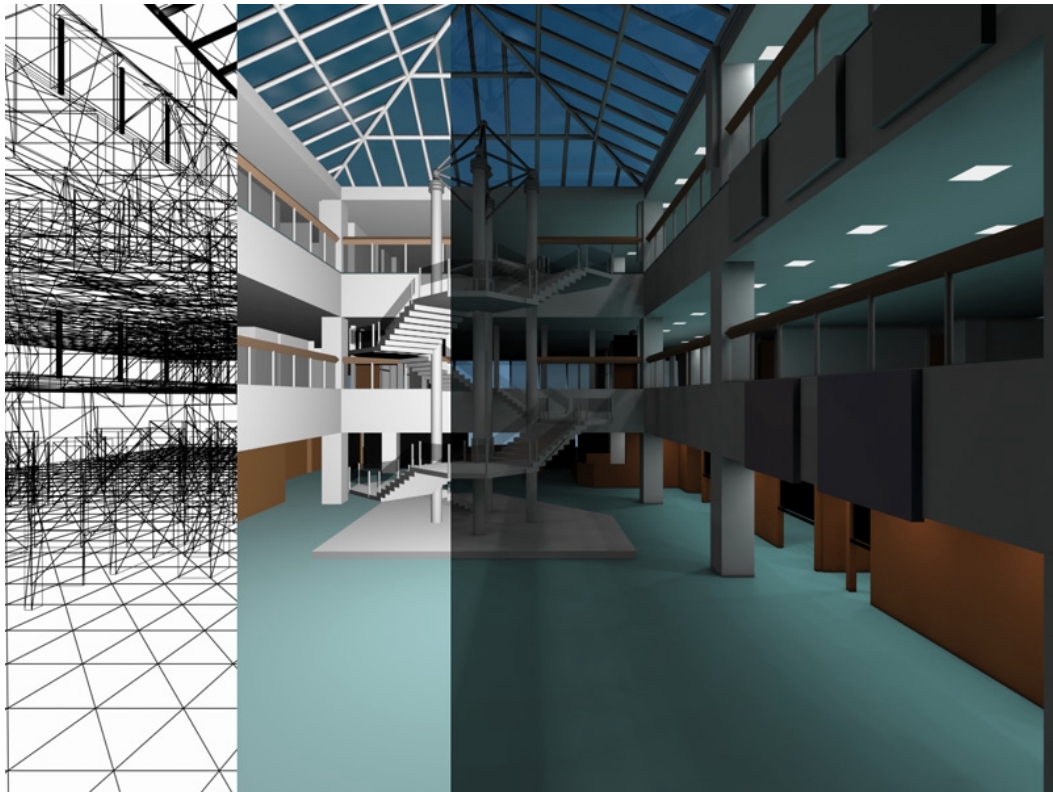


Parallele Simulation der globalen Beleuchtung in komplexen Architekturmodellen



Dissertation **von** **Olaf Schmidt**

Schriftliche Arbeit zur Erlangung des Grades
eines Doktors der Naturwissenschaften

Fachbereich Mathematik/Informatik
Universität-Gesamthochschule Paderborn

Paderborn, im Dezember 2000

Danksagung

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter im Fachbereich Mathematik/Informatik an der Universität-Gesamthochschule Paderborn. Ich möchte mich an dieser Stelle bei all jenen bedanken, die zum Gelingen dieser Arbeit beigetragen haben.

Mein Dank gilt Prof. Dr. Burkhard Monien für die Betreuung dieser Arbeit. Bei meinen Kollegen der Arbeitsgruppe Monien möchte ich mich für das angenehme Arbeitsklima bedanken, in dem diese Arbeit entstanden ist. Des weiteren bedanke ich mich bei Monika Lammers für ihr aufmerksames Korrekturlesen.

Einen besonderen Dank möchte ich auch an meine Eltern richten, die mich immer unterstützen und die meine Ausbildung ermöglichten.

Schließlich möchte ich mich herzlich bei meiner Frau Bianca bedanken, die mir in den vergangenen Jahren den Rücken gestärkt und mich stets bei meinem Tun unterstützt hat.

Vielen herzlichen Dank!

Paderborn, im Dezember 2000

Olaf Schmidt

Inhaltsverzeichnis

Kapitel 1 Einleitung	1
1.1 Prozeß der Bildgenerierung	2
1.2 Beleuchtungsmodelle in der Computergrafik	4
1.3 Parallelverarbeitung und Computergrafik	8
1.4 Erzielte Resultate	11
1.5 Überblick	13
Kapitel 2 Beleuchtungssimulation in der Computergrafik	15
2.1 Physikalische Grundlagen der Beleuchtungssimulation	16
2.1.1 Strahlungsphysikalische Größen	17
2.1.2 Lichttechnische Größen	22
2.1.3 Strahlungsaustausch zwischen Oberflächen	24
2.2 Lokale Beleuchtungsmodelle	28
2.3 Lokale Beleuchtungsalgorithmen	32
2.3.1 Gouraud-Schattierungsalgorithmus	32
2.3.2 Phong-Schattierungsalgorithmus	33
2.4 Farben	34
2.5 Ein Modell für die globale Beleuchtungsberechnung	35
2.6 Abschließende Bemerkungen	37
Kapitel 3 Die Radiosity-Methode in statischen Szenen	39
3.1 Grundlagen der Radiosity-Methode	40
3.1.1 Das Radiosity-Gleichungssystem	40
3.1.2 Bestimmung von Formfaktoren	44
3.2 Progressive-Refinement-Radiosity	56
3.3 Netzgenerierung (Meshing)	58
3.3.1 Initiale Netzgenerierung	59
3.3.2 Adaptive Verfeinerung von Patches	63
3.4 Darstellen der Radiosity-Lösung	66
3.5 Die Radiosity-Methode und spiegelnde Oberflächen	67
3.5.1 Einfache Zwei-Phasen-Methode	67
3.5.2 Vollständige Zwei-Phasen-Methode	69
3.6 Abschließende Bemerkungen	70
Kapitel 4 Radiosity in dynamisch veränderbaren Szenen	71
4.1 Radiosity-Verfahren zur Unterstützung dynamischer Szenen	72
4.2 Repropagation vs. Redistribution	73
4.3 Inkrementelle Radiosity-Berechnungen	74
4.3.1 Veränderungen von Materialattributen	75
4.3.2 Veränderung der Szenengeometrie	77
4.3.3 Berechnung inkrementeller Formfaktoren	79

4.3.4	Geometry-Queue	79
4.3.5	Auswahl des Shooting-Patches	80
4.3.6	Das Verfahren im Überblick	81
4.4	Abschließende Bemerkungen	82
Kapitel 5 Parallelisierung der Radiosity-Methode		85
5.1	Vergleich existierender paralleler Verfahren	86
5.2	Ein datenparalleles Radiosity-Verfahren auf Basis von Hemicube-Bitmaps	91
5.2.1	Statische Lastverteilung	92
5.2.2	Lokale Radiosity-Berechnungen innerhalb der Teilszenen	101
5.2.3	Energietransfer zwischen Teilszenen	103
5.2.4	Ablauf der datenparallelen Radiosity-Berechnungen	118
5.2.5	Performanz der datenparallelen Radiosity-Berechnungen	120
5.2.6	Dynamischer Lastausgleich	129
5.2.7	Dynamische Szenen	141
5.2.8	Berücksichtigung von Spiegelflächen	150
5.3	Paralleles Mehrphasen-Verfahren zur Simulation komplexer Beleuchtungseffekte	154
5.3.1	Das grundlegende sequentielle Progressive-Refinement-Verfahren	155
5.3.2	Parallelisierung der Progressive-Refinement-Methode	172
5.3.3	Parallele Radiosity-Berechnung in dynamischen Szenen	192
5.3.4	Datenparallele Berechnung von erweiterten Formfaktoren	201
5.3.5	Blickpunktabhängige Spiegelungen	206
Kapitel 6 Systemintegration		213
6.1	Architektur und Komponenten des Bildgenerierungssystems	213
6.2	Beleuchtungssimulation und Bildgenerierung als Dienstleistung in IP-Netzwerken	215
6.2.1	Architektur des Remote-Rendering-Systemes	217
6.2.2	Unterstützte Anwendungsszenarios	219
6.2.3	Bewertung des Rendering-Services	223
Kapitel 7 Zusammenfassung der erzielten Ergebnisse		225
7.1	Zusammenfassung	225
7.2	Repräsentative Visualisierungen	227
7.2.1	Konferenzraum	227
7.2.2	Wohnzimmer	227
7.2.3	Rosenthaler-Hof	228
7.2.4	Universitätsgebäude in der Fürstenallee	229
7.2.5	Sudanesischer Löwentempel	229
7.2.6	Kinokomplex	230
7.2.7	Dom zu Wetzlar	230
7.2.8	Jagdschloß <i>Platte</i> in Wiesbaden	231
Literaturverzeichnis		233

Kapitel 1

Einleitung

Just a few years ago, all one needed to be a competent researcher or practitioner in computer graphics was a solid background in geometry, algebra, calculus, topology, probability, mechanics, electromagnetism, signal processing, image processing, electrical engineering, mechanical engineering, optics, information theory, structured programming, basic algorithms and data structures, complexity theory, computer architecture, human factors, perceptual psychology, colorimetry, graphic design, industrial design, semiotics, and art!

Unfortunately, the list is growing.

(Joe Marks, SIGGRAPH 1994)

Seit ihren Anfängen in den sechziger Jahren hat die Computergrafik einen sehr weiten Weg zurückgelegt und sich zu einem wesentlichen Forschungsschwerpunkt innerhalb der Informatik entwickelt. Computergenerierte Bilder werden heutzutage in vielen verschiedenen Bereichen eingesetzt und sind praktisch Bestandteil des täglichen Lebens. Sie werden zu Unterhaltungs- und Ausbildungszwecken sowie als Planungs- und Entscheidungshilfen verwendet. Zu den wohl populärsten Anwendungen der photorealistischen Bildgenerierung zählen die visuellen Effekte (*Visual Effects*) in der Filmindustrie, Computerspiele und die Produktwerbung. Diese Anwendungen zielen hauptsächlich darauf ab, Effekte mit Hilfe von synthetisch erzeugten Bildern darzustellen, die durch konventionelle Techniken nicht realisierbar sind oder deren Umsetzung ohne Computeranimation zu teuer ist. Speziellere Anwendungen der Computergrafik sind Flug- und Fahr simulatoren sowie Beleuchtungsplanung und Architekturvisualisierung. Die Zielsetzung von Anwendungen aus diesen Kategorien ist die Unterstützung von Lern-, Planungs- und Entscheidungsprozessen. Hierbei steht die Simulation von realen Vorgängen in oftmals (noch) nicht existierenden Umgebungen im Vordergrund, wobei ein visueller Eindruck von maximaler Realitätsnähe vermittelt werden soll. Dieser Bedarf an Realitätsnähe hat jedoch seinen Preis, wie im weiteren Verlauf dieser Arbeit noch gezeigt wird.

Mit Hilfe von Computern erzeugte synthetische Bilder und Animationen gewinnen im Architektur- und Bauwesen zunehmend an Bedeutung. Während der Planung und Durchführung eines Bauvorhabens können 3D-Visualisierungen sowohl von Architekten und Innenarchitekten als auch Beleuchtungsplanern dazu verwendet werden, ihre Ideen sichtbar zu machen. Baupläne und abstrakte lichttechnische Berechnungen sind im allgemeinen nicht dazu geeignet, einen plastischen Eindruck von einem Bauvorhaben zu vermitteln. Im Rahmen von Genehmigungs- und Präsentationsverfahren können Computervisualisierungen eingesetzt wer-

den, um ein Bauvorhaben darzustellen. Hierbei wird die Kommunikation zwischen Architekten, Bauträgern, Behörden und Anliegern durch die allgemeine Verständlichkeit von realistisch wirkenden Bildern vereinfacht. Ein virtuelles 3D-Modell ermöglicht schon während der Entwurfs- und Wettbewerbsphase einen guten Einblick in die ersten Konzepte, wodurch Planungsfehler einfach zu entdecken sind. Dieses wurde als visuelle Revolution im Bereich des Beleuchtungsdesigns bezeichnet [Mis94]. Sowohl Lichtdesigner (bzw. Architekten) als auch die Kunden sind sich mittlerweile der Möglichkeiten bewußt, die durch die Erzeugung und die Analyse realistischer computergenerierter Bilder von Architekturmodellen entstehen. Die Nutzung dieser Bilder zur Unterstützung der Beleuchtungsplanung setzt die korrekte Wiedergabe der Beleuchtungsverhältnisse innerhalb real existierender oder virtueller Umgebungen voraus.

Während der Planung von Innenräumen sind die Auswirkungen von verschiedenen Beleuchtungssituationen nur schwer vorstellbar. Dies betrifft insbesondere die Quantität und die Qualität der Verteilung von diffus reflektiertem Licht. Zur Beurteilung von Kunstlicht-, Tageslicht- und Mischlicht-Situationen benötigt der Planer aber gewisse Anhaltspunkte. Ideal ist eine Simulation der Lichtverteilung, welche die Beurteilung von Raumausleuchtungen, Lichtstimmungen und die Wirkung von Materialien und deren Farben ermöglicht. Für den Lichtdesigner sind neben der qualitativen Wirkung des Lichtes auch photometrische Kennzahlen zur quantitativen Spezifikation der relevanten Lichtparameter von Bedeutung.

Das klassische Verfahren zur Lichtsimulation ist die Messung der Lichtverteilung an einem 1:20-Modell. Die Schwierigkeit beim Modellverfahren liegt in der fehlenden Innenraumausstattung und den verwendeten Materialien. Auch können bei einem 1:20-Modell keine wirklichen Lichtkörper verwendet werden. Bei der Büroplanung werdend teilweise schon Muster-Situationen im Maßstab 1:1 eingesetzt, um Kunstlichtplanungen zu beurteilen. Was ist aber zu tun, wenn das Planungsobjekt die Ausmaße eines Kinos, Theaters, Festsaals oder eines Atriums hat? Ein weiterer Nachteil der Modellverfahren ist, daß sich Tageslichtsituationen, die sich an einem bestimmten Sonnenstand orientieren, nur mit hohem Aufwand nachstellen lassen.

All diese Einschränkungen kennt die digitale Beleuchtungssimulation nicht. Die Visualisierung korrekter Beleuchtungsverhältnisse bedingt jedoch den Einsatz rechenintensiver Verfahren, die auf einer Simulation des physikalischen Phänomens Licht basieren. Innerhalb dieser Arbeit liegt das Hauptaugenmerk auf der effizienten parallelen Simulation von Beleuchtungssituationen, die innerhalb von komplexen Architekturmodellen vorherrschen. Die vorgestellten Verfahren sind jedoch nicht auf diesen Bereich beschränkt, sondern lassen sich direkt in anderen Anwendungsbereichen der photorealistischen Bildgenerierung einsetzen.

1.1 Prozeß der Bildgenerierung

Der Begriff *Computergrafik* bezeichnet ein breites Feld von verschiedenen Forschungsgebieten und Techniken. Aus diesem Grund ist es wichtig, eine klare Abgrenzung der in dieser Arbeit behandelten Thematik vorzunehmen. Gegenstand dieser Arbeit ist die schnelle photorealistische Generierung von Bildern komplexer dreidimensionaler Umgebungen durch effiziente Nutzung paralleler Rechnersysteme.

Der vollständige Prozeß der Erzeugung synthetischer Bilder von virtuellen dreidimensionalen Gebilden oder Umgebungen mit Hilfe des Computers wird im allgemeinen in drei separate Stufen unterteilt:

- **Modellierung der Szenen:** Der erste Schritt in jedem Bildgenerierungsprozeß ist die Modellierung. Die darzustellende dreidimensionale Szene wird durch eine Menge von Objekten und deren Materialien spezifiziert, die in der virtuellen Umgebung enthalten sind. Diese Beschreibung enthält die mathematische und geometrische Beschreibung aller Objekte, ihre Positionen und gegenseitige Ausrichtungen, die optischen Eigenschaften von Materialien, welche die Interaktionen von Licht mit dem Material einer Oberfläche von Objekten beschreiben sowie die Positionierung von Lichtquellen innerhalb der Szene. Diese Szenenbeschreibungen können in beliebiger Form eingegeben werden. Üblicherweise geschieht dieses mit Hilfe eines Computerprogrammes, das als *Modeller* bezeichnet wird. Das Ergebnis des Modellierungsprozesses sind Dateien mit den zur vollständigen Beschreibung einer Szene benötigten Daten. Zum Zweck der Bildgenerierung werden die Objektoberflächen einer Szene oftmals in polygonale Elemente unterteilt. Dieses bietet sich vor allen Dingen dann an, wenn moderne Grafikhardware verwendet werden soll, welche die schnelle Darstellung von triangulierten Objektoberflächen unterstützt. Die Granularität der Diskretisierung von Oberflächen hat erhebliche Auswirkungen auf die Qualität sowie die Geschwindigkeit der Bildgenerierung.
- **Bildgenerierung (Rendering):** Es existiert eine Vielzahl von Verfahren zur Bildgenerierung, zwischen denen je nach Anwendungsfeld gewählt werden kann. Bei den ersten verfügbaren Verfahren handelte es sich um projektive Methoden, bei denen jedem Polygon der Umgebungsbeschreibung ein einheitlicher konstanter Farbwert zugewiesen wurde (*Flat-Shading*). Diese Verfahren wurden später dahingehend erweitert, daß für die Polygone die Auswirkungen der direkten Beleuchtung durch Lichtquellen in der Szene auf der Basis von empirischen Modellen bestimmt wurden. Hierbei wurde der Aspekt der indirekten Beleuchtung vernachlässigt oder durch Addition eines konstanten *ambienten* Terms sehr grob approximiert. Fortschrittliche Verfahren zur Bildgenerierung basieren auf einer physikalischen Simulation der Beleuchtung von Objekten einer Umgebung. Das der Simulation zugrundeliegende Problem wird als *globales Beleuchtungsproblem* bezeichnet.

Abhängig von der gewünschten Qualität der Bildgenerierung wird zwischen einer großen Anzahl von verschiedenen Techniken und Algorithmen zur Beleuchtungsapproximation unterschieden. Diese reichen von einfachen lokalen Beleuchtungsverfahren wie Phong- und Gouraud-Shading [Gou71][BT75] bis hin zu komplexen Lösungsmethoden wie Monte-Carlo-Raytracing [Arv86][Kaj86] und Radiosity [GTGB84] [NN85].

- **Farbanpassung des Bildes für ein Ausgabemedium:** Sobald ein Bild für den gegebenen Standpunkt eines Betrachters (Kameraposition) in der Szene berechnet wurde, muß es auf einem Ausgabegerät wiedergegeben werden. Dieses ist üblicherweise ein Computerbildschirm, aber es kann ebenso ein Farbdrucker sein. Hierbei spielen die korrekte Farb- und Intensitätswiedergabe eine große Rolle, die stark von dem gewünschten Ausgabemedium abhängen.

Die in dieser Arbeit vorgestellten parallelen Methoden konzentrieren sich primär auf die Phase der Bildgenerierung.

1.2 Beleuchtungsmodelle in der Computergrafik

Verfahren zur Erzeugung von photorealistischen Bildern, welche die physikalisch korrekte Beleuchtungssituation innerhalb von dreidimensionalen Szenen wiedergeben, basieren auf der Simulation von grundlegenden physikalischen Phänomenen wie Lichtemission, Lichtreflexion und die dadurch bedingte Lichtenergieverteilung. Ziel ist es, visuelle Effekte wie indirekte Beleuchtung, Farbübertragung (*Color-Bleeding*), Schatten, Halbschatten und Reflexionen bei der Bildgenerierung zu berücksichtigen, um den Realitätsgrad der erzeugten Bilder deutlich zu steigern [MRC86][Wan92]. Zu diesem Zweck ist es notwendig, die Ausbreitung des Lichtes durch sogenannte Beleuchtungsmodelle zu beschreiben.

Bei der photorealistischen Bildgenerierung wird ein Bild im allgemeinen durch eine zweidimensionale Matrix von Farbwerten beschrieben (Raster-Bild). Die Farbe jedes Punktes (*Pixel*) des Bildes wird anhand der aktuellen Beleuchtung der an dem jeweiligen Bildpunkt sichtbaren Oberfläche der Szene berechnet (siehe Abb. 1.1).

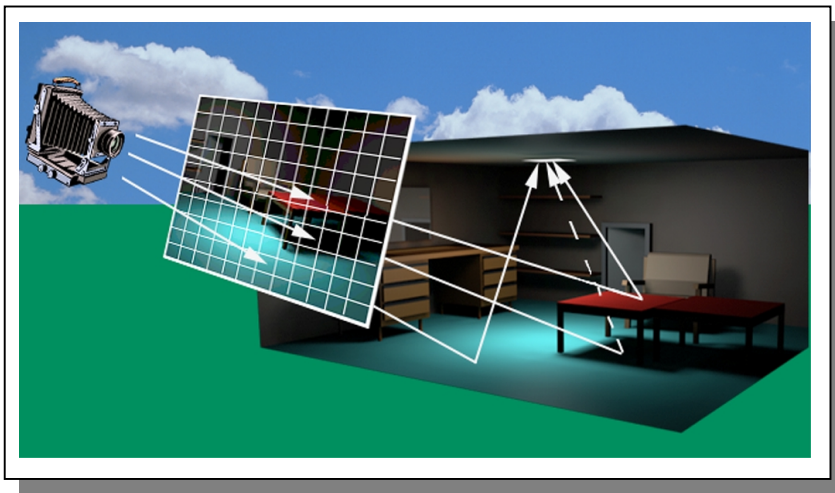


Abb. 1.1: Ermittlung von Pixel-Intensitäten (bzw. Farbwerten).

Beleuchtungsmodelle definieren eine Menge von Lichtquellen innerhalb der Szene sowie Beleuchtungsregeln für die Berechnung von Farbwerten für sichtbare Oberflächenpunkte. Hierbei wird zwischen *lokalen* und *globalen Beleuchtungsmodellen* unterschieden.

Lokale Beleuchtungsmodelle sind empirische Modelle. Sie beschreiben die Verteilung des reflektierten Lichtes als Funktion des auf die Oberfläche einfallenden Lichtes, das von Lichtquellen in die Umgebung abgegeben wird. Es wird von einem sehr idealisierten Reflexionsverhalten und von punktförmigen Lichtquellen ausgegangen. Auf aufwendige Berechnungen von Schatten und Spiegelungen wird dabei zumeist verzichtet. Der Ausdruck *lokales Beleuchtungsmodell* bedeutet in diesem Zusammenhang, daß die Beleuchtung der Oberflächen lediglich von deren lokalen Materialeigenschaften sowie den Eigenschaften der Lichtquellen abhängen, da nur die direkte Wechselwirkung zwischen Oberflächenpunkten auf Objekten und Lichtquellen modelliert wird. Dieses bedeutet jedoch, daß lokale Beleuchtungsmodelle keine korrekte Simulation der realen Vorgänge bei der Lichtausbreitung ermöglichen. Denn in der Realität wird eine Oberfläche sowohl direkt durch Lichtquellen als auch indirekt durch Reflexion des Lichtes von anderen Objekten der Umgebung beleuchtet. Dem Aspekt der indirekten Beleuchtung durch andere Objekte der Szene wird in lokalen Beleuchtungsmodellen

durch Addition eines konstanten *ambienten Terms* versucht, Rechnung zu tragen. Dieses ist jedoch eine sehr ungenaue Approximation der realen Beleuchtungssituation, da die indirekte Beleuchtung einen wesentlichen Anteil an der Beleuchtung einer Umgebung ausmacht. Trotz der erwähnten Einschränkungen werden lokale Beleuchtungsmodelle sehr häufig in Echtzeitgrafik-Systemen mit leistungsfähiger Spezial-Hardware eingesetzt.

Ein Modell zur Beschreibung der Beleuchtung von Objekten durch Mehrfachreflexionen des Lichtes innerhalb einer Umgebung wird als *globales Beleuchtungsmodell* bezeichnet. Bei globalen Beleuchtungsverfahren wird nicht nur der Lichtaustausch zwischen den Objekten und den Lichtquellen sondern auch zwischen den Objekten untereinander berücksichtigt. Dabei wird auch die Verdeckung in die Berechnung mit einbezogen, was zur Bildung von Schatten führt. Im Gegensatz zu lokalen Beleuchtungsmodellen werden durch globale Beleuchtungsmodelle Effekte wie indirekte Beleuchtung, weiche Schatten, durch Flächenlichtquellen verursachte Halbschatten sowie Farbübertragungen (*Color-Bleeding*) berücksichtigt. Die möglichst exakte Approximation der globalen Beleuchtung hat sich zu einem Forschungsschwerpunkt in der Computergrafik entwickelt. Mittlerweile stehen verschiedene Verfahren zur physikalischen Simulation von Lichtenergieverteilungen zur Verfügung. Die meisten dieser Algorithmen basieren auf zwei grundsätzlich verschiedenen Ansätzen, dem *Raytracing* und *Radiosity*.

Die ursprüngliche Methode des Raytracing wurde in ihren Grundzügen in den sechziger Jahren formuliert [Whi80] und wird seitdem in vielen kommerziellen Bildgenerierungssystemen erfolgreich eingesetzt. Das Grundprinzip basiert auf den Gesetzen der Strahlenoptik und geht von einer strahlenförmigen Natur des Lichtes aus. Die Photonenbewegung des Lichtes wird simuliert, indem virtuelle Strahlen von dem Standpunkt eines Betrachter (Augenpunkt) durch die Bildebene in der Szene gesendet werden. Für jeden dieser Sehstrahlen muß derjenige Schnittpunkt mit einem Objekt der Szene bestimmt werden, der dem Augenpunkt am nächsten ist. Ist das Objekt spiegelnd, wird ein weiterer Strahl in die Spiegelrichtung geschickt. Ist das Objekt transparent, wird mit dem Brechungsgesetz die Richtung eines transmittierten (gebrochenen) Strahls bestimmt. Für diese beiden Strahlen wird das Verfahren rekursiv durchgeführt. Wenn das geschnittene Objekt nicht vollständig spiegelnd oder transparent ist, wird am Strahlenschnittpunkt ein lokales Beleuchtungsmodell ausgewertet. Bevor eine Lichtquelle einen Beitrag zur Beleuchtung eines Oberflächenpunktes liefert, wird mit Schattenstrahlen überprüft, ob die Lichtquellen vom Objektschnittpunkt aus sichtbar sind. Bei dieser Vorgehensweise handelt es sich um eine Umkehrung des Fortpflanzungsprozesses von Photonen in einer Umgebung nach deren Emission von Punktlichtquellen. Aus diesem Grund wird das oben geschilderte Raytracing-Verfahren auch oftmals als *Backward Eye Ray Tracing* bezeichnet im Gegensatz zum *Forward Light Ray Tracing*.

Mit dem Raytracing-Verfahren lassen sich sehr eindrucksvolle Bilder generieren, besonders bei Szenen mit vorwiegend spiegelnden oder transparenten Objekten. Derartige Reflexionen haben in unserer Umwelt zwar nur einen geringen Anteil, in speziellen Situationen ist ihr Einfluß jedoch ausschlaggebend. Weitere Vorteile sind die Einfachheit des Verfahrens und die geringen Anforderungen an die Objektbeschreibungen. Die einzige Anforderung ist die Existenz eines Verfahrens zur Berechnung eines Schnittpunktes mit einem Strahl.

Ein Nachteil ist, daß das Raytracing-Verfahren konzeptionell abhängig von dem Blickpunkt eines Betrachters ist, da Beleuchtungsinformationen nur für die Punkte innerhalb der Szene bestimmt werden, die von einer vorgegebenen Position und in einer bestimmten Richtung sichtbar sind. Somit muß bei einer Änderung des Betrachterstandpunktes die Berechnung vollständig wiederholt werden. Schwächen hat das Verfahren zudem bei diffus reflektieren-

den Oberflächen. Beim herkömmlichen Raytracing wird die indirekte diffuse Beleuchtung einer Oberfläche der Szene durch andere Objekte mit Hilfe von einem konstanten ambienten Term nur sehr grob approximiert.

Monte-Carlo-Raytracing ermöglicht es, diffuse Beleuchtungseffekte zu simulieren, indem Strahlenbündel, die den Halbraum oberhalb eines Oberflächenpunktes abtasten, rekursiv weiterverfolgt werden [Arv86][Kaj86][WRC88][AK90]. Diese Methoden sind jedoch sehr zeittensiv, und aufgrund der Verfolgung von zufällig verteilten Abtast-Strahlen (*Sample-Strahlen*) weisen die erzeugten Bilder oftmals Rauscheffekte (*Spekles*) auf.

Während im Produkt-Design mit Raytracing schon seit geraumer Zeit photorealistische Bilder generieren werden konnten, wird in der Architekturvisualisierung dieses Ziel erst mit dem Radiosity-Verfahren erreicht. Raytracing berechnet zwar Reflexionen und Refraktionen physikalisch korrekt, aber berücksichtigt keine diffusen Interobjekt-Reflexionen. Gerade diese Art der Reflexion ist aber bei der Architekturvisualisierung von entscheidender Bedeutung, da hier überwiegend diffus reflektierende Flächen vorliegen (Wände, Böden, Decken, Möbelstoffe etc.).

Die Grundidee des Radiosity-Verfahrens ist es, für eine Umgebungsbeschreibung (Szene) eine physikalisch korrekte Lichtstrahlungsbilanz zu berechnen, der alle Daten für eine realistische, dreidimensionale Bildwiedergabe entnommen werden können. Das Radiosity-Verfahren wurde 1984 erstmals in der Computergrafik eingesetzt [GTGB84][NN85] und stützt sich auf Grundlagen der Thermodynamik, Radiometrie und der Photometrie [Hot54][ED59]. Diese Technik wurde schon in den vierziger Jahren in der Wärmetechnik verwendet, um Temperaturverteilungen in geschlossenen Umgebungen berechnen zu können. Licht wird hierbei als Träger von Strahlungsenergie betrachtet. Diese Betrachtungsweise ermöglicht eine Simulation der diffusen Lichtverteilung in geschlossenen Umgebungen. Mit dem Radiosity-Verfahren war es erstmals möglich, die indirekte diffuse Beleuchtung von Objekten in einer Szene effizient zu berechnen. Das Radiosity-Verfahren beruht auf der Diskretisierung von Oberflächen der Objekte einer Szene, indem sie in endlich große Elemente (*Patches*) unterteilt werden. Für jedes dieser Patches wird bei der klassischen Radiosity-Methode die Beleuchtung als konstant angenommen. Es wird davon ausgegangen, daß Flächen das Licht rein diffus reflektieren. Mit Hilfe eines linearen Gleichungssystems wird ein Energiegleichgewicht in einer gegebenen Szene beschrieben. Die Lösung dieses Gleichungssystems ordnet jedem diskreten Oberflächenelement einen Radiosity-Wert (Strahlungsenergie, die unabhängig von der Richtung pro Flächeneinheit in die Umgebung abgegeben wird) zu. Die berechneten Radiosity-Werte werden als Intensitätswerte interpretiert. Die abschließende Bildgenerierung kann mit Hilfe einfacher Schattierungsverfahren wie zum Beispiel der Gouraud-Schattierung [Gou71] auf der Basis der berechneten Intensitätswerten erfolgen. Da mit Hilfe der Radiosity-Methode, im Gegensatz zur Raytracing-Methode, die Verteilung der Lichtenergie unabhängig von einem Betrachterstandpunkt in der Szene berechnet wird, kann die Lösung zur Durchführung von virtuellen Begehungen (*Walkthrough-Animationen*) mittels spezieller Grafikhardware genutzt werden.

Die Bedeutung der Berücksichtigung von indirekter Beleuchtung wird in Abb. 1.2 demonstriert. In dem linken Bild wird die Szenerie nur durch Spot-Lichtquellen beleuchtet, welche die Lichtenergie von der Decke in Richtung Fußboden strahlen. Bei Vernachlässigung jeglicher Reflexionen des Lichtes innerhalb der Szene wird lediglich der untere Raumbereich ausgeleuchtet, wodurch das Bild unnatürlich dunkel erscheint. Das rechte Bild wurde in der identischen Szene vom gleichen Betrachterstandpunkt aus erzeugt. In diesem Fall wurde jedoch innerhalb der Szene die diffuse Reflexion der Lichtenergie mit Hilfe der Radiosity-

Methode simuliert. Es ist deutlich zu erkennen, daß die Berücksichtigung der indirekten Beleuchtung erheblich zur Steigerung des Realitätsgrades des Bildes beiträgt.

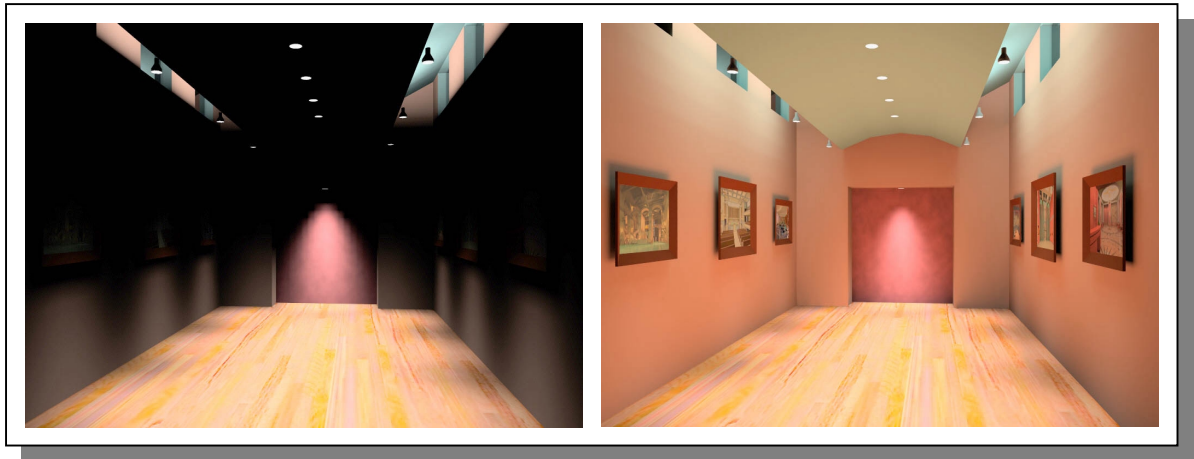


Abb. 1.2: Auswirkungen der indirekten Beleuchtung (links: nur direkte Beleuchtung durch Spot-Lichtquellen; rechts: direkte und indirekte Beleuchtung durch diffuse Reflexion).

Lange Zeit war umstritten, ob das Radiosity-Verfahren für komplexere Szenen überhaupt einsetzbar ist. Selbst einfache Szenen bestehen zumeist aus mindestens 10.000 Flächenelementen, was zu einem linearen Gleichungssystem mit über 100 Millionen Matrix-Elementen führen würde. Eine Bestimmung der kompletten Gleichungssysteme ist somit in den meisten Fällen aus Rechenzeit- und Speichergründen nicht praktikabel. Ein wichtiger Beitrag war daher das 1988 eingeführte iterative *Progressive-Refinement-Radiosity* [CCWG88]. Bei diesem Verfahren besteht nicht mehr die Notwendigkeit das komplette Gleichungssystem explizit zu erzeugen. Das Progressive-Refinement-Verfahren wird auch als *Shooting-Verfahren* bezeichnet, da in jedem Iterationsschritt die Energie einer ausgewählten Fläche an die übrigen Flächen der Umgebung verteilt (verschossen) wird. Wenn bei der Durchführung der Shooting-Iterationen mit der Verteilung der größten Energiemengen begonnen wird (hellste Flächen), kann schon mit wenigen Iterationen eine erste gute Annäherung der endgültigen Radiosity-Lösung bestimmt werden.

Die klassische Radiosity-Methode berücksichtigt ebenso wie das klassische Raytracing-Verfahren nur einen Teilaspekt der Lichtausbreitung in realen Umgebungen. Während bei Raytracing-Verfahren vorwiegend Beleuchtungseffekte simuliert werden, die durch ideal spiegelnde Reflexionen entstehen, konzentriert sich die Radiosity-Methode auf die ideal diffuse Lichtreflexion. Da aber der größte Anteil der in der Realität vorkommenden Reflexionen diffuser Natur ist, wird mit Radiosity-Algorithmen eine sehr realistische und natürliche Darstellung erzielt.

Es ist zu beachten, daß die Vorteile der verschiedenen globalen Beleuchtungsverfahren zu Mehrphasen-Methoden [CRMT91] kombiniert werden können. In der einfachsten Form kann eine Zwei-Phasen-Methode mit Hilfe von Raytracing und Radiosity realisiert werden [WCG87][SP89]. Hierbei werden durch spiegelnde und diffuse Reflexion verursachte Beleuchtungseffekte in separaten, nacheinander ausgeführten Phasen simuliert und kombiniert. Diese Vorgehensweise ermöglicht die Erzeugung von Bildern mit einem sehr hohen Realitätsgrad.

Aufgrund des großen Rechenzeit- und Speicherbedarfs, der in Abhängigkeit der Modellkomplexität wächst, ist die Nutzung dieser Mehrphasen-Methoden in den meisten Fällen bisher nicht praktikabel. Aus Sicht der Forschung stellt es eine Herausforderung dar, basierend auf globalen Beleuchtungsverfahren, eine Online-Begehung (*Walkthrough*) innerhalb eines korrekt beleuchteten Szenarios zu ermöglichen, wobei Interaktionen mit Objekten der Umgebung möglich sind und die Auswirkungen von Veränderungen auf die Beleuchtung direkt visuell kontrolliert werden können. Ein vielversprechender Ansatz zur Lösung dieses Problems ist die effiziente Nutzung der Vorteile, welche die Parallelverarbeitung bietet.

1.3 Parallelverarbeitung und Computergrafik

Die Parallelverarbeitung und die Computergrafik sind zwei Forschungsgebiete innerhalb der Informatik, die sich gegenseitig in starkem Maße ergänzen. Viele Algorithmen der Computergrafik sind durch einen enormen Bedarf an Rechenzeit und Speicherplatz gekennzeichnet. Dieser Bedarf läßt sich in idealer Art und Weise durch parallele Rechnersysteme decken. Insbesondere daten- und berechnungskomplexe Bildgenerierungsverfahren, die auf einer globalen Beleuchtungssimulation basieren, profitieren von den sehr guten Skalierbarkeitseigenschaften von Parallelrechnersystemen mit verteiltem Speicher. Zur Erzeugung einzelner qualitativ hochwertiger Bilder werden mit herkömmlichen sequentiellen Verfahren Stunden bis Tage benötigt. Dieses führt bei der Herstellung von längeren Animationssequenzen mit mindestens 25 Bildern pro Sekunde zu Produktionszeiten von mehreren Wochen bis Monaten. Durch eine Nutzung der Vorteile des Konzeptes der Parallelverarbeitung ist eine drastische Verkürzung der Produktionszeiten möglich.

Durchgeführte Forschungsarbeiten in dem Bereich der parallelen Computergrafik konzentrieren sich im wesentlichen auf zwei Fragestellungen [BL90]:

1. Wie kann ein sequentieller Algorithmus derart modifiziert werden, daß er in einer parallelen Umgebung (dedizierter Parallelrechner oder Verbund von Workstations) effizient ausgeführt werden kann?
2. Wie kann eine parallele Umgebung (Spezialhardware) konstruiert werden, um spezielle Grafikfunktionen zu unterstützen?

Im Rahmen dieser Arbeit wird ausschließlich die erste Fragestellung betrachtet.

Die Vorteile, welche die Parallelverarbeitung im Bereich der Bild- und Animationserzeugung bietet, wurde auch von den Entwicklern kommerzieller Modellierungs- und Renderingsoftware erkannt. Teilweise unterstützen diese Softwareprodukte eine verteilte Erzeugung von Animationen innerhalb eines lokalen Netzwerkes, indem die einzelnen Bilder einer Animation auf verschiedenen Rechnern des Verbundes parallel generiert und anschließend zentral zu einer Sequenz zusammengefügt werden. Dem Problem der Szenenkomplexität wird bei dieser Vorgehensweise nicht Rechnung getragen, da die vollständigen Geometrie- und Materialdaten auf jedem Rechner gespeichert werden.

Die breite Masse der verfügbaren kommerziellen Produkte verwendet bei der Bildgenerierung noch keine fortschrittlichen Verfahren wie Radiosity, Monte-Carlo-Raytracing oder Mehrphasen-Methoden zur physikalisch basierten Approximation der globalen Beleuchtung. Dieses ist darauf zurückzuführen, daß bisher nur wenige effiziente Parallelisierungsstrategien existieren, welche sowohl die Berechnungszeit deutlich reduzieren, als auch eine gute Ausnutzung des verfügbaren verteilten Speichers durch datenparallele Berechnungen gewährleisten. Daten-

parallele Ansätze zur globalen Beleuchtungssimulation sind derzeit Gegenstand intensiver Forschungsarbeiten und werden von Seiten kommerzieller Entwickler noch mit Skepsis betrachtet. Somit ist es Aufgabe der Forschung, im Bereich der parallelen Computergrafik zuverlässige Verfahren zur Verfügung zu stellen, die den Anforderungen professioneller Anwender bezüglich Qualität, Geschwindigkeit und Effizienz genügen.

Ein weiterer Grund für die zögerliche Haltung bei der Verwendung komplexer paralleler Algorithmen in kommerzieller Modellierungs- und Visualisierungssoftware ist, daß sich die Entwicklung und Implementierung von zuverlässigen Algorithmen für parallele Systeme erheblich schwieriger gestaltet als die Entwicklung herkömmlicher sequentieller Verfahren. Die Ursachen hierfür sind in den zahlreichen Aspekten zu suchen, die zusätzlich berücksichtigt werden müssen, um effiziente parallele Algorithmen zu entwickeln. Hierbei handelt es sich um fundamentale Problemstellungen innerhalb der Parallelverarbeitung wie Kommunikation, Synchronisation, Lastverteilung, Mapping und die verteilte Terminierungserkennung. Die konkrete Umsetzung dieser Aspekte in einem parallelen Programm hat einen entscheidenden Einfluß auf dessen Korrektheit sowie die Performanz des Verfahrens.

Ein wesentliches Ziel bei der Entwicklung von parallelen Algorithmen ist die Realisierung skalierbarer Verfahren. Der Begriff der *Skalierbarkeit* eines parallelen Algorithmus bezieht sich auf die Eigenschaft des Verfahrens, mit einer steigenden Anzahl von Prozessoren über eine größere Problemlösungskapazität zu verfügen. Die Skalierbarkeit wird üblicherweise über Maße wie Berechnungszeit, Speed-Up, Effizienz sowie die Komplexität der potentiell lösbaren Probleme bewertet. Es ist zu beachten, daß die Notwendigkeit zur Interprozeß-Kommunikation, Synchronisation und verteilte Terminierungserkennung zu einem Mehraufwand (*Overhead*) bei den parallelen Berechnungen führt, wodurch die Skalierbarkeit eines parallelen Verfahrens erheblich beeinträchtigt werden kann. Weiterhin kann es aufgrund von Lastungleichgewichten zwischen den Prozessoren eines Parallelrechners zu Wartezeiten (*Idle-Zeiten*) der Prozessoren und somit zu einer Verringerung der Effizienz von parallelen Berechnungen kommen.

Ein weiterer wesentlicher Aspekt bei der Entwicklung von parallelen Verfahren ist die Architektur des verwendeten Parallelrechners. Eine große Anzahl von Computerarchitekturen wurden in der Vergangenheit vorgeschlagen und zum Teil realisiert. 1972 wurde von Flynn eine Klassifizierung vorgeschlagen, die existierende Rechnerarchitekturen in vier Kategorien einordnet [Fly72]. Die Einordnung in eine der Kategorien erfolgt gemäß der Art und Weise, wie die Verarbeitung von Instruktionsfolgen und der Zugriff auf Daten durch die entsprechende Architektur realisiert wird. Hierbei wird zwischen den folgenden grundlegenden Architekturklassen unterschieden:

SISD (*Single Instruction Single Data*)

Konventionelle von Neumann Architekturen fallen in diese Kategorie. Die Instruktionen eines Programmes werden sequentiell auf einem einzelnen Datenstrom ausgeführt. Moderne Variationen dieses Einprozessorsystems verwenden Pipelining-Techniken, um die Performanz des Systems zu steigern.

SIMD (*Single Instruction Multiple Data*)

Architekturen dieser Kategorie ermöglichen die gleichzeitige Anwendung einer Instruktion auf eine Menge von Datenelementen. Hierbei wird eine einzelne Kontrolleinheit und mehrere Prozessoren verwendet, die über ein Verbindungsnetzwerk verbunden sind. Die Kontrolleinheit dekodiert Instruktionen und übermittelt sie an die Prozessoren, welche sie parallel auf den lokalen Daten ausführen.

MISD (*Multiple Instruction Single Data*)

Architekturen dieser Klasse können verschiedene Instruktionen zur gleichen Zeit auf einem einzelnen Datenelement ausführen.

MIMD (*Multiple Instruction Multiple Data*)

Die Prozessoren eines MIMD-Parallelrechners können verschiedene Instruktionssequenzen autonom auf lokalen Daten ausführen. Rechnersysteme dieser Kategorie ermöglichen asynchrone Berechnungen. Es besteht somit nicht die Notwendigkeit einer synchronen Bearbeitung von Instruktionen, wie es bei SIMD-Rechnern der Fall ist. Durch Interprozessorkommunikation ist eine gemeinsame Bearbeitung eines Problems durch mehrere Prozessoren möglich. Es werden zwei verschiedene Ansätze zur Interprozessorkommunikation verfolgt.

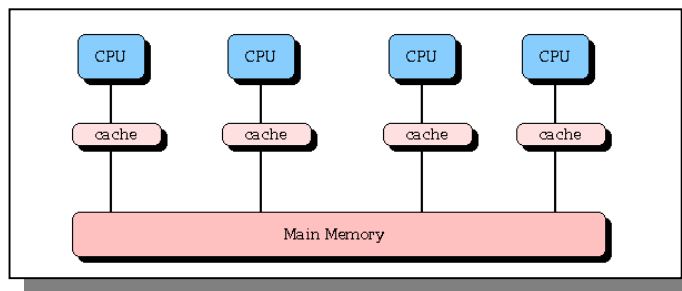


Abb. 1.3: Aufbau eines SMP-Systems (*Symmetric-Multi-Processing*).

In *Shared-Memory-Systemen* verfügen die Prozessoren über einen gemeinsamen Adressraum (siehe Abb. 1.3). Die Kommunikation erfolgt über Schreib- und Leseoperationen auf globalen Variablen. Hierbei werden Kontrollmechanismen verwendet, um einen Zugriff auf gemeinsame Ressourcen (Daten) zur gleichen Zeit zu verhindern und somit die Konsistenz von Daten zu gewährleisten.

In Systemen mit verteiltem Speicher (*Distributed Memory Systeme*) verfügen Prozessoren nur über einen lokalen Adressraum, der exklusiv genutzt wird. Eine Kommunikation zwischen den Prozessoren erfolgt durch den Austausch von Nachrichten (*Message-Passing*) auf Basis eines Verbindungsnetzwerkes zwischen den Prozessoren (siehe Abb. 1.4).

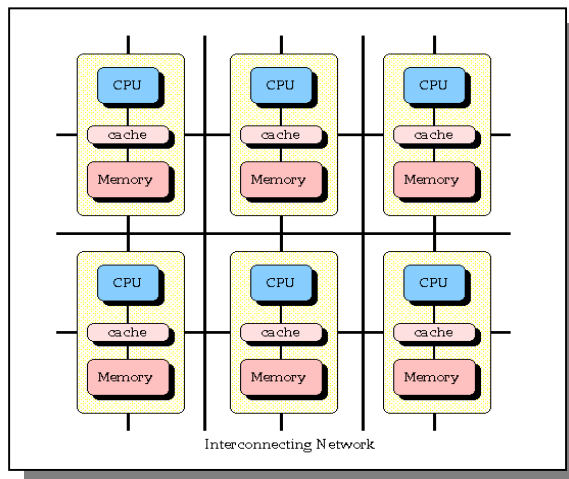


Abb. 1.4: Aufbau eines MPP-Systems.

Die Nutzung von Parallelität im Bereich der Computergrafik ist zur Normalität wenn nicht gar zum Muß geworden. Der Schwerpunkt liegt hierbei auf SIMD- und MIMD-Architekturen, wobei der Trend von ursprünglich verwendeten speziellen Architekturen hin zu universellen Parallelrechnern (*General Purpose Parallelrechnern*) und verteilten Systemen (heterogene Workstations und PCs über LAN gekoppelt) sowie kleinen SMP-Systemen geht.

In der Vergangenheit wurden für verschiedenste Problemstellungen aus dem Bereich der Computergrafik parallele Verfahren entwickelt. Beispiele hierfür sind

- Hidden-Surface-Removal-Algorithmen [SSS74]/ Paralleles Rendering,
- globale Beleuchtungsverfahren wie Raytracing und Radiosity,
- Visualisierung von Volumendaten,
- Computeranimation (verteilte Generierung von Einzelbildern),
- Parallelisierung von Grafikstandards (PHIGS+, GKS, RenderMan) und
- die Rendering-Pipeline (Transformation, Rotation, Skalierung, Clipping, HSR, Shading).

Der größte Teil der existierenden Parallelisierungen wurde für spezielle Prozessortypen und Parallelrechnerarchitekturen entwickelt. Ein Ziel der Forschung im Bereich der parallelen Computergrafik muß aus Kosten- und Verfügbarkeitsgründen die Entwicklung effizienter paralleler Methoden für General-Purpose-Architekturen sein. Hierbei sollte durch die Verwendung von Standards zum Austausch von Nachrichten zwischen Prozessoren in parallelen Umgebungen wie PVM und MPI eine Unabhängigkeit von der Hardwareplattform erreicht werden, um die Unterstützung heterogener verteilter Umgebungen zu gewährleisten.

1.4 Erzielte Resultate

Methoden zur Simulation der globalen Beleuchtung in virtuellen Umgebungen ermöglichen es, bei der Erzeugung von synthetischen Bildern eine große Anzahl der in realen Umgebungen auftretenden Beleuchtungseffekte zu berücksichtigen. Aufgrund der großen Daten- und Berechnungskomplexität dieser Methoden ist die Entwicklung von effizienten (daten-) parallelen Verfahren eine wichtige Forschungsaufgabe.

In dieser Arbeit werden zwei neue effiziente datenparallele Radiosity-Verfahren zur möglichst exakten Simulation der globalen Beleuchtung in hochgradig komplexen Modellen auf Basis eines Progressive-Refinement-Ansatzes, sowie deren Integration in ein interaktives Bildgenerierungssystem beschrieben. Beide parallele Methoden wurden dahingehend erweitert, daß interaktive Manipulationen der Geometrie und der Materialeigenschaften von Objekten innerhalb einer Szene während der Beleuchtungssimulation berücksichtigt werden, indem eine Korrektur der Radiosity-Lösung inkrementell berechnet wird. Aufgrund der parallelen Berechnung von korrigierten Lösungen werden sehr kurze Anpassungszeiten erzielt. Dieses ermöglicht es, die Auswirkungen einer Szenenmodifikation auf die globale Beleuchtung visuell zu kontrollieren, ohne das lange Wartezeiten entstehen, wie es bei einer kompletten Neuberechnung einer Lösung der Fall ist [SRRP98]. Weiterhin wurde eine neue Strategie zur dynamischen Lastverteilung entwickelt, welche die Effizienz von datenparallelen Radiosity-Berechnungen deutlich steigert [SR99].

Das Ziel von Verfahren zur Simulation globaler Beleuchtungseffekte ist, alle möglichen Pfade zu berücksichtigen, die ein Photon nach dessen Emission durch eine Lichtquelle innerhalb einer Umgebung zurücklegen kann, bevor es von einer Oberfläche absorbiert wird. Die im

Rahmen dieser Arbeit vorgestellten parallelen Mehrphasen-Verfahren ermöglichen es, bei der globalen Beleuchtungssimulation sehr komplexe Lichttransportpfade zu berücksichtigen. Dieses trägt zu einer deutlichen Steigerung des Realitätsgrades der generierten Bilder bei. Es können Beleuchtungseffekte wiedergegeben werden, die durch diffuse Reflexionen zwischen den Objekten der Szene und spiegelnde Reflexion sowie transparente Objekte entstehen [SRRP98].

Die entwickelten parallelen Methoden wurden in ein Bildgenerierungssystem integriert, welches die Vorteile von Computersystemen mit spezieller Grafikhardware und Parallelrechnern mit verteiltem Speicher nutzt, um Anwendern eine interaktive Begehung (*Walkthrough*) korrekt beleuchteter virtueller Umgebungen zu ermöglichen. Das System wurde in Form einer modularen Client/Server-Architektur konzipiert. Hierbei werden Client-Komponenten auf Computern mit spezieller Hardware gestartet, die eine beschleunigte Durchführung von Walkthrough-Animationen auf der Basis von berechneten Radiosity-Lösungen unterstützt. Die eigentliche Radiosity-Simulation wird als Server-Komponente auf einem Parallelrechner mit verteiltem Speicher datenparallel ausgeführt. Der Anwender kann zu jedem Zeitpunkt der globalen Beleuchtungssimulation Zwischenergebnisse zur Visualisierung anfordern oder Veränderungen der Szene an die Server-Komponente übermitteln. Bei der Implementierung der parallelen Radiosity-Methode wurde aus Gründen der Portabilität PVM (*Parallel Virtual Machine*) verwendet. Dieses ermöglicht die Nutzung einer großen Anzahl von Rechnersystemen für die parallele Beleuchtungssimulation. Das System hat einen hochgradig modularen Aufbau, ist einfach um zusätzliche Komponenten zu erweitern und kann einfach an die Bedürfnisse von Endanwendern angepaßt werden [SL97].

Ein wesentlicher Nachteil von leistungsfähigen Parallelrechnern ist deren Preis. Dieses führt dazu, daß der Anwenderkreis solcher HPC-Systeme im allgemeinen auf die Großindustrie und Forschungseinrichtungen beschränkt ist. Da es sich bei der Zielgruppe des in dieser Arbeit beschriebenen Systems um kleine Architektur- und Visualisierungsbüros handelt, ist davon auszugehen, daß auf Seiten der Anwender keine leistungsfähigen parallelen Systeme lokal vorhanden sind. Aus diesem Grund wurde ein *Remote-Rendering-System* entwickelt, welches die effizienten parallelen Methoden zur globalen Beleuchtungssimulation als Electronic-Commerce Dienstleistung über das Internet zugänglich macht. Im Rahmen des EU-Forschungsprojektes HiPEC (High Performance Visualization System supporting Networked Electronic Commerce Applications) wurde dieses Szenario erfolgreich in einer speziellen E-Commerce Applikation getestet [SL98]. Dieses Szenario wird in dem BMBF-Projekt HiQOS auf weitere Anwendungsfelder ausgedehnt, die über den Bereich der Architekturvisualisierung hinausgehen [SP00].

Ein Problem bei Walkthrough-Animationen in Szenen, für die mit Hilfe der Radiosity-Methode globale Beleuchtungsinformationen bestimmt wurden, wird durch die große Komplexität der Radiosity-Lösung verursacht. Um den Rechenaufwand von Radiosity-Verfahren in einem akzeptablen Bereich zu halten, werden initial zunächst große diskrete Flächenelemente (*Patches*) erzeugt. Die Patches werden im Verlauf der Berechnung an den Stellen, an denen der Helligkeitsgradient groß ist, adaptiv in kleinere Flächenelemente (Subpatches) unterteilt. Dieses ist zum Beispiel entlang von Schattengrenzen oder Flächen direkt gegenüber einer starken Lichtquelle der Fall. Da die Patches auf einem regelmäßigen Gitter liegen, ist zum Erfassen von Schattengrenzen, die nicht diesem Gitter folgen, eine sehr große Zahl von Subpatches notwendig. Die entstehenden zusätzlichen Patches erhöhen die Komplexität einer Szene erheblich und machen die Durchführung von interaktiven Walkthrough-Animationen oftmals unmöglich. Weiterhin verursachen derart komplexe Radiosity-Lösungen im Kontext des Remote-Rendering-Systems einen erheblichen Mehraufwand durch Kommunikation, da

die Lösungen über ein IP-Netzwerk mit geringer Bandbreite (Internet) zum Endanwender transferiert werden müssen. In diesem Fall wird die Effizienz des Gesamtsystems erheblich durch die Kommunikationszeiten reduziert.

Es besteht ein Bedarf an Optimierungsverfahren, welche die Komplexität der Netze auf ein geringes Maß reduzieren. Dabei soll die Szene nur möglichst kleine Qualitätseinbußen erleiden. Die Akzeptanzschwelle für Qualitätseinbußen hängt hierbei stark von der Anwendung ab. Ziel ist es, interaktive Begehungen korrekt beleuchteter Szenen und schnelle Internet-Übertragungen der Radiosity-Lösungen zu ermöglichen. Es existiert mittlerweile eine große Anzahl von Optimierungs-/Vereinfachungs-Algorithmen. Diese Algorithmen unterscheiden sich sehr stark in ihrer Laufzeit, ihrem Speicherbedarf und der erreichten Qualität. Aufgrund des enormen Speicherplatz- und Rechenzeitbedarfs dieser Algorithmen werden skalierbare parallele Methoden benötigt, die eine Vereinfachung von extrem komplexen, aus mehreren Millionen Polygonen bestehenden Netzen ermöglicht. In dem in dieser Arbeit beschriebenen Remote-Rendering-System werden zwei verschiedene Ansätze zur Vereinfachung komplexer Radiosity-Lösungen verwendet. Bei dem einen Verfahren handelt es sich um einen datenparallelen Ansatz, der eine Reduktion der Anzahl der Polygone in einer adaptiv verfeinerten Radiosity-Lösung durchführt [SR00]. Die Qualität der Radiosity-Lösung wird dabei nur in geringem Maße beeinträchtigt. Eine alternative verlustfreie sequentielle Methode wandelt die in dem komplexen hierarchisch aufgebauten Lösungs-Netz enthaltenen Radiosity-Informationen in Oberflächentexturen um, die auf eine grobe Repräsentation der Szenenobjekte gelegt werden. Beide Methoden tragen erheblich zur Effizienzsteigerung des Client/Server-Visualisierungssystems bei, da die interaktive Begehung komplexester Architekturmodelle signifikant beschleunigt wird. Die Akzeptanz des Remote-Rendering-Systems durch die Endbenutzer profitiert ebenfalls von der Vereinfachung der erzeugten Lösungen, da der Kommunikations-Overhead des Systems reduziert wird und die Nachbearbeitung der beleuchteten Szenen (*Postproduction*) selbst auf den oftmals sehr leistungsschwachen Computersystemen der Anwender möglich ist. In dieser Arbeit liegt der Fokus auf der Simulation der globalen Beleuchtung. Aus diesem Grund wird auf die Optimierungsverfahren für erzeugte Radiosity-Lösungen nicht im Detail eingegangen. Der interessierte Leser wird für eine genauere Beschreibung auf [SR00] verwiesen.

1.5 Überblick

Die vorliegende Arbeit gliedert sich in sieben Kapitel. Die physikalischen Grundlagen der globalen Beleuchtungssimulation werden in Kapitel 2 eingehend betrachtet. Eine der wesentlichen Methoden in der Computergrafik zur realistischen Bildgenerierung unter Berücksichtigung der in virtuellen Umgebungen vorherrschenden Beleuchtungssituation ist die Radiosity-Methode. Diese Methode wird in ihren Grundzügen in Kapitel 3 beschrieben. Weiterhin wird dort auf Mehrphasen-Methoden eingegangen, die als wesentliche Erweiterungen der klassischen Radiosity-Methode zu betrachten sind, und die es ermöglichen, den Realitätsgrad von synthetisch erzeugten Bildern erheblich zu steigern.

Die klassische Radiosity-Methode eignet sich zur Simulation der globalen Beleuchtung in statischen Szenen. Während der Durchführung einer virtuellen Begehung (*Online-Walkthrough-Animation*) in einem mit Hilfe der Radiosity-Methode beleuchteten Modell ist es lediglich gestattet, den Standpunkt des Betrachters innerhalb einer Szene frei zu variieren, ohne daß bei einer Veränderung des Standpunktes jeweils eine neue Radiosity-Lösung bestimmt werden muß. Sobald jedoch der Benutzer eine Veränderung in der Szene vornimmt, muß die Simulation der globalen Beleuchtung erneut ausgeführt werden. In Kapitel 4 wird ein Überblick über existierende Radiosity-Verfahren gegeben, die dynamische Veränderungen

der Szene unterstützen, wobei eine kostspielige Neuberechnung der Radiosity-Lösung im Anschluß an eine Interaktion vermieden wird.

Die zentralen Ergebnisse dieser Arbeit sind in Kapitel 5 zusammengefaßt. Dort werden zwei verschiedene datenparallele Progressive-Refinement-Verfahren zur Berechnung von Radiosity-Lösungen in hochgradig komplexen Szenen beschrieben. Beide Verfahren unterstützen sowohl statische als auch dynamisch veränderbare Szenen und ermöglichen Mehrphasen-Simulationen, um eine möglichst große Anzahl der in realen Umgebungen auftretenden globalen Beleuchtungseffekte bei der Bildgenerierung zu berücksichtigen. Die Effizienz der Verfahren wird durch umfangreiche Testmessungen dokumentiert, die auf Basis von Modelldaten mit unterschiedlicher Komplexität durchgeführt wurden.

In Kapitel 6 wird zunächst eine Systemintegration der zuvor eingeführten parallelen Verfahren beschrieben. Das Ergebnis dieser Integration ist ein Bildgenerierungssystem mit Client/Server-Architektur, welches es dem Anwender erlaubt, effiziente parallele Verfahren zur globalen Beleuchtungssimulation auf einfache Art und Weise zu nutzen. Durch ein Remote-Rendering-System ist es möglich, die parallelen Verfahren einer breiteren Anwenderschicht über Internet zur Verfügung zu stellen. Die Architektur dieses Systems sowie die einzelnen Komponenten werden in ihren Grundzügen ebenfalls in Kapitel 6 beschrieben. Weiterhin ist dort ein Überblick der verschiedenen Anwendungsszenarios des Remote-Rendering-Systems zu finden.

Den Abschluß der Arbeit bildet eine kurze Zusammenfassung der zentralen Ergebnisse in Kapitel 7.

Kapitel 2

Beleuchtungssimulation in der Computergrafik

Die Linie ist das Mittel, mit dem sich der Mensch die Wirkung des Lichtes auf die Dinge klarmacht, aber in der Natur gibt es keine Linien, da ist alles ausgefüllt: man modelliert, wenn man zeichnet, das heißt, man löst die Dinge aus ihrer Umgebung heraus und nur die Verteilung der Helligkeit verleiht dem Körper Erscheinung!

(Honoré de Balzac: Das unbekannte Meisterwerk)

Wie schon im vorangegangenen Kapitel ausgeführt, ist die Erzeugung realistischer Bilder von dreidimensionalen Objekten eine der zentralen Aufgabenstellungen der Computergrafik. Bei der photorealistischen Bildgenerierung sind vorrangig drei Teilprobleme zu bewältigen:

- Die perspektivische Abbildung eines Modells auf eine zweidimensionale Bildfläche,
- die korrekte Sichtbarkeitsberechnung
- und die Beleuchtung von Oberflächen.

Die Beleuchtungsberechnung ist der aufwendigste Teil des Bildgenerierungsprozesses. Computergrafiken werden zumeist auf Bildschirmen oder anderen in Matrixform gerasterten Ausgabemedien dargestellt, wobei bei einem Betrachter der Eindruck von maximaler Realitätsnähe erweckt werden soll. Ein gewisser Grad an Photorealismus wird bei computergenerierten Bildern mit Hilfe von Beleuchtungsmodellen erreicht. Fortschrittliche Beleuchtungsmodelle basieren auf der Modellierung des physikalischen Verhaltens von Licht und der Berücksichtigung der Eigenschaften des menschlichen visuellen Wahrnehmungssystems. In den vergangenen Jahren wurde bei der photorealistischen Bildgenerierung Verbesserungen vor allen Dingen aufgrund eines besseren Verständnisses der physikalischen Vorgänge bei der Lichtausbreitung erzielt.

Eine Beschreibung einer dreidimensionalen Umgebung besteht aus Lichtquellen, Objekten mit verschiedenen geometrischen Formen und deren optischen Materialeigenschaften. Basierend auf einer solchen Beschreibung kann simuliert werden, in welcher Art und Weise das von Lichtquellen emittierte Licht durch Objektmaterialien gestreut, reflektiert, gebrochen oder absorbiert wird. Die aus diesem physikalischen Vorgang der Lichtausbreitung in einer Umgebung resultierende Lichtenergieverteilung wird anschließend von einer Kamera oder der Netzhaut wahrgenommen. Durch die möglichst exakte Approximation dieser Lichtenergieverteilung bei der synthetischen Bildgenerierung kann der Realitätsgrad der Bilder erheblich gesteigert werden.

In den Anfängen der Computergrafik wurden empirische Beleuchtungsmodellen verwendet, die von einem sehr idealisierten Reflexionsverhalten und von punktförmigen Lichtquellen

ausgehen und auf komplexe Schattenberechnung verzichten. Die indirekte Beleuchtung durch andere Objekte in der Szene wird durch einen konstanten ambienten Beleuchtungswert approximiert. Im Zusammenhang mit dieser Art von Beleuchtungsberechnung wird von *lokaler Beleuchtung* gesprochen, da nur die direkte Wechselwirkung zwischen Objekten und Lichtquellen modelliert ist.

Die indirekte Beleuchtung von Objekten, die durch reflektiertes oder transmittiertes Licht entsteht, wird als *globale Beleuchtung* bezeichnet. Die Approximation der globalen Beleuchtung durch einen konstanten Wert reicht jedoch nicht aus, um Beleuchtungseffekte wie Halbschatten, Spiegelungen von Objekten oder Color-Bleeding zu modellieren. Algorithmen, die sowohl die lokale als auch die globale Beleuchtung von Oberflächen berücksichtigen, versuchen Interaktionen von Licht mit Objekten einer gegebenen Szenen möglichst genau zu rekonstruieren. Das Problem der globalen Beleuchtung für Umgebungen mit vorwiegend spiegelnden Oberflächen wird in einem großen Ausmaß von vielen der bekannten und weit verbreiteten Verfahren basierend auf Raytracing [WHI80] gelöst. Eine weitere Möglichkeit globale Beleuchtungseffekte zu simulieren, die durch diffuse Inter-Objektreflexionen des Lichtes hervorgerufen werden, bietet die Radiosity-Methode [GTGB84] [NN85]. Die wesentlichen Unterschiede zwischen diesen beiden Verfahren wurden schon in Kapitel 1 erläutert. Die diesen globalen Beleuchtungsverfahren zugrundeliegenden Kenngrößen und physikalischen Vorgänge werden ausführlich im weiteren Verlauf dieses Kapitel beschrieben. Die Betrachtungen werden hierbei von den folgenden zentralen Fragestellungen geleitet:

1. Was ist Licht und wie kann es charakterisiert werden?
2. Wie ist die Reflexion von Licht an Objektoberflächen charakterisiert?
3. Wie läßt sich die räumliche Ausbreitung von Licht mathematisch beschreiben?

2.1 Physikalische Grundlagen der Beleuchtungssimulation

Die globale Beleuchtungsberechnung basiert auf der Simulation des Austausches von Lichtenergie zwischen Objektoberflächen einer Umgebungsbeschreibung. Aus physikalischer Sicht kann Energie im Raum auf drei verschiedene Arten transportiert werden. Als Fluß von Energie durch die Struktur eines Materials oder Mediums (*Conduction*), als Energietransport durch Bewegung des Mediums (*Convection*) sowie als Energietransport durch elektromagnetische Felder (*Radiation*). Die bei der Beleuchtungssimulation vorgenommenen Betrachtungen beinhalten im allgemeinen eine wesentliche Vereinfachung des generellen Problems, nämlich daß der Austausch von Energie zwischen Oberflächen einer Umgebung ohne Transportmedium, d.h. im Vakuum stattfindet. Es ist zu beachten, daß ein Energietransport im Vakuum nur durch Strahlung möglich ist. Somit basieren die Modelle, die zu Zwecken der Beleuchtungssimulation in der Computergrafik eingesetzt werden auf dem Energietransport durch elektromagnetische Felder und somit durch Strahlung.

Strahlung ist definiert als das Aussenden von elektromagnetischer Energie von einem Objekt, welches als Strahlungsquelle bezeichnet wird. Zur Charakterisierung von Strahlung existieren zwei verschiedene physikalische Modellvorstellungen: das Wellenmodell und das einfachere Teilchenmodell (geometrische Optik). Diese beiden Modelle werden durch die Quantenmechanik vereinheitlicht. Eine quantentheoretische Beschreibung des Lichtes ist für die im Rahmen der Beleuchtungssimulation angestellten Betrachtungen kaum sinnvoll, da die zu simulierenden Vorgänge makroskopische Dimensionen besitzen. Den weiteren Betrachtungen und Definitionen in diesem Kapitel liegt das Teilchenmodell zugrunde.

Nur ein kleiner Teil des gesamten elektromagnetischen Spektrums kann vom menschlichen Auge wahrgenommen werden. Dieser Frequenzbereich, der als optische Strahlung bezeichnet wird und für die Belange der Computergrafik relevant ist, liegt zwischen 380 nm (Ultraviolett) und 770 nm (Infrarot). Je nach Anwendung kann optische Strahlung grundsätzlich unter zwei voneinander verschiedenen Aspekten bewertet werden. Einerseits läßt sie sich durch rein physikalische Größen charakterisieren, andererseits aber auch durch rein physiologische Größen, welche die Empfindlichkeitscharakteristik des menschlichen Auges einbeziehen. Im ersten Fall handelt es sich um die *Radiometrie* und strahlungsphysikalische Größen, im zweiten Fall um die *Photometrie* und lichttechnische Größen. Kenngrößen sind bei diesen Ansätzen analog definiert, werden jedoch im jeweiligen Anwendungsgebiet der Physik mit unterschiedlichen Einheiten verwendet. Einige Anwendungen der Computergrafik, allen voran die Lichttechnik, erfordern die Berücksichtigung lichttechnischer Kenngrößen. Aufgrund der analogen Definition der Kenngrößen gestaltet sich eine Umrechnung der radiometrischen Werte in lichttechnische Werte, unter Berücksichtigung der farbabhängigen Helligkeitsempfindung des menschlichen Auges, unproblematisch.

Im weiteren Verlauf dieses Kapitels werden die wichtigsten strahlungsphysikalischen Größen sowie die entsprechenden lichttechnischen Größen eingeführt, wobei sich diese im wesentlichen formal nur durch ihre jeweiligen Einheiten unterscheiden. Im Anschluß daran werden die Grundlagen des Strahlungsaustausches zwischen Oberflächen einer Umgebung beschrieben, welcher durch Emission und Reflexion verursacht wird. Hierbei wird der Schwerpunkt auf die Modellierung dieses Strahlungsaustausches durch globale Beleuchtungsmodelle der Computergrafik gelegt.

2.1.1 Strahlungsphysikalische Größen

Die Radiometrie beschäftigt sich mit der physikalischen Messung von elektromagnetischer Strahlung. Elektromagnetische Strahlung transportiert Energie durch den Raum und kann sowohl als Welle, als auch als Teilchen aufgefaßt werden. Die grundlegende Größe in der Radiometrie ist die *Strahlungsenergie* (engl. *Radiant Energy*), die mit dem Symbol Q bezeichnet und in Joule (J) gemessen wird. Unter der Strahlungsenergie Q wird die von einer Strahlungsquelle emittierte Strahlungsmenge verstanden (optisches Spektrum), die sich in Form elektromagnetischer Strahlung ausbreitet.

Die *Strahlungsleistung* oder auch der *Strahlungsfluß* (engl. *Radiant Flux*) bezeichnet die Strahlungsenergie, die von einer Fläche pro Zeiteinheit abgestrahlt wird oder pro Zeiteinheit auf einer Fläche auftrifft. Sie wird gemessen in Joule pro Sekunde (J/s) oder in Watt (W) und ist definiert als Strahlungsenergie pro Zeiteinheit:

$$\Phi = \frac{dQ}{dt} \quad (2.1)$$

Im allgemeinen kann nicht vorausgesetzt werden, daß eine Strahlungsquelle in jedem Zeitintervall gleich viel Energie abstrahlt. Bei der globalen Beleuchtungssimulation, die Gegenstand dieser Arbeit ist, werden jedoch üblicherweise zeitunabhängige Energieverteilungen betrachtet. Aus diesem Grund wird die Strahlungsleistung im weiteren Verlauf als zeitlich konstant angenommen, wodurch die einfachere Form der Gleichung (2.1) verwendet werden kann:

$$\Phi = \frac{Q}{t} \quad (2.2)$$

Durch die *Strahlungsstärke* I , auch *Strahlungsintensität* (engl. *Radiant Intensity*) genannt, wird die von einer Strahlungsquelle ausgesandte Strahlungsleistung pro Raumwinkel $d\omega$ beschrieben. Diese wird auch als *Raumwinkeldichte* der Strahlungsleistung bezeichnet. Ihre Einheit ist Watt pro Steradian (W/sr).

$$I = \frac{d\Phi}{d\omega} \quad (2.3)$$

Die Strahlungsstärke beschreibt die Richtungsabhängigkeit der ausgesandten Strahlung. Unter dem Raumwinkel ist hierbei die dreidimensionale Erweiterung des Bogenmaßes zu verstehen. Obwohl der Raumwinkel $d\omega$ eigentlich eine dimensionslose Größe ist, wird *Steradian* (sr) verwendet. Wird die Einheitskugel betrachtet, so ergibt sich 4π Steradian als maximaler Raumwinkel (die gesamte Fläche der Einheitskugel). Analog zum Bogenmaß im zweidimensionalen Fall ist der Raumwinkel, den ein Objekt von einem Punkt P aus betrachtet einnimmt, definiert als der Flächeninhalt der Zentralprojektion des Objektes auf die Einheitskugel mit Zentrum P (siehe Abb. 2.1).

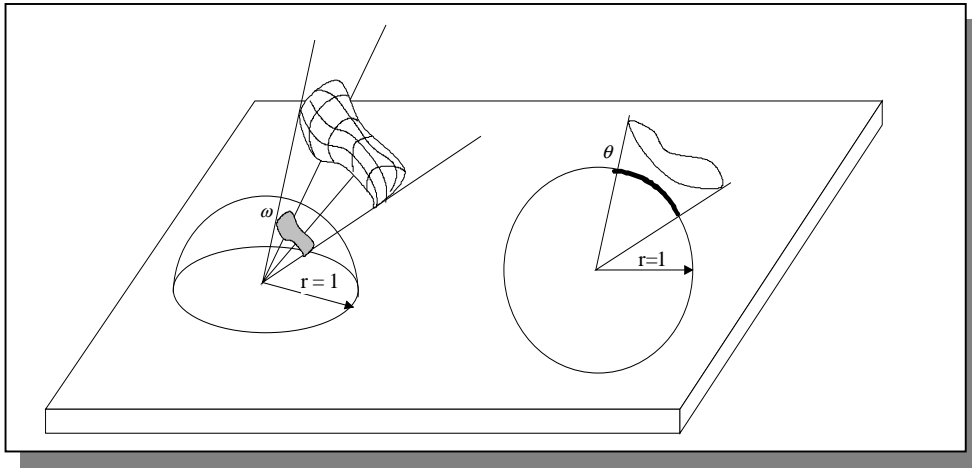


Abb. 2.1: Der Raumwinkel im Vergleich zum Bogenmaß.

Ein Maß für die pro Fläche auftreffende Strahlungsleistung ist die *Bestrahlungsstärke* E (engl. *Radiant Flux Density* oder *Irradiance*), mitunter auch Strahlungsleistungsdichte genannt. Sie bezeichnet die auf einem Empfänger aus beliebigen Richtungen auftreffende oder eine Fläche durchdringende Strahlungsleistung pro Flächeneinheit, gemessen in W/m^2 . Es ist

$$E = \frac{d\Phi}{dA} \quad (2.4)$$

Aufgrund von Eigenemission und/oder Reflexion kann Strahlungsleistung von einer Fläche abgegeben werden. Die *spezifische Ausstrahlung* M (engl. *Radiant Exitance* oder *Radiosity*) ist die auf ein selbst (primär) strahlendes oder reflektierendes (sekundär strahlendes) Flächenelement dA bezogene Strahlungsleistung $d\Phi$ einer in den Halbraum strahlenden Fläche A mit

$$M = \frac{d\Phi}{dA} \quad (2.5)$$

Hierbei wird eine homogene Ausstrahlung der Fläche vorausgesetzt. Bei nicht homogenen Strahlern wird die Strahlungsleistung Φ als Integral über alle strahlenden bzw. reflektierenden Teilflächen errechnet, woraus sich die mittlere spezifische Ausstrahlung ergibt. Die Einheit der spezifischen Ausstrahlung ist W/m^2 .

Die Abhängigkeit der Strahlungsleistung von der Richtung wird durch die Strahlungsstärke (2.3) und deren Flächenabhängigkeit durch die Bestrahlungsstärke (2.4) bzw. spezifischer Ausstrahlung (2.5) beschrieben. Mit Hilfe der *Strahlungsdichte* L (engl. *Radiance*) wird nun eine flächen- und raumwinkelspezifische Strahlungsleistung definiert:

$$L(x, \theta) = \frac{d^2\Phi}{dA_x(\cos\theta d\omega)} \quad (2.6)$$

Diese Größe läßt sich bildlich durch einen Lichtstrahl veranschaulichen, der einen Punkt x , der von einer unendlich kleinen Fläche dA_x umgeben ist, in eine durch den Winkel θ spezifizierte Richtung verläßt, oder aus dieser Richtung auf den Punkt x auftrifft. Hierbei ist θ der Winkel zwischen der Richtung der auf eine unendlich kleine Fläche dA_x einfallenden (oder von ihr abgegebenen) Strahlungsleistung und der Flächennormalen von dA_x . Durch $d\omega$ wird der differentielle Raumwinkel angegeben, in dem Strahlung abgegeben wird, oder aus dem Strahlung einfällt (siehe Abb. 2.2). Strahlungsdichte ist die Strahlungsleistung, welche durch den Lichtstrahl transportiert wird.

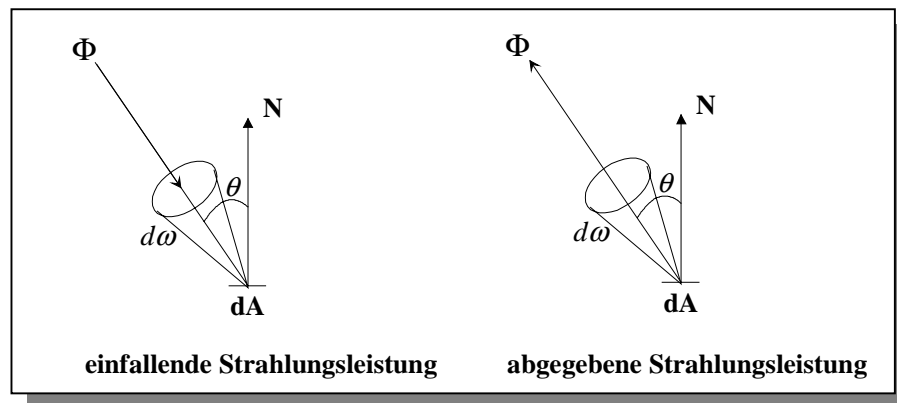


Abb. 2.2: Einfallende und abgegebene Strahlungsleistung eines Oberflächenpunktes in eine gegebene (bzw. aus einer gegebenen) Richtung.

Für die Strahlungsdichte ergibt sich durch Umformung von (2.3) und anschließendes Einsetzen in (2.6):

$$L(x, \theta) = \frac{dI_\theta}{dA_x \cos\theta} \quad (2.7)$$

Der Nenner $dA_x \cos(\theta)$ drückt hierbei die Tatsache aus, daß eine Fläche dA_x aus der durch θ spezifizierten Richtung um den Faktor $\cos(\theta)$ verkürzt erscheint (siehe Abb. 2.3).

Somit ergibt sich für Gleichung (2.7) eine anschauliche Interpretation. Die Strahlungsdichte ist die Strahlungsleistung pro Raumwinkel der aus (bzw. in) Richtung θ gesehenen und somit

um $\cos(\theta)$ verkürzten Fläche dA_x . Die Einheit der Strahlungsdichte ist Watt pro Flächeneinheit pro Steradian ($W/m^2/sr$).

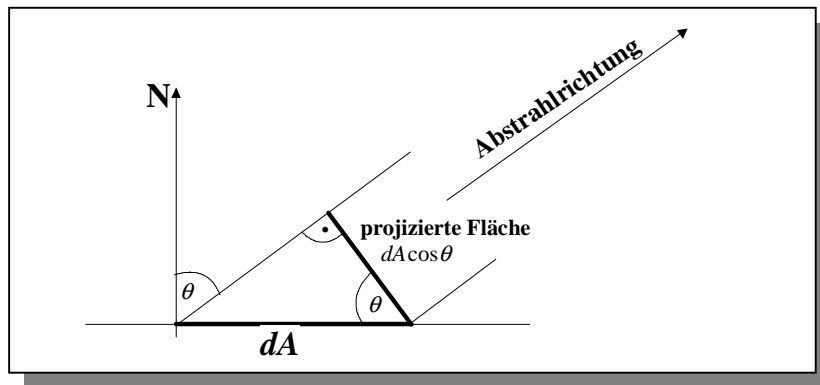


Abb. 2.3: Das unter dem Winkel θ betrachtete kleine Flächenelement dA erscheint um den Faktor $\cos(\theta)$ verkleinert.

Im Rahmen dieser Arbeit sind die sogenannten *Lambertstrahler* und *Lambertreflektoren* von besonderer Wichtigkeit. Oberflächen mit *lambertschen* Eigenschaften haben eine konstante, vom Betrachtungswinkel unabhängige Strahlungsdichte und werden oftmals auch als Oberflächen mit *ideal diffusen Reflexions-* und/oder *Emissionseigenschaften* bezeichnet. In der Realität existieren keine Objekte mit solchen Eigenschaften, wenngleich die Oberfläche der Sonne oder eine matte Glühlampe den Eigenschaften eines Lambertstrahlers sehr nahe kommen, während Kreide oder Gips annähernd ideal diffuse Reflexionseigenschaften aufweisen. Innerhalb der Computergrafik werden aus Gründen, die im weiteren Verlauf der Arbeit noch dargestellt werden, viele Oberflächenmaterialien mit lambertschen Eigenschaften definiert.

Bei Betrachtung eines unendlich kleinen Flächenelementes dA , ist dieses äquivalent zu einer punktförmigen Strahlungsquelle, für welche die abgegebene Strahlung durch geometrische Strahlen modelliert werden kann. Die Strahlungsintensität I_θ eines solchen Strahles mit einem Winkel θ zur Oberflächennormalen ist durch das *Cosinusetz von Lambert (1760)* gegeben:

$$I_\theta = I_N \cos \theta \quad (2.8)$$

Hierbei ist I_N die Strahlungsintensität des Strahles, welcher die Oberfläche in Richtung der Flächennormalen verläßt. Die Strahlungscharakteristik eines Lambertstrahlers (-reflektors) ist also eine Kugel, welche die Strahlungsoberfläche tangiert (siehe Abb. 2.4).

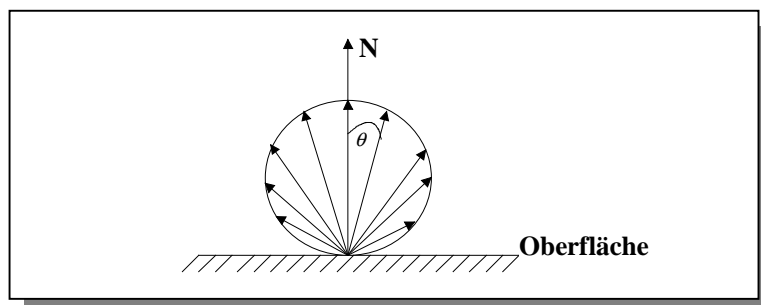


Abb. 2.4: Emission eines perfekten Lambertstrahlers.

Für die Strahlungsdichte L_d des Punktes x einer Oberfläche mit lambertschen Eigenschaften, der von einer unendlich kleinen Fläche dA_x umgeben ist, ergibt sich aus (2.7) und (2.8):

$$L_d(x) = \frac{dI_N}{dA_x} \quad (2.9)$$

Ein wichtiger Punkt für das Radiosity-Verfahren, das im weiteren Verlauf dieser Arbeit eine wesentliche Rolle spielt, ist die Formulierung der spezifischen Ausstrahlung mittels der Leuchtdichte einer Oberfläche.

Wie aus Gleichung (2.5) hervorgeht, ist die spezifische Ausstrahlung (Radiosity) definiert als die von einem Oberflächenpunkt abgegebene Strahlungsleistung pro Flächeneinheit. Im weiteren Verlauf sei durch $B(x)$ die spezifische Ausstrahlung einer Oberfläche an einem Punkt x der Fläche bezeichnet. Dieses geschieht, um konform mit der Literatur im Bereich der Computergrafik zu bleiben, die sich mit dem Thema Radiosity auseinandersetzt. Die Fachliteratur der Physik verwendet den Buchstaben M (und somit $M(x)$), um die spezifische Ausstrahlung eines Oberflächenpunktes zu bezeichnen.

Da die spezifische Ausstrahlung $B(x)$ als gesamte Strahlungsleistung definiert ist, die pro Flächeneinheit von einem Punkt x einer leuchtenden Fläche abgegeben wird gilt:

$$B(x) = \frac{d\Phi}{dA_x} \quad (2.10)$$

Somit läßt sich $B(x)$ mit Hilfe von Gleichung (2.6) und Integration über der Hemisphäre oberhalb von x wie folgt beschreiben (zur Herleitung sie auch [SP94]):

$$B(x) = \int_{2\pi} L(x, \theta) \cos \theta \, d\omega \quad (2.11)$$

Hierbei wird über die gesamte Hemisphäre oberhalb des lichtabstrahlenden Punktes integriert. Erfolgt die Abstrahlung ideal diffus, so ist die Strahlungsdichte unabhängig von der Abstrahlrichtung und der Funktionsterm $L(x, \theta)$ kann durch den von θ unabhängigen Term $L_d(x)$ (Gleichung (2.9)) ersetzt werden. Somit erhalten wir:

$$B(x) = \int_{2\pi} L_d(x) \cos \theta \, d\omega = L_d(x) \int_{2\pi} \cos \theta \, d\omega \quad (2.12)$$

Aus (2.12) folgt durch Bestimmung des Integrals die wichtige Beziehung:

$$B(x) = \pi L_d(x) \quad (2.13)$$

Die durch (2.13) gegebene Beziehung zwischen der Strahlungsdichte einer Oberfläche mit ideal diffusen Eigenschaften und der spezifischen Ausstrahlung einer Oberfläche ist wesentlicher Bestandteil der Herleitung einer vereinfachten globalen Beleuchtungsfunktion, die ein Energiegleichgewicht in einer ideal diffus reflektierenden Umgebung beschreibt (siehe Abschnitt 2.5). Diese Beleuchtungsfunktion ist die Grundlage der in Kapitel 3 im Detail beschriebenen Radiosity-Methode.

2.1.2 Lichttechnische Größen

Die Photometrie beschäftigt sich mit der Messung von Strahlungsenergie im elektromagnetischen Spektrum zwischen Ultraviolett (380 nm) und Infrarot (770 nm), d.h. dem sichtbaren Bereich des gesamten Spektrums. Hierbei werden die gemessenen Größen gemäß der Empfindlichkeit des menschlichen Auges gewichtet. Die Gewichtung basiert auf einem statistischen Modell der Empfindlichkeit des menschlichen Auges bzgl. der Helligkeit von Licht. Betrachtet über die Wellenlängen des sichtbaren Spektrums schwankt diese Helligkeits-Empfindlichkeit. So erscheint für einen menschlichen Betrachter eine grüne Lichtquelle mit einer Strahlungsdichte von $1 \text{ W/m}^2/\text{sr}$ erheblich heller, als eine rote Lichtquelle mit einer Strahlungsdichte von $1 \text{ W/m}^2/\text{sr}$ [Ash94]. Innerhalb der Photometrie wird somit nicht die Strahlungsleistung in Watt gemessen. Vielmehr wird versucht, den subjektiven Eindruck zu messen, der bei der Reizung des menschlichen visuellen Systems durch Strahlungsenergie verursacht wird. Diese Aufgabe wird erheblich durch die Tatsache erschwert, daß die Empfindlichkeit des Auges nicht linear auf die Veränderung der Helligkeit des Lichtes reagiert. Die Empfindlichkeit verändert sich nicht nur mit der Wellenlänge der Strahlungsenergie (d.h. Farbe), sondern auch mit der Menge der transportierten Strahlungsleistung, der räumlichen Komplexität der wahrgenommenen Umgebung, dem Anpassungszustand der Iris und der Retina, dem physischen und psychischen Zustand des Betrachters, sowie einer Vielzahl von weiteren Parametern.

