HOCHSCHULE
FÜR ANGEWANDTE
WISSENSCHAFTEN
MÜNCHEN

SS 2013

Ausarbeitung zum
Hauptseminar
bei Prof. Dr. A. Nischwitz

Instant Radiosity Verfahren

Simon Pfeiffer
Matrikelnummer: 03106911
Jahnstraße 18
85521 Ottobrunn

Inhaltsverzeichnis

1 Einleitung **3**
1.1 Themengebiet . . . 3
1.2 Aufgabenstellung . . . 3

2 Grundlegende Techniken und Begriffe **4**
2.1 Bildsynthese mit 'Global Illumination'-Verfahren . . . 4
2.1.1 Global Illumination . . . 6
2.1.2 Finite Elemente . . . 9
2.1.3 Radiosity . . . 9
2.2 Erster Ansatz für interaktive 'Radiosity' . . . 14

3 Der 'Instant Radiosity' Algorithmus **15**
3.1 Grundlegende Techniken . . . 15
3.1.1 'Monte Carlo'-Verfahren . . . 15
3.1.2 Importance Sampling . . . 16
3.1.3 Low-Discrepancy Sampling . . . 17
3.1.4 Quasi-Random Walk . . . 19
3.2 Instant Radiosity . . . 20

4 Instant Radiosity Erweiterungen **23**
4.1 Grundlegende Techniken . . . 23
4.1.1 Shadow Maps . . . 23
4.1.2 Deferred Rendering . . . 25
4.2 Reflective Shadow Maps . . . 25
4.2.1 Splatting Indirect Illumination . . . 30
4.3 Imperfect Shadow Maps . . . 32
4.4 Hirearchical Image-Space Radiosity . . . 36

5 Zusammenfassung **42**
5.1 Meinung des Autors . . . 42

Literaturverzeichnis **44**

Abbildungsverzeichnis **46**

1 Einleitung

'Global Illumination' oder zu deutsch 'globale Beleuchtung' ist ein fundamentaler Bestandteil der Computergrafik. Virtuelle Bilder werden immer realistischer, können immer schneller erstellt werden und sind unser täglicher Begleiter, sei es in Werbung, in Computerspielen oder am Arbeitsplatz. Neue Techniken werden erdacht und bestehende weiterentwickelt. So rücken aufwändige Verfahren mit der Zeit und mit immer schneller werdender Grafik-Hardware in den in den Echtzeitbereich vor. 'Instant Radiosity' ist eben so ein Verfahren, das die sonst so aufwändige Berechnung einer Szene mit 'Radiosity'-Verfahren durch clever gewählte Vereinfachungen für den Einsatz in interaktiven Anwendungen ermöglicht.

Diese Arbeit entstand im Rahmen des Hauptseminars der Hochschule München im Fachbereich 'Computergraphik und Bildverarbeitung' im Sommersemester 2013 unter Leitung von Prof. Dr. Alfred Nischwitz. Das Hauptseminar wird jedes Semester für angehende Master angeboten und soll Studenten die Möglichkeit geben, das selbstständige Erarbeiten einer Problemstellung und das wissenschaftliche Schreiben zu üben, um so ideal auf die darauf folgende Master-Thesis vorbereitet zu sein. Die Themen, welche bearbeitet werden sollen, wechseln dabei von Semester zu Semester und sind stets an aktuelle Themen im Bereich Computergrafik orientiert.

1.1 Themengebiet

Das Hauptseminar des Sommersemesters 2013 umfasst die Themenschwerpunkte 'Realtime Global Illumination' und 'Shadows on Programmable Graphics Hardware'. Diese beiden Themenblöcke sind unterteilt in verschiedene Unterthemen, welche von den Studenten dann bearbeitet werden. So enthält der Block 'Realtime Global Illumination' unter anderem die Themen 'Cascaded Light Propagation Volumes', 'Voxel Cone Tracing' und 'Instant Radiosity'. Im Block der Schatten-Themen sind unter anderem 'Backprojecting Shadows', 'Shadow Volumes', 'Volumetric Shadows' und 'Thermic Shadows' zugegen. Damit werden Themen behandelt, welche vor kurzer Zeit schon eingeführt wurden (zum Beispiel 'Cascading Light Propagation Volumes' oder 'Voxel Cone Tracing') oder sich derzeit auf oder schon über der Schwelle des Einsatzes in interaktiven Anwendungen befinden (zum Beispiel 'Instant Radiosity').

1.2 Aufgabenstellung

Ziel des Hauptseminars ist, idealerweise zeitlich direkt vor dem Beginn der Master-Thesis, selbstständig eine Problemstellung zu bearbeiten und eine Ausarbeitung mit wissenschaftlichen Anspruch zu erstellen. So haben die Studenten die Möglichkeit, das wissenschaftliche Arbeiten und Schreiben zu üben, sich Kritik einzuholen und zu verbessern. Zusätzlich wird noch ein aktuelles Thema der Computergrafik bearbeitet und sowohl Studenten als auch Professor können in diesem Themenbereich auf dem neusten Stand bleiben.

2 Grundlegende Techniken und Begriffe

Dieses Kapitel dient der Heranführung an das eigentliche Thema 'Instant Radiosity' durch aufzeigen der zugrunde liegenden Techniken und Verfahren. So legt Kapitel 2.1 dar, welche Bedeutung 'Global Illumination' in der heutigen Zeit hat und Kapitel 2.1.1 zeigt wie das Verfahren grundlegend funktioniert. Ebenso wird, als eigentliche 'Global-Illumination'-Lösung, das Radiosity-Verfahren beschrieben (Kapitel 2.1.3). Das Darlegen der Vor- und Nachteile, beziehungsweise der Stärken und Schwächen, stellt die Grundlage späterer Kapitel dar, um den Vergleich erweiterter 'Instant Radiosity'-Verfahren zu dieser Referenzlösung zu ziehen.

2.1 Bildsynthese mit 'Global Illumination'-Verfahren

Bei der Erstellung virtueller Bilder spielt Realismus eine entscheidende Rolle. Das liegt in dem Umstand begründet, dass in verschiedenen Wirtschaftszweigen die Notwendigkeit besteht, etwas zu zeigen, das real nicht existiert. Und das möglichst in einer Qualität, die von einem tatsächlich aufgenommenen Bild nicht zu unterscheiden ist. Es bedarf also einem Verfahren, virtuelle Beschreibungen von Objekten in ein Bild umzuwandeln.

In den 80er und 90er Jahren erfuhr das Gebiet der realistischen Bildsynthese einen regelrechten Boom (Dutre u. a., 2003, Kapitel 'Foreword') und die zu der Zeit entwickelten Algorithmen haben selbst heute noch große Bedeutung, auch wenn diese zwischenzeitlich stark weiterentwickelt wurden. So bildeten sich in diesen Jahren sowohl der 'Raytracing'- als auch einige Jahre später der 'Radiosity'-Algorithmus und damit die Grundlage für unzählige technische Ausarbeitungen, die der realistischen Bildsynthese den Gundstein legten, der zu der heutigen Relevanz in der Wirtschaft und unserem Alltag führte (Dutre u. a., 2003, Kapitel 1.1 'What is Realistic Image Synthesis?').

Der Einfluss beschränkt sich dabei nicht nur auf offensichtliche Bereiche wie zum Beispiel 'Visual Effects' für die Filmindustrie oder Computerspiele. Es gibt viele verschiedenartige Problemstellungen, in denen virtuelle Repräsentationen von (noch) nicht existierenden Objekten herangezogen werden, um zum Beispiel subjektiv beurteilen zu können, wie ein Produkt im erdachten Einsatz aussieht. Dieser Punkt ist eine beliebte Anwendung der realistische Bildgenerierung und spielt unter anderem in folgenden Bereichen eine tragende Rolle (Dutre u. a., 2003, Kapitel 1.1 'What is Realistic Image Synthesis?') :

Simulatoren: Eine Darstellung, die möglichst mit echten Szenarios übereinstimmt, nehmen bei virtuellen Simulationen eine große Rolle ein. So soll bei diesen Anwendungen das Verhalten eines oder mehrerer Objekte in einer realen Umgebung simuliert werden. Einige dieser Betrachtungen sind rein optischer Natur, wie zum Beispiel das Testen von neu entwickelten Fahrzeugen im regulären Straßenbetrieb. Es wird versucht, diese Tests möglichst früh in der Entwicklung stattfinden zu lassen. Man ist somit auf virtuelle Simulatoren angewiesen.

Lichtdesign: Zum Überprüfen komplexer Lichtsituationen, zum Beispiel bei der Planung der Beleuchtung eines Raumes, sind hochqualitative Render-Algorithmen erforderlich. Dazu müssen idealerweise alle relevanten Lichteffekte, vor allem aber das globale Verhalten des Lichtes unter Berücksichtigung physikalischer Grundsätze simuliert werden. Auch können derart realistische Bilder oder Videos Leuchtmittelherstellern helfen ihre Produkte zu verbessern, um bestimmte Lichtstimmungen erzeugen zu können, ohne auf langwierige Prototyp-Herstellung und Tests angewiesen zu sein.

Architektur: Komplexe Lichtsituationen sind auch in der Architektur von großer Bedeutung. Hierbei ist aber nicht nur die Simulation künstlichen Lichts von Bedeutung, sondern vor allem, wie das natürliche Licht ein Gebäude sowohl von außen, als auch innen Beleuchtet. In der Dämmerung und Nacht ist das Zusammenspiel von natürlichem und künstlichem Licht interessant. So können Architekten ihre Entwürfe in vielen verschiedenen Lichtsituationen betrachten und analysieren. Auch bei der Planung von Büroräumen kann eine Lichtsimulation von Bedeutung sein, um zum Beispiel beim Einhalten von gesetzlichen Auflagen zu helfen. Virtuelle Begehungen von Gebäuden sind so ebenfalls sehr realistisch möglich.

Wie ersichtlich ist, gibt es ein breit gefächertes Gebiet, in dem globale Beleuchtung und realistische Bilderzeugung ein wichtiger Aspekt ist. Was es mit der Simulation globaler Beleuchtung auf sich hat und wie diese mit 'Radiosity' errechnet, beziehungsweise angenähert wird, zeigen die folgenden Kapitel.

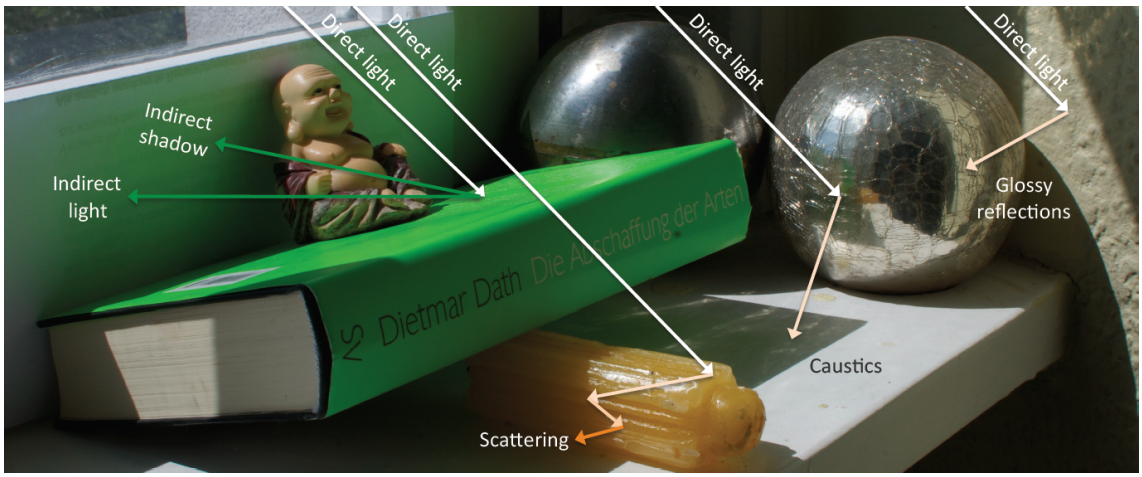

Abbildung 2.1: Fotografie einer realen Szenerie. Eingezeichnet sind die Wege verschiedener Lichtstrahlen, wie sie unterschiedliche Effekte erzeugen, die man versucht möglichst realistisch und detailgetreu nachzustellen. Diese Effekte schließen die direkte und indirekte Beleuchtung, Lichtstreuung, Reflexionen und Kaustiken ein. Abbildung übernommen von Ritschel u. a. (2012).

2.1.1 Global Illumination

Licht, ob natürlich oder künstlich, nimmt nach dem Auftreffen auf eine Oberfläche verschieden geartete Wege. Abbildung 2.1 zeigt anhand einer realen Aufnahme wie direkte Lichtstrahlung unterschiedliche Wege durch die Szenerie nehmen kann. Licht als solches wird in der Computergrafik noch hauptsächlich als Energie betrachtet, die sich Strahlenförmig von der Quelle ausbreitet, auf eine Oberfläche trifft und sich in weitere Strahlen in verschiedene Richtungen aufteilt oder absorbiert wird. Es senden somit nicht nur ausschließlich Lichtquellen Licht aus, sondern so gut wie jeder Punkt. Diese extrem komplexe Situation unterscheidet globale und lokale Beleuchtungsmodelle. Bei lokalen Modellen wird nur Licht betrachtet, welche direkt von einer Lichtquelle kommt. Globale Modelle hingegen versuchen möglichst alle Lichtanteile in einer Szene zu simulieren und deren Auswirkungen auf allen Punkten einer Szene zu errechnen.

Dies entspricht nicht exakt dem Welle-Teilchen-Dualismus. Es sind Näherungen, die vorgenommen werden um die Effekte der Lichtwellen/-teilchen und deren Wechselwirkung mit Oberflächen zu simulieren (vgl. Nischwitz u. a., 2011, Kapitel 12.1.2 'Lokale und global Beleuchtungsmodelle'). Die dadurch ignorierten minimalen Details der Beleuchtung werden toleriert, da sie für das menschliche Auge praktisch nicht wahrnehmbar sind. Anhand der Oberflächenbeschaffenheit der getroffenen Objekts entscheidet sich, ob der Lichtstrahl in das Material eintritt (transmittiert), ob es reflektiert oder absorbiert wird. In manchen Fällen ist auch sowohl Reflexion als auch ein Eintreten möglich, dann wird eintreffende Energie aufgeteilt und beide Anteile breiten sich weiterhin als Lichtstrahlen aus.

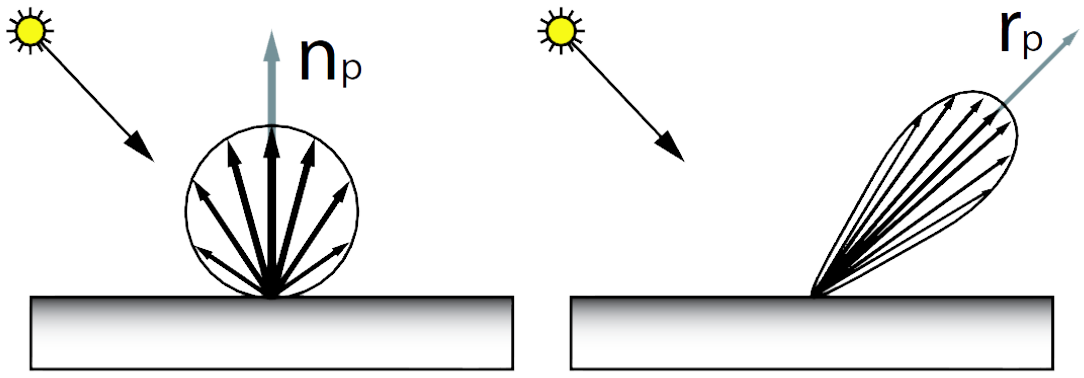

Abbildung 2.2: Diffuse und spekulare Reflexionscharakteristik im Vergleich. Bei diffuser Reflexion hat das ausgehende Licht keine Vorzugsrichtung. Spekulare dagegen ist entlang der Reflexionsrichtung r_p orientiert. Je glänzender das Material ist, desto enger liegen alle Lichtstrahlen an r_p. Abbildung übernommen von Dachsbacher u. Stamminger (2006).

Bei der Reflexion von Lichtstrahlen wird zwischen spekularer und diffuser unterschieden. Strahlen, die von einer spekularen, das heißt zu einem gewissen Grad spiegelnden, Oberfläche zurückgeworfen werden, haben eine Vorzugsrichtung, die um die Richtung des Ausfallwinkels des eintreffenden Lichts verteilt ist. Je spiegelnder das getroffene Oberflä-

chenmaterial ist, desto enger liegen die reflektierten Strahlen um diesen Ausfallvektor. Es ergibt sich die Form eines Tropfens (breite Fächerung) oder eine Zigarrenform (enge Fächerung). Ist eine Oberfläche zu keinem Anteil spiegelnd, so handelt es sich um ein rein diffuses Material. Die Reflexionsrichtungen der ausgehenden Strahlen sind gleich verteilt, ganz ohne Vorzugsrichtung und haben die Form einer perfekten Kugel, die sich über dem getroffenen Punkt auf dem Objekt aufspannt. Zusehen ist das in Bild 2.2.

Das Licht kann in verschiedenen Formen vorliegen. Die einfachste Form ist die Punktlichtquelle (vgl. Ritschel u. a., 2012, Kapitel 'Rendering equation'). Diese wird mit einer Position im Raum und einer Abstrahlcharakteristik angegeben. Sie strahlt Licht mit einem gewissen Intensitätsabfall in alle Richtungen gleichmäßig. Spotlichtquellen sind dagegen eine Erweiterung der Punktlichtquelle, in dem eine Richtung definiert wird und eine Abfall mit zunehmendem Winkel zwischen dieser Vorzugsrichtung und der Richtung des ausgesandten Lichtstrahls. So entsteht eine kreisrunde 'Öffnung' in der kugelförmigen Abstrahlung der Punktlichtquelle. Dies führt zu einem kegelförmigen Lichtaustritt mit einem, je nach gewählten Abfall, mehr oder weniger weichem Übergang zwischen hell und dunkel. Paralleles Licht ist ähnlich einfach zu berechnen. Es bezeichnet Licht, dessen Strahlen, wie der Name schon verrät, parallel ausgerichtet sind. Es muss also lediglich eine Richtung und eine Intensität angegeben werden um dieses Licht zu definieren. Die komplizierteren Lichtquellen sind Flächenlichter. So kann die Lichtquelle aus einem ebenen Rechteck bestehen, welches Licht in alle Richtungen aus der Fläche aussendet. So gibt die Normale an, aus welche Seite der Fläche Licht ausgesandt wird, um Licht nicht in alle Raumrichtungen auszusenden. Komplexer sind dagegen Lichtquellen beliebiger Form, welche wegen ihrer Flexibilität schwer zu berechnen sind.

Die Rendering-Gleichung

Die sogenannte Rendering-Gleichung, als Formel 2.1, (vgl. Pharr u. Humphreys, 2004, Kapitel 16.2.1'Basic Derivation'; Ritschel u. a., 2012, Kapitel 2.1 'Rendering Equation'; Dutre u. a., 2003, Kapitel 2.6.1 'Hemispherical Formulation'), drückt die im vorangegangenen Kapitel dargelegte Wechselwirkung des Lichts mit Oberflächen mathematisch dar. L_O ist der Lichtstrom an der Stelle einer Oberfläche x in Richtung ω.

$$L_o(x,\omega) = L_e(x,\omega) + \int_{\Omega^+} L_i(x,\omega_i) f_r(x,\omega_i \rightarrow \omega) \langle N(x),\omega_i\rangle^+ d\omega_i \tag{2.1}$$

Der Lichtstrom setzt sich zusammen aus L_e, dem in die Szene ausgesandten Lichtstrom, und dem Integral über Ω_+. Dieser bezeichnet die positive Halbkugel, die sich über dem Punkt x entlang dem Normalen $N(x)$ aufspannt. Die Bidirektionale Reflektanzverteilungsfunktion dieser Gleichung gibt der Term f_r an, $\langle\rangle^+$ bezeichnet das Skalarprodukt, welches positiv begrenzt ist. Es wird also bei der globalen Beleuchtung der Term $L_o(x,\omega)$, also der Lichtstrom auf jedem sichtbaren Punkt x einer gegeben Szene mit Materialien und Licht L_e, berechnet.

Die hemisphärische Rendering-Gleichung kann alternativ auch, statt auf die Richtungen des Lichts bezogen, auf die Oberfläche der Szene angegeben werden (Formel 2.2). Dabei wird das Integral nicht über die positive Hemisphäre, sondern über die Oberfläche der Szene aufgespannt. S entspricht nun eben jener gesamten Oberfläche.