Die Photometrie wurde 1729 von Pierre Bouguer [Boug] begründet, indem menschliche Beobachter die Helligkeit von verschiedenen Lichtquellen vergleichen sollten. Eine unbekannte Lichtquelle wurde mit Standardlichtquellen bekannter Helligkeit (zu dieser Zeit Kerzen) verglichen, um die relative Helligkeit festzustellen. Hierbei konnte Bouguer beobachten, daß ein menschlicher Beobachter keine Angabe darüber machen konnte, um welchen Faktor eine Lichtquelle heller ist als eine andere, aber zuverlässig darüber Auskunft geben konnte, ob zwei Lichtquellen die gleiche Helligkeit besitzen.

Die Quantifizierung der subjektiven Wahrnehmung des Helligkeitsempfindens erfolgte 1924 durch die CIE (Commission Internationale d' Eclairage). Hierbei wurden statistische Untersuchungen unter normalen Sichtbedingungen durchgeführt, bei denen Probanden visuell die Helligkeit von monochromen Lichtquellen mit verschiedenen Wellenlängen unter kontrollierten Bedingungen einschätzen sollten. Das statistische Ergebnis ist die sogenannte CIE Photometrie-Kurve (Hellempfindlichkeitsfunktion), welche die relative Empfindlichkeit des Auges in Abhängigkeit von der Wellenlänge des wahrgenommenen Lichtes widerspiegelt (siehe Abb. 2.5). Diese Kurve gibt eine Gewichtungsfunktion, mit deren Hilfe radiometrische in photometrische Werte umgerechnet werden können.

Die Theorie der Photometrie beschreibt nicht wie der Mensch Farben wahrnimmt. Das gemessene Licht kann sowohl monochrom als auch eine Kombination verschiedener Wellenlängen sein. Die Reaktion des Auges auf dieses Licht kann mit Hilfe der Hellempfindlichkeitsfunktion bestimmt werden. Der einzige Unterschied zwischen der Theorie der Radiometrie und der Theorie der Photometrie besteht in den Einheiten der gemessenen Größen.

Im weiteren Verlauf dieses Abschnittes werden die verschiedenen Kenngrößen der Photometrie eingeführt, da diese für die Zwecke der Innenarchitekturvisualisierung, der Beleuchtungstechnik und des Beleuchtungsdesigns von grundlegender Bedeutung sind und die mit Hilfe der Radiosity-Methode (siehe Kapitel 3) berechneten Ergebnisse quantitativ in der Terminologie der Zielgruppe ausgedrückt werden können.

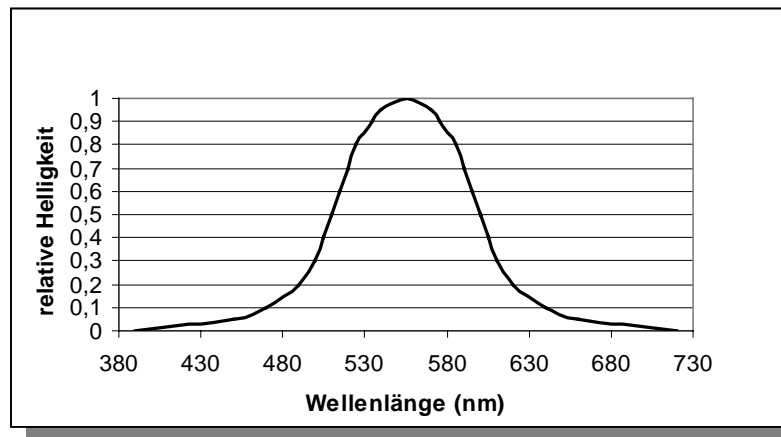


Abb. 2.5: Relative Hellempfindung von Strahlung im sichtbaren Spektralbereich.

Die *Lichtstärke* (engl. *Luminous Intensity*) ist Basisgröße des internationalen Einheitensystems und wird in Candela (*cd*) gemessen. Die internationale standardisierte Bezugsgröße ist hierbei eine theoretische Punktlichtquelle, welche die *Lichtstärke* von einem Candela hat. Diese Lichtquelle emittiert monochromes Licht der Wellenlänge 540 THz mit einer Strahlungsstärke (-intensität) von $1/683 \text{ W/sr}$ in Richtung des Betrachters. Die spezifizierte Wellenlänge entspricht ungefähr 555 nm , was mit der Wellenlänge des maximalen Wertes der Hellempfindlichkeitsfunktion übereinstimmt. Diese Basisgröße stellt somit zusammen mit der Hellempfindlichkeitsfunktion, ein Hilfsmittel zur Umrechnung von radiometrischen in photometrische Werte dar.

Im Fall einer monochromen Punktlichtquelle, welche Strahlung der Wellenlänge 510 nm und einer Strahlungsstärke (-intensität) von $1/683 \text{ W/sr}$ abgibt, hat die Hellempfindlichkeitsfunktion bei 510 nm einen Wert von $0,503$. Somit hat die Punktlichtquelle eine Lichtstärke von $0,503 \text{ cd}$.

Lichtstrom (engl. *Luminous Power*) ist die mit Hilfe der Hellempfindlichkeitsfunktion gewichtete Strahlungsleistung. Der Lichtstrom wird durch die Einheit *Lumen* (*lm*) gekennzeichnet. Das photometrische Gegenstück zur Bestrahlungsstärke ist die *Beleuchtungsstärke*, welche in *Lux* (*lx*) gemessen wird. Die Größe *Lux* ist in der Beleuchtungstechnik eine sehr gebräuchliche Einheit. So braucht beispielsweise ein Mensch zum Lesen eine Beleuchtungsstärke von mindestens 100 lx . Für normales Arbeiten werden 500 lx und für Präzisionsarbeit 1000 lx benötigt.

Mit Hilfe der im weiteren Verlauf der Arbeit vorgestellten Radiosity-Methoden, lassen sich somit die Beleuchtungssituationen beispielsweise an Büroarbeitsplätzen simulieren, bevor der Gebäudekomplex gebaut wurde und die Beleuchtung der Arbeitsplätze gemäß der gesetzlichen Anforderungen planen. Ähnliches gilt natürlich auch für die Planung von Bühnendekorationen im Theater oder der Beleuchtung von Messeständen.

Alle weiteren photometrischen Größen und deren entsprechenden strahlungsphysikalischen Größen sind in Tabelle 2.2.1 angegeben.

Radiometrie				Photometrie			
Bezeichnung		Zeichen	Einheit	Bezeichnung		Zeichen	Einheit
deutsch	englisch			deutsch	englisch		
Strahlungsleistung (Strahlungsfluß)	Radiant Power Radiant Flux	Φ_e	W Watt (Joule/sec)	Lichtstrom	Luminous Power Luminous Flux	Φ_v	lm (Lumen)
Bestrahlungsstärke (spezifische Ausstrahlung)	Radiant Flux Density oder Irradiance (Radiosity oder Radiant Existance)	E_e (M,B)	W/m^2	Beleuchtungsstärke (spez. Lichtausstrahlung)	Illuminance (Luminosity)	E_v	$lx=lm/m^2$ (Lux)
Strahlungsdichte	Radiance	L_e	$W/m^2/sr$	Leuchtdichte	Luminance	L_v	cd/m^2 ($lm/m^2/sr$)
Strahlungsstärke	Radiant Intensity	I_e	W/sr	Lichtstärke	Luminous Intensity	I_v	$cd=lm/sr$

Tabelle 2.2.1: Strahlungsphysikalische und lichttechnische Größen (Index $e = \text{energetisch}$; Index $v = \text{visuell}$)

2.1.3 Strahlungsaustausch zwischen Oberflächen

In den vorangegangenen Abschnitten wurde schon erläutert, daß Licht elektromagnetische Strahlung ist, welche vom menschlichen Auge wahrgenommen werden kann. Somit lassen sich Theorien und Verfahren aus Forschungsbereichen wie der Optik, der Beleuchtungstechnik und der Thermodynamik bei der photorealistischen Bildgenerierung zur Simulation globaler Beleuchtungsphänomene verwenden. Ein für die Computergrafik im besonderen Maße relevanter Bereich ist hierbei die geometrische Optik, da sich dieser mit der Beschreibung makroskopischer Eigenschaften von Licht beschäftigt. Die Gesetzmäßigkeiten der geometrischen Optik werden in der Computergrafik verwendet, um geeignete Simulationsmodelle zur Beschreibung der Ausbreitung von Lichtenergie in einer gegebenen Umgebung zu entwickeln. Hierbei werden neben den lokalen Wechselwirkungen des Lichtes mit den einzelnen Objektflächen einer Szene auch Emissions, Reflexions- und Transmissionsprozesse betrachtet, die sich zwischen sämtlichen Oberflächen einer Umgebungsbeschreibung ergeben. Erst durch Berücksichtigung dieser komplexen Wechselwirkungen können Beleuchtungsphänomene simuliert werden, die durch Mehrfachreflexion entstehen.

Die Emission von Strahlungsenergie wird von einer Vielzahl physikalischer Prozesse, wie Wärmeabstrahlung, chemische Reaktionen (Lumineszenz) oder Phosphoreszenz (verzögerte Emission – Nachleuchten) verursacht. Bei der Bildgenerierung spielt es im allgemeinen keine Rolle, welcher physikalische Prozeß für die Emission von Strahlung verantwortlich ist. Lediglich die Strahlungsleistung als Funktion der Position des Emitters sowie der Richtung der Ausstrahlung ist von Interesse. Diese Information wird, wie in 2.1.1 erläutert, durch Spezifikation der Strahlungsdichte (bzw. Leuchtdichte) einer Lichtquelle vorgegeben.

Fällt Strahlung auf eine Oberfläche, so kann diese reflektiert, transmittiert oder in kinetische Energie umgewandelt (absorbiert) werden. Da die Simulation der Transmission sich im wesentlichen nicht von der Reflexion unterscheidet, wird sie im weiteren Verlauf nicht explizit betrachtet. Das Hauptaugenmerk liegt auf der Beschreibung des Reflexionsverhaltens von Lichtenergie.

2.1.3.1 Die bidirektionale Reflexions-Verteilungsfunktion (BRVF)

Das Aussehen einer Oberfläche unter verschiedenen Beleuchtungsverhältnissen hängt von der Art und Weise ab, wie Lichtenergie durch das Material der Oberfläche absorbiert und reflektiert wird. Hierbei ist zu beachten, daß die selektive Absorption von Licht bestimmter Wellenlängen durch Oberflächenmaterialien die Farbe eines Objektes festlegt. Weiterhin füh-

ren komplexe Oberflächenstrukturen von Objekten der realen Welt auf mikroskopischer Ebene zu komplizierten Reflexionseigenschaften auf makroskopischer Ebene.

Innerhalb der Computergrafik wird vereinfachend zwischen vier verschiedenen Arten der Reflexion unterschieden. Einerseits wird bei der *ideal diffusen* (Lambertreflektoren) und der *ideal spiegelnden Reflexion* von einem sehr idealisierten Reflexionsverhalten ausgegangen, welches nur für sehr wenige reale Materialien annähernd angenommen werden kann. Die *gerichtet diffuse Reflexion*, welche gleichmäßig aber nicht konstant ist [SAWG91] sowie die *grob spiegelnde Reflexion*, die einfallende Strahlung in eine endliche Menge von Raumwinkeln reflektiert [Heck91], entsprechen schon mehr den Eigenschaften real existierender Materialien. Die beiden letztgenannten Reflexionsklassen besitzen jedoch nur eine sehr unscharfe Definition, so daß sie teilweise überlappend sind. Diese Klassen wurden als Erweiterung der idealen Reflexionsmodelle eingeführt, um alle möglichen Arten der Lichtreflexion abzudecken.

In Abb. 2.6 werden die verschiedenen Reflexionsklassen dargestellt. Dort strahlt eine punktförmige Lichtquelle ihre Strahlungsenergie in einen kegelförmigen Bereich ab. In der oberen Bildreihe ist das Reflexionsverhalten des aus einem bestimmten Raumwinkel einfallenden Lichtes für Materialien mit unterschiedlichen Reflektivitäten angegeben. In der unteren Bildreihe ist eine Perspektive der Oberfläche aus der Hauptreflexionsrichtung der einfallenden Strahlung wiedergegeben. Hierbei ist deutlich die verschiedene Form der Glanzlichter zu erkennen, die auf den Oberflächen durch die jeweiligen Reflexionseigenschaften des Materials entstehen.

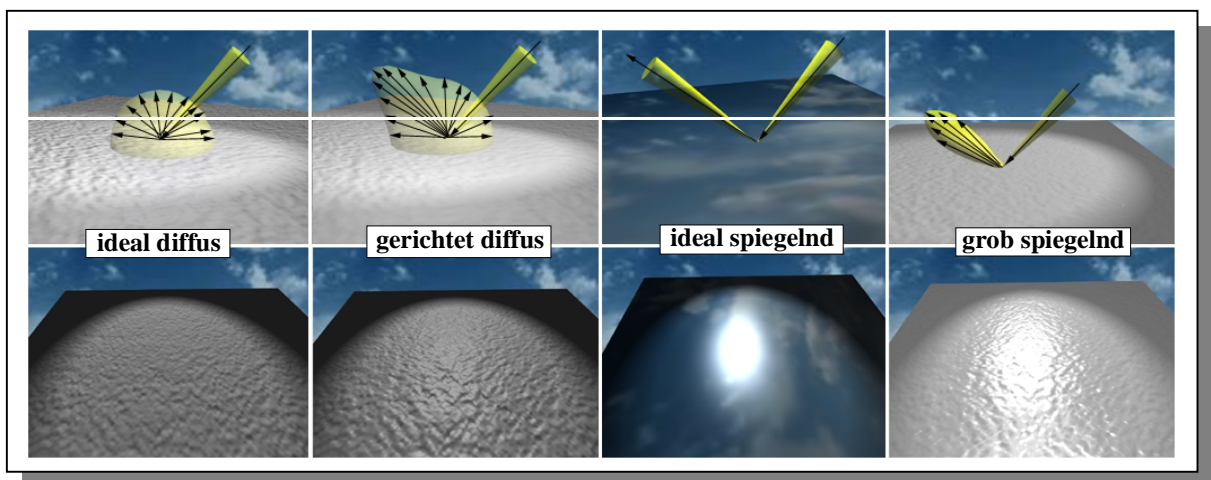


Abb. 2.6: Innerhalb der Computergrafik verwendete Reflexionsklassen.

Das Reflexionsverhalten von Oberflächen läßt sich vollständig durch die sogenannte bidirektionale Reflexions-Verteilungsfunktion (BRVF) beschreiben (siehe [SP94]). Diese Funktion ist definiert als das Verhältnis der Leuchtdichte L in Reflexionsrichtung (θ_o, ϕ_o) zur Strahlungsleistung Φ_i aus der Einfallrichtung (θ, ϕ) der Strahlung in einem Oberflächenpunkt x (siehe Abb. 2.7).

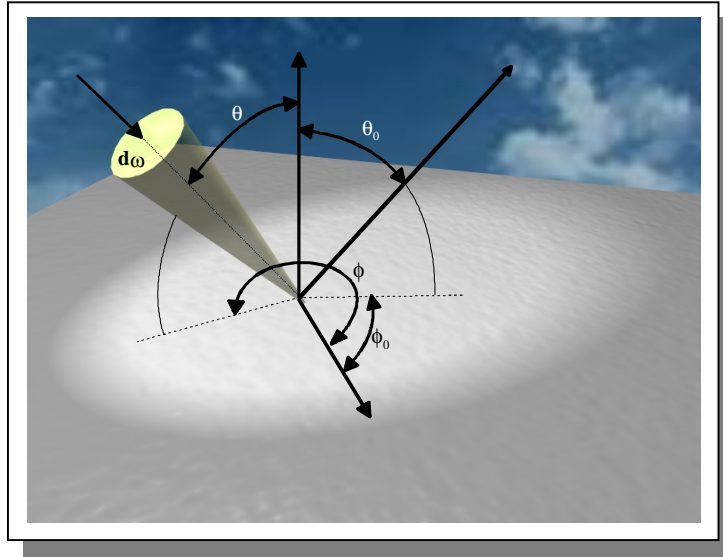


Abb. 2.7: Bezeichnungen für die BRVF.

Es ist zu beachten, daß die Richtung der Strahlung hier durch Polarkoordinaten und somit durch zwei Winkel beschrieben wird. Die BRVF wird durch ρ_{bd} bezeichnet und es gilt:

$$\rho_{bd}(\theta_0, \phi_0, \theta, \phi) = \frac{L(x, \theta_0, \phi_0)}{d\Phi_i} = \frac{L(x, \theta_0, \phi_0)}{L_i(x, \theta, \phi) \cos \theta d\omega} \quad (2.14)$$

Ideal spiegelnde und ideal diffus reflektierende Oberflächen besitzen die einfachsten bidirektionalen Reflexions-Verteilungsfunktionen. Im Fall einer ideal diffus reflektierenden Oberfläche gibt es aufgrund des Lambertschen Cosinusesgesetzes in der BRVF keine Abhängigkeiten von der Reflexionsrichtung. Weiterhin gilt aufgrund der Gesetzmäßigkeiten der Thermodynamik die Eigenschaft der Umkehrbarkeit der BRVF [SH81], d.h.

$$\rho_{bd}(\theta_0, \phi_0, \theta, \phi) = \rho_{bd}(\theta, \phi, \theta_0, \phi_0) \quad (2.15)$$

Somit ist die BRVF von Lambert-Reflektoren (ideal diffuse Reflexion) weder von der Reflexionsrichtung noch von der Einfallrichtung abhängig und es gilt für die Reflektivität ρ der Oberfläche

$$\rho_{bd}(\theta_0, \phi_0, \theta, \phi) \equiv \rho \quad (2.16)$$

Als diffuse Reflektivität ρ_d wird der Anteil der aus einer bestimmten Richtung einfallenden Leuchtdichte bezeichnet, der in beliebige Richtungen in die Hemisphäre oberhalb der Oberfläche reflektiert wird. Dieser Wert ergibt sich durch Integration der BRVF (und somit Gleichung (2.16) über alle Richtungen der Hemisphäre oberhalb des Oberflächenpunktes. Es ist:

$$\rho_d = \rho \int_{\Omega} \cos \theta_0 d\omega_0 = \pi \rho \quad (2.17)$$

Wenn somit eine Umgebungsbeschreibung nur aus ideal diffusen Reflektoren besteht, ist es nicht notwendig, den Austausch von Lichtenergie zwischen Oberflächen mit Hilfe der Leuchtdichte zu beschreiben. In diesem Fall ist der Energieaustausch nicht abhängig von der

Einfall- oder Reflexionsrichtung der Strahlung. Somit ist es naheliegend, die Energiemengen mit Hilfe der spezifischen Ausstrahlung einer Oberfläche zu quantifizieren, wobei die Reflexion durch einen konstanten Reflexionsfaktor ρ_d beschrieben werden kann.

Es ist :

$$\begin{aligned} L(x, \theta_0, \phi_0) &= \int_{\Omega} \rho_{bd}(\theta_0, \phi_0, \theta, \phi) L_i(x, \theta, \phi) \cos \theta \, d\omega \\ &= \frac{\rho_d}{\pi} \int_{\Omega} L_i(x, \theta, \phi) \cos \theta \, d\omega \\ &= \frac{\rho_d}{\pi} E(x) \end{aligned} \quad (2.18)$$

Hierbei ist E die Bestrahlungsstärke, die auf der Oberfläche ankommt (siehe Abschnitt 2.1.1). Durch Umformen von Gleichung (2.13) und Einsetzen in Gleichung (2.18) ergibt sich die für die im weiteren Verlauf dieser Arbeit betrachtete Radiosity-Methode wichtige (richtungsunabhängige) Beziehung zwischen Bestrahlungsstärke und spezifischer Ausstrahlung einer ideal diffus reflektierenden Oberfläche mit

$$B(x) = \rho_d E(x) \Leftrightarrow \rho_d = \frac{B(x)}{E(x)} \quad (2.19)$$

Für weitere Ausführungen zum Thema der BRVF sei auf [CW93] und [SP94] verwiesen.

2.1.3.2 Klassifizierung von Lichttransportpfaden

Ziel globaler Beleuchtungsverfahren ist die Simulation von *Lichttransportpfaden* (siehe Abb. 2.8). Diese sind alle möglichen Wege, die ein Photon nach dessen Emission durch eine Lichtquelle innerhalb einer Umgebung zurücklegen kann, bis es von einer Oberfläche absorbiert wird oder auf die Retina eines Betrachters innerhalb der Szene trifft. Hierbei wird davon ausgegangen, daß die Objekte der Umgebung das Licht diffus und spiegelnd reflektieren bzw. transmittieren.

Lichttransportpfade lassen sich durch reguläre Ausdrücke über dem Alphabet $\Sigma = \{L, E, S, D\}$ beschreiben. Hierbei bezeichnet L eine Lichtquelle der Szene (Ausgangspunkt der Strahlung) und E die Position des Betrachters. Der Buchstabe D bezeichnet eine Richtungsänderung des Photons, die durch diffuse Reflexion (bzw. Refraktion) verursacht wird, während der Buchstabe S eine durch spiegelnde Reflexion verursachte Richtungsänderung anzeigt. Diese regulären Ausdrücke ermöglichen eine Bewertung der Methoden zur Approximation der globalen Beleuchtung innerhalb einer Umgebung hinsichtlich der Komplexität der jeweils simulierbaren Lichttransportpfade. Ein Verfahren, welches das Beleuchtungsproblem vollständig löst, muß somit alle Lichttransportpfade der Menge $L(D/S)^*E$ simulieren.

Die zur Zeit am häufigsten verwendeten Methoden zur Approximation der Beleuchtungssituation simulieren jeweils nur Teilmengen aller möglichen Transportpfade, wie im weiteren Verlauf der Arbeit noch gezeigt wird. Dieses hat zur Folge, daß wesentliche Beleuchtungseffekte bei der Bildgenerierung nicht berücksichtigt werden.

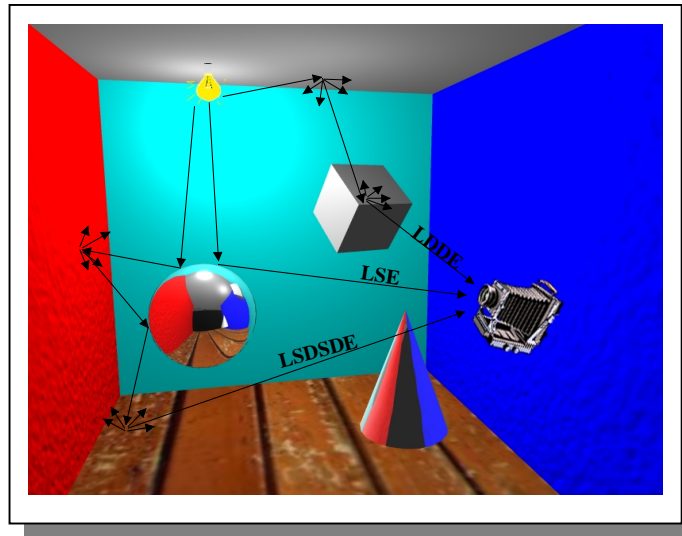


Abb. 2.8: Lichttransportpfade von der Lichtquelle zum Betrachter.

2.2 Lokale Beleuchtungsmodelle

Lokale Beleuchtungsmodelle wurden zu Anfang der siebziger Jahre entwickelt. Sie beschreiben die durch einen Betrachter wahrnehmbare Helligkeit von sichtbaren Oberflächenpunkten lediglich in Abhängigkeit von den Eigenschaften der direkt sichtbaren Flächen selbst (Farbe, Winkel zur Lichtquelle, etc.) und den Eigenschaften der primären Lichtquellen in der Szene. Die Geometrie und das Material der übrigen Objekte der Umgebung werden bei der Bestimmung von Helligkeitswerten an sichtbaren Oberflächenpunkten nicht betrachtet. Dieses hat zur Folge, daß keine Schatten und auch keine indirekten Beleuchtungseffekte simuliert werden. Der Vorteil von lokalen Beleuchtungsmodelle ist deren Einfachheit, welche eine Realisierung in Hardware ermöglicht. Es ist zu beachten, daß diese Modelle nicht physikalisch basiert sind. Somit geben die auf Basis von lokalen Beleuchtungsmodellen an sichtbaren Oberflächenpunkten berechneten Helligkeitswerte keine radiometrischen Größen (siehe Abschnitt 2.1.1) an. In der Literatur werden die berechneten Werte oftmals als *Intensität I* eines sichtbaren Punktes bezeichnet. Hierbei darf die Intensität I jedoch nicht mit der in 2.1.1 eingeführten Strahlungsstärke I verwechselt werden. Um mit der Literatur konform zu sein, wird im weiteren Verlauf dieses Abschnittes durch I ein Helligkeitswert (oder Pixel-Intensität) eines sichtbaren Oberflächenpunktes beschrieben und somit nicht die radiometrische Größe der Strahlungsleistung. Dieser Helligkeitswert wird zum Zwecke der Bildgenerierung auf Basis eines geeigneten Farbmodells in eine Pixelfarbe umgewandelt (siehe Abschnitt 2.4). Erst im Zusammenhang mit den physikalisch korrekten globalen Beleuchtungsmodellen wird durch I wieder die radiometrische Größe der Strahlungsstärke bezeichnet.

Lokale Beleuchtungsmodelle sind empirische Modelle und basieren auf einer einfachen Approximation der bidirektionalen Reflexions-Verteilungsfunktion, wobei nur die ideal diffuse, ideal spiegelnde und grob spiegelnde Reflexion des von Lichtquellen abgegebenen Lichtes modelliert ist. Es ist zu beachten, daß keine Wechselwirkungen zwischen Objekten der Szene berücksichtigt werden (d.h. Energietransfer zwischen Oberflächen verursacht durch Reflexion).

Die *ideale diffuse Reflexion* ist charakteristisch für matte, rauhe Oberflächen. Diffus reflektierende Oberflächen erscheinen aus allen Blickwinkeln gleich hell, da die Strahlungsdichte für jeden Betrachtungswinkel konstant ist (siehe Gleichung (2.9)). Die Interaktionen der Lichtstrahlen mit dem Oberflächenmaterial bestimmen die diffuse Reflexion, da bestimmte Wellenlängen des Spektrums des einfallenden Lichtes von dem Oberflächenmaterial absorbiert werden, während es andere Wellenlängen reflektiert. Der Anteil der reflektierten Lichtenergie pro Flächeneinheit ist proportional zum Kosinus des Winkels ϕ zwischen der Richtung \vec{L} zur Lichtquelle und der Oberflächennormalen \vec{N} an diesem Punkt. Diese fundamentale Beschreibung der diffusen Reflexion auf Basis des Kosinusetzes von Lambert läßt sich durch die folgende Gleichung beschreiben:

$$I_d = I_i K_d \cos \phi \quad (2.20)$$

Hierbei ist

I_i die Intensität der Lichtquelle,

K_d der materialabhängige konstante diffuse Reflexionsgrad der Oberfläche
($0 \leq K_d \leq 1$),

ϕ der Winkel zwischen der Richtung \vec{L} des einfallenden Lichtstrahls und der Oberflächennormalen \vec{N} .

Gleichung (2.20) kann mit Hilfe des Skalarproduktes umgeschrieben werden zu

$$I_d = I_i K_d \vec{L} \vec{N} \quad (2.21)$$

wobei \vec{L} ein Einheitsvektor in Richtung der Lichtquelle und \vec{N} ein Einheits-Normalenvektor der reflektierenden Oberfläche ist.

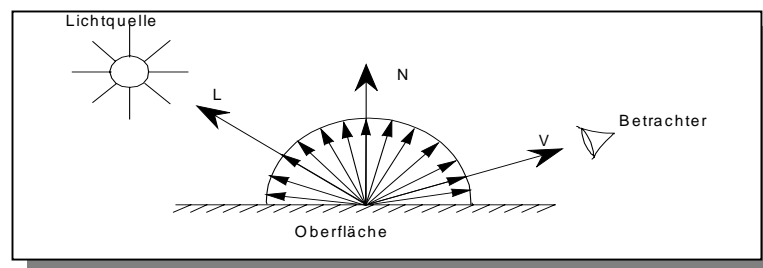


Abb. 2.9: Ideal diffuse Reflexion des Lichtes

Das wesentliche Merkmal einer diffus reflektierenden Oberfläche ist, daß die Helligkeit des reflektierten Lichtes für alle Betrachterstandpunkte gleich erscheint (siehe Abb. 2.9).

Neben diffus reflektierenden Oberflächen müssen bei der Bildgenerierung auch spiegelnde Oberflächen berücksichtigt werden. Spiegelnde Reflexionen sind, im Gegensatz zu diffusen Reflexionen, hochgradig abhängig von dem Betrachterstandpunkt. Es muß zwischen *ideal spiegelnden* und *grob spiegelnden* Oberflächen unterschieden werden, wobei ein perfekter Spiegel als ideal spiegelnde Oberfläche bezeichnet wird. Für einen perfekten Spiegel ist charakteristisch, daß ein einfallender Lichtstrahl \vec{L} in genau einer Spiegelrichtung \vec{R} reflektiert

wird, wobei sich \vec{R} aus der Achsenspiegelung von \vec{L} an dem Normalenvektor \vec{N} der Oberfläche ergibt (siehe Abb. 2.10).

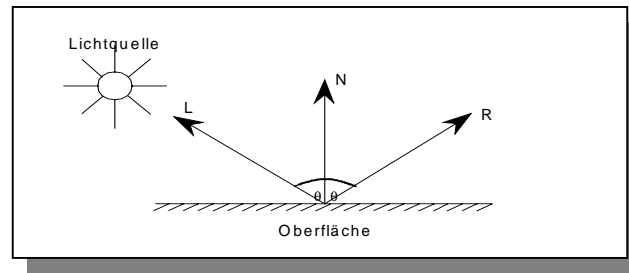


Abb. 2.10: Ideal spiegelnde Reflexion des Lichtes.

Somit wird ein reflektierter Lichtstrahl von einem Betrachter nur dann wahrgenommen, wenn er sich in Spiegelrichtung \vec{R} befindet. In der realen Welt gibt es nur wenige Oberflächen, welche die Reflexionseigenschaften eines perfekten Spiegels aufweisen. Bei *grob spiegelnden* Oberflächen wird das reflektierte Licht in einem kegelförmigen Bereich um die Spiegelrichtung gestreut und kann somit von einem Betrachter wahrgenommen werden, dessen Blickrichtung in diesem Bereich liegt (siehe Abb. 2.11).

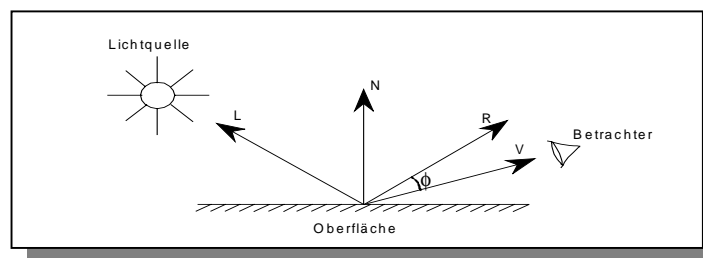


Abb. 2.11: Betrachtungsrichtung weicht um Winkel ϕ von Reflexionsrichtung R ab.

Von Phong wurde ein empirisches Modell für die spiegelnde Reflexion vorgeschlagen, in welchem die Intensität des spiegelnd reflektierten Lichtes proportional zu $\cos^n \phi$ ist. Hierbei ist ϕ der Winkel zwischen der Spiegelrichtung \vec{R} und dem Vektor \vec{V} in Richtung des Betrachters. Durch Verändern des Exponenten n kann die Größe des vom Betrachter wahrgenommenen Glanzlichtes (*Highlights*) verändert werden, das durch die spiegelnde Reflexion auf der Oberfläche hervorgerufen wird. Für $n \rightarrow \infty$ ergibt sich eine Annäherung an die Reflexionseigenschaften eines perfekten Spiegels (siehe Abb. 2.12).

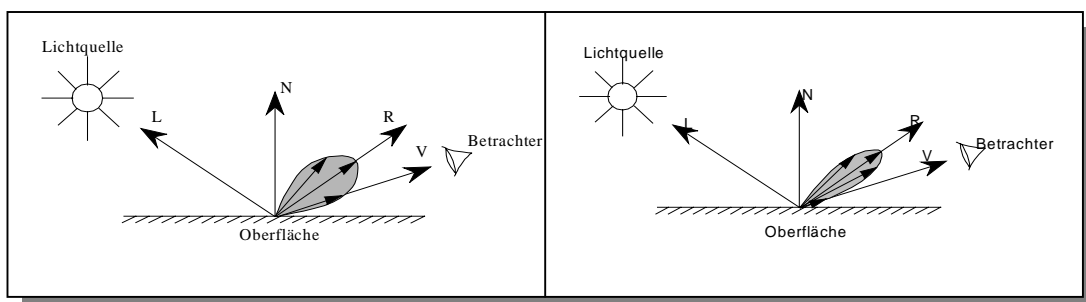


Abb. 2.12: Grob spiegelnde Reflexion mit großem n (links) und kleinem n (rechts).

Die Intensitätsgleichung für die spiegelnde Reflexion lautet somit

$$I_s = I_i K_s \cos^n \phi, \quad (2.22)$$

wobei

- I_i die Intensität der Lichtquelle;
- K_s der spiegelnde, materialabhängige Reflexionskoeffizient der Oberfläche ($0 \leq K_s \leq 1$)
- ϕ der Winkel zwischen der Spiegelrichtung \vec{R} der einfallenden Lichtenergie und dem Vektor \vec{V} der in Richtung Betrachter zeigt und
- n der spiegelnde Exponent, welcher vom Material abhängig ist.

Licht, das aus allen Richtungen der Hemisphäre einer Oberfläche einfällt, wird als *ambientes Licht* bezeichnet. Dieses ambiente Licht entsteht durch mehrfache Reflexionen der Lichtstrahlen zwischen den Objekten einer Umgebung und wird von lokalen Beleuchtungsmodellen durch den ambienten Term $I_a K_a$ beschrieben. Die ambiente Intensität I_a wird als konstant für alle Objekte betrachtet. Der Anteil des ambienten Lichtes, der von einem Objekt reflektiert wird, ist durch einen materialabhängigen ambienten Reflexionskoeffizienten K_a bestimmt.

Die zuvor beschriebenen Reflexionsmodelle wurden von Phong [BT75] zu einem grundlegenden Beleuchtungsmodell der Computergrafik zusammengefaßt. Bei diesem lokalen Beleuchtungsmodell setzt sich die Intensität eines Oberflächenpunktes aus einer Linearkombination des ambienten, des diffus reflektierten und des spiegelnd reflektierten Lichtes einer Lichtquelle wie folgt zusammen:

$$I = I_a K_a + I_i (K_d (\vec{L} \cdot \vec{N}) + K_s \cos^n \phi) \quad (2.23)$$

Sind in einer Szene N Lichtquellen vorhanden, wird die Intensität eines sichtbaren Oberflächenpunktes auf Basis der Gleichung (2.24) bestimmt.

$$I_{total} = I_a K_a + \sum_{i=1}^N (I_{d,i} + I_{s,i}) \quad (2.24)$$

Hierbei wird durch $I_{d,i}$ (bzw. $I_{s,i}$) der Intensitätsanteil des Oberflächenpunktes bezeichnet, welcher durch ideal diffuse Reflexion (bzw. spiegelnde Reflexion) der auf den Punkt aus Richtung der Lichtquelle L_i einfallenden Lichtenergie verursacht wird. Diese Anteile werden gemäß der Formeln (2.21) und (2.20) für jeder der in der Szene definierten Lichtquellen ausgewertet.

Dieses lokale Beleuchtungsmodell ist durch die folgenden Punkte charakterisiert:

- Es werden nur Punktlichtquellen berücksichtigt, wobei keine speziellen Verteilungen der Strahlendichte für Lichtquellen spezifiziert werden können.
- Bei Berechnung einer Intensität für einen sichtbaren Oberflächenpunkt werden lediglich die lokalen Eigenschaften des entsprechenden Objektes und die Eigenschaften der in der Szene enthaltenen Lichtquellen betrachtet.

- Der Abstand der Lichtquelle und des Betrachterstandpunktes zu den Oberflächenpunkten geht nicht in die Berechnung der Intensitäten ein.
- Anteile der Intensität eines Oberflächenpunktes, die durch diffuse und spiegelnde Reflexion verursacht werden, berücksichtigen nur Licht, welches von primären Lichtquellen an den entsprechenden Oberflächenpunkt abgegeben und dort reflektiert wird.
- Ein empirisches Modell wird benutzt, um die unterschiedlichen Intensitäten des spiegelnd reflektierten Lichtes in der Umgebung der Reflexionsrichtung zu berechnen und somit den Glanz von Oberflächen zu modellieren.
- Das Problem der Schattenbildung wird nicht betrachtet.
- Der globalen Beleuchtung (ambientes Licht) wird durch einen konstanten Wert Rechnung getragen.

Verbesserte lokale Beleuchtungsmodelle wurden von Blinn [Blinn77] und Cook [CT81] eingeführt. Keines dieser verbesserten lokalen Modelle ist jedoch ausreichend, um den Qualitätsanforderungen der photorealistischen Bildgenerierung aus der heutigen Sicht gerecht zu werden, da die wesentlichen durch indirekte Beleuchtung entstehenden Beleuchtungsphänomene nicht berücksichtigt werden. Im allgemeinen lassen sich mit Hilfe von lokalen Beleuchtungsmodellen lediglich Lichttransportpfade aus der Menge $L(D/S)E$ beschreiben (siehe Abschnitt 2.1.3.2).

2.3 Lokale Beleuchtungsalgorithmen

Die Beschreibungen von Modellen bzw. Szenen liegen üblicherweise in der Form von polygonalen Netzen vor. Die Interpolations-Schattierungsverfahren ermitteln mit Hilfe von Beleuchtungsmodellen die Intensitäten der Eckpunkte des polygonalen Netzes. Diese Werte werden interpoliert, um die Intensitäten für Punkte im Inneren der Polygone zu approximieren. Die Interpolationswerte werden im Rahmen eines Scan-Line-Verfahrens ermittelt.

2.3.1 Gouraud-Schattierungsalgorithmus

Die Gouraud-Schattierung [Gou71] basiert auf dem Prinzip der bilinearen Interpolation von Intensitäten, welche für die Eckpunkte eines polygonalen Netzes berechnet werden.

Mit Hilfe eines Beleuchtungsmodells werden die Eckpunkt-Intensitäten berechnet. Die Intensitäten an den Schnittpunkten einer Scan-Line mit den Kanten eines Polygons können mittels linearer Interpolation aus den Intensitäten der Eckpunkte des Polygons berechnet werden. Diese Interpolationswerte werden benutzt, um die Intensitäten der Punkte auf einer Scan-Line zu ermitteln (siehe Abb. 2.13).

Der Gouraud-Schattierungsalgorithmus eignet sich vor allem zur Schattierungsberechnung von diffus reflektierenden Oberflächen. Ein Glanzlicht (Highlight), welches durch spiegelnde Reflexionen entsteht, wird bei einer Gouraud-Schattierung nur dann wiedergegeben, wenn es auf einen Knotenpunkt des polygonalen Netzes fällt, was im allgemeinen nicht der Fall ist.

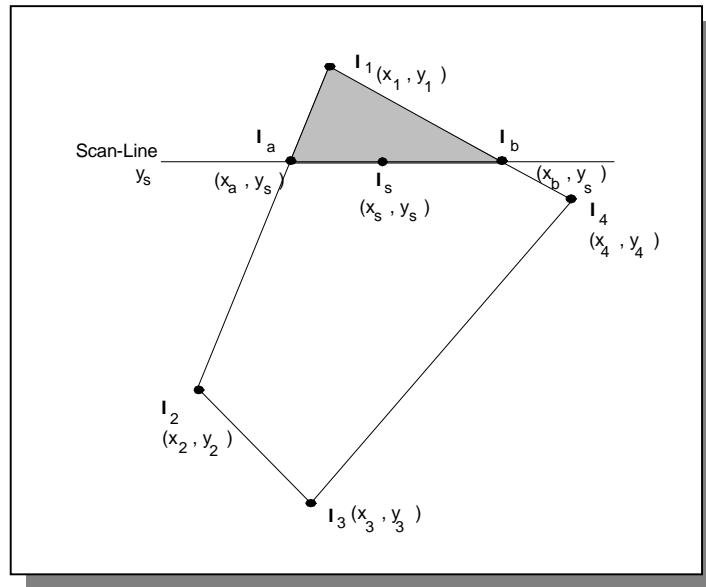


Abb. 2.13: Das Gouraud-Schattierungsverfahren.

Die Intensitäten an Punkten auf Polygonkanten und im Inneren eines Polygons werden mit Hilfe der folgenden Interpolationsgleichungen bestimmt:

$$\begin{aligned}
 I_a &= \frac{1}{y_1 - y_2} (I_1(y_s - y_2) + I_2(y_1 - y_s)) \\
 I_b &= \frac{1}{y_1 - y_4} (I_1(y_s - y_4) + I_4(y_1 - y_s)) \\
 I_s &= \frac{1}{x_b - x_a} (I_a(x_b - x_s) + I_b(x_s - x_a))
 \end{aligned} \tag{2.25}$$

Es ist zu beachten, daß das Gouraud-Schattierungsverfahren nicht nur in Kombination mit lokalen Beleuchtungsmodellen verwendet wird. Es kann auch zusammen mit einem globalen Beleuchtungsverfahren wie der Radiosity-Methode benutzt werden. Hierbei werden die während der Beleuchtungssimulation an den Eckpunkten des Polygon-Netzes unabhängig von einem Betrachterstandpunkt ermittelten Beleuchtungsinformationen für die Oberflächenschattierung (*Shading*) bei der Bildgenerierung verwendet.

2.3.2 Phong-Schattierungsalgorithmus

Die Phong-Schattierung [BT75] basiert ebenfalls auf einer bilinearen Interpolation, um die Intensitäten im Inneren eines Polygons zu berechnen. Bei diesem Verfahren werden jedoch nicht die Intensitäten der Eckpunkte des polygonalen Netzes interpoliert, sondern die Eckpunktnormalen (siehe Abb. 2.14). Die interpolierten Normalen der inneren Flächenpunkte werden dazu benutzt, um mit dem Beleuchtungsmodell von Phong einen Intensitätswert für jeden dieser Punkte direkt zu bestimmen.

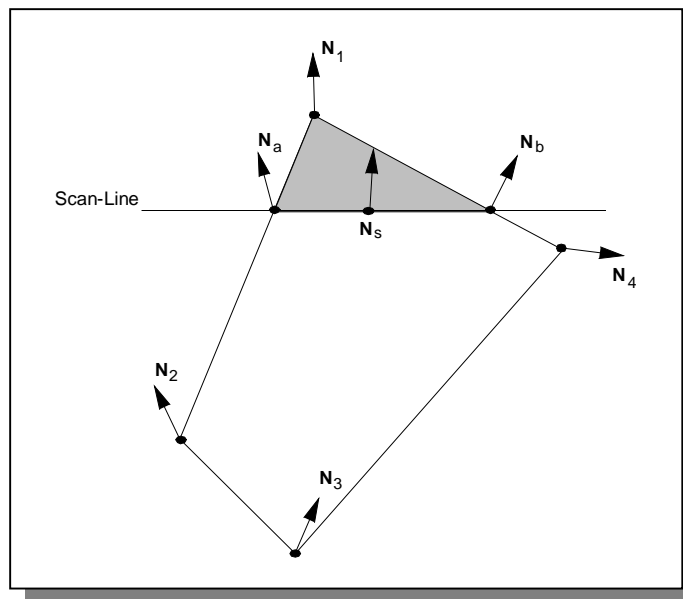


Abb. 2.14: Interpolation von Normalen im Inneren eines Patches.

Die Normalen im Flächeninneren können mit folgenden Gleichungen bestimmt werden:

$$\begin{aligned}
 N_a &= \frac{1}{y_1 - y_2} (N_1(y_s - y_2) + N_2(y_1 - y_s)) \\
 N_b &= \frac{1}{y_1 - y_4} (N_1(y_s - y_4) + N_4(y_1 - y_s)) \\
 N_s &= \frac{1}{x_b - x_a} (N_a(x_b - x_s) + N_b(x_s - x_a))
 \end{aligned}
 \tag{2.27}$$

Die Phong-Schattierung generiert Glanzlichter, im Gegensatz zur Gouraud-Schattierung, auch im Inneren der Polygonen, wodurch die höhere Qualität der erzeugten Bilder deutlich sichtbar ist. Da bei diesem Schattierungsverfahren für jeden inneren Punkt eine Intensitätsberechnung mittels eines Beleuchtungsmodells durchgeführt wird, ist der Rechenaufwand wesentlich größer als bei der Gouraud-Schattierung.

2.4 Farben

Um ein photorealistisches Bild auf einem zweidimensional gerasterten Ausgabemedium darzustellen, muß für jeden Rasterpunkt ein Farbwert berechnet werden. Das menschliche Auge ist nur für einen Teilbereich der optischen Strahlung empfindlich. Dieser Bereich liegt zwischen 380 nm (Ultraviolett) bis 770 nm (Infrarot) und wird als Licht bezeichnet. Leider ist es bei den derzeit verfügbaren Ausgabemedien wie Monitore oder Drucker nicht möglich, die mit Hilfe von Beleuchtungsmodellen ermittelten Intensitäten von Oberflächenpunkten (bzw. Helligkeitsinformationen) direkt wiederzugeben. Ein Bildpunkt (Pixel) wird üblicherweise durch seine Farbe charakterisiert. Um Farben auf einem Ausgabemedium darzustellen, werden Farbmodelle verwendet, die sich auf die Berechnung von Werten für bestimmte Wellenlängen des Farbspektrums beschränken. Das bekannteste dieser Modelle ist das *RGB-Modell*, welches auf der Grundlage der additiven Farbmischung basiert. Die darstellbaren Farben wer-

den hierbei aus den drei Grundfarben Rot, Grün und Blau zusammengesetzt. Eine farbliche Darstellung einer gegebenen Szene kann generiert werden, indem ein Schattierungsverfahren die Intensitäten der sichtbaren Oberflächenpunkte für die Wellenlängen der Farben Rot, Grün und Blau bestimmt. Das RGB-Modell wird anschließend benutzt, um aus den Intensitäten die entsprechenden Rasterpunktfarben zu berechnen. Es werden somit drei Intensitätsgleichungen gelöst, um eine darstellbare Farbinformation (Pixelfarbe) zu erhalten.

$$\begin{aligned} I_r &= I_a K_{ar} + I_i \left(K_{dr} (\vec{L} \cdot \vec{N}) + K_s \cos^n \phi \right) \\ I_g &= I_a K_{ag} + I_i \left(K_{gr} (\vec{L} \cdot \vec{N}) + K_s \cos^n \phi \right) \\ I_b &= I_a K_{ab} + I_i \left(K_{db} (\vec{L} \cdot \vec{N}) + K_s \cos^n \phi \right) \end{aligned} \quad (2.27)$$

Die Farbe der Objekte wird durch ihren diffusen Reflexionskoeffizienten festgelegt. Es ist zu beachten, daß der spiegelnde Reflexionskoeffizient nicht abhängig von der Objektfarbe ist, da durch diese Reflexion Glanzlichter in der Farbe der Lichtquelle (weiß) entstehen sollen.

Diese Vorgehensweise der Farberzeugung ist nicht auf die lokalen Beleuchtungsmodelle beschränkt, sondern wird auch bei den globalen Beleuchtungsmodellen eingesetzt. In diesem Fall werden die physikalisch korrekten Strahlungsdichten an sichtbaren Oberflächenpunkten für die relevanten Frequenzbereiche bestimmt und mit Hilfe eines Farbmodells zu einer darstellbaren Farbe kombiniert.

Es ist zu beachten, daß in der Vergangenheit eine Vielzahl von verschiedenen Farbmodellen entwickelt wurde, die ein breites Spektrum von Anwendungsszenarien abdecken. Eine ausführliche Betrachtung der Farbmatrik sowie verschiedener Farbmodelle ist in [Enc90] zu finden.

2.5 Ein Modell für die globale Beleuchtungsberechnung

Im Gegensatz zu lokalen Beleuchtungsmodellen beziehen *globale Beleuchtungsmodelle* bei der Berechnung der Beleuchtung einer Fläche jeweils den Strahlungsaustausch zwischen allen Oberflächen einer Szene in Betracht. Diese Methoden basieren auf einer globalen Strahlungsdichtefunktion, welche die Lichtausbreitung in einer Umgebung beschreibt. Diese Funktion besitzt eine wichtige Eigenschaft, die explizit oder implizit in vielen Algorithmen der Computergrafik verwendet wird. Diese Eigenschaft besagt, daß die Strahlungsdichte entlang einer Geraden in Blickrichtung konstant ist. Das bedeutet, daß eine Änderung des Betrachterstandpunktes entlang einer solchen Geraden bei konstanter Blickrichtung keine Auswirkung auf die beobachtete Strahlungsdichte hat. Hieraus folgt, daß die globale Strahlungsdichtefunktion durch die Funktionswerte aller Punkte auf den Oberflächen einer Szene vollständig spezifiziert ist.

Die Strahlungsdichte L an einem Punkt x auf einer Oberfläche setzt sich aus einem von der Oberfläche selbst emittierten Anteil L_e und einem reflektierten Anteil L_r zusammen:

$$L(x, \theta_0, \phi_0) = L_e(x, \theta_0, \phi_0) + L_r(x, \theta_0, \phi_0) \quad (2.28)$$

Der selbst emittierte Anteil L_e ist bekannt und muß somit nicht bestimmt werden. Das Hauptproblem stellt die Berechnung von L_r dar. Da die gesamte reflektierte Strahlung aus der Hemisphäre oberhalb des Punktes x einfällt, läßt sich L_r mit Hilfe der bidirektionalen Reflexionsverteilungsfunktion darstellen. Für den Fall, daß lediglich aus der Einfallrichtung (θ, ϕ) eine

von Null abweichende Strahlungsdichte auf den Oberflächenpunkt x einfällt, folgt aus Gleichung (2.14):

$$L_r(x, \theta_0, \phi_0) = \rho_{bd}(\theta_0, \phi_0, \theta, \phi) L_i(x, \theta, \phi) \cos \theta \, d\omega \quad (2.29)$$

In einer realen Umgebung existiert jedoch ein Vielzahl von Einfallsrichtungen, aus denen Strahlung auf x einfällt. Somit weicht die entsprechende einfallende Strahlungsdichte L_i aus diesen Richtungen von Null ab. Aus diesem Grund muß über alle Richtungen Ω der Hemisphäre oberhalb des Punktes x integriert werden, um die reflektierte Strahlungsdichte L_r vollständig zu beschreiben:

$$L(x, \theta_0, \phi_0) = L_e(x, \theta_0, \phi_0) + \int_{\Omega} \rho_{bd}(x, \theta_0, \phi_0, \theta, \phi) L_i(x, \theta, \phi) \cos \theta \, d\omega \quad (2.30)$$

Sei nun x' der Oberflächenpunkt, von dem Strahlung abgegeben wird, welche auf den Oberflächenpunkt x aus Richtung (θ, ϕ) einfällt. In dem lokalen Koordinatensystem des Punktes x' sind die Polarkoordinaten dieser Strahlrichtung (θ', ϕ') (siehe Abb. 2.15).

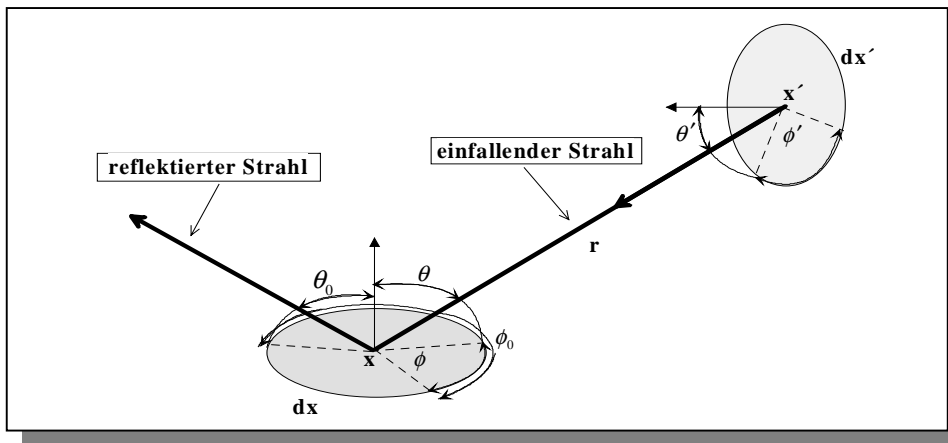


Abb. 2.15: Geometrie des Strahlungsaustausches zwischen zwei Oberflächenpunkten.

Innerhalb eines Vakuums nimmt die Strahlungsdichte nicht mit der Entfernung ab. In diesem Fall stimmt die auf x aus Richtung (θ, ϕ) auftreffende Strahlungsdichte mit der von Punkt x' in Richtung (θ', ϕ') abgegebenen Strahlungsdichte überein, d.h.

$$L_i(x, \theta, \phi) = L(x', \theta', \phi'). \quad (2.31)$$

Nun läßt sich das Integral, welches die Richtungen der Hemisphäre oberhalb eines Oberflächenpunktes als Integrationsbereich hat, in ein Flächenintegral überführen. Hierbei umfaßt der Integrationsbereich alle Punkte der Oberflächen einer gegebenen Umgebung. Der Raumwinkel, den ein Oberflächenstück dx' mit einer Entfernung r zu einem Oberflächenpunkt x und einem Winkel von θ' zur Oberflächennormalen am Punkt x' in der Hemisphäre oberhalb von x einnimmt, entspricht der projizierten Fläche auf einer Kugel um x mit Radius r geteilt durch r^2 [CW93]. Somit gilt:

$$d\omega = \frac{dx' \cos \theta'}{r^2} \quad (2.32)$$

Mit Hilfe von Gleichung (2.32) läßt sich das Richtungsintegral der Gleichung (2.30) in ein Flächenintegral überführen:

$$L(x, \theta_0, \phi_0) = L_e(x, \theta_0, \phi_0) + \int_{\Gamma} \rho_{bd}(x, \theta_0, \phi_0, \theta, \phi) L(x', \theta', \phi') \frac{\cos \theta \cos \theta'}{r^2} v dx' \quad (2.33)$$

Der Term v stellt einen sogenannten Sichtbarkeitsterm dar, der den Wert Eins annimmt, wenn die Oberflächenpunkte x und x' gegenseitig sichtbar sind (andernfalls nimmt v den Wert Null an). Der Integrationsbereich Γ umfaßt die Menge sämtlicher in der Umgebung enthaltenen Oberflächen. Die einzige unbekannte Größe in Gleichung (2.33) ist die Strahlungsdichte, die von einem Oberflächenpunkt x' in Richtung des Punktes x abgegeben wird. Die übrigen Variablen sind Funktionen von x und x' . Somit liegt eine *Freedholmsche Integralgleichung der zweiten Art* vor. Die Finite Elemente Methode und die Monte Carlo Integration stellen numerische Lösungsverfahren dar, welche die Lösung von Integralgleichungen dieser Art ermöglichen.

Die Gleichung (2.33) entspricht der von Kajjiya 1986 eingeführten *globalen Beleuchtungsgleichung (Rendering Equation)* und beschreibt ein Energiegleichgewicht in einer Umgebung [Kaj86]. Die globale Beleuchtungsgleichung läßt sich deutlich vereinfachen, wenn die Oberflächeneigenschaften der Objekte einer Szene folgende zusätzlichen Bedingungen erfüllen: sie sind nicht transparent (d.h. kein Licht transmittieren) und besitzen ideal diffuse (lambertsche) Eigenschaften bzgl. Reflexion und Emission. Wie in Abschnitt 2.1 schon gezeigt wurde, ist in diesem Fall die Strahlungsdichte und die bidirektionale Reflexion-Verteilungsfunktion lediglich eine Funktion der Position. Die Richtung der Strahlungsenergie spielt in diesem Fall keine Rolle mehr. Mit Hilfe der Gleichungen (2.12), (2.16) und (2.17) ergibt sich für die globale Beleuchtungsgleichung in einer ideal diffus reflektierenden (und emittierende) Umgebung:

$$B(x) = E(x) + \rho_d(x) \int_{\Gamma} \frac{\cos \theta \cos \theta'}{\pi r^2} v B(x') dx' \quad (2.34)$$

In Gleichung (2.34) wird mit $E(x)$ die ideal diffuse Eigenemission von Strahlungsenergie an Oberflächenpunkt x bezeichnet. Analog zur spezifischen Ausstrahlung $B(x)$ gilt hierbei $E(x) = \pi L_d(x)$.

2.6 Abschließende Bemerkungen

In diesem Kapitel wurden die grundlegenden Konzepte der physikalisch basierten Simulation der globalen Beleuchtung in der Computergrafik erläutert. Hierbei wurde insbesondere der physikalische Hintergrund aus radiometrischer und photometrischer Sicht betrachtet, sowie die wesentlichen Gleichungen zur Beschreibung des Strahlungsaustausches zwischen Oberflächen beschrieben. Die globale Beleuchtungsgleichung für Szenen mit ideal diffusen Reflexionseigenschaften (Gleichung (2.34)) bildet die Grundlage für die im weiteren Verlauf der Arbeit betrachtete Radiosity-Methode. Globale Beleuchtungsmodelle ermöglichen theoretisch die Simulation von Lichttransportpfaden der Art $L(D/S)*E$. Somit können komplexe Beleuchtungsphänomene berücksichtigt werden, wodurch der Realitätsgrad der generierten

Bilder im Vergleich zu Bildern, die mit Hilfe von lokalen Beleuchtungsmodellen erzeugt wurden, dramatisch gesteigert wird.

Kapitel 3

Die Radiosity-Methode in statischen Szenen

Die Radiosity-Methode wurde ursprünglich in der Thermodynamik zur Simulation des Austausches von Wärmeenergie zwischen den Oberflächen einer geschlossenen Umgebung entwickelt [Hot54][ED59][Spar63][Spar78][SH81]. In der Computergrafik ist dieses Verfahren erstmalig 1984 von Goral, Torrance und Greenberg [GTGB84] verwendet worden, um globale Beleuchtungsphänomene, welche durch die ideal diffuse Emission und Reflexionen des Lichtes entstehen, möglichst exakt und vollständig zu simulieren.

Die Radiosity-Methode bietet die Möglichkeit, eine Simulation der Verteilung der Lichtenergie in einer Szene, die ausschließlich Objekte mit lambertschen Eigenschaften (siehe Abschnitt 2.1.1) enthält, physikalisch korrekt durchzuführen. Hierbei wird der Austausch von Strahlungsenergie (Licht) zwischen den Objektoberflächen nachgebildet. Das globale Beleuchtungsmodell, welches Grundlage dieses Verfahrens ist, beruht auf dem Prinzip der Energieerhaltung. In diesem Beleuchtungsmodell kann jede Oberfläche Lichtenergie aufgrund von Emission oder Reflexion an die Umgebung abgeben, wodurch indirekte Beleuchtung und Farbübertragungen (*Color-Bleeding*) zwischen Oberflächen berücksichtigt werden. Zwischen der zugeführten Strahlungsenergie durch Lichtquellen und der Absorption durch die Oberflächen besteht ein Gleichgewicht, das zu einer von der Zeit unabhängigen Beleuchtungsstärke in der Szene führt. Im Anschluß an die Beleuchtungssimulation werden Bilder auf der Basis der mit Hilfe des Radiosity-Verfahrens berechneten spezifischen Ausstrahlung (siehe Kapitel 2) der Objektoberflächen generiert.

Ein wesentlicher Vorteil der Radiosity-Methode gegenüber anderen Methoden zur Simulation der globalen Beleuchtung ist, daß die Berechnungen unabhängig von einem Betrachterstandpunkt in der Szene durchgeführt werden, so daß im Anschluß an die Energiesimulation Bildsequenzen (Walkthrough-Animationen) erzeugt werden können. Eine Neuberechnung der Energieverteilung ist nur dann notwendig, wenn sich die Geometrie der Szene ändert. Die Beschränkung des globalen Beleuchtungsmodells auf ideal diffuse Materialeigenschaften der Objektoberflächen erscheint zunächst etwas restriktiv. Wenn jedoch die Tatsache berücksichtigt wird, daß der überwiegende Anteil der Objekte in realen Umgebungen einfallendes Licht primär diffus reflektiert (vor allen Dingen in Innenräumen), während spiegelnde Flächen eher die Ausnahme bilden, wird diese Einschränkung relativiert. Da jedoch das menschliche Auge sehr sensitiv für Glanzlichter und Spiegelungen ist, können diese Effekte bei der photorealistischen Bildgenerierung nicht vollständig vernachlässigt werden. Aus diesem Grund wurden verschiedene Erweiterungen der klassischen Radiosity-Methode eingeführt, die es ermöglichen, durch spiegelnde Oberflächen verursachte Beleuchtungseffekte bei der Bildgenerierung zu berücksichtigen.

3.1 Grundlagen der Radiosity-Methode

Das Grundprinzip des Radiosity-Verfahrens wird in diesem Abschnitt anhand des klassischen *Full-Matrix-Verfahren* beschrieben, wie es 1984 von Goral u.a. vorgestellt wurde [GTGB84]. Die einzelnen Teilaspekte werden hierbei detailliert dargestellt. Das *Progressiv-Refinement-Verfahren* stellt eine zur Full-Matrix-Methode alternative Vorgehensweise dar und wird in Abschnitt 3.2 erläutert. Diese Methode ermöglicht die Verarbeitung von komplexeren Eingabeszenen, sowie die Generierung von Bildern auf der Basis von Zwischenergebnissen der Radiosity-Berechnung.

3.1.1 Das Radiosity-Gleichungssystem

Die Radiosity-Methode basiert auf der Lösung des *Radiosity-Gleichungssystems*. Dieses Gleichungssystem beschreibt ein Energiegleichgewicht in einer geschlossenen Umgebung. Szenen, für die eine Radiosity-Lösung berechnet werden soll, unterliegen gewissen Voraussetzungen, um die diffuse Energieverteilung korrekt modellieren zu können.

- Alle Oberflächen in der Szene sind Lambert-Reflektoren (siehe Kapitel 2), d.h. sie besitzen ideal diffuse Emissions- und Reflexionseigenschaften.
- Alle Oberflächen sind homogen bezüglich ihrer Eigenschaften. Insbesondere besitzt jede Oberfläche eine konstante Farbe.
- Die Umgebung ist abgeschlossen, so daß ihr weder Energie von außen zugeführt noch entzogen wird.
- Jede Oberfläche kann Strahlungsquelle (primäre Lichtquelle), Reflektor (sekundäre Lichtquelle) oder beides sein.

Die Radiosity-Methode basiert auf einem radiometrischen Ansatz. Bei der Herleitung der Radiosity-Gleichung werden die folgenden grundlegenden Bezeichnungen verwendet:

- **Radiosity (B):** Die spezifische Ausstrahlung (engl. radiosity) einer Oberfläche, welche in die Umgebung abgegeben wird und mit Hilfe der Radiosity-Methode berechnet werden soll (W/m^2).
- **Emission (E):** Der Anteil der spezifischen Ausstrahlung (W/m^2) einer Oberfläche, der selbst emittiert wird. Bei Oberflächen, die keine primären Lichtquellen sind, ist dieser Wert Null.
- **Reflektivität (ρ):** Die Reflektivität (oder Reflexionskoeffizient) ist eine dimensionslose Größe und gibt den Anteil der auf eine Oberfläche einfallenden Energie an, der von dieser Oberfläche ideal diffus in die Umgebung reflektiert wird. ($0 < \rho \leq 1$).
- **Formfaktor (F_{ij}):** Der Anteil der Strahlungsleistung, den ein Oberflächenelement i an ein bestimmtes Oberflächenelement j in der Umgebung abgibt, wird durch einen sogenannten Formfaktor beschrieben. Dieser Wert ist abhängig von der Form, dem Flächeninhalt, der relativen Lage und der relativen Sichtbarkeit der entsprechenden Oberflächenelemente. Der Formfaktor wird auch oftmals als *Konfigurationsfaktor* bezeichnet und ist ein Wert ohne Einheit mit $0 \leq F_{ij} \leq 1$.

Die spezifische Ausstrahlung (Radiosity) einer Oberfläche ist als die gesamte Strahlungsleistung pro Flächeneinheit definiert, die von einem Oberflächenpunkt in sämtliche Richtungen

der Hemisphäre oberhalb des Punktes abgegeben wird. Durch die Einschränkung auf diffuse Energieausstrahlung (Emission und Reflexion), läßt sich dieser Sachverhalt durch die Gleichung (3.1) beschreiben:

$$B(x) = E(x) + \rho_d(x) \int_{y \in S} B(y) F_{yx} dy \quad (3.1)$$

Hierbei gilt:

- $B(x)$ ist die spezifische Ausstrahlung an einem Oberflächenpunkt x ;
- S ist die Menge aller Oberflächen einer Szene;
- $E(x)$ ist die Eigenemission eines Oberflächenpunktes x ;
- $\rho_d(x)$ ist die Reflektivität der Oberfläche, auf der Punkt x liegt (da die Materialeigenschaften der Oberflächen aufgrund der Voraussetzungen homogen sind, ist dieser Wert für eine Oberfläche konstant und wird im weiteren Verlauf durch ρ bezeichnet);
- F_{yx} ist der Formfaktor zwischen den Oberflächenpunkten y und x .

Gleichung (3.1) ist eine geringfügig modifizierte Version der in Abschnitt 2.5 hergeleiteten globalen Beleuchtungsgleichung für Szenen, deren Objekte ideal diffuse Reflexionseigenschaften aufweisen (Gleichung (2.34)).

Um die Anzahl der zu berechnenden Radiosities zu begrenzen, wird eine geschlossene Umgebung in eine endliche Anzahl diskreter Flächenelemente, den sogenannten *Patches*, unterteilt. Hierbei ist zu beachten, daß die Diskretisierung (d.h. Einteilung in Flächenelemente endlicher Größe – *Meshing*) einen wesentlichen Einfluß auf die Qualität der berechneten Lösung hat (siehe Abschnitt 3.3). Für ein Patch wird angenommen, daß es Lichtenergie gleichmäßig über die gesamte Fläche emittiert und reflektiert. Unter der Voraussetzung, daß eine geschlossene Umgebung in n Patches unterteilt ist und jedes Patch ein nicht-durchsichtiger, ideal diffuser Emittent oder Reflektor ist, kann mit Hilfe der Radiosity-Methode für jedes Patch die jeweilige spezifische Ausstrahlung berechnet werden. Dieser Wert bildet im Anschluß an eine Radiosity-Berechnung die Grundlage des Bildgenerierungsprozesses, indem die Radiosity-Werte in Intensitäten (bzw. Farben) umgerechnet werden und dann ein interpolatives Schattierungsverfahren zur Erzeugung von Bildern verwendet wird.

Die gesamte Strahlungsleistung ist die Summe der Energie, welches ein Patch i selbst als Strahlungsquelle emittiert, sowie der Anteil der Strahlungsleistung, welcher von übrigen Patches der Umgebung an das Patch i abgegeben und dann von Patch i wieder diffus reflektiert wird (siehe Abb. 3.1).

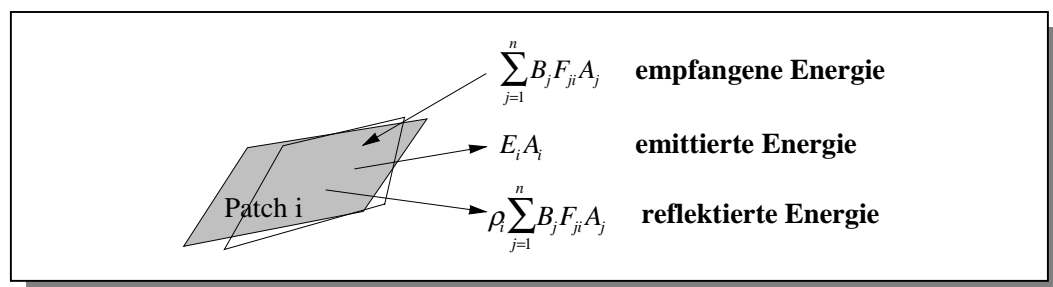


Abb. 3.1: Spezifische Ausstrahlung (Radiosity) eines Patches i .

In diskretisierten Umgebungen kann die Integralgleichung (3.1) durch die endliche Summengleichung

$$B_i A_i = E_i A_i + \rho_i \sum_{j=1}^n B_j F_{ji} A_j \quad (3.2)$$

ersetzt werden, wobei gilt:

- B_i ist die Radiosity von Patch i ;
- A_i ist der Flächeninhalt von Patch i ;
- E_i ist die Emission von Patch i ;
- ρ_i ist die Reflektivität von Patch i ;
- F_{ji} ist der Formfaktor von Patch j nach Patch i und
- n ist die Anzahl der diskreten Patches.

Diese lineare Gleichung gibt die Strahlungsleistung eines endlichen Flächenstückes i (Patch i) an (in Watt), die in die Umgebung abgegeben wird. Die Herleitung dieser diskreten Formulierung der Radiosity-Gleichung ist sehr ausführlich in [SP94] beschrieben.

Von Siegel und Howell wurden die wechselseitigen Beziehungen zwischen Emittlern und Empfängern von Lichtenergie untersucht [SH81], welche sich durch eine reziproke Beziehung von Formfaktoren darstellen lassen.

$$F_{ij} A_i = F_{ji} A_j \quad (3.3)$$

Mit Hilfe dieser Beziehung zwischen den Formfaktoren läßt sich die angegebene lineare Gleichung umformen, so daß keine Abhängigkeiten von den Größen der Patches bestehen und somit die spezifische Ausstrahlung (W/m^2) eines Patches i bestimmt wird. Die Radiosity-Gleichung ist gegeben durch:

$$B_i = E_i + \rho_i \sum_{j=1}^n B_j F_{ij} \quad (3.4)$$

Ein Energiegleichgewicht in einer geschlossenen Umgebung, die in n Patches diskretisiert wurde, läßt sich durch ein lineares Gleichungssystem beschreiben:

$$\begin{bmatrix} 1 - \rho_1 F_{11} & -\rho_1 F_{12} & \cdots & -\rho_1 F_{1n} \\ 1 - \rho_2 F_{22} & 1 - \rho_2 F_{22} & \cdots & 1 - \rho_2 F_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ -\rho_n F_{n1} & -\rho_n F_{n2} & \cdots & 1 - \rho_n F_{nn} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_n \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_n \end{bmatrix} \quad (3.5)$$

Das Gleichungssystem (3.5) kann mit Hilfe der bekannten Lösungsverfahren für lineare Gleichungssysteme gelöst werden. Aufgrund spezieller Eigenschaften der angegebenen Matrix ist es jedoch möglich, effizientere Methoden zur Berechnung der Radiositäten anzuwenden. Da die Summe der Formfaktoren einer Zeile der Matrix per Definition Eins ergibt und F_{ii} für alle Patches i gleich Null ist, liegt eine Matrix vor, welche eine diagonale Dominanz aufweist. In diesem speziellen Fall ergeben alle Terme auf der Hauptdiagonalen der Matrix den Wert Eins, während die übrigen Terme Werte kleiner als Eins bilden. Somit sind die Voraussetzungen für

eine Anwendung des iterativen Gauss-Seidel-Verfahrens zur Lösung von Gleichungssystemen erfüllt. Dieses Konvergenz-Verfahren garantiert für Gleichungssysteme mit den oben beschriebenen Eigenschaften eine schnelle Konvergenz [Horn75].

Radiosity-Verfahren, welche auf der Lösung des gesamten Gleichungssystems basieren, werden auch als *Full-Matrix-Verfahren* (FM-Verfahren) bezeichnet. Der Ablauf des Full-Matrix-Verfahrens ist in Abb. 3.2 beschrieben.

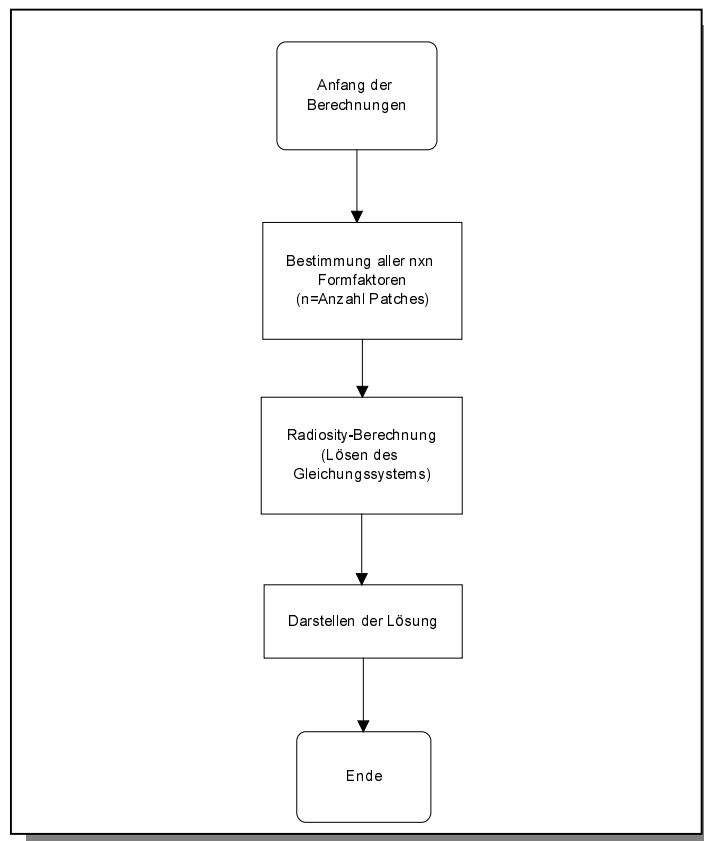


Abb. 3.2: Ablaufdiagramm des Full-Matrix-Verfahrens.

Zur Berechnung der Radiositäten mit Hilfe des Full-Matrix-Verfahrens müssen die n^2 Formfaktoren für die n Patches einer abgeschlossenen Umgebung vor dem eigentlichen Lösen des Gleichungssystems bestimmt werden, wodurch eine Platzkomplexität von $O(n^2)$ entsteht. Mit Hilfe des Iterationsverfahrens wird im Anschluß an die Formfaktorberechnungen eine konvergierte Lösung des Gleichungssystems berechnet. Somit setzen sich die Kosten für das gesamte Verfahren aus den Kosten für die Bestimmung der Formfaktoren und den Kosten für das Lösen des Gleichungssystems zusammen. Hierbei entfällt jedoch der überwiegende Anteil der Berechnungszeit auf die Bestimmung der Formfaktoren innerhalb der Szene [CCWG88].

Im weiteren Verlauf dieser Arbeit wird die Intensität eines Oberflächenpunktes durch dessen spezifische Ausstrahlung einer Wellenlänge des sichtbaren Spektrums beschrieben, wodurch eine einfarbige (monochrome) Beschreibung des Lichtes gegeben ist. In einer realen Beleuchtungssituation muß berücksichtigt werden, daß es sich bei der ausgesandten Strahlung um ein spezifisches elektromagnetisches Energiespektrum handelt, dessen Ausprägung von den Emissions- und Reflexionseigenschaften des strahlenden Objektes abhängt [Bres97]. Die verwendeten Modellierungsparameter sind abhängig von der Wellenlänge (z.B. Reflexions-

koeffizienten). Somit muß das Radiosity-Gleichungssystem prinzipiell für jede Wellenlänge des sichtbaren Spektrums berechnet werden. Der quantitativen Farbwahrnehmungsanalyse von Meyer [Mey90] entsprechend, ist es in den praktisch auftretenden Fällen ausreichend, vier charakteristische Wellenlängen für die menschliche Wahrnehmung herauszugreifen oder – noch restriktiver – die drei Primärfarben der Bildgenerierung Rot, Grün und Blau (siehe Kapitel 2) zu verwenden und entsprechend drei Radiosity-Gleichungssysteme aufzustellen. Somit ergibt sich als Lösung der Radiosity-Berechnungen für jedes Flächenelement ein Zahlentripel mit spezifischen Ausstrahlungen verschiedener Wellenlängen. Diese werden in Intensitätswerte (Farben) umgewandelt, mit denen ein Ausgabemedium wie ein Monitor angesteuert werden kann [CG85].

3.1.2 Bestimmung von Formfaktoren

Der Wert, welcher den Anteil der Strahlungsleistung eines Flächenstücks A_i spezifiziert, der auf das gesamte Flächenstück A_j übergeht, wird als Formfaktor (alternativ Konfigurationsfaktor) bezeichnet. Bei der Bestimmung dieses dimensionslosen Wertes muß die Form, der Flächeninhalt sowie die relative Lage beider Teilstücke berücksichtigt werden. Der Formfaktor ist über die Strahlungsleistung (angegeben in Watt) der Fläche i definiert und nicht über deren spezifische Ausstrahlung (angegeben in W/m^2). Der Term $A_i B_i F_{ij}$ gibt somit die insgesamt auf Fläche j auftreffende Strahlungsleistung an, die von Fläche i abgegeben wurde.

Formfaktoren können unabhängig von Standpunkten eines Betrachters oder den physikalischen Eigenschaften der Oberflächen berechnet werden. Aus diesem Grund ist es möglich, den Betrachterstandpunkt in der Szene zu ändern oder die Reflektivitäten der Patches zu variieren, ohne daß die Formfaktoren neu berechnet werden müssen. Eine Neuberechnung von Formfaktoren muß nur dann vorgenommen werden, wenn die Geometrie der Szene durch eine neue Anordnung der Objekte verändert wird.

3.1.2.1 Die Formfaktor-Gleichung

Der Formfaktor zwischen den unendlich kleinen Flächen dA_i und dA_j hängt von der Entfernung r zwischen beiden Flächen und ihrer relativen Anordnung zueinander ab, welche durch die Winkel ϕ_i und ϕ_j bestimmt ist. Diese Winkel werden durch die Flächennormalen der Patches und der Verbindungslinie zwischen dA_i und dA_j gebildet (siehe Abb. 3.3).

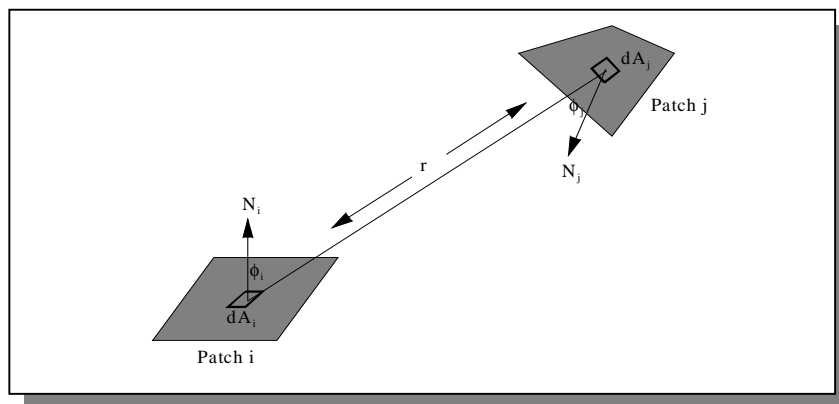


Abb. 3.3: Formfaktorberechnung zwischen Patch i und Patch j .

Der Energieanteil, der von einer unendlich kleinen Fläche dA_i an eine unendlich kleine Fläche dA_j abgegeben wird, läßt sich durch folgende Gleichung beschreiben (*Punkt-zu-Punkt Formfaktor*):

$$F_{dA_i dA_j} = \frac{\cos \phi_i \cos \phi_j dA_j}{\pi r^2} \quad (3.6)$$

Die Herleitung dieser Formfaktor-Gleichung wird ausführlich in der Arbeit von Goral, Torrance, Greenberg und Bataile geschildert [GTGB84]. Durch Integration über die Fläche j ergibt sich der Formfaktor von der unendlich kleinen Fläche dA_i zu der endlichen Fläche A_j (*Punkt-zu-Patch Formfaktor*):

$$F_{dA_i A_j} = \int_{A_j} \frac{\cos \phi_i \cos \phi_j}{\pi r^2} dA_j \quad (3.7)$$

Der Formfaktor zwischen den endlichen Flächen (Patches) A_i und A_j ist als Flächendurchschnitt definiert (*Patch-zu-Patch Formfaktor*):

$$F_{A_i A_j} = \frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{\cos \phi_i \cos \phi_j}{\pi r^2} dA_i dA_j \quad (3.8)$$

Diese Darstellung der Formfaktor-Gleichung berücksichtigt nicht den Aspekt der relativen Sichtbarkeit von Patches. Es ist jedoch zu beachten, daß die gegenseitige Sichtbarkeit zwischen zwei Patches über die Flächen betrachtet aufgrund von partiellen Verdeckungen variieren kann, wodurch natürlich auch der Energietransfer betroffen ist. Aus diesem Grund wird der Gleichung zur Bestimmung der Formfaktoren ein zusätzlicher Term H_{ij} hinzugefügt. Abhängig davon, ob eine unendlich kleinen Fläche dA_j von dA_i aus betrachtet sichtbar ist oder nicht, kann H_{ij} die Werte Eins oder Null annehmen. Somit wird die Gleichung (3.8) umgeschrieben zu:

$$F_{A_i A_j} = \frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{\cos \phi_i \cos \phi_j}{\pi r^2} H_{ij} dA_i dA_j \quad (3.9)$$

Im weiteren Verlauf der Arbeit wird aus Gründen der einfacheren Schreibweise ein Patch-zu-Patch Formfaktor zwischen zwei Patches i und j als F_{ij} bezeichnet.

Die Formfaktoren besitzen bestimmte Eigenschaften, die für die Korrektheit der Radiosity-Methode von wesentlicher Bedeutung sind.

- Um die Erhaltung der Energie in einer geschlossenen Umgebung zu garantieren, muß die gesamte Strahlungsleistung berücksichtigt werden, die von einer Oberfläche an die Umgebung abgegeben wird. Wenn die Oberflächen einer Szene in n Patches diskretisiert wurden, gilt für die Formfaktoren eines Patches i

$$\sum_{j=1}^n F_{ij} = 1 \quad (3.10)$$

- Für jedes Patch i gilt:

$$F_{ii} = 0 \tag{3.11}$$

- Im allgemeinen ist der Formfaktor von einem Patch i auf ein Patch j ungleich dem umgekehrten Formfaktor.

$$F_{ij} \neq F_{ji} \tag{3.12}$$

- Die reziproke Beziehung von Formfaktoren zwischen zwei Patches i und j läßt sich wie folgt beschreiben:

$$A_i F_{ij} = A_j F_{ji} \tag{3.13}$$

3.1.2.2 Das Analogon von Nusselt

Von Nusselt wurde eine geometrische Interpretation des Formfaktor-Integrals (3.8) entwickelt [Nuss51]. Unter der Annahme, daß der Mittelpunkt eines Patches repräsentativ für alle anderen Punkte der Patchfläche ist, kann F_{ij} durch $F_{dA_i A_j}$ approximiert werden, wobei dA_i in der Mitte des Patches i liegt. Zur Berechnung von F_{ij} wird oberhalb von dA_i eine Einheits-Halbkugel zentriert positioniert. Die sichtbaren Teile von A_j werden mittels Zentralprojektion auf diese Halbkugel projiziert. Anschließend wird eine Orthogonalprojektion vorgenommen, welche die projizierte Fläche von A_j auf die Grundfläche der Einheits-Halbkugel abbildet (siehe Abb. 3.4). Das Verhältnis von projizierter Fläche zur gesamten Grundfläche der Halbkugel ist äquivalent zu dem Formfaktor $F_{dA_i A_j}$.

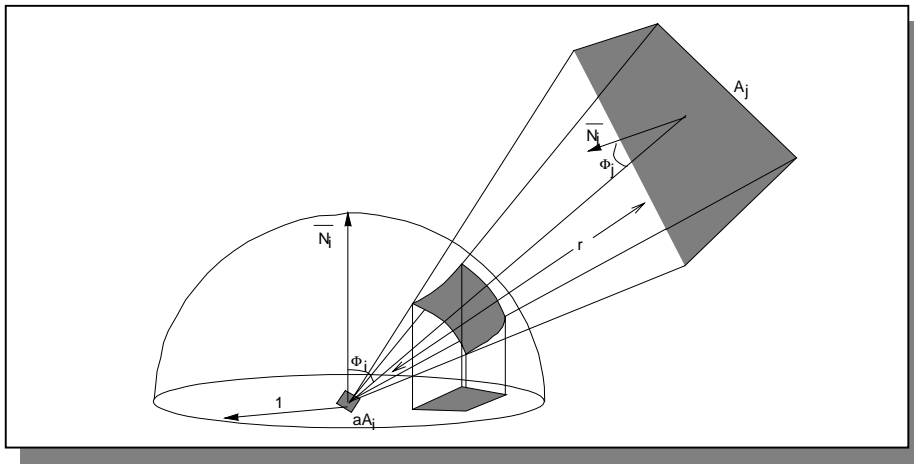


Abb. 3.4: Bestimmung des Formfaktors $F_{dA_i A_j}$ basierend auf dem Analogon von Nusselt.

3.1.2.3 Die Hemicube-Methode

Eine analytische Berechnung der Formfaktoren ist im allgemeinen für beliebige geometrische Konstellationen in einer geschlossenen Form nicht möglich. Von Cohen u.a. wurde die *Hemicube-Methode* entwickelt, die es ermöglicht, Formfaktoren in komplexen Umgebungen zu approximieren [CG85]. Aus dem Analogon von Nusselt kann gefolgert werden, daß alle Pat-

ches j , deren Projektionen auf die Halbkugel über dem Mittelpunkt von Patch i Flächen gleicher Größe belegen, auch Formfaktoren F_{ij} mit identischen Werten besitzen (siehe Abb. 3.5).

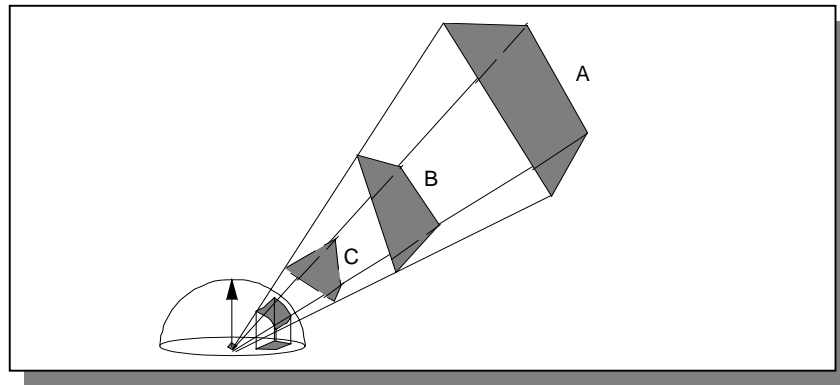


Abb. 3.5: Die Projektionsflächen von A, B und C auf der Einheitshalbkugel oberhalb des Projektionszentrums sind gleich groß.

Dieser Effekt kann auch erzielt werden, wenn die Projektionen auf einen Halbwürfel (*Hemicube*) anstatt auf eine Halbkugel durchgeführt werden. Bei dem Ansatz von Cohen und Greenberg werden die Patches der Umgebung auf einen Hemicube projiziert, welcher zentriert über einem Patch i positioniert wird und dessen obere Seite parallel zu diesem Patch verläuft [CG85]. Über jede der Seiten des Hemicubes wird eine Gitterstruktur gelegt. Jede der so entstandenen Gitterzellen definiert eine bestimmte Richtung und Raumwinkel. Somit existiert zu jeder Gitterzelle (Pixel) p_i ein *Delta-Formfaktor* ΔF_{p_i} .