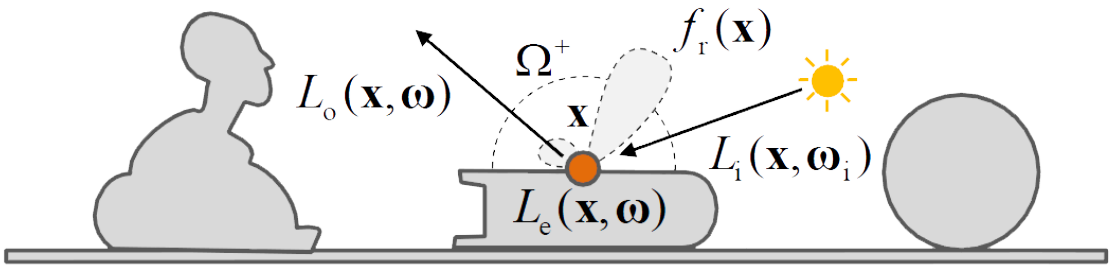

Abbildung 2.3: Die Rendering-Gleichung bildlich dargestellt. Eingezeichnet sind Hemishpäre und BRDF-Form (gestrichelt), sowie einkommendes und reflektiertes Licht (Pfeile). Abbildung übernommen von Ritschel u. a. (2012).

$$L_o(x,\omega) = L_e(x,\omega) + \int_S L_i(x,\omega_i) f_r(x,\omega_i \rightarrow \omega) \langle N(x),\omega_i\rangle^+ G(x,s)\,ds \tag{2.2}$$

Zusätzlich wird die Gleichung um einen Geometrie-Term G (Formel 2.3) erweitert. Dieser gibt die Entfernung zweier Punkte in der Szene an. Dabei wird auch die Sichtbarkeit zwischen diesen zwei Punkten errechnet:

$$G(x,s) = \frac{\langle N(s),(-\omega_i\rangle^+ V(x,s)}{\|s-x\|^2} \tag{2.3}$$

$V(x,s)$ gibt diese Sichtbarkeits-Operation an. Ist der Weg des Strahls zwischen x und s von irgendetwas versperrt, so liefert der Term null, andernfalls eins. Dieser Term wird auch als 'Ray-casting'-Term bezeichnet (Dutre u. a., 2003, Kapitel 2.6.2 'The Rendering Equation'), aus dem Grund, dass die Sichtbarkeit zweier Punkte, dadurch festgestellt werden kann, ob ein Strahl, ausgesendet von einem Punkt in Richtung des Anderen, diesen trifft oder nicht.

Die Bidirektionale Reflektanzverteilungsfunktion $f_r(x,\omega_i \rightarrow \omega)$ in den Formeln 2.1 und 2.2, abgekürzt auch BRDF ('Bi-directional Reflectance Distribution Function'), ist eine 4D-Funktion (Ritschel u. a., 2012, Kapitel 2.1 'Rendering Equation') welche angibt, wie Licht von einer Oberfläche reflektiert wird. Eintreffendes Licht aus Richtung ω_i wird zu einem gewissen Anteil in Richtung ω_o zurückgeworfen. Ergebnis einer berechneten BRDF ist, wie groß der Anteil ist. So sind Materialbeschreibungen möglich, deren Reflexionsverhalten stark winkelabhängig ist. Und das nicht nur längs um die Oberflächennormale, sondern auch quer dazu.

Ist die Reflexionscharakteristik auf einer Oberfläche nicht überall gleich, sie ändert sich also von Punkt x zu Punkt x', so spricht man von einer BTF ('Bi-directional Texture Function' oder Bidirektionale Texturfunktion). Das bedeutet, dass für jeden Punkt x auf einem Objekt eine eigene BRDF hinterlegt ist, was den Speicheraufwand stark in die Höhe treibt. Kann Licht nicht nur Reflektiert werden, sondern auch in das Material eintreten,

so benötigt man zur Berechnung statt einer BRDF oder BTF eine sogenannte BSDF (Bidirectional Scattering Distribution Function oder Bidirektionale Zerstreuungsverteilungsfunktion). Zu diesem Zweck wird die in der BRDF heran genommene Hemisphäre zu einer vollständigen Kugel erweitert. So ist es möglich, sowohl reflektierte als auch transmittierte Strahlverteilungen angeben.

2.1.2 Finite Elemente

Bevor 'Radiosity' als Lösungsverfahren für globale Beleuchtung betrachtet wird, dient dieser Einschub zur Darlegung der Methode der finiten Elemente abseits der Computergrafik. Dieses Verfahren wird auch dazu benutzt um Beleuchtung zu errechnen (Dutre u. a., 2003, Kapitel 'Stochasitc Radiosity'), ist im technischen Umfeld aber eher dafür bekannt, zum Beispiel für Belastungs- oder Thermodynamiksimulation benutzt zu werden. Dazu wird, stark vereinfacht ausgedrückt, das Objekt, an dem die Simulation stattfinden soll in eine hohe Anzahl an Elementen eingeteilt. Da diese festgelegte Anzahl an Elementen nicht unendlich oder flexibel ist (und damit finit), ist das Objekt in seiner Form oder Volumen diskretisiert. Dies ermöglicht, komplexe Integralberechnungen durchzuführen in dem das eigentliche Problem auf einfache, nur ein Element betreffende Probleme herunter gebrochen wird. Es wird dann in mehreren Iterationen die Auswirkungen der Änderung eines Elements auf seine Umliegenden simuliert bis ein Abbruchkriterium erreicht ist.

Betrachtet man nun eine 3D-Beschreibung einer Szenerie, fällt auf, dass man die Oberfläche dieser Szene auch per finite Elemente Methoden in 'Patches' einteilen kann und dann basierend auf diesen 'Patches' eine Beleuchtungsrechnung durchführen kann. Aus diesem Gedanken ergibt sich das Lösungsverfahren für globale Beleuchtung namens 'Radiosity'.

2.1.3 Radiosity

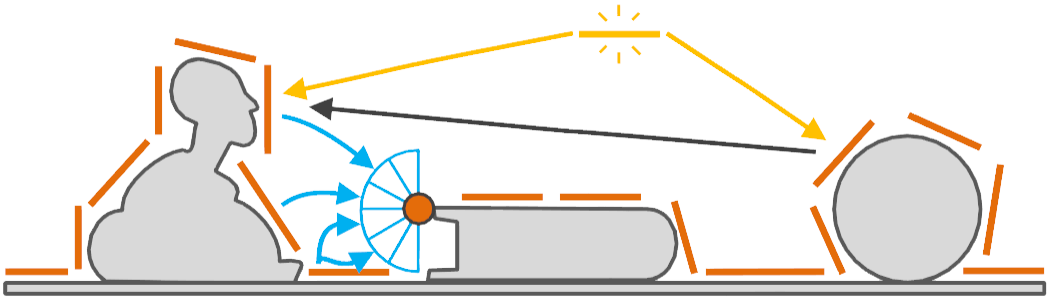

Abbildung 2.4: Das 'Radiosity'-Prinzip bildlich dargestellt. Zu sehen sind die grob aufgelösten Patches der Szenengeometrie (orange) und das direkte Licht (gelbe Pfeile). Zu berechnen sind die Formfaktoren (schwarzer Pfeil) und wieviel indirektes Licht auf einen beliebigen Punkt in der Szene reflektiert wird (blaue Pfeile). Abbildung übernommen von Ritschel u. a. (2012).

Dieses Kapitel beschreibt das globale Beleuchtungs-Lösungverfahren 'Radiosity'. Der Name 'Radiosity' kommt von der radiometrischen Größe, welche in der globalen Beleuchtung errechnet werden will. Diese Technik wurde 1984 von Goral u. a. entwickelt (vgl. Dutre u. a., 2003, Kapitel 'History of Photorealistic Rendering'), nur wenige Jahre nachdem mit 'Raytracing' ein erstes robustes Verfahren gefunden wurde um komplexe Lichtsituationen zu simulieren. Da es zu dieser Zeit aber noch kein ideales Verfahren ist um Effekte wie diffuse Reflexion oder 'Color-Bleeding' (naheliegende Flächen strahlen einen Teil des eintreffenden farbigen Lichts auf das beobachtete Objekt) zu simulieren, wurde zur Hilfe ein neues Renderingverfahren entwickelt. 'Radiosity' ist ein finite Elemente Verfahren bei dem die gesamte Oberfläche der Szenerie diskretisiert, in sogenannte 'Patches' eingeteilt wird, und zwischen allen Patch-Kombinationen sogenannte Formfaktoren errechnet und gespeichert werden. Dadurch wird in einem globalen Fokus der Lichtfluss und damit auch die Sichtbarkeit jedes möglichen Punktepaares errechnet. Sichtbarkeit ist ein Stichwort, das in Kapitel 2.1.1 schon gefallen ist. In Gleichung 2.2 und dessen Geometrie-Term 2.3 ist eben jene Punkt-zu-Punkt Sichtbarkeit mit der Funktion $V(x, s)$ beinhaltet.

So kann aus der Flächenbasierten Rendering-Gleichung 2.2 die Formel 2.4 (vgl. Dutre u. a., 2003, Kapitel 6.1 'Classic Radiosity') aufgestellt werden. Wobei angenommen wird, dass ausschließlich diffuse Reflexionen berücksichtigt werden sollen, was den Entfall der Richtungsvariablen nach sich zieht.

$$L(x) \;=\; L_e(x) + p(x) \int_S L_i(s) \langle N(x), \omega_i \rangle^+ G(x, s)\, dA_s \tag{2.4}$$

Darin bezeichnet $p(x)$ den Reflexionsfaktor im Punkt x. Ersetzt man nun, wie Dutre u. a. zeigt, das Licht mit der Radianz, also $B(x) = \pi L(x)$ und $B_e(x) = \pi L_e(x)$ und multipliziert mit π, entsteht aus der generellen Rendering-Gleichung die 'Radiosity Integralgleichung' 2.5.

$$B(x) \;=\; B_e(x) + p(x) \int_S B_i(s) \langle N(x), \omega_i \rangle^+ G(x, s)\, dA_s \tag{2.5}$$

Zieht man die in Patches eingeteilte Szene in Betracht, so gibt Dutre u. a. weiterhin an, dass sich der Lichtfluss durch Patch i folgendermaßen angegeben werden kann:

$$B_i \;=\; \frac{1}{A_i} \int_{S_i} \int_{\Omega^+} L(x, \omega_i) \langle N(x), \omega_i \rangle \, d\omega_i \, dA_x \tag{2.6}$$

Nimmt man nun die in Formel 2.5 angewandten Einschränkung vor, so ergibt sich eine stark vereinfachte Darstellung des Integrals:

$$B_i \;=\; \frac{1}{A_i} \int_{S_i} B(x)\, dA_x \tag{2.7}$$

Der Lichtfluss durch Patch i entspricht also dem Lichtfluss der gesamten restlichen Szene integriert über die Fläche von i.

Wird nun weiterhin angenommen, dass die Radianz $B(x)$ auf jedem Patch i konstant ist und auch der Reflexionsfaktor auf jeden Patch konstant bleibt, so folgt aus den Formeln

2.5 und 2.7 das klassische Radiosity-Gleichungssystem, mit welchem die Integral-Gleichung angenähert werden kann. Diese hat den Vorteil, dass das Integral über die Fläche aller Patches durch die Summe der Formfaktoren ersetzt wurde.

$$B'_i = B_{ei} + p_i \sum_j F_{ij} B'_j \tag{2.8}$$

Mit

$$F_{ij} = \frac{1}{A_i} \int_{S_i} \int_{S_j} \langle N(x), \omega_i \rangle^+ G(x,s) dA_x dA_s \tag{2.9}$$

werden die Formfaktoren errechnet. Dies sind 'nicht-triviale vierdimensionale Integrale' (Dutre u. a., 2003, Kaptiel 6.1.2 'Mathematical Problem Description') und damit schwer zu lösen.

Dutre u. a. fasst treffend zusammen, dass das 'Radiosity'-Verfahren aus 4 Schritten besteht:

1. Szene disretisieren und in Patches einteilen
2. Errechnen der Formfaktoren
3. Numerisches Lösen des Radiosity-Gleichungssystem
4. Anzeigen der Lösung

Im weiteren Verlauf dieses Kapitels wird nun kurz auf die Einteilung der Szene in Patches und die darauf berechneten Formfaktoren eingegangen. Anschließend werden die Vorteile aber vor allem auch die Probleme des Verfahrens dargelegt und mit dem klassischen 'Ray-Tracing' Algorithmus verglichen um den Stellenwert des Verfahrens einordnen zu können.

Patches

Die in Radiosity-Verfahren verwendeten Elemente zur Diskretisierung der Szene werden 'Patches' genannt. Sie dienen der Vereinfachung der vorhandenen Geometrie in eine begrenzte Anzahl an planaren Primitiven. Damit wird erreicht, dass nicht mehr ein Integral über die gesamte, beliebig geformte, Oberfläche gelöst werden muss und für jedes mögliche Punktepaar der unendlich vielen Punkte Formfaktoren errechnet werden müssen. Stattdessen muss nur noch der Formfaktor für jedes Patch-Paar iterativ errechnet werden (siehe Formel 2.7 zu Formel 2.8).

Als Patches werden meist Dreiecke verwendet, wobei auch konvexe Vierecke und quadratische Patches Verwendung finden (Dutre u. a., 2003, Kapitel 6.1 'Classic Radiosity'). Das kommt der Computergrafik natürlich entgegen, da als Geometrie-Beschreibung in den meisten Fällen eine Dreiecks- oder Viereckrepräsentation verwendet wird. Diese Geometrie-Triangles können ohne weiteres auch als 'Radiosity'-Patches benutzt werden.

Formfaktoren

Die sogenannten Formfaktoren geben an, wie sich der Lichtfluss zwischen zwei Patches verhält. Formel 2.9 zeigt, dass hierzu ein Doppelintegral über die Flächen von den Patches i und j gelöst werden muss. Sie gibt an, zu welchen Bruchteil von Patch i aus der Patch j sichtbar ist und umgekehrt. Nach Gleichung 2.8 setzt sich so die Radianz B_i aus dem von Patch i emittierten Lichtanteil, sowie der Summe aller Lichtbeiträge der restlichen Patches, abgeschwächt um den gemeinsamen Formfaktor F_{ij}. Das Ergebnis F_{ij} der Formfaktoren muss gewissen Anforderungen genügen (vgl. Dutre u. a., 2003, Kapitel 6.2.1 'Properties of the Form Factors'):

1. Alle Formfaktoren zwischen Patches sind positiv oder null. Null gibt dabei an, dass zwei Patches gegenseitig nicht sichtbar sind.

2. Die Formfaktoren zwischen Patch i und allen anderen Patches ergibt in einer geschlossenen Szene (kein Licht kann entkommen) eins. Ist die Szene nicht geschlossen, so ist ein Ergebnis kleiner eins möglich. Das nicht aufgefangene Licht entkommt so in die Unendlichkeit des Raumes.

Die Formfaktoren wurden vor allem in der Anfangszeit von 'Radiosity' durch zwei unterschiedliche Techniken errechnet, um den nicht-trivialen Integral vierter Dimension nicht numerisch lösen zu müssen. Es handelt sich bei diesen Herangehensweisen um Annäherungen des Formfaktors. Die klassische Berechnung per Hemi-Sphere- oder Hemi-Cube-Verfahren wurde später von Local- und Global Line-Algorithmen abgelöst.

Ursprüngliche Verfahren: Beim Hemi-Sphere- und -Cube-Verfahren wird die Fläche des Patches j betrachtet, welche von einem Punkt x in Richtung Patch i auf eine x umgebende Halbkugel oder einen halben Würfel projiziert wird.

Local Lines: Es wird angenommen (Dutre u. a., 2003, Kapitel 6.2.3 'Form Factor Sampling Using Local Lines'), dass die Annäherung des Formfaktor durch eine Partikelsimulation erreichen lässt. Dazu wird eine gewisse Anzahl an Partikeln N_i, welche sich wie Photonen verhalten, von Patch i aus in die Szene geschossen. Es wird eine uniforme Kosinus-Verteilung um die Oberflächennormale verwendet. Entsprechend der Anzahl an Partikeln, welche auf Patch j auftrifft, ergibt sich der Formfaktor $F_{ij} \approx N_i/N_{ij}$

Global Lines: Anstatt diese eben beschriebenen lokalen Sichtlinien ausgehend von einem Patch zu errechnen, ist es auch möglich, die Problemstellung global anzugehen (vgl. Dutre u. a., 2003, Kapitel 6.2.4 'Form Factor Sampling Using Global Lines). Dazu wird zum Beispiel auf einer, die Szene umspannenden, Kugel Punkte uniform verteilt und zu Geraden verbunden. Generell schneidet eine solche Linie mehrere Patches und wird durch die Schnittpunkte in gewisse 'Spannen' eingeteilt, welche die gegenseitige Sichtbarkeit zwischen zwei geschnitten Patches darstellen. Der Vorteil gegenüber den Local Lines ist, dass die Anzahl der redundanten Linien reduziert wird. So sind nicht mehr zwei Linien nötig um gegenseitige Sichtbarkeit auszudrücken (Linie von Patch i zu j und eine weitere Linie von Patch j zu i), sondern nur noch eine. Die Wahrscheinlichkeit, dass ein Patch von einer globalen Linie geschnitten wird, entspricht dem Verhältnis der Fläche des Patches zu der Gesamtfläche, sowie der Anzahl der erstellten Linien.

Vorteile und Probleme

Im vorherigen Kapitel wurden schon die Lösungen eines großen Problems von 'Radiosity' aufgezeigt. Das Lösen des nicht-trivialen Doppelintegrals durch Projizieren auf Hilfsgeometrie oder das Annähern durch Partikelsimulation.

Das 'Radiosity'-Verfahren stellte noch weitere Herausforderungen an die Forscher, da es naturgemäß keine perfektes ist. Entsprechend der vier, im Verlauf dieses Kapitels aufgeführten, Schritte von 'Radiosity' lassen sich diese weitere Problempunkte erörtern, welche die Berechnung von Bildern aufwändig machen. Da sich das 'Radiosity'-Gleichungssystem gutartig verhält (Dutre u. a., 2003, Kapitel 6.1.4 'Problems') kann schon mit wenigen Iterationen eine ansprechende Lösung gefunden werden, so beschränken sich die Problempunkte auf die beiden ersten Schritte:

Szene disretisieren und in Patches einteilen: Die gesamte Geometrie einer Szene muss, um die Beleuchtung mit 'Radiosity' berechnen zu können, diskretisiert werden. Das heißt, alle Oberflächen müssen durch planare Primitive angenähert werden. Liegt die Szene schon tesseliert vor, so kann es vorkommen, dass diese trotzdem noch einmal in gröberer Auflösung angenähert werden muss, denn die Anzahl der Patches in einer Szene beeinflusst, unter Verwendung des klassischen Algorithmus, die Render-Geschwindigkeit massiv.

Patches müssen in ausreichender Zahl vorliegen, um die Szene ausreichend genau zu repräsentieren. Jeder Patch muss dabei klein genug sein, um auch feine Details der Beleuchtung befriedigend detailliert wiedergeben zu können, da die 'Radiosity'-Gleichung die Einschränkung beinhaltet, dass die Radianz über einen Patch konstant ist. Feine Lichtdetails, die wesentliche kleiner sind als der Patch an der entsprechenden Stelle, werden somit 'verschluckt'. Es gilt also einen Kompromiss zu finden zwischen der Gesamtzahl der Patches und der Qualität, um ein ansprechendes Ergebnis in akzeptabler Zeit zu erlangen. Wird eine zu große Menge verwendet, so schlägt sich das sowohl auf die benötigte Zeit, als auch auf die mindestens zur Verfügung stehende Speicherkapazität aus. Denn die Patch-Repräsentation muss parallel zur Szenengeometrie im Arbeitsspeicher vorliegen (sofern nicht die originale Tesselierung der Geometrie als Patches verwendet wird).

Errechnen der Formfaktoren: Anhand der Menge der Patches für eine virtuelle Umgebung lassen sich Rückschlüsse auf die Anzahl der zu berechnenden und zu speichernden Formfaktoren bilden. Nimmt man für eine einfache Szene an, dass diese aus 50.000 Patches besteht und, um die 'Radiosity'-Gleichung zu erfüllen, für jeden Patch auch jeder weitere betrachtet werden muss, so ergibt sich eine Anzahl von 2.500.000.000 Formfaktoren. Das führt vor Augen in welcher Dimension Speicher und Rechenkapazität zur Verfügung gestellt werden muss. Dass bei komplexen Szenen durchaus 100.000 oder gar mehr Patches verwendet werden (Dutre u. a., 2003, Kapitel 6.1.4. 'Problems') verstärkt die Notwendigkeit von Algorithmen, welche die Komplexität der Formfaktorberechnung ($O(n^2)$) reduzieren.