Für die Pixel eines Hemicubes können die Delta-Formfaktoren vorab berechnet und in einer Tabelle gespeichert werden, so daß sie während der Berechnung von Patch-Formfaktoren nicht jeweils neu bestimmt werden müssen. Zur Berechnung der Formfaktoren eines Patches i werden die übrigen Patches der Umgebung durch eine Zentralprojektion auf die fünf Seiten des Hemicubes projiziert. Hierbei können mit Hilfe eines Z-Buffers [FvDFH90] gegenseitige Verdeckungen bestimmt werden, so daß nur die vom Zentrum des Patches direkt sichtbaren Teile der Patches der Umgebung auf den Hemicube abgebildet werden (siehe Abb. 3.6).

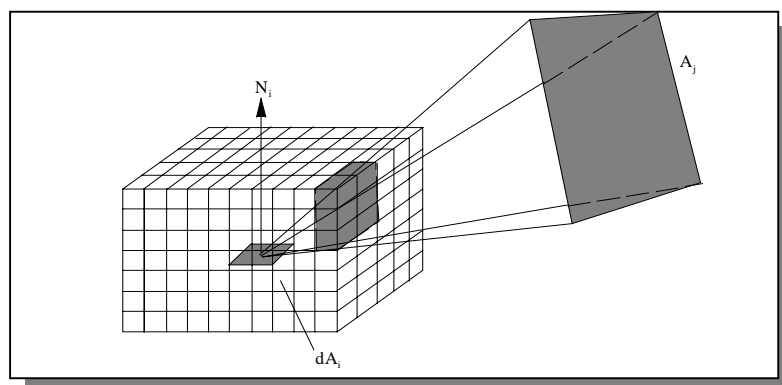


Abb. 3.6: Bestimmung von Formfaktoren mit Hilfe der Hemicube-Methode.

Ein Formfaktor F_{ij} von Patch i zu einem Patch j der Umgebung wird approximiert, indem die Delta-Formfaktoren der m Pixel des Hemicubes aufsummiert werden, die durch die Projektion von Patch j belegt sind.

$$F_{A_i A_j} \approx F_{dA_i A_j} \approx \sum_{k=1}^m \Delta F_k \quad (3.14)$$

Es ist offensichtlich, daß die Qualität der Formfaktor-Approximation von den Pixel-Auflösungen der Seitenflächen des Hemicubes abhängt.

Mit der vollständigen Berechnung eines Hemicubes, der über Patch i positioniert ist, werden sämtliche Formfaktoren der i -ten Zeile des Radiosity-Gleichungssystems bestimmt. In einer geschlossenen Umgebung, die durch n diskrete Patches modelliert wurde, müssen somit n Hemicubes vollständig berechnet werden, um alle Formfaktoren zu bestimmen.

Von Baum u.a. [BRW89] wurde gezeigt, daß die Verwendung der Hemicube-Methode zur Berechnung von Formfaktoren zu Fehlern führt, welche die Genauigkeit der berechneten Radiosity-Lösung erheblich mindern.

- **Fehler bei Patches mit einer geringen relativen Distanz (Proximity Assumption)**
Bei der Formfaktorbestimmung basierend auf der Hemicube-Methode wird lediglich das innere Integral der Gleichung (3.9) bestimmt. D.h. der *Patch-zu-Patch* Formfaktor (3.9) wird durch einen *Punkt-zu-Patch* Formfaktor (3.7) angenähert. Wenn Polygone im Verhältnis zu ihrer Größe eine geringe Distanz zueinander haben, variiert der Kern des Integrals sehr stark über der Fläche des Patches. Falls die Patches zwischen denen ein Formfaktor bestimmt werden soll eine gemeinsame Kante haben, kann es vorkommen, daß der Kern entlang der Kante singularär ist (z.B. wenn die Patches rechtwinklig aufeinander stehen).
- **Sichtbarkeitsfehler (Visibility Assumption)**
Die Sichtbarkeitsberechnungen werden bei der Formfaktorbestimmung auf Basis der Hemicube-Methode bzgl. eines stellvertretenden Punktes auf einem der Patches durchgeführt. Dieses ist natürlich nur eine sehr ungenaue Approximation der tatsächlichen relativen Sichtbarkeit zwischen zwei Patches der Umgebung.
- **Fehler durch endliche Hemicube-Auflösung (Resolution/Aliasing Assumption)**
Die Hemicube-Methode bestimmt die von einem stellvertretenden Punkt eines Patches aus betrachtet sichtbaren Patches der gesamten Szene, indem die Oberflächen der Szene auf die regelmäßig in Pixel diskretisierten Seitenflächen eines Hemicubes projiziert werden. Diese Vorgehensweise entspricht einer Abtastung (*Sampling*) der Umgebungsbeschreibung von diesem Punkt aus betrachtet in endlich viele Richtungen. Aus diesem Grund hat die Auflösung (Anzahl der Pixel) der Seitenflächen des Hemicubes und somit die Anzahl der verwendeten Sample-Richtungen einen wesentlichen Einfluß auf die Qualität der Sichtbarkeitsbestimmungen. Weiterhin führt die regelmäßige Sampling-Struktur des Hemicubes zu regelmäßigen Helligkeitsschwankungen bei der Darstellung der Radiosity-Lösung.

3.1.2.4 Analytische Bestimmung von Formfaktoren

Von Baum u.a. [BRW89] wurde ein Methode zur Formfaktorberechnung eingeführt, welche die Fehler der Hemicube-Methode vermeidet, indem in Situationen, in denen die Approxima-

tion von *Patch-zu-Patch* Formfaktoren durch *Punkt-zu-Patch* Formfaktoren zu ungenau ist, direkt *Patch-zu-Patch* Formfaktoren bestimmt werden. Hierbei wird das äußere Integral in Gleichung (3.9) numerisch integriert, während das innere Integral analytisch bestimmt wird. Bei der analytischen Bestimmung des inneren Integral (*Punkt-zu-Patch* Formfaktor) wird der *Satz von Stokes* ausgenutzt, mit dessen Hilfe das Flächenintegral in der Formel zur Bestimmung des *Punkt-zu-Patch* Formfaktors $F_{dA_i A_j}$ (Gleichung (3.7)) in ein Kontur-Integral überführt werden kann [HS67].

Für ein beliebig orientiertes Polygon A_j , kann dieses Kontur-Integral punktweise für eine unendlich kleine Fläche dA_i wie folgt in geschlossener Form ausgewertet werden:

$$F_{dA_i A_j} = \frac{1}{2\pi} \sum_{g \in G_j} N_i \cdot \Gamma_g \quad (3.15)$$

Hierbei ist

- G_j die Menge der Kanten von A_j ,
- N_i die Normale der unendlich kleinen Fläche dA_i und
- Γ_g ein Vektor der Länge γ_g (Bogenmaß) und der Richtung, die durch das Kreuzprodukt der Vektoren R_g und R_{g+1} (wie in Abb. 3.7) gegeben ist.

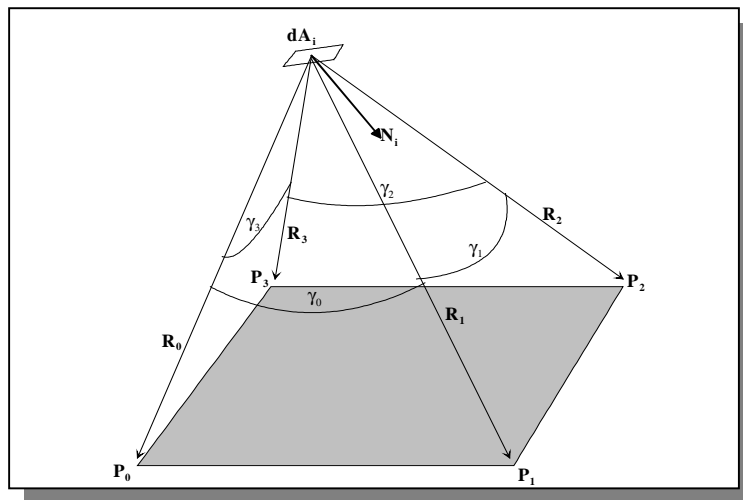


Abb. 3.7: Geometrie zur Auswertung der analytischen Formfaktoren.

Zur Berechnung von *Patch-zu-Patch* Formfaktoren (Gleichung (3.9)) wird das äußere Integral numerisch integriert, indem die angegebene analytische Formel zur Bestimmung des inneren Integrals (3.15) an verschiedenen Sample-Punkten der Fläche A_i ausgewertet wird. Werden m Sample Punkte s_1, \dots, s_m auf der Fläche A_i regelmäßig verteilt, so wird der Formfaktor F_{ij} durch

$$F_{ij} \approx \frac{A_i}{m} \sum_{k=1}^m F_{s_k A_j} V_{s_k A_j} \quad (3.16)$$

angenähert. Es ist zu beachten, daß bei der analytischen Bestimmung des inneren Integrals für jeden Sample-Punkt die relative prozentuale Sichtbarkeit $V_{s_k A_j}$ ($0 \leq V_{s_k A_j} \leq 1$) bestimmt werden muß. Dieses kann sowohl mit Hilfe der Hemicube-Technik, als auch mit Hilfe eines Strahlentests (siehe hierzu auch Abschnitt 3.1.2.5) geschehen.

3.1.2.5 Approximation von Formfaktoren mit Hilfe von Raytracing

Von Wallace u.a. wurde eine neue Methode zur Formfaktorapproximation eingeführt. Diese basiert auf einer Raytracing-Technik und vermeidet die Fehler, die bei der Berechnung von Formfaktoren mittels Hemicube-Methode auftreten [WKEH89].

Ein wesentliches Problem der Hemicube-Methode besteht darin, daß die Szene von einem repräsentativem Sample-Punkt auf einem Patch i aus betrachtet, in die durch Hemicube-Pixel festgelegten Richtungen gleichmäßig abgetastet wird (siehe Abb. 3.8).

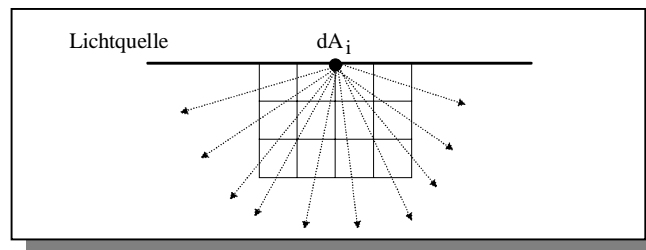


Abb. 3.8: Regelmäßige Sampling-Struktur der Hemicube-Methode.

Die auf diese Weise berechneten *Punkt-zu-Patch* Formfaktoren werden als Näherung für *Patch-zu-Patch* Formfaktoren verwendet, was jedoch nur in dem Fall von weit voneinander entfernten Patches eine brauchbare Approximation ist. Weiterhin wird bei der Hemicube-Methode die Tatsache vollkommen vernachlässigt, daß die Radiosity-Informationen eigentlich an den Eckpunkten des Polygonnetzes bestimmt werden müssen, um die Radiosity-Lösung anschließend mit Hilfe eines interpolativen Schattierungsverfahrens (z.B. Gouraud-Verfahren - siehe Abschnitt 3.4) darstellen zu können.

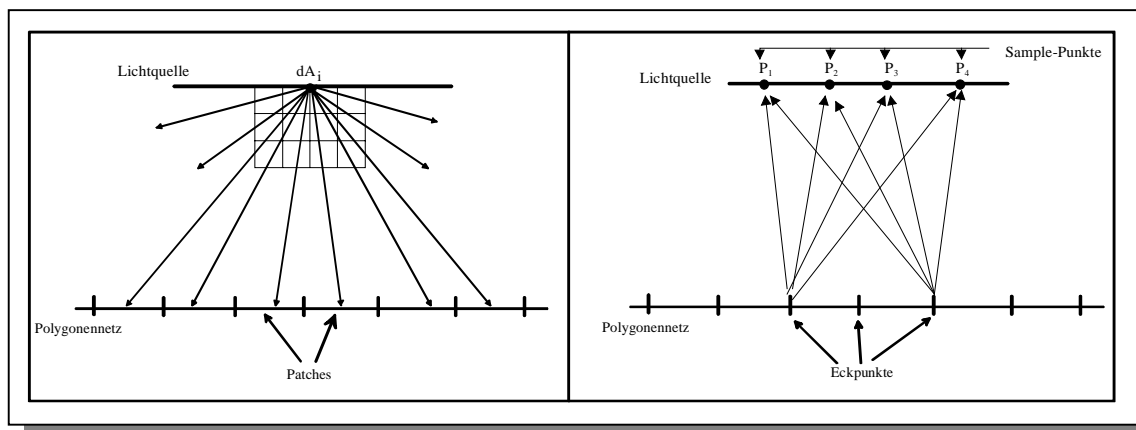


Abb. 3.9: Sampling-Ansatz bei der Berechnung von Patch-Radiositäten mit der Hemicube-Methode (links) verglichen mit der direkten Berechnung von Eckpunkt-Radiositäten (rechts).

Bei dem Verfahren von Wallace u.a. [WKEH89] wird ein Energieaustausch zwischen einem Patch und den Eckpunkten der übrigen Patches der Szene auf Basis von *Punkt-zu-Patch* Formfaktoren berechnet. Dadurch werden die Radiosity-Informationen direkt (ohne Interpolation) an den Punkten berechnet, an denen sie zu Zwecken der Bildgenerierung benötigt werden (siehe Abb. 3.9).

Diese Vorgehensweise kehrt den Sampling-Prozeß verglichen mit der Hemicube-Methode um. Die Sichtbarkeit der Lichtquelle wird bei dem Verfahren von Wallace u.a. aus Sicht der Empfängerelemente (Eckpunkte) bestimmt.

Weiterhin werden bei der Approximation von Formfaktoren alle Fehler vermieden, die durch die regelmäßige Sampling-Struktur der Hemicube-Methode bzw. einer zu geringen Auflösung der Hemicube-Seitenflächen entstehen. So kann bei der Hemicube-Methode der Fall auftreten, daß kleine Flächen bei der Verteilung von Energie eines Patches i unberücksichtigt bleiben, da sie durch die Projektion auf die Hemicube-Seiten nicht erfaßt werden (siehe Abb. 3.10), obwohl sie von dem Projektionspunkt dA_i aus sichtbar sind.

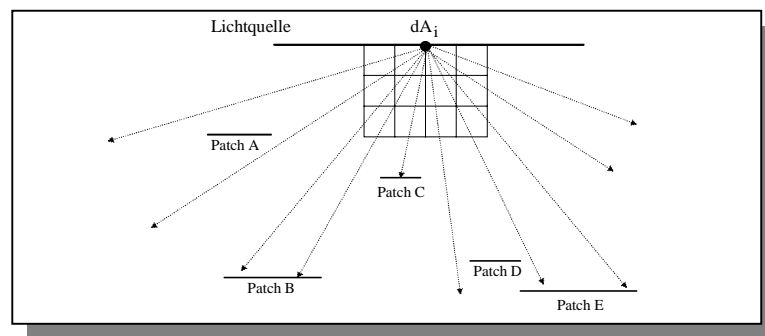


Abb. 3.10: Sampling-Fehler der Hemicube-Methode (Patches A und D werden nicht erfaßt).

Wallace u.a. führen die eigentliche Formfaktorberechnung als Kombination von Raytracing und analytischer Formfaktorberechnung durch. Das Raytracing dient dabei lediglich zur Überprüfung der relativen Sichtbarkeit zwischen einem Eckpunkt und einer endlichen Fläche (primäre oder sekundäre Lichtquelle). Da sich die Sichtbarkeitsverhältnisse über eine Fläche verändern können, wird eine variable Anzahl N von Referenzpunkten (*Sample-Punkten*) P_k ($1 \leq k \leq N$) auf einer Fläche A_j festgelegt, die gleichmäßig auf dem Patch verteilt sind (siehe Abb. 3.11).

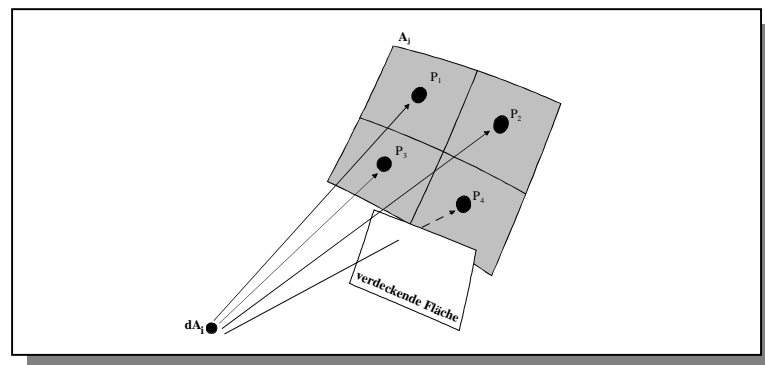


Abb. 3.11: Platzierung mehrerer Referenzpunkten auf der Fläche A_j .

Wie in Abschnitt 3.1.2.1 schon gezeigt wurde, ist der Formfaktor zwischen einer unendlich kleinen Fläche dA_i und einer endlichen Fläche A_j gegeben durch:

$$F_{dA_i A_j} = \int_{A_j} \frac{\cos \vartheta_i \cos \vartheta_j}{\pi r^2} dA_j \quad (3.17)$$

Dieses Integral kann für sehr einfache Fälle analytisch gelöst werden. Im allgemeinen wird es jedoch numerisch approximiert, indem es als eine Summe von einfacheren Formfaktoren ausgedrückt wird, die für kleinere Flächen $A_{j,k}$ berechnet werden, welche eine Überdeckung der Fläche A_j bilden (siehe Abb. 3.12).

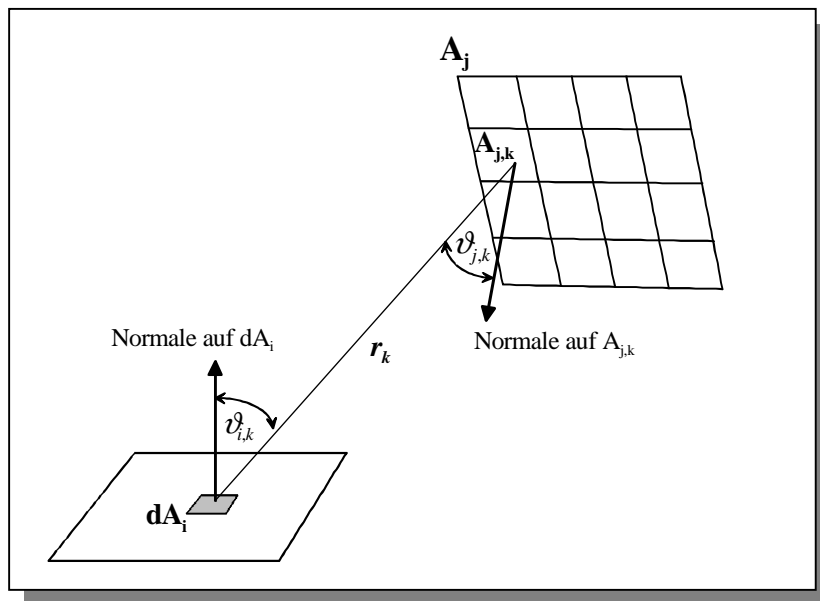


Abb. 3.12: Diskretisierung der Fläche A_j in kleinere Teilflächen.

Somit gilt:

$$F_{dA_i A_j} = \sum_{k=1}^N \frac{\cos \vartheta_{i,k} \cos \vartheta_{j,k}}{\pi r_k^2} A_{j,k} \quad (3.18)$$

Eine wichtige Voraussetzung für die Gültigkeit dieser Näherung ist, daß die Flächenstücke $A_{j,k}$ im Vergleich zu r_k klein sind. Wird r_k kleiner, so muß auch A_j feiner unterteilt werden (d.h. $A_{j,k}$ wird verkleinert).

Bei der Methode von Wallace u.a. werden N Referenzpunkte P_k gleichmäßig auf der Fläche A_j verteilt. Um einen Referenzpunkt P_k wird ein kreisförmiges Flächenelement $A_{j,k}$ mit Radius a_k gebildet (siehe Abb. 3.13), wobei gilt:

$$A_j = \sum_{k=1}^N A_{j,k} \quad (3.19)$$

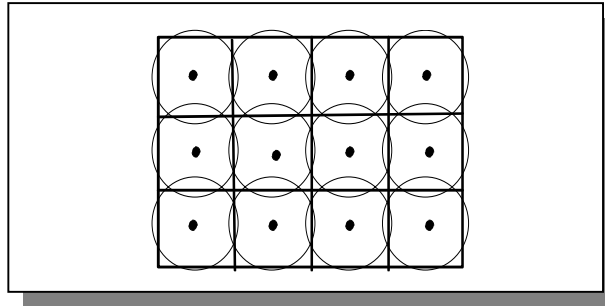


Abb. 3.13: Annäherung der Fläche eines Patches durch zwölf Kreisscheiben.

Für den exakten Formfaktor zwischen dA_i und einer Kreisscheibe $A_{j,k}$, die sich parallel gegenüber liegen, gibt es eine exakte (analytische) Formel [SH81] (siehe Abb. 3.14):

$$F_{dA_i A_{j,k}} = \frac{a_k^2}{r_k^2 + a_k^2} \quad (3.20)$$

Hierbei sind a_k der Radius der Kreisscheibe $A_{j,k}$ mit Mittelpunkt P_k und r_k der Abstand zwischen dA_i und P_k .

Wenn das unendlich kleine Flächenelement dA_i und die Kreisscheibe $A_{j,k}$ beliebig zueinander orientiert sind, gilt:

$$F_{dA_i A_{j,k}} = \frac{\cos \vartheta_{i,k} \cos \vartheta_{j,k} a_k^2}{r_k^2 + a_k^2} \quad (3.21)$$

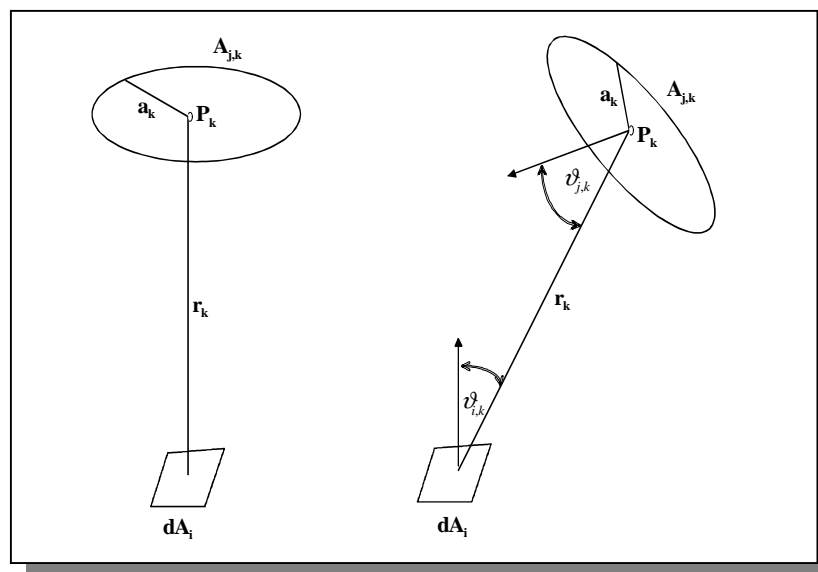


Abb. 3.14: Analytischer Formfaktor von einem Punkt zu einer Kreisscheibe.

Auch die Winkel $\vartheta_{i,k}$ und $\vartheta_{j,k}$ zwischen den Normalen und der Verbindungsstrecke sind für jeden Referenzpunkt verschieden und bekommen deshalb den Index k .

Da die $A_{j,k}$ die emittierenden Flächen sind, wird jedoch der reziproke Formfaktor $F_{A_{j,k}dA_i}$ benötigt. Mit Hilfe der Reziprozitätsbeziehung zwischen Formfaktoren und der Formel $A_{j,k} = \pi a_k^2$ erhält man:

$$F_{A_{j,k}dA_i} = \frac{\cos \vartheta_{i,k} \cos \vartheta_{j,k}}{\pi r_k^2 + A_{j,k}} dA_i \quad (3.22)$$

Zur Überprüfung der relativen Sichtbarkeit wird von dA_i ein Strahl in Richtung des Punktes P_k generiert. Dieser Strahl wird auf einen Schnittpunkt mit den übrigen Objekten der Umgebung überprüft. Liegt ein Objekt zwischen dA_i und einem Referenzpunkt P_k , so trägt $A_{j,k}$ nichts zum Formfaktor $F_{A_j dA_i}$ zwischen A_j und dA_i bei.

Der gesamte Formfaktor kann damit wie folgt berechnet werden:

$$F_{A_j dA_i} = \frac{1}{N} \sum_{k=1}^N \delta_k F_{A_{j,k} dA_i}, \quad (3.23)$$

wobei $\delta_k = 1$, wenn $A_{j,k}$ nicht verdeckt ist, sonst 0.

Mit Gleichung (3.22) ergibt sich schließlich:

$$F_{A_j dA_i} = dA_i \frac{1}{N} \sum_{k=1}^N \delta_k \frac{\cos \vartheta_{i,k} \cos \vartheta_{j,k}}{\pi r_k^2 + A_{j,k}}. \quad (3.24)$$

Der Beitrag der Sender-Fläche A_j zur spezifischen Ausstrahlung eines Eckpunktes i des Radiosity-Netztes ist gegeben durch:

$$B_i = \rho_i B_j A_j \frac{1}{N} \sum_{k=1}^N \delta_k \frac{\cos \vartheta_{i,k} \cos \vartheta_{j,k}}{\pi r_k^2 + A_{j,k}}. \quad (3.25)$$

Zusätzlich zu den schon aufgeführten Vorteilen der Methode von Wallace u.a. ergeben sich noch weitere positive Nebeneffekte, die bei der Berechnung von Radiosity-Lösungen ausgenutzt werden können. So können z.B. verschiedene Lichtquellentypen wie Flächen-, Punkt- und Spot-Lichtquellen bei der Radiosity-Simulation verwendet werden. Weiterhin lassen sich Sichtbarkeitstests mit der exakten Beschreibung der Geometrie (analytische Flächenbeschreibungen) durchführen. Einen wesentlichen Beitrag zur Qualitätssteigerung der erzeugten Bilder trägt die Tatsache bei, daß transparente- und semi-transparente Objekte bei der Schattenberechnung berücksichtigt werden können.

Der für diese Arbeit wichtigste Aspekt ist jedoch, daß das Verfahren von Wallace u.a. sehr gut parallelisierbar ist, da Formfaktoren zwischen verschiedenen Eckpunkten der Umgebungsbeschreibungen und Lichtquellen unabhängig voneinander berechnet werden können.

3.1.2.6 Monte Carlo Integration des Formfaktor-Integrals

Mit den bisher vorgestellten analytischen und numerischen Methoden zur Bestimmung eines *Punkt-zu-Patch* Formfaktors (d.h. Formfaktor zwischen einer unendlich kleinen Fläche und einer endlichen Fläche) kann der vollständige *Patch-zu-Patch* Formfaktor (siehe Gleichung (3.9)) zwischen zwei endlichen Flächen A_i und A_j approximiert werden. Zu diesem Zweck werden *Punkt-zu-Patch* Formfaktoren (siehe Gleichung (3.7)) für mehrere Punkte auf der Fläche A_i bzgl. dem Patch A_j bestimmt und der Durchschnittswert dieser Formfaktoren als *Patch-zu-Patch* Formfaktor gesetzt.

Durch diese Vorgehensweise erfolgt jedoch eine unzureichende Abtastung (Sampling) des Flächenelementes A_i . Abhilfe schafft hier ein Verfahren, welches das doppelte Flächenintegral des *Patch-zu-Patch* Formfaktors mit Hilfe von Monte Carlo Integration approximiert [CW93].

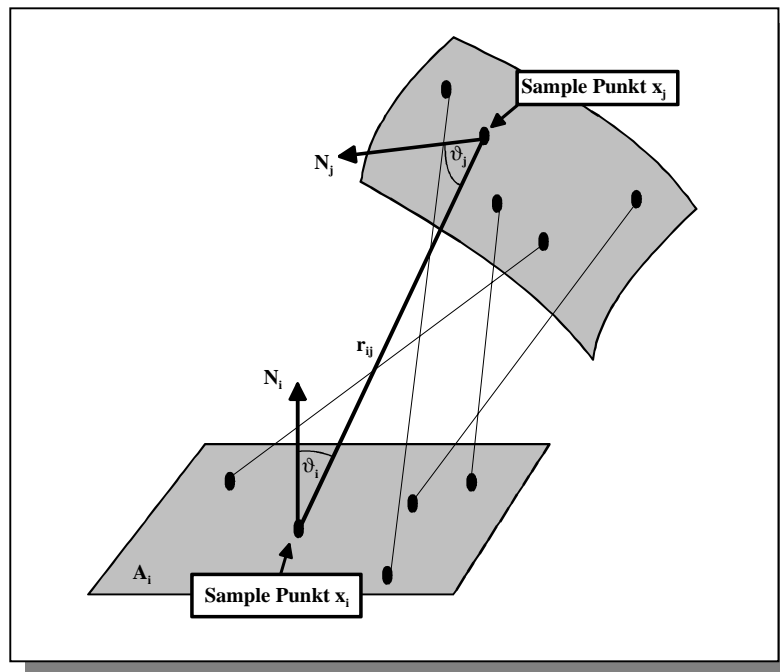


Abb. 3.15: Monte Carlo Integration des *Patch-zu-Patch* Formfaktor-Integrals.

Das doppelte Flächenintegral lässt sich deutlich exakter approximieren, indem sowohl Sample-Punkte auf der Fläche A_i als auch der Fläche A_j zufällig verteilt werden. Zwischen den Punkten auf den beiden Flächen werden zufällige Paare bestimmt, zwischen denen dann ein einfacher *Punkt-zu-Punkt* Formfaktor berechnet wird (siehe Abb. 3.15).

Die relative Sichtbarkeit eines solchen Punktepaars lässt sich hierbei sehr effizient mittels eines Strahles überprüfen, der durch die beiden Punkte verläuft. Dieser Strahl wird dann auf Schnittpunkte mit den Objekten der Szene überprüft.

In Algorithmus 3.1 ist Pseudo-Code für die Ermittlung eines *Patch-zu-Patch* Formfaktors F_{ij} auf Basis der Monte Carlo Integration angegeben.

```

 $F_{ij} = 0;$ 
for  $k = 1$  to  $n$  do
  wähle einen zufällig auf der Fläche  $i$  positionierten Punkt  $\bar{x}_i$ ;
  wähle einen zufällig auf der Fläche  $j$  positionierten Punkt  $\bar{x}_j$ ;
  überprüfe, ob  $\bar{x}_i$  und  $\bar{x}_j$  gegenseitig sichtbar sind;
  if (sichtbar)
    berechne  $r^2 = (|\bar{x}_i - \bar{x}_j|)^2$ ;
    berechne  $\cos \vartheta_i = \bar{r}_{ij} \cdot \bar{N}_i$ ;
    berechne  $\cos \vartheta_j = \bar{r}_{ji} \cdot \bar{N}_j$ ;
    berechne  $\Delta F = \frac{\cos \vartheta_i \cos \vartheta_j}{\pi r^2 + A_j / n}$ ;
    if ( $\Delta F > 0$ )  $F_{ij} = F_{ij} + \Delta F$ ;
  end if
end for
 $F_{ij} = F_{ij} A_j / n$ ;

```

Algorithmus 3.1: Monte Carlo Integration des doppelten Flächenintegrals für *einen Patch-zu-Patch* Formfaktor F_{ij} (alle Vektoren sind normierte Einheitsvektoren).

3.2 Progressive-Refinement-Radiosity

Neben der Erzeugung von möglichst realitätsnahen Bildern ist die Verwendbarkeit globaler Beleuchtungsverfahren in interaktiven Anwendungsszenarios ein weiteres wichtiges Ziel. Durch die Berechnung aller n^2 Formfaktoren in einem Preprocessing und dem vollständigen Lösen des Gleichungssystems (3.5) kann der ursprüngliche *Full-Matrix-Ansatz* zur Berechnung einer Radiosity-Lösung den Geschwindigkeitsanforderungen interaktiver Anwendungen jedoch nicht gerecht werden.

Von Bergman u.a. [BFGS86] wurde 1986 das Prinzip des *Rendering by Adaptive Refinement* eingeführt. Bei diesem Bildgenerierungsprinzip werden zunächst einfache Versionen des Bildes angezeigt, während weitere Bilder mit steigendem Realitätsgrad berechnet werden, bis die endgültige Version erzeugt ist. Hierbei ist es wesentlich, daß eine einzige Rendering-Operation zur Verfügung steht, der durch wiederholte Anwendung schrittweise verbesserte synthetische Bilder generiert.

Progressive-Refinement-Radiosity beruht exakt auf diesem Prinzip [CCWG88]. Es werden schrittweise verbesserte Radiosity-Lösungen berechnet, um diese Zwischenlösungen für den Bildgenerierungsprozeß zu verwenden, bevor eine konvergierte Lösung zur Verfügung steht. Ein weiterer wesentlicher Vorteil der Progressive-Refinement-Methode besteht darin, daß nur die n Formfaktoren einer einzelnen Zeile des Gleichungssystems zur Durchführung eines Schrittes des iterativen Lösungsverfahrens berechnet werden müssen. Somit entfällt der bei der Full-Matrix-Methode zur Speicherung der Formfaktoren notwendige Speicherbedarf von $O(n^2)$. Die Formfaktoren werden bei der Progressive-Refinement-Methode nur bei Bedarf berechnet, wodurch pro Iterationsschritt ein Speicherplatzbedarf von $O(n)$ entsteht.

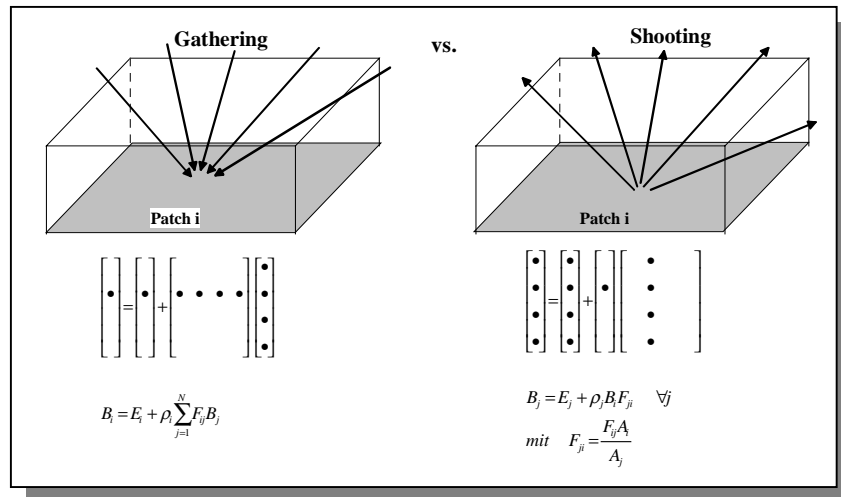


Abb. 3.16: Gathering- versus Shooting-Verfahren.

Das bei der Full-Matrix-Methode verwendete Jacobi- oder Gauss-Seidel- Iterationsverfahren geht bei der Lösung des Gleichungssystems zeilenweise vor. Die Berechnung der i -ten Zeile liefert hierbei einen Schätzwert für den Radiosity-Wert von Patch i , welcher auf den aktuellen Schätzwerten der übrigen Patch-Radiosities der Umgebung beruht. Somit wird die Radiosity von Patch i bestimmt, indem die spezifischen Ausstrahlungen (Radiosities) der übrigen Patches der Umgebung *eingesammelt* werden. Aus diesem Grund wird diese Methode als *Gathering-Methode* bezeichnet.

Der Gathering-Prozeß läßt sich umkehren. Hierbei wird berechnet, wieviel von der gesamten Strahlungsleistung $B_i A_i$ eines bestimmten Patches i jeweils an die übrigen Patches j ($i \neq j$) der Umgebung abgegeben wird. Diese Vorgehensweise wird als *Shooting-Methode* bezeichnet, da die Energie von Patch i in die Umgebung verschossen wird.

Die Strahlungsleistung, die eine Fläche A_j von A_i erhält, ist $F_{ij} B_i A_i$. Der Anteil, den A_j davon ideal diffus reflektiert ist $\rho_j F_{ij} B_i A_i$. Der Beitrag den A_j zur spezifischen Ausstrahlung von A_j leistet ist demnach

$$\rho_j \frac{A_i}{A_j} F_{ij} B_i . \quad (3.26)$$

Für jedes Patch der diskretisierten Umgebungsbeschreibung wird bei der Shooting-Methode während des iterativen Lösungsprozesses sowohl die aktuelle spezifische Ausstrahlung, als auch die durch das Patch bisher *nicht verschossene spezifische Ausstrahlung* (in W/m^2) gespeichert. Initial besitzen nur primäre Lichtquellenpatches eine nicht verschossene spezifische Ausstrahlung, die größer als Null ist (Eigenemission). In einem iterativem Prozeß wird pro Iteration die nicht verschossenen spezifische Ausstrahlung eines Patches i (*Shooting-Patch*) an die übrigen Patches der Umgebung verteilt (*Shooting*). Hierbei lassen sich beispielsweise mit Hilfe der Hemicube-Technik die Formfaktoren F_{ij} zu allen sichtbaren Patches j bestimmen (siehe Abschnitt 3.1.2.3) und die entsprechenden Energiemengen entsprechend Gleichung (3.26) verteilen (siehe auch Abb. 3.16). Hat ein Shooting-Patch seine Energie verteilt, wird die nicht verschossene spezifische Ausstrahlung des Patches auf Null gesetzt. Wenn ein Patch j auf diese Weise Energie zugeteilt bekommt (empfängt), wird der entsprechende Beitrag sowohl auf die aktuelle spezifische Ausstrahlung des Patches addiert, als auch auf dessen

nicht verschossene spezifische Ausstrahlung. Das Verfahren wird iteriert, bis die gesamte nicht verschossene spezifische Ausstrahlung aller Patches eine gewisse Schranke unterschreiten.

Das Shooting-Verfahren ist in Algorithmus 3.2 als Pseudocode zusammengefaßt.

```

 $\forall j: \Delta B_i = E_i;$ 
 $\forall j: B_i = E_i;$ 
while (nicht konvergiert) do {
    wähle neues Shooting-Patch  $i$  mit maximalem  $\Delta B_i A_i$ ;
    berechne die Formfaktoren  $F_{ij}$  mit Hilfe eines Hemicubes über Patch  $i$ ;
    for (jedes Patch  $j$ ) do {
         $\Delta Radiosity = \rho_j \Delta B_i F_{ij} \frac{A_i}{A_j};$ 
         $\Delta B_j = \Delta B_j + \Delta Radiosity;$ 
         $B_j = B_j + \Delta Radiosity;$ 
    }
     $\Delta B_i = 0;$ 
}

```

Algorithmus 3.2: Das Progressive-Refinement-Verfahren.

Mit jedem Iterationsschritt nähern sich der Schätzwert der spezifischen Ausstrahlung eines Patches dem endgültigen Wert an. Der Wert ΔB_i ist die Differenz zwischen dem letzten und dem gegenwärtigen Schätzwert für die spezifische Ausstrahlung B_i eines Patches i , und gibt die zu jedem Zeitpunkt der Berechnungen die nicht verschossene spezifische Ausstrahlung des Patches an. Durch $\Delta B_i A_i$ ist somit die gesamte *nicht verschossene Strahlungsleistung* (in W) von Patch i gegeben. Die Reihenfolge, in welcher die Patches ihre Energie in die Umgebung verschießen, ist von großer Bedeutung für die Konvergenzgeschwindigkeit des Shooting-Verfahrens. Die endgültige Radiosity B_j eines Patches j besteht aus den Energiebeiträgen aller Patches der Umgebung. Dieser Wert wird durch das Shooting-Verfahren am schnellsten erreicht, wenn in jeder Iteration bei der Auswahl des nächsten Shooting-Patches jeweils das Patch mit der größten nicht verschossenen Strahlungsleistung gewählt wird. Somit wird in jeder Iteration das Patch i mit größtem $\Delta B_i A_i$ als Shooting-Patch ausgewählt.

Es ist zu beachten, daß das Progressive-Refinement-Verfahren nicht auf die Verwendung der Hemicube-Technik beschränkt ist (siehe [BRW89][WKEH89]). Es können alle in Abschnitt 3.1.2 aufgeführten Techniken zur Formfaktorbestimmung eingesetzt werden, um die Qualität der berechneten Radiosity-Lösungen zu steigern.

3.3 Netzgenerierung (Meshing)

Die Berechnung exakter Radiosity-Lösungen ist ein Prozeß, welcher ein grundlegendes Verständnis der Radiosity-Methode auf Seiten der Anwender voraussetzt. Diese Tatsache bezieht sich insbesondere auf die Erzeugung initialer Polygonnetze, die als Eingabe für die Radiosity-Berechnungen dienen. Es ist zu beachten, daß die mit Standard-Modellierungssoftware er-

zeugten Modelle *nachbehandelt* werden müssen, um den geometrischen und topologischen Anforderungen des Radiosity-Verfahrens zu genügen [BMSW91]. Somit ist es notwendig *qualitativ gute* Polygonnetze in einer separaten initialen Netzgenerierungs-Phase (*initiales Meshing*) zu erzeugen. Diese initiale Netzgenerierung wird vor den eigentlichen Radiosity-Berechnungen durchgeführt.

Zusätzlich zur initialen Netzgenerierung muß eine adaptive Netzverfeinerung (*adapitives Meshing*) zur Laufzeit der Radiosity-Berechnungen durchgeführt werden, sobald eine Unstetigkeit der Beleuchtungsfunktion einer Objektoberfläche erkannt wird. Eine solche Unstetigkeit liegt beispielsweise vor, wenn eine Schattengrenze auf einer Oberfläche verläuft. In diesem Fall werden die Polygone, welche die entsprechende Oberfläche beschreiben, adaptiv verfeinert, um den Verlauf der Schattengrenze (und somit die Beleuchtungsfunktion) genauer approximieren zu können. Diese adaptiven Verfeinerungen werden durch Eingabeparameter gesteuert, die von Anwendern der Radiosity-Methode auf *sinnvolle* Werte gesetzt werden müssen, um qualitativ gute Ergebnisse zu erzielen. Das Setzen dieser Eingabeparameter erfordert jedoch ein grundlegendes Verständnis des Simulationsprozesses.

In Abb. 3.17 ist aufgeführt, wie diese verschiedenen Meshing-Phasen in den Lösungsprozeß der Radiosity-Methode integriert sind.

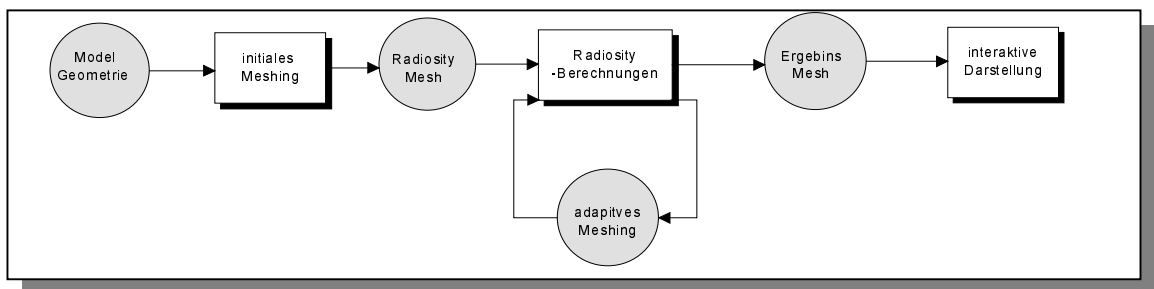


Abb. 3.17: Die Radiosity-Pipeline.

3.3.1 Initiale Netzgenerierung

Während der initialen Netzgenerierung muß das Eingabennetz in eine Form gebracht werden, die es einerseits ermöglicht fehlerfreie Radiosity-Berechnungen durchzuführen. Andererseits müssen Anforderungen erfüllt werden, deren Vernachlässigung zu deutlich sichtbaren Fehlern bei dem Bildgenerierungsprozeß im Anschluß an die Radiosity-Berechnungen führen. Demzufolge lassen sich die Anforderungen an ein *gutes* Radiosity-Netz in drei Kategorien einteilen:

- Allgemeine Anforderungen an die Geometrie des Modells
- Anforderungen bedingt durch den Radiosity-Simulationsprozeß
- Anforderungen bedingt durch den Bildgenerierungsprozeß

3.3.1.1 Allgemeine Anforderungen an die Geometrie des Modells

Die Einhaltung der allgemeinen Anforderungen an die Geometrie des Eingabemodells ist eine grundlegende Voraussetzung für fehlerfreie Radiosity-Berechnungen.

Zunächst muß ein Modell aus separaten Oberflächen modelliert sein. Eine Oberfläche ist ein zusammenhängendes Gebiet, das homogene Materialeigenschaften aufweist und dessen Flächennormalen an verschiedenen Punkten der Oberfläche nicht voneinander abweichen (d.h. keine stark gekrümmten Oberflächen). Weiterhin muß das Netz in eine Menge von konvexen Flächenelementen zerlegt werden. Hierbei ist es wesentlich, daß jedes dieser Flächenelemente nur in eine Richtung sichtbar ist. Bei diesen *zulässigen* Flächenelementen wird Energie (spezifische Ausstrahlung in W/m^2) nur in eine Richtung abgegeben. Ein *unzulässiges* Flächenelement würde Energie in beide Richtungen abstrahlen, was bei der Radiosity-Simulation nicht gewollt ist. Hierbei ist sehr wichtig, daß auf die Konsistenz der Normalen der Flächenelemente einer Oberfläche geachtet wird, da diese wesentlichen Einfluß auf die Formfaktorbestimmung haben. Die Nichteinhaltung dieser Konsistenzanforderung führt unweigerlich zu einer fehlerhaften Radiosity-Lösung. Weiterhin ist es wichtig, daß im Fall von koplanaren Flächen, die in der Szene enthalten sind, die Flächen entfernt werden, die für den anschließenden Bidgerierungsprozeß nicht wichtig sind, da sie verdeckt werden. Dieses ist zum Beispiel der Fall, wenn ein Stück Papier auf einem Tisch liegt. In diesem Fall kann der verdeckte Teil der Oberfläche des Tisches aus dem Eingabernetz entfernt (ausgeschnitten) werden.

3.3.1.2 Anforderungen bedingt durch den Radiosity-Simulationsprozeß

Um bei der Simulation der Verteilung der Lichtenergie Ergebnisse mit möglichst geringen Fehlern zu berechnen, müssen weitere Anforderungen durch das Eingabernetz erfüllt werden, die in direktem Zusammenhang mit dem Lösungsverfahren stehen. Hierbei ist zu berücksichtigen, daß die Generierung einer exakten Radiosity-Lösung sowohl die teilweise Rekonstruktion (*Sampling*) als auch die Interpolation einer initial unbekannter Funktionen bedingt, welche die Beleuchtung der Oberflächen einer Szene beschreibt [Kay86]. Hierbei basiert der Interpolationsprozeß auf der Topologie des Netzes, während die Rekonstruktion der Funktion verbunden ist mit der Anzahl von diskreten Flächenelementen sowohl auf den Empfänger- als auch den Senderoberflächen von Strahlungsenergie.

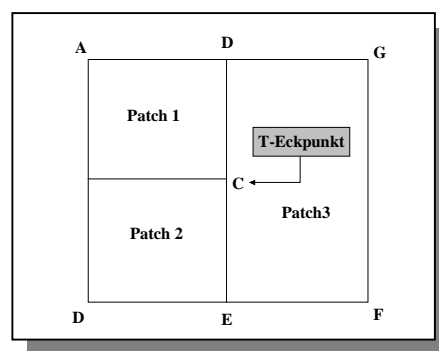


Abb. 3.18: Knoten C bildet für die 3 benachbarten Patches einen T-Eckpunkt.

Um eine kontinuierliche Interpolation der spezifischen Ausstrahlung einer Objektoberfläche zu ermöglichen, muß sich das Netz aus einfachen Flächenelementen wie Dreiecken oder Vierecken zusammensetzen, die eine schnelle und gute lokale Interpolation ermöglichen. Es ist jedoch zu beachten, daß zwischen benachbarten einfachen Flächenelementen einer Oberfläche keine *T-Knoten* (siehe Abb. 3.18) zugelassen sind, da ansonsten die Approximation einer kontinuierlichen Oberflächenbeleuchtung und deren Visualisierung nur eingeschränkt möglich ist. An den Grenzen zu benachbarten Oberflächen dürfen jedoch T-Knoten auftreten,

da an diesen Stellen die Beleuchtungsfunktionen auch ohne T-Knoten eine Unstetigkeit aufgrund abweichender Materialeigenschaften oder Oberflächennormalen aufweisen.

Eine weitere Anforderung setzt die *Wohlgeformtheit* von Flächenelementen voraus. Diese besagt, daß das Höhen-/Breitenverhältnis (*Aspect Ratio*) eines Flächenelementes möglichst nahe bei dem Wert Eins liegen soll. In diesem Fall sind die Formfaktorberechnungen exakter [BRW89].

Um eine gute Lösungsqualität zu erzielen, die in annehmbarer Zeit berechenbar ist, muß die Größe der Flächenelemente mit großer Vorsicht gewählt werden. Während des Sampling-Prozesses bei der Formfaktorberechnung muß eine ausreichende Dichte von Sample-Punkten auf den Sender-Patches (*Sender Node Density*) als auch der Empfänger-Patches (*Receiver Node Density*) gegeben sein. Sind die Patches zu groß, wird die Lösung ungenau [BRW89]. Andererseits wird bei kleinen Patchgrößen die Berechnungszeit in einen nicht akzeptablen Bereich gesteigert. Diese Anforderungen lassen sich jedoch nur bedingt vor Berechnungsbeginn erfüllen und müssen somit im Rahmen der adaptiven Substrukturierung im Verlauf der Berechnungen (siehe Abschnitt 3.3.2) berücksichtigt werden.

Eine der wichtigsten Anforderungen ist, daß Schnittflächen zweier Objekte explizit erkannt und behandelt werden müssen. Eine Mißachtung dieser Anforderung führt zu den am deutlichsten sichtbaren Fehlern bei den Radiosity-Berechnungen, den sogenannten *Shadow-* bzw. *Light-Leaks*.

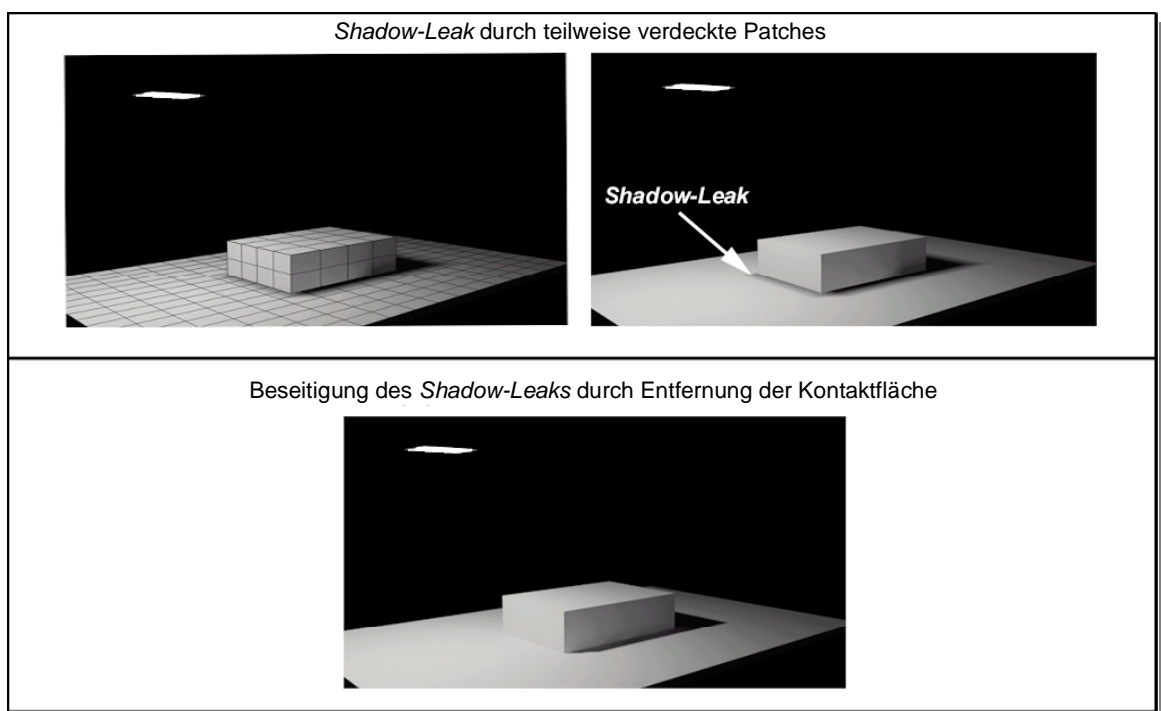


Abb. 3.19: An dem Rand der Kontaktfläche zwischen Quader und Bodenfläche entsteht ein *Shadow-Leak* (oben), welches durch Beseitigung der Kontaktfläche eliminiert wird (unten).

Ein *Shadow-Leak* läßt sich anhand einer Szene demonstrieren, in der ein Objekt auf einem anderen Objekt plaziert ist und beide Flächen von oben beleuchtet werden. Wenn nun die

Kontur der Schnittfläche der beiden Objekte nicht explizit bei der Netzgenerierung erzeugt und die entsprechende Kontaktfläche entfernt wurde, kann der Fall auftreten, daß Patches der unteren Oberfläche nur teilweise durch das obere Objekt verdeckt sind. Somit sind Eckpunkte eines Patches teilweise bzgl. der Lichtquelle verdeckt, während andere Eckpunkte Energie empfangen. Wenn nun eine lokale Interpolation der Eckpunkt-Radiosities durchgeführt wird, *fließt* der Schatten unter dem verdeckenden Objekt in den sichtbaren Bereich (siehe Abb. 3.19). Dieses Problem läßt sich zwar durch adaptive Verfeinerung (Abschnitt 3.3.2) an den Schnittgrenzen bedingt beheben, wodurch jedoch extreme Substrukturierungstiefen - und somit große Anzahlen von Patches - erzeugt werden. Dieses wirkt sich sowohl auf den Berechnungsaufwand der Radiosity-Methode als auch auf die Darstellungsgeschwindigkeit der berechneten Lösung mittels interpolativer Schattierung aus.

Ein *Light-Leak* bedeutet, daß in den auf Basis der Radiosity-Lösung generierten Bildern hell beleuchtete Regionen zu erkennen sind. Diese können eigentlich keine Lichtenergie von einer Lichtquelle empfangen, da ein verdeckendes Objekt die direkte und indirekte Lichteinstrahlung verhindert. Ein *Light-Leak* hat die gleiche Ursache, wie das zuvor geschilderte *Shadow-Leak*. Da Patches existieren, die teilweise in einer beleuchteten Region und teilweise in einer nicht beleuchteten Region liegen, *fließt* durch ein interpolatives Schattierungsverfahren Lichtenergie aus dem beleuchteten Teil des Patches in den eigentlich verdeckten Teil (siehe Abb. 3.20). Sowohl *Lightleaking* als auch *Shadowleaking* wird verhindert, indem alle Kontaktflächen, die zwischen den Objekten der Szene im initialen Netz existieren, ausgeschnitten werden.

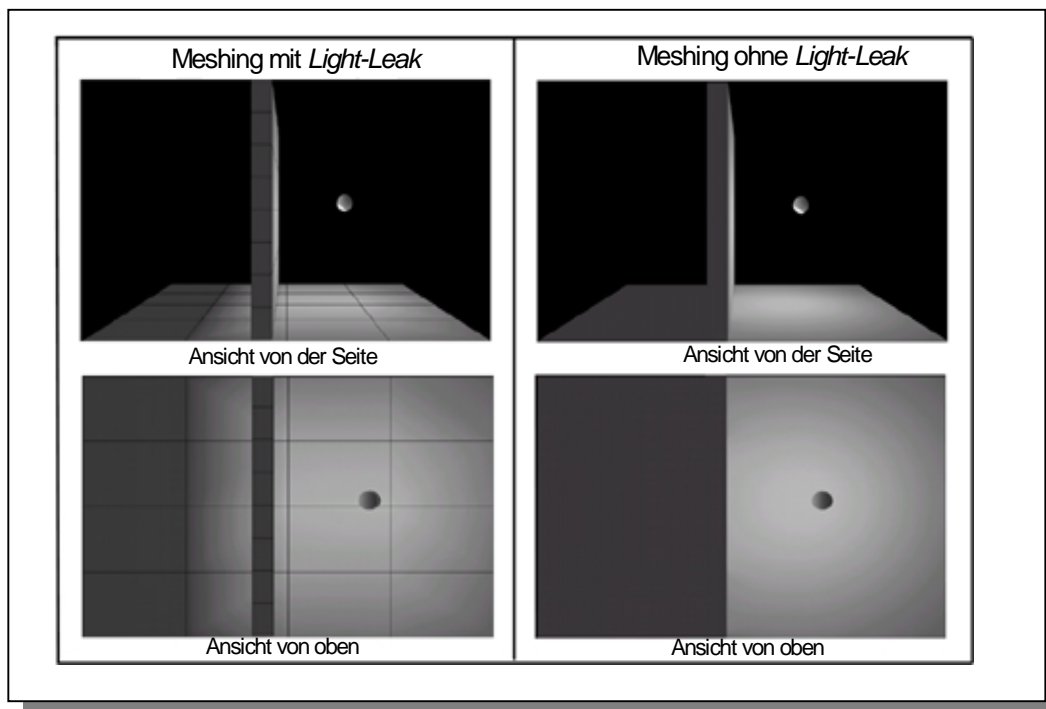


Abb. 3.20: Entstehung eines *Light-Leaks* durch Polygonnetze, die Kontaktflächen nicht explizit berücksichtigen.

3.3.1.3 Anforderungen bedingt durch den Bildgenerierungsprozeß

Der Bildgenerierungsprozeß stellt eigene Anforderungen an das während der Radiosity-Berechnungen evtl. adaptiv verfeinerte Lösungs-Netz. Auch hier dürfen, wie schon in 3.3.1.2 gefordert, keine T-Knoten auftreten, da diese zu deutlich sichtbaren Interpolationsfehlern bei der Darstellung der Lösung führen (siehe Abb. 3.21).

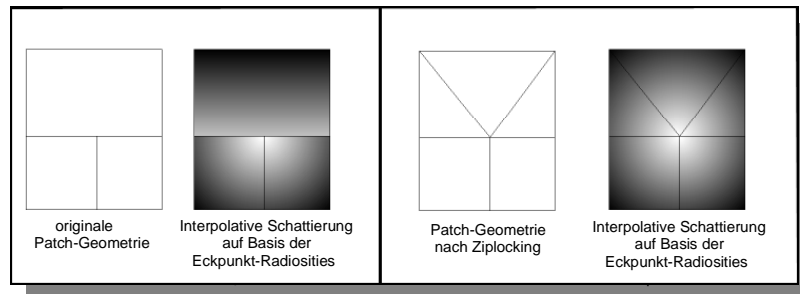


Abb. 3.21: Interpolative Schattierung vor Ziplocking (links) und nach Ziplocking (rechts).

Abhilfe schafft hierbei die Technik des *Ziplocking*, welche einen T-Knoten durch Unterteilen des größeren, an den T-Knoten angrenzenden Patches, *verankert*. Das Ziplocking kann hierbei sowohl während der Radiosity-Berechnung im Rahmen der adaptiven Verfeinerung von Patches (Substrukturierung), als auch in einem Postprocessing-Schritt vor der Darstellung der Lösung geschehen [BMSW91].

3.3.2 Adaptive Verfeinerung von Patches

Die Diskretisierung der Oberflächen von Objekten in kleine polygonale Elemente (Patches) hat einen großen Einfluß auf die Qualität der Radiosity-Berechnungen. Eine Unterteilung der Oberflächen in viele kleine Patches verursacht einerseits einen höheren Rechenaufwand bei der Bestimmung der Patch-Radiosities. Andererseits wird die diffuse Verteilung der Lichtenergie mit einer wesentlich höheren Genauigkeit bestimmt, wodurch auch die Qualität der im Anschluß an die Radiosity-Berechnungen generierten Bilder sichtbar besser ist.

Die ersten Radiosity-Algorithmen basierten auf einer Diskretisierung der Szene in annähernd gleich große Patches (*uniformes Meshing*), wobei die Unterteilung bereits vor der Durchführung der eigentlichen Berechnungen vorgenommen wurde. Der Ansatz des uniformen Meshings birgt das Problem in sich, daß die lokalen Gegebenheiten der Beleuchtung einer Szene nicht berücksichtigt werden. Dieses führt an manchen Stellen der Szene zu einer unnötig feinen Diskretisierung, während an anderen Stellen eine zu grobe Netzstruktur vorliegt, um die tatsächliche Beleuchtungsfunktion hinreichend genau rekonstruieren zu können.

Als erstes wurde von Cohen u.a. [CGIB86] erkannt, daß die Verwendung gleich großer Flächenelemente zu unflexibel ist und die Größe eines Patches von dessen Lage innerhalb der Szene abhängig gemacht werden muß. Auch der in [CGIB86] beschriebene Ansatz ist nur bedingt zufriedenstellend, da es in den meisten Fällen nicht möglich ist, die Verteilung der Lichtenergie innerhalb einer Szene hinreichend genau a priori abzuschätzen. Dieses trifft insbesondere auf große Helligkeitsgradienten zu, wie sie bei der durch Lichtquellen verursachten Schattenbildung auftreten. Dieses Problem läßt sich jedoch durch eine adaptive Verfeinerung (Substrukturierung) im Verlauf der Radiosity-Berechnungen lösen [CCWG88]. Hierbei wird zunächst von einer initial sehr groben Unterteilung der Szene in Patches ausgegangen. Wenn

sich bei der Berechnung der spezifischen Ausstrahlung eines Patches herausstellt, daß diese über die Fläche betrachtet einer Schwankung unterliegt, die einen vorgegebenen Grenzwert überschreitet, so wird das Patch in kleinere Elemente (*Sub-Patches*) unterteilt. Dieser Substrukturierungsprozeß läßt sich rekursiv fortsetzen, bis die Beleuchtung über einem Sub-Patch als annähernd konstant betrachtet werden kann oder eine untere Schranke für die Größe eines Elementes erreicht wird. Für jedes Patch der Szene kann durch diese Vorgehensweise eine Hierarchie von Sub-Patches erzeugt werden, welche sich in einer Quadtree-Datenstruktur effizient verwalten läßt.

Der wesentliche Punkt bei der adaptiven Substrukturierung ist die Entscheidung, ob ein bestimmtes Flächenelement unter einer gegebenen Beleuchtungssituation weiter verfeinert werden soll oder ob die tatsächliche Beleuchtungsfunktion in der aktuellen Substrukturierungstiefe hinreichend genau approximiert wird.

Die Qualität eines Radiosity-Verfahrens ist in beträchtlichem Ausmaß von dem sogenannten *Substrukturierungskriterium* abhängig. Ein mögliches Kriterium ist die Überprüfung des Gradienten der Beleuchtungsfunktion entlang der Kanten eines Flächenelementes. Überschreitet dieser Gradienten einen vorgegebenen Grenzwert, so wird das Patch in kleinere Sub-Patches unterteilt, für welche eigene Radiosity-Werte bestimmt werden. Die Ursache für einen zu großen Gradienten kann beispielsweise eine Schattengrenze sein, die innerhalb der Patchfläche verläuft. Durch eine weitere Substrukturierung des Patches wird der Schattenverlauf genauer eingegrenzt. Somit wird eine Verfeinerung der Substrukturierung nur an den Stellen vorgenommen, wo dieses notwendig ist (siehe Abb. 3.22).

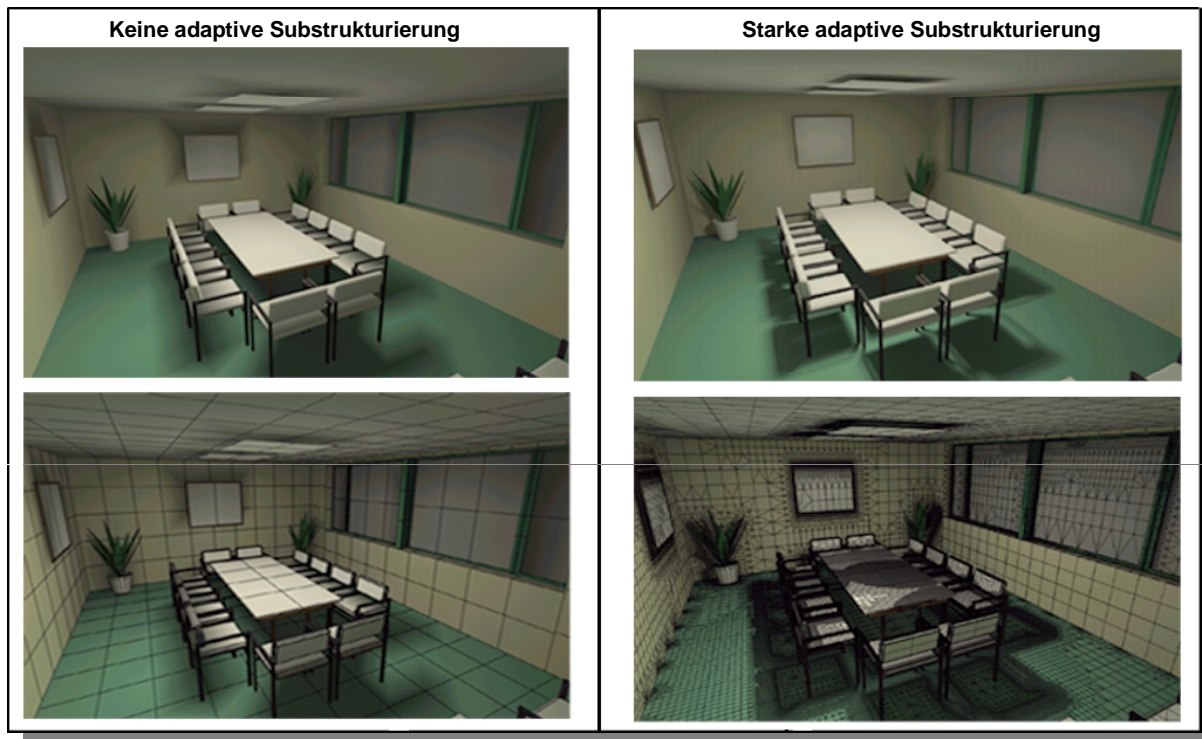


Abb. 3.22: Verschiedene Stufen der adaptiven Substrukturierungstiefe und die dadurch verbesserte Annäherung von Schattengrenzen.

Seien dazu $v_1, v_2, v_3, v_4 \in \mathbb{R}^3$ die Eckpunkte eines viereckigen Patches i und $B(v_1), B(v_2), B(v_3), B(v_4)$ die zugehörigen Eckpunkt-Radiosities, die entweder direkt oder mit der in Abschnitt 3.4 beschriebenen Methode berechnet werden. Die betragsmäßig größte Abweichung der Radiosities entlang der Patch-Kanten kann dann als Approximation für den Gradienten dienen. Somit wird Patch i genau dann verfeinert, wenn

$$\frac{|B(v_i) - B(v_j)|}{\|v_i - v_j\|} > \varepsilon \quad (3.27)$$

für ein Paar i, j mit $1 \leq i < j \leq 4$ erfüllt ist, wobei die Schranke ε geeignet vorgegeben sein muß.

Es ist zu beachten, daß es noch weitere Substrukturierungskriterien gibt, die sich deutlich bzgl. Berechnungskomplexität und Qualität unterscheiden. Auf weitere effiziente Techniken wird zu einem späteren Zeitpunkt in dieser Arbeit noch genauer eingegangen.

Die adaptive Substrukturierung läßt sich auf einfache Art und Weise in das Progressive-Refinement-Verfahren integrieren (siehe Algorithmus 3.3).

```

forall  $j$ :  $\Delta B_i = E_i$ ;
forall  $j$ :  $B_i = E_i$ ;
while (nicht konvergiert) do {
    wähle neues Shooting-Patch  $i$  mit maximalem  $\Delta B_i A_i$ ;
    berechne die Formfaktoren  $F_{ij}$  mit Hilfe eines Hemicubes über Patch  $i$ ;
    for (jedes Patch  $j$ ) do {
         $\Delta R_{adiosity} = \rho_j \Delta B_i F_{ij} \frac{A_i}{A_j}$ ;
         $\Delta B_j = \Delta B_j + \Delta R_{adiosity}$ ;
         $B_j = B_j + \Delta R_{adiosity}$ ;
    }
    Berechne auf Basis der bestimmten Patch-Radiosities Radiosity-Werte
    an den Eckpunkten des Netzes (siehe 3.4);
    Werte für alle Patches das durch Gleichung (3.27) gegebene Substrukturie-
    rungskriterium aus und Unterteile ggf. die entsprechenden Patches in
    kleinere Elemente;
    Wenn eine Substrukturierung durchgeführt wurde, verteile die Energie des
    Shooting-Patches erneut bzgl. der erzeugten Elemente;

     $\Delta B_i = 0$ ;
}

```

Algorithmus 3.3: Das Progressive-Refinement-Verfahren mit adaptiver Verfeinerung.

Auch mit adaptiver Verfeinerung können Schattenkanten nur angenähert werden. Zur Erzeugung wirklich scharfer Schattenkanten müssen die Schattengrenzen identifiziert und die

Patch-Kanten danach ausgerichtet werden. *Discontinuity Meshing* ist ein Verfahren, das Unstetigkeiten in der Beleuchtungsfunktion einer Oberfläche (wie z.B. Schattenkanten) bei dem Netzgenerierungsprozeß berücksichtigt. Es gibt Unstetigkeiten verschiedener Ordnung in der Beleuchtungsfunktion einer Oberfläche, wobei eine Unstetigkeit für den Betrachter um so auffälliger ist, je geringer deren Ordnung ist. Eine Unstetigkeit nullter Ordnung ist ein Sprung in einer Beleuchtungsfunktion (zumeist aufgrund direkter Einstrahlung von Lichtenergie). Unstetigkeiten höherer Ordnung entstehen, wenn mehrere Patches an der Schattenbildung beteiligt sind, und die Reflexionen mehrerer Patches mit einbezogen werden. Verfahren, die derartige Unstetigkeitsstellen suchen und die Patches danach unterteilen, sind extrem aufwendig. In den meisten Fällen werden nur Unstetigkeiten nullter und erster Ordnung betrachtet. Die Methode führt bei scharfen Schattenkanten jedoch zu sehr guten Ergebnissen. Eine erschöpfende Abhandlung über die Thematik des *Discontinuity Meshing* ist in [CW93] zu finden.

3.4 Darstellen der Radiosity-Lösung

Zur Bildgenerierung auf Basis einer Radiosity-Lösung bietet sich das iterative Gouraud-Schattierungsverfahren an (siehe 2.3.1). Dieses ist bei modernen Grafikkarten, wie sie in den meisten PC's vorzufinden sind, in Hardware realisiert, wodurch selbst auf diesen kleinen Systemen Online-Begehungen von Szenen mit moderater Komplexität möglich sind.

Die Intensitäten an den Eckpunkten der Patches der diskretisierten Szenenbeschreibung werden aus den zuvor bestimmten Patch-Radiosities ermittelt, da iterative Schattierungsverfahren diese Werte benötigen. Von Cohen und Greenberg [CG85] wurde eine Methode vorgestellt, um Eckpunkt-Radiosities aus den Patch-Radiosities zu bestimmen. Diese Methode berechnet die Radiosities eines inneren Eckpunktes einer diskretisierten Oberfläche als Durchschnitt der Radiosities der an die Eckpunkte angrenzenden Patches.

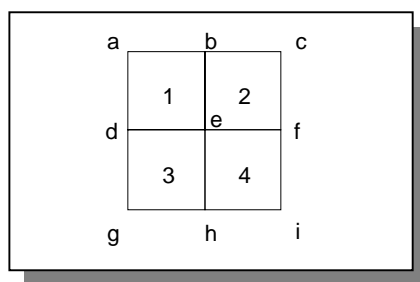


Abb. 3.23: Berechnung von Eckpunkt-Radiosities.

Somit ergibt sich der Radiosity-Wert B_e in Abb. 3.23 aus:

$$B_e = \frac{1}{2}(B_1 + B_2 + B_3 + B_4) \quad (3.28)$$

Um die Eckpunkt-Radiosities der auf dem Rand einer Oberfläche liegenden Eckpunkte zu ermitteln, muß darauf geachtet werden, daß nur die angrenzenden Patches betrachtet werden:

$$B_a = B_1 \quad (3.29)$$

und

$$B_b = \frac{1}{2}(B_1 + B_2). \quad (3.30)$$

Nachdem die Eckpunkt-Radiositäten berechnet wurden, können nach Umwandlung der radiometrischen bzw. photometrischen Werte in Farbwerte Bilder aus beliebigen Perspektiven mit Hilfe eines interpolativen Schattierungsverfahrens (z.B. [Gou71]) generiert werden, ohne daß eine erneute Radiosity-Berechnung für die Szene notwendig wird.

Es ist zu beachten, daß bei der Methode von Wallace u.a. [WKEH89] Radiosity-Werte direkt für Eckpunkte der Patches berechnet werden. Somit entfallen die Interpolationsfehler, die bei der Bestimmung von Eckpunkt-Radiositäten aus den berechneten Patch-Radiositäten auftreten können.

Wird die abschließende Bildgenerierung mittels einem Raytracing-Verfahren durchgeführt (siehe 3.5) bietet sich die bilineare Interpolation der Eckpunktradiositäten an, wie sie in [CW93] beschrieben ist.

3.5 Die Radiosity-Methode und spiegelnde Oberflächen

Die Bildgenerierung auf Basis der Radiosity-Methode ermöglicht die Berücksichtigung von Beleuchtungsphänomenen, die durch ideale diffuse Reflexion des Lichtes innerhalb einer Szene entstehen. Die Beschränkung der Reflexionseigenschaften von ideal diffuse Eigenschaften (Lambert-Reflektoren) stellt jedoch eine Einschränkung der in der Realität auftretenden Lichttransportpfade dar (siehe Kapitel 2). Um Bilder mit einem maximalen Grad an Realitätsnähe zu generieren, muß sowohl die diffuse als auch die spiegelnde Reflexion des Lichtes bei der globalen Beleuchtungssimulation berücksichtigt werden.

Wenn eine Szene aus spiegelnd und diffus reflektierenden Oberflächen besteht, ergibt sich die indirekte Beleuchtung eines Oberflächenpunktes als das Ergebnis von vier Mechanismen des Lichttransportes innerhalb von Szenen. Eine Klassifizierung dieser Transportmechanismen wurde in [Heck91] vorgenommen (siehe auch Kapitel 2). In der Vergangenheit wurden verschiedene Methoden eingeführt, welche versuchten, diese Mechanismen des Lichttransportes möglichst vollständig nachzubilden. Es muß hierbei zwischen Verfahren unterschieden werden, die vollständig unabhängig von der Radiosity-Methode sind (z.B. Monte Carlo Raytracing und bidirektionale Verfahren) [KK86] [Kay86] [WRC88] [AK90] sowie Verfahren, die als Erweiterungen der Radiosity-Methoden konzipiert wurden [WCG87] [SP89] [PSV90] [CRMT91].

Für diese Arbeit sind nur die auf der Radiosity-Methode basierenden *Mehrphasen-Methoden* von Bedeutung. Bei diesen Verfahren werden die Vorteile von Radiosity-Methoden mit denen von Raytracing-Verfahren kombiniert, um möglichst komplexe Lichttransportpfade zu simulieren. Im weiteren Verlauf dieses Abschnitts wird ein kurzer Überblick über die Zwei-Phasen-Methoden gegeben, mit deren Hilfe die Simulation von Lichttransportpfaden möglich ist, die weitestgehend alle Transportketten berücksichtigen (d.h. $L(D/S)^*E$ Lichttransportpfade).

3.5.1 Einfache Zwei-Phasen-Methode

Durch die Simulation der globalen Beleuchtung soll bei einem Betrachter von computergenerierten Bildern ein Eindruck von maximaler Realitätsnähe vermittelt werden. Aus diesem

Grund müssen alle Lichttransportpfade berücksichtigt werden, die einen signifikanten Einfluß auf das endgültige Bild haben. Das menschliche Auge erkennt sehr schnell Glanzlichter und Spiegelungen in realen Umgebungen. Diese sind offensichtlich von dem Standpunkt des Betrachters und dessen Blickrichtung abhängig. Somit ist die vom Betrachterstandpunkt unabhängige Beleuchtungssimulation mittels der Radiosity-Methode nicht ausreichend, da keine Spiegelungen und Glanzlichter bei der Bildgenerierung berücksichtigt werden. Eine Möglichkeit zur Simulation von Beleuchtungsphänomenen, die durch diffuse Reflexion und Spiegelung entstehen, besteht darin, die Vorteile der Radiosity-Methode mit denen von Raytracing-Verfahren zu kombinieren. Die Radiosity-Methode wird hierbei verwendet, um den Austausch von Lichtenergie zwischen diffus reflektierenden Oberflächen unabhängig von einem Betrachterstandpunkt zu simulieren. Ein von der Perspektive des Betrachters abhängiges Verfahren (z.B. klassisches Raytracing [Whi80]) kann in einer zweiten Phase dazu verwendet werden, um Glanzlichter und spiegelnde Effekte zur Lösung hinzuzufügen. Wenn klassisches Raytracing verwendet wird, können die Berechnungen der zweiten Phase vereinfacht werden, da zahlreiche Informationen schon in der ersten Phase durch die Radiosity-Simulation berechnet wurden. Bei der Durchführung der Raytracing-Berechnungen kann vollständig auf die Generierung von *Schattenstrahlen* bzgl. der Lichtquellen in der Szene verzichtet werden, um festzustellen, ob auf einem Oberflächenpunkt von den Lichtquellen direkt abgestrahlte Lichtenergie aus diesen Richtungen einfällt. Diese Energieanteile wurden schon in der Radiosity-Phase berücksichtigt. Da die Erzeugung von Schattenstrahlen typischerweise den Hauptaufwand bei der rekursiven Strahlenverfolgung ausmacht, wird die Berechnungszeit der zweiten Phase durch diese Vereinfachung deutlich verkürzt. Das beim Raytracing verwendete Beleuchtungsmodell wird dahingehend modifiziert, daß der spiegelnde Anteil einer Pixel-Intensität durch Verfolgung eines ideal spiegelnd reflektierten Strahles gewonnen wird, während der diffuse Anteil auf Basis der zuvor berechneten Radiosity-Lösung bestimmt werden kann (z.B. bilineare Interpolation der Eckpunkt-Radiosities eines getroffenen Patches).

Es ist zu beachten, daß bei der Durchführung der zweiten Phase auch ein einfacheres, teilweise durch Hardware unterstütztes Verfahren, wie z.B. die Phong-Schattierung verwendet werden kann, um Glanzlichter zu der Radiosity-Lösung hinzuzufügen [PSV90]. Die auf diese Art und Weise generierten Bilder enthalten jedoch keine Spiegelungen.

Das beschriebene einfache Zwei-Phasen-Verfahren basiert auf einer strikten Trennung der diffusen und spiegelnden Beleuchtungsanteile von Punkten auf Oberflächen einer Umgebungsbeschreibung. Diese Vorgehensweise entspricht jedoch nicht der physikalischen Realität. Lichtenergie ist weder spiegelnd noch diffus. Diese Ausdrücke beschreiben lediglich verschiedene Reflexionsmodelle und nicht verschiedene Arten von Energie. So kann Lichtenergie eine verschiedene Anzahl von wechselnden diffusen und nicht diffusen Reflexionen unterzogen werden, bevor sie von einer Oberfläche absorbiert wird. Der Anteil an der Beleuchtung eines Oberflächenpunktes, welcher durch diffuser Reflexionen von anderen Oberflächen abgegeben wurde, kann nicht unabhängig von dem durch spiegelnde Reflexion beigesteuerten Anteil berechnet werden. Es sind vielmehr Wechselwirkungen dieser beiden Reflexionsarten zu berücksichtigen. Wenn diese physikalische Tatsache in Betracht gezogen wird, ist die Unzulänglichkeit der einfachen Zwei-Phasen-Methode sofort ersichtlich. So ist es mit dieser Methode nicht möglich, Transportpfade der Art LS^+DE zu berechnen. Wenn während der Raytracing-Phase vom Auge des Betrachters aus zunächst eine diffus reflektierende Oberfläche gefunden wird, sind an diesem Oberflächenpunkt keine Informationen verfügbar, aus welcher Richtung von den anderen Oberflächen der Umgebung spiegelnd reflektierte Energie einfällt. Diese Informationen könnten nur gewonnen werden, wenn wie bei dem Monte Carlo Raytracing die gesamte Hemisphäre über dem gefundenen Oberflächenpunkt abgetastet würde. Dieses ist jedoch ein zu kostspieliger Prozeß. Aber gerade die Transportpfade des Typs

LS^+DE verursachen wesentliche Beleuchtungseffekte, welche in der Realität häufig vorkommen (Lichtenergie, die durch Spiegelung auf eine diffus reflektierende Oberfläche gelenkt wird). Dieses wird in Abb. 3.24 verdeutlicht, wo Licht über einen Spiegel auf die Rückseite des Würfels reflektiert wird und diesen somit indirekt beleuchtet.



Abb. 3.24: Berücksichtigung von Lichttransportpfaden unterschiedlicher Komplexität bei der Bildgenerierung.

Die Schlußfolgerung lautet somit, daß zwar sehr realistische Bilder mit Hilfe der einfachen Zwei-Phasen-Methode generiert werden können, wesentliche physikalische Eigenschaften des Lichttransportes jedoch vernachlässigt werden. Somit besteht ein Bedarf an Methoden, die eine möglichst vollständige Simulation der in realen Umgebungen auftretenden Lichttransportpfade zulassen.

3.5.2 Vollständige Zwei-Phasen-Methode

Die einfache Zwei-Phasen-Methode kann dahingehend erweitert werden, daß während der ersten Berechnungsphase (Radiosity-Berechnungen) gerade die Beleuchtungseffekte berücksichtigt werden, welche durch das in Abschnitt 3.5.1 beschriebene einfache Zwei-Phasen-Verfahren vernachlässigt wurden. Dieses ermöglicht das 1986 von Rushmeier eingeführte Konzept *der erweiterten Formfaktoren* [Rush86]. Ein erweiterter Formfaktor beschreibt hierbei die Energiemenge, die von einem Patch der Umgebung abgestrahlt wird und nach einer beliebigen Anzahl von spiegelnden Reflexionen von einem anderen Patch empfangen wird (siehe Abb. 3.25 - eine Spiegelung über Spiegelfläche b). Das von Rushmeier vorgestellte Verfahren ermöglichte die Berechnung eines erweiterten Formfaktor mit Hilfe der Hemicube-Methode. Hierbei wird eine spiegelnde Oberfläche als Fenster in eine virtuelle Welt betrachtet. Diese *Spiegelwelt* besteht aus der an Ebene der Spiegelfläche gespiegelten Szene. Die so berechneten erweiterten Formfaktoren beschreiben den Energietransfer von einem Sender-Patch zu einem Empfänger-Patch über eine planare, ideal spiegelnde Transferfläche. Dieses Verfahren wurde 1987 erstmalig von Wallace u.a. [WCG87] im Rahmen einer Zwei-Phasen-Methode zur Simulation von Lichttransportpfaden des Typs $L(D^*/D^*SD^*)S^*E$ verwendet, wobei in dem Teilpfad D^*SD^* lediglich eine ideal spiegelnde Reflexion zugelassen ist. Von Sillion u.a. [SP89] wurde eine Methode eingeführt, welches erweiterte Formfaktoren mit Hilfe eines rekursiven Raytracing-Verfahrens berechnet. Diese Vorgehensweise ermöglicht die Berücksichtigung von mehreren spiegelnden Reflexionen der Lichtenergie (DS^*D Transportpfade).

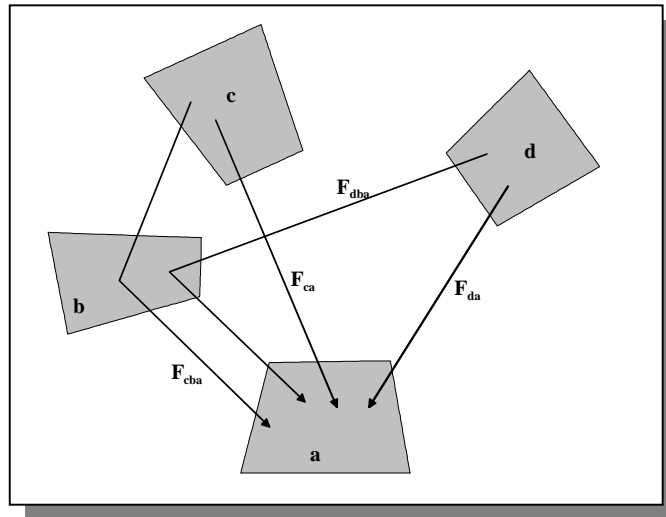


Abb. 3.25: Erweiterte Formfaktoren über Spiegelfläche *b*.

Zusammen mit den in [SAWG91] beschriebenen Erweiterungen ist durch eine Zwei-Phasen-Methode die Simulation von Lichttransportpfaden möglich, die sämtliche durch diffuse und spiegelnde Reflexion verursachten Richtungsänderungen eines Photons beschreiben (d.h. $L(D/S)^*E$).

Einige Autoren weisen darauf hin, daß die Detailgenauigkeit von Verfahren, welche die Beleuchtungsinformationen nur an bestimmten Stellen der Polygonnetze (z.B. Eckpunkte) berechnen, unzureichend bei der Wiedergabe von Beleuchtungseffekten sind, die durch primäre Lichtquellen entstehen [CRMT91]. Aus diesem Grund wird die mit Hilfe der Radiosity-Methode berechnete Lichtenergieverteilung nur zur Approximation der indirekten Beleuchtung verwendet. Die einer Oberfläche durch direkte Beleuchtung hinzugefügte Energieanteile werden hierbei vor der Weiterverarbeitung subtrahiert. Energieanteile, die durch direkte Beleuchtung entstehen, werden in einer blickpunktabhängigen Phase mittels Monte Carlo Pathtracing berechnet. Es ist zu beachten, daß mit Hilfe dieser Verfahren zwar sehr realistische Bilder generiert werden können, die Berechnungszeiten jedoch für interaktive Anwendungen nicht akzeptabel sind.

3.6 Abschließende Bemerkungen

In diesem Kapitel wurden die Grundlagen der Radiosity-Methode erläutert, die für das Verständnis der im weiteren Verlauf dieser Arbeit beschriebenen Parallelisierungsstrategien notwendig sind. Hierbei wurde insbesondere Wert auf die Beschreibung der verschiedenen Berechnungsmethoden für Formfaktoren gelegt, da diese einen entscheidenden Einfluß sowohl auf die Qualität der berechneten Lösungen, als auch auf die Effizienz der parallelen Berechnungen haben.

Auf zahlreiche Erweiterungen und Verbesserungen der Radiosity-Methode für statische Szenen wie z.B. *Hierarchisches Radiosity* und *Clustering* ist in diesem Kapitel nicht eingegangen worden, da diese Techniken zur Zeit keine Relevanz für die weiteren Betrachtungen innerhalb dieser Arbeit haben. Dem interessierten Leser sei zu dieser Thematik [CW93] und [SP94] empfohlen.

Kapitel 4

Radiosity in dynamisch veränderbaren Szenen

*Radiosity scenes are like museums,
you may look around, but do not touch anything!*

(Stefan Müller, 5th Eurographics Workshop on Rendering)

Der Anspruch an die visuelle Qualität von virtuellen Welten korreliert mit der Steigerung der Leistungsfähigkeit moderner Graphik-Workstations. Es reicht mittlerweile nicht mehr aus, eine mit Hilfe von lokalen Beleuchtungsmodellen beleuchtete Szene in Echtzeit virtuell zu durchlaufen (Walkthrough-Animation), da aufgrund fehlender globaler Beleuchtungseffekte und Schatten der Realitätseindruck äußerst gering ausfällt.