Um jene angesprochene quadratische Komplexität abzumildern, wurden unter Verwendung von 'Monte Carlo'-Verfahren Wege gefunden die Anzahl der errechneten Formfaktoren auf ein einfacher zu errechnendes Maß zu senken. So wird zum Beispiel bei 'Stochastic Relaxation Radiosity' (Dutre u. a., 2003, Kapitel 6.3 'Stochastic Relaxation Radiosity'), mit Hilfe

des Jacobi-Verfahrens zum Lösen linearer Gleichungssysteme in der Form von $Ax = b$, die 'Radiosity'-Gleichung angenähert. Mit jeder Jacobi-Iteration wird mit dieser Herangehensweise ein sogenannter 'Bounce' der Beleuchtung errechnet. Ein 'Bounce' bezeichnet die einfache Abstrahlung von Licht, komme dieses Licht nun von eine Lichtquelle oder sei es von einer beliebigen Oberfläche reflektiert. In der ersten Iteration bekommen alle Patches direktes Licht zugerechnet, in jeder Nachfolgenden wird für jeden Patch indirektes Licht aller Anderen gesammelt und aufaddiert (sogenanntes 'Gathering') oder das vorhandene Licht eines Patches auf alle Anderen der Szene verteilt (sogenanntes 'Shooting'). Bei 'Discrete Random Walk Radiosity' (Dutre u. a., 2003, Kapitel 6.4 'Discrete Random Walk Methods for Radiosity') wird dagegen der Weg von Photonen durch die Szene anhand von Wahrscheinlichkeiten verfolgt. Mehr dazu gibt es in Kapitel 3.1.4 in einem leicht abgeänderten Verfahren.

Gerade in den ersten Jahren der Existenz von 'Radiosity', war es das bestmögliche Verfahren um diffuse Abstrahlung von Licht zu berechnen. Da es zu dieser Zeit (1984) weder 'Path-Tracing' noch 'Photon-Mapping' zur Simulation globaler Beleuchtung gab, war 'Radiosity' nahezu konkurrenzlos in diesem Bereich. Was im klassischen Algorithmus wiederum nicht simuliert wird, ist die spekulare Reflexion und Refraktion, wo 'Ray-Tracing' realistische Ergebnisse liefern kann. So beschränkten sich beide Algorithmen zunächst auf jene Problemstellungen, für welche sie entwickelt wurden. Da sie sich allerdings ideal ergänzen und 'Radiosity' zuerst relevante Information errechnet und speichert, ist es möglich, beide Verfahren miteinander zu kombinieren indem zuerst per 'Radiosity' die globale diffuse Reflexion errechnet wird und in einem zweiten Render-Durchlauf diese, wenn relevant, zusätzlich aufgesammelt.

2.2 Erster Ansatz für interaktive 'Radiosity'

Bevor es möglich war, 'Radiosity' auf Grafik-Hardware zu berechnen, gab es Techniken die dadurch simulierte globale Beleuchtung in Echtzeit-Anwendungen darstellen. Es wird sich dabei dem Umstand bedient, dass 'Radiosity' zunächst einmal die Beleuchtung errechnet und erst, wenn dieser Schritt abgeschlossen ist, diese auf der originalen Szenengeometrie dargestellt werden kann. So ist es möglich, sogenannte 'interaktive Begehungen' anzubieten, welche den Algorithmus vor allem in der Architektur beliebt und verbreitet gemacht haben (vgl. Dutre u. a., 2003, Kapitel 1.1.1 'The Importance of Realistic Image Synthesis'). Dazu wird ein Echtzeit-Renderingverfahren verwendet, welches, statt die Beleuchtung selber zu errechnen, sich der von 'Radiosity' simulierten globalen Beleuchtung bedient.

Was getan werden muss, um 'Radiosity' in interaktiven Bildwiederholraten auch auf Grafik-Hardware zu berechnen, zeigt das folgende Kapitel 3.

3 Der 'Instant Radiosity' Algorithmus

1997 veröffentlichte Alexander Keller eine Arbeit, in der Wege aufgezeigt werden, wie die aufwändige Berechnung für globale Beleuchtung mittels 'Radiosity' durch clever gewählte Vereinfachungen stark beschleunigt werden kann. Ebenso zeigt Keller Techniken auf, die es ermöglichen, einen Teil dieser Berechnung direkt auf Grafik-Hardware auszuführen. Bevor dieser Algorithmus detailliert betrachtet werden kann, müssen zuerst weitere Techniken betrachtet werden, die zum Verständnis grundlegend sind.

3.1 Grundlegende Techniken

3.1.1 'Monte Carlo'-Verfahren

Unter 'Monte Carlo' werden Verfahren verstanden, die zum Beispiel Gleichungssysteme statt mit klassischen Methoden numerisch zu lösen, durch Verwendung vieler Zufallswerte anzunähern. Grundsätzlich kann mal solche Algorithmen in zwei Kategorien einteilen. Neben der hier beschriebenen 'Monte Carlo'-Klassifizierung existieren noch die sogenannten 'Las Vegas'-Algorithmen (Pharr u. Humphreys, 2004, Kapitel 14 'Monte Carlo Integration I: Basic Concepts'). 'Las Vegas'-Verfahren ergeben trotz der Verwendung von Zufallszahlen immer das gleiche Ergebnis, wohingegen 'Monte Carlo'-Verfahren die richtige Lösung immer nur annähern und im Durchschnitt aller Lösungen auf dem korrekten Wert liegen. Für ein korrektes Ergebnis vorausgesetzt ist eine Mindestzahl an betrachteten Zufallswerten. 'Monte Carlo' wird mit steigender Zahl der Werte zunehmend genauer.

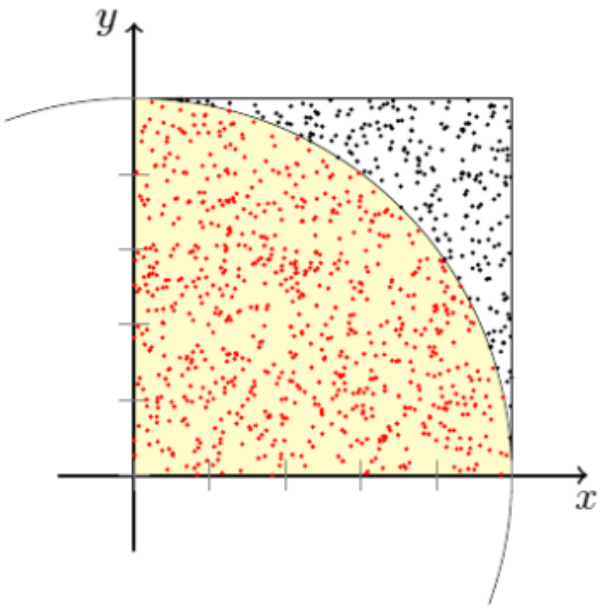

Abbildung 3.1: Will man π mit 'Monte Carlo'-Verfahren annähern, so reicht es, ein Quadrat (beispielsweise mit Seitenlänge eins) mit zufällig Verteilten Punkten zu füllen. Zieht man einen Kreis um den Ursprung, entspricht das Verhältnis der Anzahl der Punkte im Kreis zu der Gesamtzahl ungefähr $\frac{\pi}{4}$. Abbildung übernommen von `http://de.wikipedia.org/wiki/Monte-Carlo-Simulation` (Aufgerufen am 20.6.2013).

Ein Vorteil des Verfahrens liegt darin, dass es sich auf eine breitgefächerte Menge an Problemen anwenden lässt. Es lässt sich damit beispielsweise der Wert der Kreiszahl π ermitteln (Bild 3.1). Für die Bildverarbeitung und vor allem -generierung wichtig, ist die Eigenschaft, dass sich mit 'Monte Carlo' auch höherdimensionale Integrale über komplizierte oder gar nicht stetige Funktionen lösen lassen, welche mit klassischen Mitten nur schwer bis gar nicht lösbar sind (Dutre u. a., 2003, Kapitel 3.2 'Why Are Monte Carlo Techniques Useful?'). Dutre u. a. führt weiterhin den Nachteil der Verfahren an, dass die Konvergenzgeschwindigkeit vergleichsweise niedrig ist. Diese liegt im klassischen Ansatz bei $\frac{1}{\sqrt{N}}$ (N gibt hierbei die Anzahl der verwendeten Zufallswerte an) und ist für simple Probleme nicht optimal, da es Verfahren gibt, die es auf andere Weise schneller lösen können. Um ein Integral (3.1) (vgl. Dutre u. a., 2003, Kapitel 3.4.2 'Estimator') zu lösen beziehungsweise mit 'Monte Carlo' anzunähern wird dieser an zufällig ausgewählten Stellen geprüft.

$$I = \int f(x)\,dx \tag{3.1}$$

Um die Abweichung des Ergebnisses auch bei nicht optimaler, aber ausreichender Menge, an Samples zu reduzieren, werden diese nicht uniform über den Bereich des Integrals verteilt. Stattdessen wird die Menge der Samples entsprechend einer Wahrscheinlichkeitsfunktion wie zum Beispiel dem 'Importance Sampling' verteilt. Man nützt dabei aus, dass man diese Funktion so beeinflussen kann, dass die Verteilung ein für das Problem vorteilhaftes Verhalten ergibt. Diese Wahrscheinlichkeitsfunktion an der Stelle i bezeichnet in Formel 3.2 der Term $p(x_i)$.

$$\langle I \rangle = \frac{1}{N} \sum_{i=1}^{N} \frac{f(x_i)}{p(x_i)} \tag{3.2}$$

$\langle I \rangle$ ist der sogenannte 'Estimator' von I (Gleichung 3.1), also das angenäherte Ergebnis des Integrals über die Funktion $f(x)$. Gebildet wird die normalisierte Summe über alle Samples (N mal). Für jede Iteration werden sowohl die Funktion $f()$ als auch die Wahrscheinlichkeitsfunktion $p()$ an der Stelle x_i evaluiert. Für das Ergebnis gilt: $\langle I \rangle \approx I$. Für unendlich viele Samples gilt $\langle I \rangle = I$.

Eine Verbesserung der 'Monte Carlo'-Verfahren bildet das 'Russian Roulette'. Damit werden Samples verworfen, die nicht signifikant genug (festzustellen durch einen Vergleich gegen einen vordefinierten Grenzwert) zum Endergebnis beitragen. Stattdessen wird ein vordefinierter Wert als Ersatz verwendet. So werden nur für das Ergebnis relevante Samples berechnet, was Rechenzeit spart und das Ergebnis nicht beeinflusst, sofern der Grenzwert entsprechend gesetzt ist.

3.1.2 Importance Sampling

'Importance Sampling' dient dazu, in 'Monte Carlo'-Verfahren die Samples zu verwenden, welche für das das Endergebnis wichtiger sind als andere. Man versucht daher im 'Estimator' (Formel 3.2) die Funktion $p(x)$ möglichst ähnlich zu der Funktion $f(x)$ zu wählen. Werden zum Beispiel in der Beleuchtungsrechnung in gewissen Bereichen feine Details, wie Kaustiken oder Ähnliches, erwartet, sollte das Integral I in diesem Bereich auch stärker

abgetastet werden um diese Details auch erfolgreich zu erfassen. Wird das 'Importance Sampling' überreizt, so kann es passieren, dass Artefakte auftreten, weil andere Bereiche zu wenig abgetastet werden.

Ein Beispiel für 'Importance-Sampling' ist das 'Hierarchical Warping'. Dabei wird die Fläche, auf der die Samples verteilt werden sollen, in vier gleich große Abschnitte eingeteilt und entsprechend eines festgelegten Kriteriums analysiert. Für jeden Bereich muss sich ein Prozentwert ergeben, der die Verteilung des Kriteriumwertes auf der Fläche widerspiegelt. Nun wird eine Zufallsverteilung von Punkten entsprechend der analysierten Verteilung eingeteilt. Man geht dabei so vor, dass man zuerst die zwei oberen und die zwei unteren Bereiche addiert und die Zufallsverteilung dann quer entsprechend der Prozentwerte einteilt (in Bild 3.2 im Verhältnis 80 zu 40). Danach normalisiert man beide Bereiche der Zufallsverteilung, sodass beide Bereiche gleich groß sind aber der obere Teil etwa 80% der Samples beinhaltet. Danach teilt man die beiden normalisierten Abschnitte in gleicher weise auf und hat nach diesem Schritt eine Verteilung der Samples entsprechend des Werts des Analysekriteriums. Dieser Schritt lässt sich beliebig oft wiederholen um eine gewünschte Genauigkeit zu erhalten.

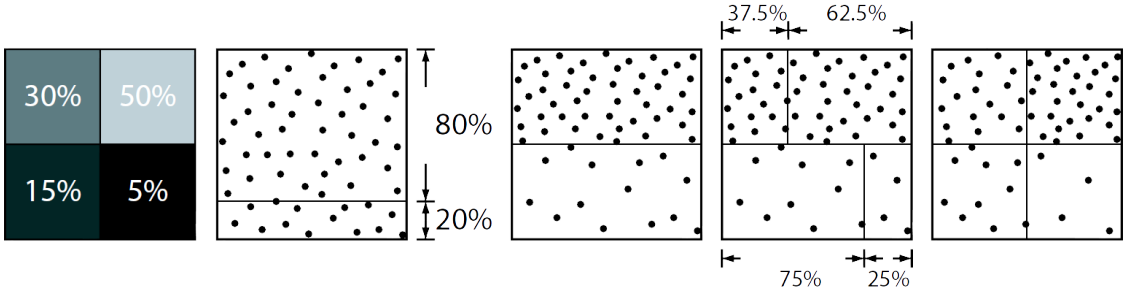

Abbildung 3.2: Es wird eine Sample-Verteilung gewünscht, sodass die vier Quadranten 30%, 50%, 15% und fünf % der verfügbaren Samples enthalten. Durch Verschieben der Samples gemäß der gewünschten Verteilung wird eine solche Zufallsverteilung erstellt. Abbildung übernommen von Dachsbacher u. Stamminger (2006).

3.1.3 Low-Discrepancy Sampling

Um Zufallswerte zu generieren, gibt es Verfahren wie pseudo-Zufallszahlen, welche zum Beispiel aus der zum Berechnungszeitpunkt aktuellen Uhrzeit errechnet werden. Meist erhält man dadurch Werte zwischen null und eins. Für Algorithmen wie 'Monte Carlo' werden aber über einen Bereich oder einer Fläche verteilte Zufallswerte benötigt. Alle Beispiele in diesem Kapitel betrachten, analog zu Pharr u. Humphreys (Kapitel 7.4 'Low-Discrepancy Sampling'), Zufallsverteilungen in einem Quadrat mit Seitenlänge eins und einem Wertebereich von null bis eins. Verteilt man Samples in so einem Quadrat naiv durch zwei zufällige Koordinaten x und y, so erhält man eine teils sehr unvorteilhafte Zufallsverteilung (siehe 3.3 oben links). Samples häufen sich an gewissen Stellen, während ganze Flächen nicht ein Sample enthalten. Abhilfe schafft da sogenanntes 'jittered sampling', wobei das Sampling-Quadrat in so viele kleinere Quadrate aufgeteilt wird wie es

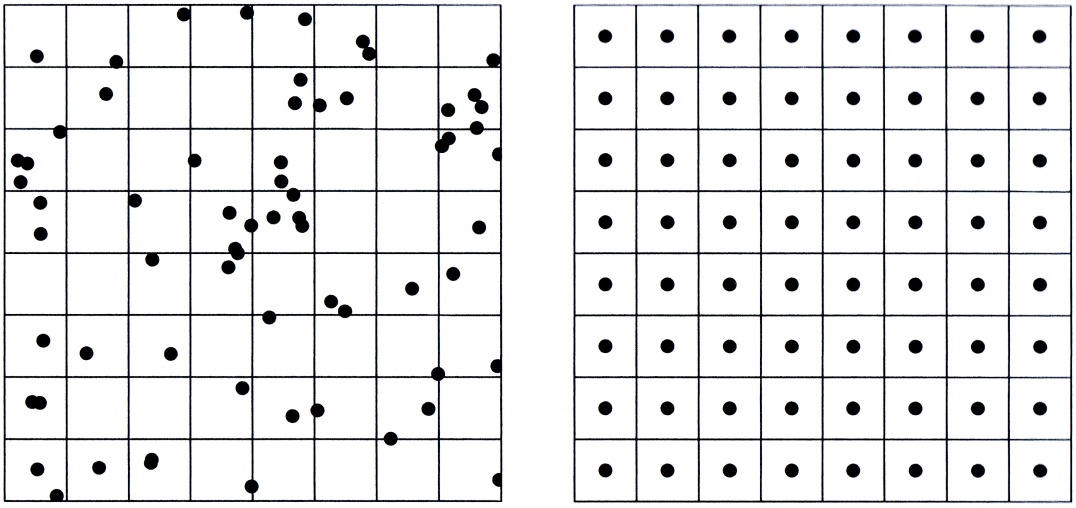

Abbildung 3.3: Zwei nicht ideale Sampling-Verfahren. Links ein rein zufällig verteiltes Muster, rechts strikt uniform verteilt. Uniforme Verteilungen sind in der Computergrafik nicht erwünscht, da sie dazu neigen, hoch-regelmäßige Aliasing-Effekte zu erzeugen, wogegen rein zufällige Verteilungen zu große Flächen unbesetzt lassen und sich an anderen Stellen häufen, was das Rauschen verstärkt. Abbildung übernommen von Pharr u. Humphreys (2004).

Samples geben soll. Jedes dieser unter-Quadrate enthält dann nur noch ein darin zufällig positioniertes Sample, was dazu führt, dass die Abstände zwischen den Zufallswerten gleichmäßiger verteilt sind und eine Häufung von mehr als vier Samples unmöglich wird.

Die Gleichmäßigkeit der Verteilung ist ein Ziel der Sampling-Techniken ohne aber die Zufälligkeit zu verlieren. Denn zu uniform verteilte Samples sind anfällig für Aliasing-Artefakte. Die Abweichung zwischen Zufallsverteilung und uniformer Verteilung wird 'Diskrepanz' genannt. Diese muss also zwischen einem Mindest- und einem Maximalwert liegen, vorzugsweise nahe dem Mindestwert. Berechnen kann man den Diskrepanz-Wert einer beliebigen Form auf dem Einheitsquadrat beispielsweise (in **??** veranschaulicht) indem man prüft, wie viele Samples innerhalb dieser Form liegen.

Für 'Instant Radiosity'-Verfahren wichtig ist, dass man beispielsweise mit der Halton-Sequenz eine Sample-Verteilung mit niedriger Diskrepanz erstellen kann. Wobei es sich damit nicht mehr um zufällig platzierte Werte handelt, sondern lediglich um eine Verteilung mit niedriger Diskrepanz. Dazu wird sich der 'Radical Inverse' bedient. Diese Technik basiert, wie Pharr u. Humphreys in Kapitel 7.4.2 'Contructing Low-Doscrepancy Sequences' schreibt, auf dem Fakt, dass sich jeder positive Integer-Wert n als eine Basis b und einer Sequenz an Ziffern $d_m \dots d_2 d_1$ eindeutig durch

$$n = \sum_{i=1}^{\infty} d_i b^{i-1} \tag{3.3}$$

ausdrücken lässt. Die 'Radical Inverse'-Funktion Φ_b zur Basis b rechnet einen positiven Integer-Wert in einen Floating-Point-Wert zwischen null und eins um, indem sie deren

Binärwert um den Dezimalpunkt spiegelt.

$$\Phi_b(n) \;=\; 0.d_1 d_2 \ldots d_m \tag{3.4}$$

Für eine Halton-Sequenz in n Dimensionen wird für jede Dimension eine andere Basis b der 'Radical Inverse' verwendet, wobei die verwendeten Basen gegenseitig unteilbar sein, weswegen gerne die ersten n Primzahlen verwendet werden. Der Algorithmus teilt dann Dimension eins in $b_1 = 2$ Teile, Dimension zwei in $b_2 = 3$ Teile, ..., Dimension n in b_n Teile und kaskadiert so den n-dimensionalen Raum. Die Grenzen der Unterteilungen liefern dann die Koordinaten (z.B im 2D-Fall liegt der erste Punkt in $(^1/_2\ ,\ ^1/_3)$). Da die Samples nicht mehr zufällig, aber weiterhin nicht uniform verteilt sind, spricht man nun mehr von einer Quasi-Zufallsverteilung. 'Monte-Carlo'-Verfahren, welche statt echten Zufallswerten Quasi-Zufallsverteilungen benutzen bezeichnet man als 'Quasi-Monte-Carlo'-Verfahren.

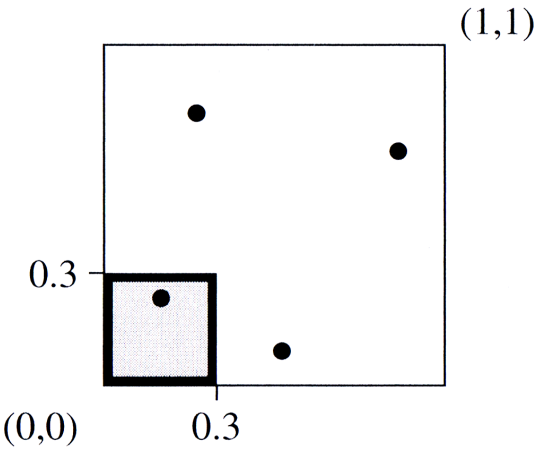

Abbildung 3.4: Um die Diskrepanz einer Sample-Verteilung und einer zu überprüfenden beliebigen Form zu errechnen, prüft man, wie viele Samples (n) der gesamten Samples (N) genau in dieser Fläche liegen. So zieht man von $\frac{n}{N}$ den Flächeninhalt der betrachteten Form ab. Der größte gefundene Wert aller möglichen oder festgelegten Formen ist die Diskrepanz. Grafik übernommen von Pharr u. Humphreys, 2004, Kapitel 7.4.1 'Definition of Discrepancy'.

3.1.4 Quasi-Random Walk

Die sogenannten 'Random Walk'-Verfahren (speziell das in Dutre u. a. genannte 'Photon Density Estimation'-Verfahren) ist eine weitere, zustätzlich zu den in Kapitel 2.1.3 aufgeführten, Möglichkeit die Photonendichte, welche sich proportional zur Radianz verhält, eines Patches in einer für 'Radiosity' vorbereiteten Szene zu simulieren. Analog zum 'Photon Mapping' wird daher der Weg der Photonen verfolgt, die das Licht ausmachen. Dazu wird eine festgelegte Menge an Photonen ausgehend von der Lichtquelle in die Szene geschossen. Deren 'Flugbahnen' werden simuliert und bei jedem auftreffen auf eine Oberfläche wird entschieden, ob das Photon diesen Aufprall überlebt und reflektiert oder stattdessen absorbiert wird. Im klassischen 'Random Walk' wird dafür eine Zufallsentscheidung oder 'Russian Roulette' verwendet. Keller schlägt vor (Keller, 1997, Abschnitt 3.2 'The Quasi Random Walk), basierend auf der Annahme, dass die diffuse Reflexivität der getroffenen Oberfläche nicht stark von der Durchschnittlichen $\overline{p}$ abweicht, diesen Wert für die Entscheidung herzunehmen.

$$\overline{p} := \frac{\sum_{k=1}^{K} p_{d,k}|A_k|}{\sum_{k=1}^{K} |A_k|} \approx \|T_{f_d}\| \tag{3.5}$$

Formel 3.5 zeigt, dass die Summe über eine Menge K an Oberflächenelementen von der diffusen Reflexivität auf allen Flächen dieser Elemente pro Summe der Flächen die eigentliche diffuse Reflexivität in diesem Bereich approximiert.

Die Photonen werden solange durch die Szene verfolgt und an allen Schnittpunkten mit einer Fläche der Überlebens-Test gemacht bis kein Photon mehr übrig bleibt. Für jeden Patch in der Szene erhält man jetzt die Anzahl an Photonen die davon absorbiert wurden und wie viele auf ihrem Weg diesen Patch 'besucht' haben. 'Quasi-Random-Walk' benutzt zum Festellen der Startpunkte der Photonen auf der Lichtquelle und der Richtungen, in welche sie geschossen werden, Werte der quasi-zufälligen Halton-Sequenz. Daher der Name 'Quasi-Random Walk'.

3.2 Instant Radiosity

Alexander Keller, zu dieser Zeit an der Universität Kaiserslautern im Bereich der numerischen Algorithmen tätig, veröffentlichte 1997 eine wissenschaftliche Arbeit zum Thema 'Instant Radiosity'. Durch diverse Erkenntnisse und Weiterentwicklungen im Umfeld von globaler Beleuchtung und damit auch 'Radiosity,' ergaben sich für Keller neue Optionen, die benötigte Zeit zur Erstellung von Bildern mit global errechneter Beleuchtung näher an für Interaktivität notwendige Zeiten anzunähern. Dazu wird in seiner Arbeit ein Weg beschrieben, um eine Annäherung der ersten diffusen Reflexion des Lichts mithilfe des 'Quasi-Radnom Walk'-Verfahrens, welches wiederum auf 'Quasi-Monte-Carlo'-Verfahren basiert, zu errechnen.

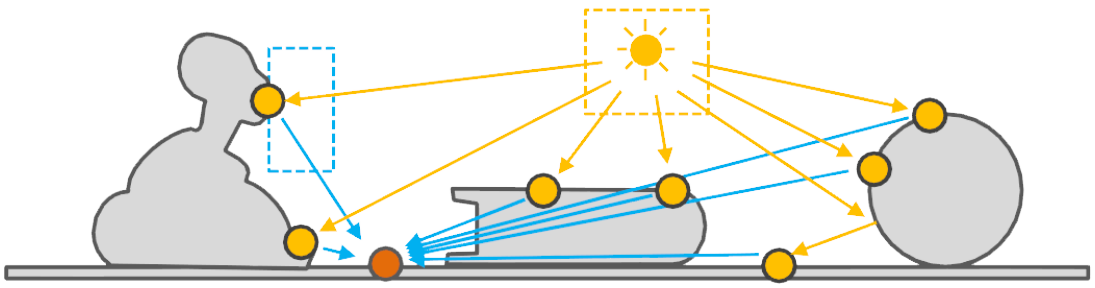

Abbildung 3.5: Das Prinzip von 'Instant Radiosity' bildlich dargestellt. Licht (gelbe Pfeile) wird in die Szene ausgestrahlt. An den Schnittpunkten zwischen den Lichtstrahlen und der Oberfläche werden Punktlichtquellen (gelbe Punkte) platziert. Diese schicken ihr reflektiertes Licht zu jedem Sample das evaluiert wird (blaue Pfeile). Abbildung übernommen von Ritschel u. a. (2012).

Die Grundidee ist dabei nicht wirklich kompliziert: Ausgehend von der Lichtquelle, werden N Photonen in quasi-zufällig gerichtete Bewegungsbahnen geschickt. Trifft das Photon auf eine Oberfläche so wird getestet ob es reflektiert oder absorbiert wird. Wird es reflektiert wiederholt sich das Vorgehen. Nach dem 'Quasi-Random Walk'-Prinzip (im Speziellen Formel 3.5) setzen dabei $\lfloor \overline{p}N \rfloor$ Partikel ihren Weg fort, solange bis alle Partikel absorbiert wurden. Bei jeder Interaktion wird an den entsprechenden Punkt im Raum eine Lichtquelle gesetzt. Die Menge an Punktlichtquellen M korreliert dabei mit der durchschnittlichen Reflexivität der Szene (siehe Formel 3.6). $\overline{l}$ gibt darin die durchschnittliche Länge eines 'Quasi-Random Walk'-Pfades an.

$$M \;<\; \sum_{j=0}^{\infty} \overline{p}^{j} N \;=\; \frac{1}{1-\overline{p}} N \;=:\; \overline{l} N \tag{3.6}$$

Bei ausreichend hoher Zahl an Photonen wird also realistisch das physikalische Verhalten von Licht nachgebildet.

Sind alle Photonen verteilt, werden für jede Punktlichtquelle in M ein eigenes Bild, vollständig mit original texturierter Szenen-Geometrie und Schatten, gerendert. Es ist also jeweils ein vollständiger Durchlauf sowohl von Vertex- als auch von Pixelshader nötig, zusätzlich noch der eine Render-Durchlauf für die eigentliche Schattenberechnung. Das geschieht aus der Kameraposition und jeweils eine Lichtquelle aus M wird dabei alleine in der Szene platziert. Prinzip-bedingt liegen viele Samples im Bereich der originalen Lichtquelle der Szene, sofern diese vom Kamerastandpunkt sichtbar ist. Sind alle Punktlichtquellen gerendert, werden alle Bilder aufsummiert und gewichtet. Das Resultat ist ein Bild mit globaler Beleuchtung.

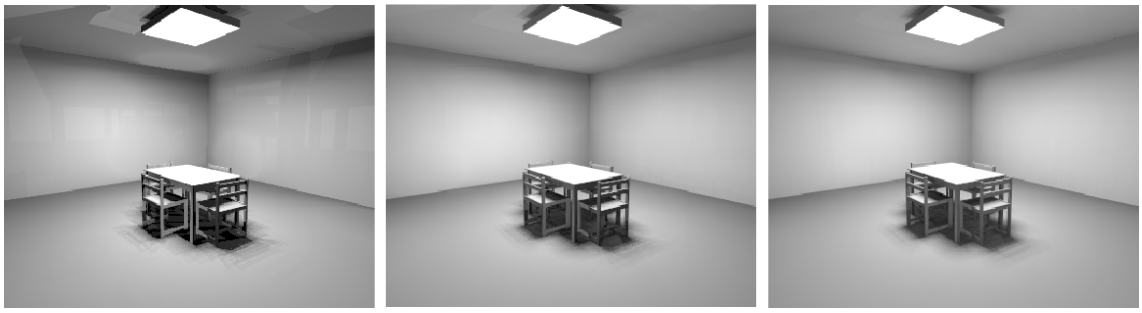

Abbildung 3.6: Drei Ergebnisbilder erstellt mit 'Instant Radiosity' und unterschiedlicher Sample-Anzahl. Sichtbar bei zehn Samples und 20 Lichtquellen (links) sind die wenigen überlagerten Schatten des Tisches. Bei mehr Samples und Lichtquellen werden die Schatten zunehmend weicher. Auch die Artefakte an den Wänden nehmen ab. Abbildung übernommen von Keller (1997).

Man erhält mit diesem Algorithmus zwar ein Bild, das globale Lichtanteile beinhaltet, trotzdem ist es nur eine Annäherung der exakteren 'Radiosity'-Lösung. So beschränkt die Anzahl der simulierten Photonen und damit implizit die Anzahl der Punktlichtquellen, die Genauigkeit der Lösung. Um ein perfektes Ergebnis zu erreichen, wären unendlich viele Lichtquellen nötig. Durch die Verwendung von Punktlichtquellen für die Reflexion von

Licht an Oberflächen ist das Verfahren ausschließlich auf diffuse Reflexion beschränkt. Die Abstrahlcharakteristik ermöglicht nicht, das Verhalten einer beliebigen, durch eine BRDF beschriebene, nicht-diffuse Reflexion. Es wird also nur kugelförmig reflektiert und nicht zigarrenförmig, wie die spekulare Reflexion es voraussetzt. Keller fürht noch zwei weitere wichtige, dem Algorithmus eigene, Punkte an, die zu Bedenken sind: Nähert sich der Abstand zwischen einer Punktlichtquelle und einem Empfängerpunkt null an, so führt das dazu, dass der Wert im Framebuffer zu groß wird und nach oben hin begrenzt werden muss, was einen gewissen Informationsverlust bedeutet. Ein anderes Problem ist, dass Punktlichtquellen, die entsprechend der Oberflächen-Farbe, eingefärbt sind einen sehr großen Einfluss auf die Gesamtfarbe einer Szene hat. Da die Bilder aber mit $1/N$ gewichtet werden, ist die Abweichung nur schwer erkennbar.

Das Verfahren beschleunigt die Erzeugung von 'Radiosity'-Bildern um mindestens eine Größenordnung. So gibt Keller an,dass sich die Renderzeit pro Bild 1997 von Stunden auf wenige Minuten reduzierte. Und das bei ähnlicher Qualität. Möglich macht das der Verzicht auf den aufwändigen Zwischenschritt der Einteilung der Szenen-Geometrie in Patches. Sowie der durch das 'Quasi-Monte Carlo'-Prinzip bedingte Verzicht auf die Formfaktor-Errechnung, womit ein extrem großer Teil der Komplexität entfällt. Dabei bleibt es möglich die mehrfache diffuse Reflexion zu simulieren, ebenso kann die spekulare Reflexion durch wenige Anpassungen im Algorithmus hinzufügen.

Keller gibt noch weitere Vereinfachungen und Techniken an, um 'Instant Radiosity' Echtzeit-fähig zu machen. So schlägt er vor, den 'Quasi-Random Walk' durch ein Verfolgen der Photonen mit fester Länge der Pfade zu ersetzen. So wird sichergestellt, dass die Photonen-Simulation jeden Frame eine ähnlich lange Zeit benötigt. Des weiteren werden die letzten N Bilder der Photonen-Pfade gespeichert und nur, wenn ein neuer Pfad erstellt wurde, das jeweils älteste Bild verworfen und durch das Neue ersetzt. Das hält den Rechenaufwand in Grenzen, führt aber zu Flackern oder 'Nachziehen' der Beleuchtung bei schnellen Bewegungen von Kamera oder originaler Lichtquelle in der Szene.

4 Instant Radiosity Erweiterungen

Zwischen der Vorstellung von 'Instant Radiosity' im Jahr 1997 und dem heutigen Tage gab es zahlreiche Weiterentwicklungen in diesem Bereich. So wurde das Verfahren von Keller gezielt weiterentwickelt um noch bessere Bildwiederholraten bei gleichbleibender Qualität zu erzielen. Ebenso wird an Techniken geforscht, die Prinzip-bedingten Einschränkungen aufzuweichen oder gar aufzuheben. In dieser Zeit sind einige große Schritte in diesen Bereichen gemacht worden. Dieses Kapitel beschreibt nun drei dieser Techniken, welche unterschiedliche Aspekte des ursprünglichen Algorithmus verbessern. Zuerst wird der Fokus auf grundlegende Render-Techniken gelegt, die für das Verständnis der darauf folgenden Verfahren hilfreich ist (Kapitel 4.1). Danach wird zunächst Carsten Dachsbachers und Marc Stammingers Technik der 'Reflective Shadow Maps' in Kapitel 4.2 dargelegt, bei dem es sich um eine schnelle Möglichkeit handelt, die Punktlichtquellen zu verteilen. Darauf folgt das 'Imperfect Shadow Maps'-Verfahren (Kapitel 4.3), was das Problem der Punkt-zu-Punkt-Sichtbarkeiten durch niedrig aufgelöste 'Shadow Maps' löst. Abschließend wird in Kapitel 4.4 das von Greg Nichols, Jeremy Shopf und Chris Wyman entwickelte 'Instant Radiosity'-Verfahren 'Hierarchical Image-Space Radiosity', welches mithilfe von 'Multi-Resultion'-Ansätzen und 'Stencil'-Techniken eine robuste und schnelle 'Radiosity'-Lösung ermöglicht.

Für alle Methoden wird zuerst auf das allgemeine Prinzip eingegangen. Ebenso werden die vorgenommen Annahmen und Einschränkungen sowie eine kurze Analyse der Qualität der Ergebnisse und der Performance vorgenommen.

4.1 Grundlegende Techniken

Bevor die weiterführenden Techniken aufgeführt werden, werden zwei grundlegende Render-Techniken beschrieben, die von den komplexeren 'Instant Radiosity'-Vorgehen benutzt werden. So beschreibt Kapitel 4.1.1 das Verfahren der 'Shadow Maps'. Ein 'Mulit-Pass'-Rednering-Verfahren, das das Rendern in den Textur-Speicher ausnutzt um schnell Schatten rechnen zu können. 'Deferred Shading' hingegen, beschrieben in Kapitel 4.1.2, beschleunigt solche 'Multi-Pass'-Verfahren wie das genannte 'Shadow-Mapping'.

4.1.1 Shadow Maps

Mit dem 'Shadow-Mapping'-Verfahren, schon 1978 erstmals beschrieben von Williams, kann man in interaktiven Szenen einen nahezu korrekten Schattenwurf der darin enthaltenen Objekte berechnen. Dies ist nötig um einen realistischen Eindruck der Beleuchtungssituation zu erreichen. Ein Objekt ohne Schatten scheint im Raum zu schweben. Die korrekte Position relativ zum Boden (sofern das Objekt denn auf dem Boden steht) ist dann nur schwer für das menschliche Auge festzustellen.

In 'Ray-Tracing'-Verfahren wird für jeden zu berechnenden Pixel ein sogenannter Schattenfühler von der Position des Punktes im Raum in Richtung der Lichtquelle(n) geschickt. Für jeden dieser Schattenfühler muss ein Kollisionstest mit der kompletten Szene durchgeführt werden. Befindet sich zwischen Lichtquelle und Punkt ein Objekt, so befindet er sich im Schatten. Dieses Vorgehen ist, wie man erkennen kann, sehr aufwändig und damit Rechenzeit-intensiv. Daher wurden andere Möglichkeiten entwickelt, um den korrekten Schattenwurf schnell auf Grafik-Hardware berechnen zu können.

Diese Techniken sind auf Grafik-Hardware praktisch immer 'Multi-Pass'-Verfahren. Das bedeutet, dass die Schatten in mehreren Schritten berechnet werden. Beim 'Stencil Shadowing' beispielsweise wird in einem erst Schritt (auch 'Pass' genannt) die Szene mit direkter Beleuchtung ohne Schatten gerendert. Danach wird die Geometrie aller Objekte entlang ihrer Umrisse und entlang des Öffnungsvolumens des Lichts ins Unendliche verlängert. Dieses Volumen wird aus Kamerasicht evaluiert, welcher Punkt im Schatten liegt und welcher nicht. Was dann im Schatten liegt wird entsprechend ausgeschnitten (daher der Begriff 'Stencil', zu deutsch Schablone) und nur dort wird dann das Bild abgedunkelt.

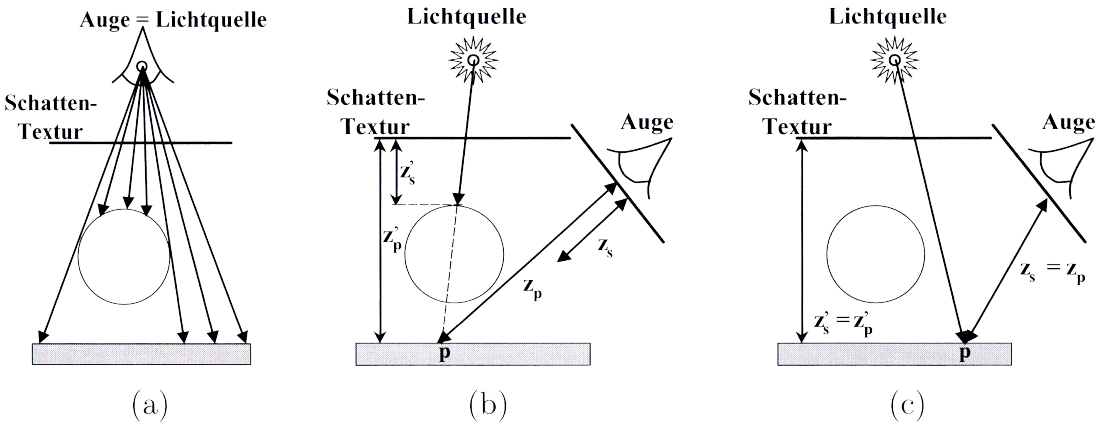

Abbildung 4.1: Drei Bilder erklären das 'Shadow Mapping'-Prinzip. Im ersten Pass werden die Tiefenwerte aus Lichtposition und -richtung gerendert (a). (b) zeigt einen positiven Schattentest, weil der Abstand zwischen 'Shadow Map' und Punkt kleiner ist als der von Kamera und Punkt. Sind beide Abstände gleich, liegt der Punkt nicht im Schatten (c). Abbildung übernommen von Nischwitz u. a. (2011).

'Shadow Maps' hingegen wählen einen umgekehrten Ansatz die Verdeckung von Licht durch Geometrie zu errechnen. Statt zu testen, ob ein Punkt die Lichtquelle 'sehen' kann, wird abgeprüft und zwischengespeichert, welche Punkte auf der Oberfläche der Szene die Lichtquelle direkt sehen kann. Dazu werden aus Position und Richtung des Lichts die sichtbaren Tiefenwerte in eine Textur gespeichert, den sogenannten 'z-Buffer'. Alles was nicht in dieser 'Depth Map' verzeichnet ist befindet sich damit entweder außerhalb des Lichtvolumens oder ist verdeckt. Ausgewertet wird die 'Depth Map' einen Renderdurchlauf später. Die Szene wird ganz normal, also als würde man nur die direkte Beleuchtung berechnen wollen, aus Kamera-Sicht gerendert. Nur wird zusätzlich für jeden Punkt p ein Schattentest durchgeführt. Dabei werden die Abstände zwischen der Schattentextur

und p (in Formel 4.1.1 z_s) und zwischen der Kameraposition und p (in Formel 4.1.1 z_p) verglichen. Ist der Abstand z_s kleiner als z_p so liegt der Punkt p in im Schatten, da eine Fläche zwischen Licht und p existiert. Sind die Abstände gleich, so gibt es keine Hindernis, das Licht blockieren könnte.