Zur Simulation der globalen Beleuchtung kann die Radiosity-Methode eingesetzt werden. Sie wird im Bereich der Architekturvisualisierung zu Präsentations- und Planungszwecken verwendet, um ansprechende Bilder von teilweise in der Realität noch nicht existierenden Umgebungen (Gebäuden) zu erzeugen. Ebenso bei der Erstellung von Animationen und virtuellen Spielumgebungen wird die Radiosity-Methode häufig zur Simulation globaler Beleuchtungseffekte verwendet. In virtuellen Welten tragen die mit Radiosity berechneten Schatten zur Verbesserung des Tiefeneindrucks und zur Erleichterung der Orientierung bei. Schließlich können mit dieser Methode exakte Beleuchtungsstärken berechnet werden, weshalb sie auch in der Beleuchtungstechnik verwendet wird. Der Wunsch nach Interaktion führt jedoch zum Konflikt mit diesem globalen Beleuchtungsverfahren, da die klassische Radiosity-Methode lediglich die Berechnungen von Radiosity-Lösungen in statischen Szenen unterstützt. Somit ist es einem Benutzer nicht erlaubt, Modifikationen an der Szene vorzunehmen, während der Berechnungsprozeß durchgeführt wird. Im Falle einer Szenenmodifikation muß der vollständige Lösungsprozeß von Beginn an wiederholt werden. Diese Neuberechnung ist im allgemeinen jedoch zu zeitaufwendig, weshalb mit Radiosity-Methoden beleuchtete Szenen in interaktiven Anwendungen mit Echtzeitanforderungen praktisch nicht zum Einsatz kommen. Zuerst wurde die Notwendigkeit der Berücksichtigung von Benutzerinteraktionen während der Beleuchtungssimulation von Baum u.a. [BWCG86], Buckalew u.a. [BF89] sowie Puech u.a. [PSV90] erkannt und explizit behandelt.

Verschiedene Anwendergruppen von interaktiven Bildgenerierungssystemen haben unterschiedliche Anforderungen hinsichtlich der Exaktheit der Beleuchtungssimulation und der Antwortzeiten des Systems im Anschluß an einer Modifikation der Szene [Zedl93]. Eine Übersicht über die verschiedenen Anforderungen ist in Tabelle 4.1 gegeben.

Anwendungsgebiet	Interaktionsgrad	Erforderliche Exaktheit der Lösung	Maximale Antwortzeit
<i>Virtual Reality</i>	hoch	niedrig	Bruchteile einer Sekunde
<i>Innen-Architektur / Design</i>	mittel	niedrig - mittel	wenige Sekunden
<i>Beleuchtungstechnik</i>	gering – mittel	sehr hoch (physikalisch korrekt)	Sekunden oder wenige Minuten
<i>Animationen</i>	hoch	niedrig - mittel	-

Tabelle 4.1: Unterschiedliche Anforderungen an ein interaktives Radiosity-System hinsichtlich des Zeitverhaltens und der Genauigkeit der Simulation.

Die typischen Modifikationen, die durch Anwender innerhalb einer virtuellen Umgebung vorgenommen werden, sind das Ändern von Oberflächeneigenschaften von Objekten (Materialattributen) sowie Veränderungen der Anordnung von Objekten (Szenengeometrie). Oberflächeneigenschaften von Objekten sind in diesem Kontext deren Reflexionseigenschaften (z.B. Farbe) oder Abstrahleigenschaften von Lichtquellen (z.B. Lichtquellen an- oder ausschalten). Im Fall von Geometrieänderungen innerhalb der Szene ändert sich die Form oder die Position von Objekten. Eine komplette Neuberechnung im Anschluß an eine Modifikation ist aus zeitlichen Gründen in interaktiven Anwendungen nicht möglich, da ein Anwender möglichst kurze Antwortzeiten erwartet. Es ist zu beachten, daß der größte Teil der möglichen Modifikationen nur sehr lokale Auswirkungen auf die Beleuchtungssituation innerhalb von komplexen virtuellen Umgebungen hat. Aus diesem Grund liegt es nahe, die Radiosity-Methode dahingehend zu erweitern, daß eine schon berechnete Radiosity-Lösung im Anschluß an eine Modifikation der Szene lediglich lokal korrigiert wird. Somit werden nur Teilbereiche der gesamten Szene bei der Berechnung von Korrekturwerten betrachtet. Die Radiosity-Werte von Patches innerhalb von Regionen, deren Beleuchtungssituation durch die Veränderung nicht betroffen sind, bleiben unverändert. Durch diese Vorgehensweise sind sehr kurze Systemantwortzeiten im Anschluß an eine Benutzerinteraktion möglich. In der Vergangenheit wurden verschiedene Ansätze zur Berechnung von Radiosity-Lösungen innerhalb von dynamisch veränderbaren Szenen eingeführt, deren Effizienz auf der Ausnutzung von lokalen Kohärenzen beruht.

4.1 Radiosity-Verfahren zur Unterstützung dynamischer Szenen

Von Baum u.a. [BWCG86] wurde das auf der Hemicube-Methode [CG85] basierende *Back-Buffer*-Verfahren vorgestellt. Das Verfahren ist für Berechnungen von Animationssequenzen entwickelt worden, in welchen sich Objekte auf vordefinierten Pfaden bewegen. Vor Beginn der eigentlichen Radiosity-Berechnungen werden die Sichtbarkeitsverhältnisse in dem statischen Teil einer gegebenen Szene ermittelt. Bei der Animationserzeugung werden für jedes Bild der Animation die Sichtbarkeitsverhältnisse für den dynamischen Teil der Szene neu ausgewertet. Im Anschluß daran werden sämtliche Radiosity-Werte durch ein erneutes Lösen des Gleichungssystems bestimmt. Somit nutzt das *Back-Buffer*-Verfahren die Tatsache aus, daß zuvor berechnete Informationen, welche durch eine Veränderung der Szene nicht betroffen sind, wiederverwendet werden können. Dieses bedeutet, daß durch Ausnutzung von Kohärenzen zwischen den einzelnen Bildern einer Animation die Berechnung aufeinanderfolgender Bilder stark beschleunigt wird. Hierbei korreliert der erreichbare Beschleunigungsfaktor mit der Anzahl der statischen Objekte in der Szene. Ein wesentlicher Nachteil dieser Methode ist, daß sie auf die Verwendung der Hemicube-Methode zur Formfaktorbestimmung beschränkt ist. Eine weitaus größere Einschränkung der Verwendbarkeit des *Back-Buffer*-Verfahrens stellt jedoch die Konzeption der Methode zur Berechnung von Animationen dar,

bei denen die Bewegungsbahnen dynamischer Objekte vor Berechnungsbeginn festgelegt sein müssen. Aus diesem Grund sind beliebige Interaktionen mit der Szene nicht möglich, wodurch sich diese Methode für interaktive Systeme disqualifiziert.

Von George u.a. wurde 1990 ein auf der Progressive-Refinement-Methode basierendes Verfahren zur Berechnung von Radiosity-Lösungen in dynamisch veränderbaren Umgebungen vorgestellt [GSG90]. Die grundlegende neue Idee dieser Methode besteht in der Durchführung einer Umverteilung einer zuvor berechneten Lichtverteilung. Bei dieser Umverteilung wird eine zuvor berechnete Radiosity-Lösung an eine neue räumliche Anordnung von Objekten in der Szene oder an veränderte Materialeigenschaften von Objekten angepaßt. Das Verfahren berechnet positive und negative Korrekturwerte zur Aktualisierung der zuvor berechneten spezifischen Ausstrahlungen von Patches. Die Korrekturwerte werden zunächst nur für die von einer Benutzerinteraktion direkt betroffenen Patches berechnet und dann in der Szene propagiert, um auch die indirekten Beleuchtungseffekte anzupassen (Redistribution-Operation). Sobald die Umverteilung abgeschlossen ist, werden die herkömmlichen Progressive-Refinement-Iterationen bzgl. der neuen Szene fortgesetzt (Propagation-Operation), bis eine konvergierte Lösung berechnet ist oder eine weitere Szenenmodifikation durchgeführt wird. Um die Anzahl der zu betrachtenden Empfänger-Patches bei dem Umverteilungs-Prozeß im Anschluß an eine Geometrieänderung einzuschränken, werden Kohärenzeigenschaften in der Szene ausgenutzt.

Fast zeitgleich zu dem von George u.a. [GSG90] vorgestellten Verfahren wurde von Chen [Che90] eine ebenfalls auf dem Progressive-Refinement-Verfahren aufbauende inkrementelle Radiosity-Methode entwickelt. Dieses Verfahren ermöglicht ebenfalls die Durchführung von Interaktionen mit den Objekten einer Szene während des Ablaufs der Radiosity-Simulation. Anders als bei dem Verfahren von George u.a. wird der Lösungsprozeß des Progressive-Refinement-Verfahrens nicht explizit unterbrochen, um zunächst eine vollständige Umverteilung der Energie vorzunehmen. Lediglich für die durch eine Modifikation direkt betroffenen Patches der Szene werden sofort im Anschluß an die Benutzerinteraktion *inkrementelle Korrekturwerte* der entsprechenden spezifischen Ausstrahlungen berechnet. Diese inkrementellen Korrekturwerte werden je nach Wichtigkeit (d.h. Auswirkung auf die globale Beleuchtungssituation) im weiteren Berechnungsverlauf mittels der normalen Progressive-Refinement-Iterationen innerhalb der Szene propagiert, bis eine konvergierte Lösung für die aktuelle Szene berechnet ist. Um einen möglichst großen Grad an Interaktivität zu erreichen, wurde das von Chen vorgestellte System auf Basis von zwei separate Komponenten realisiert: Einer Modellierungskomponente (*Modeler*), mit dem der Benutzer die Szene verändern kann, und einer Radiosity-Komponente (*Renderer*), welche die globale Beleuchtung auf Basis des modifizierten Progressive-Refinement-Verfahrens simuliert und das Ergebnis nach jeder Iteration darstellt. Die Modellierungskomponente informiert das Radiosity-Programm nach jeder Iteration mittels Nachrichten über die zwischenzeitlich erfolgten Modifikationen der Szene. Durch inkrementell berechnete Korrekturwerte wird die Lichtenergieverteilung innerhalb der Szene im Rahmen der Progressive-Refinement-Iterationen aktualisiert. Da die Beleuchtungssimulation bei Verwendung dieses Konzeptes bereits beginnen kann bevor eine Szene vollständig modelliert wurde, spricht Chen vom Paradigma des *Rendering while Modeling*, welches das bisherige bei der photorealistischen Bildgenerierung gültige Paradigma *Modeling then Rendering* ersetzen soll.

4.2 Repropagation vs. Redistribution

Es gibt eine grobe Klassifizierung der Radiosity-Verfahren zur Unterstützung von dynamisch veränderbaren Umgebungen in *Redistribution*- und *Repropagation*-Verfahren. Bei Repropaga-

tion-Verfahren wird zunächst eine feste Anzahl von Iterationen durchgeführt. Im Anschluß an diese Iterationen wird dem Anwender die Durchführung einer Modifikation gestattet. Unmittelbar nach dieser Benutzerinteraktion werden die zuvor durchgeführten Iterationen erneut durchgeführt, wobei die modifizierte Szene als Berechnungsgrundlage dient. Spezielle Datenstrukturen ermöglichen die Einsparung von Formfaktorberechnungen (z.B. *Shadow-Form Factor-List* – siehe Müller u.a. [MS95]). Hierbei werden zuvor berechnete Formfaktoren gespeichert und bei Korrekturberechnungen wiederverwendet, falls sie von der Geometrieänderung nicht betroffen sind. Diese Verfahren eignen sich besonders gut in interaktiven VR-Anwendungen, die extrem kurze Antwortzeiten benötigen und dabei Qualitätseinbußen in Kauf nehmen. Es ist zu beachten, daß die Anzahl der Iterationen sehr klein sein muß, um diese kurzen Antwortzeiten zu gewährleisten. In den meisten Fällen können lediglich die Energie der primären Lichtquellen verteilt werden. Somit fehlen die für Radiosity-Visualisierungen typischen Color-Bleeding-Effekte und indirekte Beleuchtung. Um diese Einschränkungen teilweise zu umgehen, wird in existierenden VR-Systemen oftmals mit Fake-Schatten und virtuellen Lichtquellen gearbeitet, welche durch spezielle Hardware-Unterstützung schneller bearbeitet werden können. Es ist zu beachten, daß bei der Verwendung einer Shadow-Form-Factor-List die Durchführung von adaptiver Verfeinerung nicht möglich ist, bzw. nicht effizient unterstützt wird. Somit muß das initiale Netz von Patches hinreichend fein sein, um Schattenverläufe approximieren zu können (*uniforme Netzgenerierung*). Dieses führt jedoch dazu, daß auch Bereiche des Netzes sehr stark adaptiv verfeinert werden, in denen es nicht notwendig ist.

Um den höheren Qualitätsanforderungen im Bereich der Architekturvisualisierung und der Beleuchtungstechnik gerecht werden zu können, müssen die zuvor beschriebenen Redistribution-Verfahren von George u.a. [GSG90] und Chen [Chen90] eingesetzt werden. Hierbei wird eine Konvergenzschranke für die Qualität der Lösung vorgegeben. Im Anschluß an eine Interaktion wird eine Lösung mit der geforderten Qualität berechnet. Durch den Progressive-Refinement-Ansatz dieser Redistribution-Verfahren ist die Visualisierung der wesentlichen veränderten Beleuchtungseffekte schon nach sehr kurzer Zeit möglich, da die Iterationen nach Wichtigkeit, d.h. Auswirkung auf die globale Beleuchtung in der Umgebung, durchgeführt werden.

4.3 Inkrementelle Radiosity-Berechnungen

Die in dieser Arbeit beschriebenen Methoden zur parallelen Berechnung von Radiosity-Lösungen in dynamisch veränderbaren Umgebungen basieren auf dem inkrementellen Radiosity-Verfahren von Chen [Chen90], welches als Erweiterung des Progressive-Refinement-Verfahrens [CCWG88] betrachtet werden kann.

Das inkrementelle Radiosity-Verfahren überprüft zu Beginn jeder Iteration, ob eine Szenenmodifikation durchgeführt wurde. Ist dieses der Fall, wird der Lösungsprozeß zeitweise unterbrochen. Zur Behandlung der Benutzerinteraktion müssen innerhalb des Radiosity-Programmes den Interaktionen entsprechende Methoden zur inkrementellen Anpassung von Radiosity-Werten (siehe 4.3.1 und 4.3.2) aufgerufen werden. Anschließend wird das normale Progressive-Refinement-Verfahren fortgesetzt, um die inkrementellen Radiosity-Werte in der Szene zu propagieren und somit auch die indirekten Beleuchtungseffekte anzupassen.

Bei erlaubten Modifikationen durch den Benutzer wird unterschieden zwischen

- Änderungen von Materialattributen wie z.B. die Eigenemission einer Lichtquelle oder die Reflektivität einer Oberfläche, sowie

- Änderungen der Szenengeometrie durch Entfernen oder Einfügen von Objekten.

Objekte der Szene, die durch eine Interaktion des Benutzers betroffen sind, werden im weiteren Verlauf als *dynamische Objekte* bezeichnet.

Das inkrementelle Radiosity-Verfahren von Chen ermöglicht die Durchführung von mehreren aufeinanderfolgenden Geometrieänderungen. Hierbei werden die Modifikationen in einer speziellen Datenstruktur (*Geometry-Queue* – siehe 4.3.4) gespeichert und nacheinander abgearbeitet, ohne das der Benutzer auf die Konvergenz der Lösung warten muß, bevor er eine weitere Modifikation durchführen kann.

4.3.1 Veränderungen von Materialattributen

Nachdem ein Benutzer innerhalb der Modellierungskomponente die Materialattribute eines Objektes verändert hat und diese Modifikation der Radiosity-Komponente mittels Nachricht bekannt gemacht wurde, werden für alle Patches des dynamischen Objektes *inkrementelle Radiosity-Werte* (Differenz-Radiosities) berechnet. Ein inkrementeller Radiosity-Wert eines Patches gibt einen lokalen Korrekturwert an, welcher den zuvor berechneten Radiosity-Wert des Patches an die neuen Gegebenheiten in der Szene anpaßt. Die inkrementellen Änderungen der Radiosity-Werte des dynamischen Objektes werden durch weitere Iterationen des Progressive-Refinement-Verfahrens innerhalb der Szene propagiert.

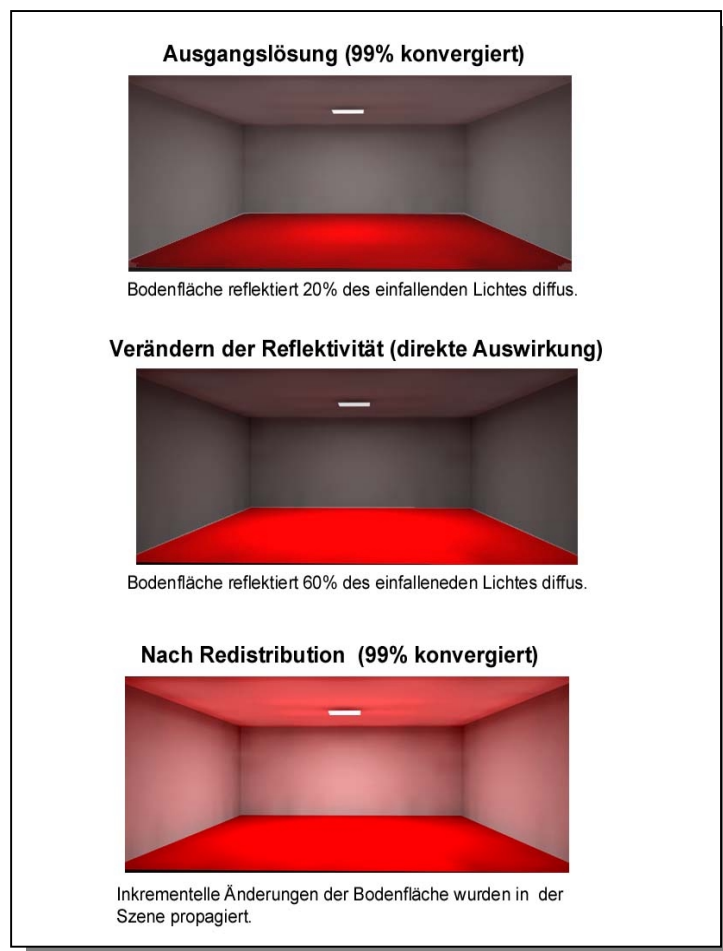


Abb. 4.1: Ändern von Materialattributen.

In Abb. 4.1 wurde durch den Anwender die Reflektivität der Bodenfläche verändert. Somit ist in diesem Fall die Bodenfläche das dynamische Objekt. Durch das inkrementelle Radiosity-Verfahren werden direkt im Anschluß an die Interaktion die Radiosity-Werte auf der Bodenfläche durch Berechnung von inkrementellen Radiosity-Werten angepaßt. Da unter Umständen die Energie der Oberflächenpatches in den vorangegangenen Iterationen des Progressive-Refinement-Verfahrens in die Umgebung reflektiert wurde, muß die Korrektur im weiteren Berechnungsverlauf in der Umgebung propagiert werden. D.h. die von dem dynamischen Objekt zuvor reflektierte Energie, die von den anderen Objekten der Umgebung empfangen wurde, muß angepaßt werden.

Die Radiosity eines Patches i vor einer Modifikation ist (wie in Abschnitt 3.1.1 gezeigt) gegeben durch

$$B_i^{alt} = E_i^{alt} + \rho_i^{alt} \sum_{j=1}^n B_j F_{ij}. \quad (4.1)$$

Seien E_i^{neu} die neue Emissivität von Patch i an und ρ_i^{neu} die neue Reflektivität von Patch i , falls die Werte vom Benutzer geändert wurden. Dann ist der neue Radiosity-Wert von Patch i unter Berücksichtigung der Veränderung der Materialattribute gegeben durch

$$B_i^{neu} = E_i^{neu} + \rho_i^{neu} \sum_{j=1}^n B_j F_{ij}. \quad (4.2)$$

Die inkrementelle Radiosity von Patch i ergibt sich durch Differenzbildung zwischen den Formeln für B_i^{alt} und den neuen B_i^{neu} . Somit ist

$$\Delta B_i = (E_i^{neu} - E_i^{alt}) + \frac{(\rho_i^{neu} - \rho_i^{alt})(B_i^{alt} - E_i^{neu})}{\rho_i^{alt}}, \quad \text{mit } \rho_i^{alt} \neq 0. \quad (4.3)$$

Die Berechnung der inkrementellen Radiosity-Werte im Anschluß an eine Materialänderung kann wie folgt in das Progressive-Refinement-Verfahren integriert werden:

1. Zunächst wird für jedes Patch i des dynamischen Objektes ein inkrementeller Radiosity-Wert ΔB_i gemäß Gleichung (4.3) berechnet.
2. Für jedes Patch i des dynamischen Objektes wird der inkrementelle Radiosity-Wert ΔB_i auf die aktuelle Radiosity B_i und auf die nicht verschossene spezifische Ausstrahlung B_i'' addiert.

$$\begin{aligned} B_i &= B_i + \Delta B_i \\ B_i'' &= B_i'' + \Delta B_i \end{aligned} \quad (4.4)$$

3. Die Iterationen des Progressive-Refinement-Verfahrens werden wie üblich fortgesetzt.

Zu beachten ist, daß bei der Bestimmung des Shooting-Patches die absoluten Werte der nicht verschossene spezifische Ausstrahlung betrachtet werden, um die Verteilung von negativer Energie zu ermöglichen (siehe 4.3.5). Dieses ist z.B. dann notwendig, wenn die Emission (die von einer Lichtquelle abgegebene primäre Energie) verringert wird. In diesem Fall wird die inkrementelle Radiosity nach Gleichung (4.3) der entsprechenden primären Lichtquellenpat-

ches negativ. Wenn die negative Energie in der Szene propagiert wird, reduziert sich auf dem jeweiligen Empfängerpatch die indirekt empfangene Energiemenge der Modifikation entsprechend.

4.3.2 Veränderung der Szenengeometrie

Im Gegensatz zu der Veränderung von Materialattributen sind bei der Veränderung der Geometrie hiervon die Formfaktoren innerhalb der Szene betroffen. Somit gestaltet sich die Berechnung von inkrementellen Radiositäten für die durch die Geometrieänderung betroffenen Patches schwieriger. Nachdem die Szenengeometrie verändert wurde, muß eine Umverteilung der Energieanteile vorgenommen werden, die vor dem Modifikationszeitpunkt in der Szene verteilt wurden. Hierbei kann der Fall auftreten, daß sich die relative Sichtbarkeit zwischen Patches der Szene aufgrund der Modifikation verändert, wodurch auch die Formfaktoren verändert werden. Bei der Umverteilung von Energieanteilen müssen somit in Regionen, die von der Modifikation betroffen sind, Formfaktoren bzgl. der alten und der neuen Szene berechnet werden.

Ein Beispiel für ein Umverteilungsprozeß im Anschluß an eine Geometrieänderung (verschieben eines Tisches) ist in Abb. 4.1 gegeben. Den Berechnungen liegt eine Szene zugrunde, welche einen Büroraum repräsentiert.

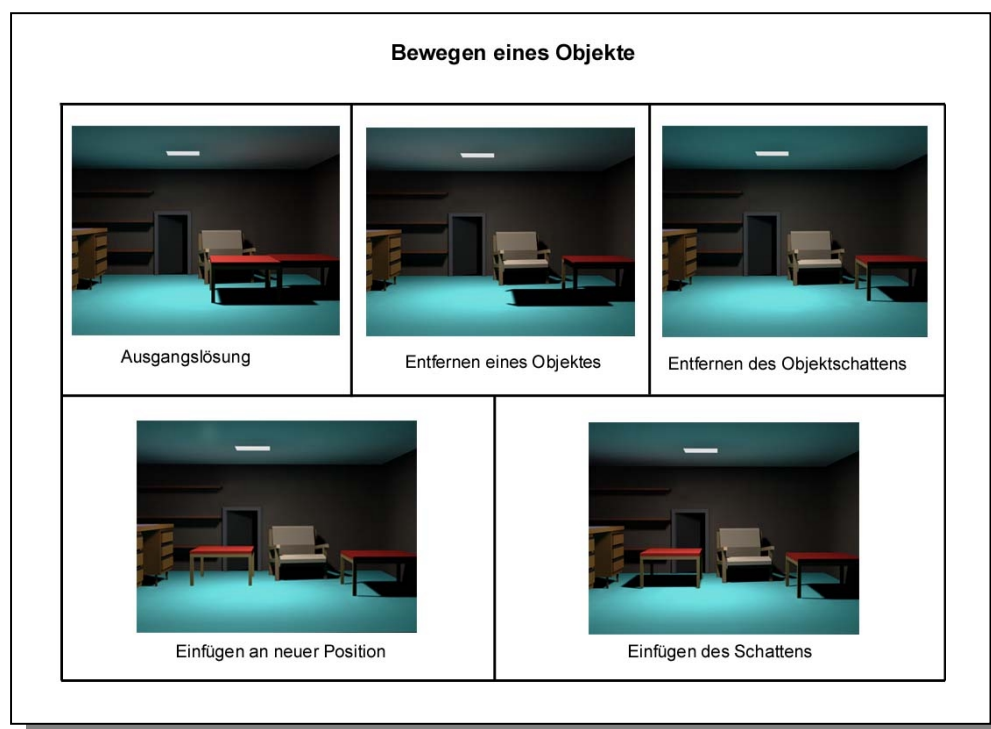


Abb. 4.2: Korrekturberechnungen nach einer Objektbewegung.

Die neuen Radiosity von einem Patch j berechnet sich bzgl. eines Shooting-Patches i wie folgt:

1. Der Radiosity-Anteil, welcher von Patch i an Patch j vor der Geometrieänderung abgegeben wurde, muß entfernt werden:

$$B'_j = B_j - \rho_j B_i^s F_{ji}^{alt} \quad (4.5)$$

Hierbei ist $B_i^s = B_i - B_i^u$ der Energieanteil, der von Patch i vor der Änderung in die Umgebung abgestrahlt wurde. B_i ist die gesamte Radiosity von Patch i und B_i^u die aktuell nicht verschosene spezifische Ausstrahlung.

2. Die gesamte spezifische Ausstrahlung (Radiosity) von Patch i wird gemäß dem neuen Formfaktor F_{ji}^{neu} verteilt:

$$B''_j = B_j + \rho_j (B_i^s \Delta F_{ji} + B_i^u F_{ji}^{neu}) \quad (4.6)$$

Hierbei ist $\Delta F_{ji} = F_{ji}^{neu} - F_{ji}^{alt}$ der inkrementelle Formfaktor.

Es ist zu beachten, daß keine expliziten Redistribution-Schritte wie bei George u.a. [GSG90] durchgeführt werden, sondern die Umverteilung der Korrekturwerte in den Progressive-Refinement-Iterationen integriert sind.

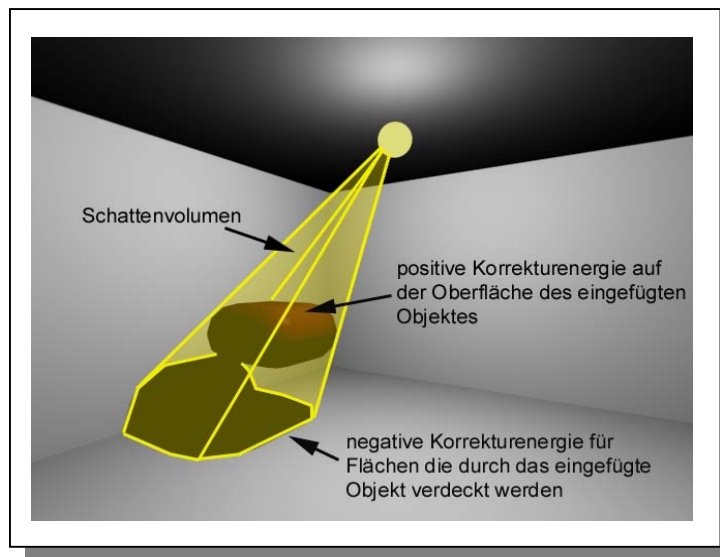


Abb. 4.3: Verteilung von Korrekturenergie unter Ausnutzung von lokalen Kohärenzeigenschaften.

In Abbildung 4.3 ist das Einfügen eines dynamischen Objektes in eine Szene dargestellt. Hierbei wird davon ausgegangen, daß zuvor die Energie der primären Lichtquelle an die Patches des statischen Teils der Szene (Wände) verteilt wurde. Nach dem Einfügen des Objektes wird den Patches auf der Oberfläche des dynamischen Objektes positive Korrekturenergie von der primären Lichtquelle zugewiesen. Dieses ist notwendig, da die Patches des dynamischen Objektes in der vorangegangenen Progressive-Refinement-Iteration noch nicht Bestandteil der Szene waren und somit keine Energie an das dynamische Objekt abgegeben wurde. An die Patches der Wandfläche im Schattenbereich des eingefügten Objektes wird negative Korrekturenergie verteilt. Da nach der Änderung der Szenengeometrie die Lichtquelle für diese Patches verdeckt ist, muß der Anteil der Energie abgezogen werden, der zuvor von der Lichtquelle direkt empfangen wurde. Durch Bildung eines Schattenvolumens, das durch die Lichtquelle und das dynamische Objekt festgelegt ist, können die Patches sehr effizient bestimmt

werden, für welche der inkrementelle Formfaktor einen Wert ungleich Null hat. Nur für diese Patches muß eine Korrektur der zuvor berechneten spezifischen Ausstrahlung durchgeführt werden. Somit wird die Lokalität von Änderungen der Szenengeometrie ausgenutzt, um die Komplexität der Berechnung von inkrementellen Formfaktoren zu reduzieren.

4.3.3 Berechnung inkrementeller Formfaktoren

Da in der Regel nur ein geringer Teil der Formfaktoren von einer Änderung der Geometrie betroffen ist, kann die in 4.3.2 beschriebene Aktualisierung der direkten Auswirkungen durch Ausnutzung von lokaler Kohärenz beschleunigt werden. Hierbei ist zu beachten, daß nur ein Patch, welches von der Veränderung betroffen ist, einen inkrementellen Formfaktor ungleich Null hat. Somit müssen nur die Gebiete innerhalb der Szene betrachtet werden, in denen die Geometrieänderung Einfluß auf die relative Sichtbarkeit von Patches hat.

Der triviale Ansatz zur Berechnung der inkrementellen Formfaktoren besteht darin, jeweils die alten und neuen Formfaktoren in zwei Berechnungsschritten separat zu bestimmen. Dieses ist sehr zeitaufwendig und unnötig. Bei der Berechnung der Formfaktoren kann Kohärenz je nach Berechnungsmethode der Formfaktoren auf unterschiedliche Art und Weise ausgenutzt werden:

- Hemicube-Methode [CG85]
Bei Verwendung der Hemicube-Methode werden zwei Hemicubes benötigt, um inkrementelle Formfaktoren zu berechnen. Der statische Teil der Szene muß nur einmal auf den Hemicube projiziert werden. Dieser Hemicube wird dupliziert und der dynamische Teil wird auf jeden dieser Hemicubes separat projiziert (alte und neue Position). Im Anschluß daran lassen sich sehr leicht die Hemicube-Pixel identifizieren, welche durch das dynamische Objekt belegt werden. Somit sind aufgrund der Informationen, die in den Hemicube-Pixeln gespeichert sind, auch die Patches bekannt, für welche der inkrementelle Formfaktor einen Wert ungleich Null hat. Diese Vorgehensweise ist ähnlich wie bei dem Back-Buffer-Verfahren von Baum u.a. [BWCG86].
- Raytracing Techniken [WKEH89]
Mit Hilfe hierarchischen Begrenzungsvolumen [GS87] [KK86] und Schattenvolumen (siehe Abbildung 4.3) [NN85] oder anderen Techniken zur Aufteilung des Objektraumes (z.B. [HW91]) kann der Bereich innerhalb der Szene identifiziert werden, welcher durch die Geometrieänderung betroffen ist, wodurch die Anzahl der zu berechnenden Formfaktoren deutlich reduziert wird.

4.3.4 Geometry-Queue

Ein wesentliches Konzept des hier beschriebenen inkrementellen Radiosity-Verfahrens für dynamisch veränderbare Umgebungen ist die *Geometry-Queue*. Nach der Durchführung einer Geometrieänderung müssen alle Patches, die zuvor Energie in der Szene verteilt haben, ihre Energie erneut bzgl. der neuen Szenengeometrie verteilen, bevor der normale Lösungsprozeß fortgesetzt werden kann. Weiterhin darf während dieser Umverteilungsphase keine weitere Geometrieänderung durchgeführt werden, da es sonst zu Fehlern bei den inkrementellen Berechnungen kommen kann (siehe [Chen90]). Diese Einschränkungen können durch die Verwendung einer Geometry-Queue aufgehoben werden. Die vorgenommenen Geometrieänderungen werden in eine Geometry-Queue vermerkt (z.B. *Einfügen Objekt x* oder *Löschen Ob-*

jekt y). Die Verwendung der Geometry-Queue ermöglicht beliebig viele nacheinander ausgeführte Geometrieänderungen durch den Benutzer, da jede Version der Szene zu jedem Zeitpunkt der inkrementellen Berechnungen rekonstruierbar ist. Hierbei entsteht eine Szene S_k aus der ursprünglichen Szene S_0 , wenn die ersten k in der Geometry-Queue vermerkten Geometrieänderungen sukzessive ausgeführt werden. Somit ist $S_{Länge_Queue-1}$ die jeweils aktuellste Szene. Sobald eine Geometrieänderung vorliegt, wird diese in die Geometry-Queue eingefügt und der normale Lösungsprozeß fortgesetzt.

Jedes Patch i hat einen Queue-Counter σ_i , welcher die Anzahl der Geometrieänderungen in der Queue angibt, die durch dieses Patch beim Verteilen seiner nicht verschossenen spezifischen Ausstrahlung bisher berücksichtigt wurden (initial sind die Queue-Counter aller Patches Null). Die Szenen, die zur Berechnung der inkrementellen Formfaktoren benötigt werden, lassen sich mit Hilfe der Queue-Counter generieren. Die *alte* Szene, bzgl. welcher das ausgewählte Shooting-Patch zuletzt seine nicht verschossene spezifische Ausstrahlung verteilt hat, ist durch den Queue-Counter des Patches bestimmt. Die *neue* Szene, welche bei dem aktuellen Shooting-Vorgang berücksichtigt werden muß, wird durch die gesamte Queue beschrieben.

Ein wesentlicher Vorteil der Geometry-Queue ist, daß mehrere Veränderungen die nacheinander durchgeführt werden, keinen großen Extra-Aufwand erzeugen. Der Algorithmus betrachtet nur die Netto-Änderung in der Szene, d.h. das wenn ein Objekt mehrfach bewegt wird nur die für das ausgewählte Shooting-Patch bisher berücksichtigte Position des Objektes und die aktuelle Position des Objektes in der Umgebung bei der Umverteilung berücksichtigt werden. Nachdem das Shooting-Patch seine nicht verschossenen (bzw. inkrementelle) Radiosity bzgl. einer Szene S_k verteilt hat, wird der Geometry-Queue Counter dieses Patches auf k gesetzt.

Es ist zu beachten, daß in der Geometry-Queue lediglich Einfüge- und Löschoperationen von Objekten gespeichert werden. Dieses stellt jedoch keine Einschränkung des Verfahrens dar, da sich Objektbewegungen und Objektverformungen durch aufeinanderfolgende Lösch- und Einfüge-Operationen realisieren lassen (siehe Abbildung 4.2).

4.3.5 Auswahl des Shooting-Patches

Die Reihenfolge, in der die nicht verschossene spezifische Ausstrahlungen der Patches während der Progressive-Refinement-Iterationen innerhalb der Umgebung verteilt werden, hat einen wesentlichen Einfluß auf das Konvergenzverhalten des Verfahrens [CCWG88]. Hierbei ist es sinnvoll, das Patch als Shooting-Patch zu wählen, welches in der jeweiligen Iteration die größte nicht verschossene spezifische Ausstrahlung besitzt und somit den größten Einfluß auf die Beleuchtung der Szene hat. Dieses gestaltet sich im Fall der inkrementellen Radiosity-Berechnungen etwas schwieriger, da der Einfluß eines Patches auf die Beleuchtung von der Art und Anzahl der vorgenommenen Änderungen in der Szene abhängen. Die Energie, die von einem Patch i bei jedem Shooting-Vorgang an die Szene abgegeben wird, setzt sich wie folgt zusammen:

$$B_{i-to-scene} = \sum_{j=1}^n (B_i^u \rho_j F'_{ji} + B_i^s \rho_j \Delta F_{ji}) \quad (4.7)$$

Der Formfaktor F'_{ji} läßt sich grob approximieren durch (siehe [CCWG88]):

$$F'_{ji} \approx \frac{A_i}{\sum_{j=1}^n A_j} \quad (4.8)$$

Der inkrementelle Formfaktor von einem beliebigen Patch j zu Patch i ist die Differenz zwischen dem alten und dem neuen Formfaktor. Eine mögliche Approximation für diese Differenz läßt sich mit Hilfe des Geometry-Queue-Zählers σ_i von Patch i angeben. Hierbei wird davon ausgegangen, daß der inkrementelle Formfaktor für die Patches am größten ist, die bisher die wenigsten Geometrieänderungen beim Verteilen ihrer nicht verschossenen spezifischen Ausstrahlung berücksichtigt haben. Somit ist eine mögliche Approximation für den inkrementellen Formfaktor

$$\Delta F_{ji} \approx (1 - \frac{\sigma_i}{\sigma}) F'_{ji} \quad (0 \leq \sigma_i \leq \sigma), \quad (4.9)$$

wobei

- σ_i der Patch i zugeordnete Queue-Counter (Anzahl der durch Patch i beim Verteilen der Energie bisher berücksichtigten Geometrieänderungen) und
- σ die Länge der Geometry-Queue (Anzahl aller Geometrieänderungen) ist.

Es ist zu beachten, daß der Schätzwert nicht immer eine gute Approximation für den inkrementellen Formfaktor ist (z.B. wenn die Änderungen von dem Patch nicht sichtbar sind).

Durch Einsetzen in die Formel für $B_{i-to-scene}$ und Umformen der Formel erhält man:

$$B_{i-to-scene} \approx \left(|B_i^u| + (1 - \frac{\sigma_i}{\sigma}) |B_i^s| \right) A_i \frac{\sum_{j=1}^n \rho_j}{\sum_{j=1}^n A_j} \quad (4.10)$$

Da die Summen in Zähler und Nenner konstant und positiv sind, brauchen sie bei der Bestimmung des Shooting-Patches nicht ausgewertet werden. Somit wird das Patch i mit dem größten Wert β_i als Shooting-Patch für die nächste Iteration ausgewählt, wobei

$$\beta_i = \left(|B_i^u| + (1 - \frac{\sigma_i}{\sigma}) |B_i^s| \right) A_i. \quad (4.11)$$

Bei der Bestimmung des Patches i mit maximalen β_i ist es notwendig, die Absolutbeträge von B_i^u und B_i^s zu betrachten.

4.3.6 Das Verfahren im Überblick

In Abbildung 4.4 ist der Gesamtablauf des inkrementellen Radiosity-Verfahrens beschrieben.

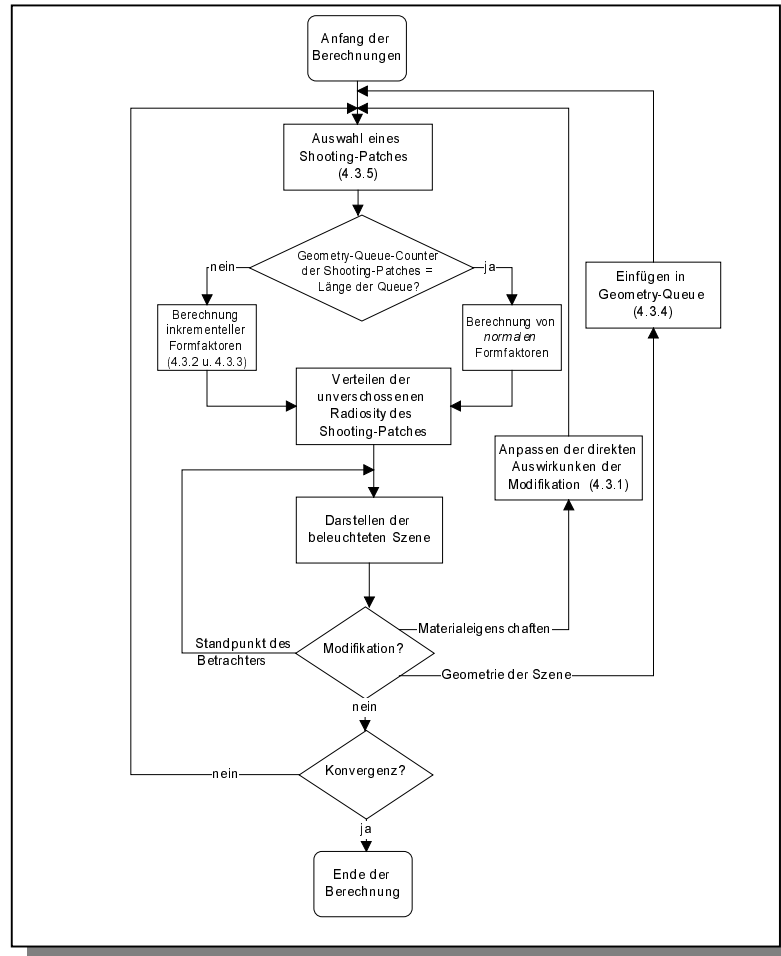


Abb. 4.4: Ablaufdiagramm des inkrementellen Radiosity-Verfahrens.

4.4 Abschließende Bemerkungen

Während Korrekturen der Lichtenergieverteilung bei Änderungen von Materialeigenschaften mit den Verfahren von George u.a. [GSG90] und Chen [Chen90] sehr schnell berechnet werden können, ist eine Aktualisierung nach Veränderungen der Geometrie mit den vorgestellten Verfahren nicht in Echtzeit möglich. Wie schnell die direkten Auswirkungen aktualisiert werden können, hängt stark von der Änderung selbst ab. Im allgemeinen erfordern kleine Modifikationen weniger Neuberechnungen als umfangreiche Umbauten in der Szene.

Beide Methoden berechnen bei der Bestimmung von Korrekturwerten im Anschluß an eine Veränderung der Geometrie differentielle Formfaktoren. Dieses bedingt die Bestimmung von Formfaktoren bzgl. der Szene vor und nach der Veränderung. Wären nun die Formfaktoren, die vor einer Modifikation durch das Progressive-Refinement-Verfahren berechnet wurden zum Zeitpunkt der Berechnung der differentiellen Formfaktoren noch bekannt, könnte der notwendige Berechnungsaufwand halbiert werden. Der Speicherplatzbedarf des Verfahren würde sich dann jedoch von $O(n)$ auf $O(n^2)$ vergrößern ($n = \text{Anzahl der Patches}$). Dieses ist jedoch für komplexe Szenen nicht praktikabel.

Die Entwickler beider Methoden weisen darauf hin, daß die Parallelisierung der Verfahren eine vielversprechende Möglichkeit darstellt, auch die Auswirkungen von Veränderungen der

Geometrie einer Szene auf die globale Beleuchtungssituation mit sehr kurzen Systemantwortzeiten bei der Bildgenerierung wiederzugeben.

Kapitel 5

Parallelisierung der Radiosity-Methode

Globale Beleuchtungsverfahren wie Monte-Carlo-Raytracing und Radiosity ermöglichen zwar eine qualitativ gute Bildgenerierung, besitzen jedoch im allgemeinen eine sehr große Berechnungs- und Datenkomplexität. Aus diesem Grund ist der photorealistische Bildgenerierungsprozeß auf Basis von globalen Beleuchtungsverfahren für komplexe Szenen sehr zeitintensiv und oftmals mit herkömmlichen Computersystemen nicht realisierbar. Sehr früh wurden die Vorteile der Parallelverarbeitung für die Belange der Computergrafik erkannt und genutzt. Zunächst wurden Parallelisierungen von Hidden-Surfac-Algorithmen [HF85] [GF85], spezieller Graphik-Funktionen [Clark82] [NOKSO83] sowie der Raytracing-Methode [DS84] [PB85] [CWBV86] [NO86] durchgeführt, wodurch erhebliche Reduktionen der Berechnungszeiten erzielt werden konnten.

Mittlerweile ist die Parallelisierung von neu entwickelten sequentiellen Algorithmen eine Standard-Methode innerhalb der Computergrafik, um eine Beschleunigung der in den meisten Fällen hochgradig berechnungsintensiven Verfahren zu erzielen. Dieses gilt insbesondere für globale Beleuchtungsmethoden. Die entwickelten parallelen Verfahren zur globalen Beleuchtungssimulation lassen sich grob in zwei Kategorien einordnen.

Parallelisierungen der ersten Kategorie gestatten jedem Berechnungsprozeß (*Worker*) den Zugriff auf die komplette Szenenbeschreibung. Dieses wird ermöglicht, indem die Szene in den lokalen Speicherbereichen aller Prozessoren repliziert wird. In diesem Fall können die Prozessoren unabhängig voneinander arbeiten, wobei lediglich mit einem zentralen Prozeß (i.a. als *Master* bezeichnet) kommuniziert wird. Dieser spezielle Prozeß ist für die Zuteilung von Lastelementen an die Berechnungsprozesse verantwortlich. Ein Nachteil einer solchen Vorgehensweise ist, daß der Master schon bei Szenen mit moderater Komplexität einen Engpaß bildet, wenn die Anzahl der Prozessoren erhöht wird. In diesem Fall entstehen auf Seiten der Worker große Wartezeiten (Idle-Zeiten), welche die Effizienz der parallelen Berechnungen erheblich beeinträchtigen. Der entscheidende Nachteil ist jedoch, daß die vollständige Szenenbeschreibung in den lokalen Speicherbereichen aller Prozessoren gespeichert wird. Somit ist die Komplexität der Szenen durch die Größe der lokalen Speicherbereiche beschränkt. Im allgemeinen wird durch Parallelisierungsstrategien aus dieser Kategorie ein entscheidender Vorteil von parallelen Systemen nicht genutzt, nämlich das komplexere Problemstellungen gelöst werden können, indem mehr Prozessoren für den Lösungsprozeß verwendet werden. Somit sind die Skalierbarkeitseigenschaften dieser parallelen Verfahren stark begrenzt.

Verfahren der zweiten Kategorie werden als datenparallele Verfahren bezeichnet. Bei diesen Ansätzen wird eine Aufteilung der Eingabeszene vorgenommen und die so gebildeten Teilszenen auf die lokalen Speicherbereiche der Prozessoren verteilt. Die Berechnungsprozesse

lösen somit Teilprobleme auf Basis der jeweils lokal verfügbaren Szenendaten. Die berechneten Teillösungen werden am Berechnungsende zu einer vollständigen Lösung kombiniert. Es ist zu beachten, daß die Berechnungsprozesse im allgemeinen während der Berechnungen kommunizieren müssen, da nicht alle benötigten Daten lokal verfügbar sind. Eine entscheidende Rolle für die Effizienz dieser Verfahren stellt somit die initiale Partitionierung des Problems dar, welche die Ausnutzung von Lokalitätseigenschaften bei den parallelen Berechnungen (Daten-Kohärenz) ermöglichen sollte, um den Kommunikationsbedarf so gering wie möglich zu halten. Die Aufteilung der Szenen stellt eine statische Lastverteilung dar. Es muß somit gewährleistet werden, daß die gebildeten Teilszenen eine annähernd gleiche Komplexität besitzen. Eine solche statische Lastverteilung vor Beginn der parallelen Berechnungen kann die zur Laufzeit dynamisch entstehende Lastsituation nur unzureichend abschätzen. Somit spielt der dynamische Lastausgleich während der Berechnungen oftmals eine entscheidende Rolle bei der Steigerung der Effizienz von parallelen Verfahren der zweiten Kategorie.

Die Implementierung von Parallelisierungsstrategien der ersten Kategorie ist in den meisten Fällen sehr einfach und ohne spezielle Kenntnisse im Bereich der Parallelverarbeitung möglich. Sollen jedoch Verfahren mit sehr guten Skalierbarkeitseigenschaften entwickelt werden, ist die Verwendung einer Parallelisierungsstrategie der zweiten Kategorie unabdingbar. Dieses setzt jedoch fundierte Kenntnisse im Bereich der Parallelverarbeitung voraus, da die Aspekte der Kommunikation, des Prozeß-Mappings, der Ende-Erkennung und der Lastverteilung erheblichen Einfluß auf die Effizienz der parallelen Berechnungen hat und die entsprechenden Methoden oftmals sehr schwer praktisch umsetzbar sind.

Im weiteren Verlauf werden die zentralen Ergebnisse dieser Arbeit präsentiert. Hierbei handelt es sich insbesondere um zwei neue datenparallele Strategien für die schnelle und effiziente parallele Berechnung von Radiosity-Lösungen auf Basis eines Progressive-Refinement-Ansatzes. Durch diese Strategien werden sowohl komplexe statische als auch dynamisch veränderbare Szenen unterstützt.

5.1 Vergleich existierender paralleler Verfahren

Die Radiosity-Methode ist eines der Verfahren innerhalb der Computergrafik mit der größten Berechnungs- und Datenkomplexität und ist somit ein offensichtlicher Kandidat für eine Parallelisierung. Trotz der Tatsache, daß seit der Einführung der Radiosity-Methode im Jahr 1984 zahlreiche Verbesserungen vorgeschlagen wurden, die sowohl den Berechnungsaufwand als auch die Menge der gespeicherten Daten erheblich reduzieren, sind Berechnungszeiten von mehreren Stunden selbst für moderat komplexe Szenen üblich. Die Parallelisierung der Radiosity-Methode ermöglicht es, sehr komplexe Szenen zu verarbeiten und die Berechnungszeiten signifikant zu verkürzen. Ziel ist es, parallele Verfahren zu entwickeln, die möglichst gute Skalierbarkeitseigenschaften aufweisen und die Rechenkapazität der verwendeten Parallelrechner möglichst effizient nutzen.

Die bei der parallelen Radiosity-Berechnung auftretenden Probleme setzen sich somit wie folgt zusammen [CR98]:

- Die Formfaktorberechnungen stellen den berechnungsintensivsten Teil der Radiosity-Methode dar. Hierbei ist die Bestimmung der relativen Sichtbarkeit zwischen allen Paaren von diskreten Flächenelementen (Patches) innerhalb der Szene notwendig. Sind diese Flächenstücke bei einer datenparallelen Strategie auf verschiedenen Prozessoren gespeichert, ist Kommunikation notwendig, um korrekte Formfaktoren zu bestimmen.

- Während der Radiosity-Berechnungen wird für jedes Flächenelement dessen spezifische Ausstrahlung (Radiosity) gespeichert. Wenn eine Parallelisierungsstrategie verwendet wird, bei der die vollständigen Szenendaten auf jedem Prozessor lokal verfügbar sind (d.h. Mehrfachspeicherung), muß der für ein Patch berechnete Radiosity-Wert allen am Berechnungsprozeß beteiligten Prozessoren mittels Broadcasting bekannt gemacht werden. Dieses kann zu einem erheblichen Kommunikationsengpaß führen.
- Wenn Umgebungsbeschreibungen als Grundlage einer Radiosity-Berechnung verwendet werden, die in separate Regionen segmentiert sind (z.B. ein Haus mit Räumen, die unabhängig voneinander sind), beschränkt sich der Energietransfer auf einzelne, räumlich begrenzte Regionen. In solchen Fällen muß besonderes Augenmerk auf die Lastverteilung gelegt werden, da durch eine Verteilung solcher Szenen auf mehrere Prozessoren ein Lastungleichgewicht verursacht werden kann.
- Eine weitere Ursache für ein dynamisch entstehendes Lastungleichgewicht ist die Tatsache, daß die Berechnungszeiten für verschiedene Formfaktoren aufgrund unterschiedlicher geometrischer Beziehungen zwischen den Patches deutlich voneinander abweichen können. Dieses wird durch adaptive Verfeinerung (Substrukturierung) noch weiter verstärkt, da die Substrukturierungstiefen von Patches in verschiedenen Regionen der Szene teilweise erheblich voneinander abweichen.

Parallelisierungen der Radiosity-Methode, welche in der Vergangenheit durchgeführt wurden, berücksichtigen diese Aspekte auf unterschiedliche Art und Weise. Die wesentlichen Unterscheidungsmerkmale dieser parallelen Verfahren sind:

- *Verwendete Hardware:* Parallelisierungen existieren sowohl für Distributed-Memory Architekturen (Transputersystemen und Workstations mittels LAN gekoppelt) als auch Shared Memory-Systeme.
- *Parallelisierungsstrategie:* Der größte Teil der existierenden Parallelisierungen der Radiosity-Methode lassen sich direkt in die beiden zu Anfang dieses Kapitels beschriebenen Kategorien zur Parallelisierung von globalen Beleuchtungsverfahren einordnen. Es existieren jedoch auch Mischformen, bei denen zwar eine zentrale Instanz (Master) für die Zuordnung von Arbeitspaketen zu den Berechnungsprozessen (Worker) zuständig ist, die Szenendaten jedoch nicht mehrfach gespeichert werden. Zu Berechnungszwecken benötigte Szenendaten werden bei diesen Strategien zwischen den Workern mittels Nachrichten (*Message-Passing*) ausgetauscht.
- *Grundlegende Radiosity-Methode:* Ziel der Radiosity-Methode ist die Lösung des Radiosity-Gleichungssystems, welches die Verteilung der Lichtenergie innerhalb einer Umgebung beschreibt (siehe 3.1). Die Verwendung der Full-Matrix-Methode [GTGB84] als Grundlage einer Parallelisierung bietet sich nicht an, da sich diese Methode in der Vergangenheit sowohl in Bezug auf den Speicherplatz- und Rechenzeitbedarf als nicht effizient herausgestellt hat. Bei den effizientesten Verfahren handelt es sich um die Progressive-Refinement-Methode [CCWG88] und das hierarchische Radiosity-Verfahren [HS91]. Aufgrund der schlechten lokalen Eigenschaften der hierarchischen Verfahren gestaltet sich eine datenparallele Implementierung auf Distributed-Memory-Architekturen sehr schwierig, wie in [BG95] zu sehen ist. Der größte Teil der existierenden parallelen Radiosity-Verfahren basiert auf einem Progressive-Refinement-Ansatz.

- *Methode zur Bestimmung von Formfaktoren:* Wie in Kapitel 3 gezeigt wurde, können die Formfaktoren mit Hilfe von verschiedenen Techniken bestimmt werden. Diese unterscheiden sich sowohl bzgl. der Qualität der Approximation als auch bzgl. der benötigten Berechnungszeit erheblich voneinander. Die Hemicube-Methode [CG85] ist die bei parallelen Implementierungen die am häufigsten verwendete Methode zur Bestimmung von Formfaktoren. Aufgrund der unter Umständen sehr groben Approximation von Formfaktoren [BRW89] wird bei Parallelisierungen neueren Datums vermehrt auf die Methode von Wallace u.a. [WKEH89] zurückgegriffen.

Die ersten parallelen Radiosity-Verfahren basierten auf einem reinen Master/Worker-Ansatz, bei dem Szenendaten mehrfach gespeichert wurden [Chen89][RGG90][PSV90]. Diese Methoden sind die bis heute am meisten verbreitetsten parallelen Berechnungsverfahren zur Bestimmung von Radiosity-Lösungen in Szenen mit geringer Komplexität. Hierbei wird einem Prozessor die Aufgabe eines zentralen Lastverteilers (Master) zugeteilt, welcher den übrigen Prozessoren (Worker) Lastelemente zur Bearbeitung zuordnet. Im Kontext der parallelen Radiosity-Berechnungen ist ein Master-Prozeß für die Auswahl von Patches zuständig, deren nicht verteilten spezifischen Ausstrahlungen gemäß dem Progressive-Refinement-Verfahren an die übrigen Patches der Umgebungsbeschreibung verteilt werden sollen (Shooting-Patches). Ausgewählte Shooting-Patches werden an die Worker übermittelt, welche die Formfaktoren zwischen dem übermittelten Shooting-Patch und den übrigen Patches der Szene berechnen. Jeder Worker-Prozeß hat Zugriff auf die vollständige Szenenbeschreibung, da diese in den lokalen Speicherbereichen der an den Berechnungen beteiligten Prozessoren gespeichert ist. Die berechneten Formfaktoren werden an den Master-Prozeß geschickt, welcher den eigentlichen Energietransfer gemäß der empfangenen Formfaktoren durchführt. Ein ähnlicher Ansatz wurde von Baum u.a. [BW90] auf einem Mehrprozessorsystem mit gemeinsamem Speicher und spezieller 3D-Grafikhardware realisiert. Die Grafikhardware wird bei diesem Ansatz zur beschleunigten Durchführung der Hemicube-Projektionen bei den Formfaktorbestimmung verwendet.

Mit Hilfe von parallelen Radiosity-Verfahren auf Basis des Master/Worker-Ansatzes lassen sich gute Ergebnisse für kleine Szenen mit einer geringen Anzahl von Prozessoren erzielen. Die begrenzte Kommunikationsbandbreite zwischen Master- und Worker-Prozessen führt sehr schnell zu einer drastischen Verringerung der Effizienz, wenn die Szenenkomplexität oder die Anzahl der Prozessoren erhöht wird. Weiterhin ist die Komplexität der Szenen bei diesen parallelen Verfahren durch die Größe der lokalen Speicherbereiche der Prozessoren beschränkt. Um diese Restriktion zu umgehen, wurden Master/Worker-Ansätze entwickelt, die ohne Mehrfachspeicherung der Szene innerhalb von Rechnersystemen mit verteiltem Speicher auskommen [CP91][FP91]. Jeder der P Prozessoren des Rechnersystems bekommt initial die gleiche Anzahl von $N/(P-1)$ Patches zugeordnet, wobei N die Anzahl sämtlicher Patches der Umgebungsbeschreibung ist. Mehrere Iterationen ($P-1$) des sequentiellen Progressive-Refinement-Algorithmus werden gleichzeitig ausgeführt, indem ein zentraler Master-Prozeß $P-1$ Shooting-Patches auswählt und jedem der Worker-Prozesse eines dieser Patches zuordnet. Die Worker-Prozessoren projizieren zunächst die lokal verfügbaren Patches auf einen über dem entsprechenden Shooting-Patch positionierten Hemicube. Es ist zu beachten, daß zur korrekten Formfaktorberechnung bzgl. eines Shooting-Patches sämtliche Patches der Szene auf den entsprechenden Hemicube projiziert werden müssen. Um dieses zu ermöglichen, tauschen die Worker-Prozessoren die lokal gespeicherten Patches mittels Nachrichten aus (z.B. auf einem ringförmigen Verbindungsnetzwerk). Nachdem diese Austauschschritte insgesamt ($P-1$)-mal ausgeführt wurden, hat jeder Prozessor alle Patches der Umgebungsbeschreibung auf seinen Hemicube projiziert und speichert wieder seine ursprünglichen

Geometriedaten. Nun berechnet jeder Prozessor sämtliche Formfaktoren bzgl. des ihm zugeordneten Shooting-Patches gemäß der vollständigen Hemicube-Informationen. Diese Formfaktoren werden an die übrigen Worker-Prozesse übermittelt, welche eine Aktualisierung der lokal gespeicherten Patches gemäß der übermittelten Formfaktoren durchführen.

Durch Aufteilung der Szenendaten auf die Prozessoren eines Rechnersystems mit verteiltem Speicher wird erreicht, daß Radiosity-Berechnungen auch in komplexen Szenen durchgeführt werden können. Die Kommunikation zwischen den Prozessoren führt jedoch sehr schnell zu einer starken Auslastung der Kommunikationsverbindungen, wodurch die Performanz der parallelen Berechnungen mit steigender Anzahl von Prozessoren deutlich abnimmt. Problematisch bei diesen Ansätzen ist auch, daß sich eine adaptive Substrukturierung nur sehr schwer realisieren läßt und der Kommunikationsaufwand hierdurch erheblich gesteigert wird. Somit entspricht die Genauigkeit der berechneten Radiosity-Lösungen zur Approximation der tatsächlichen Beleuchtungsfunktionen von Objektoberflächen nicht den heutigen Anforderungen.

Zur besseren Ausnutzung der guten Skalierbarkeitseigenschaften von Parallelrechnersystemen mit verteiltem Speicher werden asynchrone datenparallele Ansätze benötigt, welche lokale Eigenschaften der Radiosity-Methode ausnutzen, um den Kommunikationsbedarf so gering wie möglich zu halten. In der Vergangenheit wurden nur sehr wenige Parallelisierungsstrategien dieser Art in Veröffentlichungen beschrieben. Eine effiziente Realisierung eines asynchronen datenparallelen Verfahrens läßt sich nur unter Berücksichtigung zahlreicher spezieller Aspekte der Parallelverarbeitung durchführen. Als erstes wurde eine solche Strategie von Arnaldi u.a. [APV91] sowie von van Liere [vLiere91] beschrieben.

Beide Verfahren teilen das Radiosity-Gleichungssystem in eine Menge unabhängiger Gleichungssysteme für Teilszenen (lokale Umgebungen) der vollständigen Eingabeszene auf. Die Begrenzungen der lokalen Umgebungen (sogenannte Virtuelle Wände) werden in den lokalen Gleichungssystemen als *virtuelle Patches* berücksichtigt, welche die Energie speichern, um sie an die benachbarten Teilszenen abzugeben. Eine verteilte Berechnung der globalen Radiosity-Lösung wird ermöglicht, indem die Patch-Radiosities der lokalen Umgebungen mit Hilfe der lokalen Gleichungssysteme berechnet und die akkumulierten Energien der virtuellen Patches in die benachbarten Umgebungen abgegeben werden. Hierbei können die Gleichungssysteme mit Hilfe eines Iterationsverfahrens von verschiedenen Prozessoren unabhängig voneinander gelöst werden. Der Energieaustausch über die virtuellen Patches bildet einen Synchronisationspunkt zwischen den Prozessoren.

Die Verfahren, die auf einer Aufteilung der Szenengeometrie beruhen, unterscheiden sich hauptsächlich in der Art und Weise, wie die gegenseitigen Einflüsse der lokalen Umgebungen nachgebildet werden, wobei alle bisherigen Parallelisierungen den tatsächlichen Energieaustausch zwischen Patches verschiedener Teilszenen nur mehr oder weniger gut approximieren. Durch eine feinere Aufteilung der Begrenzungswände in virtuelle Patches kann die Qualität der Approximation verbessert werden, da die Richtung der Energie, welche eine lokale Umgebung verläßt, genauer berücksichtigt wird. Der gegenseitige Einfluß der Beleuchtungssituationen der Teilszenen kann jedoch nicht ohne Fehler berücksichtigt werden. Der Fehler vergrößert sich mit der Anzahl von Partitionen, in die das Radiosity-Gleichungssystem unterteilt wird. Somit wird der Fehler um so größer, je mehr Prozessoren am Lösungsprozeß beteiligt sind.

Basierend auf der Partitionierungsidee von Arnaldi u.a. wurde eine Parallelisierungsstrategie für das Progressive-Refinement-Verfahren entwickelt, welche die virtuellen Wände lediglich

als Kommunikations-Interfaces zwischen benachbarten Teilszenen interpretiert [Schm94]. Im Gegensatz zu der Arbeit von Arnaldi u.a. werden die virtuellen Wände nicht als expliziter Empfänger oder Sender von Energie betrachtet, sondern definieren lediglich relative Sichtbarkeitsbeziehungen zwischen Teilszenen. Das Verfahren verwendet die Hemicube-Methode [CG85] zur Bestimmung von Formfaktoren zwischen Patches innerhalb der einzelnen Teilszenen, sowie zur Formfaktorbestimmung zwischen Patches aus verschiedenen Teilszenen. Die parallel berechnete Radiosity-Lösung weist weder Qualitätseinbußen im Vergleich zu einer sequentiell berechneten Lösung auf, noch wird während der datenparallelen Berechnungen ein großer Kommunikations-Overhead verursacht. Die grundlegende Neuerung bei diesem Ansatz ist, daß zur parallelen Berechnung der Hemicube-Projektionen keine Szenendaten zwischen den Prozessoren ausgetauscht werden müssen (wie bei [CP91][FP91]). Dieses wird durch den Austausch von Sichtbarkeitsinformationen zwischen den Prozessoren eines Parallelrechners mit verteiltem Speicher ermöglicht. Hierbei sind Informationen über die Belegung der Hemicube-Pixel durch lokale Patchprojektionen in sogenannten Hemicube-Bitmaps kodiert. Weitere Einzelheiten zu dieser Parallelisierungsstrategie werden in dem folgenden Abschnitt 5.2 angegeben. Ein den Hemicube-Bitmaps ähnliches Konzept (die sogenannten *Visibility-Masks*) wurde später von Arnaldi u.a. [APRP96][RAPP97] für ein Shared-Memory-System vorgestellt.

Ein Hauptproblem bei der Parallelisierung der Radiosity-Methode ist ein Lastungleichgewicht, welches dynamisch im Verlauf der parallelen Berechnungen aufgrund der unterschiedlichen Berechnungszeiten für Formfaktoren und adaptiver Substrukturierung in verschiedenen Teilszenen entsteht. Die Problematik des dynamischen Lastausgleiches wurde neben der im weiteren Verlauf dieser Arbeit präsentierten Strategie [SR99] bisher nur in einer Arbeit von Stürzlinger u.a. [SSV95] explizit betrachtet. Die Effizienz der von Stürzlinger u.a. vorgestellte Methode verringert sich jedoch deutlich, wenn die Prozessorenanzahl vergrößert wird. Dieses widerspricht jedoch dem Grundgedanken einer dynamischen Lastverteilungsstrategie, welche ja gerade die effiziente Nutzung einer größeren Anzahl von Prozessoren ermöglichen soll.

Eine weitere Problemstellung, welche bisher bei sämtlichen vorgestellten Verfahren zur parallelen Berechnung von Radiosity-Lösungen in komplexen Umgebungen vollständig vernachlässigt wurde, ist die Unterstützung von dynamisch veränderbaren Szenen (siehe Kapitel 4). Obwohl die Autoren der wesentlichen Veröffentlichungen zu dieser Thematik [Chen90][GSG90] die Parallelisierung ihrer Methoden als erfolgversprechende Möglichkeit bezeichnen, um kurze Aktualisierungszeiten der Radiosity-Lösung bzgl. einer veränderten Szenengeometrie auch in komplexen Szenen zu garantieren, wurde bisher kein solches Verfahren vorgestellt.

In den folgenden Abschnitten dieses Kapitels werden zwei neue datenparallele Strategien zur Berechnung von Radiosity-Lösungen in dynamisch veränderbaren Szenen beschrieben und die erzielten Ergebnisse präsentiert. Es ist zu beachten, daß diese beiden parallelen Verfahren grundsätzlich verschiedene Ansätze zur Bestimmung von der Formfaktoren verwenden. Die Parallelisierungsstrategien weichen ebenfalls erheblich voneinander ab. Bei der Darstellung der beiden Verfahren wird jeweils sowohl auf wesentlichen Einzelheiten der grundlegenden Radiosity-Methode (z.B. Strategie der Bestimmung von Formfaktoren und adaptive Substrukturierung) eingegangen, als auch die speziellen Aspekte der Parallelisierung detailliert beschrieben und analysiert.

5.2 Ein datenparalleles Radiosity-Verfahren auf Basis von Hemicube-Bitmaps

Der größte Teil der existierenden Parallelisierungsstrategien für die Radiosity-Methode basiert auf den zuvor beschriebenen Master/Worker-Ansätzen. Existierende Parallelisierungsstrategien, die auf einer verteilten Speicherung der Szenendaten basieren, sind entweder durch einen sehr großen Kommunikations-Overhead gekennzeichnet oder es wird eine Lösung berechnet, die von einer sequentiell berechneten Lösung abweicht, wobei der Fehler mit steigender Anzahl von Prozessoren größer wird. Somit besteht erheblicher Bedarf an einer Parallelisierungsstrategie, die datenparallel arbeitet, einen geringen Kommunikations-Overhead aufweist und die parallele Berechnung einer exakten Radiosity-Lösung ermöglicht. Diese sollte selbst bei großen Anzahlen von Prozessoren keinen bzw. einen möglichst kleinen Fehler im Vergleich zur entsprechenden sequentiell berechneten Lösung aufweisen.

Der in diesem Abschnitt beschriebene Parallelisierungsansatz basiert auf der von Arnaldi u.a. [APV91] und von van Liere [vLiere91] beschriebenen Grundidee, die Eingabeszene in kleinere Teilszenen zu zerlegen und für diese parallel lokale Radiosity-Lösungen zu berechnen. Die Radiosity-Lösung für die gesamte Szene setzt sich dann aus den berechneten Teillösungen zusammen. Das grundsätzliche Problem bei den Ansätzen von Arnaldi u.a. und van Liere ist die Berücksichtigung von Wechselwirkungen in der Beleuchtung der Teilszenen. Die dort gewählten Lösungsansätze führen zu erheblichen Fehlern in den berechneten Lösungen, da Formfaktoren zwischen Flächen unterschiedlicher Teilszenen nicht direkt bestimmt werden. Es wird lediglich ein akkumulierter Energietransfer über virtuelle Begrenzungsflächen der Teilszenen durchgeführt, wobei die Informationen über die Richtung der Lichtstrahlung beim Übergang zwischen zwei Teilszenen verloren geht. Das hier beschriebene Verfahren ermöglicht die fehlerfreie Simulation dieser Wechselwirkungen (Austausch von Lichtenergie zwischen den Teilszenen). Dieses wird möglich, indem die Formfaktoren zwischen Flächen verschiedener Teilszenen analog zu den sequentiellen Berechnungen korrekt approximiert werden, wobei nur eine geringe Datenmenge zwischen den an dem Berechnungsprozeß beteiligten Prozessoren ausgetauscht wird.

Die datenparallele Strategie zur Berechnung von Radiosity-Lösungen wurde erstmalig in ihren Grundzügen in [Schm94] beschrieben und basiert auf dem Progressive-Refinement-Verfahren [CCWG88] (siehe auch Abschnitt 3.2). Die grundlegende Idee dieses Verfahrens besteht darin, die vorgegebene Szenenbeschreibung gemäß einer geometrischen Objektraumaufteilung auf die verschiedenen Prozessoren eines Parallelrechners zu verteilen. Innerhalb der Partitionen werden asynchron lokale Iterationen des Progressive-Refinement-Verfahrens ausgeführt. Partitionsübergreifende Beleuchtungseffekte werden hierbei durch Message-Passing berücksichtigt. Zu Zwecken der Bildgenerierung werden die berechneten Teillösungen auf einem Frontend-Rechner des Parallelrechners zusammengefügt (siehe Abb. 5.1).

Der beschriebene Ansatz aus [Schm94] konnte bzgl. dessen Skalierbarkeitseigenschaften und der Qualität der berechneten Radiosity-Lösung erheblich verbessert werden. Die überarbeitete Parallelisierungsstrategie wurde in [SL97][SRRP98] veröffentlicht. Hierbei sind die speziellen Aspekte

- der statischen Lastverteilung (Aufteilung der Szenendaten) und des Mappings,
- das gewählte Approximationsverfahren für Formfaktoren,

- der Energietransfer zwischen den Teilszenen,
- der dynamische Lastausgleich zwischen Prozessoren,
- der Erweiterung für dynamisch veränderbare Szenen sowie
- die Integration von Spiegeleffekten

von besonderer Wichtigkeit sind. Auf diese Punkte soll nun im Einzelnen eingegangen werden.

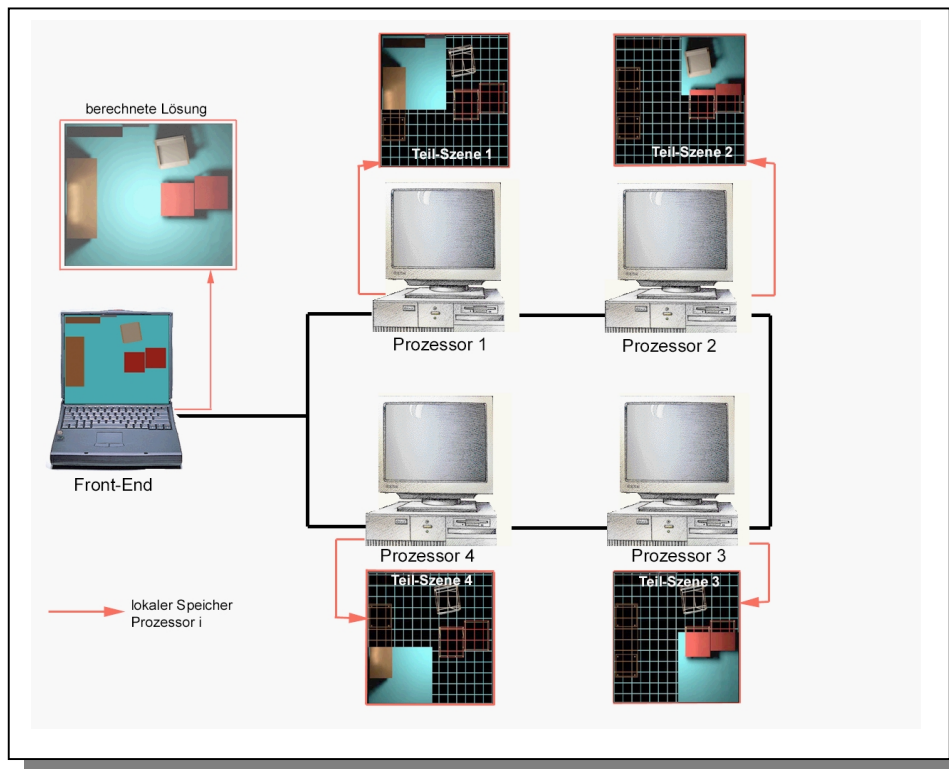


Abb. 5.1: Aufteilen der Eingabeszene in Teilszenen und asynchrone parallele Radiosity-Simulation.

5.2.1 Statische Lastverteilung

Das in diesem Abschnitt beschriebene datenparallele Verfahren basiert auf einer geometrischen Aufteilung des Objektraumes. Ziel ist die effiziente Ausnutzung von Systemen mit verteiltem Speicher, indem eine gleichmäßige Verteilung der Szenengeometrie auf die Prozessoren eines Parallelrechners vorgenommen wird. Eine *gute* Partitionierung der Szene muß im allgemeinen zwei Anforderungen genügen, die nur sehr schwer gleichzeitig zu erfüllen sind: Eine hohe *Datenlokalität* und eine gleichmäßige *statische Lastverteilung* bei den datenparallelen Radiosity-Berechnungen.

Die Partitionierung der Szene soll die Ausnutzung von Datenlokalität bei den Radiosity-Berechnungen ermöglichen, um den Kommunikationsbedarf des parallelen Verfahrens zu reduzieren. Radiosity-Berechnungen sollen durch die Prozessoren möglichst lokal durchgeführt werden können, ohne das große Datenmengen benötigt werden, die aufgrund der verteilten Speicherung der Szenengeometrie nicht lokale zur Verfügung stehen. Auf der anderen

Seite soll eine gute initiale statische Verteilung der Last erreicht werden, indem die Szenendaten möglichst gleichmäßig auf die Prozessoren verteilt werden. Hierdurch soll verhindert werden, daß die Berechnungszeiten der Radiosity-Lösungen für die durch Partitionierung der Eingabeszene gebildeten Teilszenen zu stark voneinander abweichen. Andernfalls wird die Effizienz der parallelen Berechnungen durch Wartezeiten (*Idle-Zeiten*) von Prozessoren deutlich verringert. Der Berechnungsaufwand der Radiosity-Methode korreliert mit der Anzahl der in der Szene enthaltenen Flächen. Ziel der initialen statischen Lastverteilung muß es somit sein, durch die Partitionierung der Eingabeszene Teilszenen zu generieren, die annähernd die gleiche Anzahl von Flächenelementen (Patches) enthalten.

Um eine hohe Datenlokalität zu erzielen, wird die Eingabeszene in aneinandergrenzende Partitionen (Zellen) unterteilt. Diese Zellen sind dreidimensionale Begrenzungskörper, die Teile der Szene einschließen, in denen lokale Radiosity-Berechnungen weitestgehend unabhängig voneinander durchgeführt werden können. Jedem Prozessor des Parallelrechners wird die Geometrie genau einer Zelle zugeordnet. Ziel ist es, eine gleichmäßige Objektraumaufteilung einer beliebigen Szene in eine vorgegebene Anzahl von Teilszenen vorzunehmen, wobei jede Zelle eine annähernd gleiche Anzahl von Flächenelementen einschließt. Dieses ist jedoch im allgemeinen nur möglich, wenn beliebig geformte Begrenzungskörper verwendet werden, um eine Unterteilung der Szene vorzunehmen.

Eine effiziente Ausnutzung von Datenlokalität bei den parallelen Radiosity-Berechnungen bedeutet insbesondere, daß der Formfaktor zwischen zwei Patches einer Teilszene durch einen Prozessor berechnet werden kann, ohne das mit anderen Prozessoren des Rechnersystems kommuniziert werden muß. Die Korrektheit einer solchen lokalen Formfaktorberechnung ist nur garantiert, wenn alle Flächen, welche die relative Sichtbarkeit zwischen diesen beiden Patches beeinflussen können, in der gleichen Teilszene enthalten sind. Dieses ist auf jeden Fall möglich, wenn die Teilszenen innerhalb von geschlossenen konvexen Zellen angeordnet sind.

Die Definition eines geschlossenen konvexen Polytops legt fest, daß die gedachte Verbindungslinie zwischen zwei beliebigen Punkten auf der Hülle des Polytops (und somit auch im Inneren) vollständig innerhalb des Polytops verläuft. Übertragen auf die Sichtbarkeitsbestimmung zwischen Punkten im Raum bedeutet dieses, daß die relative Sichtbarkeit der Punkte nur durch Objekte beeinflusst werden kann, die ganz oder teilweise innerhalb der konvexen Hülle dieser Punkte liegen.

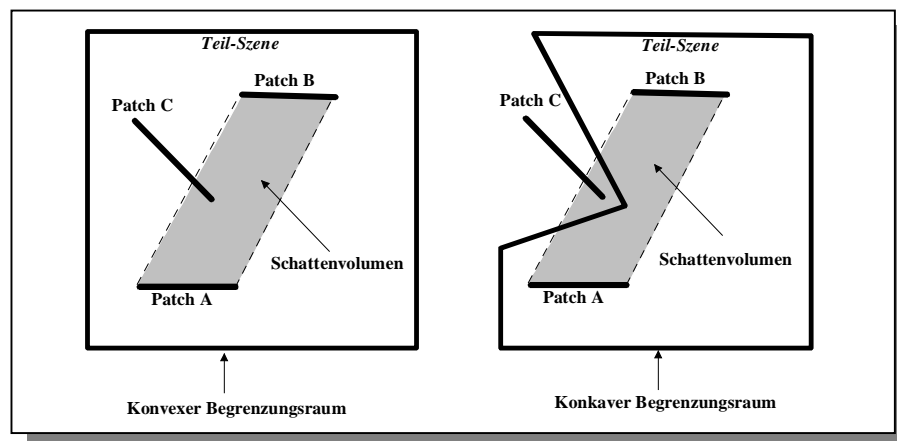


Abb. 5.2: Konvexe und konkave Begrenzungskörper für Teilszenen.

In Abb. 5.2 ist ein zweidimensionales Beispiel angegeben, welches die Problematik der Bestimmung von lokalen Formfaktoren innerhalb einer Teilszene veranschaulicht. Zwischen Patch *A* und Patch *B* der Teilszene soll ein Formfaktor berechnet werden. Wie in Kapitel 3 gezeigt wurde, hängt ein Formfaktor neben der relativen geometrischen Anordnung zweier Patches auch von der relativer Sichtbarkeit der Patches ab. Die Sichtbarkeit kann nur von Flächen beeinflusst werden, die in das Schattenvolumen der Patches ragen. Ein Schattenvolumen ist hierbei das kleinste Begrenzungsvolumen, das beide Patches einschließt. In dem angegebenen Beispiel liegt das Patch *C* teilweise innerhalb des Schattenvolumens von Patch *A* und Patch *B*. Somit wird ein Teil der Energie, die zwischen Patch *A* und Patch *B* ausgetauscht wird, von Patch *C* abgeblockt. Dies ist ein Aspekt, der durch alle Methoden zur Bestimmung von Formfaktoren entweder durch Raytracing-Techniken oder die Hemicube-Methode berücksichtigt wird. Wenn die Teilszenen, wie im linken Bild angegeben, innerhalb von konvexen Begrenzungskörpern angeordnet sind und jedes Patch der Szene vollständig innerhalb einer solchen Zelle liegt, kann kein Patch einer benachbarten Teilszene in das Schattenvolumen zwischen Patch *A* und Patch *B* ragen. Dieses gilt für sämtliche Formfaktoren, die zwischen den Patches der entsprechenden Teilszene bestimmt werden müssen. Wenn jedoch konkave Begrenzungskörper bei der Partitionierung zugelassen werden, kann wie im linken Bild gezeigt, der Fall auftreten, daß ein Patch einer benachbarten Teilszene teilweise oder vollständig innerhalb des Schattenvolumens zwischen Patch *A* und Patch *B* liegt. In diesem Fall ist eine korrekte Formfaktorberechnung ohne Kommunikation nicht möglich, da die entsprechende Verdeckung bei der Approximation der zwischen Patch *A* und Patch *B* ausgetauschten Energiemenge berücksichtigt werden muß.

Eine optimale Ausnutzung der Datenlokalität ist somit nur dann möglich, wenn die Szene in konvexe Zellen partitioniert wird.

Wenn Formfaktoren innerhalb einer Teilszene lokal ohne Kommunikation bestimmt werden sollen, muß bei der Verwendung von konvexen Zellen eine weitere Bedingung bei der Partitionierung der Szene beachtet werden. Jedes Patch der Szene muß mit seiner vollständigen Fläche innerhalb genau einer Zelle der Partitionierung liegen. Nur in diesem Fall können die Formfaktoren zwischen den Patches einer Teilszene unabhängig von der Geometrie der angrenzenden Teilszenen berechnet werden. Andernfalls können Situationen auftreten, wie in Abb. 5.3 durch ein zweidimensionales Beispiel skizziert.

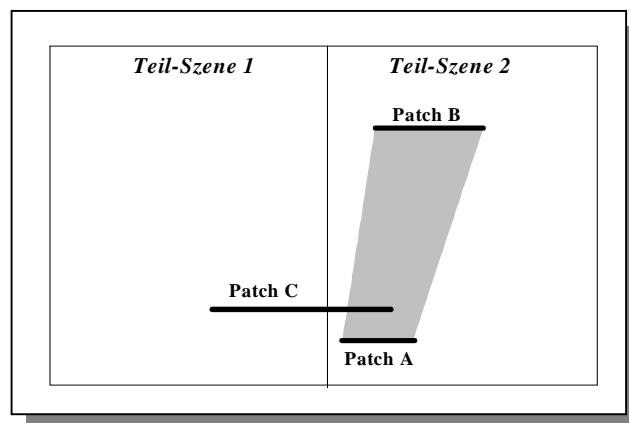


Abb. 5.3: Fehler bei lokalen Formfaktorberechnungen durch Patches, die über Zell-Grenzen hinausragen.

Das Patch *C* der Teilszene 1 schneidet die Begrenzung der Zellen von Teilszene 1 und Teilszene 2. Die Geometrie des Patches *C* ist jedoch nur dem Prozessor bekannt, welchem Teilszene 1 zugeordnet wurde. Wenn innerhalb der Teilszene 2 ein Formfaktor zwischen Patch *A* und Patch *B* berechnet wird, ist Patch *C* lokal unbekannt und kann somit bei der Bestimmung der relativen Sichtbarkeit zwischen Patch *A* und Patch *B* nicht berücksichtigt werden. Dieses kann zu erheblichen Approximationsfehlern führen. Eine Lösung des Problems besteht in der Kommunikation von Geometriedaten zwischen benachbarten Zellen. Eine solche Vorgehensweise widerspricht jedoch dem Ziel der optimalen Ausnutzung von Datenlokalität bei den parallelen Radiosity-Berechnungen.

Es ist zu beachten, daß unter einer optimalen Ausnutzung der Datenlokalität lediglich zu verstehen ist, daß die Formfaktoren zwischen den Patches einer Teilszene unabhängig von den durch andere Prozessoren verwalteten Geometriedaten berechnet werden können. Ist dieses gewährleistet, können lokale Progressive-Refinement-Berechnungen innerhalb der Teilszenen ohne Kommunikation durchgeführt werden. Zur Berechnung einer global korrekten Radiosity-Lösung muß jedoch auch ein Energieaustausch zwischen Patches verschiedener Teilszenen ermöglicht werden. Hierzu müssen Formfaktoren zwischen Patches berechnet werden, die nicht zu einer Teilszene gehören. Dieses ist bei einem datenparallelen Ansatz nur mit Hilfe von Kommunikation zwischen den Prozessoren zu realisieren. Wenn eine bessere Ausnutzung der Datenlokalität durch die erzeugte Partitionierung der Eingabeszene ermöglicht wird, verringert sich daher der gesamte Kommunikations-Overhead des datenparallelen Verfahrens.

Aus den vorangegangenen Betrachtungen ergeben sich somit die drei folgenden Anforderungen an eine Partitionierung der Eingabeszene, die im Rahmen der statischen Lastverteilung für das hier beschriebene Verfahren durchgeführt wird:

1. Die Begrenzungsvolumen (Zellen) von Teilszenen sind konvex.
2. Die Partitionierung erzeugt eine gute Verteilung der initialen Last. Insbesondere sollen die Geometriedaten der Szene möglichst gleichmäßig auf den zur Verfügung stehenden verteilten Speicher aufgeteilt werden.
3. Jedes Patch der Eingabeszene liegt mit seiner vollständigen Fläche innerhalb einer eindeutig bestimmten Zellen.

Die einfachste Möglichkeit zur Partitionierung der Eingabeszene in konvexe Zellen ist die Durchführung einer rekursiven Unterteilung einer großen Zelle in jeweils zwei kleinere Zellen (siehe Abb. 5.4). Hierbei wird zunächst ein konvexes Begrenzungsvolumen bestimmt, welches die Patches der gesamten Szene enthält. Dieses Begrenzungsvolumen wird rekursiv in zwei Teilvolumen zerlegt, wobei versucht wird die Unterteilung derart vorzunehmen, daß die Last (Anzahl der Patches) gleichmäßig auf die Zellen verteilt wird.

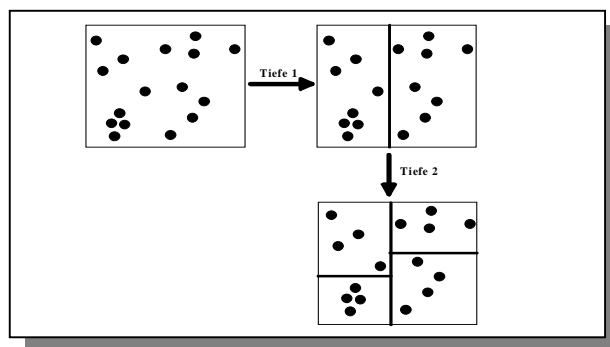


Abb. 5.4: Rekursive Partitionierung der Szene.

Das Problem bei dieser Vorgehensweise besteht darin, daß eine gute initiale Verteilung der Last nur für 2^n ($n=1,2,3,\dots$) Prozessoren erzielt werden kann. In allen anderen Fällen existieren Zellen, die aufgrund verschiedener Rekursionstiefen bei dem Unterteilungsprozeß deutlich voneinander abweichende Anzahlen von Patches enthalten, wodurch ein statisches Ungleichgewicht der Last verursacht wird.

Grundlage der statischen Lastverteilung der in diesem Abschnitt beschriebenen Parallelisierung ist die *dimensionsweise Partitionierung*. Bei diesem Ansatz wird die Szene in P Teilszenen zerlegt, die in aneinandergrenzenden konvexen Zellen liegen. Die Anzahl der Zellen der Partitionierung entlang der Dimensionen des kartesischen Koordinatensystems wird durch das Zahlen-Tripel (P_x, P_y, P_z) beschrieben, wobei $P=P_x P_y P_z$ gilt.

Zunächst wird ein quaderförmiges Begrenzungsvolumen der Szene bestimmt. Die Kanten des Begrenzungsvolumens verlaufen dabei parallel zu den Koordinatenachsen des kartesischen Koordinatensystems. In der x-Dimension wird das globale Begrenzungsvolumen mit Hilfe von P_x-1 zur y/z-Ebene parallelen Ebenen unterteilt, wodurch sich P_x konvexe Zellen ergeben. Jede dieser Zellen kann unabhängig voneinander entlang der y-Dimension durch zur x/z-Ebene parallele Ebenen unterteilt werden. Insgesamt werden P_y-1 dieser Begrenzungsebenen in der y-Dimension eingefügt. Nach diesem Schritt wurde die Szene in $P_x P_y$ konvexe Zellen zerlegt. Als letzter Schritt wird analog eine Unterteilung der bisher gebildeten Zellen entlang der z-Dimension vorgenommen, wodurch insgesamt $P_x P_y P_z$ entstehen (siehe Abb. 5.5).