Abgeänderte 'Shadow Mapping'-Verfahren werden in 'Reflective Shadow Maps' und 'Imperfect Shadow Maps' in unterschiedlicher Weise verwendet um globale Beleuchtung zu simulieren.

4.1.2 Deferred Rendering

In den frühen 90er Jahren entwickelten Saito u. Takahashi eine Technik, um Geometrie in sogenannte 'g-Buffer' zu rendern, um dann auf den Puffern komplexe Berechnungen auszuführen. Mit der Zeit wurde die Idee aufgegriffen und zum 'Deferred Shading' weiterentwickelt um 'Multi-Pass'-Verfahren zur Beleuchtung zu beschleunigen und vereinfachen. Dazu zählen vor allem Situationen mit vielen Lichtquellen. Dies geschieht dadurch, dass die Durchläufe der Szenen-Geometrie von den nachfolgenden Berechnungen entkoppelt wird und man möglichst selten die Geometrie rendern muss. Dafür werden alle benötigten Informationen der Szene in sogenannte 'g-Buffer' gerendert, darunter mindestens Weltkoordinaten und Oberflächennormale. Es ist dem Benutzer überlassen weitere Informationen zu speichern, um weitere Effekte zu erzielen. Um eine klassische direkte Beleuchtung verzögert (also 'deferred') rendern zu können, benötigt man neben Weltkoordinaten und Normalen noch den Farbwert der Oberflächen. Anschließend wird nach einem beliebigen Beleuchtungsmodell wie zum Beispiel Phong, das endgültige Bild errechnet werden.

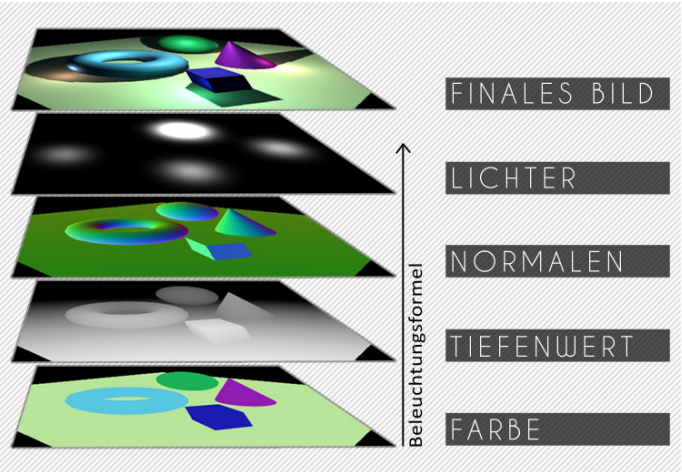

Abbildung 4.2: Eine Übersicht, wie ein 'Deferred Shading'-Pass aufgebaut ist. Das endgültige Bild wird errechnet aus den in einem vorherigen Pass gerenderten Bildern, die Farbinformation, Tiefenwerte, Oberflächennormalen und Lichter enthalten. Abbildung übernommen von `http://de.wikipedia.org/wiki/Deferred_Shading` (Aufgerufen am 20.6.2013).

In 'Instant Radiosity'-Methoden wird 'Deferred Rendering' dazu verwendet, die sehr komplexen 'Multi-Pass'-Berechnungen von der Szenenkomplexität zu entkoppeln. Man erkauft sich so eine höhere Rendergeschwindigkeit mit massiv gesteigertem Platzbedarf in Grafikspeicher und Anzahl an Texturzugriffen.

4.2 Reflective Shadow Maps

Dachsbacher u. Stamminger kombinieren in ihrem Ansatz 'Instant Radiosity'-Methoden

mit dem populären 'Shadow-Mapping'-Algorithmus um Punktlichtquellen zu generieren. Die Schattentextur soll die auf Grafik-Hardware nicht ideal ausführbare Partikelsimulation zur Photonenverteilung aus der Methode von Keller ersetzen. Es wird im Folgenden auch eine Analyse des 'Splatting Indirect Illumination'-Algorithmus (Kapitel 4.2.1) vorgenommen, welche zu Teilen auf dem selben Ansatz basiert, aber eine andere Vorgehensweise der Evaluation der indirekten Lichtanteile benutzt.

Prinzip

Wie bereits erwähnt, ersetzen die 'Reflective Shadow Maps' die Partikelsimulation der Photonen zum Verteilen der Punktlichtquellen (in dieser Methode als 'Pixellicht' betitelt). Das Prinzip folgt der Überlegung, dass alle Oberflächen, die Licht reflektieren können von den Lichtquellen aus sichtbar sein müssen. Hat man nur eine Lichtquelle bietet sich die Schattentextur natürlich an, da diese schnell auf Grafik-Hardware berechenbar ist. Auf dieser Schattentextur wird jeder Texel (Textur-Pixel) als Pixellicht betrachtet. Dazu müssen in einer 'Reflective Shadow Map' nicht nur Tiefenwerte, wie im klassischen Ansatz, gespeichert werden. Zusätzlich werden, analog zum 'Deferred Rendering', noch weitere Informationen wie Weltkoordinate des Punkts, Oberflächennormale und Lichtfluss abgespeichert. Um den korrekten Wert von Φ feststellen zu können, muss vor dem Speichern der Lichtfluss durch jeden Pixel der 'Reflective Shadow Map' festgestellt werden und mit dem Reflexivitätswert der darunterliegenden Oberfläche multipliziert werden. Dachsbacher u. Stamminger verzichten dabei ausdrücklich auf die Speicherung der Radianz der reflektierten Lichtstrahlung, da diese Werte Flächenabhängig sind und sich so Aufwand sparen lässt.

Die Radianz I in Punkt p lässt sich mit Hilfe dieser Information folgendermaßen berechnen (vgl. Dachsbacher u. Stamminger, Abschnitt 3.1 'Data'):

$$I_p \;=\; \Phi_p max(0, \langle n_p, \omega \rangle) \tag{4.1}$$

So gibt in Formel 4.1 Φ_p den reflektierten Flux (Lichtfluss) in Punkt p an, $\langle\rangle$ gibt das Skalarprodukt zwischen der Normalen n von p und der Lichtrichtung ω an. Die eingehende Lichtstrahlung von Pixellicht p E in Punkt x auf einer Oberfläche mit Normale n gibt somit Formel 4.2 an.

$$E_p(x, n) \;=\; \Phi_p \frac{max(0, \langle n_p, x - x_p \rangle) max(0, \langle n, x_p - x \rangle)}{\|x - x_p\|^4} \tag{4.2}$$

Welche Auswirkungen Formel 4.2 mit sich bringt, wird im nachfolgenden Abschnitt beleuchtet.

Durch die vorgenommenen Geschwindigkeitsverbesserungen des 'Instant Radiosity'- Algorithmus ist nun die Evaluation der indirekten Beleuchtung der aufwändigste Schritt. Um das Verfahren noch performanter zu machen, wird ein Interpolations-Algorithmus hinzugefügt, der sich zu Nutze macht, dass indirekte Beleuchtung auf Flächen nur in Ausnahmefällen niederfrequente Details aufweist. Um das eingehende indirekte Licht aufzusammeln genügt es prinzipiell, für jeden zu rendernden Pixel den Einfluss aller in der Szene verteilten Pixellichter aufzusummieren. Bei einer Beispielhaft gewählten Auflösung der 'Reflective Shadow Map' von $N_p = 1024^2 \approx 1M$ Lichtquellen ist das eine gewaltige

Datenmenge. Dachsbacher u. Stamminger reduzieren diese Komplexität durch die Verwendung von 'Importance Sampling'-ähnlichen Ansätzen um die Anzahl der verwendeten Pixellichter von über einer Millionen auf einen Wert zu beschränken der einfacher handhabbar ist. Es werden die Lichter so ausgewählt, dass möglichst nur Lichtquellen platziert werden, die auch wirklich relevant für einen realistischen Eindruck der indirekten Beleuchtung sind. Dazu wird angenommen, dass benachbarte Punkte in der Szene sich auch als Pixel in der erweiterten Schattentextur in direkter Nachbarschaft befinden. Unterscheiden sich nun die gespeicherten Normalen zwischen einem Punkt, der aktuell gerendert werden soll, und dem in der Schattentextur gespeicherten Normalen eines Pixellichts, wird entweder kein Licht angenommen (Normalen zeigen voneinander weg) oder wird indirekte Beleuchtung berücksichtigt (Normalen zeigen aufeinander).

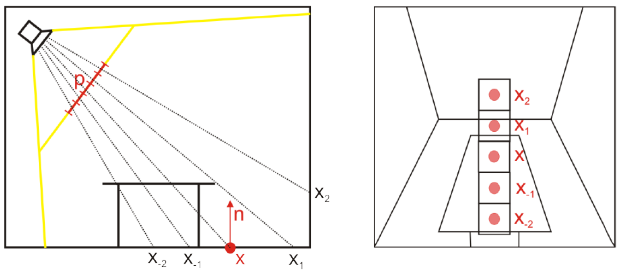

Abbildung 4.3: Der Punkt x und seine Nachbarpunkte x_{-1}, x_{-2}, x_1 und x_2 sind auch in der erweiterten Schattentextur (rechts) benachbart. Da die Normalen von x_{-1} und x_{-2} von x weg zeigen und x_1 auf der selben Ebene liegt, wird x nur von Pixellicht x_2 beleuchtet. Abbildung übernommen von Dachsbacher u. Stamminger (2005).

Durch die lokale Begrenzung des Einflusses der Pixellichter verwendeten Dachsbacher u. Stamminger ein Sampling-Muster um sich Pixellichter aus der Umgebung von einem Punkt zu holen. Mit diesem Sample wird dann der Test der Normalen ausgeführt und das eintreffende Licht nur von einer stark begrenzten, statt der maximal verfügbaren, Anzahl an Pixellichtern aufgesammelt.

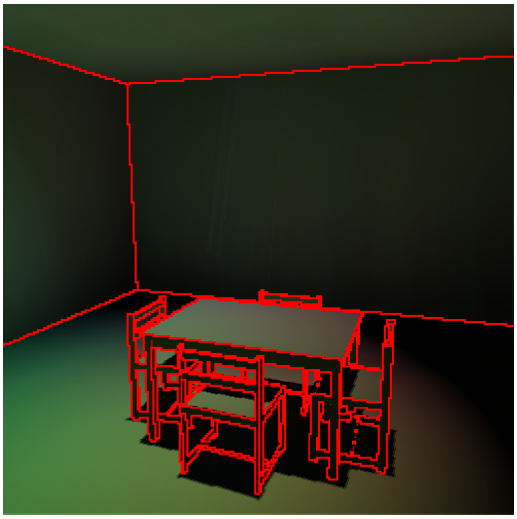

Abbildung 4.4: Alle roten Punkte in diesem Bild sind Pixel die in vollständiger Auflösung evaluiert werden müssen. Sichtbar ist, dass sich diese an Stellen häufen, wo es große Unterschiede zwischen den Normalen gibt, also Kanten. Abbildung übernommen von Dachsbacher u. Stamminger (2005).

Um die Anzahl der zu evaluierenden Objekte noch weiter zu reduzieren, wird die Szene zuerst in einem sehr grob aufgelöstem Render-Pass aus Kamerasicht aufgenommen. Direkt danach wird die Szene in voller Auflösung gerendert und für jeden Pixel entschieden, ob der Wert aus der niedrig aufgelösten Version interpoliert werden kann oder ob er genau

evaluiert werden muss. Als Kriterium für eine Interpolation wird die Normale und die Weltkoordinate herangezogen. Weichen diese zu stark ab, kann man davon ausgehen, dass dieser Pixel beispielsweise eine Kante bedeckt und daher höherfrequente Details vonnöten sind (siehe dazu Bild 4.4).

Annahmen & Einschränkungen

Die Beschleunigungen im Rendervorgang kommen teilweise von Vereinfachungen in den Berechnungen, die optisch nicht direkt wahrnehmbar sind aber viel Rechenzeit sparen können. Manche eingesetzten Techniken haben aber auch Nachteile, die sich im Endergebnis bemerkbar machen.

'Reflective Shadow Maps' nimmt ebenfalls Vereinfachungen vor, um die mathematische Problemstellung einzugrenzen. Prinzip-bedingt gibt es auch Einschränkungen, die eine genauere Lösung verhindern oder zumindest deutlich aufwändiger in der Berechnung machen. Die erste Einschränkung betrifft die Form des eingesetzten Lichtes. Bedingt durch das 'Shadow Mapping'-Verfahren und dem Umstand, dass die Kamera zum Rendern der Schattentextur einen Öffnungswinkel hat, werden deshalb nur Spot- und parallele Lichtquellen unterstützt. Mehrere Lichtquellen sind theoretisch möglich, führen aber zu einem drastisch steigenden Rechenaufwand. So müsste nicht nur aus einer Schattentextur Pixellichter ausgesucht werden, sondern aus allen Texturen für jede Hauptlichtquelle. Ebenso sind Flächenlichter prinzipiell möglich und mit Algorithmen für weichen Schattenwurf realistisch berechenbar, auch wenn der Rechenaufwand auch hier ansteigt. Zusätzlichen Aufwand bedeutet auch die Unterstützung von Punktlichtquellen. Hierzu müsste statt einer Schattentextur pro Lichtquelle deren sechs berechnet werden um jegliche Richtung der Lichtausbreitung abdecken zu können.

Weiterhin wird in der Schattentextur nur der Flux von diffusen Oberflächen gespeichert. Das limitiert die globale Beleuchtung auf das indirekte Licht von diffusen Oberflächen. Punktlichtquellen haben ihre Kugelförmige Abstrahcharakteristik mit dem Reflexionsverhalten von diffusen Oberflächen gemeinsam. Hier ist die Simulation von nicht-diffusen Oberflächen realisierbar, indem statt dem Flux in der erweiterten Schattentextur beispielsweise eine Material-ID gespeichert wird, um beim Evaluierungsschritt dann gemäß einer BRDF zu rechnen. Da bei der Evaluation eines Punktes auf der Oberfläche alle Pixellichtquellen der Umgebung betrachtet werden und beim Sampling-Verfahren nur ganz rudimentär ein Punkt-zu-Punkt Sichtbarkeitstest durchgeführt wird, kann es passieren, dass Punkte indirekt Beleuchtet werden, die von keiner (oder einer falschen) Pixellichtquelle beleuchtet werden.

Dem gegenüber stehen Vorteile, die 'Shadow Maps' mit sich bringen: Das Verfahren ermöglicht sowohl dynamische Geometrie, als auch eine bewegliche Kamera und Lichtquelle, da die 'Reflective Shadow Map' jeden Frame neu gerendert wird und damit immer auf dem aktuellen Stand der Szene ist.

Qualität & Performance

Einige Qualitätsmerkmale lassen sich schon aus den aufgeführten Einschränkungen ableiten. Der nur rudimentär vorhandenen Punkt-zu-Punkt Sichtbarkeitstest führt natürlich zu einer nicht korrekten Lösung der Beleuchtung. Pixel, die in der Realität kein indirektes Licht empfangen würden, tun dies trotzdem, weil ein blockierendes Objekt nicht erkannt

wird, das zwischen Pixel und Pixellichtquelle liegt. Generell lässt sich allerdings sagen, dass allein schon der angedeutete Effekt von indirekter diffuser Reflexion zu befriedigenden Ergebnissen führen kann (vgl. hierzu Dachsbacher u. Stamminger, Kapitel 3.3 'Evaluation'). Weiterhin negativ für die Qualität bemerkbar macht sich das Fehlen von hochfrequenten Details in der indirekten Beleuchtung abseits von Kanten, da in diesen Bereichen das diffus reflektierte Licht aus einer grob aufgelösten Version der Szene interpoliert wird statt dies Pixelweise in der originalen Auflösung zu tun. Auch die fehlende Selbstabschattung der indirekten Beleuchtung (auch dies ist bedingt durch die fehlenden Punkt-zu-Punkt Sichtbarkeitstests) kann im Ergebnisbild stören. Dachsbacher u. Stamminger schlagen zur Lösung vor, 'Ambient Occlusion' zu verwenden, um die Selbstverdeckung der Szenengeometrie hinzuzufügen. Einen weiteren Vorteil, neben den eingesparten Rechenoperationen, haben die fehlenden Sichtbarkeitstest. So hat das Beleuchtungsintegral 2.5 an Stellen, wo zwei Ebenen aufeinandertreffen eine Singularität, welche sich umgehen lässt, indem man die Pixellichter leicht entgegen der Oberflächennormalen verschiebt.

Eine zu niedrig gewählte Anzahl an Pixellichtern führt zu einer sehr grob angenäherten globalen Beleuchtung und Artefakte können sichtbar werden. Ebenfalls ist bei niedriger Anzahl der sekundären Lichtern und einer Kamerabewegung ein deutliches Flackern feststellbar, das daher kommt, dass Pixellichtquellen in der Zwischenzeit zwischen zwei aufeinanderfolgenden gerenderten Bildern ihre Position wechseln und so kurzzeitig für sichtbar abweichende indirekte Beleuchtung führt.

Die Rendergeschwindigkeit wird maßgeblich von der Auflösung der 'Reflective Shadow Map' und der Anzahl der Pixellichtquellen, welche pro Punkt evaluiert werden sollen, beeinflusst. Auch die Anzahl der regulären Lichtquellen hat einen starken Einfluss. Durch Verwendung von 'Deferred Shading' für indirekte Beleuchtung ist die Geschwindigkeit von der Komplexität der Szene teilweise entkoppelt. Teilweise deshalb, weil neben dem initialen Geometriedurchlauf Kanten aus Sicht der Kamera in voller Auflösung evaluiert werden müssen, da sie nicht aus der niedrigen Auflösung interpoliert werden können.

Da zusätzlich zum 'Deferred Rendering' noch 'Shadow Mapping' verwendet wird, ist als hauptverantwortlicher Flaschenhals die Speicherbandbreite zu nennen. Mit steigender Auflösung steigt sowohl der Speicherverbrauch als auch die nötigen Texturzugriffe auf die g-Buffer und die erweiterte Schattentextur. So erreichten Dachsbacher u. Stamminger 2005 auf einer Nvidia Quadro FX4000 in Szenen mit geringer Komplexität in guten Fällen (deterministisch ausgesuchte Parameter) circa 18 bis $27 Bilder/Sekunde$ bei einer Auflösung von 512^2 Pixeln. Um diese Zahl etwas besser einordnen zu können sollte man sich die Entwicklung der Speicherbandbreite über die Jahre vor Augen führen. So hatte die verwendete Grafikkarte, welche mit der Nvidia Geforce 6800GT direkt vergleichbar ist, für heutige Verhältnisse geringe $32 GB/Sekunde$ Speicherbandbreite (Spille u. Becker (2004)). Heute sind Grafikkarten eines vergleichbaren Segments bei circa $224 GB/Sekunde$ Speicherbandbreite (Andermahr (2013)). So kann man sagen, dass mit diesem Algorithmus heute selbst komplexe Szenen mit interaktiven Bildwiederholraten möglich sind, da fast zehnmal so viel Bandbreite zur Verfügung steht wie damals. Geht man davon aus, dass sich die Restlichen Parameter der Grafikhardware ähnlich explosiv entwickelt haben scheint eine mindestens zehnmal höhere Performance möglich, also liegt die theoretisch mögliche Bildwiederholrate heute in einem Bereich von 180 bis 270 Bilder in der Sekunde, allerdings in sehr einfachen Szenen.

Es gilt zu bedenken, dass diese errechneten Werte rein theoretischer Natur sind und aus-

schließlich einer groben Vergleichbarkeit zwischen den verschiedenen Ansätzen für 'Instant Radiosity' dienen. Da bei allen Techniken, sofern nicht anders angegeben, das begrenzende Element die Speicherbandbreite ist, wird dieser Wert auch dazu hergenommen die Leistung linear auf moderne Grafikhardware hochzurechnen. Als Vergleich dient der neuste Spross der Nvidia Performance-Grafikkarten, die Geforce GTX 770. Effizientere Berechnungen moderner Grafikkarten werden ebenso wenig mitberechnet, wie unterschiedlich optimierte 'Instant Radiosity'-Algorithmen. Abhängig von der Szenenkomplexität, Auflösung und Qualität soll erkenntlich sein wie gut der Grundgedanke der Rendergeschwindigkeit zuträglich ist.