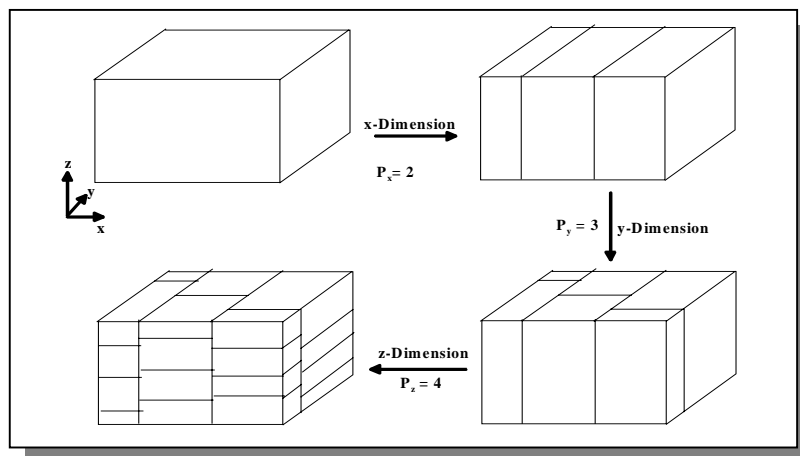


Abb. 5.5: Dimensionsweise Partitionierung der Szene.

Die entscheidende Frage ist nun, wie die Positionen der Unterteilungsebenen in den einzelnen Dimensionen bestimmt werden. Um eine effiziente Auslastung der Prozessoren während der datenparallelen Radiosity-Berechnungen zu erreichen, müssen die Patches der Eingabeszene derart auf die Prozessoren verteilt werden, daß die Berechnungszeit auf allen Prozessoren annähernd gleich sind. Somit wird ein Maß für den Berechnungsaufwand in einer generierten Teilszene benötigt. Der hier beschriebenen Parallelisierung liegt die Progressive-Refinement-Methode zugrunde (siehe Abschnitt 3.2). Der Berechnungsaufwand dieser Methode für die Durchführung einer einzelnen Iteration beträgt $O(n)$, wobei n die Anzahl der in der Szene enthaltenen Patches ist. Somit hängt die gesamte Berechnungszeit von der Anzahl der Patches in der Szene und von der Anzahl der Iterationen ab. Die Anzahl der Iterationen ist wiederum abhängig von der Energiemenge, die in der Szene zu verteilen ist, bis eine vorgegebene Konvergenzschranke unterschritten wird. Jedem Patch A_i wird ein Gewicht $w(A_i)$ zugeordnet. Dieses Gewicht beschreibt den Berechnungsaufwand, den Patch A_i bei der Durchführung einer

Iteration verursacht. Da dieses Gewicht a priori nicht bekannt ist, wird eine einfache Heuristik verwendet. Unter der Annahme, daß während jeder Iteration annähernd die gleiche Energiemenge verteilt und für jedes der Patches eine adaptive Substrukturierung der gleichen Tiefe durchgeführt wird, kann für $w(A_i)$ der Wert Eins gewählt werden ($i=1, \dots, n$). In diesem Fall werden Teilszenen gesucht, die aus annähernd der gleichen Anzahl von Patches bestehen. Dieses ist eine Heuristik, welche den tatsächlichen Berechnungsaufwand innerhalb der Teilszenen nur grob annähert, aber im allgemeinen eine gute statische Lastverteilung ermöglicht. Eine genauere Approximation ist möglich, wenn durch die Gewichtungsfunktion der Flächeninhalt der Patches und deren primäre Energiemengen berücksichtigt werden.

Eine statische Lastverteilung durch die zuvor beschriebene dimensionsweise Partitionierung der Szene läßt sich mit Hilfe von Präfixsummen der Gewichtungsfunktion der Patches bestimmen. Eine Szene, deren Objektoberflächen in n Patches A_1, \dots, A_n diskretisiert wurden, soll in $P=P_x P_y P_z$ konvexe Zelle C_1, \dots, C_P zerlegt werden. Für die summierten Gewichte der Patches in jeder Zelle C_j ($1 \leq j \leq P$) soll hierbei gelten:

$$\sum_{A_i \in C_j} w(A_i) \approx \frac{\sum_{i=1}^n w(A_i)}{P} \quad (5.1)$$

Zunächst werden die Schwerpunkte der Patches bestimmt. Zur Partitionierung der Szene entlang der x-Dimension in P_x Zellen wird eine Sortierung der Patches gemäß der x-Koordinaten der Schwerpunkte bestimmt. Nun wird für jedes Patch A_i ($i=1, \dots, n$) dessen Präfixsumme

$$s_i = \sum_{j=1}^i w(A_j) \quad (5.2)$$

innerhalb der Sortierung bestimmt. Mit Hilfe dieser Präfixsummen lassen sich nun die Positionen der P_x-1 Begrenzungsebenen parallel zur y/z-Ebene bestimmen. Durch die Aufteilung sollen Zellen gebildet werden, die Teilszenen enthalten, deren aufsummierte Gewichtungsfunktionen der Patches annähernd gleich sind.

Sei nun

$$W = \sum_{i=1}^n w(A_i) \quad (5.3)$$

die gesamte zu verteilende statische Last. Es werden nun Indizes i_1, \dots, i_{P_x-1} in der sortierten Patchliste bestimmt, für die gilt:

$$\left(s_{i_{j-1}} < \frac{jW}{P_x} \right) \wedge \left(s_{i_j} \geq \frac{jW}{P_x} \right) \quad (5.4)$$

Diese Indizes lassen sich in $O(n)$ bestimmen.

Durch SP_i sei nun der Schwerpunkt von Patch A_i bezeichnet. Die Partitionierungsebenen verlaufen nun parallel zur y/z-Ebene durch die Punkte

$$X_j = \frac{(SP_{i_{j-1}} + SP_{i_j})}{2}, \quad 1 \leq j < P_x. \quad (5.5)$$

Ein Patch wird einer Zelle eindeutig zugeordnet, sobald dessen Mittelpunkt zwischen den entsprechenden Begrenzungsebenen liegt.

Analog wird nun jede der im ersten Schritt bestimmten Zellen in P_y Zellen entlang der y-Dimension aufgeteilt. Hierbei erfolgt die Sortierung der Schwerpunkte, die Berechnung der Präfixsummen und die Bestimmung der Partitionierungsgrenzen für jede der zuvor bestimmten P_x Zellen separat. Anschließend wird entsprechend eine Unterteilung der $P_x P_y$ Zellen in jeweils P_z Zellen entlang der z-Dimension vorgenommen, wodurch insgesamt $P_x P_y P_z$ quaderförmige konvexe Zellen entstehen.

Die erzeugte Partitionierung erfüllt zwei der drei gestellten Bedingungen an eine gute Partitionierung. Die Begrenzungskörper der Teilszenen sind konvex und die initiale statische Last wird gemäß einer vorgegebenen Gewichtungsfunktion gleichmäßig auf die Zellen der Partitionierung verteilt. Bei diesem Ansatz ist in der beschriebenen Form noch nicht gewährleistet, daß jedes Patch der Szene mit seiner vollständigen Fläche in genau einer Zelle liegt. Bei der beschriebenen Partitionierungsmethode wurde ein Patch einer Zelle zugeordnet, sobald der Schwerpunkt des Patches innerhalb eines Begrenzungsraumes lag. Somit können Teile des Patches über die Zellengrenzen hinausragen. Dieses kann jedoch, wie zuvor dargestellt, zu Fehlern bei der Bestimmung von Formfaktoren führen.

Ein einfacher Ansatz, dieses Problem im Anschluß an die Partitionierung zu beheben, besteht darin, betroffene Patches an den Zellengrenzen zu zerteilen (*Clipping*), so daß die entstehenden Teilflächen eindeutig einer Zelle zugeordnet werden können. Problematisch bei dieser Vorgehensweise ist die Tatsache, daß aufgrund der Partitionierung der Eingabeszene das initiale Polygonnetz (Radiosity-Netz) modifiziert werden muß. Die durch die Partitionierung bedingte Änderung des initialen Radiosity-Netzes variiert mit der Anzahl der zu generierenden Teilszenen (und somit der Anzahl von Prozessoren), da die Begrenzungen der Zellen jeweils anders positioniert werden, um die Last gleichmäßig zu verteilen. Insbesondere bei der Verwendung von vielen Prozessoren wird das Radiosity-Netz unnötigerweise sehr komplex, da in diesem Fall mehr Zellengrenzen existieren, an denen Patches zerteilt werden müssen. Weiterhin sind die Berechnungszeiten von parallelen Berechnungen und die Simulationsergebnisse nur noch bedingt vergleichbar, wenn die Diskretisierung von Objektoberflächen einer Eingabeszene bei verschiedenen Berechnungen voneinander abweichen.

Eine andere Möglichkeit, die dritte Anforderung an eine gute Partitionierung zu erfüllen, besteht in einer *vorausschauenden* Netzgenerierung. Hierbei wird die gesamte Szene bei der initialen Netzgenerierung (siehe Kapitel 3) derart in Patches unterteilt, daß bei der statischen Lastverteilung auf jeden Fall eine Aufteilung in Teilszenen vorgenommen werden kann, so daß alle gestellten Bedingungen an eine zulässige Partitionierung für beliebige Anzahlen von Prozessoren erfüllt sind. Zu diesem Zweck wird ein zweidimensionales (m,n) -Gitter definiert, wobei m und n ganze Zahlen sind, welche die Anzahl der Gitterzellen entlang der entsprechenden Dimension angeben. Die Kantenlänge jeder Gitterzelle ist in beiden Dimensionen durch $l \in \mathbb{R}$ gegeben. Die Ausdehnung des Gitters entspricht der Ausdehnung einer gegebenen Szene in der xy-Ebene. Die Gitterstruktur wird orthogonal zur xy-Ebene auf die Objektoberflächen der Szene projiziert. Bei der initialen Netzgenerierung werden die Objektoberflächen entlang der projizierten Kanten der Gitterstruktur in kleinere Polygone unterteilt (siehe Abb. 5.6). Es ist zu beachten, daß hierbei Polygone mit mehr als vier Eckpunkten entstehen können, da die Objekte einer Szene nicht notwendigerweise an der Gitterstruktur ausgerichtet

sind. Die entsprechenden Polygone werden in Dreiecke oder Vierecke unterteilt. Weiterhin kann bei der Generierung einer Patchunterteilung für Objektflächen, die nicht parallel zur xy -Ebene verlaufen, Patches mit einer Kantenlänge größer als $2l$ entstehen. Diese Patches werden durch Anwendung eines Standard-Netzgenerierungsverfahrens in kleinere Patches unterteilt, so daß eine vorgegebene Kantenlänge nicht überschritten wird.

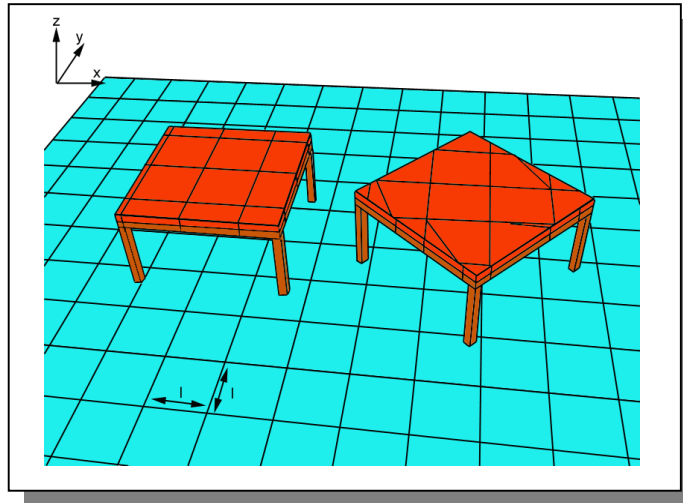


Abb. 5.6: Netzgenerierung entlang der Kanten eines projizierten 2D-Gitters.

In Abb. 5.6 ist ein einfaches Beispiel für eine solche vorausschauende Netzgenerierung angegeben. Es ist deutlich zu erkennen, daß die Kanten der Patch der Bodenfläche vollständig dem Verlauf von projizierten Kanten des zweidimensionalen Gitters folgen. Die Oberfläche des linken Tisches konnte ebenfalls in eine regelmäßige Patchstruktur unterteilt werden, da die Seitenkanten des Tisches parallel zu den Kanten des Gitters verlaufen und die Oberfläche des Tisches parallel zur xy -Ebene liegt, in der auch das Gitter angeordnet wurde. Die Seitenkanten der rechten Tischoberfläche verlaufen nicht parallel zu den Kanten des Gitters. Bei der Projektion der Gitterstruktur auf diese Oberfläche werden am Objektrand Polygone mit mehr als vier Eckpunkten erzeugt. Diese wurden in einem zusätzlichen Netzgenerierungsschritt in Vierecke und Dreiecke zerlegt, so daß eine zulässige Patchunterteilung der Tischoberfläche entsteht.

Nach der Durchführung der vorausschauenden Netzgenerierung kann die beschriebene dimensionsweise Partitionierung der Szene erfolgen. Damit garantiert wird, daß jedes Patch mit seiner vollständigen Fläche innerhalb einer der generierten Zelle liegt, muß bei der Festlegung der Partitionierungsebenen darauf geachtet werden, daß diese nur entlang der bei der Netzgenerierung verwendeten Gitterkanten verlaufen dürfen. Dieses ist einfach zu garantieren, indem die Bestimmung der Ebenengleichungen mit Hilfe der Gleichung (5.5) derart modifiziert wird, daß zunächst die Gitterkanten des (m,n) -Gitters bestimmt werden, die in der entsprechenden Dimension durch die berechneten Intervalle verlaufen. Die Partitionierungsebenen werden dann orthogonal zur xy -Ebene entlang dieser Gitterkanten orientiert. Es ist zu beachten, daß bei der hier beschriebenen Netzgenerierung eine Partitionierung der Szene nur in der x - und y -Dimension vorgenommen werden kann, da die initiale Netzgenerierung die Patches an einem zur xy -Ebene parallelen, zweidimensionalen Gitter ausgerichtet wurden. Dieses ist jedoch einfach zu erweitern, indem bei der Netzgenerierung eine zusätzliche Gitterstruktur orthogonal zur yz - und oder zur xz -Ebene auf die Oberflächen projiziert wird.

Die statische Aufteilung der Last, die mit Hilfe dieses gekoppelten Netzgenerierungs- und Partitionierungsverfahrens vorgenommen wird, ist im allgemeinen nicht optimal. Weiterhin erfüllt das initiale Netz nicht alle Gütekriterien, die an ein *gutes* Netz (siehe Kapitel 3) gestellt werden, in gleichem Maße. Gegenüber anderen Partitionierungsverfahren bietet es jedoch den Vorteil, daß Partitionierungen für beliebige Anzahlen von Prozessoren korrekt durchgeführt werden können, wobei ein hohes Maß an Datenlokalität bei den parallelen Berechnungen gewährleistet ist. Der wesentliche Vorteil ist jedoch, daß parallele Radiosity-Berechnungen für verschiedene Anzahlen von Prozessoren auf dem gleichen Eingabe-Netz durchgeführt werden können, wodurch die Berechnungszeiten und Berechnungsergebnisse vergleichbar sind.

Prozessoren	<i>Büro-Szene</i> (1809 Patches)		<i>Townhouse-Szene</i> (40518 Patches)		<i>Fürstenallee-Szene</i> (110439 Patches)	
	Verlauf der Partitionsgrenzen		Verlauf der Partitionsgrenzen		Verlauf der Partitionsgrenzen	
	beliebig	Gitterkanten	beliebig	Gitterkanten	beliebig	Gitterkanten
2	1,5 %	6,7 %	0,0 %	1,2 %		
4	3,4 %	6,4 %	0,1 %	7,5 %		
6	6,2 %	10,1 %	0,2 %	13,0 %		
8	7,5 %	13,3 %	0,1 %	8,6 %		
16			0,3 %	14,4 %		
24			0,9 %	15,5 %	0,4 %	20,4 %
30			1 %	22,5 %	0,2 %	20,2 %
36			0,6 %	23,6 %	0,2 %	27,2 %
40			0,9 %	21,8 %	0,4 %	22,3 %

Tabelle 5.1: Durchschnittliche Abweichung der Anzahl der statisch zugewiesenen Patches von einer optimalen Zuordnung.

In Tabelle 5.1 wurde für drei Eingabeszenen unterschiedlicher Komplexität und verschiedenen Anzahlen von Prozessoren eine statische Aufteilung der Geometriedaten (Patches) auf die Prozessoren bestimmt. Es wurde das zuvor beschriebene dimensionsweise Partitionierungsverfahren verwendet, wobei einmal die Partitionierungsgrenzen beliebig plziert wurden und einmal nur entlang der Gitterkanten verlaufen durften. Die Tabelle gibt die jeweilige durchschnittliche Abweichung der den Prozessoren zugewiesenen Patchanzahlen von einer optimalen Aufteilung wieder. Es ist deutlich zu erkennen, daß durch das kombinierte Netzgenerierungs- und Partitionierungsverfahren Szenenaufteilungen erzeugt werden, die deutlich von der optimalen statischen Lastverteilung abweichen. Dieses gilt insbesondere für größere Anzahlen von Prozessoren. Offensichtlich lassen sich komplexe Szenen besser partitionieren als kleine Szenen. In der Büro Szene, die initial in 1809 Patches unterteilt wurde, konnte selbst bei einer beliebigen Positionierung der Zell-Grenzen keine absolut gleichmäßige Verteilung der Last erzielt werden. Die Situation verbessert sich jedoch erheblich, wenn die Komplexität der Eingabeszene vergrößert wird. Dieses ist dadurch zu Begründen, daß sich ein statisches Ungleichgewicht, welches durch Objekte mit einer großen lokalen Komplexität in Szenen mit hoher Gesamtkomplexität besser ausgleichen läßt. Wenn bei der Partitionierung die Begrenzungsebenen nur entlang der Kanten des zweidimensionalen Projektions-Gitters verlaufen dürfen, bleibt die durchschnittliche Abweichung von der optimalen Lastverteilung etwa auf dem gleichen Niveau (20% - 30%), wenn komplexe Szenen auf eine größere Anzahl von Prozessoren (20-40) verteilt werden.

Während der parallelen Radiosity-Berechnungen kann aufgrund von adaptiver Substrukturierung der Patches und unterschiedlichen Energieverteilungen in den Teilszenen ein dynamisches Lastungleichgewicht zwischen den Prozessoren entstehen. Dieses läßt sich im Rahmen

eines statischen Lastverteilungsverfahrens nur sehr grob durch einfache Heuristiken annähern. Weiterhin ist es bei Szenen, die einige wenige Objekte mit großer lokaler Komplexität enthalten, nicht möglich, eine gute statische Lastverteilung durch Partitionierung der Szene in konvexe Zelle zu berechnen. Aus diesem Grund ist es unerlässlich, auf unterschiedliche Lastsituationen, die sich im Verlauf der Berechnungen für verschiedene Prozessoren eines Parallelrechners ergeben, mit Hilfe eines Verfahrens zum dynamischen Lastausgleich zu reagieren.

5.2.2 Lokale Radiosity-Berechnungen innerhalb der Teilszenen

Die Radiosity-Berechnungen innerhalb der durch die statische Lastverteilung generierten Teilszenen erfolgen auf Basis des in Abschnitt 3.2 beschriebenen Progressive-Refinement-Verfahrens [CCWG88]. Ein entscheidender Aspekt hierbei ist die möglichst genaue Approximation der Energiemenge, die von einem bestimmten Patch (Shooting-Patch) an die übrigen Patches der Umgebung abgegeben wird. Zur Bestimmung der Formfaktoren stehen, wie in Abschnitt 3.1 gezeigt, verschiedene Techniken zur Verfügung. Die Güte dieser Verfahren unterscheiden sich hinsichtlich ihrer Approximationsgenauigkeit und der Berechnungsgeschwindigkeit. Die in [Schm94] beschriebene Parallelisierung der Radiosity-Methode verwendet die Hemicube-Methode [CG85] (siehe auch 3.1.2.3) zur Formfaktorbestimmung. Mit Hilfe dieses Verfahrens können die Formfaktoren zwar schnell bestimmt werden, die Genauigkeit ist jedoch im allgemeinen nicht ausreichend, um eine adäquate Radiosity-Lösung zu berechnen [BRW89]. Aus diesem Grund muß ein exakteres Verfahren als Grundlage der parallelen Radiosity-Berechnungen gewählt werden, welches einerseits eine möglichst exakte Approximation des Energieaustausches zwischen den Patches ermöglicht. Andererseits muß sich das Verfahren effizient in den gewählten Parallelisierungsansatz integrieren lassen.

5.2.2.1 Bestimmung analytischer Formfaktoren

Die analytische Bestimmung von Formfaktoren nach Baum u.a. [BRW89] bietet die Möglichkeit, eine sehr genaue Approximation der Formfaktoren zwischen zwei Patches vorzunehmen (siehe 3.1.2.4). Die relative Sichtbarkeit zwischen zwei Patches erfolgt zwar mit Hilfe des Projektionsansatzes der Hemicube-Methode, die eigentliche Bestimmung des entsprechenden Formfaktors wird hingegen analytisch durchgeführt. Da der ursprüngliche Parallelisierungsansatz aus [Schm94] darauf basiert, die relative Sichtbarkeit zwischen Patches unterschiedlicher Teilszenen mit Hilfe von Hemicube-Projektionen zu bestimmen, kann das parallele Verfahren dahingehend modifiziert werden, daß die analytische Methode von Baum u.a. zur Berechnung von Formfaktoren verwendet wird. Der Kommunikations-Overhead bei den parallelen Radiosity-Berechnungen bleibt hierbei unverändert, die Berechnungskomplexität nimmt jedoch erheblich zu. Somit kann durch Verwendung der Methode von Baum u.a. sowohl die Lösungsqualität des parallelen Verfahrens gesteigert als auch die Effizienz der parallelen Berechnungen verbessert werden.

Zum besseren Verständnis der Darstellungen in den folgenden Abschnitten soll nun kurz die Vorgehensweise bei der Bestimmung der Formfaktoren mit Hilfe der analytischen Methode innerhalb einer Teilszene skizziert werden.

Nach der Auswahl eines Shooting-Patches innerhalb der lokalen Teilszene wird ein Hemicube über Schwerpunkt des Shooting-Patches positioniert. Die Patches der Teilszene werden auf die in Pixel diskretisierten Seitenflächen des Hemicube projiziert. Nach Durchführung dieser Projektionen ist in jedem Pixel der Hemicube-Seiten die relative Distanz zwischen dem Schwerpunkt des Shooting-Patches und der in der entsprechenden Richtung sichtbaren Oberflächen gespeichert. Diese Vorgehensweise entspricht soweit der normalen Hemicube-

Methode. Die eigentliche Formfaktorbestimmung der analytischen Methode, die im Anschluß an die Sichtbarkeitsberechnung durchgeführt wird, unterscheidet sich jedoch grundlegend von der Hemicube-Methode.

Wie in 3.1.2.4 beschrieben, basiert die Approximation des Patch-zu-Patch Formfaktors auf der analytischen Berechnung von Punkt-zu-Patch Formfaktoren für mehrere Sample-Punkte. Mit einer größer werdenden Anzahl von Sample-Punkten nimmt hierbei die Genauigkeit der bestimmten Formfaktoren zu. Zu diesem Zweck werden auf jedem Empfänger-Patch i eine fest vorgegebene Anzahl von m Sample-Punkten s_1, \dots, s_m plazierte. Ziel ist es nun, gemäß der Gleichung (3.15) einen analytischen Punkt-zu-Patch Formfaktor $F_{s_k j}$ zwischen den Sample-Punkten s_k ($k=1, \dots, m$) und dem Shooting-Patches j zu berechnen. Anschließend wird mit Hilfe von Gleichung (3.16) der Patch-zu-Patch Formfaktor F_{ij} zwischen Patch i und Patch j bestimmt, wobei die jeweilige relative Sichtbarkeit zwischen den Sample-Punkten s_k und dem Shooting-Patch j berücksichtigt werden muß (Sichtbarkeitsfaktor $V_{s_k j}$).

Mit Hilfe der in den Hemicube-Pixeln gespeicherten Distanzwerten läßt sich nun einfach bestimmen, ob ein Sample-Punkt s_k auf einem Empfänger-Patch, von dem Schwerpunkt des Shooting-Patches aus betrachtet, sichtbar ist. Hierzu muß der Punkt s_k lediglich auf den Hemicube projiziert werden. Anschließend wird ein Vergleich zwischen dem in dem entsprechenden Hemicube-Pixel gespeicherten Distanzwert mit der Entfernungen zwischen s_k und dem Schwerpunkt des Shooting-Patches durchgeführt. Ist die in einem Hemicube-Pixel gespeicherte Distanz geringer, liegt eine andere Oberfläche der Szene näher zum Projektionszentrum als der Sample-Punkt s_k . Somit ist die Sichtbarkeit zwischen s_k und dem Schwerpunkt des Shooting-Patches blockiert. In diesem Fall wird der Sichtbarkeitsfaktor $V_{s_k j}$ auf den Wert Null gesetzt, andernfalls auf den Wert Eins.

Es ist zu beachten, daß bei dieser Vorgehensweise ein Formfaktor von Empfänger-Patch i und Shooting-Patch j und somit F_{ji} berechnet wird. Für die Progressive-Refinement-Methode wird jedoch der umgekehrte Formfaktor F_{ij} benötigt. Dieser läßt sich mit Hilfe der reziproken Beziehung für Formfaktoren direkt aus F_{ji} bestimmen (siehe 3.1.1). Problematisch ist hierbei, daß die relative Sichtbarkeit bzgl. eines Punktes auf dem Shooting-Patch berechnet wird. Dieses kann zu Approximationsfehlern führen, welche mit Hilfe eines in [BRW89] angegebenen Verfahrens erkannt und verhindert werden können. In einem erkannten Fehlerfall kann eine adaptive Unterteilung des Shooting-Patches vorgenommen und eine erneute Formfaktorberechnung bzgl. der erzeugten kleineren Teilflächen des Shooting-Patches durchgeführt werden.

5.2.2.2 Adaptive Verfeinerung des Polygonnetzes

Ein weiterer entscheidender Aspekt bei der Berechnung einer Radiosity-Lösung, welche die tatsächlichen Beleuchtungsfunktionen von Objektoberflächen möglichst genau approximieren soll, ist die adaptive Unterteilung von Empfänger-Patches (siehe 3.3.2). Wesentlich für die Durchführung der adaptiven Substrukturierung eines Patches ist das Kriterium, das als Entscheidungsgrundlage dient, ob ein bestimmtes Patch weiter unterteilt werden muß. Ein Patch soll genau dann weiter unterteilt werden, wenn die Approximation der spezifischen Ausstrahlung eines Patches durch einen konstanten Wert einen zu großen Fehler in der Radiosity-Lösung verursacht. Dieses ist der Fall, wenn über die Fläche des Patches betrachtet, deutliche Unterschiede in der Beleuchtung auftreten. Ein Energieaustausch wird dann rekursiv zwischen dem aktuellen Shooting-Patch und den durch die Patch-Verfeinerung gebildeten Sub-Patches durchgeführt.

Das Kriterium für die adaptive Unterteilung eines Patches, welches in dem hier beschriebenen Parallelisierungsansatz verwendet wird, orientiert sich an den berechneten Punkt-zu-Patch Formfaktoren zwischen den auf einem Empfänger-Patch i verteilten Sample-Punkten s_1, \dots, s_m und dem jeweiligen Shooting-Patch j . Zunächst werden der maximale und der minimale dieser Formfaktoren bestimmt mit:

$$F_{\min} = \underset{k \in \{1, \dots, m\}}{\text{MIN}}(F_{s_k j}) \quad (5.6)$$

und

$$F_{\max} = \underset{k \in \{1, \dots, m\}}{\text{MAX}}(F_{s_k j}) \quad (5.7)$$

Mit Hilfe dieser Werte läßt sich ein Näherungswert \tilde{r} für die maximale relative Abweichung r der bestimmten Punkt-zu-Patch Formfaktoren über der Fläche des Empfänger-Patches angeben. Es ist:

$$r \approx \tilde{r} = \frac{F_{\max} - F_{\min}}{F_{\max}} \quad (5.8)$$

Die maximale zulässige relative Abweichung ε ist vorgegeben. Eine Unterteilung des Patches wird vorgenommen, wenn $\tilde{r} > \varepsilon$. Die analytische Formfaktorberechnung wird in diesem Fall rekursiv für die erzeugten Sub-Patches durchgeführt, bis $\tilde{r} \leq \varepsilon$ ist.

Damit keine unendliche Unterteilung von Patches vorgenommen wird, deren Beleuchtungsfunktion eine Unstetigkeit aufweist (z.B. entlang Patches durch die eine Schattengrenze verläuft), wird ein minimaler Flächeninhalt vorgegeben, welcher bei der adaptiven Substrukturierung nicht unterschritten werden darf.

Der Vorteil des beschriebenen Substrukturierungskriteriums liegt darin, daß keine Radiosity-Werte an den Eckpunkten der Patches durch Interpolation der an die Eckpunkte angrenzenden Patch-Radiosities bestimmt werden müssen (siehe Substrukturierungskriterium in Abschnitt 3.3.2). Bei den datenparallelen Radiosity-Berechnungen sind nicht in allen Fällen die an einen Eckpunkt des Patch-Netzes angrenzenden Patches lokal verfügbar. Für die entlang der Zell-Grenzen angeordneten Patches könnte ein auf Eckpunkt-Radiosities basierendes Substrukturierungskriterium nur mit Hilfe von Kommunikation ausgewertet werden. Dieses würde jedoch zu einem erheblichen Kommunikations-Overhead führen.

5.2.3 Energietransfer zwischen Teilszenen

Bei dem hier beschriebenen Verfahren zur datenparallelen Berechnung einer Radiosity-Lösung sind verschiedene Prozessoren für die Berechnung von Radiosity-Lösungen in den während der statischen Lastverteilung definierten Teilszenen verantwortlich. Die parallele Berechnung einer global korrekten Radiosity-Lösung bedingt die Berücksichtigung von Beleuchtungseffekten, die über eine Teilszene hinaus Auswirkungen auf benachbarte Teilszenen haben. Somit müssen die Prozessoren kommunizieren, um diese Beleuchtungseffekte berücksichtigen zu können. Dieses kann anhand von zwei aneinandergrenzenden Räumen, die durch eine geöffnete Tür verbunden sind, veranschaulicht werden. Wenn nur in einem der beiden Räume eine Lichtquelle angeschaltet ist, fällt ein Teil der Lichtenergie durch die Tür in den

benachbarten Raum und beleuchtet diesen teilweise. Bei einer Aufteilung der Szene auf zwei Prozessoren, so daß jedem Prozessor genau ein Raum zugeordnet wird, enthält nur eine der Teilszenen eine initiale Lichtquelle. Somit kann eine Energieverteilung nur innerhalb der Teilszene mit der initialen Lichtquelle berechnet werden, wenn die Prozessoren Radiosity-Berechnungen unabhängig voneinander (ohne Kommunikation) durchführen. Die zweite Teilszene würde vollständig dunkel bleiben, da initial keine Energie in diesem Teil der Szene vorhanden ist. Dieses entspricht jedoch nicht der Realität (siehe Abb. 5.7).



Abb. 5.7: Beleuchtung eines Raumes durch die geöffnete Tür.

Das datenparallele Verfahren erlaubt den Austausch von Lichtenergie zwischen benachbarten Teilszenen. Hierbei kann auch eine *Energiediffusion* durch mehrere Teilszenen auftreten. Für die Effizienz der parallelen Berechnungen ist die Kommunikationsstruktur der Berechnungsprozesse von besonderer Bedeutung, wobei die Verbindungsstruktur der Prozessoren (Kommunikationshardware) optimal genutzt werden soll, um den Kommunikations-Overhead zu reduzieren. Die Kommunikationsverzögerung läßt sich im allgemeinen durch eine möglichst gute Abbildung der Berechnungsprozesse auf die Prozessoren unter Berücksichtigung der Verbindungsstruktur des Parallelrechners reduzieren.

5.2.3.1 Der Kommunikationsgraph des parallelen Programmes

Während der initialen statischen Lastverteilung werden konvexe Zellen C_1, \dots, C_P bestimmt, welche die Eingabeszene in Teilszenen annähernd gleicher Komplexität aufteilen. Jede der generierten Zellen wird durch 6 planare Polygone begrenzt. Zwei Zellen werden hierbei als direkt benachbart bezeichnet, wenn diese Zellen in einer zur xy -, yz - oder xz -Ebene parallelen Partitionierungsebene der Eingabeszene direkt aneinandergrenzen. In diesem Fall existiert für die beiden Zellen entweder ein gemeinsames begrenzendes Polygon oder es existieren zwei Begrenzungspolygone der Zellen, deren Schnittmenge wiederum ein Polygon ist. Eine direkte Nachbarschaftsbeziehung ist nicht gegeben, wenn zwei Zellen an einen gemeinsamen Punkt (Schnittpunkt von m Partitionierungsebenen im m -dimensionalen Raum) oder eine gemeinsame Geraden (Schnittpunkt von $m-1$ Begrenzungsebenen im m -dimensionalen Raum) angrenzen. Dieser Sachverhalt wird in dem zweidimensionalen Beispiel in Abb. 5.8 veranschaulicht. Dort werden direkte Nachbarschaftsbeziehungen über gemeinsame Segmente der Begrenzungsebenen im zweidimensionalen Raum definiert. Wenn Zellen an einen gemeinsamen Punkt angrenzen, impliziert dieses keine direkte Nachbarschaftsbeziehung. Somit ist die

Zelle 1 direkt benachbart zu den Zellen 0, 2, und 4. Zwischen Zelle 1 und 3 besteht keine direkte Nachbarschaftsbeziehung, da sie lediglich an einen Punkt im zweidimensionalen Raum angrenzen.

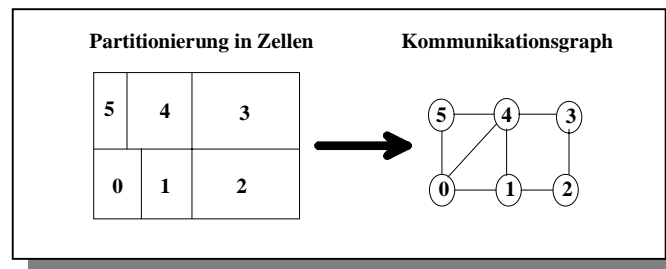


Abb. 5.8: Direkte Nachbarschaftsbeziehungen bei einer zweidimensionalen Partitionierung.

Für jede Teilszene, die innerhalb einer konvexen Zelle C_i ($i \in \{1, \dots, P\}$) liegt, berechnet ein Prozeß gemäß des Progressive-Refinement-Verfahrens eine Radiosity-Lösung. Zur Berechnung einer global korrekten Lösung kommunizieren Prozesse miteinander, die für die Berechnung von Teillösungen in direkt benachbarten Zellen verantwortlich sind.

Definition 5.1: (Kommunikationsgraph)

Sei $PART = \{C_1, \dots, C_P\}$ die Menge der während der statischen Lastverteilung generierten konvexen Zellen, welche paarweise disjunkte Teilszenen der Eingabeszene enthalten. Der Kommunikationsgraph $G = \{V, E\}$ des parallelen Programmes ist vollständig definiert durch die Knotenmenge $V = \{C_1, \dots, C_P\}$ und die Kantenmenge $E = \{\{C_i, C_j\}; i, j \in \{1, \dots, P\} \text{ mit } i \neq j \text{ und } C_i \text{ und } C_j \text{ sind direkt benachbarte Zellen}\}$.

Das *Einbettungsproblem*, das sich im Rahmen des datenparallelen Radiosity-Verfahrens stellt, beschäftigt sich mit der Frage der injektiven Abbildung von Prozessen auf Prozessoren. Da ein generierter Kommunikationsgraph und der Verbindungsgraph des verwendeten Parallelrechners im allgemeinen nicht isomorph sind, wird eine Einbettungsheuristik benötigt, welche Kanten des Kommunikationsgraphen auf Pfade des Verbindungsgraphen abbildet, wobei die Länge der Pfade minimiert werden soll. Eine umfangreiche Darstellung der Einbettungsproblematik ist in [Heiss94] zu finden.

5.2.3.2 Einbettung des Kommunikationsgraphen in einen Verbindungsgraphen

Ziel ist es nun, einen generierten Kommunikationsgraph (und somit die Teilszenen) injektiv auf die Prozessoren eines Parallelrechners abzubilden, so daß diese Abbildung gewissen Gütekriterien genügt. Die Abbildung soll gewährleisten, daß die Kommunikation zwischen den Berechnungsprozessen auf der entsprechenden Netzwerktopologie des Parallelrechners möglichst effizient ausgeführt werden kann. Die Topologie des Parallelrechners wird durch den sogenannten Verbindungsgraphen beschrieben. Die hier verwendeten Heuristiken basieren auf der Voraussetzung, daß die Topologie des Zielsystems durch ein zweidimensionales Gitter beschrieben wird.

Eine möglichst kurze Berechnungszeit ist eine wesentliche Voraussetzung, die durch eine Heuristik zur Einbettung des Kommunikationsgraphen in den Verbindungsgraphen erfüllt werden muß. Dieses liegt darin begründet, daß die Zeit zur Bestimmung einer Einbettung in den Gesamt-Overhead der parallelen Radiosity-Berechnungen eingeht.

Von Diekman u.a. wurde mit GEOM-MAP eine einfache, koordinatenbasierte Einbettungsheuristik für geometrische Graphen in Gitter vorgestellt [DMM95]. Dieses Verfahren bettet einen Graph, für den an Knoten zweidimensionale Koordinaten gegeben sind, in ein zweidimensionales Gitter ein. Die Grundidee von GEOM-MAP ist, daß ein in zwei Dimensionen angeordnete gitterförmige Struktur von Rechtecken über den zweidimensionalen Gastgraphen gelegt wird. Hierbei entspricht die Anzahl der Rechtecke in jeder Dimension der Anzahl der Knoten entlang der Dimensionen eines zweidimensionalen Gitters, welches die Prozessortopologie eines Parallelrechners beschreibt. Jeder Prozessor entspricht somit einem der Rechtecke. Die Grenzen der Rechtecke werden solange verschoben, bis jedes Rechteck genau einen Knoten des Gastgraphen enthält. Die dadurch entstandene Zuordnung von Knoten des Gastgraphen zu den Rechtecken beschreibt eine Einbettung des Gastgraphen in den Verbindungsgraphen (siehe Abb. 5.9).

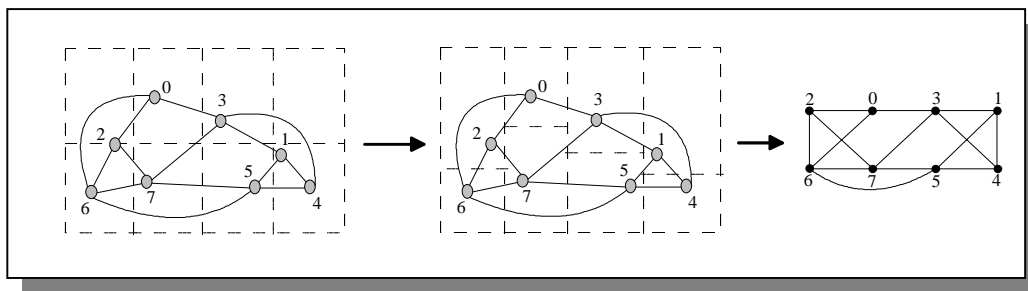


Abb. 5.9: Vorgehensweise des GEOM-MAP-Verfahrens.

Einbettungen können mit Hilfe von GEOM-MAP in sehr kurzer Zeit berechnet werden und weisen eine geringe maximale Kantenstreckung auf [DMM95].

Diese Methode wurde für den Einsatz zur Bestimmung von Einbettungen im Kontext der datenparallelen Radiosity-Berechnungen dahingehend erweitert, daß ein Gastgraph an dessen Knoten dreidimensionale Koordinaten gegeben sind, in ein zweidimensionales Gitter eingebettet werden kann. Zunächst werden den Knoten des Kommunikationsgraphen die kartesischen Koordinaten der Mittelpunkte der entsprechenden Zellen einer generierten Partitionierung zugewiesen. Anschließend werden die beiden Dimensionen des kartesischen Koordinatensystems bestimmt, entlang welcher die meisten Zellen der Partitionierung angeordnet sind. Dieses sind gerade die beiden größten Werte des Zahlentripels (P_x, P_y, P_z) , das die Partitionierung der Szene festlegt (siehe 5.2.1). Die entsprechenden Dimensionen legen eine Projektionsebene fest. Auf diese Ebene werden die Knoten des Kommunikationsgraphen orthogonal projiziert. Die Projektion erzeugt eine Punktmenge, die vollständig in der xy -, xz -, oder yz -Ebene liegt. Für jeden Punkt wird die Koordinate auf den Wert Null gesetzt, die der orthogonal zur Projektionsebene verlaufenden Dimension zugeordnet ist. Es ist zu beachten, daß während der Projektion partitionierungsbedingt mehrere Punkte auf den gleichen Punkt in der Projektionsebene abgebildet werden können. In diesem Fall werden verschiedene Punkte erzeugt, indem diese in einem engen Radius um den tatsächlichen Projektionspunkt auf der Projektionsebene zufällig generiert werden. Die den Knoten des Kommunikationsgraphen zugeordneten dreidimensionalen Koordinaten werden durch die zweidimensionalen Koordinaten der entsprechenden Projektion ersetzt. Für den resultierenden Kommunikationsgraphen mit zugewiesenen zweidimensionalen Koordinaten an den Knoten kann nun eine Einbettung in das Zielgitter mit Hilfe des GEOM-MAP-Verfahrens bestimmt werden.

Von Bokhari [Bokh81] wurde eine Heuristik vorgestellt, die auf der iterativen Verbesserung von zuvor generierten Einbettungen eines Gastgraphen in einen Wirtsgraphen basiert. Die mit Hilfe von GEOM-MAP berechnete Einbettung kann als Starteinbettung für das Bokhari-Verfahren verwendet werden, um eine verbesserte Einbettung zu generieren [DMM95]. Durch das Bokhari-Verfahren werden Einbettungen mit großer Kardinalität generiert, d.h. es wird eine große Anzahl von Kanten des Gastgraphen direkt auf Kanten des Wirtsgraphen abgebildet (Kanten werden mit Kantenstreckung Eins eingebettet). Dieses geht jedoch auf Kosten einiger Kanten, die mit großer Kantenstreckung eingebettet werden.

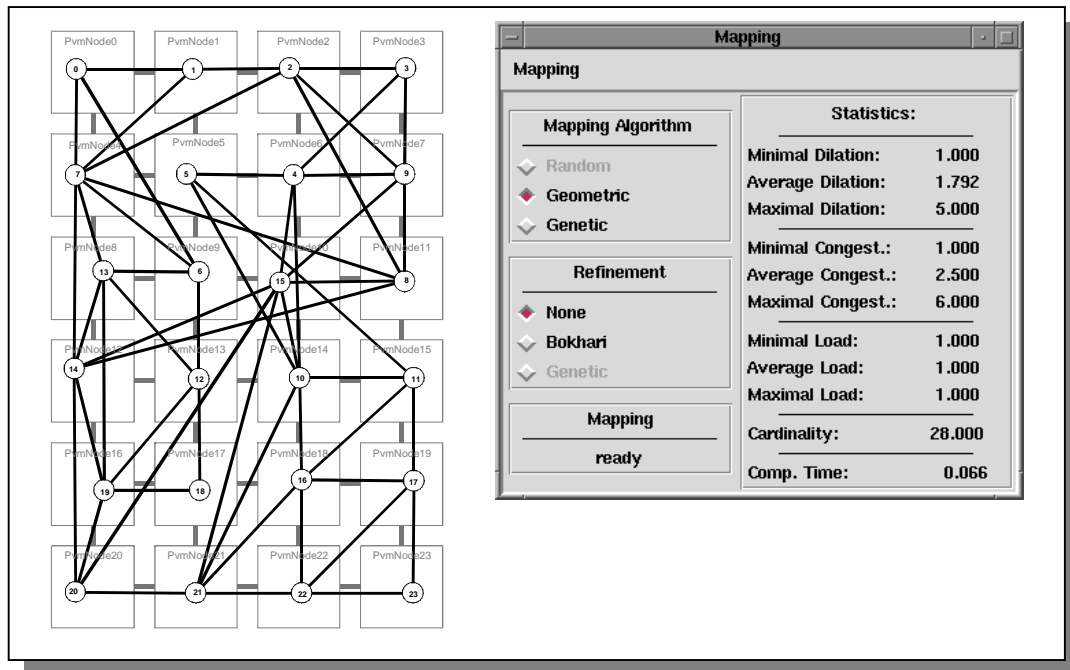


Abb. 5.10: Einbettung eines Kommunikationsgraphen (links) und die zugehörige Einbettungs-Statistik (rechts).

In Abb. 5.10 ist ein Beispiel für eine mit Hilfe des erweiterten GEOM-MAP-Verfahrens (*GeomMap3D*) bestimmte Einbettung eines Kommunikationsgraphen mit 24 Knoten in einen gitterförmigen 6x4 Verbindungsgraphen angegeben.

Durch eine Kombination von *GeomMap3D* mit der Einbettungsheuristik von Bokhari werden Einbettungen eines Kommunikationsgraphen in ein zweidimensionales Gitter ermöglicht, wobei die folgenden Qualitätskriterien (Zielfunktionen) einer Einbettung möglichst gut erfüllt werden sollen:

- Minimierung der maximalen Kantenstreckung (Ziel: Minimierung der Kommunikationsverzögerung)
- Minimierung der durchschnittlichen Kantenstreckung
- Maximierung der Anzahl der Kanten des Gastgraphen die direkt auf Kanten des Zielgraphen abgebildet werden (Kardinalität).

In Tabelle 5.2 und Tabelle 5.3 sind die Ergebnisse für die Berechnung von Einbettungen von zwei Kommunikationsgraphen auf Basis unterschiedlicher Szenenpartitionierungen in ein Zielgitter angegeben.

Verfahren	mittlere Dilation	maximale Dilation	Kardinalität	Berechnungszeit (Sekunden)
GeomMap3D	1,607	3	15	< 0,1
Geom3D+Bokhari	1,536	4	17	< 0,1
genetisches	1,321	2	19	33
	1,357	2	18	32
	1,321	2	19	16

Tabelle 5.2: Einbettung eines Graphen mit 16 Knoten in ein 4×4 Gitter.

Verfahren	mittlere Dilation	Maximale Dilation	Kardinalität	Berechnungszeit (Sekunden)
GeomMap3D	2,687	7	35	< 0,1
Geom3D+Bokhari	3,098	9	57	1
	2,571	7	54	1
	3,423	9	55	1
genetisch	2,067	5	54	116
	2,080	4	52	152
	2,043	4	50	156
	2,000	4	57	252

Tabelle 5.3: Einbettung eines Graphen mit 48 Knoten in ein 8×6 Gitter.

Den Ergebnissen der GeomMap3D-Heuristik bzw. dessen Kombination mit dem Bokhari-Verfahren sind Ergebnisse gegenübergestellt, die mit Hilfe einer genetischen Einbettungsheuristik erzielt wurden. Hierbei ist zu beachten, daß es sich bei den implementierten Algorithmen um Heuristiken handelt, die aufgrund von zufälligen Ereignissen bei verschiedenen Läufen zu unterschiedlichen Ergebnissen führen. Es ist deutlich ersichtlich, daß mit Hilfe eines genetischen Verfahrens einerseits die besten Ergebnisse bzgl. sämtlicher Zielfunktionen erzielt werden, andererseits die Berechnungszeiten jedoch deutlich größer sind als die der kombinierten Einbettungsheuristik.

Ein Vergleich von parallelen Berechnungszeiten von Radiosity-Lösungen auf Basis von Einbettungen in den Verbindungsgraphen eines Zielsystems (hier: GC/PP und CC-48 von Parsytec), die mit den verschiedenen Einbettungsheuristiken generiert wurden, hat keine signifikanten Unterschiede aufgezeigt. Schlußfolgerung hieraus ist, daß sich der Mehraufwand, der sich durch die Verwendung einer genetischen Einbettungsheuristik ergibt, im Kontext des hier beschriebenen datenparallelen Radiosity-Verfahrens nicht in Kauf genommen werden sollte. Dieses gilt insbesondere, wenn die Tatsache berücksichtigt wird, daß die Berechnungszeiten der genetischen Einbettungsheuristik bei der Verwendung einer größeren Anzahl von Prozessoren auf dem Zielsystem länger waren als die Zeit zur Durchführung der eigentlichen parallelen Radiosity-Berechnung.

5.2.3.3 Relative Sichtbarkeitsbeziehungen zwischen Teilszenen

Das hier beschriebene Verfahren basiert auf der parallelen Durchführung von Iterationen des Progressive-Refinement-Verfahrens innerhalb der Teilszenen einer, wie in Abschnitt 5.2.1 beschrieben, partitionierten Eingabeszene. Die Auswahl der Shooting-Patches, die Formfaktorberechnung zwischen den Patches der jeweiligen lokalen Teilszene und die anschließende

Verteilung der Energie gemäß der berechneten Formfaktoren erfolgt hierbei asynchron innerhalb der Teilszenen.

Für jedes der gewählten Shooting-Patches wird dessen bisher nicht verschossene spezifische Ausstrahlung zunächst lokal innerhalb der jeweiligen Teilszene verteilt. Die Lichtenergie eines Shooting-Patches hat jedoch nicht nur Auswirkungen auf die lokale Beleuchtungssituation, da Teile der spezifischen Ausstrahlung an Patches angrenzender Teilszenen übergeht. Um global korrekte Radiosity-Lösung zu bestimmen, muß die Energie eines Shooting-Patches in allen Teilszenen verteilt werden, welche Objekte enthalten, die in Abstrahlrichtung der spezifischen Ausstrahlung des Patches liegen und somit möglicherweise Energie empfangen können. Es ist zu beachten, daß somit bei der Bestimmung von Formfaktoren relative Sichtbarkeitsbeziehungen zwischen Objekten verschiedener Teilszenen ausgewertet werden müssen.

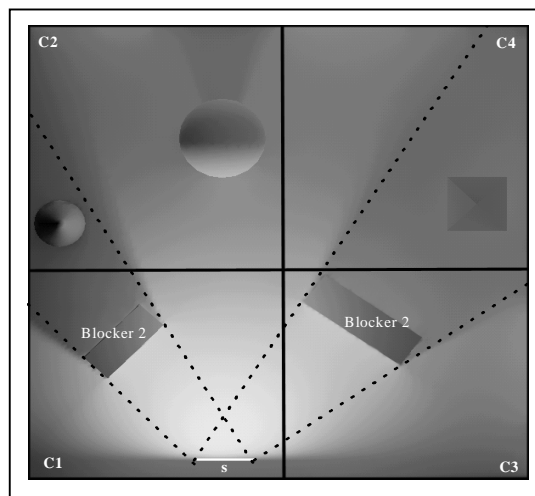


Abb. 5.11: Problematik der Formfaktorbestimmung bei verteilter Speicherung der Szene.

Ein Beispiel hierfür ist in Abb. 5.11 angegeben. In diesem Beispiel wurde eine Szene in vier Teilszenen zerlegt. Lediglich die Teilszenen innerhalb der Zelle C_1 enthält eine primäre Lichtquelle s . Es ist deutlich zu sehen, daß die beiden Objekte *Blocker 1* und *Blocker 2* durch die direkte Bestrahlung Schatten werfen, die nicht auf die jeweiligen lokalen Szenen C_1 bzw. C_3 beschränkt sind, sondern sich auf die Teilszenen innerhalb der Zellen C_2 und C_4 ausweiten. Somit ist es notwendig, bei der Verteilung der Energie der Lichtquelle s an die Patches der Teilszenen innerhalb von C_2 und C_4 Informationen über potentielle Schattenbildungen, die durch Objekte benachbarter Teilszenen verursacht werden, bei den lokalen Berechnungen zu berücksichtigen.

Bei der datenparallelen Berechnung von Formfaktoren müssen für ein gewähltes Shooting-Patch s die Teilszenen bestimmt werden, deren Objekte Lichtenergie von s empfangen können.

Jedes Patch wurde während der statischen Lastverteilung einer eindeutig bestimmten Teilszene zugeordnet, die innerhalb einer konvexen Zelle C_k ($k \in \{1, \dots, P\}$) liegt. Wenn das Patch s als lokales Shooting-Patch ausgewählt wird, kann sich dessen Energie potentiell in den Zellen ausbreiten, die ganz oder nur teilweise in dem Halbraum in Richtung des Normalenvektors von s liegen und durch die Ebene begrenzt ist in der Patch s liegt. Diese Zellen werden als die *sichtbare Zellen* von Patch s bezeichnet und sind durch einen einfachen Test zu identifizieren.

Dieser Test überprüft, ob mindestens ein Eckpunkt der konvexen Zelle in dem in Abstrahlrichtung von s liegenden Halbraum liegt.

In Abb. 5.12 ist ein Beispiel angegeben, welches diesen Sachverhalt veranschaulicht. Wenn die spezifische Ausstrahlung von Patch s in der gesamten Szene verteilt werden soll, können nur Objekte innerhalb der Zellen C_1 , C_2 , C_3 und C_4 diese empfangen.

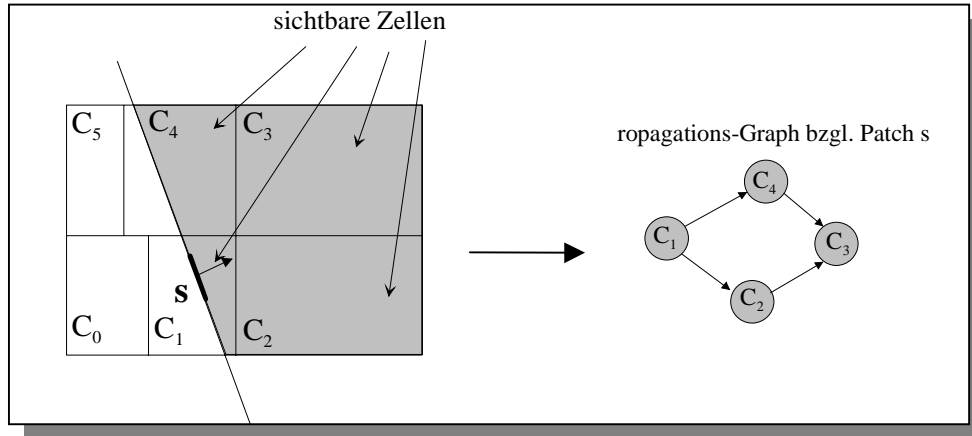


Abb. 5.12: Sichtbare Zellen von Patch s in einer zweidimensionalen Partitionierung (links) und der resultierende Propagationsgraph (rechts).

Definition 5.2: (Sichtbarkeitsrelation $<_s$)

Sei $PART = \{C_1, \dots, C_P\}$ die Menge der während der statischen Lastverteilung generierten konvexen Zellen, welche paarweise disjunkte Teilszenen der Eingabeszene enthalten. Es kann nun eine Sichtbarkeitsrelation $<_s$ auf $PART \times PART$ für ein beliebiges Patch s der Szene definiert werden. Für zwei sichtbare Zellen $C_i, C_j \in PART$ von Patch s gilt $C_i <_s C_j$ genau dann, wenn die relative Sichtbarkeit jedes beliebigen Punktes innerhalb von C_i und dem Patch s nicht durch ein innerhalb von C_j beliebig angeordnetes Objekt beeinflusst werden kann.

Für eine gegebene Partitionierung der Eingabeszene in konvexe quaderförmige Zellen C_1, \dots, C_P lassen sich die Sichtbarkeitsrelationen $C_i <_s C_j$ ($i, j \in \{1, \dots, P\}$) zwischen direkt benachbarten Zellen unter Berücksichtigung von Proposition 5.1 einfach bestimmen. Die direkte Nachbarschaftsbeziehung zwischen zwei Zellen bezieht sich hierbei analog zur Definition des Kommunikationsgraphen (Definition 5.1) auf Nachbarschaften bzgl. einer gemeinsamen Polygonfläche der Zell-Begrenzungen im dreidimensionalen Raum bzw. eines gemeinsamen Segmentes im zweidimensionalen Raum (siehe 5.2.3.1).

Proposition 5.1: Sei $PART = \{C_1, \dots, C_P\}$ die Menge der während der statischen Lastverteilung generierten konvexen quaderförmigen Zellen, die paarweise disjunkte Teilszenen der Eingabeszene enthalten. Seien $C_i, C_j \in PART$ zwei direkt benachbarte sichtbare Zellen eines Patches s , dann gilt:

Es existiert eine Ebene E_{ij} in \mathbb{R}^3 , die parallel zur xy -, xz - oder yz -Ebene verläuft und ein Separator für $C_i, C_j \in PART$ ist (d.h. C_i, C_j liegen in verschiedenen Halbräumen des durch E_{ij} aufgeteilten \mathbb{R}^3), so daß Patch s auf der gleichen Seite von E_{ij} liegt wie die Zelle C_i , genau dann wenn $C_i <_s C_j$.

Beweis:„ \Leftarrow “:

Seien $C_i, C_j \in \text{PART}$ zwei direkt benachbarte sichtbare Zellen von s , wobei s ein beliebiges Patch der Szene (planares Polygon mit 3 oder 4 Eckpunkten) ist und es gelte: $C_i <_s C_j$.

C_i und C_j wurden gebildet, indem ein konvexes quaderförmiges Begrenzungsvolumen der Eingabeszene in $P=P_xP_yP_z$ kleinere konvexe quaderförmige Zellen unterteilt wurde. Hierbei wurden Partitionierungsebenen parallel zu der xy -, xz - und yz -Ebene zur Unterteilung einer größeren Zelle in kleinere Zellen verwendet. Somit existiert für jedes beliebige Paar von Zellen mindestens eine Ebene parallel zur xy -, xz - oder yz -Ebene, welche den \mathbb{R}^3 derart in zwei Halbräume aufteilt, daß die beiden Zellen auf verschiedenen Seiten dieser Ebene liegen.

Insbesondere für zwei direkt benachbarte Zellen C_i und C_j existiert eine eindeutige Partitionierungsebene E_{ij} parallel zur xy -, xz - oder yz -Ebene, an welche die Zellen gemeinsam angrenzen. Somit existiert für jede der beide Zellen jeweils eine eindeutig bestimmte Begrenzungsfläche des konvexen Begrenzungskörpers, die innerhalb der Partitionierungsebene E_{ij} liegt. Der Schnitt dieser beiden Begrenzungsflächen der Zellen C_i und C_j innerhalb der Ebene ist ein planares Polygon (beachte: dieses war gerade die Definition einer direkten Nachbarschaftsbeziehung von Zellen der Partitionierung – Abschnitt 5.2.1).

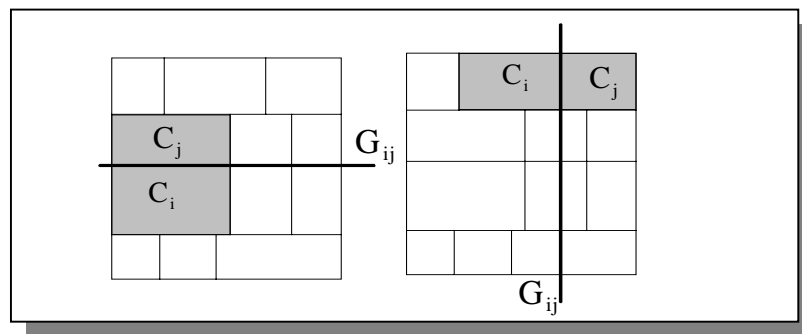


Abb. 5.13: Partitionierungsgeraden G_{ij} für direkt benachbarte Zellen C_i und C_j .

In Abb. 5.13 ist ein zweidimensionales Beispiel angegeben. Hier existiert für direkt benachbarte Zellen eine Partitionierungsgerade G_{ij} , welche die Ebene in zwei Halbebenen aufteilt, so daß die beiden Zellen in verschiedenen Halbebenen liegen.

Da laut Voraussetzung $C_i <_s C_j$ gilt, kann auf einer Geraden zwischen einem beliebigen Oberflächenpunkt des Patches s und einem beliebigen Punkt innerhalb von C_i kein Punkt aus dem Inneren der Zelle C_j liegen (siehe Definition 5.2). Angenommen das Patch s liegt auf derselben Seite der Partitionierungsebene E_{ij} wie die Zelle C_j . In diesem Fall kann jedes Objekt innerhalb von C_j , welches einen Schnittpunkt mit einer Geraden durch einen beliebigen Punkt auf Patch s und einen beliebigen Punkt auf der Fläche des Begrenzungspolygons zwischen C_i und C_j besitzt, die relative Sichtbarkeit zwischen Patch s und Punkten im Inneren der Zelle C_i verdecken (siehe 2D-Beispiel in Abb. 5.14).

Aus diesem Grund können Objekte innerhalb von C_j die relative Sichtbarkeit zwischen einem Punkt innerhalb von C_i und dem Patch s beeinträchtigen. Dieses steht jedoch im Widerspruch zur Voraussetzung $C_i <_s C_j$. Somit liegen s und C_i auf derselben Seite von E_{ij} .

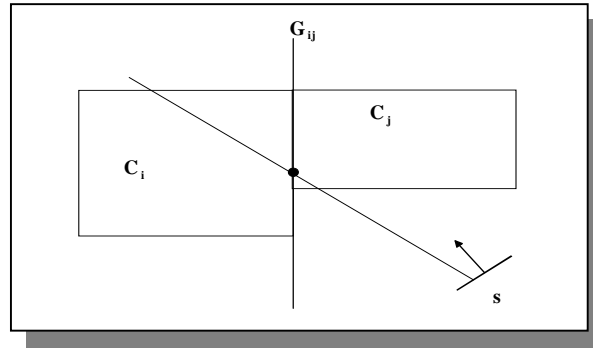


Abb. 5.14: Gerade durch einen Punkt auf Patch s und einem Punkt des gemeinsamen Begrenzungssegmentes zwischen C_i und C_j .

„ \Rightarrow “:

Annahme: Es existiert eine Ebene E_{ij} in \mathbb{R}^3 , die parallel zu xy -, xz - oder yz -Ebene verläuft, wobei die direkt benachbarten sichtbaren Zellen $C_i, C_j \in \text{PART}$ von Patch s auf verschiedenen Seiten von E_{ij} liegen und das Patch s auf der gleichen Seite von E_{ij} liegt wie die Zelle C_i . Unter diesen Voraussetzungen gilt nicht $C_i <_s C_j$.

In diesem Fall existieren Punkte P_i, P_j und P_s in \mathbb{R}^3 , wobei P_s ein Punkt des Patches s , P_i ein Punkt innerhalb C_i und P_j ein Punkt innerhalb C_j ist, für welche gilt:

Auf der Geraden durch die beiden Punkte P_s und P_i liegt der Punkt P_j zwischen P_s und P_i . In diesem Fall blockiert P_j die relative Sichtbarkeit zwischen P_s und P_i . Hieraus folgt, daß P_i, P_j und P_s auf der gleichen Seite von E_{ij} liegen müssen (beachte: die Zellen sind konvex). Dieses ist jedoch ein Widerspruch zur Voraussetzung, daß C_i und C_j auf verschiedenen Seiten von E_{ij} liegen.

■

Damit ist Proposition 5.1 bewiesen.

Aus Satz 5.1. folgt unmittelbar das folgende Korollar 5.1.

Korollar 5.1: Für zwei direkt benachbarte sichtbare Zellen $C_i, C_j \in \text{PART}$ von Patch gilt entweder $C_i <_s C_j$ oder $C_j <_s C_i$.

Definition 5.3: (Propagationsgraph eines Patches s)

Der Propagationsgraph eines beliebigen Patches s der Eingabeszene ist ein gerichteter Graph $G_s = (V_s, E_s)$ mit der Knotenmenge $V_s \subseteq \text{PART} = \{C_1, \dots, C_P\}$. V_s enthält alle sichtbaren Zellen des Patches s aus PART . Die Kantenmenge ist gegeben durch E_s , wobei gilt:

$E_s = \{(C_i, C_j); C_i, C_j \in \text{PART} (i \neq j) \text{ und } C_i, C_j \text{ sind direkt benachbart mit } C_i <_s C_j\}$.

Der Knoten, welcher die Zelle repräsentiert, in dessen Teilszene das Patch s liegt, wird als Ursprungsknoten des Propagationsgraphen bezeichnet.

In Abb. 5.12 ist im rechten Teil des Bildes ein einfaches Beispiel für einen Propagationsgraph angegeben.

Eine gerichtete Kante des Propagationsgraphen von Patch s gibt eine Beziehung an, die zwischen den in der Zelle des Startknotens und der Zelle des Zielknotens enthaltenen Teilszenen gilt. Die relative Sichtbarkeit eines Patches in der Zelle des Zielknotens und dem Patch s kann durch in der Zelle des Startknotens liegende Objekte beeinträchtigt werden, umgekehrt ist dieses jedoch nicht möglich. Soll nun ein Formfaktor F_{sp} zwischen einem beliebigen Patch s und einem Patch p der Teilszene in Zelle $C_i \in \text{PART}$ berechnet werden, können nur Objekte die relative Sichtbarkeit zwischen s und p beeinflussen, die zu Teilszenen der Zellen gehören, welche auf einem beliebigen Pfad vom Ursprungsknoten des Propagationsgraphen von s zum Knoten C_i liegen.

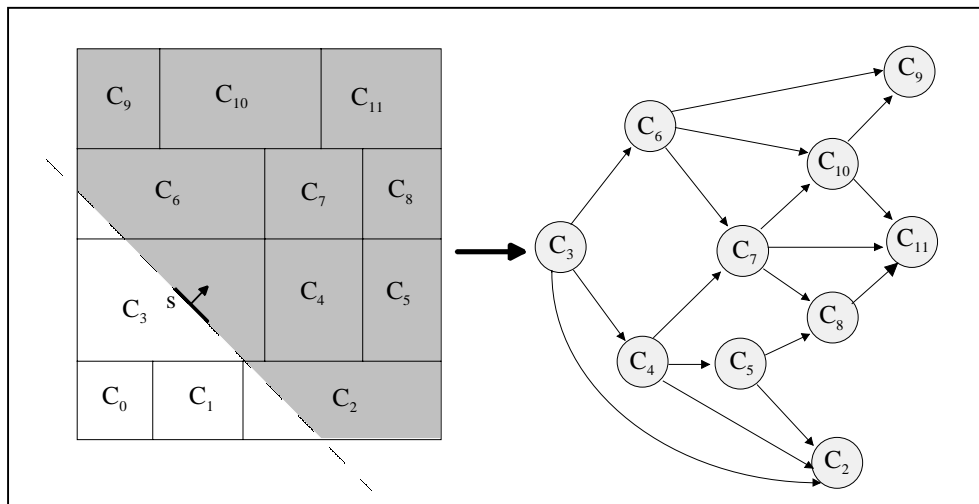


Abb. 5.15: Komplexer Propagationsgraph des Patches s in einer 2D-Partitionierung.

Ein komplexeres Beispiel für einen Propagationsgraphen ist in Abb. 5.15 angegeben. In diesem Beispiel müssen bei der Berechnung von Formfaktoren zwischen dem Shooting-Patch s und den Patches der Teilszene innerhalb der konvexen Zelle C_2 Verdeckungen berücksichtigt werden, die durch Objekte innerhalb der Zellen C_3 , C_4 , C_5 und C_2 verursacht werden können. Bei der Formfaktorberechnung zwischen Patch s und den Patches der Teilszene in Zelle C_5 können höchstens Objekte innerhalb der Zellen C_3 , C_4 und C_5 s verdecken. Für die Berechnung von Formfaktoren zwischen s und Patches innerhalb von Zelle C_7 sind die Teilszenen der Zellen C_3 , C_4 , C_6 und C_7 relevant.

Mit Hilfe von Proposition 5.1 und Korollar 5.1 läßt sich der Propagationsgraph eines beliebigen Patches s der Eingabeszene sehr einfach berechnen. Zunächst sind die sichtbaren Zellen von Patch s zu bestimmen (siehe oben). Für alle direkt benachbarten sichtbaren Zellen C_i und C_j wird nun gemäß Proposition 5.1 überprüft, ob $C_i <_s C_j$ gilt. Hierzu wird bestimmt, in welchem der durch die Partitionierungsebene E_{ij} erzeugten Halbräume das Patch s liegt. Liegen s und C_i im gleichen Halbraum bzgl. E_{ij} , gilt $C_i <_s C_j$ ansonsten gilt $C_j <_s C_i$ (siehe Korollar 5.1).

Es ist zu beachten, daß bei der Bestimmung des Propagationsgraphen eines Patches s lediglich die Anordnung der konvexen Zellen der Partitionierung einer Eingabeszene berücksichtigt wird, nicht jedoch der Inhalt der Zellen (d.h. die Teilszenen). Ziel ist es, eine pessimistische Abschätzung der möglichen gegenseitigen Verdeckungen zwischen den Teilszenen vorzunehmen. Diese Abschätzung soll eine Formfaktorberechnung zwischen Patches beliebiger

Teilszenen ermöglichen, ohne das Fehler bei der Überprüfung der relativen Sichtbarkeit von Patches aufgrund der verteilten Speicherung der Szene auftreten können.

Die Approximation der Sichtbarkeitsbeziehungen innerhalb der partitionierten Eingabeszene durch Propagationsgraphen ist als pessimistisch zu bezeichnen, da diese Abschätzung alle möglichen Anordnungen von Objekten innerhalb der Zellen abdeckt. Die tatsächliche Geometrie einer Teilszene, die in einer Zelle eingeschlossen ist, wird nicht betrachtet, da hierzu das vollständige Sichtbarkeitsproblem zwischen s und den Teilszenen gelöst werden muß. Dieses ist jedoch gerade das Problem, welches mit Hilfe des Propagationsgraphen bei verteilter Speicherung der Teilszenen gelöst werden soll.

5.2.3.4 Bestimmung von Formfaktoren bei einer verteilten Speicherung der Szene

Der Propagationsgraph eines Patches s kann dazu verwendet werden, Formfaktoren zwischen s und den übrigen Patches der verteilt gespeicherten Szene parallel zu berechnen. Die einem Prozessor des Parallelrechners zugeordnete Teilszene wird hierbei als *lokal* bzgl. dieses Prozessors bezeichnet.

Bei der Durchführung der parallelen Progressive-Refinement-Berechnungen wird unterschieden zwischen *lokalen* und *externen Shooting-Iterationen*, die von einem Prozessor ausgeführt werden. Eine Iteration wird als *lokale Shooting-Iteration* (oder kurz *lokales Shooting*) bezeichnet, wenn das entsprechende Shooting-Patch, dessen Energie zu verteilen ist, zur lokalen Teilszene des ausführenden Prozessors gehört. Das Patch wurde lokal bestimmt und dessen nicht verschossene spezifische Ausstrahlung bildet zum Auswahlzeitpunkt ein lokales Maximum innerhalb der Teilszene. Um die Auswirkungen der spezifischen Ausstrahlung dieses Patches auf die globale Beleuchtung innerhalb der gesamten Szene vollständig zu berücksichtigen, muß eine Verteilung dieser Energie auch innerhalb der Teilszenen durchgeführt werden, die in sichtbaren Zellen des Shooting-Patches liegen. Dieses wird im Rahmen von *externen Shooting-Iterationen* (oder kurz *externe Shootings*) realisiert, bei denen ein Shooting-Patch nicht zur lokalen Teilszene des ausführenden Prozessors gehört.

Während eines lokalen Shootings kann die Bestimmung von Formfaktoren und die entsprechende Verteilung der Energie des Shooting-Patches vollständig ohne Kommunikation erfolgen (siehe 5.2.1). Wenn ein Prozessor eine externe Shooting-Iteration durchführt, muß berücksichtigt werden, daß die relative Sichtbarkeit zwischen den Patches der lokalen Teilszene und dem entsprechenden externen Shooting-Patch durch Objekte beeinträchtigt werden kann, die nicht zur lokalen Teilszene gehören. Somit benötigt der Prozessor zur Formfaktorberechnung innerhalb der lokalen Teilszene zusätzliche Informationen, um eine korrekte Bestimmung der relativen Sichtbarkeit zwischen externem Shooting-Patch und lokalen Empfänger-Patches zu ermöglichen. Dieses läßt sich mittels Kommunikation realisieren, wobei es zwei verschiedene Ansätze gibt.

Eine Möglichkeit besteht darin, Geometriedaten der Teilszenen zwischen den Prozessoren auszutauschen. Dieser Ansatz führt jedoch zu einem extremen Kommunikations-Overhead, da gerade bei komplexen Szenen große Mengen von Geometriedaten zwischen den Prozessoren kommuniziert werden müssen. Ein anderer Ansatz, bei welchem deutlich geringere Datenmengen zwischen den Prozessoren ausgetauscht werden müssen, basiert auf dem Transfer eines ausgewählten Shooting-Patches inklusive berechneter Sichtbarkeitsinformationen.

Das hier vorgestellte datenparallele Radiosity-Verfahren verwendet die in 5.2.2.1 beschriebene Methode zur Berechnung von Formfaktoren. Als Grundlage der Bestimmung von relativen

Sichtbarkeiten zwischen einem Shooting-Patch s und Sample-Punkten auf den Empfänger-Patches dient hierbei die Hemicube-Technik. Somit müssen die Patches sämtlicher Teilszenen auf einen Hemicube projiziert werden, der oberhalb des Schwerpunktes von Patch s positioniert ist. Mit Hilfe der Z-Buffer-Technik werden die Patches identifiziert, welche von s aus betrachtet sichtbar sind. Diese sind gerade die Patches, die in den durch die Hemicube-Pixel definierten Raumwinkel, die geringste euklidische Distanz relativ zum Projektionszentrum des Hemicubes haben. Eine parallele Bestimmung von Formfaktoren bei verteilter Speicherung der Szene kann durchgeführt werden, indem die in den Hemicube-Pixeln gespeicherten Distanzwerte zwischen den Prozessoren ausgetauscht werden. In diesem Fall kann ein Prozessor seine lokale Teilszene auf den Hemicube projizieren und ggf. die in den Hemicube-Pixeln gespeicherten Distanzwerte aktualisieren, wenn ein lokales Patch eine geringere Distanz zum Projektionszentrum hat. Die parallele Verteilung der Energie eines Shooting-Patches wird dann in zwei Phasen ausgeführt. In der ersten Phase werden alle Teilszenen auf den Hemicube des Patches projiziert, um die globalen Sichtbarkeitsverhältnisse in der Szene zu identifizieren. In einer zweiten Phase der parallelen Berechnungen wird die berechnete Hemicube-Belegung allen Prozessoren bekannt gemacht. Die Prozessoren können dann unabhängig voneinander Formfaktoren zwischen dem entsprechenden Shooting-Patch und den Patches der lokalen Szene unter Berücksichtigung der zuvor berechneten Sichtbarkeitsinformationen bestimmen. Auf Basis dieser Formfaktoren kann dann die Energieverteilung innerhalb der Teilszenen durchgeführt werden.

Diese Vorgehensweise hat jedoch erhebliche Nachteile. Es wird eine Synchronisation bei den Berechnungen erzwungen, da eine Verteilung der Energie eines Shooting-Patches erst dann möglich ist, wenn die Hemicube-Projektionen vollständig ausgeführt wurden (d.h. alle Teilszenen wurden projiziert). Der größere Nachteil besteht jedoch in dem Kommunikations-Overhead, der erzeugt wird. Da für jedes Hemicube-Pixel ein Distanzwert gespeichert und zwischen den Prozessoren ausgetauscht wird, hat eine Nachricht schon bei einer moderaten Hemicube-Auflösung von 256×256 Pixel für die obere Hemicube-Seite und jeweils 256×128 Pixel für die sechs Seitenflächen des Hemicubes eine Größe von 8 MByte (32 Byte pro Pixel).

Mit Hilfe des in Abschnitt 5.2.3.2 eingeführten Konzeptes des Propagationsgraphen läßt sich eine datenparallele Strategie zur Berechnung von Formfaktoren realisieren, die eine vollkommen asynchrone Verteilung der Energie eines Patches in den Teilszenen ermöglicht. Hierbei müssen nur Nachrichten geringer Größe zwischen den Prozessoren ausgetauscht werden. Die Grundidee dieser Strategie ist, daß Formfaktoren zwischen einem Shooting-Patch s und den Patches einer Teilszene in Zelle $C_i \in \text{PART}$ berechnet werden können, bevor alle von s sichtbaren Teilszenen auf den Hemicube projiziert wurden. Dieses ist möglich, wenn garantiert ist, daß alle Patches mit einer geringeren Distanz zu dem jeweiligen Shooting-Patch schon auf den Hemicube projiziert wurden, bevor ein externes Shooting mit Patch s innerhalb von Zelle C_i durchgeführt wird. Ist dieses gewährleistet, besteht nicht die Notwendigkeit die Distanzwerte zwischen den Prozessoren auszutauschen, sondern lediglich die boolesche Information, ob auf ein bestimmtes Hemicube-Pixel schon eine Projektion durchgeführt wurde (Pixelwert = TRUE) oder ob in dem entsprechenden Raumwinkel kein Patch einer schon projizierten Teilszene die Sichtbarkeit von s einschränkt (Pixelwert = FALSE). Bei einer externen Shooting-Iteration sind nur Projektionen auf Hemicube-Pixel zugelassen, deren Eintrag in der empfangenen Bitmap den Wert FALSE hat. Es reicht somit aus, eine Bitmap-Kodierung der Hemicube-Flächen zwischen den Prozessoren auszutauschen und diese im Verlauf der Berechnungen zu aktualisieren, sobald ein Patch auf ein bisher unbelegtes Hemicube-Pixel projiziert wird (siehe Abb. 5.16). Bei dieser Vorgehensweise reduziert sich die Nachrichtengröße von 8 MByte auf 32 KByte, wenn nur ein Bit pro Hemicube-Pixel übertragen werden muß.

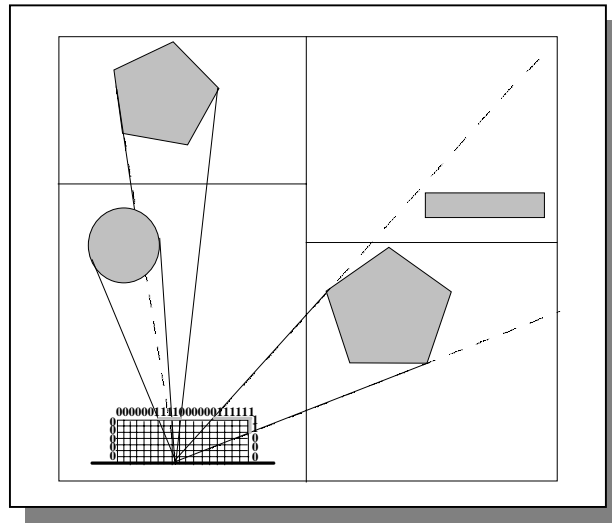


Abb. 5.16: Kodierung einer Belegung der Hemicube-Pixel durch boolesche Werte.

Wenn keine Distanzwerte zwischen den Prozessoren ausgetauscht und eine Energietransfer unmittelbar nach Durchführung der lokalen Projektionen vorgenommen werden soll, ist die Reihenfolge der Projektionen der Teilszenen entscheidend für die Korrektheit der Lösung. Ein möglicher Fehler während der Formfaktorbestimmung bei verteilter Speicherung der Szene ist in Abb. 5.17 beschrieben.

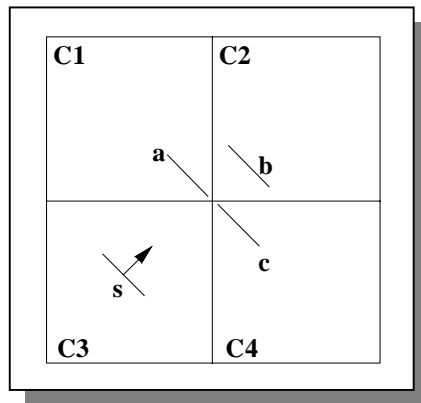


Abb. 5.17: Beispiel für Sichtbarkeitsfehler bei paralleler Berechnung von Formfaktoren.

Wenn die relative Sichtbarkeit zwischen Patch s und den Patches a , b und c in verschiedenen Teilszenen mittels Projektionen auf einen über s positionierten Hemicube ermittelt werden soll, darf die Projektion von Patch b auf den Hemicube erst dann erfolgen, wenn schon alle Patches innerhalb der Teilszenen C_3 , C_4 und C_1 durchgeführt wurden. In diesem Beispiel verdecken die Patches a und c jeweils teilweise die relative Sichtbarkeit zwischen s und b . Wenn bei der Projektion von b auf den Hemicube nicht die Projektionen von a und c berücksichtigt werden, wird der Formfaktor F_{sb} einen zu großen Wert bekommen, da die relative Sichtbarkeit falsch approximiert wurde. Somit bekommt b bei der anschließenden Verteilung der Energie des Patches s auf Basis der berechneten Formfaktoren einen zu großen Energiewert zugewiesen, obwohl der größte Teil der spezifischen Ausstrahlung von s durch die Patches a und c abgeblockt wird, bevor die Energie auf das Patch b auftreffen kann. Die berechnete Ra-

diosty-Lösung wäre in diesem Fall falsch. Mögliche Reihenfolgen für die Projektionen der Teilszenen auf den Hemicube von Patch s , die zu einer korrekten Bestimmung der Sichtbarkeitsverhältnisse führen, sind: C_3, C_1, C_4, C_2 oder C_3, C_4, C_1, C_2 .

Die Teilszenen der Zellen C_1 und C_4 können keine gegenseitigen Verdeckungen bzgl. dem Patch s verursachen, da sowohl $C_1 <_s C_4$ als auch $C_4 <_s C_1$ gilt. C_1 und C_4 sind nicht direkt benachbart, da sie an einen gemeinsamen Punkt angrenzen und Korollar 5.1 aus diesem Grund für C_1 und C_4 nicht gilt. In diesem Fall ist es nicht möglich, daß Patches aus C_1 und C_4 auf gleiche Hemicube-Pixel projiziert werden. Nachdem die Teilszene innerhalb von Zelle C_3 auf den Hemicube projiziert wurde, ist es somit möglich, Projektionen auf den Hemicube innerhalb dieser Zellen parallel und unabhängig voneinander auszuführen, ohne daß es hierbei zu falschen Ergebnissen kommt. Bevor jedoch die Projektionen innerhalb von C_2 ausgeführt werden können, müssen die in C_1, C_4 berechneten Pixelbelegungen der Hemicube-Seiten kombiniert werden, um alle möglichen Verdeckungen durch Objekte benachbarter Teilszenen (Patch a und Patch c) bei den Formfaktorberechnungen zwischen Patch s und Patch b berücksichtigen zu können.

Eine Projektionsreihenfolge der Teilszenen auf den Hemicube oberhalb von Patch s , welche eine fehlerfreie Berechnung von Formfaktoren zuläßt, ist durch den Propagationsgraph von s gegeben. Hierbei darf eine Teilszene innerhalb einer Zelle $C_i \in \text{PART}$, die von Patch s sichtbar ist, erst auf den Hemicube projiziert werden, wenn alle Teilszenen auf den Hemicube projiziert wurden, die innerhalb von Zellen auf einem beliebigem Pfad vom Ursprungsknoten C_s des Propagationsgraphen zum Knoten C_i liegen. Aufgrund der konvexen Zellform von C_s können die Formfaktoren während der lokalen Shooting-Iteration innerhalb der Teilszene ohne Kommunikation durchgeführt werden (siehe Abschnitt 5.2.1). Während der Hemicube-Projektionen wird die Pixel-Belegung der Hemicube-Flächen durch boolesche Werte in einer Bitmap kodiert. Im Anschluß an die lokalen Berechnungen wird eine Kopie der Daten zur Beschreibung des Shooting-Patches s (geometrische und energetische Informationen) und die kodierte Hemicube-Belegung an alle Prozessoren verschickt, welche die Berechnungen für Teilszenen innerhalb von Zellen durchführen, die im Propagationsgraph von s direkte Nachfolger von C_s sind. Ein Empfänger-Prozessor verwendet die in der Nachricht enthaltenen kodierten Informationen der Hemicube-Belegung, um eine externe Shooting-Iteration durchzuführen. Hierbei werden Projektionen von lokalen Patches nur auf Hemicube-Pixel zugelassen, deren Eintrag in der Bitkodierung einen Wert FALSE hat. Wenn ein Pixel durch eine lokale Projektion belegt wird, verändert sich der entsprechende Wert in der Bitkodierung von FALSE auf TRUE. Die aktualisierten Informationen werden gemäß dem Propagationsgraph an Nachfolgerknoten geschickt (bzw. an die für die Berechnung zuständigen Prozessoren).

Wenn der Knoten C_i im Propagationsgraph einen Eingangsgrad $m > 1$ hat, müssen m Hemicube-Belegungen kombiniert werden, bevor eine externe Shooting-Iteration mit s als Shooting-Patch durchgeführt werden kann. Dieses ist notwendig, um eine vollständige Hemicube-Belegung zu erhalten, die alle Projektionen von Teilszenen auf den Hemicube berücksichtigt, welche Auswirkungen auf die relative Sichtbarkeit zwischen dem Shooting-Patch s und den Patches der lokalen Teilszene innerhalb von C_i haben können.

Als Verknüpfungsfunktion der in Bitmaps kodierten Belegungen der Hemicube-Flächen dient das logische ODER. Die als boolesche Werte kodierte Information über die Belegung eines Hemicube-Pixels in den m empfangenen Bitmaps werden mittels logischem ODER verknüpft. Somit wird bei dieser Verknüpfung ein Hemicube-Pixel als belegt (TRUE) markiert, sobald mindestens in einer der empfangenen Bitmaps das entsprechende Pixel als belegt markiert ist.

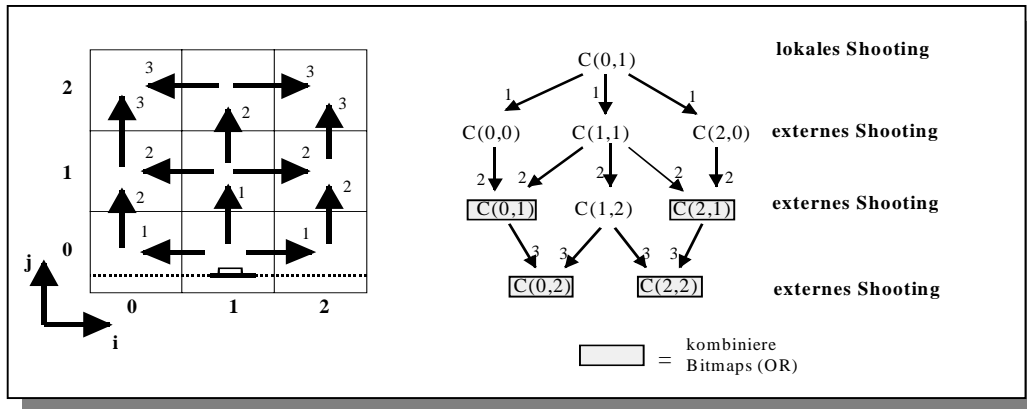


Abb. 5.18: Routing der Hemicube-Bitmaps auf Basis des Propagationsgraphen des Shooting-Patches.

Ein Beispiel für die Reihenfolge der Projektionen von Teilszenen auf den Hemicube eines ausgewählten Shooting-Patches ist in Abb. 5.18 gegeben. Im rechten Teil des Bildes ist der Propagationsgraph des Patches angegeben, wobei alle Knoten gemäß ihrer Distanz zum Ursprungsknoten hierarchisch angeordnet wurden (zunehmende Distanz von oben nach unten). In allen Knoten auf einer Hierarchiestufe können die Hemicube-Projektionen parallel ausgeführt werden, wobei die Informationen über die Hemicube-Belegungen von ihren jeweiligen Vorgängern im Propagationsgraph benötigt und ggf. kombiniert werden. Hierbei ist jedoch zu beachten, daß die entsprechenden externen Shooting-Iterationen in den Zellen auf einer Hierarchiestufe aufgrund der asynchronen Berechnungen nicht wirklich zur gleichen Zeit erfolgen.

5.2.4 Ablauf der datenparallelen Radiosity-Berechnungen

Aufgrund der Komplexität der betrachteten Einzelaspekte des parallelen Verfahrens soll an dieser Stelle noch einmal eine kurze Zusammenfassung der Methode gegeben werden, welche den Ablauf der datenparallelen Berechnungen beschreibt.

Wenn ein Prozessor ein lokales Shooting-Patch auswählt und dessen Energie in der lokalen Szene verteilt, müssen die anderen Prozessoren diese Iteration ebenfalls durchführen, damit das Energiegleichgewicht, welches durch das Radiosity-Gleichungssystem beschrieben wird, erhalten bleibt. Da die Parallelisierung auf einer Objektraumaufteilung basiert, ist es notwendig, Informationen über Pixel-Belegungen eines Hemicubes zwischen den Berechnungsprozessen auszutauschen, um relative Sichtbarkeitsbeziehungen zwischen den Patches unterschiedlicher Teilszenen berücksichtigen zu können. Zu diesem Zweck werden die Belegungen der Hemicube-Pixel in einer Nachricht binär codiert (Hemicube-Bitmap) und an Prozessoren übermittelt, die Teilszenen in Abstrahlrichtung des Shooting-Patches bearbeiten. Prozessoren, die eine Nachricht mit einer Hemicube-Bitmap empfangen, können diese dazu verwenden, die Energie des entsprechenden Shooting-Patches im Rahmen einer externen Shooting-Iteration in der lokalen Teilszene zu verteilen.

Die in angrenzenden Teilszenen vorherrschenden Sichtbarkeitsverhältnisse können aufgrund der in der Hemicube-Bitmap kodierten Belegungsinformationen der Hemicube-Pixel berücksichtigt werden. Projektionen werden dabei nur noch auf Hemicube-Pixel gestattet, die nicht schon durch Projektionen von Patches aus Teilszenen belegt sind, welche die Sichtbarkeit relativ zu dem externen Shooting-Patch beeinträchtigen können (und somit die entsprechende Bitmap-Inhalt den Wert Eins hat – siehe Abb. 5.19).

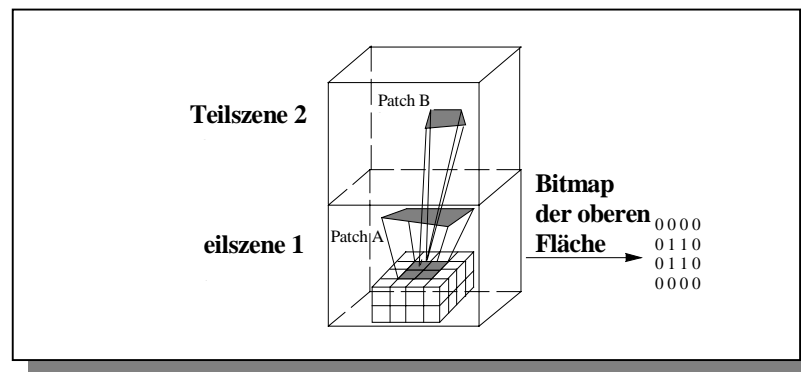


Abb. 5.19: Patch *B* der Teilszene 2 wird unter Berücksichtigung der Bitmap auf den Hemicube projiziert.

Die Verwendung der Hemicube-Bitmap ist nur dann möglich, wenn ein spezielles Routing-Schema für diese Bitmaps verwendet wird. Dieses garantiert, daß Teilszenen immer mit einer zunehmenden Distanz zum jeweiligen Shooting-Patch auf den entsprechenden Hemicube projiziert werden. Wenn zusätzlich garantiert ist, daß die Teilszenen innerhalb von konvexen Begrenzungsvolumen liegen, besteht nicht die Notwendigkeit, die in den Hemicube-Pixeln gespeicherten Distanzwerte zwischen Prozessoren zu kommunizieren. Es reicht lediglich die Information aus, daß das entsprechende Hemicube-Pixel durch die Projektion eines Patches belegt ist, welches zu einer Teilszene gehört, die eine geringere Distanz zu dem entsprechenden Shooting-Patch besitzt. Ein solches Routing-Schema ist durch den Propagationsgraph eines Shooting-Patches gegeben. Dieser definiert die Reihenfolge der Energieausbreitung der bisher nicht verschossenen spezifischen Ausstrahlung eines Patches innerhalb der sichtbaren Teilszenen des jeweiligen Shooting-Patches. Es ist zu beachten, daß der Propagationsgraph während der Berechnungen nicht vollständig erzeugt und zwischen den Prozessoren ausgetauscht wird. Jeder Prozessor kann die relevanten Informationen des Propagationsgraph (e eingehende und ausgehende Kanten der lokalen Zelle) aufgrund der geometrischen Position eines lokalen oder externen Shooting-Patches und den geometrischen Informationen der lokalen Zelle (Ebenengleichungen der Begrenzungspolygone) sowie den Nachbarschaftsbeziehungen aus dem Kommunikations-Graphen sehr schnell und einfach generieren.

Die asynchronen Berechnungen werden in den Teilszenen solange fortgesetzt, bis die verbleibende noch zu verteilende Restenergie einen vorgegebenen Grenzwert unterschreitet. Hierbei ist zu berücksichtigen, daß einer Teilszene nicht nur aufgrund von lokalen primären Lichtquellen Energie zugeführt wird, sondern daß jede empfangene Nachricht implizit eine gewisse Energiemenge transportiert, die der entsprechenden Teilszene von außen zugeführt wird. Jeder Prozessor hat eine Nachrichtenwarteschlange, in der die ankommenden Nachrichten gespeichert werden. Aus dieser Schlange wird immer die Nachricht entnommen, welche das Patch mit der größten nicht verschossenen spezifischen Ausstrahlung enthält und dessen Sichtbarkeitsinformationen gemäß dem Propagationsgraph des Patches vollständig sind (Kombination von Bitmaps – siehe 5.2.3.4). Wie bei dem sequentiellen Progressive-Refinement-Verfahren wird immer das Shooting-Patch mit maximaler nicht verschossener spezifischer Ausstrahlung ausgewählt. Ein Prozessor kann somit entweder das lokale Patch mit der größten nicht verteilten Energiemenge oder das Patch aus einer Nachricht mit maximaler nicht verteilter Energiemenge auswählen. Von beiden Möglichkeiten wird das Patch mit der größeren Energiemenge ausgewählt. Die nicht verteilte spezifische Ausstrahlung dieses Shooting-Patches wird in der lokalen Teilszene verteilt, indem eine Iteration des sequenti-

ellen Progressive-Refinement-Algorithmus, unter Berücksichtigung der in den Bitmaps kodierten Sichtbarkeitsinformationen ausgeführt wird. Bei diesen Berechnungen werden die Bitmaps entsprechend der durchgeführten Projektionen aktualisiert und im Anschluß an die lokalen Berechnungen gemäß des Routing-Schemas weitergeleitet.

Zu einem beliebigen Zeitpunkt der parallelen Berechnungen werden aufgrund der innerhalb der Teilszenen asynchron ausgeführten lokalen Iterationen verschiedene Shooting-Patches gleichzeitig durch die Prozessoren des Parallelrechners bearbeitet. Aus diesem Grund verändert sich das Konvergenzverhalten des parallelen Verfahrens für unterschiedliche Anzahlen von Prozessoren. Dieses ist damit zu begründen, daß bei den sequentiellen Berechnungen in jeder Iteration ein Patch ausgewählt wird, dessen nicht verschossene spezifische Ausstrahlung maximal innerhalb der gesamten Szene ist. Wenn jedoch bei den parallelen Berechnungen die Shooting-Patches asynchron innerhalb der Teilszenen bestimmt werden, bilden die entsprechenden nicht verschossenen spezifischen Ausstrahlungen jeweils nur lokale Maxima. Dieses führt im allgemeinen dazu, daß bei den parallelen Berechnungen mehr Iterationen (mit jeweils geringeren Energiemengen) ausgeführt werden als bei der vergleichbaren sequentiellen Berechnung, die bis zum gleichen Konvergenzstadium der globalen Radiosity-Lösung durchgeführt wird.

Bei den parallelen Berechnungen muß unterschieden werden zwischen einer lokalen Konvergenz innerhalb von einzelnen Teilszenen und einer globalen Konvergenz der berechneten Lösung. Eine lokale Konvergenz liegt vor, wenn in einer Teilszene die maximale nicht verschossene spezifische Ausstrahlung eines lokal ausgewählten Shooting-Patches einen vorgegebenen Grenzwert unterschreitet. In diesem Fall wird keine weitere lokale Shooting-Iteration ausgeführt. Der Grenzwert, welcher die lokale Konvergenz festlegt, orientiert sich an der maximalen Energiemenge, die initial einem in der Szene enthaltenen primären Lichtquellen-Patch zugewiesen wurde. Die parallelen Berechnungen dürfen jedoch erst dann terminieren, wenn für alle Teilszenen eine lokale Konvergenz erkannt und im gesamten System keine weitere Nachricht verschickt wurde, welche einer anderen Teilszene Energie zuführen kann. Diese globale Konvergenz der parallelen Berechnungen kann mit Hilfe eines Verfahrens zur Terminierungs-Erkennung von parallelen Berechnungen zuverlässig erkannt werden [Diso80].

5.2.5 Performanz der datenparallelen Radiosity-Berechnungen

Zur Untersuchung der Skalierbarkeitseigenschaften des vorgestellten datenparallelen Verfahrens wurden Testmessungen in Szenen verschiedener Komplexität mit variierender Anzahl von Prozessoren durchgeführt. Die Komplexität der Szenen bezieht sich hierbei auf die Anzahl der während der initialen Netzgenerierung erzeugten Patches und auf die Anzahl der primären Lichtquellen in einer Szene.

Die Messungen wurden auf dem Parallelrechner *CC-48* der Firma *Parsytec* durchgeführt. Jeder Knoten des Systems ist ein *PowerPC 604* mit 133 MHz Taktfrequenz und 64 MByte Hauptspeicher, von denen bei Nutzung eines Nano-Kernels als Ausführungsumgebung 59 MByte von der Applikation verwendet werden können. Die Peak-Performance des Systems beträgt 12.76 GFlops. Die Interprozeß-Kommunikation basiert auf einer speziellen Implementierung der PVM-Library für dieses System (*PowerPVM*), die auf PARIX aufgesetzt wurde.

Während der Testmessungen wurde eine Hemicube-Auflösung d 256x256 Hemicube-Pixeln für die obere Seite verwendet. Der Grenzwert für die Erkennung der lokalen Konvergenz ist bei den Testmessungen derart gewählt worden, daß einerseits die Berechnungszeit für eine sequentielle Referenz-Lösung nicht zu groß wurde, andererseits die zu verteilende Restenergie in den Szenen keine großen Auswirkungen auf die globale Beleuchtung der Szene mehr hatte (d.h. visuell nicht mehr wahrnehmbar ist). Während der Radiosity-Berechnungen ist eine adaptive Unterteilung der Patches gemäß dem in 5.2.2.2 angegebenen Substrukturierungskriterium vorgenommen worden. Hierbei ist zu beachten, daß die Substrukturierungsparameter so gewählt wurden, daß der zur Verfügung stehende lokale Speicher der Knoten des *CC-48* ausreicht, um auch eine sequentielle Referenz-Messung durchführen zu können.

Ein Maß für die Skalierbarkeitseigenschaften eines parallelen Verfahrens ist der erzielte *Speed-Up*. Dieser Wert setzt die Zeit, die benötigt wird ein Problem auf einem Prozessor eines Parallelrechners zu lösen in Beziehung zu der Zeit, die benötigt wird, das gleiche Problem auf mehreren Prozessoren des Systems zu lösen. Der Speed-Up ist definiert als

$$S(n) := \frac{T(1)}{T(n)}, \quad (5.9)$$

wobei $T(1)$ die sequentielle Berechnungszeit und $T(n)$ die Berechnungszeit auf n Prozessoren ist. Aus dem Speed-Up läßt sich die *Effizienz* der parallelen Berechnungen als ein weiteres Maß für die Skalierbarkeit eines parallelen Verfahrens ableiten. Die Effizienz gibt an, wie gut die zur Verfügung stehende Rechenkapazität der an den Berechnungen beteiligten Prozessoren ausgenutzt wird. Die Effizienz der Berechnungen auf n Prozessoren ist definiert als

$$E(n) := \frac{S(n) \times 100}{n}. \quad (5.10)$$

Ein paralleles Verfahren besitzt ideale Skalierbarkeitseigenschaften, wenn der Speed-Up linear mit der Anzahl der während der Berechnungen verwendeten Prozessoren wächst (d.h. die Berechnungszeit auf n Prozessoren ist n mal kürzer als die entsprechende sequentielle Berechnungszeit) und die Effizienz somit für jede Anzahl von Prozessoren 100% beträgt. Dieses wird jedoch von den wenigsten parallelen Verfahren erreicht, da die Interprozeß-Kommunikation einen Mehraufwand (Kommunikations-Overhead) verursacht und die Last während der Berechnungen im allgemeinen nicht derart gleichmäßig verteilt ist, daß Prozessoren zu jedem Zeitpunkt vollständig ausgelastet sind.

Verfahren	Speicherung der Szene	Effizienz	Anzahl der Prozessoren	Anzahl der Patches
[Chen89]	Mehrfachspeicherung	Fast linear	8	6.000
[BW90]	Shared-Memory	50%	8	38.000
[PSV90]	Mehrfachspeicherung	68%	6	1.000
[CP91]	zirkulierend	31%	42	448
[FP91]	zirkulierend	60%	9	4.500
[CAO93]	zirkulierend	86%	16	8.000
[Š96]	Mehrfachspeicherung	90%	8	9.800

Tabelle 5.4: Vergleich von parallelen Progressive-Refinement-Verfahren auf Basis der Hemicube-Methode.

Die wenigsten der bisher vorgenommenen Parallelisierungen der Progressive-Refinement-Methoden auf Basis von Hemicube-Projektionen zur Sichtbarkeitsbestimmung konnten eine gute Effizienz auf mehr als 8 Prozessoren erreichen (siehe Tabelle 5.4 und [Gib98]). Keines der angegebenen Verfahren erlaubt die adaptive Substrukturierung von Patches während der parallelen Berechnungen.

Es ist zu beachten, daß in Tabelle 5.4 lediglich Parallelisierungen aufgeführt wurden, die mit dem vorgestellten Verfahren vergleichbar sind, da sie gleiche Techniken zur Bestimmung der relativen Sichtbarkeit der Formfaktorberechnung verwenden und somit ähnliche Problemstellungen bei der Parallelisierung betrachtet werden müssen.

Im weiteren Verlauf dieses Abschnittes werden zunächst die Testszenen beschrieben, sowie die jeweiligen sequentiellen und parallelen Berechnungszeiten einer Radiosity-Lösung angegeben. Im Anschluß daran werden die erzielten Ergebnisse im Zusammenhang interpretiert.

Als erste Testszene wurde eine sehr einfache Szene gewählt, die aus zwei aneinandergrenzenden Räumen besteht, die teilweise möbliert sind. Jeder Raum enthält zwei Lichtquellen und die Räume sind durch eine offene Tür verbunden. Die Szene wurde initial in 6281 Patches unterteilt. Bei der Formfaktorberechnung wurden 9 Sample-Punkte auf den Empfänger-Patches plaziert. Die Konvergenzschranke wurde mit $\epsilon = 0.001$ vorgegeben, d.h. die Berechnungen terminieren, sobald die spezifische Ausstrahlung der lokal ausgewählten Shooting-Patches weniger als 0.1 % der spezifischen Ausstrahlung der in der Szene enthaltenen hellsten Lichtquelle ist. Die Berechnungszeiten für verschiedene Anzahlen von Prozessoren sowie der daraus resultierende Speed-Up und die Effizienz der parallelen Berechnungen sind in Tabelle 5.5 angegeben.

Prozessoren	Berechnungszeit (in Sekunden)	Speed-Up	Effizienz (%)
1	1969	1.00	100
2	1110	1.77	89
4	672	2.93	73
6	513	3.83	64
8	418	4.71	59
12	284	6.93	58
18	262	7.51	42
20	250	7.87	39

Tabelle 5.5: Berechnungszeiten für eine einfache Testszene.

Eine etwas komplexere Szene repräsentiert den Eingangsbereich der Universität-GH Paderborn vor den Hörsälen C1 und C2. Die Szene wurde initial in 22234 Patches unterteilt, wobei in der Szene 97 primäre Lichtpatches enthalten sind. Bei der Formfaktorberechnung wurden 9 Sample-Punkte auf den Empfänger-Patches plaziert. Die Konvergenzschranke wurde mit $\epsilon = 0.0003$ vorgegeben. Die entsprechenden Ergebnisse sind in Tabelle 5.6 aufgeführt. Während der parallelen Berechnungen wurde eine adaptive Verfeinerung des initialen Radiosity-Netzes durchgeführt. Das Lösungsnetz ist somit erheblich komplexer und enthält am Ende der ca. 66.000 Patches.

Prozessoren	Berechnungszeit (in Sekunden)	Speed-Up	Effizienz (%)
1	7370	1.00	100
2	3739	1.90	97
4	1961	3.76	94
6	1476	4.99	83
8	1171	6.29	79
12	821	8.97	75
18	651	11.32	63
24	511	14.41	60
27	487	15.12	56
36	387	19.04	53
40	353	20.86	52

Tabelle 5.6: Berechnungszeiten für die Testszene *Universitäts-Foyer*.

Eine deutlich komplexere Testszene repräsentiert einen *Hörsaal* mit angrenzendem Seminarraum. Das Modell wurde von der FhG/IGD-Darmstadt im Rahmen des BMBF-Projektes PARAGRAPH modelliert und als Testszene zur Verfügung gestellt. Das initiale Netz, welches als Grundlage der Radiosity-Simulation dient, wurde in 41111 Patches unterteilt. In der Szene sind initial 212 primäre Lichtpatches enthalten. Bei der Formfaktorberechnung wurden 9 Sample-Punkte auf den Empfänger-Patches plaziert. Die Konvergenzschranke wurde mit $\epsilon = 0.001$ vorgegeben. In Tabelle 5.7 sind die erzielten Ergebnisse angegeben. Nach Beendigung der Berechnungen wurde die Ausgangsszene aufgrund von adaptiver Verfeinerung in 75.000 Patches unterteilt.

Prozessoren	Berechnungszeit (in Sekunden)	Speed-Up	Effizienz (%)
1	2068	1.00	100
4	603	3.43	86
8	325	6.36	80
12	213	9.71	81
16	173	11.95	75
20	151	13.70	68
24	118	17.53	73
28	109	18.97	68
32	103	20.08	63
36	105	19.70	55
40	94	22.00	55

Tabelle 5.7: Berechnungszeiten für die Testszene *Hörsaal*.

Bei der Szene *Townhouse* handelt es sich um ein Modell eines kleinen Gebäudes. In diesem Gebäude sind verschiedene mit Möbeln ausgestattete Räume auf drei Stockwerken angeordnet. Jeder dieser Räume ist durch Lichtquellen beleuchtet, wobei insgesamt 250 primäre Lichtquellenpatches in der Szene enthalten sind. Während der initialen Netzgenerierung wurden 45747 Patches erzeugt. Bei der Formfaktorberechnung wurden 9 Sample-Punkte auf den Empfänger-Patches plaziert. Die Konvergenzschranke wurde mit $\epsilon = 0.001$ vorgegeben. Die Berechnungszeiten und der daraus resultierende Speed-Up sowie Effizienz sind in Tabelle 5.8 angegeben. Durch die adaptive Verfeinerung der initialen Patches wurden insgesamt 80.000 Patches während der Berechnung erzeugt.

Prozessoren	Berechnungszeit (in Sekunden)	Speed-Up	Effizienz (%)
1	4117	1.00	100
4	1185	3.47	87
8	640	6.43	80
12	429	9.60	80
16	362	11.37	71
20	335	12.29	61
24	222	18.55	77
28	206	19.99	71
32	189	21.78	68
36	183	22.50	62
40	170	24.22	61

Tabelle 5.8: Berechnungszeiten für die Testszene *Townhouse*.

Das komplexeste Modell, mit welchem Testmessungen durchgeführt wurden, repräsentiert den *Rosenthaler-Hof*. Hierbei handelt es sich um einen Teil der Planung für den Umbau des ehemaligen Kaufhauses Wertheim im Zentrum von Berlin. Die Szene stellt einen überdachten Innenhof mit Bistro, Brunnen und umgebende Gebäudefassaden dar. Das Modell wurde von der Berliner Firma *System Connect* modelliert und diente in dem BMBF-Projekt PARAGRAPH als repräsentative Demonstrationsszene. Das Modell wurde initial in 154588 Patches unterteilt. Bei der Formfaktorberechnung wurden 4 Sample-Punkte auf den Empfänger-Patches plaziert. Die Konvergenzschranke wurde mit $\epsilon = 0.0005$ vorgegeben. Bei diesen Berechnungen wurde keine Substrukturierung durchgeführt, da eine sequenzielle Referenzmessung aufgrund von Speichermangel auf dem ausführenden Prozessor nicht möglich war. Die entsprechenden Ergebnisse der Testmessungen wurden in Tabelle 5.9 zusammengefaßt.

Prozessoren	Berechnungszeit (in Sekunden)	Speed-Up	Effizienz (%)
1	8236	1.00	100
2	4417	1.86	93
4	2368	3.48	87
8	1209	6.81	85
16	658	12.51	78
24	470	17.52	73
32	422	19.50	61
36	405	20.32	56

Tabelle 5.9: Berechnungszeiten für die Szene *Rosenthaler-Hof*.

In Abb. 5.20 und Abb. 5.21 sind die Speed-Up bzw. Effizienz der parallelen Berechnungen für die verschiedenen Testszenen zusammengefaßt und gegenübergestellt, um einen direkten Vergleich zu erleichtern.

Den Abbildungen ist zu entnehmen, daß die Effizienz der parallelen Berechnungen für größere Anzahlen von Prozessoren deutlich zunimmt, wenn die Szenenkomplexität erhöht wird. Bei der sehr einfachen Testszene fällt die Effizienz unter 50% wenn mehr als 12 Prozessoren verwendet werden, während bei Szenen mit moderater und großer Komplexität sehr gute Skalierbarkeitseigenschaften auf bis zu 16 Prozessoren zu beobachten sind. Die Effizienz liegt

hierbei in allen Fällen über 70%. Für die komplexeren Szenen wie *Hörsaal*, *Townhouse* und *Rosenthaler Hof* liegt selbst bei Verwendung von 40 Prozessoren die Effizienz noch oberhalb von 50%.

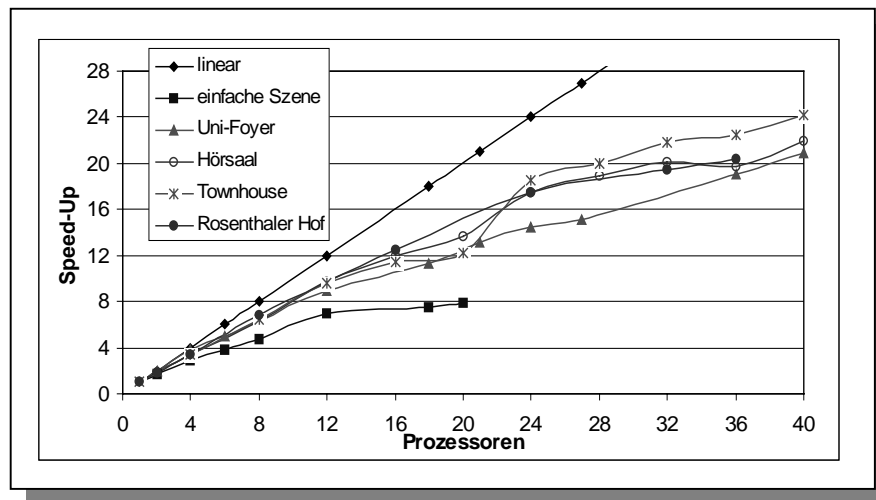


Abb. 5.20: Speed-Up der parallelen Berechnungen für verschiedene Testszenen.

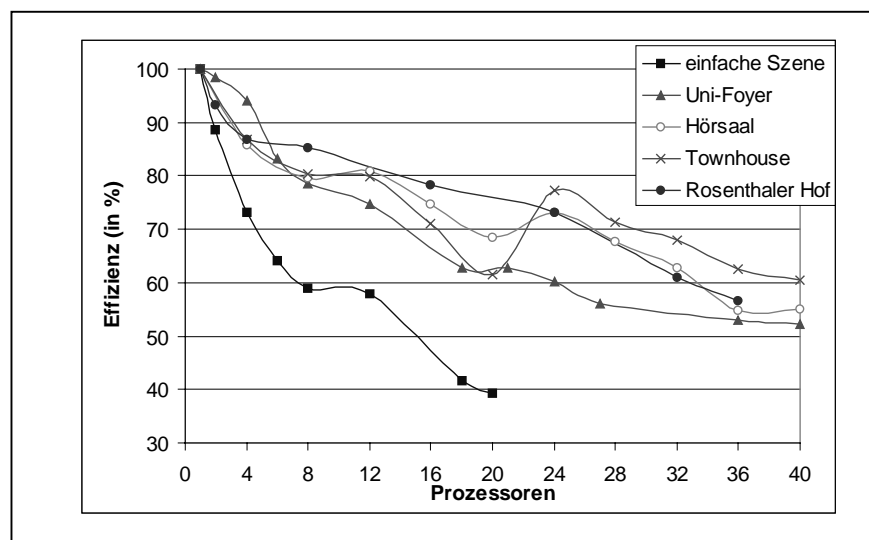


Abb. 5.21: Effizienz der parallelen Berechnungen für verschiedene Testszenen.

In Abb. 5.21 sind erhebliche Schwankungen der Effizienz bei den parallelen Berechnungen auf mehr als 16 Prozessoren zu erkennen. Dieses betrifft vor allen Dingen die Berechnungen mit den Szenen *Hörsaal* und *Townhouse*. In diesen Fällen ist ein deutlicher Einbruch der Effizienz von 75% auf 68% bei der *Hörsaal*-Szene und von 71% auf 61% bei der *Townhouse*-Szene zu beobachten. Bei der Verwendung von 24 Prozessoren steigt die Effizienz in beiden Fällen wieder über 70%. Die Ursache für dieses Verhalten ist in der statischen Verteilung der Last zu suchen. Aufgrund von lokalen geometrischen Eigenschaften der Szenen kann nicht für jede Anzahl von Prozessoren eine gute Aufteilung der Szene gefunden werden, die den Prozessoren Teilszenen mit annähernd gleicher Komplexität (Anzahl von Patches) zuordnet. Dieses trifft vor allen Dingen dann zu, wenn die Szenen sehr kleine Objekte enthalten, die aus sehr vielen Patches bestehen, wie es bei der *Hörsaal*-Szene und der *Townhouse*-Szene der Fall ist. Die Testszenen *Universitäts-Foyer* und *Rosenthaler-Hof* enthalten keine Objekte mit gro-

ßer lokaler Komplexität. Weiterhin ermöglicht die Anordnung der Objekte innerhalb der Szene für fast alle Anzahlen von Prozessoren eine Aufteilung der Szene in Teilszenen mit annähernd der gleichen Anzahl von Patches. An dem Verlauf der Effizienz-Kurven für die entsprechenden Meßreihen ist zu sehen, daß keine derartigen Schwankungen auftreten, wie es bei den Szenen *Townhouse* und *Hörsaal* der Fall ist.

Die durchschnittliche Effizienz der parallelen Berechnungen innerhalb der *Townhouse*-Szene ist, verglichen mit der *Hörsaal*-Szene, größer. Insbesondere bei der Verwendung von mehr als 32 Prozessoren fällt die Effizienz der parallelen Berechnungen auf Basis der *Hörsaal*-Szene im Vergleich zu Berechnungen für die *Townhouse*-Szene stark ab (ca. 7%), obwohl die Komplexität der Szenen ähnlich ist. Grund hierfür ist, daß die initiale Energiemenge, die in der *Townhouse*-Szene enthalten ist, aufgrund der Positionen der Lichtquellen besser auf die Teilszenen verteilt werden kann, als es bei der *Hörsaal*-Szene der Fall ist. Somit werden im Verlauf der Berechnungen für die Szene *Townhouse* gerade bei größeren Anzahlen von Prozessoren im Mittel pro Iteration größere Energiemengen in den lokalen Szenen verteilt. Dieses führt zu einer schnelleren Konvergenz der parallelen Berechnungen.

Während der Testmessungen konnten verschiedene Aspekte identifiziert werden, die teilweise erhebliche Auswirkungen auf die Effizienz der parallelen Berechnungen haben. Den größten Einfluß hat hierbei die statische Lastverteilung. Eine ungünstige statische Lastverteilung ist der Hauptfaktor für die dynamische Entstehung eines Lastungleichgewichtes während der parallelen Berechnungen. Wie schon zuvor erwähnt, ist in Abb. 5.20 und Abb. 5.21 sehr deutlich zu erkennen, daß für einige Anzahlen von Prozessoren aufgrund der Restriktionen des verwendeten Partitionierungsverfahrens keine gute Aufteilung der Eingabeszene in Teilszenen vorgenommen werden konnte. In diesen Fällen weicht schon die initiale Last der Prozessoren (Komplexität und der gebildeten Teilszenen und Verteilung der primären Lichtquellen) erheblich voneinander ab.

Neben den geometrischen Aspekten bei der statischen Lastverteilung spielt auch die Verteilung der Lichtquellen auf die Prozessoren eine sehr wichtige Rolle. Aufgrund der Verteilung der Lichtquellen innerhalb der Szene (z.B. lokale Konzentrationen) oder eine zu geringe Anzahl von Lichtquellen, können während der statischen Lastverteilung Teilszenen gebildet werden, deren initialen Energiemengen erheblich voneinander abweichen. Wenn Teilszenen keine primären Lichtquellen enthalten, können die entsprechenden lokalen Berechnungen erst beginnen, wenn diesen Teilszenen Energie in Form einer externen Shooting-Iteration zugeführt wird. Somit kann der Fall auftreten, daß Prozessoren schon zu Berechnungsbeginn Wartezeiten (*Idle-Zeiten*) haben, wodurch die Effizienz der parallelen Berechnungen beeinträchtigt wird. Dieser Aspekt läßt sich begrenzt bei der statischen Lastverteilung berücksichtigen, indem die Gewichtungsfunktion bei der statischen Lastverteilung (siehe 5.2.1) so modifiziert wird, daß Teilszenen aus weniger Patches bestehen, sobald in ihnen eine Lichtquelle enthalten ist. In diesem Fall wird die primäre Energie schneller an benachbarte Teilszenen transferiert.

Zusätzlich zu der Verteilung der primären Energie der Lichtquellen in der Szene hat auch die Verteilung der reflektierten Energie innerhalb der Szene großen Einfluß auf die Entstehung eines dynamischen Lastungleichgewichtes während der parallelen Berechnungen. Dieses wird verursacht durch abweichende Materialeigenschaften in den Teilszenen und ein dadurch bedingtes unterschiedliches Reflexionsverhalten der lokalen Objekte. Dieser Aspekt kann nur sehr grob bei der statischen Lastverteilung berücksichtigt werden. Das Verteilen der Energie von sekundären Lichtquellen (Energie die aufgrund von diffuser Reflexion innerhalb der Szene propagiert werden muß), stellt den Hauptaufwand bei den Berechnungen dar. Somit kön-

nen deutlich voneinander abweichende Reflexionseigenschaften von Objekten innerhalb der Teilszenen eine ungleichmäßige Verteilung der Energie zwischen den Teilszenen verursachen, auch wenn die primären Lichtquellen gut verteilt wurden. Dieses hat vor allen Dingen gegen Ende der parallelen Berechnungen große Auswirkungen auf die Idle-Zeiten der Prozessoren. In der Regel ist zu diesem Zeitpunkt nur noch in einigen wenigen Teilszenen Patches enthalten, deren nicht verteilte spezifische Ausstrahlung oberhalb des Grenzwertes zur Erkennung einer lokalen Konvergenz liegt.

Dieser Sachverhalt wird in Tabelle 5.10 veranschaulicht. Dort werden Zeiten, die Prozessoren während der Berechnung einer Radiosity-Lösung für die reine Problemlösung (Berechnung + Kommunikation) benötigten, den Wartezeiten (*Idle-Zeiten*) der Prozessoren gegenübergestellt. Die Zeiten wurden bei der parallelen Berechnung einer Radiosity-Lösung für die Szene *Townhouse* ermittelt (siehe Tabelle 5.8). Ein Prozessor hat hierbei den Status *idle*, sobald er eine lokale Konvergenz innerhalb der ihm zugeordneten Teilszene erkannt hat. Es ist zu beachten, daß dieser Zustand nicht notwendigerweise nur gegen Berechnungsende auftreten kann. Wenn ein Prozessor eine lokale Konvergenz erkannt hat, kann der entsprechenden Teilszene weitere Energie durch ein externes Shooting-Patch zugeführt werden, woraufhin der Prozessor das lokale Iterationsverfahren fortsetzt, bis erneut eine lokale Konvergenz erkannt wird. Die Ausführungszeit der parallelen Berechnungen setzt sich somit aus Kommunikations-, Berechnungs und Idle-Zeiten der Prozessoren sowie der Zeit zur Erkennung der globalen Konvergenz zusammen. In Tabelle 5.10 ist deutlich zu erkennen, daß die Hauptlast während der parallelen Berechnungen bei Prozessor 6 liegt, während die anderen Prozessoren für große Zeitspannen untätig sind. Der Prozessor mit der größten Idle-Zeit (Prozessor 4) ist ca. 19.7 % der Ausführungszeit der parallelen Berechnungen untätig. Diese Tatsache ist dafür verantwortlich, daß die Effizienz der parallelen Berechnungen auf acht Prozessoren für die *Townhouse*-Szene lediglich 80% beträgt (siehe Tabelle 5.8).

Proz. Nr.	Berechnung + Kommunikation (Sek.)	idle (Sek.)	Ausführungszeit (Sek.)
0	538.6	101.6	640.2
1	558.3	81.8	640.1
2	529.5	110.8	640.3
3	599.3	41.0	640.3
4	513.4	126.6	640.0
5	524.6	115.6	640.2
6	637.7	2.4	640.1
7	549.5	90.8	640.3

Tabelle 5.10: Idle-Zeiten der Prozessoren während der parallelen Radiosity-Berechnungen für die *Townhouse*-Szene auf acht Prozessoren.

Weitere Auswirkungen auf die Effizienz der parallelen Berechnungen konnten beobachtet werden, wenn sehr kleine Grenzwerte zur Erkennung der lokalen Konvergenz vorgegeben wurden. In diesem Fall werden aufgrund der asynchronen Berechnungen lokale Shooting-Patches ausgewählt, deren nicht verschossene spezifische Ausstrahlung zum Auswahlzeitpunkt zwar ein lokales, aber kein globales Maximum bilden. Somit werden viele Iterationen ausgeführt, in denen nur geringe Energiemengen verteilt werden. Die spezifische Ausstrahlung eines lokal ausgewählten Shooting-Patches muß in verschiedenen Teilszenen gemäß dem Propagationsgraph des Patches verteilt werden, um eine global korrekte Radiosity-Lösung zu berechnen. Somit werden viele Nachrichten erzeugt, die nur geringe Energiemengen zwischen den Zellen transportieren. Dieses fällt vor allen Dingen ins Gewicht, je mehr Prozessoren am

Berechnungsprozeß beteiligt sind (siehe Testmessungen für Szene *Universitäts-Foyer*). Hierdurch entsteht ein erheblicher Mehraufwand für Kommunikation und es sind teilweise deutlich mehr Shooting-Iterationen notwendig, um während der parallelen Berechnungen die gleiche Energiemengen zu verteilen (bzw. um ein gleiches Konvergenzstadium zu erreichen), wie bei den entsprechenden sequentiellen Berechnungen. Somit spielt der Kommunikations-Overhead des vorgestellten datenparallelen Verfahrens erst dann eine Rolle, wenn sehr kleine Konvergenzschranken vorgegeben werden. Es ist jedoch zu beachten, daß es sehr einfache und effektive Methoden gibt dieses Problem zu beheben. Indem während der Berechnungen die Konvergenzschranken stufenweise verringert werden, kann erzwungen werden, daß zu jedem Zeitpunkt der Berechnung lediglich Shooting-Patches ausgewählt werden können, die eine Mindestmenge an nicht verteilter Energie besitzen. Zu diesem Zweck müssen jedoch Synchronisationspunkte während der parallelen Berechnungen eingeführt werden.

Ein weiterer Aspekt, der bei den parallelen Berechnungen unbedingt berücksichtigt werden muß, ist die Tatsache, daß der Kommunikations-Overhead überwiegt, sobald der lokale Berechnungsaufwand innerhalb der Zellen zu gering ist oder zu gering wird. Dieses betrifft vor allen Dingen Berechnungen auf einer großen Anzahl von Prozessoren. Der Berechnungsaufwand innerhalb der lokalen Zellen korreliert mit der Anzahl der Patches einer Szene, der Anzahl der Sample-Punkte bei der Formfaktorberechnung und der Substrukturierungstiefe. Bei Szenen geringer Komplexität (z.B. die einfache Testszene) kann in den meisten Fällen nur mit einer moderaten Anzahl von Prozessoren (12-16) noch eine Effizienz $> 50\%$ erzielt werden. Wenn jedoch die lokale Berechnungskomplexität erhöht wird (d.h. Anzahl der verwendeten Sample-Punkte oder Substrukturierungstiefe), kann auch für Szenen mit geringer Komplexität auf größeren Anzahlen von Prozessoren eine gute Effizienz erzielt werden. Bei der Szene *Rosenthaler-Hof* ist sehr deutlich zu sehen, daß ab ca. 24 Prozessoren der lokale Berechnungsaufwand zu gering wird, um noch bessere Effizienzen zu erzielen. Dieses ist mit der geringen Anzahl von Sample-Punkten (4 Sample-Punkte) zu begründen, die bei den Berechnungen verwendet wurden. Weiterhin wurde keine adaptive Substrukturierung durchgeführt. Bei den Berechnungen auf weniger als 24 Prozessoren reicht die Komplexität der Szene (die Anzahl der Patches in den Teilszenen) aus, um eine hohe lokale Berechnungskomplexität zu erzeugen. Dieses ist sehr deutlich daran zu erkennen, daß bei den Berechnungen auf bis zu 24 Prozessoren für die *Rosenthaler-Hof*-Szene die besten Effizienz-Werte, verglichen mit den anderen Testszenen, erzielt werden konnten. Als Schlußfolgerung ist hieraus zu ziehen, daß die parallelen Berechnungen besser skalieren, wenn entweder die (Teil-) Szenen eine angemessene Komplexität besitzen oder der lokale Berechnungsaufwand aufgrund einer größeren Anzahl von Sample-Punkten bei der Formfaktorberechnung oder einer großen Substrukturierungstiefe ausreichend ist. In beiden Fällen ist der Kommunikationsaufwand verglichen mit dem Berechnungsaufwand gering. Es ist jedoch nicht unbedingt sinnvoll, aus Skalierbarkeitsgründen die Anzahl der Sample-Punkte zu erhöhen oder die Substrukturierungstiefe zu vergrößern, da dieses einzig und alleine aus Gründen der Qualitätssteigerung der berechneten Radiosity-Lösung geschehen sollte. Im Fall der einfachen Testszene ist es somit nicht sinnvoll diese Parameter zu modifizieren, da die Lösungsqualität ausreichend ist. Im Fall der Szene *Rosenthaler-Hof* ist die Modifikation beider Parameter durchaus sinnvoll. Dieses würde zu einer erheblichen Verbesserung der Effizienz der parallelen Berechnungen gerade für große Anzahlen von Prozessoren führen. Es ist zu erwarten, daß in diesem Fall auf deutlich mehr als 40 Prozessoren noch eine sehr gute Effizienz zu erzielen ist.

Aus den Testmessungen kann das Fazit gezogen werden, daß die Berechnungszeiten zur Durchführung einer Radiosity-Simulation durch das vorgestellte parallele Verfahren selbst für komplexe Eingabeszenen vom Stundenbereich auf einen kleinen Minutenbereich reduziert werden. Die Skalierbarkeitseigenschaften sind hierbei im Vergleich zu anderen existierenden

parallelen Progressive-Refinement-Verfahren (siehe Tabelle 5.4) als sehr gut zu bezeichnen. Selbst mit mehr als 32 Prozessoren kann für moderat komplexe und komplexe Szenen eine Effizienz von deutlich über 50% erzielt werden.

Der größte Vorteil des datenparallelen Verfahrens ist jedoch, daß eine Radiosity-Berechnung in komplexen Szenen ermöglicht wird, deren Durchführung mit herkömmlichen sequentiellen Methoden nicht möglich ist. Es ist zu beachten, daß in diesem Abschnitt zu Zwecken der Skalierbarkeitsuntersuchungen die Szenen und Parameter zur Steuerung der Berechnungen derart gewählt wurden, daß eine sequentielle Referenzmessung ermöglicht wurde. Da die parallelen Berechnungen jedoch datenparallel ausgeführt werden, können Szenen beliebiger Komplexität als Grundlage einer Radiosity-Berechnung verwendet werden, indem eine entsprechende Anzahl von Prozessoren verwendet wird, um den Speicherplatzbedarf zu decken. Die größte bisher verwendete Eingabeszene wurde initial in mehr als 1.000.000 Patches unterteilt, wobei die entsprechende Radiosity-Lösung auf 40 Prozessoren berechnet wurde.

5.2.6 Dynamischer Lastausgleich

In dem vorangegangenen Abschnitt 5.2.5 wurde gezeigt, daß die parallelen Radiosity-Berechnungen für komplexere Szenen bei der Verwendung von bis zu 32 Prozessoren gut skalieren (Effizienz von 60 %). Bei Berechnungen auf mehr als 32 Prozessoren ist keine weitere deutliche Reduktion der Berechnungszeiten zu beobachten. Der Grund hierfür ist, daß während der parallelen Berechnungen dynamisch ein nicht unerhebliches Lastungleichgewicht zwischen den Prozessoren entstehen kann, welches die Effizienz der Berechnungen gerade für eine größere Anzahl von Prozessoren beeinträchtigt. Die Ursachen für eine ungleichmäßige Auslastung der Prozessoren während der parallelen Radiosity-Berechnung sind spezielle Eigenschaften der Eingabeszene, die nur unzureichend durch eine statische Lastverteilung vor Berechnungsbeginn berücksichtigt werden können. Um effiziente datenparallele Berechnungen auf einer großen Anzahl von Prozessoren durchführen zu können, muß zur Laufzeit der Berechnungen dynamisch auf eine unterschiedliche Auslastung der Prozessoren des Parallelrechners reagiert werden, indem ein Ausgleich der Last zwischen den Prozessoren vorgenommen wird.

Im allgemeinen ist die Zielsetzung eines Verfahrens zum dynamischen Lastausgleich gegeben durch die Minimierung der Zeiten, in denen ein Prozessor während der parallelen Berechnungen ohne Arbeit ist. Dieses impliziert, daß die Prozessoren annähernd zum gleichen Zeitpunkt die Bearbeitung von zugeordneten Arbeitspaketen (oder Lastelemente) beenden und die parallelen Berechnungen mit möglichst geringen Wartezeiten der Prozessoren terminieren können.

Eine Strategie zum dynamischen Lastausgleich läßt sich in zwei separate Teile gliedern, den Entscheidungs- und den Migrationsteil [LMR92]. In dem Entscheidungsteil wird festgelegt, wann ein Prozessor ein Lastelement behalten oder an einen anderen Prozessor abgeben soll. Diese Entscheidung wird entweder aufgrund der lokalen Lastsituation eines Prozessors und dessen topologischen Nachbar-Prozessoren oder aufgrund der Lastsituation aller Prozessoren eines Parallelrechners getroffen. Im ersten Fall wird von einer *lokalen Entscheidungsbasis* gesprochen, während im zweiten Fall eine *globale Entscheidungsbasis* vorliegt. Im Migrationsteil werden Lastelemente zwischen den Prozessoren migriert. Um einen Lastausgleich zu erreichen, können Lastelemente zu topologisch benachbarten Prozessoren (*lokaler Migrationsraum*) oder beliebigen Prozessoren des Parallelrechners (*globaler Migrationsraum*) migriert werden. Die Auswahl der Entscheidungsbasis und des Migrationsraumes hat einen erheblichen Einfluß auf die Effizienz des dynamischen Lastausgleichsverfahrens. Eine globale

Entscheidungsgrundlage und globale Transfermöglichkeiten für Lastelemente ermöglichen zwar im allgemeinen einen besseren Lastausgleich zwischen den Prozessoren. Dieser Vorteil muß aber in den meisten Fällen durch einen erheblichen Mehraufwand an Kommunikation erkauft werden.

Grundsätzlich ist bei Verfahren zum dynamischen Lastausgleich zwischen zentralen und dezentralen Ansätzen zu unterscheiden [Heiss94]. Bei einem zentralen Ansatz existiert eine zentrale Instanz, die aufgrund von globalen Informationen entscheidet ob Prozessoren unter- bzw. überlastet sind und die Prozessoren festlegt, zwischen denen eine Migration von Lastelementen durchgeführt wird. Bei einem dezentralen Ansatz existieren auf allen Prozessoren identische Instanzen, die kooperativ einen Lastausgleich erzielen. Dezentrale Ansätze weisen bessere Skalierbarkeitseigenschaften auf, als es für zentrale Ansätze der Fall ist. Vor allen Dingen auf massiv parallelen Systemen sind die dezentralen den zentralen Ansätzen überlegen, da eine zentrale Instanz einen erheblichen Engpaß darstellt, wenn ein Lastausgleich für eine sehr große Anzahl von Prozessoren durchgeführt werden soll. Der Vorteil von zentralen Ansätzen besteht darin, daß die Verwendung einer globalen Entscheidungsbasis und eines globalen Migrationsraumes einen besseren Lastausgleich auf Systemen moderater Komplexität zuläßt.

An einem Ausgleich der Last sind immer mindestens zwei Prozessoren eines Parallelrechners beteiligt. Die Migration von Lastelementen zwischen den beteiligten Prozessoren sollte im Idealfall nicht zu einem neuen Lastungleichgewicht an anderer Stelle führen. Somit sollte eine Lastausgleichsoperation eine möglichst geringe Anzahl von Prozessoren betreffen, insbesondere eine globale Synchronisation der Prozessoren ist zu vermeiden.

In der Vergangenheit wurde im Kontext von datenparallelen Radiosity-Berechnungen nur in einer Arbeit die Problematik des dynamischen Lastausgleichs explizit betrachtet [SSV95]. Das dort geschilderte Verfahren basiert jedoch auf einem vollständig anderem Parallelisierungsansatz, als das hier vorgestellte datenparallele Verfahren. Weiterhin wird eine grundsätzlich verschiedene Methode zur Formfaktorberechnung verwendet. Aus diesen Gründen ist eine Nutzung dieses Ansatzes nicht möglich, um einen dynamischen Lastausgleich für das beschriebene datenparallele Verfahren durchzuführen.

5.2.6.1 Verschiedene Ansätze zum dynamischen Lastausgleich

Während der datenparallelen Radiosity-Berechnungen wird ein Prozessor als *ausgelastet* betrachtet, solange weitere Iterationen des Progressive-Refinement-Verfahrens ausgeführt werden können. Dieses ist nur dann möglich, wenn lokale Patches in der dem Prozessor zugeordneten Teilszene existieren, deren nicht verschossene spezifische Ausstrahlung oberhalb eines vorgegebenen Grenzwertes liegen oder wenn dem Prozessor noch externe Shooting-Patches zur Verfügung stehen, deren Energie innerhalb der lokalen Teilszene verteilt werden muß. Sobald ein Prozessor eine lokale Konvergenz erkennt, ist der Prozessor nicht ausgelastet und hat den Status *idle*. Ziel eines Verfahrens zum dynamischen Ausgleich der Last zwischen den Prozessoren muß es somit sein, dafür Sorge zu tragen, daß den Prozessoren zu jedem Zeitpunkt der parallelen Berechnungen eine annähernd gleiche Anzahl von Shooting-Patches zur Verfügung steht, deren Energie noch zu verteilen ist, bevor eine globale Konvergenz erkannt wird. Im Idealfall sollte annähernd zur gleichen Zeit in allen Teilszenen eine lokale Konvergenz erreicht werden.

Für die Durchführung datenparalleler Raytracing-Berechnungen wurden in der Vergangenheit verschiedene Ansätze zum dynamischen Lastausgleich vorgestellt. Für einige dieser Metho-

den erscheint eine Verwendung im Kontext des beschriebenen datenparallelen Radiosity-Verfahrens durchaus möglich. Aus diesem Grund sollen diese Ansätze zunächst auf ihre Verwendbarkeit überprüft werden.

Dynamische Veränderung der Teilszenen

Ein dynamischer Lastausgleich kann erzielt werden, indem die Komplexität der Teilszenen zur Laufzeit verändert wird [DS84]. Die Teilszenen wurden während der statischen Lastverteilung festgelegt. Jede der Teilszenen liegt innerhalb einer konvexen Zelle, wobei die Anzahl der Patches in jeder der Zellen annähernd gleich ist. Die Komplexität der Teilszenen läßt sich variieren, indem Form und Ausdehnung der Zellen verändert wird (siehe Abb. 5.22).

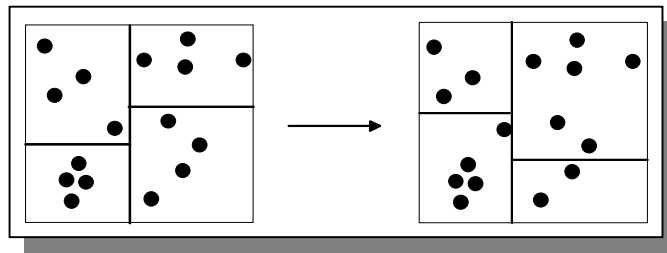


Abb. 5.22: Veränderung der Ausdehnung und Form von Zellen, indem Eckpunkte oder Zellwände verschoben werden.

Voraussetzung an eine gültige Partitionierung der Szene ist, daß jedes Patch eindeutig einer Zelle zugeordnet werden kann und das die Patches jeweils mit ihrer vollständigen Fläche innerhalb einer Zelle liegen (siehe 5.2.1). Unter Berücksichtigung dieser Restriktionen werden nun Patches zwischen benachbarten Teilszenen gemäß der neuen Form und Anordnung der Zellen ausgetauscht.

Der Lastausgleich bzw. die Veränderung der Szenenpartitionierung kann sowohl auf Basis eines lokalen als auch eines globalen Entscheidungsraumes durchgeführt werden. Dieses ermöglicht die Realisierung eines zentralen und eines dezentralen Lastausgleichsverfahrens. Bei der Verwendung eines lokalen Entscheidungsraumes kann der Lastausgleich zwischen benachbarten Prozessoren zu einem neuen Lastungleichgewicht im gesamten System führen. Im schlimmsten Fall wird eine sukzessive Modifikation aller Teilszenen notwendig.

Während eines Lastausgleiches bei der datenparallelen Radiosity-Simulation müssen die Berechnungen für die betroffenen Teilszenen synchronisiert werden, um eine korrekte Verteilung der Energie von ausgewählten Shooting-Patches zu ermöglichen. Andernfalls kann während der Migration von Objekten zwischen den Teilszenen der Fall auftreten, daß die entsprechenden Objekte bei der Verteilung von Energie gar nicht oder doppelt berücksichtigt werden.

Weiterhin ist zu beachten, daß durch eine Modifikation der Zelltopologie ein veränderter Kommunikationsgraph entsteht. Die Anzahl der Knoten bleibt hierbei konstant, aber die Kantenmenge wird verändert. Somit kann die Qualität der initial bestimmten Einbettung des Kommunikationsgraphen in den Verbindungsgraphen beliebig schlecht werden. Außerdem müssen die neu bestimmten Nachbarschaftsbeziehungen zwischen den Zellen den betroffenen Prozessoren bekannt gemacht werden, um weiterhin eine fehlerfreie Verteilung der nicht verschossenen spezifischen Ausstrahlung eines Shooting-Patches gemäß dessen Propagationsgraphen zu ermöglichen. Nach der Durchführung eines Lastausgleiches und der damit ver-

bundenen Veränderung des Kommunikationsgraphen wird der Propagationsgraph eines Shooting-Patches auf Basis des neuen Kommunikationsgraphen bestimmt. Die Propagationsgraphen aller bisher ausgewählten Shooting-Patches verändern sich. Ein externes Shooting-Patch, dessen Energie noch nicht vollständig in den Teilszenen gemäß des entsprechenden Propagationsgraphen verteilt wurde, muß gemäß der alten Partitionierung Energie verteilen. Wenn jedoch schon Objekte migriert wurden, kann keine korrekte Verteilung durchgeführt werden. Somit müssen Prozessoren verschiedene Versionen der Teilszene aufrechterhalten.

Ein dynamischer Lastausgleich durch Veränderung der Komplexität der Teilszenen erscheint kein vielversprechender Ansatz für das beschriebene datenparallele Radiosity-Verfahren. Dieses ist durch den erheblichen Einfluß eines Lastausgleichschrittes auf die Kommunikationsstruktur des parallelen Verfahrens zu begründen, welcher einen erheblichen Reorganisationsaufwand verursacht. Der entscheidende Nachteil ist jedoch, daß bei dem Lastausgleich in erster Linie geometrische Aspekte der Teilszenen berücksichtigt werden. Aufgrund der Restriktionen, denen eine gültige Partitionierung der Szene für das vorgestellte datenparallele Radiosity-Verfahren unterliegt, ist keine beliebige Änderung der Zellformen möglich. Aus diesem Grund ist kein exakter Lastausgleich zwischen benachbarten Teilszenen (und somit den Prozessen im Kommunikationsgraph) möglich.

Feingranulare Partitionierung der Szene

Einen weiteren Ansatz zur dynamischen Lastverteilung bietet die initiale Aufteilung der Szene in mehr Teilszenen als Prozessoren im Parallelrechner vorhanden sind [IAÖ91]. Der entsprechende Kommunikationsgraph wird derart in das Verbindungsnetzwerk eingebettet, daß jedem Prozessor mehrere Teilszenen zur Bearbeitung zur Verfügung gestellt werden. Dieses entspricht dem Problem einer *Many-to-One Einbettung* des Kommunikationsgraphen in den Verbindungsgraphen (siehe Abb. 5.23).

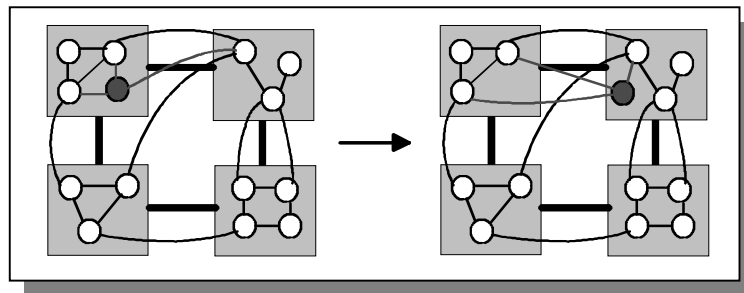


Abb. 5.23: Mehrere Teilszenen werden durch einen Prozessor bearbeitet.

Jeder Prozessor ist somit für die Ausführung mehrerer Berechnungsprozesse zur Bestimmung lokaler Radiosity-Lösungen in verschiedenen Teilszenen zuständig. Um einen Lastausgleich zu erzielen, werden Prozesse des Kommunikationsgraphen zwischen den Prozessoren migriert.

Dieser Ansatz birgt jedoch erhebliche Nachteile in sich. Wenn auf einem Prozessor mehr als ein Prozeß lauffähig ist, muß die Rechenkapazität des Prozessors auf die Prozesse aufgeteilt werden. Dieses geschieht durch die Multitasking, wobei der Prozessor der Reihe nach den Prozessen für eine gewisse Zeitspanne zur Verfügung gestellt wird. Dieses Multitasking verursacht jedoch einen nicht zu unterschätzenden Mehraufwand, wenn viele Prozesse auf einem Prozessor plaziert werden.

Ein viel größeres Problem stellt jedoch die Größe des Kommunikationsgraphen dar. Durch die Aufteilung der Szene in mehr Teilszenen als der Parallelrechner Prozessoren hat, wird ein Kommunikationsgraph erzeugt, der wesentlich mehr Knoten und Kanten hat, als es bei der ursprünglichen Eins zu Eins Partitionierung der Fall war. Somit wird auch die Anzahl der Knoten in den Propagationsgraph der Shooting-Patches vergrößert. Da mit jedem externen Shooting auch Verwaltungsaufwand für die Hemicube-Bitmaps verbunden ist (lesen der Nachrichten, Hemicube initialisieren, etc.), wächst der Aufwand für die globale Energieverteilung mit der Anzahl der Knoten im Propagationsgraph. Zusätzlich sind erheblich mehr Nachrichten notwendig, um die Energie eines Shooting-Patches in allen sichtbaren Teilszenen zu verteilen.

Der Lastausgleich zwischen Prozessoren, der durch Migration von Prozessen realisiert wird, verursacht erhebliche Kommunikationskosten, da die Teilszenen der entsprechenden Prozesse zu den Zielprozessoren transferiert werden müssen. Wenn während der Radiosity-Berechnungen eine adaptive Unterteilung von Patches durchgeführt wird, führt dieses dazu, daß die Komplexität der Teilszenen im Berechnungsverlauf erheblich anwächst. Da das größte Lastungleichgewicht zwischen den Prozessoren gerade gegen Berechnungsende auftritt, wird der Transfer von erheblichen Datenmengen bei der Migration von Prozessen notwendig, wodurch die Effektivität des Lastausgleiches zunichte gemacht wird.

Adaptive Verkleinerung der Teilszenen

Eine weitere Möglichkeit des Lastausgleiches besteht darin, die Teilszene eines überlasteten Prozessors während der Berechnungen adaptiv zu verkleinern. [NO86]. Für die gebildeten kleineren Teilszenen werden neue Berechnungsprozesse generiert, die auf unterlasteten Prozessoren gestartet werden (siehe Abb. 5.24).

Dieser Ansatz verursacht jedoch im Kontext der datenparallelen Radiosity-Berechnungen ähnliche Probleme, wie die beiden zuvor beschriebenen Lastausgleichsstrategien. Es wird eine lokale Modifikation des Kommunikationsgraphen vorgenommen. Aus diesem Grund ergibt sich eine ähnliche Problematik bei der Durchführung der externen Shooting-Iterationen durch Migration von Berechnungsprozessen, wie bei dem Ansatz der feingranularen Kommunikationsgraphen. Weiterhin werden bei der adaptiven Unterteilung von Teilszenen nur geometrische Aspekte berücksichtigt, wobei aufgrund der Anforderungen an zulässige Partitionierungen (siehe Abschnitt 5.2.1) nicht in allen Fällen kleinere Teilszenen gleicher Komplexität generiert werden können.

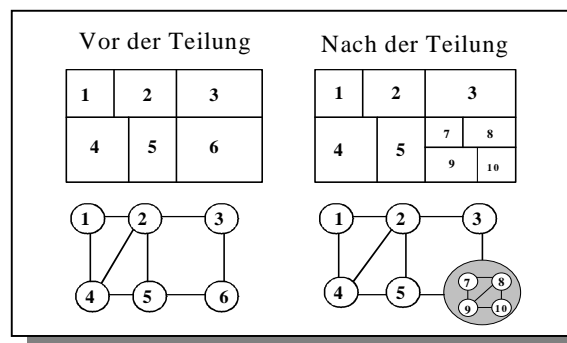


Abb. 5.24: Adaptive Unterteilung von Teilszenen.

Aus den vorangegangenen Betrachtungen kann das Fazit gezogen werden, daß verschiedene Ansätze zum dynamischen Lastausgleich während der datenparallelen Radiosity-Berechnungen aus dem Kontext der parallelen Raytracing-Berechnungen übernommen werden können. Diese Ansätze weisen jedoch erhebliche Nachteile auf und es ist zu vermuten, daß durch den notwendigen Mehraufwand, die durch diese Ansätze verursacht wird, keine wesentliche Verbesserung der Effizienz der datenparallelen Radiosity-Berechnungen erzielt werden kann. Weiterhin ist zu beachten, daß im Fall von parallelen Raytracing-Berechnungen bei einem Lastausgleich lediglich geometrische Aspekte berücksichtigt werden müssen, da die Berechnungskomplexität mit der Anzahl von lokalen Schnittpunktberechnungen korreliert. Für parallele Radiosity-Berechnungen sind jedoch Energieverteilungen innerhalb der Teilszenen von besonderer Bedeutung, die jedoch nur teilweise von geometrischen Aspekten abhängig sind.

Aus diesen Gründen wird im Rahmen des hier vorgestellten parallelen Radiosity-Verfahrens ein vollständig neuer Ansatz zum dynamischen Lastausgleich verfolgt [SR99]. Dieser Ansatz wird als *lokales Farming* bezeichnet.

5.2.6.2 Dynamischer Lastausgleich durch *lokales Farming*

Die neu entwickelte Strategie zum dynamischen Lastausgleich basiert auf der einfachen Beobachtung, daß Prozessoren, deren Teilszenen nur eine geringe Menge der insgesamt in der Szene zu verteilenden Lichtenergie empfangen (d.h. dunkle Bereiche innerhalb der Szene), sehr große Idle-Zeiten im Verlauf der Radiosity-Berechnungen aufweisen. Ein Prozessor ist nur solange ausgelastet, wie lokale oder externe Shooting-Patches zur Verfügung stehen, deren Energie innerhalb der lokalen Teilszene verteilt werden muß, bevor eine lokale Konvergenz erkannt wird. Da ein Energieaustausch zwischen den Teilszenen der partitionierten Szene durch Message-Passing realisiert wird, ist die Anzahl von Nachrichten in der Nachrichtenwarteschlange eines Prozesses ein Indikator für die externe Energiemenge, die von den lokalen Objekten der entsprechenden Zelle empfangen werden kann.

Die relative Abweichung der Anzahl von externen Shooting-Patches in den lokalen Warteschlangen der Prozessoren ist Entscheidungsgrundlage des hier vorgestellten Verfahrens zum Lastausgleich während der datenparallelen Radiosity-Berechnungen. Für jedes durch einen Prozessor empfangenes externes Shooting-Patch, für welches die relevanten Sichtbarkeitsinformationen vollständig bekannt sind (siehe 5.2.3 – logische ODER-Verknüpfung der Hemisphere-Bitmaps), ist eine weitere externe Shooting-Iteration innerhalb der entsprechenden lokalen Teilszene durchzuführen. Das Grundprinzip des Lastausgleichsverfahrens basiert auf der Durchführung der externen Shooting-Iterationen eines überlasteten Prozessors durch mehrere Prozessoren. Ziel des Verfahrens ist es, die angestauten externen Shooting-Iterationen eines überlasteten Prozessors möglichst schnell auszuführen. Hierzu wird das Konzept des *lokalen Farmings* verwendet, welches bei einem Lastausgleich primär energetische Aspekte berücksichtigt.

Das *Farming* ist in der Parallelverarbeitung ein weit verbreitetes Konzept und ermöglicht triviale Parallelisierung von sequentiellen Verfahren. Hierbei verteilt ein *Master-Prozeß* dynamisch Arbeitspakete auf eine Menge von *Worker-Prozessen*. Die durch die *Worker-Prozesse* berechneten Teilergebnisse werden an den *Master-Prozeß* übermittelt, der die Teillösungen zu einer Lösung des Ausgangsproblems kombiniert. Den *Worker-Prozessen* werden solange neue Arbeitspakete zugeordnet, bis sämtliche Teilprobleme gelöst wurden. Die ersten Parallelisierungen der Raytracing- und der Radiosity-Methode basierten exakt auf diesem Prinzip.

Beim *lokalen Farming* kann jeder Prozeß der parallelen Applikation eine eigene (lokale) Farm aufbauen. Somit existiert eine Menge von Master-Prozessen, denen eindeutig eine variable Anzahl von Worker-Prozessen zugeordnet werden kann. Durch dieses Konzept kann ein Lastausgleich zwischen den Prozessoren eines Parallelrechners erzielt werden, indem für überlastete Berechnungsprozesse eine lokale Farm generiert wird, deren Worker-Prozesse auf unterlasteten Prozessoren gestartet werden (siehe Abb. 5.1).

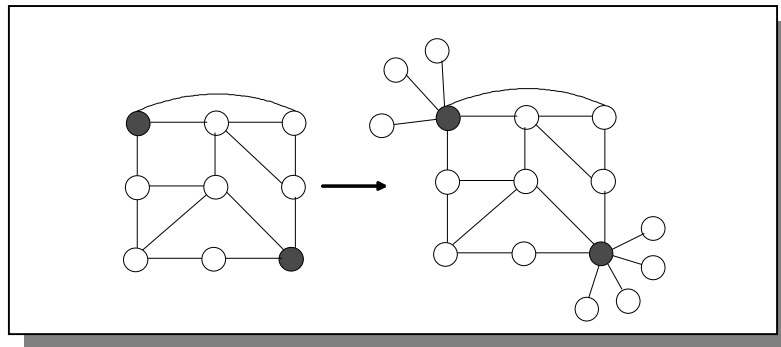


Abb. 5.25: Lokale Farmen für zwei überlastete Prozesse mit verschiedenen Anzahlen von Worker-Prozessen.

Der Lastausgleich durch lokales Farming wurde zur Durchführung eines dynamischen Lastausgleiches bei datenparallelen Radiosity-Berechnungen erstmals in [Schm94] vorgeschlagen.

Wenn für einen Prozessor während der datenparallelen Radiosity-Berechnungen eine Überlastung erkannt wird, erfolgt die Generierung einer lokalen Farm, die aus einem Master-Prozeß und einer variablen Anzahl von Worker-Prozessen besteht. Der Master-Prozeß ist identisch mit dem ursprünglichen Radiosity-Berechnungsprozeß auf dem überlasteten Prozessor. Die Worker-Prozesse sind neu generierte Berechnungsprozesse, die auf Prozessoren gestartet werden, für die eine Unterlast erkannt wurde. Da aufgrund des datenparallelen Ansatzes jeder Prozessor ursprünglich nur Zugriff auf einen kleinen Teil der Szenendaten hat, muß bei der Initialisierung einer lokalen Farm eine Kopie der entsprechenden Teilszene an die Worker-Prozesse übermittelt werden. Der Master-Prozeß verteilt nun seine Warteschlange mit externen Shooting-Patches gleichmäßig auf die Worker-Prozesse, welche die entsprechenden externen Shooting-Iterationen in den Kopien der lokalen Teilszene durchführen. Es ist zu beachten, daß die Worker-Prozesse einer lokalen Farm lediglich die ihnen zugeordneten externen Shooting-Iterationen ausführen, während der Master-Prozeß weiterhin Energie von lokalen Shooting-Patches verteilt. Nach Durchführung eines externen Shootings wird das entsprechende Shooting-Patch sowie die aktualisierten Hemicube-Bitmaps durch den Worker-Prozeß gemäß dem Propagationsgraph des Patches verschickt. Empfängt ein Master-Prozeß neue externe Shooting-Patches, so werden diese unter Berücksichtigung der Lastsituation innerhalb der lokalen Farm den Worker-Prozessen zugeordnet. Wenn die Warteschlange der externen Shooting-Patches vollständig abgearbeitet wurde oder die lokale Farm aufgrund einer veränderten Lastsituation vorzeitig abgebaut wird, übermitteln die Worker-Prozesse ihre berechneten Teilergebnisse an den Master-Prozeß. Dieser generiert eine aktuelle Radiosity-Lösung durch Addition der Patch-Radiosities aus den Kopien der Teilszene. Hierbei müssen ggf. unterschiedliche Substrukturierungstiefen eines Patches in verschiedenen Kopien der Teilszene berücksichtigt werden.

Um eine korrekte Erkennung der lokalen Konvergenz zu ermöglichen, ist es notwendig, daß die in den Kopien der Teilszenen enthaltenen Energiemengen berücksichtigt werden. Aus diesem Grund übermitteln die Worker-Prozesse in regelmäßigen Zeitintervallen die Radiosity-Werte der Patches, die den größten Teil der externen Energie empfangen haben, an den Master-Prozeß. Diese Vorgehensweise ermöglicht es auch, daß bei der Auswahl von lokalen Shooting-Patches durch die Master-Prozesse die Patches mit der jeweils annähernd größten lokal nicht verschossenen spezifischen Ausstrahlung ausgewählt werden. Somit wird durch das lokale Farming nur eine geringe Veränderung des lokalen Konvergenzverhaltens verursacht. Alternativ können die Worker-Prozesse nach jeder externen Shooting-Iteration aktualisierte Patch-Radiositäten an den Master-Prozeß übermitteln, was jedoch zu einem erheblichen Mehraufwand an Kommunikation führt.

Bei dem Lastausgleich auf Basis des lokalen Farmings handelt es sich um ein zentrales Verfahren. Der Lastausgleich zwischen den Prozessoren wird durch einen zentralen Prozeß gesteuert, welcher in zwei Phasen vorgeht. In der ersten Phase wird die relative Last der Prozessoren ermittelt. Dieses geschieht auf Basis von Lastwerten, die regelmäßig von den Radiosity-Berechnungsprozessen übermittelt werden. Die Lastwerte geben die Anzahl der externen Shooting-Patches an, die in lokalen Warteschlangen der Prozessoren gespeichert sind und für die noch externe Shooting-Iterationen ausgeführt werden müssen. Der zentrale Lastausgleichprozeß entscheidet aufgrund dieser Informationen darüber, welche der Prozesse unterlastet oder überlastet sind und berechnet die Anzahlen von Worker-Prozessen, die in den lokalen Farmen benötigt werden, um das Lastungleichgewicht innerhalb des Parallelrechners zu reduzieren. Dieses beinhaltet die Erzeugung von neuen lokalen Farmen, den Abbau bestehender lokaler Farmen sowie die Veränderung der Anzahl der Worker-Prozessen innerhalb der lokalen Farmen.

In der zweiten Phase berechnet der Lastausgleichsprozeß eine Abbildung der Worker-Prozesse auf die unterlasteten Prozessoren. Hierbei hat es sich als sinnvoll erwiesen, die Anzahl der auf einem Prozessor auszuführenden Prozesse zu beschränken, um dem begrenzten Speicherplatz auf einem Prozessor und dem durch Task-Switching entstehenden Overhead Rechnung zu tragen. In der Praxis wurden mit maximal drei Prozessen pro Prozessor die besten Resultate erzielt.

In Abb. 5.26 ist ein Beispiel für die Platzierung der Worker-Prozesse von zwei lokalen Farmen auf unterlasteten Prozessoren angegeben. Das in diesem Beispiel angegebene Szenario ist häufig am Berechnungsende einer datenparallelen Radiosity-Simulation anzutreffen. Einige wenige Prozessoren (in Abb. 5.26 sind es zwei Prozessoren) haben noch Energie in ihren lokalen Teilszenen zu verteilen, während andere Prozessoren eine lokale Konvergenz erkannt haben und somit auf die Terminierung der parallelen Berechnungen warten.

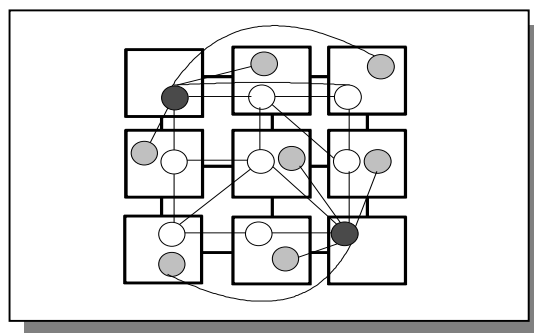


Abb. 5.26: Worker-Prozesse der lokalen Farmen werden auf unterlastete Prozessoren platziert.

Nachdem die Position der Worker-Prozesse festgelegt worden ist, werden auf den entsprechenden Prozessoren neue Berechnungsprozesse generiert. Jeder Worker-Prozeß ist eindeutig einem Master-Prozess zugeordnet. Unmittelbar nach der Erzeugung eines Worker-Prozesses signalisiert dieser dem ihm zugeordneten Master-Prozess durch eine Nachricht, daß er bereit ist Szenendaten sowie externe Shooting-Patches zur Bearbeitung zu empfangen. Im Anschluß daran werden die Berechnungen innerhalb der lokalen Farmen wie zuvor beschrieben ausgeführt.

Der Hauptvorteil des Lastausgleichs auf Basis des lokalen Farmings ist, daß ein globaler Entscheidungsraum und globaler Migrationsraum verwendet wird. Die Lastsituation aller Prozessoren wird somit bei einem Lastausgleich berücksichtigt. Bei der Verwendung von Parallelrechnern mit einer moderaten Anzahl von Prozessoren entsteht kein Engpaß durch das zentrale Verfahren. Wenn eine größere Zahl von Prozessoren verwendet wird, kann das Kommunikationsaufkommen zwischen den Berechnungsprozessen und dem zentralen Lastverteilungsprozeß verringert werden, indem die Aktualisierungszyklen der Lastinformationen verlängert werden. Dieses führt zu keiner erheblichen Verschlechterung des Lastausgleichsverfahrens, da aufgrund der langen Bearbeitungszeiten für einzelne Iterationen keine plötzlichen extremen Schwankungen der Last von Prozessoren zu befürchten ist. Ein weiterer Vorteil des lokalen Farmings ist, daß lediglich eine lokale Synchronisation zwischen Master- und Worker-Prozessen notwendig ist. Nachdem die externen Shooting-Patches innerhalb einer lokalen Farm verteilt wurden, können die Worker-Prozesse unabhängig voneinander arbeiten. Benachbarte Prozesse innerhalb des ursprünglichen Kommunikationsgraphen müssen nicht synchronisiert werden. Die Propagationsgraphen der Shooting-Patches bleiben unverändert, da der Nachrichtenaustausch auf Basis des ursprünglichen Kommunikationsgraphen (ohne lokale Farmen) erfolgt. Innerhalb einer lokalen Farm existiert eine optimale Verteilung der Last, da der Master-Prozeß bei der Zuteilung neuer externer Shooting-Patches die Auslastung der Worker-Prozesse berücksichtigt.

Trotz der genannten Vorteile des beschriebenen Lastausgleichsverfahrens auf Basis des lokalen Farmings existieren Faktoren, welche die Performanz einschränken und somit das Erreichen von einem linearen Speed-Up bei den datenparallelen Radiosity-Berechnungen verhindern. Diese Faktoren sind ein zusätzlicher Berechnungs- und Kommunikationsaufwand, der durch das Lastausgleichsverfahren verursacht wird. Die Auslastung der Prozessoren wird durch einen zentralen Prozeß kontrolliert, der auf einem Prozessor des Parallelrechnersystems gestartet wird. Dieser Prozeß verursacht somit eine Mehrbelastung eines Prozessors durch Berechnungen, die bei einer sequentiellen Radiosity-Berechnung nicht ausgeführt werden müssen. Weiterhin wird der Kommunikationsaufwand für die Bestimmung einer Radiosity-Lösung erhöht, da bei der Durchführung eines dynamischen Lastausgleiches zusätzliche Kommunikation zwischen den Berechnungsprozessen und dem zentralen Lastverteilungsprozeß und zwischen dem Master-Prozeß einer lokalen Farm und den jeweiligen Worker-Prozessen notwendig ist. Es ist jedoch zu beachten, daß diese einschränkenden Faktoren durch die Häufigkeit der Übermittlung von Lastwerten an den zentralen Lastverteilungsprozeß und die maximale Anzahl von Worker-Prozessen einer lokalen Farm beeinflusst werden können.

5.2.6.3 Ergebnisse

Um eine Bewertung der Qualität des Lastausgleiches auf Basis des lokalen Farming-Ansatzes vornehmen zu können, wurden Leistungsmessungen für die Testszenen *Hörsaal* und *Townhouse* auf dem CC-System von *Parsytec* durchgeführt. Wie in Abschnitt 5.2.5 beschrieben, waren gerade bei diesen Testszenen die Auswirkungen ungünstiger statischer Szenenpartitio-

nierungen und das dadurch entstehende dynamische Lastungleichgewicht sehr deutlich zu erkennen.

In Tabelle 5.11 sind die Berechnungszeiten für die parallelen Radiosity-Berechnungen innerhalb der Testszene *Hörsaal* auf verschiedenen Anzahlen von Prozessoren angegeben. Zusätzlich ist der entsprechende Speed-Up und die Effizienz der parallelen Berechnungen aufgeführt. Der Tabelle kann entnommen werden, daß bei den datenparallelen Berechnungen mit dynamischem Lastausgleich auf bis zu 12 Prozessoren eine Effizienz größer als 90 % und auf bis zu 28 Prozessoren eine Effizienz von mehr als 80% erreicht wird (mit der Ausnahme von 77% für 24 Prozessoren). Bei der Verwendung von bis zu 32 Prozessoren wird durch einen dynamischen Lastausgleich eine durchschnittliche Verbesserung der Effizienz von fast 10% im Vergleich zu den datenparallelen Berechnungen ohne dynamischen Lastausgleich erreicht.

Prozessoren	Berechnungszeit (in Sekunden)	Speed-Up	Effizienz (%)
1	2068	1.0	100
4	539	3.84	96
8	273	7.58	95
12	184	11.24	94
16	147	14.07	88
20	128	16.16	81
24	112	18.46	77
28	91	22.73	81
32	87	23.77	74
36	105	19.70	55
40	90	22.98	57

Tabelle 5.11: Datenparallele Berechnungen mit dynamischem Lastausgleich für die Szene *Hörsaal*.

In Abb. 5.27 wird der Speed-Up der parallelen Berechnung mit und ohne dynamischen Lastausgleich gegenübergestellt.

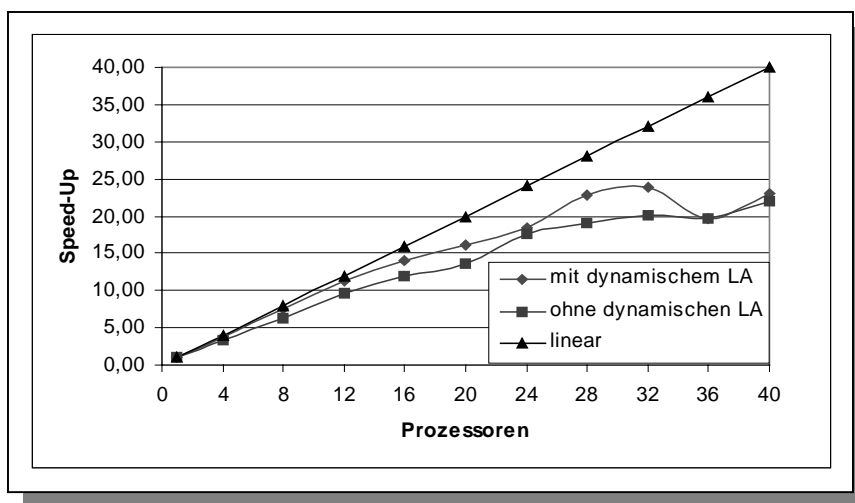


Abb. 5.27: Speed-Up der datenparallelen Radiosity-Berechnungen für die Szene *Hörsaal* mit und ohne dynamischem Lastausgleich.

In dem Diagramm ist deutlich sichtbar, daß bei der Verwendung von mehr als 32 Prozessoren keine Verbesserung der Berechnungszeiten durch das dynamische Lastausgleichsverfahren erzielt wurde. Verantwortlich für dieses Verhalten ist jedoch nicht eine Schwäche des lokalen Farmings. Die Ursache hierfür ist vielmehr in einer fehlerhaften Implementierung der PVM-Library (PowerPVM) für das CC-System zu suchen. PowerPVM hat für eine gegebene Anzahl von Prozessoren eine interne Obergrenze für die Anzahl von PVM-Tasks, die zeitgleich auf einer Partition von Prozessoren des Parallelrechners ausgeführt werden können. Für eine Partition von 40 Prozessoren liegt diese Obergrenze bei 44 PVM-Tasks, für eine Partition mit 36 Prozessoren dürfen maximal 40 PVM-Tasks gestartet werden und bei 32 Prozessoren sind nur 38 PVM-Tasks zugelassen. Aufgrund dieser Einschränkungen der Task-Anzahlen ist die Anzahl der lokalen Farmen die im Verlauf der datenparallelen Berechnungen aufgebaut werden können sowie die Farmgrößen stark begrenzt. Dieses führt dazu, daß ein dynamischer Lastausgleich durch lokales Farming auf dem CC-System nicht mehr effizient ausgeführt werden kann, sobald für mehrere Prozessoren ein Überlastung erkannt wird. Wenn bei einer Berechnung auf 40 Prozessoren initial auf jeden Prozessor ein Berechnungsprozeß plazierte wurde und im Verlauf der Radiosity-Berechnungen für fünf Prozessoren eine Überlastung erkannt wird, können aufgrund der PowerPVM-Restriktion insgesamt nicht mehr als vier Worker-Prozesse dynamisch erzeugt werden, obwohl bis zu 35 Prozessoren eine relativ geringe Last haben.

Von dieser Problematik sind insbesondere datenparallele Berechnungen mit Eingabeszenen betroffen, die sich aufgrund ihrer geometrischen Eigenschaften sehr schlecht partitionieren lassen. In diesem Fall kann es zu erheblichen Schwankungen der effektiven lokalen Berechnungszeiten für die generierten Teilszenen kommen (siehe Abschnitt 5.2.5). Dieses trifft gerade für die Szene *Hörsaal* zu. Da in dieser Szene sehr viele Objekte mit großer lokaler Komplexität existieren, ist eine gleichmäßige statische Lastverteilung mit dem verwendeten Partitionierungsverfahren nicht möglich. Hierbei werden Teilszenen gebildet, die in ihrer räumlichen Ausdehnung sehr klein sind und die im Berechnungsverlauf empfangene Energiemenge somit eher gering ist. Das energetische Ungleichgewicht zwischen den Teilszenen wird somit in der Szene *Hörsaal* sehr groß, sobald eine große Anzahl von Prozessoren bei der datenparallelen Radiosity-Berechnung verwendet wird. Es besteht somit ein großer Bedarf an Lastausgleich zwischen den Prozessoren, der durch den Aufbau von lokalen Farmen für überlastete Prozessoren realisiert wird. Da jedoch die Anzahl der dynamisch zu generierenden Worker-Prozesse durch die PowerPVM-Beschränkung stark eingeschränkt ist, kann nur ein stark eingeschränkter Lastausgleich erfolgen. Dieses resultiert in datenparallelen Berechnungen auf dem CC-System, für welche ein dynamischer Lastausgleich durch lokales Farming zu keiner Verbesserung der Effizienz bei parallelen Berechnungen führt, sobald eine zu große Anzahl von Prozessoren verwendet wird (siehe Berechnungen auf 36 und 40 Prozessoren des CC-Systems für die Szene *Hörsaal*).

Die Szene *Townhouse* läßt sich aufgrund der geometrischen Eigenschaften der Objekte und der Positionen der Lichtquellen auch für größere Anzahlen von Prozessoren derart statisch in kleinere Teilszenen unterteilen, so daß die in den Teilszenen empfangenen Energiemengen deutlich weniger voneinander Abweichen, als es für die Szene *Hörsaal* der Fall ist. Somit existieren im Berechnungsverlauf nur wenige Prozessoren, für die zur gleichen Zeit eine Überlastung erkannt wird. In diesem Fall kann trotz Beschränkungen der dynamisch zu generierenden Worker-Prozesse ein guter Lastausgleich erzielt werden.

In Tabelle 5.12 sind die Berechnungszeiten für die parallelen Radiosity-Berechnungen innerhalb der Testszene *Townhouse* für verschiedene Anzahlen von Prozessoren sowie der sich daraus ergebende Speed-Up und die Effizienz angegeben. Bei den durchgeführten Berech-

nungen konnte die durchschnittliche Effizienz auf bis zu 40 Prozessoren durch einen dynamischen Lastausgleich auf Basis des lokalen Farmings um annähernd 12% verbessert werden. Auf 40 Prozessoren wird noch eine Effizienz von 79% erzielt.

Prozessoren	Berechnungszeit (in Sekunden)	Speed-Up	Effizienz (%)
1	4117	1.0	100
4	1152	3.57	89
8	589	6.99	87
12	398	10.34	86
16	310	13.28	83
20	240	17.15	86
24	188	21.90	91
28	172	23.94	85
32	160	25.73	80
36	143	28.79	80
40	131	31.43	79

Tabelle 5.12: Datenparallele Berechnung mit dynamischem Lastausgleich in der Szene *Townhouse*.

In Abb. 5.28 wird der Speed-Up der parallelen Berechnung mit und ohne dynamischen Lastausgleich gegenübergestellt. Es ist deutlich zu erkennen, daß in diesem Fall der Lastausgleich besonders wirksam ist, wenn eine größere Anzahl von Prozessoren am Berechnungsprozeß beteiligt ist.

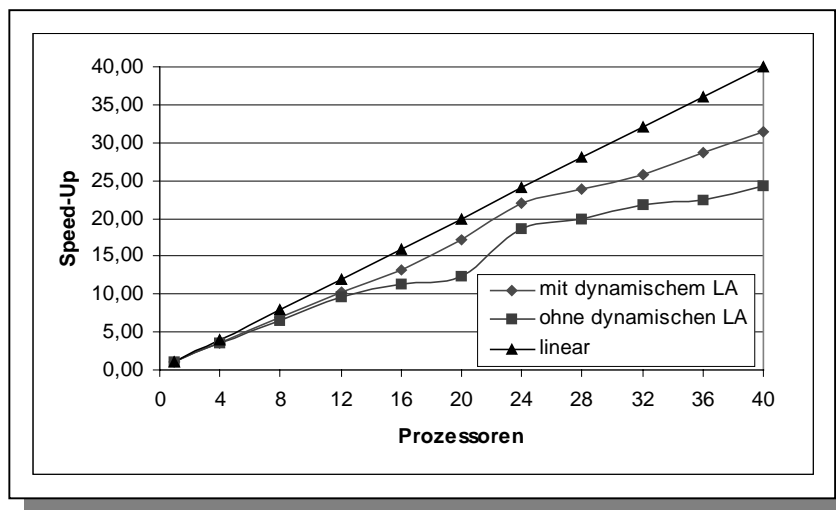


Abb. 5.28: Speed-Up der datenparallelen Radiosity-Berechnungen für die Szene *Townhouse* mit und ohne dynamischem Lastausgleich.

Wie schon in Abschnitt 5.2.5 ausgeführt wurde, kommt es aufgrund der statischen Lastverteilung zu Abweichungen der in den Teilszenen zu verteilenden Energiemengen. Hierdurch werden teilweise sehr große Idle-Zeiten der Prozessoren verursacht (siehe Tabelle 5.10 in Abschnitt 5.2.5). Ziel des Lastausgleiches auf Basis des lokalen Farming-Ansatzes ist es, diese Idle-Zeiten der Prozessoren erheblich zu reduzieren. In Tabelle 5.13 ist eine Gegenüberstellung der Idle-Zeiten der Prozessoren bei einer Radiosity-Berechnung für die Szene *Townhouse* auf acht Prozessoren mit und ohne dynamischem Lastausgleich gegeben.

	Ohne dynamischen Lastausgleich			Mit dynamischem Lastausgleich		
	<i>Berechnung + Kommunikation (Sek.)</i>	<i>idle (Sek.)</i>	<i>Ausführungs- zeit (Sek.)</i>	<i>Berechnung + Kommunikation (Sek.)</i>	<i>idle (Sek.)</i>	<i>Ausführungs- zeit (Sek.)</i>
0	538.6	101.6	640.2	585.2	4.0	589.2
1	558.3	81.8	640.1	589.2	0.0	589.2
2	529.5	110.8	640.3	589.3	0.0	589.3
3	599.3	41.0	640.3	574.6	14.6	589.2
4	513.4	126.6	640.0	589.2	0.0	589.2
5	524.6	115.6	640.2	589.3	0.0	589.3
6	637.7	2.4	640.1	585.6	3.6	589.2
7	549.5	90.8	640.3	589.3	0.0	589.3

Tabelle 5.13: Gegenüberstellung der Berechnungs- und der Wartezeiten (Idle-Zeiten) während einer datenparallelen Berechnung einer Radiosity-Lösung für die Szene *Townhouse* auf acht Prozessoren.