4.2.1 Splatting Indirect Illumination

Einen anderen Weg der Evaluierung der indirekten Beleuchtung wählten Dachsbacher u. Stamminger in ihrem Ansatz zum 'Splatting Indirect Illumination'. es wird nun nicht mehr ein 'Gathering'-Ansatz gewählt, wie noch bei 'Reflective Shadow Maps', sondern ein 'Shooting' der indirekten Lichtanteile. Dazu wird eine Teilmenge von Pixellichtern ausgewählt und deren Beitrag zum indirekten Licht in die Szene verteilt (oder 'geschossen'). Dieses 'Shooting' passiert in einem separaten Pass, indem auf der Position in Bildschirmkoordinaten einer ausgewählten Pixellichtquelle zunächst ein Viereck erzeugt wird. Jeder Pixel der Szene (durch Nutzung von 'Deferred Rendering' handelt es sich hierbei um einen g-Buffer), der von diesem Viereck überdeckt ist, bekommt indirektes Licht ab. Die Größe wird dadurch bestimmt, dass intensiver Strahlende Pixellichter eine größere Fläche in Bildschirmkoordinaten einnehmen, je weiter weg das entsprechende Licht ist, desto kleiner wird die Fläche.

Die sekundären Lichtquellen werden über den 'Hierarchical Warping'-Algorithmus (siehe Kapitel 3.1.2) entsprechend des Lichtflusses verteilt. Stellen, die mehr direktes Licht erhalten, weisen eine gehäufte Verteilung der Punktlichtquellen auf. So kommt diese Methode mit signifikant weniger Punktlichtquellen aus als noch 'Reflective Shadow Maps'. Wurde dort noch 400 evaluierte Pixellichter pro gerendertem Pixel angegeben um realistische Ergebnisse zu erhalten, so sind mit 'Splatting Indirect Illumination' deutlich weniger sekundäre Lichtquellen nötig um ein Bild ohne Artefakte der indirekten Beleuchtung zu erhalten.

Dachsbacher u. Stamminger geben eine einfache Möglichkeit an, die diffuse Reflexion auf eine nicht-diffuse Oberflächen unterstützende Reflexion zu erweitern. So wird der Term 4.1, der auch in 'Reflective Shadow Maps' Verwendung fand, auf eine Phong-basierte Berechnung erweitert (Formel 4.3).

$$I_p \;=\; \Phi_p max(0, \langle r_p, \omega \rangle)^P \tag{4.3}$$

r_p gibt nun die schon in Bild 2.2 aufgezeigte Reflektionsrichtung den einfallenden Lichts aus Richtung ω an. P ist nun gemäß des Phong'schen Beleuchtungsmodells der spekulare Exponent. Um diesen Term während dem Rendern evaluieren zu können, müssen inder erweiterten Schattentextur entsprechend dem Material die Parameter für die spekulare Reflexion gespeichert werden. Neben P bietet es sich an, beim Erstellen der Schattentextur schon die Reflexionsrichtung r_p zu errechnen und mit zu speichern.

Spekulare Reflexionen setzen allerdings voraus, dass es möglich ist, hochfrequente Details zu simulieren. Das ist mit den vorher genannten Bildschirmkoordinaten-Vierecken

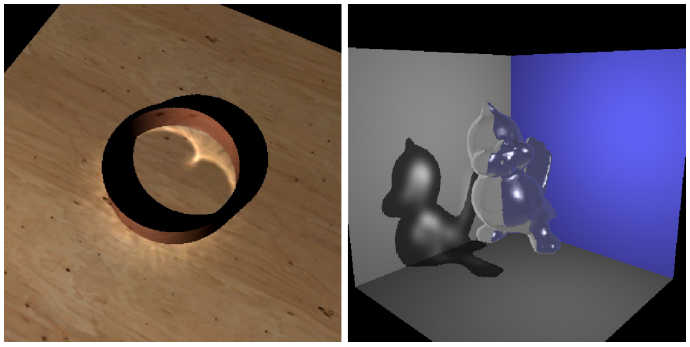

Abbildung 4.5: Ein glänzender Ring (links) mit physikalisch korrekt simulierten Kaustiken auf der Tischplatte. Ein transparentes Objekt (rechts) wirft Kaustiken auf eine nahe Wand. Abbildung übernommen von Dachsbacher u. Stamminger (2006).

nicht ohne weiteres machbar. Dazu braucht es Geometrien, die sich enger um die Reflexionsrichtung eines Pixellichts anpassen. Dachsbacher u. Stamminger, 2006 schlägt dafür Ellipsoiden vor, die zwar einen höheren Transformationsaufwand als Vierecke haben, aber deutlich genaueres 'Splatting' ermöglichen.

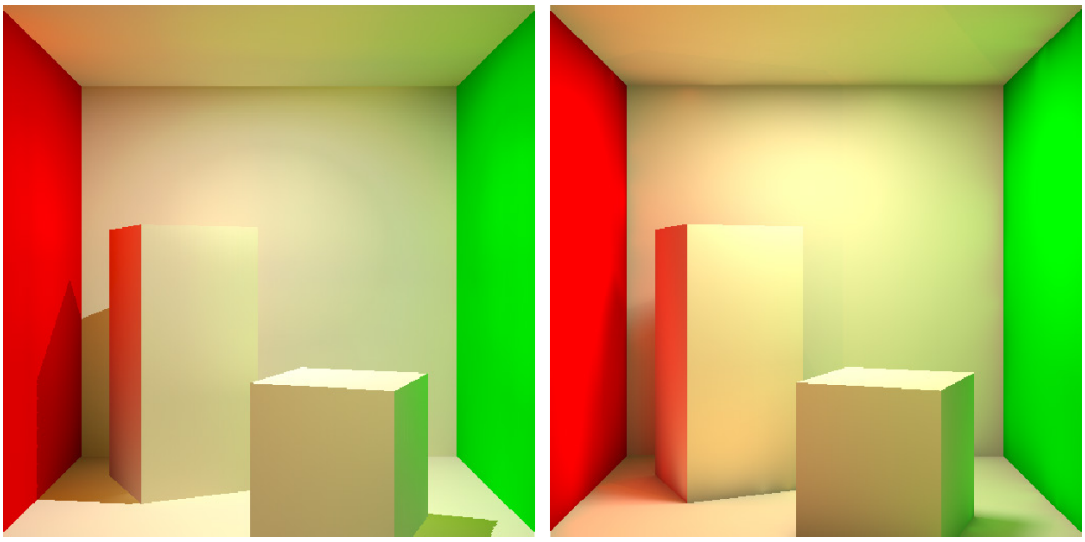

Abbildung 4.6: Der Vergleich zwischen 'Splatting Indirect Illumination' und einem aufwändig gerenderten 'Radiosity'-Bild zeigt wenige Unterschiede. Der klassische 'Radiosity'-Ansatz produziert weiche Schatten. Die Echtzeit-Implementation zeigt dagegen die nicht vorhandene Verdeckung für indirektes Licht. Sichtbar ist das besonders an der der roten Wand zugerichteten Seite des großen Quaders. Es sind klar grüne Lichtanteile der gegenüberliegenden Wand erkennbar. Abbildung übernommen von Dachsbacher u. Stamminger (2006).

Das eigentliche Ziel dieser Technik war nicht, die Qualität gegenüber 'Relfective Shadow Maps' zu verbessern, sondern eine Technik aufzuzeigen, die eine deutliche Reduzierung der zu evaluierenden Pixellichter ermöglicht und deren Einflussbereich verkleinert. So evaluiert beim klassischen Ansatz jeder Pixel eine Teilmenge der Pixellichter, 'Splatting Indirect Illumination' hingegen verteilt das Licht von einer Teilmenge von Pixellichtern auf eine Teilmenge der Pixel und erzwingt so noch weiter die lokale Beschränktheit des Lichteinflusses eines sekundären Lichtes. Als Nebeneffekt ist mit feinerer Geometrie nicht-diffuse

Reflexion möglich (Bild 4.5). Der neue Algorithmus erbt also nahezu alle Einschränkungen seiner Vorgängertechnik. In Bild 4.6 erkennt man gut die fehlenden Punkt-zu-Punkt Sichtbarkeitstests. Die grüne Wand strahlt Licht auf die abgewandte Seite des großen Würfels ab, was im klassischen 'Radiosity'-Algorithmus nicht vorkommt, da dort die Verdeckung für indirektes Licht korrekt berechnet wird.

Auch die Geschwindigkeitscharakteristik erbt diese Methode in großen teilen von 'Reflective Shadow Maps'. 'Deferred Rendering' und 'Shadow Mapping' belasten die Speicherbandbreite stark. Zusätzlich belegt der Ansatz die Fragment-Shader der Grafik-Hardware mit Rechenaufwand, da für jeden 'Splat' (Viereck oder feinere Geometrie) die darunterliegenden Pixel evaluiert werden müssen, was bei einer realistischen Zahl an Pixellichtern dazu führt, dass viele Pixel mehrfach bezeichnet werden müssen. Bei entsprechend eingestellter Größe der 'Splats' sind insgesamt weniger Berechnungen zu tätigen, da jede Lichtquelle nur eine stark beschränkte Anzahl an Pixeln beeinflusst. Die Geschwindigkeit skaliert also nicht mehr linear zu der Anzahl der Punktlichtquellen, sondern nur noch mit der Anzahl der Lichtquellen, die pro Pixel evaluiert werden müssen.

Die Vergleichsergebnisse zwischen 'Reflective Shadow Maps' und 'Spaltting Indirect Illumination' von Dachsbacher u. Stamminger, 2006 zeigen deutliche Geschwindigkeits-Steigerungen von circa 300% auf nahezu identischer Hardware. Wobei davon auszugehen ist, dass für physikalisch korrekte Ergebnisse eine größere Anzahl an Punktlichtquellen als die verwendeten 768 nötig sind, ebenso wie genauere Grenzwerte der Intensität, um die Abstrahlcharakteristik bestmöglich anzunähern.

4.3 Imperfect Shadow Maps

Mit den 'Imperfect Shadow Maps' stellten Ritschel u. a. 2008 einen 'Instant Radiosity'-Algorithmus vor, der einen gänzlich anderen Weg nimmt indirekte Beleuchtung zu simulieren, als die beiden vorangegangenen Methoden. Es basiert auf korrekten Sichtbarkeitstests zwischen Punktlichtquellen und beliebigen Punkten in der Szene. Die Verteilung der sekundären Lichtquellen erfolgt in beliebiger Form. Die Punkt-zu-Punkt Sichtbarkeitstests folgen dem Prinzip der 'Shadow Maps'. So wird für jede Punktlichtquelle eine eigene Schattentextur gerendert. Da das mit originaler Szenengeometrie viel zu aufwändig wäre, werden diese Texturen auf einer Punkt-basierten Szenenrepräsentation erstellt.

Prinzip

Für 'Imperfect Shadow Maps' ist die Art der Verteilung der sekundären Lichtquellen nächst einmal irrelevant. Ob diese nun von einem 'Quasi-Random Walk' einer vollständigen Photonensimulation nach Ray-Tracing oder aus einer 'Reflective Shadow Map' in der Szene verteilt werden. Sie sollten lediglich in ihrer Zahl beschränkt sein und an Stellen positioniert sein, die den maximalen Effekt für globale Beleuchtung erzielen. Deshalb unterscheidet sich dieser spezielle, vorbereitende Schritt nicht von den vorhergehenden Techniken.

Nachdem die Punktlichtquellen verteilt sind wir ein einem einzigen Renderpass für jedes Licht eine 'Imperfect Shadow Map' erstellt. Dies stellt sicher, dass für jede Lichtquelle gespeichert wird, was von dieser aus Sichtbar ist. Dazu muss in der 'Shadow Map' allerdings ein hemisphärisches Sichtfeld entlang der Oberflächennormalen gerendert sein, da die diffuse Reflexion sich Kugelförmig aus einem Punkt ausbreitet. Das besondere dieser 'Imperfect

Shadow Maps' ist die Auflösung, welche sich im Bereich zwischen 32^2 und 256^2 Pixeln bewegt. Alle Schattentexturen werden in einer einzigen großen Textur zusammengefasst. So lassen sich 256 niedrig aufgelöste 256^2-Texturen in einer 4096^2 Pixel zählenden Textur speichern. Diese Zahlen legen nahe, dass es sich um einen extrem großen Aufwand handelt 256 mal eine Schattentextur auf der ganzen Szene zu berechnen, auch wenn die einzelnen Bilder relativ niedrig aufgelöst sind. Ritschel u. a. schlägt deshalb einen vorbereitenden Schritt vor um die Szenenkomplexität für die Schattentextur-Erstellung zu verringern. Für diese Reduzierung wird eine grobe Punktbasierte Repräsentation der Szene erstellt. Dreiecke werden zufällig, mit einer gewissen Wahrscheinlichkeit, basierend auf der eingenommenen Fläche in Bildschirmkoordinaten, ausgewählt und als Punkt in einer eigenen Datenstruktur abgespeichert. Der Punkt wird auf einer zufälligen Stelle auf diesem Dreieck in baryzentrischen Koordinaten abgenommen und mit samt des Dreiecks-Index abgelegt. Diese Art die Punkte zu speichern führt zu einer besonderen Eigenart des Verfahrens. Es funktioniert problemlos mit sich deformierender Geometrie, obwohl eine vereinfachte Szenenrepräsentation verwendet wird. Da die Koordinaten relativ zum Dreieck gespeichert sind, bleibt die Punktwolke ein korrektes Abbild der Szene.

Um aus der Punktrepräsentation eine 'Impefect Shadow Map' zu generieren wird eine Teilmenge der Punkte in den Tiefenpuffer der Grafikkarte gesetzt. Je nach quadriertem Abstand des Punktes zur entsprechenden Punktlichtquelle wird die Größe des Punktes angepasst. Dabei ist es nicht zu vermeiden, dass Punkte, die in der originalen Szene verdeckt sind in die Schattentextur aufgenommen werden, es wird also ein falscher Tiefenwert für diese Position verzeichnet. Ebenso ist es durchaus möglich, dass Texel, die eigentlich den realistischen Tiefenwert enthalten sollten, den Maximalwert der Schattentextur enthalten. Die Schattentexturen können Löcher aufweisen. Dies kann zu falschen Ergebnissen in der Beleuchtung führen, deswegen wird ein Korrektur-Schritt eingeführt, der über Bildpyramiden und Interpolation aus niedriger aufgelösten Varianten viele dieser Löcher schließt.

Abschließend wird ein reguläres 'Gathering' auf dem g-Buffer durchgeführt um die gesammelte Sichtbarkeitsinformation in indirekte Beleuchtung umzurechnen. Dabei wird für jeden Pixel eine Teilmenge an Punktlichtquellen zur Evaluation ausgewählt.

Annahmen & Einschränkungen

'Impefect Shadow Maps' erbt natürlich viele der Einschränkungen der Methode, mit derer die Punktlichtquellen in der Szene verteilt werden. So bietet sich an, auch hierfür 'Reflective Shadow Maps' zu verwenden, was die Licht-Typen auf parallele und Spot-Lichtquellen bei einer erweiterten Schattentextur und Punktlichtquellen bei Verwendung einer erweiterten 'Cube-Map' begrenzt. Die Anzahl der Punktlichtquellen ist nur in Schritten von 2^n sinnvoll, da sonst Platz auf der 'Imperfect Shadow Map' verschwendet wird, da diese in einer einzigen großen Textur gesammelt werden.

Nicht-diffuse Oberflächen sind nur eingeschränkt möglich, da eine stark spekulare Abstrahlcharakteristik durch eine komplexere Projektion des 'Imperfect Shadow Map'-Sichtfeldes simuliert werden muss. So bleibt die indirekte Beleuchtung diffusen oder nur schwach spekularen Oberflächen vorbehalten. Eine erhöhte Menge an sekundären Lichtquellen erhöht aber die Fähigkeit, höherfrequente Details der indirekten Beleuchtung darzustellen. Trotzdem bleiben Schattierungen in der indirekten Beleuchtung eher weich (für einen Vergleich siehe Bild 4.7).

Auch dieses Verfahren bietet in der Theorie die Möglichkeit, mehrere Reflexionsstufen zu simulieren. Dazu wird pro sekundärer Punktlichtquelle anstatt einer regulären Schattentextur eine 'Reflective Shadow Map' erstellt. Es wird also zusätzlich zu den Tiefenwerten weitere Informationen abgelegt, womit sich dann ein weiterer 'Bounce' der globalen Beleuchtung erstellt werden. Setzt man nun basierend auf den neuen erweiterten Schattentexturen neue Lichtquellen und rendert aus deren Position weitere 'Imperfect Shadow Maps' lassen sich auch weitere Stufen simulieren, auch wenn das mit stetig steigendem Rechenaufwand verbunden ist.

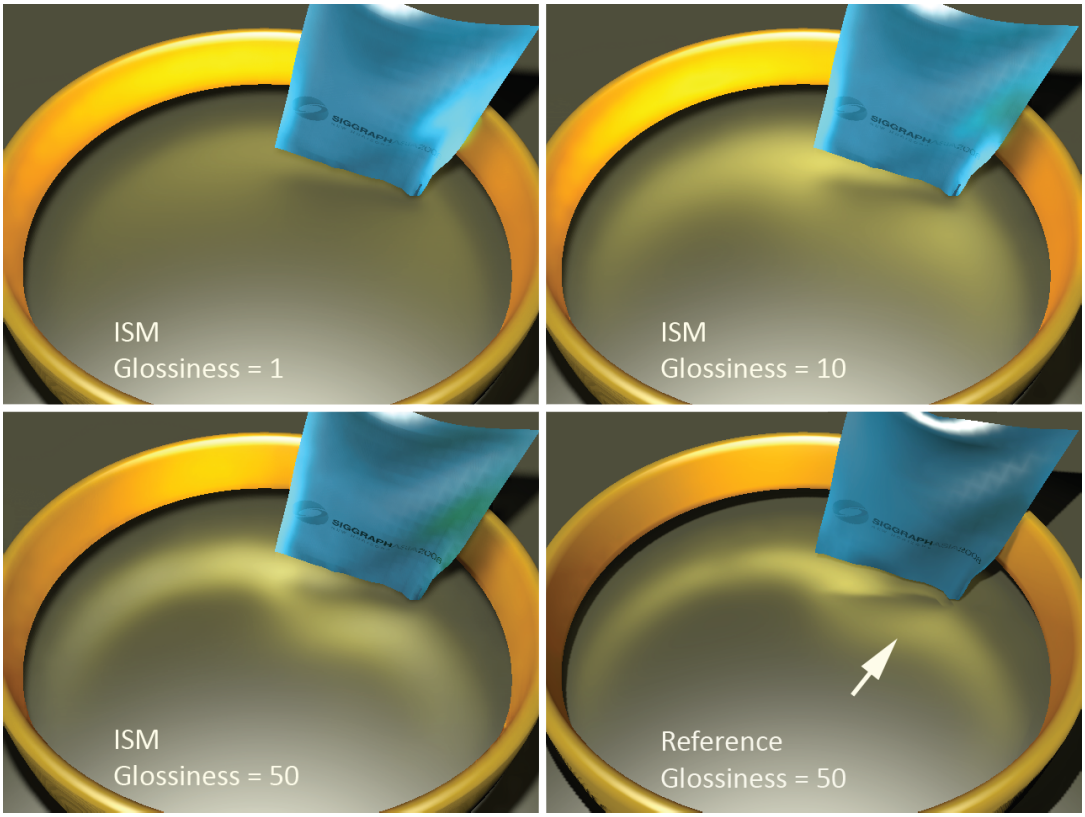

Abbildung 4.7: Vier Vergleichsbilder zeigen verschiedene 'Glossiness'-Werte und die hochfrequenten Schattendetails in der indirekten Beleuchtung. Die ersten drei Bilder zeigen verschieden intensiven Glanz des goldenen Ringes, das letzte Bild rechts unten zeigt ein Referenzbild. Die Schattendetails sind im Referenzbild deutlich schärfer, das 'Instant Radiosity'-Bild aber ist akzeptabel. Abbildung übernommen von Ritschel u. a. (2008).

Qualität & Performance

'Imperfect Shadow Maps' profitiert sehr von den nahezu korrekt ablaufenden Sichtbarkeitstests. Dem Emitter abgewandte Flächen bekommen nun keine Lichtanteile mehr ab, was den Realismus steigert. Aber auch hier spielen einige der Qualitätsabstriche der Verwendeten Technik zum Verteilen der Punktlichtquellen eine Rolle. So ist ein Flackern bei bewegter Kamera ebenfalls vorhanden wie die Artefakte bei einer zu geringen Anzahl von

sekundären Lichtquellen.

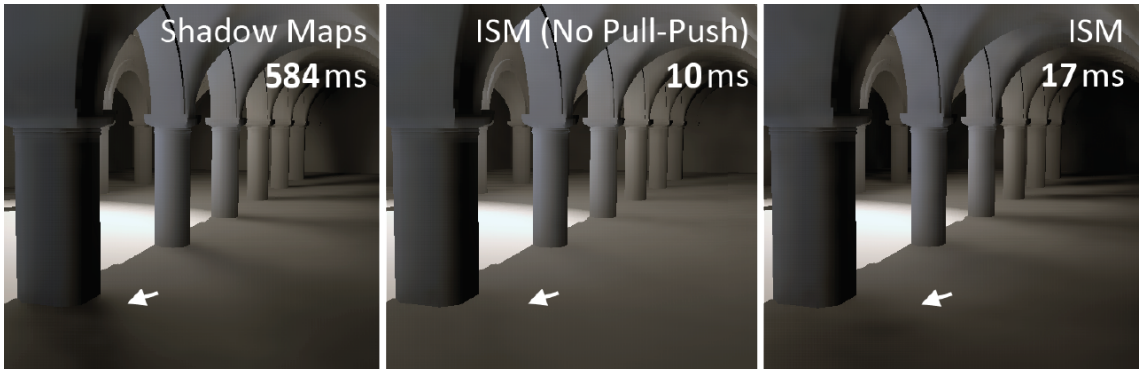

Abbildung 4.8: Ein Vergleich zwischen präzisen 'Shadow Maps', nicht rekonstruierten 'Imperfect Shadow Maps' und rekonstruierte Texturen. Im Schattenberich der Säule entstehen bei nicht rekonstruierten Schattentexturen 'Light Leaks'. Abbildung übernommen von Ritschel u. a. (2008).

Die sehr grobe Repräsentation der Szene zur Erstellung der vielen Schattentexturen führt nicht zwangsweise zu fehlerhafter Lichtsimulation. Generell gilt, dass eine Mindestanzahl an Punkten existieren muss. Diese Anzahl ist jedoch stark abhängig von der Komplexität der Szene. Feine Geometrie, wie zum Beispiel Zaunlatten, Geländer oder Säulen, können bei zu niedriger Punktzahl nicht repräsentiert werden, was zu sogenannten 'Light Leaks' führt. Das indirekte Licht läuft durch Szenengeometrie durch und erzeugt Lichtflecken im Schattenbereich dünner Objekte. Das selbe Verhalten ist zu beobachten, wenn die Schattentexturen nicht korrekt rekontruiert werden können. In einer 'Shadow Map' ist dann ein fehlerbehafteter Tiefenwert vermerkt, der dazu führt, dass der Bereich im Schatten beleuchtet wird, weil das verdeckende Objekt nicht erkannt wird (siehe Bild 4.8). Zu niedrig Aufgelöste Schattentexturen führen dazu, dass eigentlich indirekt beleuchtete Teile der Szene zu dunkel dargestellt werden. Detailreiche Geometrie wird nicht mehr zuverlässig genug erkannt um korrekt zu beleuchten.

Die in jeder einzelnen Schattentextur vorhandenen falschen Tiefenwerte äußern sich nicht in grob fehlerhaften Ergebnissen der indirekten Beleuchtung. Dadurch, dass mehrere hundert Schattentexturen erstellt werden, nivellieren sich die Fehler zuverlässig aus und es entsteht eine gute Annäherung der globalen Lichtsituation. Ist eine Szene mit diesem Algorithmus gerendert, so muss sich nicht extra um eine Schattenberechnung gekümmert werden. Das Verfahren bietet automatisch, bedingt durch die korrekten Sichtbarkeitsberechnungen, einen weichen Schattenwurf.

Die Rendergeschwindigkeit hängt von einigen Parametern ab. Für eine 'Instant Radiosity'-Technik klassisch ist einer der wichtigsten Parameter die Anzahl der Punktlichtquellen. Die Menge der Punkte in der Szenen-Repräsentation und die Auflösung der Schattentexturen haben ebenso Einfluss auf die Geschwindigkeit. Pro Lichtquelle (Anzahl N_{PL}) müssen zwischen 1.024 und 65.536 (N_{Tex}) Texel einer Schattentextur errechnet werden, dafür müssen wiederum N_P Punkte evaluiert werden. Insgesamt ergibt sich so eine Komplexität von $\mathcal{O}(N_{PL} \times (N_{Tex} + N_P))$. Dennoch erzielt das Verfahren auch in komplexen Szenen (beispielsweise die Palazzo Szene mit 380.000 Polygonen in Ritschel u. a., 2008) in-

teraktive Bildwiederholraten (5.5 Bilder pro Sekunde bei 256^2 Texturauflösung und 16.000 Punkte in der Szenenrepräsentation bei zweifacher diffuser Reflexion und einer Auflösung von 640×480 Pixeln). Das wird auch hier durch den Einsatz von 'Deferred Rendering' erreicht. Deshalb ist die Speicherbandbreite aber auch die Rohleistung ein entscheidender Flaschenhals. Rechnet man die Leistung der Test-Hardware aus Ritschel u. a., 2008 grob auf heutige Grafikhardware hoch, es handelt sich dabei um eine Nvidia 8800 GTX, so sind durchaus beeindruckende Bildwiederholraten möglich.

Relevante Eckdaten der Test-Hardware sind $518 GFLOP/Sekunde$ (FLOP = 'Floating Point Operation', deutsch 'Fließkommaoperation'), fasst Operationen auf Fließkomma-Datentypen zusammen aus denen Shader-Anweisungen zusammengesetzt sind) und $86.400 MB/Sekunde$ Speicherbandbreite. Rechnet man, wie in Kapitel 4.2, diese Daten auf eine vergleichbar positionierte Nvidia GTX 770, mit $3.213 GFLOP/Sekunde$ und $224.384 MB/Sekunde$ hoch, so ergibt sich eine theoretische Leistungssteigerung, basierend auf dem Flaschenhals der 'Imperfect Shadow Maps'-Technik und der Rohleistung, von circa 260% bis 620%. Umgelegt auf die Bildwiederholrate ergeben sich theoretische $14,3$ bis $34,1 Bilder/Sekunde$. Für eine derart komplexe Szene und zweifacher diffuser indirekter Beleuchtung ein gutes Ergebnis und nicht allzu weit von Computerspiel-typischen Szenarien entfernt. Rechnet man ein einfacheres Szenario hoch (1024 Punktlichtquellen, 4.000 Punkte, 256^2 Schattentextur, 76k Polygone und nur einfache diffuse Reflexion), welches eine Bildwiederholrate von circa $20.4 Bilder/Sekunde$ erreichte, so erhält man auf einer moderner Grafikkarte umgerechnet circa 53 bis $126,5 Bilder/Sekunde$.

4.4 Hirearchical Image-Space Radiosity

Lensch u. a. gehen bei ihrem 2008 vorgestellten Ansatz einen besonderen Weg. Ihre Arbeit basiert auf 'Multi-Resolution'-Methoden an diversen Stellen des Algorithmus um die maximale Leistung zu erreichen und dabei auch die Qualität der Bilder zu verbessern. Die Verteilung der Punktlichtquellen folgt auch hier vorrangig auf 'Reflective Shadow Maps' von Dachsbacher u. Stamminger, wobei hier Verbesserungen vorgenommen wurden. In einem 'Deferred Render'-Pass werden Stellen im Bild erkannt, die hochfrequente Details benötigen. Durch 'Stencil Refinement' werden auch nur diese Stellen dann in voller Auflösung evaluiert. Ähnlich zu 'Reflective Shadow Maps' kommt auch hier ein, wenn auch optimierter, 'Gathering'-Algortimus zum Einsatz, der die indirekte Beleuchtung simuliert.

Prinzip

Die wichtigste Neuerung des 'Hierarchical Image-Space Radiosity'-Verfahrens ist die Verwendung von 'Multi-Resolution'-Ansätzen in allen Schritten des Ablaufes. Darunter versteht man die Verwendung von 'Mip-Maps', also Bildpyramiden, die mit dem Bild in voller Auflösung beginnen und mit jeder Ebene in die Tiefe das Bild in nur zu einem Viertel aufgelöster und gefilterten Größe speichert. Dies wiederholt sich bis eine vom Nutzer festgelegte Ebenentiefe erreicht wird oder das Bild nicht weiter verkleinert werden kann.

Lensch u. a. erweiterten den Prozess zum Erstellen einer 'Reflective Shadow Map' um eine solche Bildpyramide. Grund dafür ist, dass Punktlichtquellen bei Kamera- oder Geometriebewegungen zwischen Flächen hin und her springen können, was zu einem sichtbaren Flackern führt. Also werden Punktlichtquellen in einem Quadtree ('Mip-Map') zusammen-

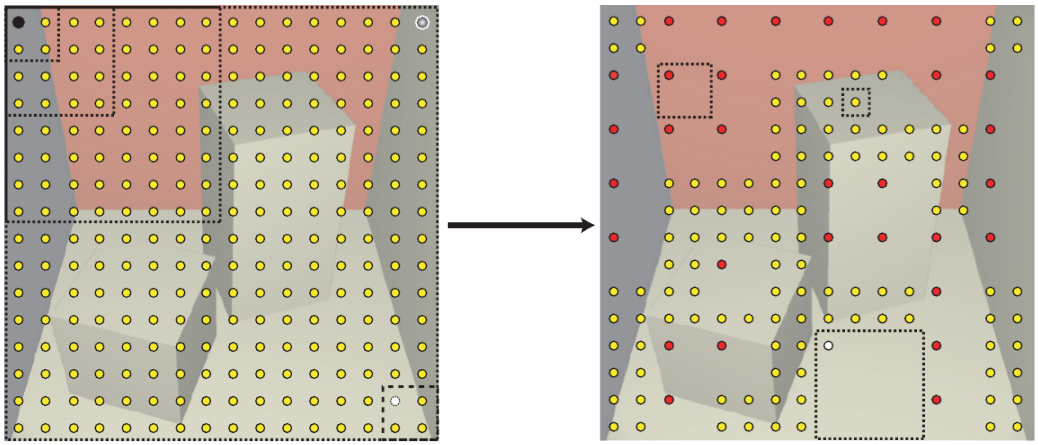

Abbildung 4.9: Auf der linken Seite sind die Punktlichtquellen (gelb) noch nicht zusammengefasst, jedoch sind verschiedene Cluster-Größen für zwei Beispielpixel eingezeichnet (die Lichtquellen sind schwarz und weiß, die Cluster mit gestrichelten Linien umrandet). Das rechte Bild zeigt zusammengefasste Pixel der Schattentextur in drei Tiefen (gelb für Stufe 0, rot für Stufe 1 und weiß für die dritte Stufe),die gestrichelten Rahmen zeigen den Einflussbereich des Lichtes auf die Szene. Abbildung übernommen von Lensch u. a. (2009).

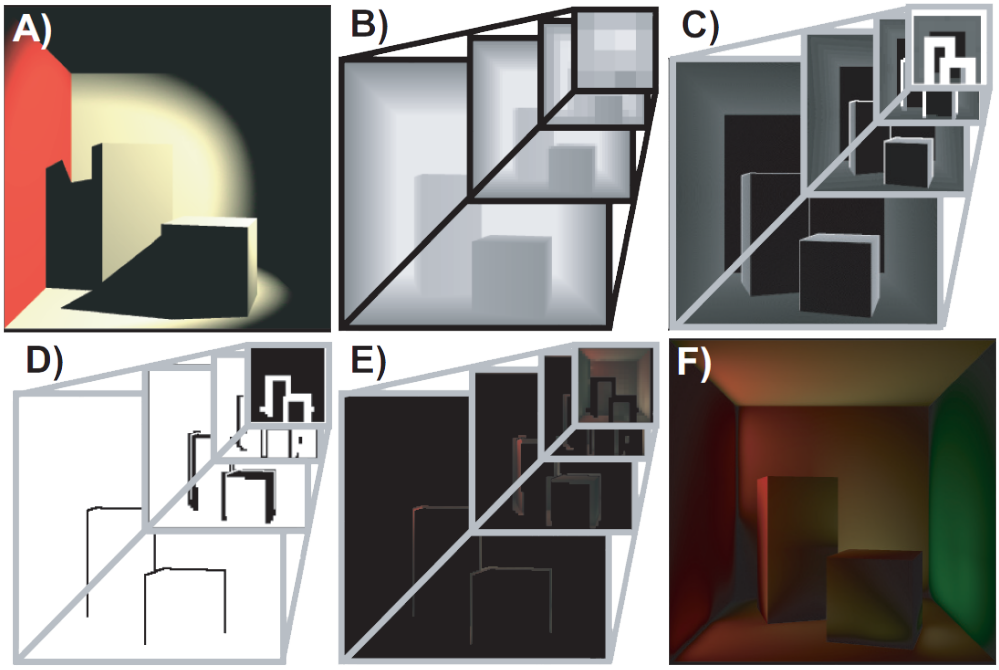

Abbildung 4.10: Das Vorgehen von 'Hierarchical Image-Space Radiosity' in sechs Schritten: Zuerst wird das direkte Licht gerendert (a), dann 'Mip-Maps' für Tiefenwerte, abgeleitete Tiefenwerte, Schablonen um hochfrequente Details und die indirekten Lichtanteile basierend auf der Schablone. Abbildung übernommen von Lensch u. a. (2009)

gefasst und damit sogenannte Licht-Patches erstellt. Dazu analysiert ein Geometry-Shader die Szene aus Sicht der erweiterten Schattentextur und sucht nach Diskontinuitäten in Tiefenwerten und Normalen. Dort, wo keine auftreten, werden die Pixel der Textur zu einem Patch in einer tiefer liegenden Ebene zusammengefasst (Bild 4.9). Dies wird bis zu einer beliebigen Tiefe der Pyramide wiederholt. Zusammengefasste Licht-Cluster werden entsprechend der beinhalteten Lichtquellen gewichtet um einen korrekten Lichtwert wiedergeben zu können.

Wird ein Pixel schlussendlich evaluiert, so wird bei der geringsten Auflösung der Lichtquellen-Bildpyramide begonnen. In Bild 4.9 sieht man, dass manche Licht-Patches kleine Teile von Diskontinuitäten überdecken können. Wird also eine höhere Genauigkeit benötigt, so wird das Cluster aufgebrochen und die nächsthöhere Ebene an dieser Stelle betrachtet. Als Kriterium für die Entscheidung wird eine Metrik benutzt, in wie weit sich der Pixel mit feiner unterteilten Pixellichtern von der groben Varianten unterscheidet. Unterschreitet diese Abweichung einen vorgegeben Grenzwert, so wir keine Verfeinerung vorgenommen. Mathematisch ausdrücken lässt sich das Kriterium wie folgt:

$$\frac{2(l-\tau)}{3\tau * RSM_{res}^2} \| V_j \|^2. \tag{4.4}$$

τ ist der Benutzerdefinierte Schwellwert, RSM_{res} die Auflösungs-Kantenlänge der erweiterten Schattentextur und V_j der Vektor zwischen von Lichtpatch j zur Lichtquelle. Diese recht einfache Metrik und der Fakt, dass die volle Anzahl an Lichtquellen nur evaluiert wird, wenn diese auch benötigt werden, führt zu einer gesteigerten Rendergeschwindigkeit.

Ein ähnliches Vorgehen zeigt sich bei der Erstellung von sogenannten 'Screen-Space Patches' (Bild 4.11). Wie in Bild 4.10 dargestellt, werden hier gleich mehrere 'Mip-Maps' eingesetzt. In einem Render-Pass werden neben der direkten Beleuchtung Bildpyramiden für Tiefenwerte und deren erster Ableitung gerendert. Aus dieser Ableitung lassen sich Diskontinuitäten lokalisieren. Der Wert in der Textur nimmt an solchen Stellen den Wert 255 (bei einer 8-Bit-Textur) an. Daraus lässt sich eine Schablonen-Textur erstellen, die später beim Evaluieren nur eine Hierarchie-Ebene von Bildschirm-Patches betrifft. In dieser Textur wird das Schablonen-Bit nur gesetzt, wenn das entsprechende Pixel nicht auf einer Diskontinuität liegt, aber das Element in der nächst-gröberen Ebene. So wird nur vermerkt, wo wirklich in dieser Auflösung der Ebene evaluiert werden muss.

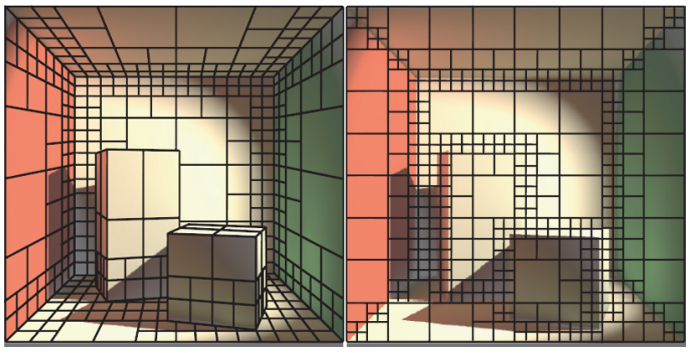

Abbildung 4.11: Ein Vergleich zwischen Patches erstellt auf Weltkoordinaten (links) versus Patches erstellt auf Bildschirmkoordinaten (rechts). Abbildung übernommen von Lensch u. a. (2009).

Der Evaluationsprozess der indirekten Beleuchtung erinnert mit Bildschirm- und Licht-Patches nun an den klassischen 'Radiosity'-Algorithmus. Benutzt wird eine 'disc-to-point'-

Projektion, es wird also eine Scheibenfläche auf einen Punkt projiziert (Formel 4.5). Allerdings wird die Sichtbarkeit zwischen Patches nicht berücksichtigt.

$$F_{i \to j} \;=\; \frac{A_j (N_i \, V_{ij})(N_j \, V_{ji})}{\pi \| \, V_{ij} \|^2 + A_j} \tag{4.5}$$

Darin gibt A die Fläche eines Patches an, N die jeweiligen Normale und V Sichtbarkeits-Vektoren zwischen Patches oder Licht-/Kameraposition. i ist der Bildschirm-Patch, j demnach der aktuelle Licht-Patch. Der indirekt eintreffende Lichtfluss eines Bildschirm-Patches i ergibt sich aus der diffusen Farbe p des Bildschirm-Patches und der Summe aller Intensitäten der indirekten Lichtquellen I_j, dessen diffuser Farbe p und der Scheiben-zu-Punkt-Annäherung $F_{i \to j}$ in Formel 4.6

$$C_i = p_i \sum_j I_j p_j F_{i \to j} \tag{4.6}$$

Die Fläche eines Licht-Patches lässt sich ungefähr errechnen, indem der Abstand des Patches zur Lichtquelle mit dem Raumwinkel ω_j des Patches, also der Strahlungsrichtung des Lichts, multipliziert wird (Formel 4.7).

$$A_j \;\approx\; \omega_j \| \, V_j \|^2 \tag{4.7}$$

Sind alle indirekten Lichtanteile in der Bildpyramide gespeichert, werden anschließend die Ebenen der Bildpyramide interpoliert und akkumuliert und mit der direkten Beleuchtung verrechnet. Die am niedrigsten aufgelöste Ebene stellt dabei die indirekte Beleuchtung großer Flächen zur Verfügung, mit jeder höheren Ebene kommen weitere Details hinzu bis ein vollständiges Bild entsteht.

Annahmen & Einschränkungen

'Hierarchical Screen-Space Radiosity' ist ein Verfahren für diffuse indirekte Beleuchtung. Das rührt daher, dass beim Sammeln die Lichtanteile pro Patch, wie in Formel 4.6 zu sehen, nicht entsprechend einer Reflexionsrichtung abgeschwächt werden. Somit reflektieren diffuse Oberflächen immer ideal kugelförmig. Wenn man den Term beispielsweise um ein BRDF-Term eines in der erweiterten Schattentextur vermerkten Materials erweitert, würde spekulare Reflexion möglich. Somit kann entsprechend der Einfallsrichtung des indirekten Lichts entschieden werden, ob es auf einem Bildschirm-Patch ankommt oder nicht.

In der Arbeit von Lensch u. a. wird keine mehrfache diffuse Reflexion simuliert, es bleibt bei der Annäherung der ersten diffusen indirekten Lichtanteile. Weiterhin werden durch das Verfahren kein Schattenwurf der Szene berechnet. Eine Simulation davon muss also zusätzlich in beliebiger Form hinzugefügt werden, sind die relevanten Informationen dafür doch in der erweiterten Schattentextur schon gespeichert. Durch die fehlende Selbstabschattung wird jede Fläche in der Szene von indirektem Licht getroffen, was in Extremsituationen zu sichtbar falschen Ergebnissen führen kann. In den meisten Fällen führt diese unkorrekte Lichtberechnung aber nicht zu einem gestörten subjektiven Empfinden.