Es ist deutlich zu erkennen, daß die durchschnittliche Idle-Zeit der Prozessoren bei Durchführung eines dynamischen Lastausgleiches erheblich reduziert und somit die gesamte Ausführungszeit für die datenparallele Radiosity-Berechnung verkürzt wird. Für den größten Teil der Prozessoren können die Idle-Zeiten vollständig eliminiert werden. Lediglich für einige wenige Prozessoren sind noch Idle-Zeiten zu erkennen. Diese Idle-Zeiten sind dadurch zu begründen, daß die Lastwerte nur in festgelegten Zeitintervallen an den zentralen Lastverteilungsprozeß übermittelt werden. Ist dieses Zeitintervall zu groß gewählt, tritt eine Verzögerung bei dem Ausgleich eines Lastungleichgewichtes auf, wodurch es zu kleineren Idle-Zeiten kommen kann, die sich aber im Verlauf der Berechnungen summieren.

Das Fazit aus den durchgeführten Testmessungen ist, daß das Hauptziel des Lastausgleiches durch lokales Farming erreicht wurde. Das Ziel bestand hierbei in der Reduzierung von Idle-Zeiten der Prozessoren und die daraus resultierende Verkürzung der Berechnungszeiten. Die Effizienz der datenparallelen Berechnungen kann durch einen dynamischen Lastausgleich auf Basis des lokalen Farmings deutlich verbessert werden. Lediglich die geschilderte PowerPVM-Problematik beeinträchtigt die Performanz.

5.2.7 Dynamische Szenen

Mit Hilfe der in dieser Arbeit beschriebenen parallelen Radiosity-Methoden soll eine Alternative zu dem bisher in Modellierungs-Systemen gültigem Paradigma *Modeling then Rendering* geschaffen werden, bei welchem die Prozesse der Modellierung und der Bildgenerierung strikt getrennt sind und nacheinander ablaufen. Aus Anwendersicht ist es wünschenswert, unmittelbar nach der Durchführung einer Modifikation der Szenenbeschreibung anhand einer photorealistischen Darstellung einen Eindruck davon zu bekommen, wie sich diese Veränderung auf die globale Beleuchtungssituation der Umgebung auswirkt. Dieses bedeutet einen Paradigmawechsel von *Modeling then Rendering* zu *Rendering while Modeling* [Chen90].

In Kapitel 4 wurden existierende sequentielle Verfahren beschrieben, die einem Benutzer die Modifikation der Reflexionseigenschaften von Oberflächen, das Setzen und Löschen von Lichtquellen sowie das Verändern der Geometrie der Szene zur Laufzeit der Radiosity-Berechnung erlauben. Die effizienten inkrementellen Verfahren (siehe Abschnitt 4.3) passen im Anschluß an eine Benutzerinteraktion die berechnete Radiosity-Lösung an die veränderte Situation an und setzen den regulären Konvergenzprozeß (Progressive-Refinement-

Verfahren) fort. Diese Vorgehensweise verhindert, daß der vollständige Lösungsprozeß im Anschluß an eine Benutzerinteraktion mit der veränderten Szenenbeschreibung wiederholt werden muß. Die Prozesse der Modellierung und der Beleuchtungssimulation werden hierbei durch eine Modellierungs- und eine Radiosity-Komponente realisiert, die auf verschiedenen Rechnersystemen gestartet werden können und miteinander kommunizieren. Somit kann ein Benutzer Veränderungen an der Szenenbeschreibung vornehmen, während die Berechnung einer Radiosity-Lösung parallel dazu ausgeführt wird. Im Fall einer Veränderung der Szene durch den Benutzer wird durch die Modellierungskomponente eine entsprechende Nachricht initiiert und an die Berechnungskomponente übermittelt. Der Benutzer kann hierbei mehrere Interaktionen in Folge ausführen, ohne auf die Konvergenz der Berechnungen warten zu müssen. Aufgrund der Progressive-Refinement-Berechnungen können zu jedem Zeitpunkt aktualisierte Radiosity-Lösungen visualisiert werden.

Trotz der sehr kurzen Berechnungszeiten der inkrementellen Methode im Anschluß an eine Benutzerinteraktion sind die Aktualisierungszeiten noch zu groß, um den Anforderungen interaktiver Anwendungen gerecht zu werden. Die parallelen inkrementellen Radiosity-Berechnungen stellen eine Möglichkeit dar, die Aktualisierungszeiten einer gegebenen Radiosity-Lösung im Anschluß an eine Veränderung von Objektattributen (Farbe, Emission) und der Geometrie der Szene deutlich zu verkürzen.

5.2.7.1 Modifikation des datenparallelen Radiosity-Verfahrens

Die in den vorangegangenen Abschnitten beschriebene datenparallele Radiosity-Methode wurde derart modifiziert, daß in den lokalen Teilszenen das inkrementelle Radiosity-Verfahren von Chen [Chen90] ausgeführt wird [SRRP98]. Da Änderungen in einer Szene vorwiegend lokale Auswirkungen haben, sind auch die Bereiche der Szene räumlich begrenzt, für die inkrementelle Radiosity-Werte zu berechnen sind. Diese Lokalität der inkrementellen Berechnungen kommt der auf Objektraumaufteilung basierende Parallelisierungsstrategie in idealer Weise entgegen, da im Falle einer Szenenmodifikation die Korrekturberechnungen hauptsächlich nur innerhalb der durch die Veränderung direkt betroffenen Teilszenen notwendig sind. In den übrigen Teilszenen wird sehr schnell der normale Radiosity-Lösungsprozeß fortgesetzt, bis eine globale Konvergenz erkannt wird. Wenn mehrere Modifikationen in Folge vorgenommen werden, haben diese in der Regel Auswirkungen auf die Beleuchtung verschiedener Teilszenen der partitionierten Eingabeszene, was zu einer natürlichen Verteilung der Berechnungslast während der parallelen Berechnungen führt.

Die grundlegenden Mechanismen der datenparallelen Berechnungen bleiben unverändert. Dieses gilt insbesondere für die statische Lastverteilung, die Einbettung des Prozeßgraphen auf den Verbindungsgraphen des Parallelrechners sowie die Kommunikationsmechanismen und Kommunikationsstruktur bei den parallelen Berechnungen. Lediglich das Verfahren zur Radiosity-Berechnung innerhalb der lokalen Teilszenen muß gemäß der in Abschnitt 4.3 beschriebenen Methoden erweitert werden. Hierzu sind jedoch nur einige wenige Änderungen notwendig.

Eine Modifikation der Szene wird mittels einer *Modifikationsnachricht* der Modellierungskomponente allen Prozessoren (Berechnungsprozessen) bekannt gemacht. Im Fall einer Änderung von Eigenschaften eines Objektes wie Farbe oder Energieabstrahlung können inkrementelle Radiosity-Werte für das betroffene Objekte direkt nach erhalten der Modifikationsnachricht gemäß Abschnitt 4.3.1 bestimmt werden. Im Anschluß daran werden die lokalen Progressive-Refinement-Berechnungen fortgesetzt. Die Korrekturwerte werden im weiteren

Verlauf der Berechnungen innerhalb der gesamten Szene propagiert, wodurch auch die indirekten Beleuchtungseffekte an die veränderten Gegebenheiten angepaßt werden.

Wenn die Geometrie verändert wird, indem ein neues Objekt in die Szene eingefügt wird, muß die Geometrie dieses Objektes an betroffene Prozessoren übermittelt werden. Hierbei werden die Patches des dynamischen Objektes gemäß der initial bestimmten Struktur von konvexen Zellen (siehe Abschnitt 5.2.1) auf die Prozessoren verteilt. Ein Prozessor bekommt genau die Patches zugeordnet, die innerhalb der konvexen Begrenzungszelle seiner lokalen Teilszene liegen. Die entsprechenden Objektdaten werden in die Datenstruktur übernommen, welche die jeweilige lokale Teilszene speichert. Es ist zu beachten, daß sich durch eine Modifikation der Szenengeometrie die initiale statische Szenenaufteilung und somit die Komplexität der Teilszenen mitunter erheblich verändert, wodurch ein Lastungleichgewicht verursacht werden kann, wenn sich Einfüge- und Löschoptionen auf eine Teilszene konzentrieren.

Bei einer Löschoption darf das entsprechende Objekt nicht aus den lokalen Datenstrukturen entfernt werden, da die Geometriedaten bei der Berechnung von inkrementellen Formfaktoren zur Korrektur von Radiosity-Werten benötigt werden (siehe 4.3.2). Für das gelöschte Objekt wird lediglich der Löszeitpunkt vermerkt, welcher anzeigt, daß dieses Objekt nicht Bestandteil der aktuellen Szene ist (siehe 4.3.4). Alle weiteren Operationen, welche die Geometrie der Szene beeinflussen (z.B. Objektbewegung), werden durch eine Löschoption gefolgt von einer Einfügeoption berücksichtigt.

Jeder Prozessor verwaltet eine lokale Geometry-Queue (4.3.4), in der sämtliche Geometrieänderungen der Szene protokolliert werden. Die Geometry-Queue ermöglicht die lokale Rekonstruktion der für eine Shooting-Iteration relevanten Szene. Das Konzept der Geometry-Queue erlaubt die Durchführung mehrerer aufeinanderfolgender Geometrie-Änderungen, ohne auf die Konvergenz der Radiosity-Berechnungen bzgl. einer zuvor ausgeführten Modifikation warten zu müssen. Aktualisierungsberechnungen werden je nach Wichtigkeit (d.h. Einfluß auf die globale Beleuchtung) durchgeführt.

Die Auswahl eines lokalen Shooting-Patches erfolgt unter Berücksichtigung der absoluten nicht verschossenen spezifischen Ausstrahlung eines Patches sowie der Anzahl von Geometrieänderungen, die bei der Verteilung der Energie beachtet werden müssen. Das entsprechende Auswahlverfahren wurde in Abschnitt 4.3.5 beschrieben. Hierbei werden normale Progressive-Refinement-Iterationen bzgl. der aktuellen Szene und Iterationen zur Korrektur von Radiosity-Werten, die bzgl. einer älteren Szenenbeschreibung verteilt wurden, verzahnt ausgeführt, um eine schnellere Konvergenz der inkrementellen Berechnungen zu erreichen.

Im Anschluß an eine Veränderung der Szene müssen zuvor berechnete Radiosity-Werte inkrementell an die neuen Gegebenheiten angepaßt werden. Im Fall der Änderung von Objekteigenschaften gestaltet sich dieses recht einfach, da zunächst nur lokale Korrekturwerte für das veränderte Objekt berechnet werden müssen, die im weiteren Verlauf der Berechnungen automatisch innerhalb der Szene propagiert werden. Wurde jedoch eine Veränderung der Geometrie durchgeführt, müssen inkrementelle Formfaktoren berechnet werden (siehe 4.3.2). Wie in 4.3.3 und [Chen90] beschrieben, werden zwei Hemicubes benötigt, um inkrementelle Formfaktoren zu berechnen. Auf einen Hemicube wird lediglich der statische Teil der jeweils aktuellen Szenen projiziert. Auf den anderen Hemicube wird der dynamische Teil der Szene projiziert. Die Berechnung von inkrementellen Formfaktoren läßt sich sehr einfach in die bei den datenparallelen Radiosity-Berechnungen verwendete analytische Methode zur Bestimmung von Formfaktoren (siehe 5.2.2.1) integrieren. Die inkrementelle Sichtbarkeit für einen Sample-Punkt der Umgebung wird dann durch einen Vergleich des Tiefenwertes eines

projizierten Sample-Punktes mit den Inhalten der entsprechenden Hemicube-Pixel der beiden Hemicubes bestimmt. Die für den Shooting-Vorgang relevante statische Szene wird auf Basis des Geometry-Queue Zählers des aktuellen Shooting-Patches bestimmt (siehe 4.3.4).

Da zwei separate Hemicubes bei der Berechnung inkrementeller Formfaktoren benötigt werden, verändert sich der Aufbau der Nachrichten bei datenparallelen Berechnungen. Wenn eine Iteration zur Korrektur von zuvor verteilten Energieanteilen durchgeführt wird, enthalten die entsprechenden Nachrichten die Bitcodierungen von zwei Hemicubes, die bei externen Shooting-Iterationen berücksichtigt werden müssen. Das Routing-Schema der Nachrichten gemäß dem Propagationsgraph eines Shooting-Patches (Abschnitt 5.2.3.4) verändert sich nicht.

5.2.7.2 Ergebnisse

Um die Leistungsfähigkeit des parallelen Radiosity-Verfahrens zur Unterstützung dynamisch veränderbarer Szenen zu demonstrieren, wurden Testmessungen auf dem CC-System von Parsytec mit Szenen mittlerer Komplexität durchgeführt.

Es ist zu beachten, daß bei den Testmessungen ein eingeschränktes Anwendungsszenario verwendet wurde. Dieses Anwendungsszenario gewährleistet, daß die Radiosity-Komponente eine Szenenmodifikation während der sequentiellen Referenzmessungen und den jeweiligen datenparallelen Berechnungen in dem gleichen Konvergenzstadium empfängt und somit die bis zum Empfangszeitpunkt der Modifikationsnachricht berechneten Radiosity-Lösungen übereinstimmen. Aus diesem Grund wird bei jeder Messung zunächst eine Konvergenzschranke vorgegeben und die entsprechende Radiosity-Lösung berechnet. Die Berechnungsprozesse gehen im Anschluß daran in eine Warteschleife über und warten auf den Empfang einer Modifikationsnachricht. Gemäß dieser Modifikation werden dann Korrekturberechnungen durchgeführt, bis erneut eine konvergierte Lösung bestimmt ist. Dieses Anwendungsszenario wird als eingeschränkt bezeichnet, da bei einer normalen Verwendung des inkrementellen Verfahrens eine sehr kleine Konvergenzschranke vorgegeben wird. Des Weiteren werden die Modifikationen der Umgebungsbeschreibungen vorgenommen, bevor die Radiosity-Berechnungen zu einer konvergierten Lösung führen. Weiterhin werden normalerweise mehrere Modifikationen in Folge ausgeführt, die i.a. Auswirkungen auf verschiedene Bereiche der Szene haben. Hierdurch wird eine natürliche Verteilung der Berechnungslast bei den datenparallelen Berechnungen erzeugt. Diese ist in dem eingeschränkten Anwendungsszenario nicht gegeben. Somit sind die datenparallelen Berechnungen in einem realen Anwendungsszenario noch deutlich effizienter, als es in dem hier verwendeten Anwendungsszenario möglich ist. Weiterhin ist zu beachten, daß in dem eingeschränkten Szenario die Zeiten zur Berechnung einer konvergierten Lösung gemessen werden. Die direkten Auswirkungen einer Szenenmodifikation (z.B. Schattenbildung durch Einfügen eines neuen Objektes) sind erheblich schneller in Zwischenlösungen des Progressive-Refinement-Verfahrens enthalten und können somit bei der Bildgenerierung mit Verzögerungen im sehr kleinen Sekundenbereich berücksichtigt werden.

Zunächst wurden Messungen auf Grundlage einer Szene durchgeführt, die zwei aneinandergrenzende Büroräume repräsentiert. Innerhalb dieser Szene wurden verschiedene Modifikationen vorgenommen und die inkrementellen Korrekturberechnungen in dem eingeschränkten Meßszenario durchgeführt. Die Berechnungszeiten für die datenparallelen inkrementellen Berechnungen und die entsprechenden sequentiellen Referenzmessungen sind in Tabelle 5.14 angegeben. Es wurden maximal 12 Prozessoren bei den parallelen Berechnungen verwendet und es wurde eine sehr kleine Konvergenzschranke vorgegeben.

Büroräume			
Prozessoren	Modifikationen		
	Verändern der Reflektivität Update (Sek.)	Verschieben eines Tisches Update (Sek.)	Verschieben einer Lichtquelle Update (Sek.)
1	199	1630	895
2	105	912	473
4	67	636	305
6	50	522	242
8	42	427	206
12	27	380	174

Tabelle 5.14: Zeiten für inkrementelle Radiosity-Berechnungen in einer dynamisch veränderbaren Szene für verschiedene Modifikationen und Anzahlen von Prozessoren.

Es ist ersichtlich, daß die Aktualisierungszeiten durch Verwendung von mehreren Prozessoren bei der Durchführung der inkrementellen Berechnungen deutlich verkürzt werden können. Insbesondere die inkrementellen Berechnungen im Anschluß an die Veränderung der Reflektivität (hier: Farbe) eines Objektes, führen in sehr kurzer Zeit zu einer konvergierten Radiosity-Lösung.

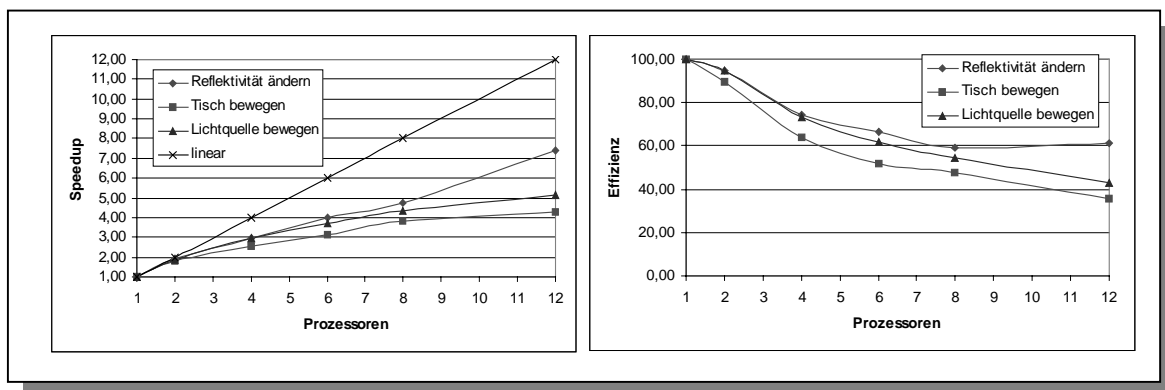


Abb. 5.29: Speed-Up (links) und Effizienz (rechts) der inkrementellen Radiosity Berechnungen für die Testszene *Büroräume*.

Der Speed-Up und die Effizienz der datenparallelen inkrementellen Berechnungen sind in Abb. 5.29 graphisch dargestellt.

Die beste Effizienz wird bei den parallelen Aktualisierungsberechnungen im Anschluß an die Änderung von Objekteigenschaften (Farbe eines größeren Objektes - Wand) erreicht. Hierbei ist die Effizienz auf bis zu 12 Prozessoren größer als 60 %. Da Aufgrund der Größe des veränderten Objektes die Veränderung der Wandfarbe Auswirkungen auf eine größere Anzahl von Teilszenen hat, sind die meisten der verwendeten Prozessoren an den inkrementellen Berechnungen beteiligt. Die inkrementelle Berechnungslast ist dabei bis zu einem gewissen Grad gleichmäßig auf die Prozessoren verteilt. Bei der Modifikation der Szenengeometrie wird eine deutlich geringere Effizienz erreicht, wobei jedoch zwischen den Arten der Änderungen unterschieden werden muß. Die Bewegung einer Lichtquelle beeinflusst nicht nur geometrische Eigenschaften der Szene (statische Lastverteilung), sondern hat zusätzlich erhebliche Auswirkungen auf die globale Energieverteilung innerhalb der Szene. Somit werden komplexere inkrementelle Berechnungen nötig, die mehrere Teilszenen beeinflussen. Dieses führt zu einer besseren dynamischen Verteilung der Last, als es bei einer einfachen Änderung

der Geometrie der Fall ist, die lediglich sehr lokale Auswirkungen hat. Im Fall einer solchen einfachen Geometrieänderung müssen nur für eine kleine Anzahl von Teilszenen inkrementelle Berechnungen durchgeführt werden. Aus diesem Grund ist die Effizienz der datenparallelen inkrementellen Berechnung im Anschluß an die Bewegung der Lichtquelle besser, als wenn lediglich ein Objekt bewegt wird, das keine Strahlung in die Umgebung abgibt.

Interessant ist die Fragestellung, wieviel schneller die inkrementellen Berechnungen im Anschluß an eine Modifikation der Szene zu einer konvergierten Lösung führen, als es bei einer vollständigen Propagation der Lichtenergie bzgl. der veränderten Szenenbeschreibung (d.h. parallele Neuberechnung der Radiosity-Lösung) der Fall ist. Dieses soll nun im weiteren Verlauf dieses Abschnittes untersucht werden. Grundlage dieser Untersuchungen ist eine Szene mit moderater Komplexität. Die Szene repräsentiert eine Offizierskabine in dem Segelschiff *Endeavour* von Captain Cook. Das vollständige Schiffsmodell wurde von der schweizer Firma AIA im Rahmen des EU-Projektes SICMA modelliert. In Tabelle 5.15 und Tabelle 5.16 sind die Berechnungszeiten einer kompletten Neuberechnung der Radiosity-Lösung gemäß der veränderten Umgebungsbeschreibung den Berechnungszeiten des inkrementellen Verfahrens gegenübergestellt. Hierbei wurde eine grundsätzliche Trennung zwischen Modifikationen der Geometrie der Szene und Veränderungen von Objekteigenschaften vorgenommen. Die Zeiten sind im Rahmen des eingeschränkten Anwendungsszenarios ermittelt worden.

Green's Cabin (6221 Patches)						
Berechnungszeiten in Sekunden						
Prozessoren	Objekt verschieben		Objekt löschen		Objekt einfügen	
	Propagation	inkrementell	Propagation	inkrementell	Propagation	Inkrementell
1	464,30	430,40	465,09	300,39	504,53	324,10
2	342,88	295,94	342,89	207,87	378,48	227,24
4	184,24	182,12	184,23	119,24	220,85	138,14
6	161,59	111,59	162,32	79,43	160,51	78,88
8	215,55	118,56	219,34	85,99	219,23	85,89
12	205,72	117,39	205,11	85,07	221,81	85,63
16	266,30	101,05	260,68	71,31	268,60	71,52

Tabelle 5.15: Modifizieren der Geometrie der Szene.

Green's Cabin (6221 Patches)						
Berechnungszeiten in Sekunden						
Prozessoren	Lampe aus		Lampe ein		Reflektivität ändern	
	Propagation	inkrementell	Propagation	Inkrementell	Propagation	Inkrementell
1	450,02	121,93	526,07	143,33	554,74	71,90
2	333,37	108,07	378,81	104,44	431,06	67,72
4	192,49	67,94	221,41	62,48	253,22	37,83
6	146,07	43,94	162,85	43,30	179,67	23,34
8	210,86	35,96	220,81	41,66	240,84	34,44
12	203,49	30,03	212,86	44,54	233,47	34,11
16	250,54	33,20	271,75	39,26	279,89	32,35

Tabelle 5.16: Modifizieren von Objekteigenschaften.

Es ist zu beachten, daß bei der vollständigen Propagation keine neue statische Lastverteilung (d.h. Szenenpartitionierung) durchgeführt wird, bevor die Neuberechnungen in der veränderten Szene durchgeführt werden. Ein neu eingefügtes Objekt wird analog zu den inkrementellen Berechnungen in die betroffenen Teilszenen eingefügt. Diese Vorgehensweise ermöglicht

einen besseren Vergleich der entsprechenden Berechnungszeiten. Die datenparallelen Berechnungen wurden auf maximal 16 Prozessoren durchgeführt, wobei im Berechnungsverlauf kein dynamischer Lastausgleich vorgenommen wurde.

Aus Gründen der Übersicht sind die Berechnungszeiten aus Tabelle 5.15 und Tabelle 5.16 in Abb. 5.30 graphisch dargestellt.

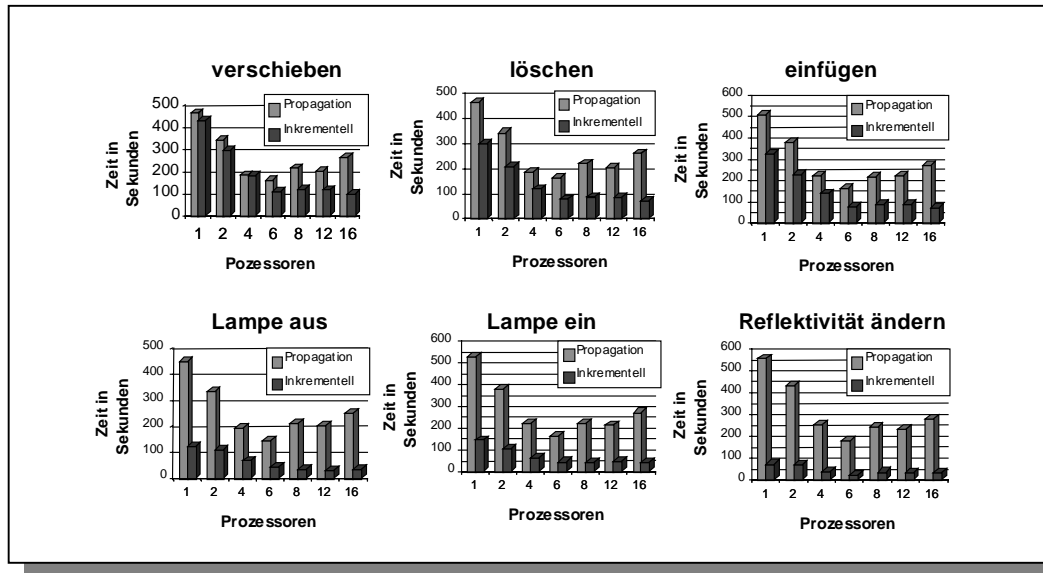


Abb. 5.30: Gegenüberstellung der Berechnungszeiten der inkrementellen Berechnungen und der erneuten Propagation der Lichtenergie in der modifizierten Szene.

Die Berechnungszeiten können sowohl bei den inkrementellen als auch bei den vollständigen Neuberechnungen erheblich verkürzt werden, indem mehr Prozessoren bei den Berechnungen verwendet werden. Die Aktualisierungszeiten der inkrementellen Berechnungen sind jedoch erheblich kürzer als die entsprechenden Zeiten zur vollständigen Neuberechnung einer Radiosity-Lösung. Dieses trifft insbesondere auf die Aktualisierungsberechnungen im Anschluß an die Modifikation von Objekteigenschaften (Lampe ein, Lampe aus, Reflektivität ändern) zu. Hauptursache hierfür ist, daß bei Veränderungen von Objekteigenschaften keine inkrementellen Formfaktoren berechnet werden müssen. Außerdem verteilt das inkrementelle Verfahren zur Berechnung einer konvergierten Lösung lediglich Korrekturenergie, d.h. Energieanteile welche aufgrund der Modifikation der Szene notwendigerweise umverteilt werden müssen. Bei einer vollständigen Neuberechnung einer konvergierten Lösung müssen hingegen auch Energieanteile verteilt werden, die in keinem Zusammenhang mit der Szenenmodifikation stehen. Dieses führt zu einem erheblichen Mehraufwand. Wenn Modifikationen an der Szene vorgenommen werden, die große Auswirkungen auf die geometrische Anordnung von Objekten innerhalb der Szene haben, muß eine große Anzahl von inkrementellen Formfaktoren berechnet werden. Bei solchen Berechnungen bringt die sequentielle inkrementelle Radiosity-Methode keine deutliche Verkürzung der Aktualisierungszeiten im Vergleich zur erneuten vollständigen Radiosity-Berechnung in der veränderten Szene mit sich (siehe Objekt verschieben). Bei den parallelen inkrementellen Berechnungen auf mehreren Prozessoren können jedoch deutliche Geschwindigkeitsvorteile erreicht werden. Dieses ist durch eine Veränderung der Lastverteilung aufgrund der Modifikation zu erklären. Eine Reduktion der Zeiten für eine Neuberechnung der Radiosity-Lösung können nur auf bis zu sechs Prozessoren erzielt werden. Bei Verwendung von mehr als sechs Prozessoren verlängern sich die Berechnungszeiten. Dieses ist bei den parallelen inkrementellen Berechnungen nicht der Fall. Eine Verbes-

serung der Berechnungszeiten wird auf bis zu 16 Prozessoren erzielt. Somit bringt die vorgestellte Parallelisierungsstrategie insbesondere Vorteile für inkrementelle Radiosity-Berechnungen mit sich.

Der Sachverhalt der Beschleunigung der Aktualisierungsberechnungen einer Radiosity-Lösung durch Verwendung des parallelen inkrementellen Verfahrens im Anschluß an eine Modifikation wird in Abb. 5.31 veranschaulicht.

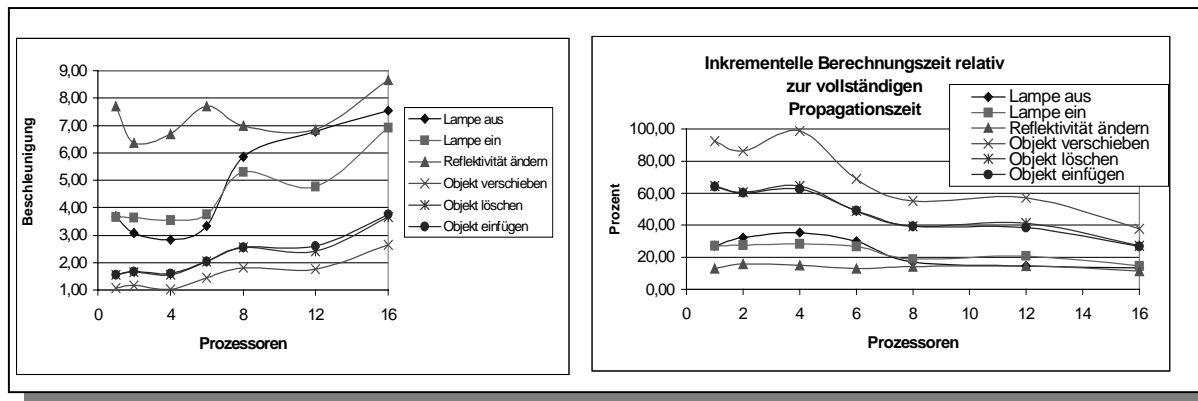


Abb. 5.31: Beschleunigung der Aktualisierungsberechnungen einer Radiosity-Lösung durch Verwendung des parallelen inkrementellen Verfahrens im Anschluß an eine Modifikation .

In dem linken Diagramm ist der Faktor graphisch dargestellt, um den die Aktualisierungsrechnungen beschleunigt werden, wenn das parallele inkrementelle Verfahren verwendet wird, anstatt eine vollständige Neuberechnung der Radiosity-Lösung vorzunehmen ($T_{prop}(n)/T_{inc}(n)$, $n \geq 1$). Das rechte Diagramm gibt an, wieviel Prozent der Zeit zur Berechnung einer vollständigen neuen Lösung durch die entsprechenden inkrementellen Berechnungen benötigt werden, um eine identische korrigierte Lösung zu bestimmen ($(T_{inc}(n)/T_{prop}(n)) * 100$, $n \geq 1$). Diesen Diagrammen ist sehr deutlich zu entnehmen, daß Korrekturlösungen im Anschluß an Änderungen von Objekteigenschaften wesentlich schneller mit dem inkrementellen Verfahren berechnet werden können. Weiterhin ist gerade in diesen Diagrammen sehr schön zu erkennen, daß die Parallelisierung des inkrementellen Verfahrens erhebliche Geschwindigkeitssteigerungen im Vergleich zur vollständigen Propagation mit sich bringt.

Wenn die Effizienz der parallelen Neuberechnung einer Radiosity-Lösung in der modifizierten Szene betrachtet wird (Abb. 5.32 - rechts), ist deutlich zu sehen, daß die Kurven für alle Modifikationen einen annähernd gleichen Verlauf haben. Geringfügige Abweichungen entstehen durch einer Veränderung der statischen Lastsituation. Hierbei wird entweder die Patchanzahl einiger weniger Teilszenen (Geometrieänderungen) oder die initiale Energieverteilung innerhalb der Teilszenen durch eine Modifikation verändert (Lichtquelle an- oder ausschalten). Da die Auswirkungen einer einzelnen Modifikation jedoch global betrachtet geringfügig sind und bei den vollständigen Propagationsberechnungen keine Lokalitätseigenschaften der Szenenmodifikationen berücksichtigt werden, ändert sich die Komplexität der globalen Berechnungen nicht wesentlich. Somit wird auch das Konvergenzverhalten bei den parallelen Berechnungen nur sehr wenig beeinflusst, was für den einheitlichen Verlauf der Effizienzkurven entscheidend ist.

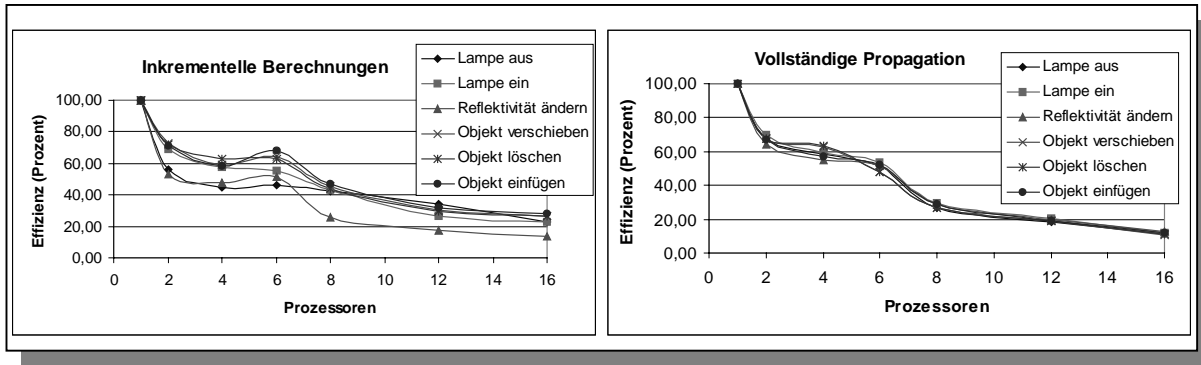


Abb. 5.32: Effizienz der parallelen inkrementellen Berechnungen (links) und der parallelen Berechnungen zur vollständigen Propagation (rechts).

Bei der Effizienz datenparalleler inkrementeller Berechnungen ist ein von der grundlegenden Tendenz ähnlicher Verlauf wie bei den vollständigen Propagationberechnungen zu beobachten (Abb. 5.32 - links). Es treten jedoch deutliche Schwankungen der Effizienz im Anschluß an eine Geometrieänderung auf. Die Ursache hierfür ist in der Lokalität dieser Modifikation zu suchen. Jede der Modifikationen hat Auswirkungen unterschiedlicher Komplexität auf die globale Beleuchtung innerhalb der gesamten Szene. Wenn eine Modifikation lediglich sehr lokale Auswirkungen hat, sind nur wenige Iterationen zur inkrementellen Anpassung einer zuvor berechneten Radiosity-Lösung notwendig. An diesen Berechnungen werden jedoch aufgrund der datenparallelen Berechnungen nur sehr wenige Prozessoren beteiligt. Dieses führt auf der einen Seite zwar zu sehr kurzen Aktualisierungszeiten, auf der anderen Seite wird jedoch die Effizienz schlechter. Dieser Zusammenhang zwischen Komplexität der inkrementellen Berechnungen im Anschluß an eine Veränderung der Szene und der Effizienz der entsprechenden datenparallelen Berechnungen führt zu dem uneinheitlichen Verlauf der Effizienzkurven. Die Effizienz der datenparallelen inkrementellen Berechnungen ist gerade für größere Anzahlen von Prozessoren wesentlich besser, als die der vergleichbaren parallelen Neuberechnungen der konvergierten Radiosity-Lösungen in der veränderten Szene.

Bei einzelnen Modifikationen, die lediglich lokale Auswirkungen haben, ist nur ein kleiner Teil der Prozessoren an dem Aktualisierungsberechnungen beteiligt. Dieses führt bei dem verwendeten Anwendungsszenario zu einer schlechten Ausnutzung der zur Verfügung stehenden Prozessoren. Somit fällt die Effizienz der parallelen Berechnungen mit einer größeren Anzahl von Prozessoren deutlich geringer aus, als es in einem normalen interaktiven Szenario der Fall ist, in dem mehrere Interaktionen im Verlauf der Radiosity-Berechnungen vorgenommen werden. Bei der Verwendung von größeren Szenen und der Durchführung von mehreren Benutzeraktionen in Folge ist zu erwarten, daß die Effizienz der parallelen inkrementellen Berechnungen auch für größere Anzahlen von Prozessoren deutlich besser ist. Problematisch hierbei ist jedoch, ein geeignetes Meßszenario praktisch zu realisieren.

Der dynamische Lastausgleich durch lokales Farming (siehe 5.2.6) stellt im Rahmen der inkrementellen Radiosity-Berechnungen keine erfolgversprechende Strategie dar, die Effizienz der parallelen Berechnungen zu steigern. Dieses ist durch den großen Overhead des Verfahrens beim Aufbau einer lokalen Farm zu begründen, welcher nur bei längeren Berechnungszeiten amortisiert wird. Weiterhin muß bei interaktiven Änderungen der Szene sehr schnell auf eine veränderte Lastsituation einzelner Prozessoren reagiert werden. Somit müssen Lastwerte in sehr kurzen Zeitabständen an den zentralen Lastverteilungsprozeß übermittelt wer-

den. Dieses würde zu einer deutlichen Steigerung des Kommunikationsoverheads führen, wodurch die Effizienz der parallelen Berechnungen verringert wird.

Abschließend soll noch einmal erwähnt werden, daß es sich bei den angegebenen Zeiten um die Berechnungszeiten einer konvergierten Lösung handelt. Die visuell deutlich sichtbaren Auswirkungen einer Szenenmodifikation auf die globale Beleuchtung können aufgrund der Progressive-Refinement-Berechnungen schon nach Korrekturberechnungen im kleinen Sekundenbereich bei der Bildgenerierung berücksichtigt werden. Somit handelt es sich bei dem vorgestellten Verfahren um einen ersten Schritt in Richtung des von Chen [Chen90] vorgeschlagenen Paradigmas *Modeling while Rendering*. Dem Autor dieser Arbeit ist zur Zeit kein weiteres paralleles inkrementelles Verfahren zur Unterstützung dynamisch veränderbarer Umgebungsbeschreibung im Kontext von Radiosity-Berechnungen bekannt.

5.2.8 Berücksichtigung von Spiegelflächen

Wie schon in Kapitel 2 ausführlich beschrieben wurde, können mit Hilfe der Radiosity-Methode nicht alle Beleuchtungseffekte simuliert werden, die in realen Umgebungen zu beobachten sind. Die Radiosity-Methode ermöglicht lediglich die Simulation der ideal diffusen Reflexion von Licht zwischen den Objekten einer gegebenen Szene. Gemäß der in Abschnitt 2.1.3 eingeführten Notation berücksichtigt die Radiosity-Methode somit Lichttransportpfade aus der Menge LD^*E . Insbesondere einfache Spiegelungen werden bei einer Bildgenerierung auf Basis einer berechneten Radiosity-Lösung durch das Gouraud-Schattierungsverfahren nicht berücksichtigt.

Eine einfache Zwei-Phasen-Methode (siehe 3.5.1) ermöglicht die Wiedergabe von perfekten Spiegelungen, Glanzlichtern und Refraktion in der abschließenden blickpunktabhängigen Bildgenerierungsphase, die im Anschluß an die Radiosity-Berechnungen durchgeführt wird. Zu diesem Zweck wird ein modifiziertes Raytracing-Verfahren verwendet, das als Eingabe die zuvor berechnete Radiosity-Lösung sowie einen Betrachterstandpunkt erhält. Anstatt die globale Beleuchtung durch einen konstanten Wert zu approximieren, berechnet das modifizierte Raytracing-Verfahren für jeden ermittelten Schnittpunkt zwischen einem Strahl und einem Objekt der Szene einen Radiosity-Wert durch bilineare Interpolation der Eckpunkt-Radiosities des *getroffenen* Patches [CW93]. Die interpolierten Radiosity-Werte werden als diffuser Beleuchtungsanteil im Beleuchtungsmodell des Raytracing-Verfahrens verwendet. Durch diese Vorgehensweise können Lichttransportpfade aus der Menge LD^*S^*E simuliert werden. Hierdurch ist eine erhebliche Qualitätssteigerung der erzeugten Bilder im Vergleich zur Visualisierung der reinen Radiosity-Methode gegeben.

Durch Kombination des zuvor beschriebenen datenparallelen Radiosity-Verfahrens mit einem parallelen Raytracing-Verfahren wurde eine einfache Zwei-Phasen-Methode realisiert [SRRP98]. Im Anschluß an die Berechnung einer blickpunktunabhängigen Radiosity-Lösung wird ein Bild mit Hilfe von Raytracing für eine bestimmte Blickrichtung unter Ausnutzung der berechneten Radiosity-Informationen generiert.

Die Parallelisierung des Raytraing-Verfahrens basiert auf einer einfachen Strategie, bei welcher die Szenendaten auf jedem Prozessor vollständig gespeichert werden. Der größte Teil der existierenden Parallelisierungen der Raytracing-Methode führt zu Zwecken der statischen Lastverteilung eine Unterteilung des gesamten Bildbereiches in Bildausschnitte gleicher Größe durch. Diese Bildausschnitte werden zu Berechnungsbeginn durch einen Master-Prozeß gleichmäßig auf Worker-Prozesse verteilt (siehe Abb. 5.33 - links).

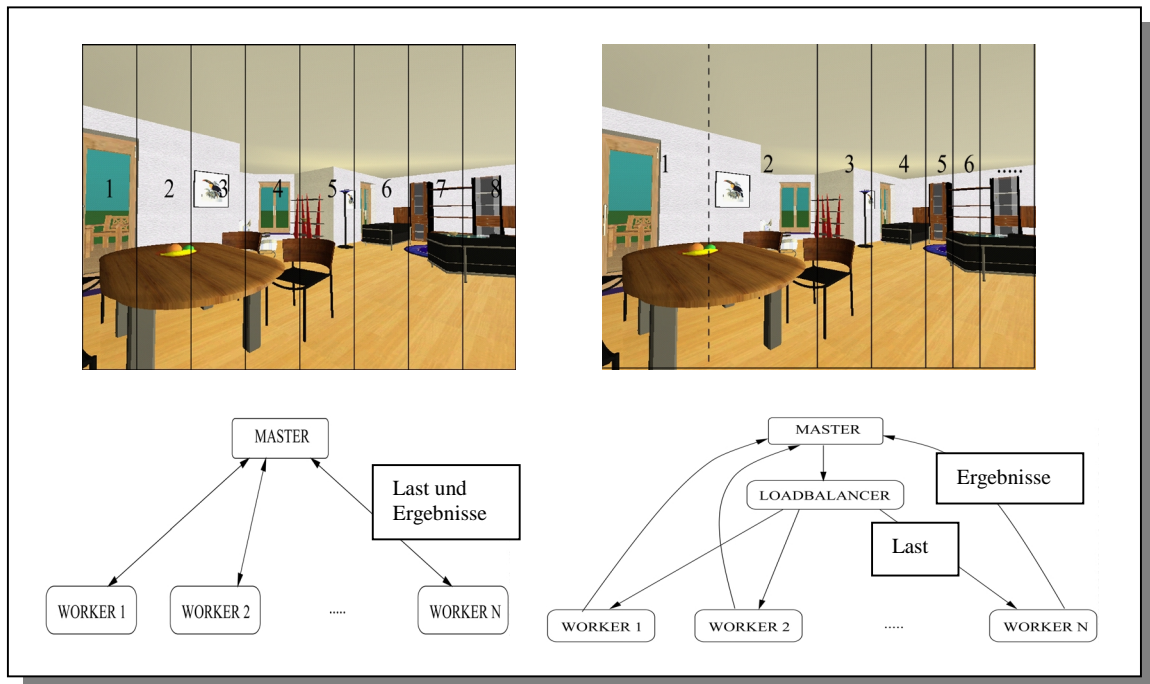


Abb. 5.33: Statische Zuweisung fester Bildausschnitte (links) vs. dynamische Zuweisung adaptiv verkleinerter Bildausschnitte (rechts).

Diese Vorgehensweise berücksichtigt nicht, daß die Berechnungszeiten von Bildausschnitten mit der gleichen Anzahl von Pixelelementen aufgrund der Materialeigenschaften der direkt oder indirekt sichtbaren Objekte (und somit der Rekursionstiefe der Strahlenbäume) erheblich voneinander abweichen können. Bei einer statischen Zuweisung von Bildausschnitten kann es zu deutlich voneinander abweichenden Berechnungszeiten der Worker-Prozesse und somit zu Idle-Zeiten kommen. Aus diesem Grund ist die Effizienz von parallelen Berechnungen auf Basis einer statischen Zuweisung von Bildbereichen schon für kleine Anzahlen von Prozessoren gering. Dieses wird im weiteren Verlauf dieses Abschnittes anhand von Testmessungen belegt.

Die Lastverteilung des im Kontext der einfachen Zwei-Phasen-Methode verwendeten parallelen Raytracing-Verfahrens basiert ebenfalls auf einer Aufteilung des Bildraumes auf die Prozessoren. Es wird jedoch eine neue adaptive, *bedarfsgesteuerte (demand-driven)* Lastverteilungsstrategie verwendet [SRRP98]. Den Prozessoren des Parallelrechners werden hierbei von einem zentralen Lastverteilungsprozeß verschiedene Bildbereiche zugeordnet, die unabhängig voneinander berechnet werden können (siehe Abb. 5.33 - rechts). Nach Berechnung eines Bildausschnittes bekommt der frei gewordene Prozessor auf Anfrage einen neuen Bildbereich zugeordnet. Die Größen der Bildausschnitte werden hierbei adaptiv in Abhängigkeit von der Anzahl der Prozessoren und der Größe des verbleibenden Bildraumes verkleinert. Diese Vorgehensweise verringert die Idle-Zeiten der Prozessoren gegen Ende der Bildgenerierung, wodurch während der parallelen Raytracing-Phase auch für eine große Anzahl von Prozessoren annähernd linearer Speed-Up erzielt wird.

Die Vorgehensweise des zentralen Lastverteilungsprozesses bei der Zuteilung von neuen Bildausschnitten an einen anfragenden Worker-Prozeß ist in Algorithmus 5.1. in Pseudocode angegeben.

```

WorkLoad = C;
Ratio = 1/(1+T*(N-1));
JobSize = (WorkLoad*Ratio);
Weise jedem Worker einen Bildausschnitt bestehend aus JobSize Pixelspalten zu;
while(WorkLoad > 0)
{
    Warte bis eine Anfrage auf einen neuen Bildbereich vorliegt;
    JobSize = WorkLoad*Ratio;
    Weise dem anfragenden Worker einen Bildausschnitt bestehend aus
    JobSize Pixelspalten zu;
    WorkLoad = WorkLoad-JobSize;
}

```

Algorithmus 5.1: Pseudocode des zentralen Lastverteilungsprozesses.

Hierbei ist N die Anzahl der Worker-Prozesse, C die Anzahl der für das Bild zu berechnenden Pixelspalten und $T \geq 1$ eine Konstante. Es soll die Annahme gelten, daß für jeweils zwei verschiedene Bildbereiche mit der gleichen Anzahl von Pixelspalten (d.h. Arbeitspakete gleicher Größe) eine Schranke T existiert mit

$$\frac{\text{Zeit für die Berechnung von Bildausschnitt}_1}{\text{Zeit für die Berechnung von Bildausschnitt}_2} \leq T. \quad (5.11)$$

Es ist zu beachten, daß der Parameter T , für beliebige Berechnungsaufträge nicht im Vorfeld bekannt ist und somit empirisch gesetzt wird. Im Verlauf von Testmessungen haben sich Werte $3 \geq T \geq 2,5$ als sinnvoll herausgestellt. Die Grundidee basiert darauf, daß wenn T bekannt ist, die Bearbeitungszeit für ein durch den angegebenen Algorithmus zugewiesenes Arbeitspaket nicht länger dauern kann, als zur Bearbeitung des verbleibenden Bildbereiches notwendig ist. In diesem Fall werden die Idle-Zeiten am Ende der parallelen Berechnungen des gesamten Bildes gering gehalten.

Im weiteren Verlauf dieses Abschnitts werden nun Meßergebnisse für die blickpunktabhängige Phase des einfachen Zwei-Phasen-Verfahrens angegeben, um die Effizienz des adaptiven Lastverteilungsstrategie zu demonstrieren. Die Messungen wurden auf dem CC-System der Firma Parsystec, welcher im *Paderborn Center for Parallel Computing* (PC²) installiert ist, durchgeführt. Als Eingabe für das modifizierte parallele Raytracing-Verfahren wurden Modelle verwendet, für die zuvor mit Hilfe des Radiosity-Verfahrens der diffuse Lichtenergieaustausch simuliert wurde. In Tabelle 5.17 und Tabelle 5.18 werden die Berechnungszeiten für zwei verschiedene Lastverteilungsstrategien gegenübergestellt. Hierbei handelt es sich auf der einen Seite um eine statische Lastverteilungsstrategie, bei welcher der Bildraum vor Berechnungsbeginn statisch in kleine Bildausschnitte unterteilt wird, wobei jedem Prozessor genau ein Bildausschnitt zugeordnet wird. Analoge Berechnungen wurden unter Verwendung der zuvor beschriebenen adaptiven, bedarfsgesteuerten Lastverteilungsstrategie durchgeführt. Für die Meßreihen sind jeweils Bilder mit einer Auflösung von 640 mal 480 Bildpunkten berechnet worden.

Szene Rosenthaler-Hof				
Anzahl der Lichtquellen: 12				
Anzahl der Polygone: 86090				
Worker	statisch (Sek.)	Speed-Up	adaptiv (Sek.)	Speed-Up
1	1768	1,0	1768	1,0
2	934	1,89	886	2,00
6	364	4,86	298	5,93
14	174	10,16	129	13,71
30	92	19,22	62	28,52
38	73	24,22	49	36,08

Tabelle 5.17: Berechnungszeiten und Speed-Up für die parallele Raytracing-Phase auf Basis einer Radiosity-Lösung für die Szene *Rosenthaler-Hof*.

Szene Badezimmer				
Anzahl der Lichtquellen: 16				
Anzahl der Polygone: 49491				
Worker	statisch (Sek.)	Speed-Up	adaptiv (Sek.)	Speed-Up
1	454	1,0	454	1,0
2	298	1,52	231	1,97
6	109	4,17	79	5,75
14	48	9,46	36	12,61
30	24	18,92	18	25,22
38	19	23,89	15	30,27

Tabelle 5.18: Berechnungszeiten und Speed-Up für die parallele Raytracing-Phase auf Basis einer Radiosity-Lösung für die Szene *Badezimmer*.

Bei komplexen Eingabeszenen werden mit Hilfe des bedarfsgesteuerten Ansatzes für große Prozessoranzahlen wesentlich bessere Ergebnisse erzielt als bei Szenen geringerer Komplexität. Dieses ist darauf zurückzuführen, daß die Lastverteilung durch eine zentrale Instanz realisiert wird, die einem Prozessor auf Anfrage ein neues Arbeitspaket übermittelt. Bei einfachen Szenen bearbeiten die Prozessoren die ihnen zugeordneten Teilaufgaben in sehr kurzer Zeit und stellen somit häufiger Anfragen nach neuen Bildausschnitten an den zentralen Lastverteiler, der in diesem Fall zu einem Flaschenhals wird. Dieser Sachverhalt ist deutlich durch Vergleich der Berechnungszeiten für die beiden Testszenen *Rosenthaler-Hof* (Tabelle 5.17) und *Badezimmer* (Tabelle 5.18) zu erkennen.

Die Überlegenheit der adaptiven Lastverteilungsstrategie gegenüber der statischen Zuordnung wird durch die in den Tabellen angegebenen Speed-Up-Werte verdeutlicht. Für die komplexe Eingabeszene *Rosenthaler-Hof* kann ein annähernd linearer Speed-Up erreicht werden. In dem Fall der initialen statischen Zuordnung von Bildausschnitten zu den Worker-Prozessen kann durch variieren der Größen und der Platzierung von zugewiesenen Bildausschnitten die Effizienz der parallelen Berechnungen beeinflusst werden. Bei allen durchgeführten Messungen konnte jedoch für statische Zuweisungsverfahren nie eine Performanz erzielt werden, welche die des dynamischen Verfahrens annähernd erreicht hätte.

Es soll an dieser Stelle nicht verschwiegen werden, daß problematische Aspekte bei der Durchführung des beschriebenen einfachen Zwei-Phasen-Verfahrens existieren. Der größte

Schwachpunkt ist die Speicherung der Szene auf jedem Prozessor des Parallelrechners. Hierdurch ist die Größe der Szenen beschränkt, die während der zweiten Bildgenerierungsphase verwendet werden können. Da im Anschluß an die Durchführung der ersten Berechnungsphase (Radiosity-Berechnungen) aufgrund der adaptiven Verfeinerung der Szenenbeschreibung selbst sehr einfache Szenen eine große Komplexität aufweisen können, wird eine datenparallele zweite Phase benötigt. Bei der hier beschriebenen einfachen Zwei-Phasen-Methode werden Radiosity-Berechnungen und Raytracing durch zwei separate parallele Programme ausgeführt. Die Übergabe der Radiosity-Werte an die zweite Berechnungsphase erfolgt hierbei mit Hilfe einer Datei, die am Ende der Radiosity-Berechnungen erzeugt wird und anschließend in einer Konvertierungsphase in das Dateiformat des Raytracing-Verfahrens überführt wird. Die resultierende Datei muß vor der abschließenden Bildgenerierungsphase gelesen und an alle Prozessoren übermittelt werden. Somit entsteht ein erheblicher Mehraufwand durch den notwendigen Datenaustausch.

Eine Lösung für die beschriebenen problematischen Aspekte ist die Realisierung einer integrierten Zwei-Phasen-Methode, bei welcher beide Berechnungsphasen auf einer gemeinsamen verteilten Datenstruktur basieren. In diesem Fall kann einerseits eine datenparallele Bildgenerierung auf Basis von Raytracing durchgeführt werden, andererseits entfällt die Notwendigkeit eines expliziten Datenaustausches zwischen zwei separaten Programmen über Dateien und der damit verbundene Mehraufwand. Eine solches integriertes paralleles Verfahren wird im Verlauf von Abschnitt 5.3 beschrieben. Hierbei handelt es sich um eine vollständige Neuimplementierung, wobei sich das parallele Radiosity-Verfahren sowohl aus Sicht der eigentlichen Radiosity-Methode als auch aus Sicht der Parallelisierungsstrategie grundsätzlich von dem bisher beschriebenen Verfahren unterscheidet.

5.3 Paralleles Mehrphasen-Verfahren zur Simulation komplexer Beleuchtungseffekte

In diesem Abschnitt wird ein neues paralleles Verfahren beschrieben, das die Simulation komplizierter globaler Beleuchtungseffekte zu Zwecken der photorealistischen Bildgenerierung ermöglicht. Hierbei werden die Vorteile verschiedener globaler Beleuchtungsverfahren kombiniert, um sowohl interaktive Begehungen in realistisch beleuchteten Umgebungen zu ermöglichen, als auch die Generierung von photorealistischen Einzelbildern zu unterstützen.

Die Grundlage des hier beschriebenen parallelen Mehrphasen-Verfahrens zur globalen Beleuchtungssimulation bildet erneut eine Progressive-Refinement-Methode [CCWG88]. Somit ist es weiterhin möglich, Zwischenergebnisse zu jedem Zeitpunkt des Simulationsprozesses zur Generierung von Einzelbildern oder zur Durchführung von interaktiven Begehungen von korrekt beleuchteten virtuellen Umgebungen zu verwenden. Im Vergleich zu der in Abschnitt 5.2 verwendeten Radiosity-Methode wurden wesentliche Veränderungen vorgenommen. Ziel dieser Veränderungen ist es, die Qualität der berechneten Radiosity-Lösungen durch Verwendung von verbesserten Methoden zur Formfaktorapproximation und zur Bestimmung der relativen Sichtbarkeit zwischen Sample-Punkten zu steigern.

Weiterhin wurde eine Erweiterung der grundlegenden Radiosity-Methode vorgenommen, welche es ermöglicht Beleuchtungseffekte zu simulieren, die durch vergleichbare parallele Radiosity-Methoden bisher nicht berücksichtigt wurden. Hierbei handelt es sich um Beleuchtungseffekte, die durch einen Lichttransport von spiegelnd reflektierenden oder transparenten Flächen auf diffus reflektierende Flächen entstehen (z.B. LSDE-Lichttransportpfade). Mit den klassischen Radiosity- und Raytracing-Methoden läßt sich zum Beispiel die indirekte

Beleuchtung einer Umgebung über einen Spiegel, der direkt von einer Lichtquelle angestrahlt wird und die Energie spiegelnd in die Umgebung reflektiert, nicht realisieren. Die Simulation dieser Transportpfade kann unabhängig von dem Standpunkt eines Betrachters innerhalb der Szene erfolgen (siehe 3.5.2). Eine spezielle Klasse dieser Beleuchtungsphänomene wird als *Caustics* (Lichtverstärkungen oder Lichtbündelungen) bezeichnet und ist in realen Umgebungen häufig anzutreffen. Die Berücksichtigung dieser Effekte ermöglicht eine wesentliche Erhöhung des Realitätsgrades von computergenerierten Bildern.

Um den Realitätsgrad der erzeugten Bilder noch weiter zu steigern, werden in einer zweiten Berechnungsphase Spiegelungen von Lichtenergie berücksichtigt, welche daß Licht in Richtung Betrachter reflektieren. Dieses geschieht nach dem Prinzip des Backward-Eye-Raytracing. Aufgrund der direkten Integration dieser zweiten Berechnungsphase in das parallele Radiosity-Verfahren entfallen die in 5.2.8 aufgeführten problematischen Aspekte der einfachen Zwei-Phasen-Methode.

Das in diesem Abschnitt beschriebene parallele Mehrphasen-Verfahren zur Simulation komplexer globaler Beleuchtungseffekte in statischen und dynamisch veränderbaren Umgebungen ist eine vollständige Neuentwicklung. Verbesserungen im Vergleich zu der in Abschnitt 5.2 eingeführten parallelen Methode wurden sowohl hinsichtlich der Qualität der Beleuchtungssimulation als auch der Effizienz der parallelen Berechnungen erzielt.

5.3.1 Das grundlegende sequentielle Progressive-Refinement-Verfahren

In diesem Abschnitt sollen die Grundlagen für das Verständnis eines neuen parallelen Verfahrens zur integrierten Simulation komplexer Beleuchtungseffekte geschaffen werden. Zunächst werden die wesentlichen Aspekte der grundlegenden Radiosity-Methode beschrieben, bevor das eigentliche parallele Verfahren im Detail vorgestellt wird. Hierbei werden vorerst nur Berechnungen in statischen Szenen betrachtet. Die notwendigen Erweiterungen für dynamisch veränderbaren Szenen werden zu einem späteren Zeitpunkt erläutert.

5.3.1.1 Verwendete Methoden zur Approximation von Formfaktoren

Die verwendete Methode zur Approximation von Formfaktoren hat sowohl einen wesentlichen Einfluß auf die Genauigkeit einer berechneten Radiosity-Lösung als auch auf die Geschwindigkeit der globalen Beleuchtungssimulation. Die Verwendung von ungenauen Formfaktoren ist eine der häufigsten Ursachen für deutlich sichtbare Artefakte bei der Visualisierung der Radiosity-Lösung (siehe 3.1.2). Im Rahmen des hier vorgestellten parallelen Verfahrens werden zwei verschiedene Methoden zur Berechnung von Formfaktoren unterstützt. Die Auswahl der zu verwendenden Methode erfolgt parametergesteuert.

Bei der ersten Methode handelt es sich um das von Wallace u.a. eingeführte Verfahren [WKEH89] (siehe Abschnitt 3.1.2.5). Dieses Verfahren ermöglicht eine exakte Approximation von Formfaktoren zwischen den Eckpunkten der Patches eines Radiosity-Netzes und einem beliebigen Shooting-Patch (oder *Sender-Patch*). Somit können die Radiosity-Werte der Eckpunkte direkt berechnet werden, wodurch die bisher übliche fehlerbehaftete Interpolation der Eckpunkt-Radiosities aus angrenzenden Patch-Radiosities nicht mehr notwendig ist (vgl. 3.4). Die berechneten Eckpunkt-Radiosities können direkt zur Bildgenerierung mittels Gouraud-Schattierung verwendet werden. Eine wesentliche Rolle bei der Formfaktorberechnung nimmt die Bestimmung der relativen Sichtbarkeit zwischen einem Empfänger-Element (Patch oder Punkt) und dem Sender von Lichtenergie ein. Bei der Methode von Wallace u.a. wird eine Menge von zufällig gewählten Sample-Punkten auf dem Sender-Patch festgelegt. Von

jedem Eckpunkt des Radiosity-Netzes wird bei der Formfaktorberechnung die relative Sichtbarkeit des Sender-Patches bestimmt, indem Strahlen in Richtung der Sample-Punkte auf dem Sender-Patch generiert werden. Jeder dieser Strahlen wird auf Schnittpunkte mit den Objekten der Szene überprüft. Durch diese Vorgehensweise wird die prozentuale Sichtbarkeit zwischen Sender-Patch und einem Empfänger-Eckpunkt approximiert. Die Genauigkeit der Approximation kann durch Steigerung der Anzahl der Sample-Punkte und durch eine gute Verteilung der Sample-Punkte auf dem Sender-Patch deutlich gesteigert werden. Die Energiemenge, die von einem Sender-Patch auf ein Empfänger-Punkt übergeht, wird auf Basis der in 3.1.2.5 angegebenen Formel für Patch-zu-Punkt-Formfaktoren ermittelt.

Die zweite der unterstützten Methoden bestimmt Formfaktoren zwischen Patches der Umgebung, basierend auf einer Monte-Carlo-Integration des doppelten Flächenintegrals eines Patch-zu-Patch Formfaktors ([CW93] und 3.1.2.6). Zur Sichtbarkeitsbestimmung wird eine Menge von zufällig verteilten Sample-Punkten sowohl auf dem Empfänger- als auch auf dem Sender-Patch erzeugt. Von jedem Sample-Punkt des Empfänger-Patches wird ein Strahl in Richtung eines Sample-Punktes auf dem Sender-Patch generiert. Für jeden dieser Strahlen wird überprüft, ob auf der Strecke zwischen den entsprechenden Sample-Punkten ein anderes Objekt der Szene getroffen. Für jedes Paar von Sample-Punkten, dessen relative Sichtbarkeit nicht beeinträchtigt ist, wird ein Punkt-zu-Punkt Formfaktor berechnet. Die bestimmten Punkt-zu-Punkt-Formfaktoren werden verwendet, um gemäß 3.1.2.6 (Algorithmus 3.1) einen Patch-zu-Patch-Formfaktor zu berechnen. Ebenso wie bei der Methode von Wallace u.a. läßt sich die Genauigkeit der Formfaktorapproximation durch Erhöhung der Anzahl der Sample-Punkte (somit der Erhöhung der Anzahl der Sample-Strahlen) und durch eine gute Verteilung der Sample-Punkte auf Empfänger- und Sender-Patch steigern.

Mit der ursprünglichen Methode von Wallace u.a. lassen sich Unstetigkeiten der Beleuchtungsfunktion im Inneren einer Patchfläche nicht erkennen, da Formfaktoren nur an den Eckpunkten eines Patches bestimmt werden. Wenn somit eine Schattengrenze im Inneren eines Patches verläuft, ohne daß einer der Eckpunkte des Patches innerhalb des Schattens liegt, kann diese Situation nicht erkannt werden (siehe Abb. 5.34).

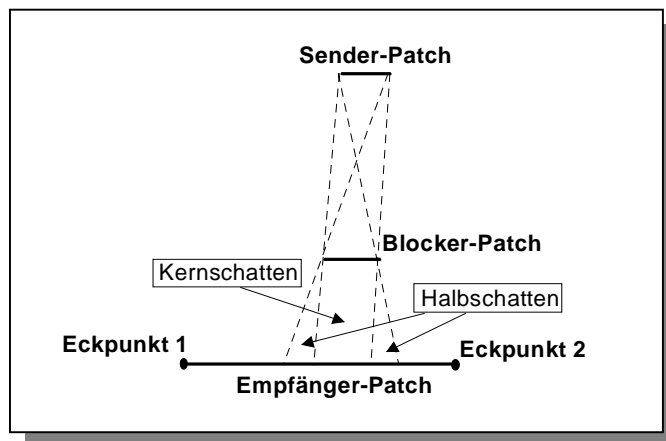


Abb. 5.34: Schattenwurf eines Blocker-Patches, der vollständig innerhalb des Empfänger-Patches verläuft.

Dieses kann in bestimmten Situationen zu deutlich sichtbaren Fehlern führen. Bei der Monte-Carlo-Integration wird das doppelte Flächenintegral des Patch-zu-Patch Formfaktors erheblich genauer approximiert als bei den übrigen Methoden zur Berechnung von Formfaktoren.

Die Variation der relativen Sichtbarkeit zwischen verschiedenen Punkten auf dem Sender- und dem Empfänger-Patch wird hierbei in Betracht gezogen. Somit werden Unstetigkeiten der Beleuchtungsfunktion eines Empfänger-Patches nicht nur entlang der Kanten des Patches, sondern auf der gesamten Fläche des Patches erkannt.

5.3.1.2 Generieren von Sample-Punkten

Wesentliche Auswirkungen auf die Qualität der berechneten Formfaktoren hat die Genauigkeit, mit der die relative Sichtbarkeit zwischen Empfänger und Sender von Energie approximiert wird. Die Genauigkeit steht in direktem Zusammenhang mit der Anzahl und der Position von Sample-Punkten auf den entsprechenden Patches.

Bei dem hier beschriebenen Verfahren kann die Anzahl m der zu verwendenden Sample-Punkten durch den Anwender festgelegt werden. Bei der Platzierung der Sample-Punkte auf einem Oberflächenelement wird zwischen statisch und zufällig platzierten Sample-Punkten unterschieden. Wenn lediglich ein Sample-Punkt verwendet werden soll, wird dieser grundsätzlich im Schwerpunkt der Dreiecks- oder Vierecksfläche platziert. Hierbei ist zu beachten, daß bei der Monte-Carlo-Integration zur Formfaktorberechnung aufgrund des Substrukturierungskriteriums in diesem Fall keine adaptive Verfeinerung von Patches durchgeführt werden kann (siehe 5.3.1.4).

Wenn die Anzahl m der Sample-Punkte einen Wert größer als Eins hat, wird für jedes Patch eine bestimmte Anzahl von Sample-Punkten derart statisch platziert, daß eine gute Abdeckung der Fläche durch Sample-Punkte gewährleistet ist. Auf einer Dreiecksfläche werden vier Sample-Punkte statisch platziert, während auf einer Vierecksfläche fünf Sample-Punkte statisch platziert werden (siehe Abb. 5.35).

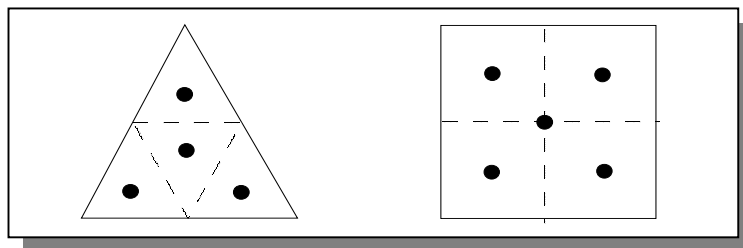


Abb. 5.35: Platzierung von Sample-Punkten in den Schwerpunkten von kleineren Flächenelementen.

Wenn l die Anzahl der statisch platzierten Sample-Punkte auf der Fläche eines Patches ist, werden die übrigen $m-l$ Punkte zufällig auf dieser Fläche verteilt. Diese Vorgehensweise hat zu einer deutlichen Verbesserung der Approximation von Schattengrenzen geführt, da gerade bei einer geringen Anzahl von Sample-Punkten die Fläche durch zufällig verteilte Punkte nicht ausreichend abgedeckt wird. Weiterhin hat es sich als günstig erwiesen, bei der Berechnung von Formfaktoren zwischen einem Patch einer primären Lichtquelle und einem Empfänger-Element zusätzliche Sample-Punkte statisch auf den Eckpunkten der Sender-Patches zu platzieren, um die komplette Ausdehnung der Flächen bei der Sichtbarkeitsberechnung zu erfassen.

Es ist zu beachten, daß bei dem Monte-Carlo-Verfahren zur Approximation der Formfaktoren sowohl auf dem Empfänger- als auch auf dem Sender-Patch m Sample-Punkte platziert und somit m relative Sichtbarkeitsbeziehungen ausgewertet werden. Wenn in der Szene n Patches

vorhanden sind, müssen pro Iteration nm Sichtbarkeitsbeziehungen ausgewertet werden. Aufgrund der adaptiven Verfeinerung vergrößert sich diese Zahl jedoch exponentiell, da für jedes neu erzeugte Patch wiederum m Sichtbarkeitsbeziehungen ausgewertet werden müssen. Somit ist die Anzahl der tatsächlich ausgewerteten Sichtbarkeitsbeziehung sehr stark von den energetischen und geometrischen Gegebenheiten innerhalb einer Szene abhängig (siehe 5.3.1.4).

Bei der Methode von Wallace u.a. werden pro Iteration lediglich auf dem Sender-Patch m Sample-Punkte erzeugt, die von jedem Eckpunkt des Radiosity-Netzes abgetastet werden. Da jedes Patch maximal vier Eckpunkte besitzt, ist die Anzahl der Eckpunkte in einer Szene bestehend aus n Patches durch $4n$ beschränkt. Diese Zahl ist bei konkreten Szenen jedoch deutlich geringer, da diese in den meisten Fällen Dreiecksflächen enthalten. Weiterhin werden gemeinsame Punkte benachbarter Patches nur einmal in der Datenstruktur gespeichert, und der entsprechende Energietransfer von einem Sender-Patch an diese Eckpunkte wird jeweils nur einmal berechnet. Die Anzahl der Ausgewerteten Sichtbarkeitsbeziehungen ist somit pro Iteration durch $4nm$ beschränkt, wenn keine adaptive Substrukturierung durchgeführt wird. Da bei jeder adaptiven Verfeinerung eines Patches neue Punkte (maximal 5) erzeugt werden, erhöht sich die Anzahl der tatsächlich pro Iteration auszuwertenden Sichtbarkeitsbeziehungen ebenfalls exponentiell in Abhängigkeit der Substrukturierungstiefe eines Patches (siehe 5.3.2.1).

Aufgrund der großen Anzahl von auszuwertenden Sichtbarkeitsbeziehungen pro Iteration ist die zur Auswertung benötigte Zeit entscheidend für die Berechnungszeit einer vollständigen Radiosity-Lösung.

5.3.1.3 Überprüfen der relativen Sichtbarkeit von Sample-Punkten

Der zeitaufwendigste Teil bei der Bestimmung von Formfaktoren auf Basis der in 5.3.1.2 beschriebenen Methoden ist die Auswertung der relativen Sichtbarkeit zwischen zwei Punkten $S_1, S_2 \in \mathbb{R}^3$. Hierbei wird ein Strahl ausgehend von S_1 in Richtung S_2 generiert mit,

$$X = S_1 + t(S_2 - S_1), t \geq 0. \quad (5.12)$$

S_1 und S_2 sind gegenseitig sichtbar, wenn der Strahl auf der Verbindungsstrecke zwischen den beiden Punkten kein Objekt der Szene schneidet. Um dieses zu überprüfen, müssen Schnittpunkte des Strahles mit den Objekten der Szene bestimmt werden. Für einen gefundenen Schnittpunkt wird überprüft, ob dieser eine geringere Distanz zu S_1 hat als S_2 . In diesem Fall sind S_1 und S_2 nicht gegenseitig sichtbar. Es ist zu beachten, daß nur die Information benötigt wird, ob ein verdeckendes Objekt existiert oder nicht. Es wird nicht das Objekt mit der geringsten Distanz zu S_1 gesucht, welches die Sichtbarkeit zwischen S_1 und S_2 einschränkt.

Die grundlegende Problemstellung entspricht somit dem Verfolgen eines Schattenfühlers (*Shadow-Ray*) in einem rekursiven Raytracing-Verfahren. Aus diesem Grund können die beim Raytracing verwendeten Beschleunigungstechniken auch im Kontext der beschriebenen Methoden zur Bestimmung von Formfaktoren eingesetzt werden.

Eine globale Beleuchtungssimulation auf Basis der Radiosity-Methode bedingt die Beschreibung der Objekte einer Szene durch Mengen von einfachen planaren Polygonen (Patches) mit drei oder vier Eckpunkten (Patches), welche während einer initialen Netzgenerierungs-Phase erzeugt werden (siehe Abschnitt 3.3). Um einen Schnittpunkt zwischen einem bestimmten Objekt und einem Strahl zu finden, muß somit jedes Patch des Objektes auf einen Schnittpunkt mit dem Strahl überprüft werden. Hierbei wird zunächst der Schnittpunkt des Strahles

mit der Ebene im dreidimensionalen Raum bestimmt, in welcher das Patch liegt. Wenn ein solcher Schnittpunkt existiert, erfolgt eine Überprüfung, ob der Punkt innerhalb des durch die Eckpunkte des Patches festgelegten Polygonzug liegt (*Punkt-in-Polygon-Test* [Glass89]). Diese Vorgehensweise bedingt aufgrund der Unterteilung von großen planaren Flächen in deutlich kleinere Flächen (Patches) eine große Zahl von unnötigen Schnittpunkttests, da für jedes Patch dieser Fläche eine Schnittpunktberechnung durchgeführt werden muß. Eine effiziente Vorgehensweise besteht darin, einen Strahl auf einen Schnittpunkt mit Oberflächenbeschreibungen durchzuführen, welche nicht in Patches unterteilt sind. Solche Oberflächenbeschreibungen lassen sich durch die Vereinigung der Patches eines Objektes zu planaren Polygone maximaler Größe erzeugen.

Dieser Sachverhalt ist in Abb. 5.36 dargestellt. Um die Objektoberfläche auf einen Schnittpunkt mit dem Strahl ausgehend von S_1 in Richtung S_2 zu überprüfen, muß zunächst in beiden Fällen der Schnittpunkt des Strahles mit der Ebene gefunden werden. Im Anschluß daran müssen im links angegebenen Fall fünf Punkt-in-Polygon-Tests ausgeführt werden, bevor feststeht, daß der Strahl die Objektoberfläche trifft (unter der Voraussetzung das Tests in Reihenfolge der Patchnummerierung durchgeführt werden). Wenn der Strahl die Ebene der Objektoberfläche schneidet, die Oberfläche selbst jedoch verfehlt, werden neun negative Tests durchgeführt, bevor die Aussage getroffen werden kann, daß der Strahl die Objektoberfläche nicht schneidet. Für den rechts skizzierten Fall reicht in jedem Fall ein Punkt-in-Polygon-Test aus.

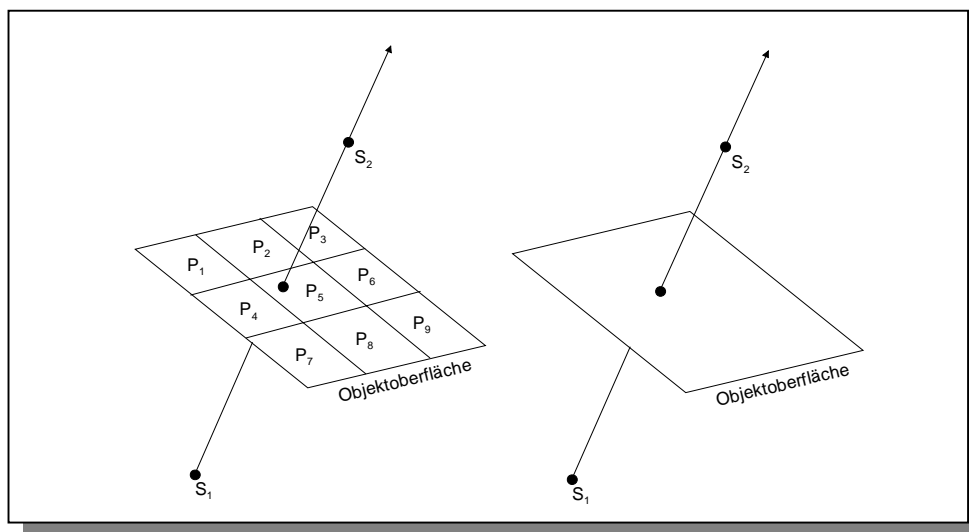


Abb. 5.36: Schnittpunkttest auf Basis von Patches (links) und auf Basis eines maximalen planaren Polygons (rechts).

Die hier beschriebene Radiosity-Methode basiert auf der Verwendung von zwei verschiedenen Repräsentationen der Eingabeszene in Form von Polygonnetzen unterschiedlicher Komplexität. Es wird hierbei eine Trennung zwischen einer komplexeren Datenstruktur zur Speicherung von Beleuchtungsinformationen auf Objektoberflächen der Szene (*Radiosity-Netz*) und einer wesentlich einfacheren Datenstruktur vorgenommen, welche die Geometrie der Eingabeszene in Form von großen planaren Polygonen beschreibt (*Intersection-Netz*). Es ist zu beachten, daß die Repräsentation einer Szene durch das Intersection-Netz und das Radiosity-Netz aus geometrischer Sicht identisch ist (siehe Abb. 5.37).

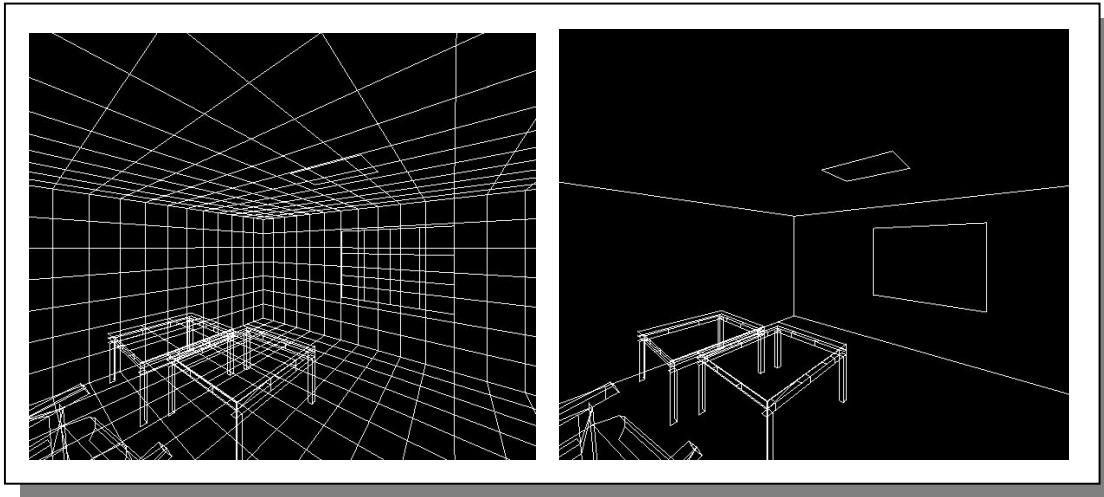


Abb. 5.37: Repräsentation einer Szene durch das *Radiosity-Netz* (links) und das *Intersection-Netz* (rechts).

Das Radiosity-Netz wird zur Approximation der Beleuchtungsfunktionen von Oberflächen verwendet. Da diese über großen Objektflächen sehr stark variieren können, ist eine wesentlich feinere Diskretisierung notwendig, als es für die Bestimmung von relativen Sichtbarkeitsbeziehungen notwendig ist. Zur Überprüfung der relativen Sichtbarkeit zweier beliebiger Sample-Punkten innerhalb der Szene werden Schnittpunkttests auf Basis des Intersection-Netzes durchgeführt. Die Komplexität dieses Netzes bleibt im Verlauf von Radiosity-Lösungen innerhalb von statischen Szenen unverändert, während die Komplexität des Radiosity-Netzes aufgrund von adaptiven Verfeinerungen von Patches deutlich zunehmen kann. Das Intersection-Netz der Eingabeszene wird ebenso wie das Radiosity-Netz während der initialen Netzgenerierungs-Phase bestimmt.

Ein Vergleich der Komplexität des Radiosity- und des Intersection-Netzes von verschiedenen Szenen ist in Tabelle 5.19 zu finden.

Szene	Radiosity-Netz		Intersection-Netz	
	Anzahl der Polygone	Speicherplatz (KB)	Anzahl der Polygone	Speicherplatz (KB)
<i>Büro</i>	1289	353	301	35
<i>Uni-Foyer</i>	18361	4152	661	85
<i>Wohnzimmer</i>	58993	15059	21224	1890
<i>Townhouse</i>	62725	16804	13627	1281
<i>Fürstenallee</i>	93621	23138	21531	1896

Tabelle 5.19: Größenvergleich Radiosity- vs. Intersection-Netz für verschiedene Szenen.

Es ist deutlich zu sehen, daß die Radiosity-Netze der Szenen wesentlich mehr Polygone (Patches) enthalten als die entsprechenden Intersection-Netze. Weiterhin ist zu sehen, daß der zur Speicherung des Radiosity-Netzes benötigte Hauptspeicher um ein Vielfaches größer ist als der für die Intersection-Netze benötigte Speicherplatz. Dieses ist sowohl auf die geringere Anzahl von Polygonen in einem Intersection-Netz zurückzuführen, als auch auf die Tatsache, daß in der Datenstruktur eines Intersection-Netzes keine Materialeigenschaften gespeichert werden, sondern lediglich die benötigten geometrischen Informationen. Es ist zu beachten, daß die Anzahl von Polygonen in Intersection-Netz sehr stark von den Formen der Objekte innerhalb der Szenen abhängt. Große Flächen lassen sich z.B. sehr gut durch einige wenige

Polygone approximieren. Im Gegensatz dazu können bei stark gekrümmten Objektoberflächen, die zumeist durch eine große Anzahl von kleinen Polygonen modelliert sind, keine Polygone zusammengefaßt werden. Ein Beispiel für eine solche Szene ist das *Wohnzimmer* in Tabelle 5.19. Diese Szene enthält sehr viele gekrümmte Oberflächen. Aus diesem Grund enthält das Intersection-Netz noch annähernd 36% der Anzahl der Polygone, die in dem entsprechenden Radiosity-Netz enthalten sind. Dieses Verhältnis ist bei den übrigen Szenen deutlich besser.

Ein naiver Ansatz zur Überprüfung der relativen Sichtbarkeit von zwei Sample-Punkten besteht darin, jede Oberfläche der Umgebungsbeschreibung auf einen Schnittpunkt mit dem entsprechenden Strahl zu überprüfen. Diese Vorgehensweise verursacht einen erheblichen Berechnungsaufwand, da viele Schnittpunkttests mit Oberflächen durchgeführt werden, die aufgrund ihrer geometrischen Position im Raum keinen Einfluß auf die gegenseitige Sichtbarkeit der beiden Punkte haben können. Im Kontext der Raytracing-Methode gibt es sehr effiziente Techniken, welche es durch Ausnutzung von bestimmten Kohärenzeigenschaften die Anzahl der zu überprüfenden Objekte bei einem Schattentest deutlich einschränken, ohne das es hierbei zu Fehlern kommt [Glass89]. Diese Techniken lassen sich größtenteils auf die hier betrachtete Problemstellung der Bestimmung der relativen Sichtbarkeit zweier Punkte im dreidimensionalen Raum direkt übertragen. Es wird hierdurch eine mitunter dramatische Reduktion der Zeiten zur Bestimmung von Formfaktoren und somit der gesamten Berechnungszeit einer Radiosity-Lösung erreicht. An dieser Stelle soll ein kurzer Überblick über die verwendeten Beschleunigungstechniken gegeben werden.

Shadow-Cache

Eine der einfachsten Techniken zur Reduktion der Anzahl von Schnittpunkttests ist das Shadow-Caching. Hierbei wird ein Verweis auf die Oberfläche gespeichert, für die zuletzt ein Schnittpunkt mit einem Sample-Strahl gefunden wurde. Jeder neu generierte Strahl wird zunächst auf einen Schnittpunkt mit dieser Oberfläche überprüft. Ein Shadow-Cache macht sich die sogenannten Strahlen-Kohärenz (*Ray-Coherence*) zunutze. Diese besagt, daß Strahlen mit einer ähnlichen Strahlenbahn im Raum auch mit sehr großer Wahrscheinlichkeit die gleichen Oberflächen einer Szene treffen.

Begrenzungskörper (Bounding-Boxes)

Da Objekte oftmals aus einer großen Anzahl von einzelnen, sehr kleinen Oberflächen bestehen, ist es eine zeitintensive Vorgehensweise, jede einzelne Fläche des Objektes auf einen Schnittpunkt mit dem Teststrahl zu überprüfen. Vor allen Dingen wenn der Strahl nicht auf das zu überprüfende Objekt trifft, werden sehr viele überflüssige Schnittpunktberechnungen durchgeführt. Die Verwendung von einfachen Begrenzungskörpern wie Kugeln oder Quader bietet die Möglichkeit einer schnellen Überprüfung, ob ein Strahl auf ein Objekt treffen kann oder nicht. Hierzu wird für jedes Objekt der Szene ein Begrenzungskörper bestimmt, der die Oberflächen des Objektes vollständig einschließt [RW80]. In dem hier beschriebenen Verfahren werden Quader als Begrenzungskörper verwendet. Während eines Preprocessing-Schrittes werden die maximalen Koordinaten der Objekte bestimmt. Mit Hilfe dieser Koordinaten können jeweils sechs Begrenzungsebenen des Objektes festgelegt werden, wobei die Ebenen parallel zu den Grundebenen des euklidischen Weltkoordinatensystems (xy -, xz - und yz -Ebene) verlaufen. Ein Schnittpunkttest eines Strahles mit diesen einfachen Begrenzungskörpern ist sehr effizient durchführbar (siehe [Glass89]). Wenn eine Schnittpunktberechnung zwischen einem Strahl und einem Objekt durchgeführt werden soll, wird zunächst überprüft, ob der Strahl den Begrenzungskörper schneidet. Ist dieses nicht der Fall, kann der Strahl auf keinen

Fall das eingeschlossene Objekt treffen. Somit besteht keine Notwendigkeit, Schnittpunktberechnungen mit den Oberflächen des Objektes durchzuführen. Im Falle eines positiven Tests müssen die einzelnen Oberflächen des Objektes auf einen Schnittpunkt mit dem Strahl überprüft werden, um eine endgültige Aussage treffen zu können. Es ist zu beachten, daß auch hierarchische Begrenzungskörper verwendet werden können, um die Anzahl der zu überprüfenden Oberflächen eines Objektes weiter einzuschränken, wenn ein Strahl das Begrenzungskörper trifft.

Octree-Methode

Eine Vielzahl von Beschleunigungstechniken basiert auf einer räumlichen Unterteilung der Szene in Teilregionen (*Spatial Subdivision*). Um eine Reduktion der Anzahl von Schnittpunkttests zu erzielen, werden Datenstrukturen aufgebaut, die eine räumliche Partitionierung der Umgebung in Teilregionen (*Voxel*) definieren. Während der Schnittpunktberechnung werden mit Hilfe dieser Datenstrukturen genau die Voxel identifiziert, durch welche die Strahlen verlaufen. Nur Objekte innerhalb dieser Voxel können durch den Strahl getroffen werden. Die bekanntesten Methoden zur Reduktion der Anzahl von Schnittpunkttests durch Raumunterteilung sind die *BSP-Trees* [Kap85] und die *Octrees* [Glass84], welche auf einer ungleichmäßigen räumlichen Unterteilung der Szene beruhen (*Nonuniform Spatial Subdivision*), sowie die erstmals von Fujimoto u.a. [FPI86] eingeführten Methoden auf Basis von gleichmäßigen räumlichen Unterteilungen (*Uniform Spatial Subdivision*).

Das hier beschriebene Radiosity-Verfahren nutzt die Octree-Methode von Glassner [Glass84] zur Beschleunigung von Schnittpunkttests. Die Wahl ist auf diese Methode gefallen, da bei dem Aufbau des Octrees aufgrund der ungleichmäßigen Raumunterteilung die geometrischen Eigenarten verschiedener Szenen besonders gut berücksichtigt werden, indem die Raumunterteilung in Regionen großer Komplexität besonders fein vorgenommen wird. Es ist zu beachten, daß bei größeren Octree-Tiefen deutlich mehr Speicher benötigt wird und durch die große Anzahl von erzeugten Voxeln eine Octree-Traversierung zu einer teuren Operation wird. Andererseits wird jedoch die Anzahl der notwendigen Schnittpunkttests bei einer feinen Raumunterteilung erheblich reduziert. Weiterhin garantiert ein von Glassner eingeführter, auf Hashing-Tabellen basierender Zugriff auf die Voxel-Daten des Octrees, eine schnelle Traversierung der Octree-Struktur entlang einer gegebenen Strahlenbahn.