Analog zu 'Reflective Shadow Maps' lässt sich das Verfahren auch auf Flächen- und Punktlichter anwenden. Die dazu erforderliche Anzahl an zusätzlichen Bildpyramiden würden den Rechenaufwand allerdings stark in die Höhe treiben. Durch das Interpolieren niedrig Aufgelöster Pyramidenstufen für die indirekte Beleuchtung auf größeren Flächen

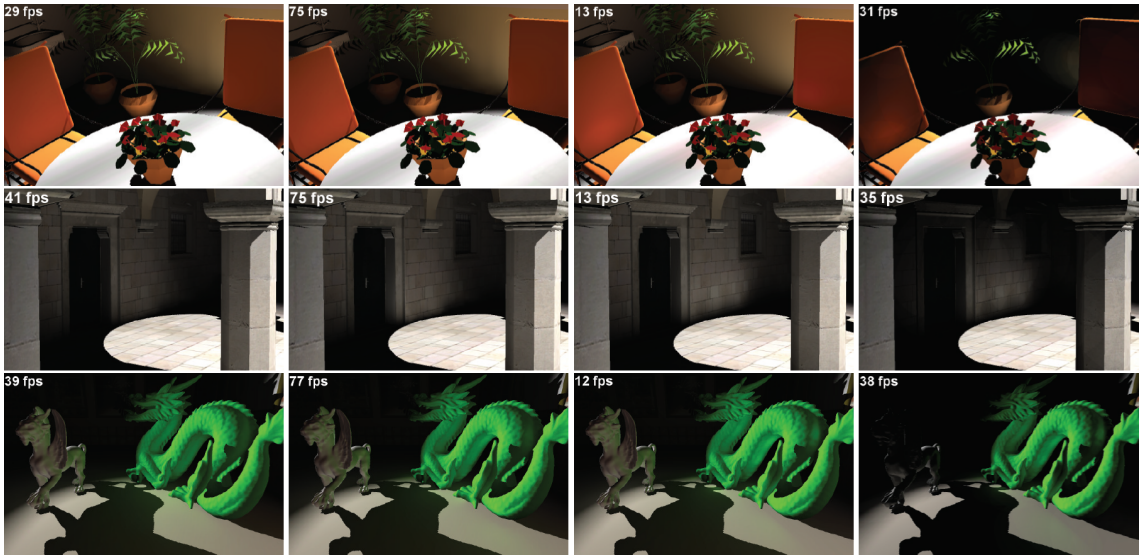

Abbildung 4.12: Drei verschiedene Szenen mit zwei verschiedenen Verfahren in jeweils zwei Ausführungen. Von links nach rechts sind diese : 'Hierachical Image-Space Radiosity' ohne Schablonen, 'Hierachical Image-Space Radiosity' mit Schablonen, 'Splatting Indirect Illumination' mit bildschirmfüllenden 'Splat'-Vierecken und 'Splatting Indirect Illumination' mit kleineren 'Splat'-Vierecken. Grobe Qualitätsunterschiede sind in den ersten drei Methoden nicht zu erkennen, wohl aber in den Bilder der letzten Spalte. Dort ist klar zu erkennen, dass die Bereiche von einzelnen Punktlichtquellen beschränkt sind, was zu einigen dunklen Stellen führt, die eigentlich indirektes Licht abbekommen müssten. Abbildung übernommen von Lensch u. a. (2009).

verhindert nahezu komplett die Existenz hochfrequenter Details, da deren Information nur aus sehr wenigen Pixeln interpoliert wird.

Qualität & Performance

Das Zusammenfassen und spätere Interpolieren von Lichtquellen führt zu einer starken Reduzierung des von anderen Methoden bekannten Flackern, das von re-positionierten Lichtquellen herrührt. Die Pixellichter bei 'Hierachical Image-Space Radiosity' sammeln sich an Kanten der Szenengeometrie. Da dadurch die Anzahl und Genauigkeit an jenen Kanten höher als im Rest der Szene ist, sind die Position der Lichter von Bild zu Bild konsistenter. Wenn es trotzdem von einer Fläche zu einer anderen überspringen sollte ist dessen Einfluss begrenzt, da sich in direkter Umgebung höchstwahrscheinlich noch einige weitere fein aufgelöste Lichtquellen befinden.

Nachteilig auf die Qualität wirkt sich aus, dass die Schatten nicht beim Berechnen der indirekten Beleuchtung erstellt werden. Korrekter Schattenwurf muss separat berechnet werden. Die fehlende Selbstabschattung der indirekten Lichtanteile führt dazu, dass eigentlich der Punktlichtquelle abgewandte Flächen dessen Licht erhalten, statt im Schatten zu liegen. Diese Qualitätseinbußen lassen sich zum Beispiel mit 'Ambient Occlusion'-Techniken reduzieren.

Die Leistung des Algorithmus ist abhängiger von der Szenengeometrie als vorhergehende Ansätze, weil nur an geometrisch komplexen Stellen voll evaluiert wird. Des weiteren muss zum hierarchischen Verteilen der Lichtquellen ein Durchgang auf dem Geometry-Shader durchgeführt werden. Das führt zu einer sinkender Bildwiederholrate, wenn aus Kamerasichten gerendert werden muss, in der sehr viele feine geometrische Details sichtbar sind. Es kann dann weniger Information aus niedrigen Auflösungen hergenommen werden, mehr Stellen müssen in voller Auflösung evaluiert werden und Punktlichtquellen können nicht mehr so effektiv zusammengefasst werden, weswegen insgesamt mehr Pixel mit mehr Lichtquellen berechnet werden müssen.

Bild 4.12 zeigt den Geschwindigkeitsvergleich zwischen jeweils zwei Ausprägungen von 'Hierachical Image-Space Radiosity' und 'Splatting Indirect Illumination'. Die erste Zeile zeigt eine Szene mit feinen Details der Pflanzen und Armlehnen der Stühle. Das erklärt den deutlichen Geschwindigkeitsunterschied von 'Hierachical Image-Space Radiosity' (in der ersten Spalte ohne die hierarchischen Schablonen um hochfrequenten Geometriedetails) zu den beiden anderen Szenen und Techniken. Im Vergleich der Techniken lässt sich erkennen, dass das durch hierarchische Schablonen verfeinerte Verfahren deutliche Geschwindigkeitszuwächse ermöglicht, selbst die feinen Details der Pflanzen wirkt sich nicht mehr stark auf die Performanz aus. 'Splatting Indirect Illumination' hingegen ist weniger Geometrieabhängig, erzielt aber bei Evaluation aller Pixellichter eine sehr geringe Bildwiederholrate. Grenzt man den Evaluationsbereich ein, so zeigt sich eine deutliche Steigerung, die allerdings zu Lasten der Qualität geht.

Zur Vergleichbarkeit zwischen den Methoden herzustellen, wird auch hier der Vergleich zu einer Modernen Nvidia Geforce GTX 770 gezogen. Als Ausgangsbasis dienen hier Ergebnisse aus Bild 4.12 (Spalte 2). Diese wurden laut Lensch u. a. auf einer Nvidia Geforce GTX 280 gerendert. Deren relevante Eckdaten sind $933\,GFLOP/Sekunde$ Rohleistung und $141.696 MB/Sekunde$ Speicherbandbreite Andermahr (2008). Da die Texturzugriffe nicht mehr das hauptsächlich Begrenzende Element sind, wird stattdessen auf Basis der Rohleistung hochgerechnet. Die relevanten Daten der Referenzhardware sind $3.213\,GFLOP/Sekunde$ Rohleistung und $224.384 MB/Sekunde$ Speicherbandbreite. 'Hierachical Image-Space Radiosity' mit Schablonen-Verfeinerung liefert zwischen 75 und $77\,Bilder/Sekunde$ bei 1024^2 Pixeln. Umgerechnet auf die GTX 770 ergibt sich so eine Leistungssteigerung von circa 344% und eine theoretische Bildwiederholrate von circa 258 und $265\,Bilder/Sekunde$. Der Vergleich zu 'Reflective Shadow Maps' ist der hier beschriebene Ansatz bei besserer Qualität und höherer Auflösung ähnlich schnell. Die Optimierungen im 'Gathering' haben sich also bezahlt gemacht. Der Vergleich mit 'Imperfect Shadow Maps' ist deutlich schwieriger, da dessen Ergebnisse auf einer deutlich komplexeren Szene (600k gegen circa 60k bis 70k Polygone) mit zweifacher diffuser Reflexion und korrektem Schattenwurf bei direkter und indirekter Beleuchtung erstellt wurden. Vergleicht man mit einem einfacheren Szenario so zeigt sich, dass 'Hierachical Image-Space Radiosity' zwar der schnellere Ansatz ist, aber nicht an die Qualität der imperfekten Schattentexturen herankommt.

5 Zusammenfassung

Im Laufe dieser Arbeit wurde der originale 'Instant Radiosity'-Algorithmus von Alexander Keller aus dem Jahre 1997 vorgestellt und dessen Funktionsweise beschrieben. Es wurden die von Keller vorgenommenen Vereinfachungen dargelegt und die Auswirkungen davon. Ebenfalls wurde detailliert auf grundlegende Techniken eingegangen, welche von der Gruppe der 'Instant Radiosity'-Algorithmen benutzt werden. 'Instant Radiosity' verteilt durch eine Photonensimulation Punktlichtquellen und rendert für jedes Licht eine Schattentextur. Schlussendlich werden alle Schattentexturen kombiniert.

Jeder der drei den originalen Algorithmus erweiternden Ansätze wurde in Bereichen der Funktionsweise, Einschränkungen, die die Methode vornimmt, und Qualitäts- und Geschwindigkeitsaspekten beleuchtet. 'Reflective Shadow Maps' verteilt Punktlichter basierend auf in der Schattentextur gespeicherten Lichtflusswerten der Szene. Beim 'Einsammeln' des Lichts wird pro Pixel eine Teilmenge dieser Punktlichter ausgewertet und der Punkt entsprechend beleuchtet. Dies beschleunigt die aufwändige Photonensimulation des originalen Ansatzes enorm auf Kosten der Speicherbandbreite und -platzes der Grafikkarte. 'Imperfect Shadow Maps' setzt sich zum Ziel, die Punkt-zu-Punkt Sichtbarkeit, welcher im originalen Algorithmus per voll aufgelöster Schattentexturen passiert, zu beschleunigen. Dazu wird eine grobe Punktrepräsentation der Szenengeometrie errechnet. Mit der werden dann mehrere hundert Schattentexturen für jedes Punktlicht gerendert. Dies beschleunigt diese Tests um mindestens eine Größenordnung. 'Hierarchical Screen-Space Radiosity' beruhigt die Lichtverteilung durch hierarchisches Sampling der Punktlichtquellen, durch Verwendung von Bildpyramiden und Stencil-Puffern werden nur Stellen in voller Auflösung evaluiert, wo Geometrie hochfrequente Details aufweist. Wo große Flächen existieren wird die indirekten Beleuchtung aus niedrig aufgelöster Ebenen der Bildpyramide interpoliert.

5.1 Meinung des Autors

Globale Beleuchtung realistisch in Echtzeit zu simulieren ist eine durchaus ambitionierte Aufgabe. Integralprobleme lassen sich in akzeptabler Geschwindigkeit mehr oder minder genau annähern. Mit diesem Problem kämpft auch 'Instand Radiosity'. Da in der Echtzeit-Computergrafik fast ausschließlich Polygonalmodelle eingesetzt werden, ist die Patch-Repräsentation also implizit schon abgeschlossen. Um die indirekte Beleuchtung nun anzunähern wird die clevere Idee umgesetzt, in der Szene virtuelle Punktlichtquellen zu verteilen, die ihrerseits die Szene beleuchten. Allerdings ist der folgende Schritt extrem aufwändig. Es wird aus der Sicht jedes Lichts die Tiefe der gesamten Szene gerendert. Da dies durchaus Minuten dauern kann ist klar. Also versuchen die weiteren Verfahren an verschiedenen Stellen durch clever gewählte Techniken und Annahmen die Geschwindigkeit zu verbessern.

Zwei der drei Ansätze ignorieren die Punkt-zu-Punkt Sichtbarkeit in der Szene, dadurch wird indirekte Beleuchtung in der Gesamten Szene verteilt. Die dadurch entstehenden

Fehler sind in machen Fällen deutlich sichtbar. Dass zwei Flächen, die gegenseitig nicht sichtbar sind, trotzdem Licht austauschen, stört mich besonders in der farbigen 'Cornel Box'-Szene. Doch ein weißer Würfel zwischen einer grünen und einer roten Fläche ist natürlich so ein Extremfall. In texturierten Szenen fällt dieser Umstand allerdings deutlich weniger stark aus und stört so in der Regel nicht mehr.

Ohne 'Reflective Shadow Maps' wäre Echtzeit-'Radiosity' meiner Meinung nach in der jetzigen Form nicht möglich. Das jeder Pixel in der Schattentextur als Lichtquelle interpretiert wird verschiebt die Berechnungen auf die Grafikkarte und man hat die Möglichkeit, aus den vielen Lichtquellen mit einem geeigneten Sampling-Verfahren die idealen Auszusuchen.

Auch 'Imperfect Shadow Maps' sind ein hochinteressanter Ansatz. Nicht mehr die volle Szene in jeder Schattentextur zu rendern, sondern stattdessen nur eine grobe Repräsentation dafür zu verwenden. Somit werden die so wichtigen Sichtbarkeiten zwischen Punktlichtquelle und Szene extrem beschleunigt. Die Qualität der automatisch von diesem Verfahren mit erstellten Schatten hat mich tief beeindruckt. Selbst bei feiner Geometrie (genug Punkte in der Repräsentation vorausgesetzt) bleiben die Schatten korrekt. Dass das alles auch bei voll deformierbarer Geometrie, flexiblen Lichtern und Kamera-Positionen funktioniert ist umso beeindruckender. Selbst mit zweifacher diffuser Reflexion in einer komplexen Szene sind ohne weiteres interaktive Bildwiederholraten möglich.

Sollte es möglich sein, 'Imperfect Shadow Maps' mit den 'Hierarchical Screen-Space Radiosity' zu verbinden, wären schöne Ergebnisse in hoher Geschwindigkeit erreichbar. Dazu könnten die imperfekten Schattentexturen auch als 'Mip-Maps' vorliegen. wird nun ein großflächiger Patch evaluiert, könnte man auch bei den Schattentexturen der indirekten Lichter eine grobe Auflösungsebene verwenden. Nur für die fein zu renderden Punkte werden auch die Schattentexturen in voller Auflösung evaluiert. Da auch die Lichtquellen hierarchisch in einem Quadtree verteilt sind, sollten sich dadurch hochqualitative Ergebnisse erzielen lassen.

Abschließend ist generell zu sagen, dass die Rendergeschwindigkeiten noch nicht hoch genug sind, um in AAA-Computerspielen zum Einsatz zu kommen. Es ist mir keine Engine bekannt, die explizit 'Instant Radiosity'-Techniken Benutzt. Globale Beleuchtung ist in den meisten Fällen immer noch zu Rechenintensiv um mit Echtzeit-Bildwiederholraten eingesetzt werden zu können. Die Zeit wird zeigen, wie lange das noch der Fall sein wird.

Literaturverzeichnis

[Andermahr 2006] Andermahr, Wolfgang: *Test: nVidia GeForce 8800 GTX*. http://www.computerbase.de/artikel/grafikkarten/2006/test-nvidia-geforce-8800-gtx/, November 2006. – Aufgerufen am 22.06.2013

[Andermahr 2008] Andermahr, Wolfgang: *Test: Nvidia GeForce GTX 280 (SLI)*. www.computerbase.de/artikel/grafikkarten/2008/test-nvidia-geforce-gtx-280-sli/, Juni 2008. – Aufgerufen am 24.06.2013

[Andermahr 2013] Andermahr, Wolfgang: *Nvidia GeForce GTX 770 im Test*. http://www.computerbase.de/artikel/grafikkarten/2013/nvidia-geforce-gtx-770-im-test/, Mai 2013. – Aufgerufen am 22.06.2013

[Dachsbacher u. Stamminger 2005] Dachsbacher, Carsten ; Stamminger, Marc: Reflective shadow maps. In: *SI3D*, 2005, S. 203–231

[Dachsbacher u. Stamminger 2006] Dachsbacher, Carsten ; Stamminger, Marc: Splatting indirect illumination. In: *SI3D*, 2006, S. 93–100

[Dutre u. a. 2003] Dutre, Phil ; Bekaert, Philippe ; Bala, Kavita: *Advanced Global Illumination, 1st Edition*. Natick, MA : A K Peters, 2003

[Goral u. a. 1984] Goral, Cindy ; Torrance, Kenneth ; Greenberg, Donald ; Battaile, Bennett: Modeling the interaction of light between diffuse surfaces. In: *Proceedings of the 11th annual conference on Computer graphics and interactive techniques*. New York, NY, USA : ACM, 1984 (SIGGRAPH '84). – ISBN 0–89791–138–5, S. 213–222

[Keller 1997] Keller, Alexander: *Instant Radiosity*. 1997

[Lensch u. a. 2009] Lensch, Hendrik P. A. ; Sloan, Peter pike ; Nichols, Greg ; Shopf, Jeremy ; Wyman, Chris: Hierarchical Image-Space Radiosity for Interactive Global Illumination, 2009

[Nischwitz u. a. 2011] Nischwitz, A. ; Fischer, M. ; Haberäcker, P. ; Socher, G.: *Computergrafik und Bildverarbeitung: Band I: Computergrafik*. Vieweg+Teubner Verlag, 2011 (Computergrafik und Bildverarbeitung / Alfred Nischwitz). – ISBN 9783834813046

[Pharr u. Humphreys 2004] Pharr, M. ; Humphreys, G.: *Physically Based Rendering: From Theory to Implementation*. Elsevier Science, 2004 (The Morgan Kaufmann series in interactive 3D technology). – ISBN 9780080538969

[Ritschel u. a. 2012] Ritschel, Tobias ; Dachsbacher, Carsten ; Grosch, Thorsten ; Kautz, Jan: The State of the Art in Interactive Global Illumination. In: *Comput. Graph. Forum* 31 (2012), Februar, Nr. 1, S. 160–188. – ISSN 0167–7055

[Ritschel u. a. 2008] Ritschel, Tobias ; Grosch, Thorsten ; Kim, Min H. ; Seidel, Hans-Peter ; Dachsbacher, Carsten ; Kautz, Jan: Imperfect Shadow Maps for Efficient Computation of Indirect Illumination. In: *ACM Trans. Graph. (Proc. of SIGGRAPH ASIA 2008)* 27 (2008), Nr. 5

[Saito u. Takahashi 1990] Saito, Takafumi ; Takahashi, Tokiichiro: Comprehensible Rendering of 3-D Shapes, 1990, S. 197–206

[Spille u. Becker 2004] Spille, Carsten ; Becker, Christoph: *Test: X800 Pro, XT PE und GeForce 6800 GT, Ultra.* `http://www.computerbase.de/artikel/grafikkarten/2004/test-x800-pro-xt-pe-und-geforce-6800-gt-ultra/`, Juni 2004. – Aufgerufen am 20.06.2013

[Williams 1978] Williams, Lance: Casting curved shadows on curved surfaces. In: *Computer Graphics* (1978), S. 270–274

Abbildungsverzeichnis

2.1 Fotografie mit eingezeichneten Lichtwegen 5
2.2 Diffuse und spekulare Reflexionscharakteristik 6
2.3 Rendering-Gleichung bildlich dargestellt 8
2.4 Radiosity Prinzip 9

3.1 Errechnung von π mit 'Monte-Carlo'-Verfahren 15
3.2 'Hierarchical Warping' 17
3.3 Nicht ideale Sampling-Typen 18
3.4 Errechnung der Diskrepanz 19
3.5 'Instant Radiosity' Prinzip 20
3.6 'Instant Radiosity' Resultate 21

4.1 'Shadow Mapping' Prinzip 24
4.2 'Deferred Shading' Prinzip 25
4.3 'Reflective Shadow Mapping Screen-Space Interpolation' 27
4.4 'Reflective Shadow Mapping Screen-Space Interpolation Evaluation' 27
4.5 'Splatting Indirect Illumination' für spekulare Reflexion und Refraktion . . 31
4.6 Vergleich 'Splatting Indirect Illumination' und 'offline Radiosity' 31
4.7 Details in indirekter Beleuchtung mit 'Imperfect Shadow Maps' 34
4.8 Light Leaks mit 'Imperfect Shadow Maps' 35
4.9 Licht-Sampling von 'Hierarchical Image-Space Radiosity' 37
4.10 Vorgehen von 'Hierarchical Image-Space Radiosity' 37
4.11 'Screen-Space Patches' von 'Hierarchical Screen-Space Radiosity' 38
4.12 Qualitätsvergleich 'Hierarchical Image-Space Radiosity' 40