Der Octree wird auf Basis des Intersection-Netzes aufgebaut. Für jedes Voxel wird wahlweise eine Liste von Verweisen auf die vollständigen Objekte oder auf einzelne Objektoberflächen bestimmt, die ganz oder teilweise innerhalb dieses Voxels liegen. Die Listeneinträge repräsentieren Kandidaten für einen Schnittpunkt mit jedem Strahl, der durch das entsprechende Voxel verläuft. Der Aufbau eines Octrees für eine gegebene Szene und die damit verbundene Bestimmung von Kandidatenlisten für einzelne Voxel erfolgt in einer Top-Down Vorgehensweise. Hierbei wird ein größeres quaderförmiges Raumvolumen (Voxel) in acht kleinere Voxel unterteilt. Auf Basis der Kandidatenliste des größeren Voxels werden die Kandidatenlisten der kleineren Voxel bestimmt. Die Unterteilung wird rekursiv solange vorgenommen, bis bestimmte Kriterien zum Abbruch der Rekursion erfüllt sind. Dies Kriterien legen die maximale Länge einer Kandidatenliste eines Voxels sowie die maximal zulässige Rekursionstiefe fest. Sobald die Octree-Datenstruktur in einem Preprocessing-Schritt aufgebaut wurde, kann sie dazu verwendet werden, schnell die Objekte bzw. Objektoberflächen zu identifizieren, welche in der Nähe einer bestimmten Strahlenbahn angeordnet sind und somit potentielle Kandidaten für einen Schnittpunkt sind. Hierzu muß ein schneller Zugriff auf die Voxel innerhalb des Octrees realisiert werden, die von einem Strahl geschnitten oder berührt werden. Dieses geschieht auf Basis von zwei grundlegenden Funktionen, die in einer Schleife solange ausge-

führt werden, bis der aktuelle Strahl den durch die Voxel des Octrees eingeschlossenen Raum verläßt. Mit Hilfe der ersten Funktion wird das Voxel identifiziert, welches einen gegebenen Punkt im dreidimensionalen Raum enthält. Die zweite Funktion berechnet einen Punkt auf der Strahlenbahn, der garantiert in dem nächsten direkt angrenzenden Voxel liegt. Bei der Methode von Glassner wird der Octree nicht wie üblich direkt als Baumstruktur mit Hilfe von Zeigern gespeichert, sondern mit Hilfe eines sehr effizienten Hashing-Verfahrens. Diese Art der Speicherung ermöglicht einen schnellen Zugriff auf einzelne Voxel des Octrees und somit auf die entsprechenden Kandidatenlisten.

Für einen Strahl, der durch einen Startpunkt und eine Richtung vollständig beschrieben ist, wird zunächst der Oktant des Wurzelknotens bestimmt, in welchem der Startpunkt des Strahls liegt. Für diesen Oktanten wird rekursiv das Voxel bestimmt, welches auf Blattebene des Octrees liegt und den entsprechenden Startknoten enthält. Der Strahl wird auf einen Schnittpunkt mit allen in der Kandidatenliste des Voxels enthaltenen Elemente überprüft. Wird ein Schnittpunkt gefunden, kann der Schnittpunkttest abgebrochen werden. Andernfalls wird ein Punkt auf der Strahlenbahn bestimmt, der garantiert in dem nächsten benachbarten Voxel auf Blattebene des Octrees liegt (siehe hierzu [Glass84]). Für diesen Punkt werden die Berechnungen analog zu den Berechnungen für den Startpunkt des Strahles fortgesetzt. Der Octree-Durchlauf für einen Strahl terminiert, sobald ein Schnittpunkt mit einem Kandidaten gefunden wurde, oder der Strahl den durch den Octree eingeschlossenen Raum verläßt.

In dem in Abb. 5.38 angegebenen Beispiel wird verdeutlicht, daß während eines Octree-Durchlaufes für einen gegebenen Strahl nur ein geringer Teil der Objekte einer Szene auf einen Schnittpunkt überprüft werden muß.

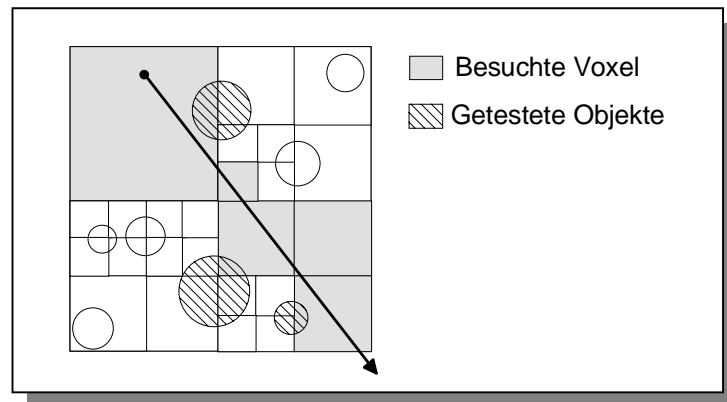


Abb. 5.38: Besuch von bestimmten Voxel während des Octree-Durchlaufes für einen gegebenen Strahl.

Die Begrenzungskörper für Objekte des Intersection-Netzes lassen sich direkt mit der Octree-Methode kombinieren, um eine weitere Reduktion der Anzahl von Schnittpunkttests zu erzielen. Dieses ist insbesondere dann von Vorteil, wenn der Octree nicht auf Objektbasis, sondern auf Oberflächenbasis aufgebaut wurde. D.h. in den Voxel des Octrees sind Verweise auf Objektoberflächen gespeichert, die ganz oder teilweise innerhalb des eingeschlossenen Raumbereiches liegen. Wenn nun während der Octree-Traversierung die Oberflächen in einem Voxel auf einen Schnittpunkt mit einem Strahl überprüft werden, wird für jede Fläche zunächst getestet, ob der Begrenzungskörper des entsprechenden Objektes von dem Strahl getroffen wird. Sollte dieses nicht der Fall sein, wird für den Begrenzungskörper vermerkt, daß dieses Objekt nicht von dem Teststrahl getroffen wird und somit auch keine der Objekto-

berflächen getroffen werden kann. Bei der weiteren Traversierung des Octrees wird für keine weitere Obflächen dieses Objektes ein Intersection-Test durchgeführt. Wird der Begrenzungskörper jedoch getroffen, müssen alle Oberflächen des entsprechenden Objektes, die in den besuchten Voxeln liegen, auf einen Schnittpunkt überprüft werden. Nachdem die Octree-Traversierung für einen gegebenen Strahl durchgeführt wurde, werden die entsprechenden Flags der Begrenzungskörper auf ihren Ursprungszustand zurückgesetzt, damit die nächste Traversierung mit einem neuen Strahl zu korrekten Ergebnissen führt.

Shaft-Culling

Eine weitere Methode, welche eine deutliche Reduktion der Anzahl von durchzuführenden Schnittpunkttests ermöglicht, ist das Shaft-Culling [HW91]. Die Effizienz dieser Methode beruht auf der Tatsache, daß Sample-Strahlen, die zur Bestimmung der Sichtbarkeit zwischen einem Sender-Patch und einem Empfänger-Patch (bzw. Empfänger-Eckpunkt) des Radiosity-Meshes generiert werden, nur in einem sehr eingeschränkten Raumvolumen der gesamten Szene verlaufen (Strahlen-Kohärenz). Objekte, welche die Sichtbarkeit zwischen den beiden Patches beeinflussen können, müssen innerhalb dieses Raumvolumens liegen (siehe Abb. 5.39).

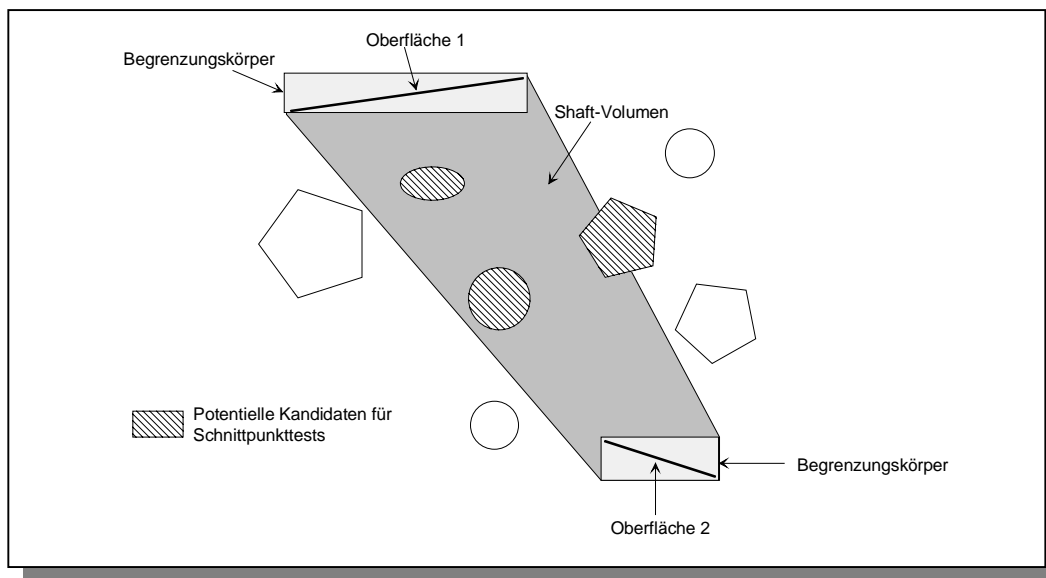


Abb. 5.39: Reduktion der Anzahl von Kandidaten für Schnittpunkttests von Sample-Strahlen zwischen zwei Oberflächen der Szene durch Konstruktion eines Shaft-Volumens.

Die wesentliche Idee des Shaft-Culling besteht nun darin, dieses Raumvolumen möglichst exakt zu bestimmen, um Objekte zu identifizieren, die vollständig innerhalb dieses Bereiches liegen, den Bereich teilweise berühren oder ihn überdecken. Bei der Approximation von Formfaktoren auf Basis der hier verwendeten Methoden (siehe 5.3.1.1) werden die Sample-Strahlen zwischen einem Sender-Patch und einem Empfänger-Patch nur gegen diese Objekte auf einen Schnittpunkt überprüft. Für eine detaillierte Beschreibung dieser Methode sei an dieser Stelle auf die Originalarbeit von Haines und Wallace [HW91] verwiesen.

Während der Radiosity-Berechnungen kann ein Shaft-Volumen wahlweise zwischen einem Sender-Patch und einem Empfänger-Objekt (Menge von Empfänger-Patches) oder zwischen einem Sender-Patch und einem Empfänger-Patch aufgebaut werden. Im ersten Fall wird das

Shaft-Volumen auf Basis eines Begrenzungskörpers für das Shooting-Patch sowie eines Begrenzungskörpers für ein Empfänger-Objekt aufgebaut. Im Anschluß daran wird das konstruierte Shaft-Volumen dazu benutzt, die Formfaktorberechnungen zwischen dem Sender-Patch und sämtlichen Patches der Empfänger-Objekte zu beschleunigen. Im zweiten Fall wird ein Shaft-Volumen auf Basis von Begrenzungskörpern für Sender-Patch und für jeweils ein Empfänger-Patch bestimmt. Durch diese Vorgehensweise wird das Shaft-Volumen erheblich verkleinert und somit auch die Anzahl der potentiellen Schnittpunkt-Kandidaten wesentlich verringert. Im Gegenzug wird jedoch die Anzahl der pro Iteration zu berechnenden Shaft-Volumen vergrößert, wodurch die Kosten der Konstruktion von Shaft-Volumen in bestimmten Fällen deren Nutzen zunichte machen (z.B. wenn zuwenig Sample-Strahlen zur Bestimmung der relativen Sichtbarkeit verwendet werden).

Die Octree-Methode oder das Shaft-Culling können alternativ verwendet werden, um die Schnittpunktberechnungen bei dem hier vorgestellten parallelen Mehrphasen-Verfahren zu beschleunigen. Beide Techniken nutzen sowohl die Begrenzungskörper der Objekte als auch das Shadow-Caching, um eine zusätzliche Beschleunigung zu erzielen.

5.3.1.4 Adaptive Verfeinerung von Patches

Um den Verlauf von Schattengrenzen oder Schwankungen der Beleuchtungsfunktionen von Objektoberflächen genauer wiedergeben zu können, werden die Patches des Radiosity-Netzes im Verlauf der Progressive-Refinement-Berechnungen adaptiv in kleinere Flächenelemente unterteilt (substrukturiert). Hierzu muß mit Hilfe von Substrukturierungstest eine signifikante Schwankung der Lichtverteilung auf einer Fläche erkannt und die betroffenen Patches in jeweils vier kleinere Flächenelemente unterteilt werden (siehe 3.3.2 und Abb. 5.40).

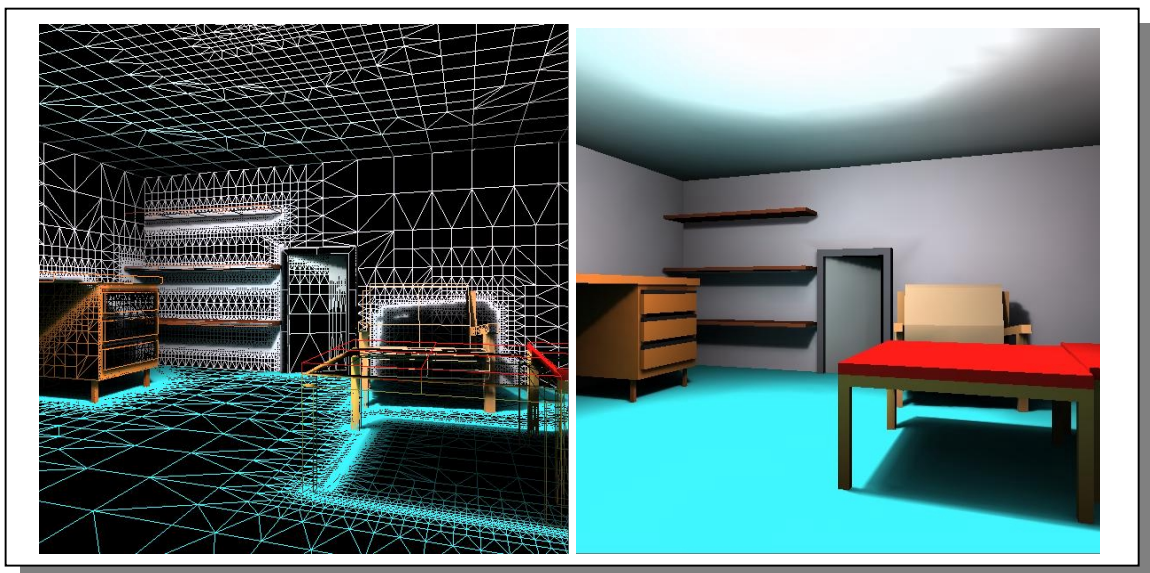


Abb. 5.40: Approximation von Helligkeitsschwankungen durch adaptive Substrukturierung.

Je nach Art des verwendeten Approximationsverfahrens für Formfaktoren werden verschiedene Substrukturierungstests benötigt, um hinreichend gute Resultate zu erzielen. Bei der direkten Berechnung von Eckpunkt-Radiositäten mit der Methode von Wallace u.a. [WKEH89] werden in dem hier beschriebenen Radiosity-Verfahren zwei grundsätzlich verschiedene Substrukturierungstests nacheinander ausgeführt, um die Energieverteilung auf einer Oberfläche möglichst gut durch ein adaptiv verfeinertes Radiosity-Netz anzunähern. Bei dem ersten Test

wird die Differenz der Eckpunkt-Radiosities entlang der Kanten eines Patches überprüft (siehe 3.3.2 – Gleichung 3.27). Überschreitet der Unterschied von Eckpunkt-Radiosities entlang einer der Kanten einen vorgegebenen Grenzwert, wird das entsprechende Patch weiter unterteilt. Im Fall einer zu großen Differenz kann davon ausgegangen werden, daß mindestens ein Eckpunkt des Patches nur eine geringe Energiemenge empfängt, weil dieser beispielsweise verdeckt ist, während ein über eine Kante benachbarter Eckpunkt wesentlich mehr Energie empfängt. Eine solche Situation würde bedeuten, daß eine Schattengrenze durch die Fläche des Patches verlaufen würde, wobei mindestens ein Eckpunkt innerhalb und mindestens ein Eckpunkt außerhalb dieses Schattens liegt.

Der beschriebene Substrukturierungstest kann keine Schattenverläufe erkennen, wenn alle Eckpunkte des Patches außerhalb des Schattens liegen und der Schatten vollständig innerhalb des Patches verläuft. Tritt eine solche Situation auf, fällt der Substrukturierungstest negativ aus und das Patch wird nicht verfeinert. Dieses kann insbesondere bei größeren Patches auftreten. Eine Lösung für dieses Problems besteht darin, im Anschluß an einen negativen ersten Substrukturierungstest einen zusätzlichen Test durchzuführen, der die oben beschriebene problematische Situation erkennt. Die Grundidee dieses zweiten Tests beruht auf der zusätzlichen Berechnung von Punkt-Radiosities für Sample-Punkte im Inneren des Patches. Die direkt berechneten Radiosity-Werte werden mit interpolierten Radiosity-Werten verglichen, die durch bilinearer Interpolation aus den zuvor bestimmten Eckpunkt-Radiosities des Patches an den Sample-Punkten bestimmt werden. Sollte die relative Abweichung zwischen direkt berechnetem und interpoliertem Radiosity-Wert für mindestens einen Sample-Punkt eine vorgegebene Fehlerschranke überschreiten, wird das Patch rekursiv in kleinere Flächenelemente unterteilt. Die Anzahl der zusätzlichen Sample-Punkte pro Patch kann hierbei frei variiert werden, wobei die Wechselwirkungen zwischen Genauigkeit und Berechnungszeit zu berücksichtigen sind. Prinzipiell wird durch diesen zusätzlichen Test überprüft, ob der Fehler bei einer Visualisierung der Radiosity-Lösung mit Hilfe der Gouraud-Schattierung (oder eines anderen interpolativen Schattierungsverfahrens) zu einer fehlerhaften Darstellung führt.

Bei der Berechnung von Formfaktoren mit Hilfe der Monte-Carlo-Integration wird lediglich ein Substrukturierungstest benötigt. Grund hierfür ist, daß diese Methode auf der Berechnung eines Energieaustausches zwischen Patches und nicht zwischen Sender-Patch und Eckpunkten von Patches basiert. Wie in 3.1.2.6 und 5.3.1.1 beschrieben, wird bei dieser Methode ein Formfaktor zwischen Sender- und Empfänger-Patch bestimmt, indem zwischen den Sample-Punkten auf den beiden Patches Punkt-zu-Punkt Formfaktoren berechnet und flächengewichtet addiert werden, um einen Patch-zu-Patch Formfaktor zu approximieren (siehe 3.1.2.6). Die berechneten Punkt-zu-Punkt Formfaktoren (3.1.2.1 – Gleichung 3.6) können zusätzlich dazu verwendet werden, den Energietransfer von Sample-Punkten auf dem Sender-Patch zu den Sample-Punkten auf dem Empfänger-Patch direkt zu berechnen. Diese Energiewerte werden dann in einem Substrukturierungstest als Entscheidungsgrundlage herangezogen. Hierzu werden die berechneten Energiewerte der Sample-Punkte auf dem Empfänger-Patch verglichen. Sollte die maximale relative Abweichung der Energiewerte zweier Sample-Punkte einen vorgegebenen Grenzwert überschreiten, wird daß Empfänger-Patch weiter adaptiv unterteilt. Die Qualität dieses Substrukturierungstests ist in großem Maße abhängig von der Anzahl der verwendeten Sample-Punkte und deren Verteilung. Bei einer ausreichenden Anzahl von Sample-Punkten (i.a. größer gleich 16 Sample-Punkte pro Patch) und deren gleichmäßiger Verteilung werden selbst kleinere Schwankungen der Beleuchtungsfunktion eines Patches zuverlässig erkannt.

Die beschriebenen Substrukturierungstests lassen sich sehr einfach in das Progressive-Refinement-Verfahren integrieren. In jeder Iteration wird ein Energietransfer von einem aus-

gewählten Shooting-Patch zu den übrigen Patches der Szene auf Basis der berechneten Formfaktoren ausgeführt. Wenn durch die zuvor beschriebenen Substrukturierungstests ein zu großer Approximationsfehler für ein Empfänger-Patch (bzw. die Eckpunkte eines Patches) erkannt wird, werden die Formfaktorberechnungen mit anschließenden Substrukturierungstests rekursiv für kleiner Flächenelemente durchgeführt, bis die vorgegebenen Fehlerschranken eingehalten werden oder eine maximale Rekursionstiefe erreicht wird.

Als Datenstruktur zur Speicherung eines adaptiv verfeinerten Radiosity-Netzes wird eine hierarchische Datenstruktur verwendet. Diese Datenstruktur unterscheidet zwischen Objekten, Patches und Punkten. Ein Objekt der Szene ist durch homogene Materialeigenschaften charakterisiert und wird durch eine Menge primitiver Polygone (Patches) beschrieben. Bei diesen primitiven Polygonen handelt es sich um Dreiecke und Vierecke. Für jedes Objekt wird eine Liste der Patches verwaltet. Da zur Approximation von Schattengrenzen Patches im Berechnungsverlauf adaptiv in jeweils vier kleinere Elemente unterteilt (substrukturiert) werden können, ist es notwendig die Substrukturierung für jedes Patch in Form einer baumartigen Datenstruktur zu speichern. Die Verwendung eines Quadtrees hat sich hierfür aus verschiedenen Gründen als ideal erwiesen [HSA91] (siehe Abb. 5.41).

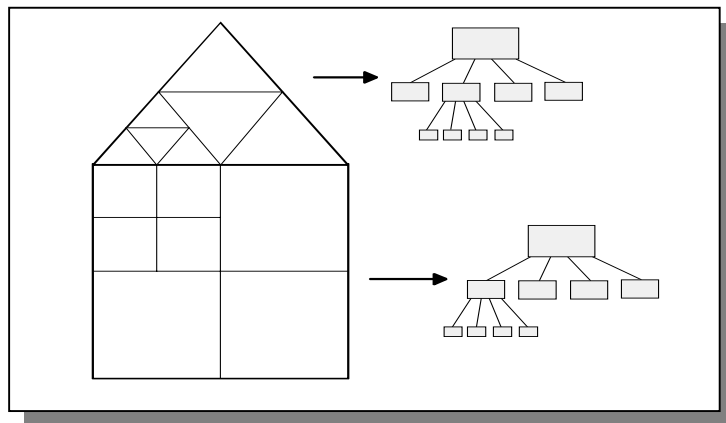


Abb. 5.41: Eine Quadtrees-Datenstruktur speichert die adaptiv erzeugten Patch-Hierarchien.

5.3.1.5 Energietransfer zwischen Patch-Hierarchien

Die hier beschriebene Radiosity-Methode basiert auf einem Progressive-Refinement-Ansatz. In jeder Iteration wird die nicht verteilte spezifische Ausstrahlung eines Sender-Patches (Shooting-Patch) von der obersten Hierarchie-Stufe der entsprechenden Patch-Hierarchie an die übrigen Patches der Szene verteilt. Auf Seiten der Empfänger-Patches werden mit Hilfe der zuvor beschriebenen Substrukturierungstests Empfänger-Elemente auf verschiedenen Ebenen der entsprechenden Patch-Hierarchien festgelegt.

Bei den Berechnungen wird Energie vom Sender-Patch zu einem Empfänger-Element oder Eckpunkt des Empfänger-Elementes nur auf der Hierarchie-Ebene transportiert, auf der durch die Substrukturierungstests in der aktuellen Iteration kein weiterer Bedarf an einer Flächenverfeinerung erkannt wird. Ein Energietransport auf einer gröberen Hierarchiestufe würde zu einem größeren Approximationsfehler führen, während ein Transfer auf einer tieferen Ebene die Lösungsqualität nicht verbessert, den Berechnungsaufwand jedoch erhöht.

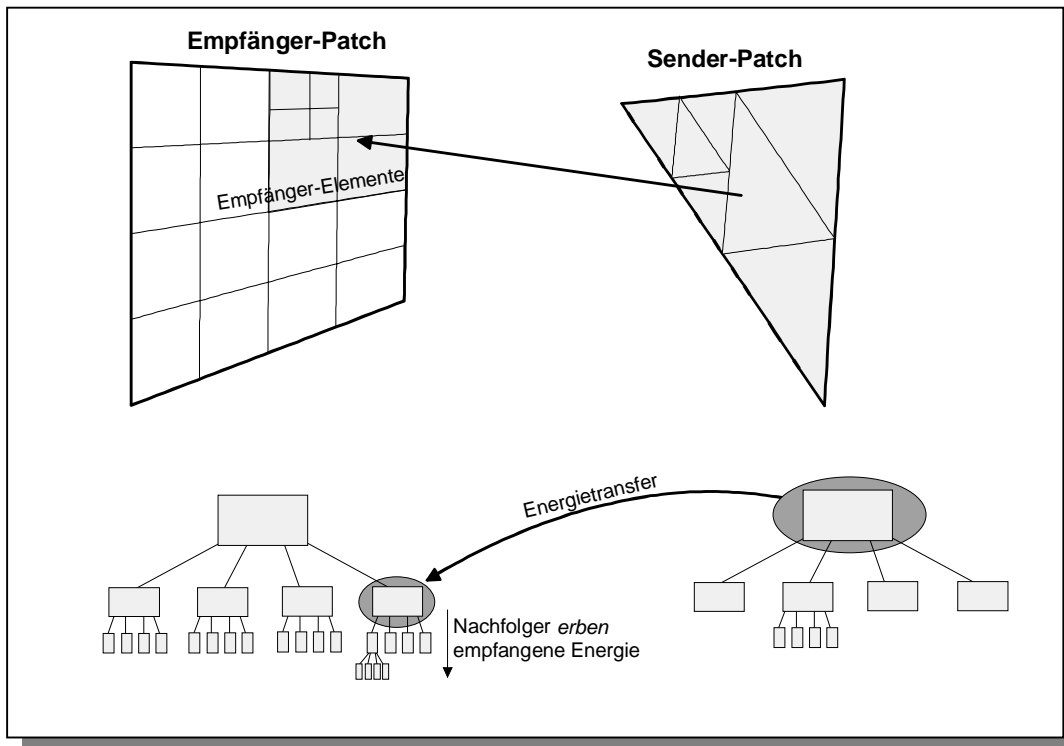


Abb. 5.42: Energietransport zwischen zwei Patch-Hierarchien.

Wie in Abb. 5.42 skizziert, kann der Fall auftreten, daß während der Substrukturierungstests für ein Element auf einer der mittleren Ebenen einer Patch-Hierarchie kein weiterer Bedarf für eine weitere adaptive Verfeinerung in der aktuellen Iteration besteht, obwohl dieses Element in einer der vorangegangenen Iterationen schon weiter unterteilt wurde (d.h. das Element hat in der Patch-Hierarchie weitere Nachfolger). In diesem Fall erfolgt der Energietransfer auf der bestimmten Hierarchie-Ebene und die Nachfolger *erben* die empfangene Energie (siehe unten – *Push/Pull-Operation*). Die Energie von verschiedenen Shooting-Patches wird somit im allgemeinen aufgrund abweichender geometrischer Positionen und Orientierungen sowie unterschiedlichen relativen Sichtbarkeitsverhältnissen auf verschiedenen Ebenen der Patch-Hierarchien empfangen.

Die Berechnungszeit kann durch diese Vorgehensweise erheblich verkürzt werden, da nicht alle Energieanteile grundsätzlich auf der Blatt-Ebene der Patch-Hierarchien verteilt werden müssen. Der Fehler bleibt aufgrund der Abbruchkriterien der Substrukturierungstest kontrollierbar. Dieses entspricht der grundlegenden Idee der *hierarchischen Radiosity-Methode*, bei der ein Energietransport nur auf den Ebenen der Patch-Hierarchien durchgeführt wird, auf denen eine ausreichende Genauigkeit der berechneten Formfaktoren garantiert ist [HSA91]. Dieser Ansatz beruht auf der Beobachtung, daß z.B. zwischen weit entfernten unverdeckten Patches eine weitaus ungenauere Approximation von Formfaktoren ausreicht, ohne das erhebliche Fehler in der Radiosity-Lösung entstehen, als es bei Patches mit einer geringen Distanz zueinander der Fall ist.

Bei der Verwendung von Patch-Hierarchien ist darauf zu achten, daß Elemente auf allen Ebenen Radiosity-Werte zugewiesen sind, die vollständige Radiosity-Lösungen für verschiedene Substrukturierungstiefen beschreiben. Insbesondere bei der Auswahl des nächsten Shooting-Patches werden Radiosity-Werte für die Elemente (bzw. Patches) auf den oberen Hierarchiestufen der Patch-Hierarchien benötigt, wohingegen die Visualisierung der Radiosity-Lösung

im allgemeinen auf Basis der Radiosity-Werte auf den Blattebenen der Patch-Hierarchien erfolgt. Es besteht somit die Notwendigkeit Energie, die während der aktuellen Iteration von dem entsprechenden Shooting-Patch auf verschiedenen Ebenen an Elementen der Patch-Hierarchie eines Empfänger-Patches abgegeben wurden, an die entsprechenden Vorgänger- und Nachfolger-Elemente zu übertragen. Dieses geschieht mit Hilfe der sogenannten *Push/Pull-Operation* [HSA91][CW93][SP94], die in Abb. 5.43 graphisch dargestellt ist. Hierbei deuten die dunkleren Farben größere Radiosity-Werte an.

Die Push/Pull-Operation basiert auf einem rekursiven Durchlauf durch eine Patch-Hierarchie (Quadtree) gemäß einer Tiefensuche, nachdem die nicht verschossene spezifische Ausstrahlung eines Shooting-Patches an die Elemente einer Patch-Hierarchie transferiert wurde. Hierbei werden Radiosity-Werte während dem Herabsteigen von der obersten Ebene entlang der Pfade bis zur Blattebene aufsummiert (*Push*). Beim Heraufsteigen wird der Radiosity-Wert eines Patches als flächengewichteter Durchschnitt seiner direkten Nachfolger innerhalb der Hierarchie gesetzt (*Pull*). Als Ergebnis einer Push/Pull-Operation hat jede Ebene der Patch-Hierarchie eine Approximation der Energiemenge, die sowohl auf den jeweils oberhalb und unterhalb in der Hierarchie angeordneten Ebenen empfangen wurde.

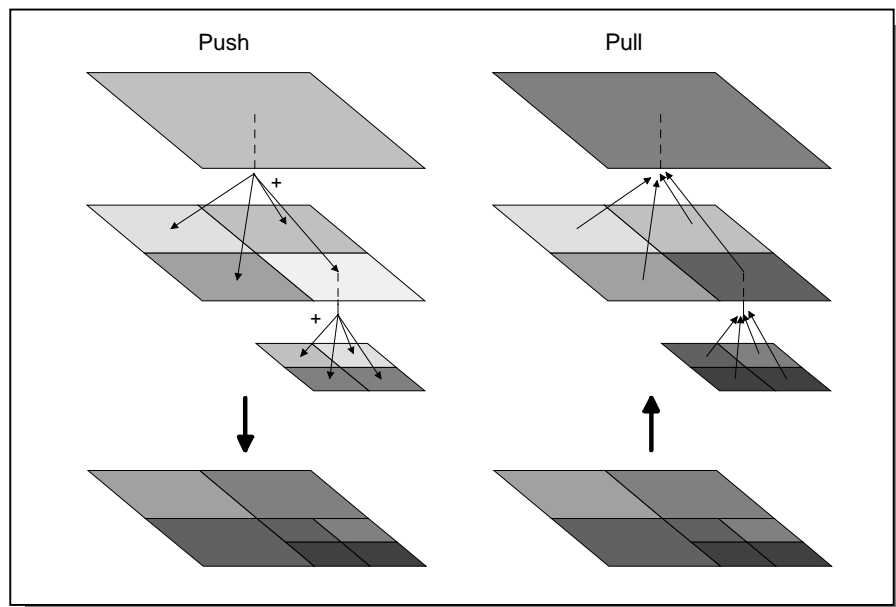


Abb. 5.43: Die Push/Pull-Operation auf einer Patch-Hierarchie.

Bei der Bestimmung von Formfaktoren gemäß dem Monte-Carlo-Verfahren (siehe 3.1.2.6), kann die oben beschriebene Push/Pull-Operation direkt verwendet werden, um die während einer Iteration empfangenen Energiemengen innerhalb der Patch-Hierarchien zu propagieren. Bei der Methode nach Wallace u.a. hingegen wird Energie von Patches an Eckpunkte des adaptiv verfeinerten Radiosity-Netzes abgegeben. Analog zum Monte-Carlo-Verfahren wird ein rekursiver Abstieg innerhalb der Patch-Hierarchien durchgeführt, bis jeweils die Abbruchkriterien der Substrukturierungstests erfüllt sind. Hierbei werden sämtliche Eckpunkt-Radiositäten der besuchten Elemente innerhalb der Hierarchie berechnet. Die Eckpunkt-Radiositäten der Elemente auf den darunterliegenden Ebenen der Hierarchie werden nicht automatisch berechnet. Somit müssen diese Energiewerte aus den Energiewerten der Eckpunkte von Flächenelementen auf höher liegenden Ebenen interpoliert oder direkt berechnet werden. Letzteres führt zu einer genaueren aber wesentlich berechnungsintensiveren Lösung, da Form-

faktoren für alle Eckpunkte der Elemente innerhalb einer Patch-Hierarchie bestimmt werden müssen. Aus diesem Grund bietet sich eine Interpolation dieser Werte an. Hierzu werden die Eckpunkt-Radiosities auf tieferen Ebenen der Patch-Hierarchie auf Basis der Eckpunkt-Radiosities des jeweiligen direkten Vorgängers in der Patch-Hierarchie interpoliert (siehe Abb. 5.44). Diese Interpolation wird während eines rekursiven Abstiegs von der Interaktionsebene innerhalb der Quadree-Datenstruktur bis zu Blattebene fortgesetzt.

Sobald für alle Eckpunkte der Elemente einer Patch-Hierarchie Radiosity-Werte entweder direkt berechnet oder interpoliert wurden, wird in einem Tiefensuch-Durchlauf für jedes Element der Hierarchie ein Patch-Radiosity als Mittelwert der Eckpunkte festgelegt. Dieses ist notwendig, da das Verfahren von Wallace u.a. einen Energietransfer von den Patches an die Eckpunkte des Radiosity-Netzes durchführt. Somit werden für die Auswahl des nächsten Shooting-Patches insbesondere auf der oberen Ebene der Patch-Hierarchien die spezifische Ausstrahlung des Patches benötigt.

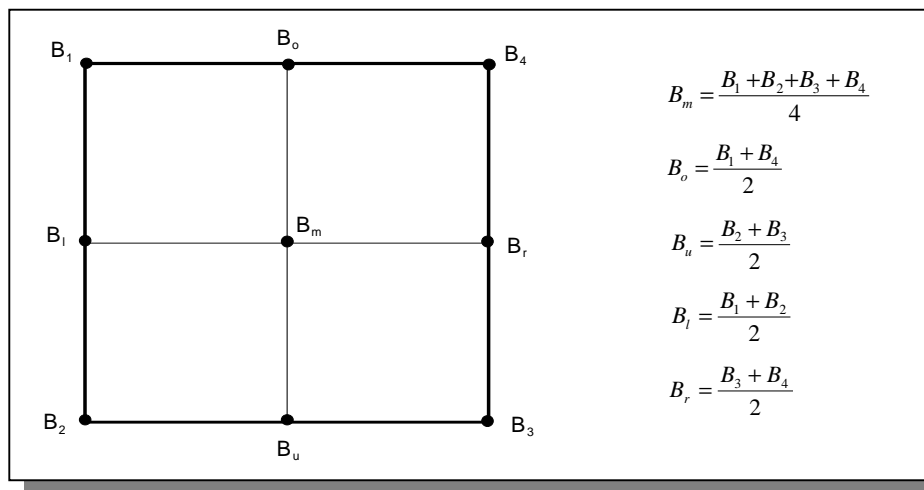


Abb. 5.44: Interpolation von Punkt-Radiosities an Eckpunkten von Elementen auf Basis der Eckpunkt-Radiosities des Vorgänger-Patches in der Patch-Hierarchie.

5.3.1.6 Unterstützte Typen von Lichtquellen

Im Gegensatz zu dem größten Teil der existierenden Radiosity-Verfahren, die zumeist lediglich perfekt diffus strahlende Flächenlichtquellen (Lambertstrahler) unterstützen, werden durch das hier beschriebene Verfahren verschiedene Typen von Lichtquellen mit unterschiedlichen Eigenschaften unterstützt. Somit lassen sich gezielt Beleuchtungssituationen simulieren, die mit Flächenlichtern unmöglich oder nur sehr umständlich erzielt werden können. Unterstützt werden die in Modellierungswerkzeugen gängigsten Lichtquellentypen wie

- Flächenlichtquellen,
- Punktlichtquellen,
- Spot-Lichtquellen sowie
- Parallellichtquellen.

Bei den Flächenlichtquellen handelt es sich um die bisher berücksichtigten primären Lichtquellen, d.h. Patches, denen initial Energie zugewiesen wurde, die ideal diffus in die Umgebung abgestrahlt wird (Lambertstrahler). Punktlichtquellen strahlen Licht von einem Punkt

mit gleicher Strahlungsstärke in alle Richtungen ab. Die spezifische Ausstrahlung (Radiosity) nimmt dabei umgekehrt proportional mit dem Quadrat der Entfernung zur Lichtquelle ab. Ein Spot-Licht strahlt Lichtenergie von einem bestimmten Punkt der Umgebung ab, wobei die spezifische Ausstrahlung proportional zu der Größe des Winkels zu einer Hauptabstrahlrichtung und umgekehrt proportional mit Quadrat zur Entfernung abnimmt. Für Parallellichtquellen wird angenommen, daß sie sich in einer bestimmten Richtung im unendlichen befinden. Somit ist die spezifische Ausstrahlung konstant.

Obwohl die verschiedenen Lichtquellentypen konzeptionell sehr einfach sind, müssen bei deren Verwendung im Kontext von Radiosity-Berechnungen einige Eigenarten berücksichtigt werden. Da Punkt-, Spot- und Parallellichtquellen keine Flächen besitzen und die Abstrahleigenschaften nicht mehr ideal diffus sind, ergibt sich für jeden Lichtquellentyp eine modifizierte Formel zur Formfaktorapproximation. Hierbei handelt es sich im wesentlichen um Punkt-zu-Patch-Formfaktoren, die an die Gegebenheiten der einzelnen Lichtquellentypen angepasst wurden. In [CW93] ist eine genaue Beschreibung und Herleitung der einzelnen Formfaktor-Formeln angegeben.

5.3.1.7 Terminierung des Verfahrens

Ein wesentlicher Aspekt bei einem Progressive-Refinement-Verfahren ist die Festlegung des Terminierungszeitpunktes der Beleuchtungssimulation. Dieses kann sowohl über die Vorgabe einer festen durchzuführenden Iterationszahl als auch über die Vorgabe einer Konvergenzschranke geschehen. Bei der Vorgabe einer maximalen Anzahl von Iterationen liegt der Nachteil auf der Hand. Im allgemeinen weiß ein Anwender nicht, wieviel Energie pro Iteration in der Szene verteilt wird und welcher Anteil dieser Energie von den Oberflächen der Szene absorbiert oder reflektiert wird. Bevor der Berechnungsprozeß nicht abgeschlossen ist, kann aus diesem Grund die Anzahl der Iterationen nicht abgeschätzt werden, die durchzuführen ist, um eine 50%, 80% oder 99% konvergierte Lösung zu berechnen. Somit kann bei Vorgabe der Iterationszahl nicht berücksichtigt werden, wie nah die zu berechnende Lösung an einer vollständig konvergierten Lösung liegen soll.

Der bessere Ansatz besteht darin, den Terminierungszeitpunkt in Abhängigkeit der gesamten nicht verschossenen spezifischen Ausstrahlung zu definieren, die zu einem bestimmten Zeitpunkt in einer Szene enthalten ist. Hierzu wird vor jeder neuen Iteration die aktuell in der Szene enthaltene, zur Zeit noch nicht verschossene spezifische Ausstrahlung ermittelt und mit der initial in der Szene enthaltenen summierten spezifischen Ausstrahlungen der primären Lichtquellen verglichen. Wenn die Summe der aktuell in der Szene enthaltenen nicht verschossenen spezifischen Ausstrahlungen von Patches einen vorgegebenen Prozentsatz der initialen Energie unterschreitet, werden die Berechnungen beendet.

Da während einer Radiosity-Simulation in statischen Szenen keine neue Energie zugeführt wird und pro Iteration aufgrund der Eigenschaften von Formfaktoren (siehe 3.1.2) von den Empfänger-Patches in der Summe nicht mehr Energie empfangen als von dem Shooting-Patch abgegeben wird, ist die Terminierung des Verfahrens garantiert. Weiterhin weiß der Anwender, wie weit die berechnete Lösung von einer 100% konvergierten Lösung entfernt ist. Eine Lösung ist hierbei zu 100% konvergiert, wenn die noch zu verteilende Restenergie 0% der initialen Energie beträgt. Dieses Ziel ist aufgrund numerischer Ungenauigkeiten jedoch im allgemeinen nicht zu erzielen. Eine Konvergenzschranke von 99% liegt jedoch im Bereich der numerischen Genauigkeit des beschriebenen Radiosity-Verfahrens.

5.3.2 Parallelisierung der Progressive-Refinement-Methode

Selbst für Szenen mit moderater Komplexität sind die Laufzeiten des in 5.3.1 beschriebenen sequentiellen Verfahren sehr groß, sobald eine intensive adaptive Substrukturierung der Szene erzeugt wird oder wenn größere Anzahlen von Sample-Punkten verwendet werden, um die Genauigkeit der Formfaktorapproximation zu erhöhen. Dieser Sachverhalt wird durch die in Tabelle 5.20 angegebenen sequentiellen Laufzeiten zur Berechnung einer zu 99% konvergierten Radiosity-Lösung auf Basis der Methode von Wallace u.a. [WKEH89] (siehe 5.3.1.1) für eine Szene mit einer sehr geringen Anzahl von initialen Patches verdeutlicht. Die Berechnungskomplexität und die Tiefe der adaptiven Unterteilungen von Patches wurde hierbei variiert. Zunächst wurden Substrukturierungen nur bei der Verteilung der Energie von primären Lichtquellen durchgeführt. In einer zweiten Stufe wurde ein zweiter Substrukturierungstest verwendet, um bestimmte Unstetigkeiten in der Beleuchtungsfunktion eines Patches zu erkennen (siehe 5.3.1.4). Im Anschluß daran wurden adaptive Unterteilungen von Patches in jeder Iteration ermöglicht. Letztendlich wurde die Anzahl von Sample-Strahlen erhöht, um ein genaueres Abtasten der Geometrie der jeweiligen Shooting-Patches zu ermöglichen. Der Tabelle ist zu entnehmen, daß mit jeder zusätzlichen Anforderung an den Lösungsprozeß eine deutliche Steigerung der sequentiellen Laufzeit verbunden ist. Ein ähnliches Laufzeitverhalten ist auch bei der Verwendung des Monte-Carlo-Verfahrens zur Approximation von Formfaktoren zu beobachten.

Szene: Büro (initial 1289 Patches mit einer Lichtquelle)				
<i>Maximale Tiefe der Patch-Hierarchie</i>	<i>Substrukturierung nur bzgl. Lichtquelle</i>	<i>Zusätzlicher Substrukturierungstest</i>	<i>Substrukturierung für jede Iteration</i>	<i>Erhöhen der Anzahl von Sample-Punkten</i>
3	1m 41s	5m 28s	6m 07s	26m 38s
4	2m 22s	7m 50s	11m 28s	39m 21s
5	5m 03s	17m 35s	33m 48s	1h 25m 18s

Tabelle 5.20: Sequentielle Laufzeiten zur Berechnungen einer 99% konvergierten Radiosity-Lösung für verschiedene Substrukturierungstiefen und variierende Anzahlen von Sample-Strahlen.

Die angegebenen Laufzeiten verdeutlichen, daß eine Parallelisierung dieses Verfahrens notwendig ist, um Berechnungen mit großer Genauigkeit auch in komplexen Umgebungen in einem vertretbaren Zeitrahmen durchführen zu können. In Abschnitt 5.3.2 wird nun ein neuer Parallelisierungsansatz der Progressive-Refinement-Methode für statische Szenen beschrieben und dessen Effizienz demonstriert.

5.3.2.1 Abschätzung der Sample-Strahlen

Der zeitaufwendigste Teil bei den Radiosity-Berechnungen ist die Approximation von Formfaktoren, wobei wiederum der größte Teil der Zeit für die Auswertung von Sichtbarkeitsbeziehungen benötigt wird. Hierbei wird die relative Sichtbarkeit zwischen Sample-Punkten auf Empfänger- und Shooting-Patches mit Hilfe der in Abschnitt 5.3.1 beschriebenen Raycasting-Techniken überprüft. Die Genauigkeit der Formfaktorapproximation korreliert bei den hier verwendeten Methoden mit der Anzahl der verwendeten Sample-Punkten auf den Patches sowie den Substrukturierungstiefen (Tiefe der Patch-Hierarchien) der Patches. Durch eine Veränderung dieser beiden Faktoren wird die Anzahl der Sample-Strahlen beeinflusst, für welche die sehr teuren Schnittpunktüberprüfungen mit den Objekten der Szene durchgeführt werden müssen. Es ist möglich auf Basis dieser Einflußfaktoren eine Abschätzung für die

Anzahl der Sample-Strahlen anzugeben, die bei der Bestimmung von Formfaktoren zwischen einem Shooting-Patch und den Elementen einer Patch-Hierarchie generiert werden.

Die Anzahl der Elemente einer Patch-Hierarchie (d.h. Knoten innerhalb des entsprechenden Quadrees), die bis auf eine Ebene m vollständig substrukturiert wurde, ist gegeben durch

$$\sum_{l=0}^m 4^l = \frac{4^{m+1} - 1}{3}. \quad (5.13)$$

Die Anzahl von Eckpunkten in einer Patch-Hierarchie, die bis auf Ebene m vollständig substrukturiert wurde, unterscheidet sich für dreieckige und viereckige Patches. Sie ist gegeben durch

$$\left(2 + \sum_{l=0}^{m-1} 2^l\right)^2 = (2^m + 1)^2 = 2^{2m} + 2^{m+1} + 1 \quad (5.14)$$

für ein viereckiges Patch und

$$(2^m + 1)^2 - \sum_{l=0}^{2^m} l = \frac{(2^m + 1)^2 + 2^m + 1}{2} \quad (5.15)$$

für ein dreieckiges Patch.

Bei der implementierten Methode zur Bestimmung von Formfaktoren zwischen einem Patch und den Eckpunkten des Radiosity-Netzes nach Wallace u.a. hängt die Anzahl der insgesamt generierten Sample-Strahlen von den durchgeführten Substrukturierungstest ab (siehe 5.3.1.4). Die Betrachtungen werden hier für viereckige Patches vorgenommen. Die Abschätzungen lassen sich analog für dreieckige Patches angeben.

Zur Durchführung des ersten Substrukturierungstests werden Formfaktoren nur an den Eckpunkten des Radiosity-Netzes bestimmt. Hierzu werden von den Eckpunkten der Elemente einer Patch-Hierarchie Sample-Strahlen in Richtung von k auf einem Shooting-Patch zufällig verteilten Sample-Punkten generiert. Die Anzahl der Sample-Strahlen die zur Durchführung eines Energietransfers auf Ebene m generiert werden, ist somit gegeben durch

$$k \cdot (2^{2m} + 2^{m+1} + 1) = O(k \cdot 2^{2m}) = O(k \cdot 4^m). \quad (5.16)$$

Wenn zusätzlich der zweite in 5.3.1.4 beschriebene Substrukturierungstest ausgeführt wird, werden zusätzliche Sample-Punkte auf den Elementen der Patch-Hierarchie eines Empfänger-Patches generiert. Für diese Punkte sind ebenfalls Formfaktoren bzgl. des Shooting-Patches zu bestimmen. Wenn die Anzahl der Sample-Punkte auf den Empfänger-Elementen und dem Shooting-Patch hierbei durch k vorgegeben ist, werden

$$k^2 \cdot \left(\frac{4^{m+1} - 1}{3}\right) = O(k^2 \cdot 4^{m+1}) \quad (5.17)$$

zusätzliche Sample-Strahlen generiert.

Aus den Abschätzungen (5.16) und (5.17) ergibt sich, daß für einen Energietransfer zwischen einem Shooting-Patch und einer bis auf Ebene m vollständig substrukturierten Patch-Hierarchie maximal $O(k^2 \cdot 4^{m+1})$ Sample-Strahlen generiert werden. Hieraus ist ersichtlich, daß die Substrukturierungstiefe eines Patches der entscheidende Faktor für die Komplexität der Durchführung eines Energietransfers ist. Diese ist sowohl von der initialen Netzgenerierung als auch von der minimalen zulässigen Kantenlänge eines adaptiv erzeugten Elementes (Abbruchkriterium der Substrukturierung) abhängig.

Wie in 5.3.1.4 beschrieben, wird bei der auf Monte-Carlo-Integration basierenden Methode zur Formfaktorapproximation lediglich ein Substrukturierungstest benötigt. Wenn eine Patch-Hierarchie bis zu einer Tiefe m vollständig substrukturiert ist und während einer Iteration auf dieser Ebene ein Energietransfer erfolgt, werden insgesamt

$$k \cdot \left(\frac{4^{m+1} - 1}{3} \right) = O(k \cdot 4^{m+1}) \quad (5.18)$$

Sample-Strahlen generiert. Hierbei ist k die Anzahl der Sample-Punkte, die sowohl auf dem Shooting-Patch als auch auf einem Empfänger-Element generiert werden.

Es ist zu beachten, daß die vorgenommenen Abschätzungen für ein Patch der Szene gilt. Da die Substrukturierungstiefen verschiedener Patches der Szene deutlich voneinander abweichen können, schwanken auch die Anzahlen von Sample-Strahlen, die zur Durchführung eines Energietransfers von einem Shooting-Patch an verschiedene Patch-Hierarchien generiert werden, mitunter sehr stark.

Aus den vorangegangenen Betrachtungen folgt direkt, daß ein Parallelisierungsansatz sich auf die parallele Auswertung von Sichtbarkeitsbeziehungen konzentrieren sollte, da dieses sowohl die am häufigsten auszuführende als auch die teuerste Operation bei der Berechnung einer Radiosity-Lösung ist.

5.3.2.2 Die grundlegende Idee des parallelen Verfahrens

Die Granularität der parallelen Berechnungen, d.h. die Festlegung der atomaren Berechnungsoperationen, die unabhängig voneinander ausgeführt werden können, ist von entscheidender Bedeutung für die Effizienz der parallelen Berechnungen. Ein feingranularer Ansatz zur Parallelisierung der in 5.3.1 beschriebenen sequentiellen Progressive-Refinement-Methode besteht darin, sämtliche relativen Sichtbarkeitsbeziehungen zwischen Sample-Punkten bei der Approximation von Formfaktoren vollständig parallel auszuwerten. Dieser Ansatz bietet zwar einen maximalen Parallelitätsgrad, da jedoch zur Bestimmung eines Formfaktors mehrere Sample-Strahlen generiert werden, um die Genauigkeit der Approximation zu erhöhen, erfordert diese Vorgehensweise einen großen Kommunikationsaufwand. Parallel berechnete Sichtbarkeitsinformationen müssen auf einem Prozessor gesammelt werden, um den eigentlichen Formfaktor zu bestimmen und die entsprechenden Radiosity-Werte der Empfänger-Elemente bzw. den Eckpunkte des Radiosity-Netzes gemäß des ermittelten Formfaktors zu berechnen.

Ein besserer Ansatz besteht darin, den Energietransfer von einem gegebenen Shooting-Patch an die Elemente innerhalb der Patch-Hierarchie eines Empfänger-Patches vollständig auf einem Prozessor auszuführen. Somit werden sämtliche relativen Sichtbarkeitsbeziehungen, die zur Bestimmung eines Formfaktors ausgewertet werden müssen, auf einem Prozessor berech-

net. Ein solcher Ansatz ist zwar durch eine gröbere Granularität der parallelen Berechnungen gekennzeichnet, ermöglicht aber die Berechnung von Formfaktoren, ohne das eine Notwendigkeit zur Kommunikation ausgewerteter Sichtbarkeitsbeziehungen zwischen den Prozessoren besteht. Die lokale Bestimmung der relativen Sichtbarkeit zwischen zwei Sample-Punkten wird im Detail in 5.3.2.4 beschrieben.

Ziel ist somit die Entwicklung eines datenparallelen Verfahrens, bei welchem das Radiosity-Netz verteilt auf den Prozessoren eines Parallelrechners gespeichert wird und Formfaktoren zwischen Shooting-Patches und Elementen innerhalb der Patch-Hierarchien von Empfänger-Patches lokal auf einem Prozessor bestimmt werden. Während der parallelen Progressive-Refinement-Berechnungen führt jeder Prozessor einen Energietransfer zwischen einem ausgewählten Shooting-Patch und den jeweils lokal verfügbaren Patches durch. Hierbei wird gemäß dem in 5.3.1.4 und 5.3.1.5 beschriebenen rekursiven (hierarchischen) Ansatz vorgegangen. Problematisch ist jedoch, daß aufgrund der rekursiven Berechnung von Formfaktoren innerhalb der Patch-Hierarchien für verschiedene Empfänger-Patches unterschiedliche Anzahlen von Sample-Strahlen generiert werden (siehe 5.3.2.1). Dieses führt bei einer ungünstigen statischen Verteilung von Patches auf die Prozessoren eines Parallelrechners mitunter zu erheblichen Abweichungen der Berechnungszeiten eines Energietransfers von einem Shooting-Patch an Patch-Hierarchien verschiedener Empfänger-Patches und somit zu deutlichen Schwankungen der Iterationszeiten.

5.3.2.3 Statische Lastverteilung

Ziel der statischen Lastverteilung ist es, eine gleichmäßige initiale Aufteilung des Eingabeproblems auf die zur Verfügung stehenden Prozessoren zu erreichen. Hierbei soll einerseits bei datenparallelen Berechnungen eine gute Ausnutzung des verteilten Speichers eines Rechnersystems erreicht werden. Andererseits soll auch die Berechnungskomplexität der definierten Teilprobleme nicht zu stark variieren, um möglichst effiziente parallele Berechnungen zu ermöglichen.

Im Kontext der parallelen Radiosity-Berechnungen muß somit eine Verteilung der während der initialen Netzgenerierung erzeugten Patches auf die Prozessoren eines Parallelrechners vorgenommen werden. Hierbei ist den Prozessoren eine annähernd gleiche Anzahl von Patches zuzuordnen. Die Grundidee ist, daß durch eine verteilte Speicherung des Radiosity-Netzes auch für sehr große Szenen Radiosity-Berechnungen ermöglicht werden. Die Berechnungskomplexität einer Iteration des Progressive-Refinement-Verfahrens korreliert direkt mit der Anzahl der auf einem Prozessor lokal gespeicherten Patches. Durch eine möglichst *sinnvolle* Verteilung der Patches kann erreicht werden, daß die Abweichungen der Zeiten zur Durchführung einer Shooting-Iteration auf den verschiedenen Prozessoren bzgl. eines gegebenen Shooting-Patches lediglich geringfügig voneinander abweichen. Nur in diesem Fall werden Wartezeiten von Prozessoren während der parallelen Berechnungen gering gehalten und somit eine gute Effizienz erzielt.

Das in 5.2 eingeführte datenparallele Verfahren basiert auf einer Aufteilung der Szene in geometrisch zusammenhängende Teilszenen, die innerhalb von konvexen Begrenzungszellen angeordnet sind. Durch diese Vorgehensweise sollen Eingabeszenen beliebiger Komplexität in Teilszenen mit annähernd der gleichen Anzahl von Patches aufgeteilt werden, um beliebig skalierbare datenparallele Berechnungen zu ermöglichen. Weiterhin soll die Ausnutzung der Datenlokalität durch diese Art der Szenenaufteilung ermöglicht werden, um den Mehraufwand des parallelen Verfahrens gering zu halten, der durch Kommunikation bei der Bestimmung von Sichtbarkeitsbeziehungen zwischen den Patches der Teilszenen entsteht.

Wie in 5.2.1 gezeigt wurde, kann durch Aufteilung der Szene in zusammenhängende Teilszenen nicht immer eine gleichmäßige initiale Verteilung der Patches auf die Prozessoren erreicht werden. Weiterhin entsteht gerade dadurch, daß geometrisch zusammenhängende Teilszenen generiert werden, ein dynamisches Lastungleichgewicht während der parallelen Berechnungen. Ein Grund hierfür ist, daß die Materialeigenschaften der Objekte (Patches) innerhalb der Teilszenen und deren Positionen relativ betrachtet zu den Lichtquellen der Szene, ähnlich sind. Dieses verursacht deutliche Abweichungen der Energiemengen, die von den Patches verschiedener Teilszenen empfangen werden. Somit müssen in einigen Teilszenen mehr lokale und externe Shooting-Iterationen ausgeführt werden als in anderen, um eine lokal konvergierte Radiosity-Lösung zu bestimmen. Einen noch entscheidenderen Einfluß auf die unterschiedlichen Berechnungszeiten der Prozessoren zur Durchführung von Shooting-Iterationen innerhalb der lokalen Teilszenen hat jedoch die adaptive Verfeinerung von Patches. Schattengrenzen verlaufen zumeist nicht nur durch einzelne Patches, sondern auch durch angrenzende Patches. Alle betroffenen Patches werden während der Radiosity-Berechnung adaptiv unterteilt, um den Verlauf einer Schattengrenze genauer zu approximieren. Somit bringt die Partitionierung in geometrisch zusammenhängende Regionen der Eingabeszene einen weiteren entscheidenden Nachteil mit sich. Es besteht die Gefahr, daß Teilszenen gebildet werden, für die im Verlauf der Berechnungen eine große Anzahl adaptiver Verfeinerungen von Patches vorgenommen werden müssen, während Patches anderer Teilszenen kaum unterteilt werden. Dieses führt zu deutlich abweichenden Iterationszeiten innerhalb der einzelnen Teilszenen und somit zu Wartezeiten von Prozessoren, welche die Effizienz der parallelen Berechnungen erheblich beeinträchtigen.

Ein anderer Ansatz zur statischen Lastverteilung im Kontext datenparalleler Radiosity-Berechnungen ist durch eine gleichmäßige Verteilung der Patches auf die Prozessoren eines Parallelrechners gegeben. Hierbei wird für jeden verfügbaren Prozessor eine Teilmenge der während der Netzgenerierung erzeugten Patches ausgewählt. Bei der Aufteilung wird darauf geachtet, daß jedem Prozessor die gleiche Anzahl von Patches zugeordnet wird, wobei die einem Prozessor zugeordneten Patches möglichst nicht unmittelbar benachbart sein sollen. Ein Vorteil dieser Vorgehensweise besteht darin, daß beliebige Anzahlen von Prozessoren verwendet werden können, wobei es möglich ist, jedem Prozessor annähernd die gleiche Anzahl von Patches zuzuordnen. Es kann zu einer maximalen Abweichung von einem Patch zwischen den Prozessoren kommen. Wie in 5.2.1 gezeigt wurde, ist ein solches Resultat mit der dort eingeführten Partitionierungsstrategie nicht möglich.

geometrische Partitionierung					gleichmäßige Verteilung				
1	1	2	2	2	1	2	1	2	4
1	1	2	2	2	3	4	3	1	3
3	3	3	4	4	4	2	1	4	2
3	3	3	4	4	1	4	3	2	3

Abb. 5.45: Gegenüberstellung einer geometrischen Partitionierung und einer gleichmäßigen Verteilung von Patches auf vier Prozessoren.

In Abb. 5.45 ist ein Beispiel für die Aufteilung einer Menge von Patches auf vier Prozessoren gegeben. Die Zuordnung im linken Teil des Bildes wird anhand des in 5.2.1 beschriebenen geometrisch basierten Partitionierungsverfahren bestimmt, während im rechten Teil des Bildes eine zufällige Zuordnung der Patches vorgenommen wird.

Weitere Vorteile der zufälligen Verteilung von Patches auf die Prozessoren ergeben sich aus bestimmten Beobachtungen, die bei Radiosity-Berechnungen mit adaptiver Substrukturierung gemacht werden können.

Patches die dicht beieinander liegen, haben ungefähr die gleiche Distanz zu einem ausgewählten Shooting-Patch. Zusätzlich haben direkt benachbarte Patches auf einer Oberfläche mit einer gemeinsamen Kante oder mindestens einem gemeinsamen Eckpunkt eine ähnliche Ausrichtung im dreidimensionalen Raum. Aufgrund der Definition der Formfaktoren, wie sie in 3.1.2 gegeben ist, kann gefolgert werden, daß direkt benachbarte Patches annähernd die gleiche Menge der spezifischen Ausstrahlung eines Shooting-Patches empfangen. Diese Energiemenge kann lediglich aufgrund von Verdeckungen größeren Schwankungen unterliegen. Weiterhin ist zu beobachten, daß der größte Teil aller adaptiven Verfeinerungen von Patches entlang von Schattengrenzen erfolgt. Somit haben die Patch-Hierarchien direkt benachbarter Patches mit sehr großer Wahrscheinlichkeit eine ähnliche Substrukturierungstiefe.

Aus diesen Beobachtungen kann gefolgert werden, daß ein Energietransfer von einem gegebenen Shooting-Patch an direkt benachbarte Patch-Hierarchien auf annähernd der gleichen Substrukturierungsstufen (Ebenen innerhalb der Patch-Hierarchien) erfolgt. Aus diesem Grund ist die Anzahl der zu generierenden Sample-Strahlen, und somit die Berechnungskomplexität dieses Energietransfers, für direkt benachbarte Patches mit großer Wahrscheinlichkeit gleich. Aufgrund der Abhängigkeit der Zeit zur Durchführung eines Energietransfers an die Elemente einer Patch-Hierarchie von der Anzahl der generierten Sample-Strahlen, sind diese Zeiten für direkt benachbarte Patches während einer Iteration annähernd gleich, können jedoch von Iteration zu Iteration mitunter erheblichen Schwankungen unterliegen. Bei der gleichmäßigen Verteilung der Patches auf die Prozessoren wird darauf geachtet, daß direkt benachbarte Patches auf verschiedene Prozessoren plaziert werden. Dadurch weisen die Zeiten zur Durchführung einer vollständigen Iteration bzgl. eines gewählten globalen Shooting-Patches auf allen Prozessoren nur geringe Abweichungen voneinander auf. Somit wird das dynamische Lastungleichgewicht durch die statische Plazierung der Patches des Radiosity-Netzes auf die Prozessoren gering gehalten, ohne das ein expliziter Lastausgleich mittels Kommunikation zwischen den Prozessoren notwendig wird.

Die oben aufgeführten Beobachtungen implizieren, daß durch die Verteilung benachbarter Patches auf verschiedene Prozessoren während der parallelen Berechnungen eine gleichmäßige Auslastung des verteilten Speichers gegeben ist. Dieses gilt auch wenn adaptive Verfeinerungen von Patches vorgenommen und somit die Tiefen der Patch-Hierarchien vergrößert werden. Im Gegensatz zur geometrischen Partitionierung der Szene kommt es nur zu einer geringeren Abweichung des auf den Prozessoren jeweils lokal benötigten Speichers. Dieser Sachverhalt wird in Tabelle 5.21 verdeutlicht. Dort ist für verschiedene Testszenen angegeben, wie sich die Elemente der Patch-Hierarchien im Anschluß an parallelen Berechnungen mit adaptiver Substrukturierung auf die einzelnen Prozessoren verteilen. Eine optimale Verteilung bei den durchgeführten Berechnungen auf vier Prozessoren ist gegeben, wenn jeder Prozessor 25% der in sämtlichen Patch-Hierarchien enthaltenen Elemente speichert. Es ist sehr deutlich zu sehen, daß bei einer initialen Verteilung benachbarter Patches auf verschiedene Prozessoren die Elemente der Hierarchien im Anschluß an die parallelen Berechnungen wesentlich gleichmäßiger verteilt sind.

Szene	Geometrische Partitionierung				Verteilung benachbarter Patches			
	Proz. 1	Proz. 2	Proz. 3	Proz. 4	Proz. 1	Proz. 2	Proz. 3	Proz. 4
<i>Büro</i>	14,92 %	41,82 %	18,59 %	24,67 %	26,13 %	20,56 %	27,32 %	25,98 %
<i>Konferenzraum</i>	31,60 %	23,73 %	24,21 %	20,47 %	25,03 %	26,36 %	24,24 %	24,37 %
<i>Townhouse</i>	17,60 %	28,86 %	24,20 %	29,33 %	24,35 %	25,72 %	25,13 %	24,79 %

Tabelle 5.21: Vergleich der Verteilung von lokal gespeicherten Elementen der Patch-Hierarchien verschiedener Szenen nach einer parallelen Radiosity-Berechnung mit unterschiedlichen statischen Lastverteilungen.

Eine gleichmäßige initiale Verteilung von Patches kann für das in 5.2 beschriebene datenparallele Verfahren nicht verwendet werden, da die lokalen Berechnungen und die verteilte Sichtbarkeitsberechnungen voraussetzen, daß die den Prozessoren zugeordneten Teilszenen in konvexen Begrenzungszellen angeordnet sind.

Bei einer statischen Lastverteilung, die auf einer gleichmäßigen Verteilung von Patches auf die Prozessoren basiert, kann nicht mehr von Bestimmung von lokalen Teilszenen im eigentlichen Sinne gesprochen werden. Die lokalen Radiosity-Netze repräsentieren keine zusammenhängenden Teilregionen der Szene mehr. Die einzelnen Patch-Hierarchien können vielmehr als unabhängige Arbeitspakete betrachtet werden, wobei die Aufgabe darin besteht, die Energie eines gegebenen Shooting-Patches gemäß vorgegebener Qualitätskriterien an die Elemente der Patch-Hierarchien zu verteilen (siehe 5.3.1.4).

5.3.2.4 Auswahl von Shooting-Patches

Zu Beginn jeder Iteration des Progressive-Refinement-Verfahrens wird ein Patch der Umgebungsbeschreibung ausgewählt, dessen nicht verschossene spezifische Ausstrahlung maximal ist. Diese Energiemenge wird an die übrigen Patches der Szene verteilt. Der Grundgedanke bei dieser Vorgehensweise ist, daß zunächst die Energieanteile innerhalb der Szene verteilt werden sollen, welche die größten Auswirkungen auf die globale Beleuchtung haben. Somit wird sehr schnell eine gute Ausleuchtung der Szene erreicht.

Bei dem hier beschriebenen Verfahren werden die Iterationen des Progressive-Refinement-Verfahrens parallel auf mehreren Prozessoren ausgeführt. Jeder Prozessor hat die Aufgabe, die nicht verschossene spezifische Ausstrahlung eines ausgewählten Shooting-Patches an die Elemente der lokal gespeicherten Patch-Hierarchien zu verteilen. Entscheidend für die Effizienz der parallelen Berechnungen ist die Auswahlstrategie der Shooting-Patches. Aufgrund der verteilten Speicherung der Szenen ist bei der Auswahl von Shooting-Patches Kommunikation zwischen den Prozessoren notwendig, da die Energie eines Shooting-Patches grundsätzlich an alle Patches der Szene verteilt werden muß und nicht nur an eine lokal verfügbare Teilmenge. Andernfalls entspricht die parallel berechnete Radiosity-Lösung nicht der Lösung des Radiosity-Gleichungssystems, das ein Energiegleichgewicht innerhalb der Eingabeszene beschreibt (siehe 3.1).

Analog zu dem in 5.2 beschriebenen parallelen Verfahren wird bei der Auswahl eines Shooting-Patches grundsätzlich zwischen einem *lokalen* und einem *externen Shooting-Patch* unterschieden (siehe 5.2.3.4). Zusätzlich wird zwischen einem synchronen und einem asynchronen Ansatz zur Auswahl von Shooting-Patches unterschieden.

Bei der synchronen Auswahlstrategie wird für die parallel ausgeführten Shooting-Iterationen jeweils das Shooting-Patch ausgewählt, das aus globaler Sicht die maximale nicht verschos-

sene spezifische Ausstrahlung besitzt. Zunächst bestimmt jeder Prozessor das lokal gespeicherte Patch mit der größten nicht verschossenen spezifischen Ausstrahlung. In einer synchronisierten Kommunikationsphase werden die Geometriedaten und Energiewerte dieser lokal bestimmten Patches zwischen den Prozessoren ausgetauscht. Jeder Prozessor wählt das eindeutig bestimmte Patch mit der maximalen nicht verschossenen spezifischen Ausstrahlung als Shooting-Patch für die nächste Iteration. Bei Verwendung dieser Auswahlstrategie entsprechen die parallel ausgeführten Iterationen exakt denen der entsprechenden sequentiellen Progressive-Refinement-Berechnung. Wie in 5.3.2.3 erläutert wurde, hat die initiale statische Lastverteilung erhebliche Auswirkungen auf die Ausführungszeiten der Iterationen. Aufgrund unterschiedlicher Ausführungszeiten der einzelnen Iterationen auf den verschiedenen Prozessoren kann es bei diesem synchronisierten Ansatz zu Wartezeiten unmittelbar vor den Synchronisations-Phasen der nächsten Iteration kommen. Dieses beeinträchtigt die Effizienz der parallelen Berechnungen.

Die asynchrone Strategie basiert auf einer lokalen Auswahl der Shooting-Patches durch die Prozessoren, ohne das in einer synchronen Kommunikationsphase das Patch mit der maximalen nicht verschossenen spezifischen Ausstrahlung bestimmt wird. Die Geometriedaten sowie die nicht verschossene spezifische Ausstrahlung eines lokal ausgewählten Shooting-Patches wird an alle übrigen Prozessoren mittels einer Broadcasting-Nachricht übermittelt. Die Empfänger-Prozessoren fügen das entsprechende externe Shooting-Patch in eine lokale Warteschlange ein. Jeder Prozessor führt seine Shooting-Iterationen unabhängig von den anderen Prozessoren aus, wobei die entsprechenden Shooting-Patches jeweils lokal bestimmt oder der Warteschlange entnommen werden. Ob ein Prozessor eine lokale oder eine externe Shooting-Iteration ausführt, hängt von der nicht verschossenen spezifischen Ausstrahlung der Patches ab. Mit Hilfe eines Parameters kann festgelegt werden, ob grundsätzlich eine externe Shooting-Iteration ausgeführt werden soll, sobald ein entsprechendes externes Shooting-Patch in der lokalen Warteschlange enthalten ist, oder ob alternativ die Energie eines lokalen Shooting-Patches verteilt werden soll, wenn dessen nicht verschossene spezifische Ausstrahlung deutlich größer ist, als die der in der Warteschlange enthaltenen externen Shooting-Patches.

Es ist zu beachten, daß bei der asynchronen Auswahlstrategie der Shooting-Patches, im Gegensatz zu der synchronen Auswahlstrategie, nicht der sequentielle Lösungsprozeß nachvollzogen wird. Aufgrund der Bestimmung von Shooting-Patches deren nicht verschossene spezifischen Ausstrahlungen jeweils ein lokales Maximum bilden, werden bei größeren Anzahlen von Prozessoren deutlich mehr Shooting-Patches bearbeitet als im sequentiellen Fall. Hierbei werden jedoch pro Iteration geringere Energiemengen an die lokalen Patch-Hierarchien verteilt. Durch Verwendung eines Terminierungskriteriums, das auf der gesamten in der Szene enthaltenen nicht verschossenen spezifischen Ausstrahlungen von Patches (*Restenergie*) basiert, ist jedoch gewährleistet, daß der sequentielle und der parallele Lösungsprozeß gegen die gleiche Radiosity-Lösung konvergieren (siehe 5.3.2.6).

Während der statischen Lastverteilung wird eine gleichmäßige Verteilung der Patches einer Umgebungsbeschreibung auf die Prozessoren vorgenommen. Im Gegensatz zu einer geometrischen Aufteilung der Szene werden bei dieser Vorgehensweise primäre Lichtquellen-Patches auf verschiedene Prozessoren verteilt, selbst wenn diese räumlich dicht beieinander liegen. Somit kann jeder Prozessor, dem initial ein primäres Lichtquellen-Patch zugeordnet wurde, unmittelbar mit den lokalen asynchronen Berechnungen beginnen und muß nicht warten, bis ihm von einem anderen Prozessor ein externes Shooting-Patch übermittelt wird.

Die Daten, die während der Auswahl von Shooting-Patches ausgetauscht werden, beschränken sich auf einige wenige Informationen, welche die Geometrie des entsprechenden Patches

(Eckpunkte der Patches) sowie dessen nicht verschossene spezifische Ausstrahlung vollständig beschreibt. Somit ist der durch Kommunikation erzeugte Mehraufwand während der parallelen Berechnungen zu vernachlässigen.

Die speziellen Lichtquellentypen, wie sie in 5.3.1.6 beschrieben wurden, nehmen während der parallelen Berechnungen eine Sonderstellung ein. Jeder Prozessor speichert eine vollständige Liste der initial in der Szene enthaltenen speziellen Lichtquellen. Die Energie dieser Lichtquellen wird asynchron und ohne jegliche Kommunikation zwischen den Prozessoren an die lokalen Patch-Hierarchien verteilt, bevor die eigentlichen parallelen Progressive-Refinement-Iterationen ausgeführt werden. Durch diese Vorgehensweise ist garantiert, daß die wesentlichen Energiemengen (und somit die entsprechenden Beleuchtungseffekte), die durch spezielle Lichtquellen abgegeben werden, sehr schnell berücksichtigt werden. Da zur vollständigen Beschreibung der speziellen Typen von Lichtquellen nur sehr wenige Daten notwendig sind, ist deren Speicherung auf allen Prozessoren unproblematisch.

5.3.2.5 Parallele Durchführung von Shooting-Iterationen

Das hier vorgestellte neue parallele Verfahren basiert auf dem in 5.3.1 beschriebenen sequentiellen Progressive-Refinement-Algorithmus. Bei den parallelen Berechnungen ist es die Aufgabe eines Prozessors, die Energie von ausgewählten Shooting-Patches in einem iterativem Prozeß an die lokal verfügbaren Patch-Hierarchien zu verteilen. Vor jeder Iteration wählt ein Prozessor ein Shooting-Patch wie in 5.3.2.5 beschrieben entweder gemäß einer asynchronen lokalen Strategie oder einer synchronen globalen Strategie aus. Hierbei sind Shooting-Patches entweder den Prozessoren während der statischen Lastverteilung zugeordnet (lokales Shooting-Patch) oder von einem beliebigen anderen Prozessor übermittelt worden (externes Shooting-Patch). Der grundlegende Parallelisierungsansatz ist unabhängig von der verwendeten Methode zur Approximation von Formfaktoren. Es kann wahlweise die Methode von Wallace u.a. oder die Monte-Carlo-Integration verwendet werden (siehe 5.3.1.1). Die parallelen Berechnungen werden ausgeführt, bis ein globales Terminierungs-Kriterium erfüllt ist (5.3.1.7 und 5.3.2.6).

Das zentrale Problem besteht darin, nach der Auswahl eines Shooting-Patches durch einen Prozessor die entsprechende nicht verschossene spezifische Ausstrahlung an die Elemente von lokal gespeicherten Patch-Hierarchien oder deren Eckpunkte zu verteilen. Bisher wurde nicht explizit darauf eingegangen, wie bei einer verteilten Speicherung des Radiosity-Netzes gemäß einer zufälligen Verteilung der Patches eine Auswertung der relativen Sichtbarkeitsbeziehungen durchgeführt wird, d.h. wie die Bestimmung von Schnittpunkten der Sample-Strahlen mit Objekten der Eingabeszene erfolgt. Ziel ist es, die parallelen Radiosity-Berechnungen mit möglichst geringem Kommunikationsaufwand auszuführen. Insbesondere während der Auswertung von Sichtbarkeitsbeziehungen soll möglichst wenig Kommunikationsbedarf bestehen. Dieses ist insofern problematisch, da die vollständige Geometrie der Szene bei der Berechnung von Formfaktoren (bzw. den Sichtbarkeitsberechnungen) berücksichtigt werden muß.

Der hier in Abschnitt 5.3 beschriebene neue Ansatz zur Durchführung paralleler Progressive-Refinement-Berechnungen verwendet eine Strahlverfolgungs-Technik, um relativen Sichtbarkeitsbeziehungen während der Approximation von Formfaktoren zu überprüfen. Wie in 5.3.1.3 gezeigt wurde, reicht eine grobe Repräsentation der Geometrie einer Eingabeszene aus, um Schnittpunkte zwischen Sample-Strahlen und Objekten der Szene zu ermitteln. Diese als *Intersection-Netz* bezeichnete Szenenrepräsentation hat eine wesentlich geringere Komplexität als das *Radiosity-Netz*, welches aus adaptiv verfeinerten Patch-Hierarchien besteht und zur

Approximation der Beleuchtungsfunktionen von Objektoberflächen verwendet wird. Durch die funktionale Trennung von verschiedenen Szenenrepräsentationen ist es möglich, während der parallelen Berechnungen relative Sichtbarkeitsbeziehungen zwischen Sample-Punkten ohne Kommunikation zwischen Prozessoren zu realisieren. Da die Komplexität eines Intersection-Netzes im Vergleich zu dem entsprechenden Radiosity-Netz sehr gering ist, kann es ohne Probleme vollständig auf jedem Prozessor gespeichert werden (Tabelle 5.22). Das Intersection-Netz repräsentiert die Geometrie der gesamten Eingabeszene und somit können auf jedem Prozessor relative Sichtbarkeitsbeziehungen ausgewertet werden, ohne dass die Notwendigkeit zur Kommunikation besteht.

Szene	Radiosity-Netz				Intersection-Netz	
	Vor Substrukturierung		Nach adapt. Substrukturierung		Anzahl der Polygone	Speicherplatz
	Anzahl der Polygone	Speicherplatz	Anzahl der Polygone	Speicherplatz		
<i>Büro</i>	1289	0,35 MB	32681	8,7 MB	301	0,03 MB
<i>Tempel</i>	7611	2,03 MB	36204	9,4 MB	3795	0,32 MB
<i>Konferenzraum</i>	5174	1,83 MB	100799	27,51 MB	3880	0,36 MB
<i>Uni-Foyer</i>	18361	4,05 MB	82204	20,6 MB	661	0,08 MB
<i>Townhouse</i>	44254	11,65 MB	1095892	281,5 MB	13627	1,25 MB
<i>Raumschiff</i>	139925	35,60 MB	279299	71,59 MB	56342	4,83 MB
<i>Fürstenallee</i>	93150	22,51 MB	1752426	431,0 MB	22199	1,91 MB

Tabelle 5.22: Speicherverbrauch des Intersection-Netzes im Vergleich zu dem Speicherverbrauch des Radiosity-Netzes für verschiedene Szenen.

Der beschriebene Ansatz ist als datenparallel zu bezeichnen, da die großen Datenmengen eines adaptiv verfeinerten Radiosity-Netzes verteilt gespeichert und die Iterationen des Progressive-Refinement-Verfahrens parallel ausgeführt werden. Das Verfahren ist somit skalierbar, da Radiosity-Lösungen für komplexe Szenen berechnet werden können, indem Radiosity-Netze durch die zuvor beschriebene statische Lastverteilungsstrategie auf eine größere Anzahl von Prozessoren verteilt werden. Eine vollständige datenparallele Lösung kann durch Speicherung des Intersection-Netzes in einem verteilten gemeinsamen Speicher (*Distributed Shared Memory*) erreicht werden. Da lediglich ein lesender Zugriff auf die entsprechenden Daten notwendig ist, kann dieses selbst mit einer Software-Simulation des verteilten gemeinsamen Speichers (z.B. [LKL96]) sehr effizient realisiert werden. Bei ersten Tests mit einer solchen Software-Simulation wurden zufriedenstellende Ergebnisse erzielt.

Zur Beschleunigung der Schnittpunktberechnungen können während der parallelen Berechnungen alle in 5.3.1.3 beschriebenen Techniken wahlweise verwendet werden. Unmittelbar nachdem ein Prozessor die Daten des Intersection-Netzes eingelesen hat, werden jeweils analog zu den sequentiellen Berechnungen die Begrenzungskörper der Objekte bestimmt und gespeichert. Wenn die Octree-Methode verwendet werden soll, wird die entsprechende Datenstruktur zu Berechnungsbeginn auf jedem Prozessor lokal erzeugt. Diese Datenstrukturen werden während der anschließenden Progressive-Refinement-Berechnungen zur Reduktion der Anzahl von auszuführenden Schnittpunkttests verwendet. Wenn die Technik des *Shaft-Cullings* eingesetzt wird, können die entsprechenden Shaft-Volumen mit der zugehörigen Kandidatenliste ebenfalls auf Basis des Intersection-Netzes lokal erzeugt werden. Dieses geschieht jeweils unmittelbar vor einem Energietransfer von einem ausgewählten Shooting-Patch an eine lokale Patch-Hierarchie.

5.3.2.6 Terminierung der parallelen Berechnungen

Ein wesentlicher Punkt bei der parallelen Berechnung von Radiosity-Lösungen ist deren korrekte Terminierung. Zu Berechnungsbeginn muß durch den Benutzer ein Abbruchkriterium für die zu berechnende Lösung vorgegeben werden. Wenn während der Berechnungen dieses Kriterium erfüllt ist, werden die Berechnungen unmittelbar beendet und die bis zu diesem Zeitpunkt berechneten Radiosity-Werte als Lösung ausgegeben.

Für das beschriebene parallele Verfahren existieren zwei grundsätzlich verschiedene Ansätze zur Terminierung der parallelen Berechnungen (siehe auch 5.3.1.7). Eine Möglichkeit besteht darin, als Abbruchkriterium eine feste Anzahl von Iterationen vorzugeben, die durch das Progressive-Refinement-Verfahren durchgeführt werden sollen. Sobald diese Anzahl von Iterationen durchgeführt wurde, bricht das Verfahren ab. Ein anderer Ansatz besteht in der Vorgabe einer *Lösungsqualität*, die eine berechnete Radiosity-Lösung aufweisen soll. Die Qualität der Lösung wird hierbei in Abhängigkeit der in der Szene enthaltenen Lichtenergie spezifiziert. Hierbei bietet sich das Verhältnis von in der Szene enthaltener primärer Energie (spezifische Ausstrahlung von primären Lichtquellen) und der nach jeder Iteration des Progressive-Refinement-Verfahrens in der Szene insgesamt enthaltenen nicht verschossenen spezifischen Ausstrahlung (*Restenergie*) an. Je kleiner dieser Wert ist, desto weiter ist der Konvergenzprozeß fortgeschritten.

Der Vorteil des ersten Abbruchkriteriums ist dessen Einfachheit und der geringe zusätzliche Berechnungsaufwand (inkrementieren eines Zählers und eine Vergleichsoperation pro Iteration). Wenn parallele synchrone Progressive-Refinement-Berechnungen durchgeführt werden, stellt dieses Abbruchkriterium die einfachste Art der Ende-Erkennung dar, da keine zusätzliche Kommunikation zwischen den Prozessoren notwendig ist. Jeder Prozessor kann lokal die Anzahl synchronen Iterationen zählen und seine Berechnungen sofort beenden, sobald die vorgegebene maximale Anzahl ausgeführt wurde. Der gravierende Nachteil ist jedoch, daß der Anwender bei der Vorgabe einer maximalen Anzahl von durchzuführenden Iterationen nur eine vage Vorstellung von der Qualität der später berechneten Lösung hat.

Das zentrale Problem bei der Terminierung der asynchronen parallelen Berechnungen ist das verteilte Erkennen einer global konvergierten Radiosity-Lösung. In dem hier vorgestellten Verfahren wird das Verhältnis von noch zu verteilter Restenergiemenge in der Szene zu der initial in der Szene enthaltenen (primärer) Energiemenge verwendet, um zu bewerten, wie weit die aktuelle Lösung von der optimalen (konvergierten) Lösung entfernt ist. Im Fall von sequentiellen Berechnungen ist die Bestimmung dieses Verhältnisses sehr einfach, da während der Auswahl des nächsten Shooting-Patches, die gesamte nicht verschossene spezifische Ausstrahlung aller Patches summiert werden kann. Die primäre Lichtenergiemenge in der Szene wird während des Einlesens der Szenenbeschreibung vor Berechnungsbeginn bestimmt, indem die primären Energiemengen der Lichtquellenpatches addiert werden. Dieses ist bei datenparallelen Berechnungen nicht so einfach möglich, da jeder Prozessor nur auf einen Teil aller in der Szene vorhandenen Patches Zugriff hat. Somit können nur die nicht verschossenen spezifischen Ausstrahlungen der lokal verfügbaren Lichtpatches addiert werden. Um die in der Szene vorhandene globale Energiemenge bestimmen zu können, müssen die Summen der lokalen primären Energiewerte in einer Kommunikationsphase im Anschluß an das Einlesen der Szene zwischen den Prozessoren ausgetauscht und summiert werden. Ein weiteres Problem bei der parallelen Konvergenzbestimmung ist, daß bei asynchron ausgeführten Iterationen des Progressive-Refinement-Verfahrens die insgesamt auf einem Prozessor lokal verfügbare Restenergiemenge nur ein Teil der aktuell in der gesamten Szene global enthaltenen Menge an Restenergie ist.

Zur verteilten Erkennung der globalen Konvergenz besteht somit die Notwendigkeit, die Summe der lokalen Restenergiemengen aller Prozessoren asynchron zu bilden. Dieser Vorgang kann in den Vorgang des Austausches von Shooting-Patches zwischen den Prozessoren integriert werden. Jeder Prozessor verwaltet eine Liste, in der die aktuelle verbleibende Restenergie jedes einzelnen Prozessors gespeichert ist. Die Summe dieser Werte ist zu jedem Zeitpunkt die in der Szene global existierende Menge an Restenergie. Wenn von einem Prozessor eine Nachricht mit dem aktuellen lokalen Shooting-Patch an die übrigen Prozessoren verschickt wird (siehe 5.3.2.4), enthält diese Nachricht ebenfalls die Summe der zum Auswahlzeitpunkt lokal existierenden Restenergiemenge. Sobald diese Nachricht durch die anderen Prozessoren empfangen wird, können diese den entsprechenden Listeneintrag aktualisieren. Zusätzlich bestimmt ein Prozessor nach jeder externen Shooting-Iteration die jeweilige lokal vorhandene Restenergie und übermittelt diese an die übrigen Prozessoren mittels einer Broadcasting-Nachricht. Somit hat jeder Prozessor ein vollständiges Bild der energetischen Verhältnisse im System, d.h. den Konvergenzfortschritt bei den parallelen Berechnungen.

Ein Prozessor führt Shooting-Iterationen durch, bis er aufgrund seiner lokalen Listeneinträge eine globale Konvergenz erkannt hat, und keine externen Shooting-Patches mehr in der lokalen Warteschlange zur Bearbeitung anstehen. Ist dieses der Fall, wechselt der Prozessor seinen Berechnungsstatus von aktiv nach passiv. Mit Hilfe des Verfahrens zur Ende-Erkennung von Dijkstra [DiSo80] läßt sich sehr einfach überprüfen, ob sämtliche Prozessoren einen passiven Berechnungsstatus eingenommen haben. Sobald dieser Test positiv ausfällt, werden die parallelen Berechnungen beendet und die erzeugte Radiosity-Lösung ausgegeben.

5.3.2.7 Leistungsmessungen für statische Szenen

Zur Untersuchung der Skalierbarkeitseigenschaften des in Abschnitt 5.3 vorgestellten parallelen Progressive-Refinement-Verfahrens wurden Testmessungen in Szenen verschiedener Komplexität und mit variierender Anzahl von Prozessoren durchgeführt. Die Komplexität der Szenen bezieht sich hierbei auf die Anzahl der während der initialen Netzgenerierung erzeugten Patches und auf die Anzahl der primären Lichtquellen-Patches in einer Szene. Die in Abschnitt 5.3.1.1 beschriebenen Methoden zur Bestimmung von qualitativ guten Formfaktoren wurden beide bei den Messungen berücksichtigt und die erzielten Ergebnisse werden im weiteren Verlauf dieses Abschnitts gegenübergestellt.

Für die parallelen Verfahren existieren Einflußgrößen (z.B. Tiefe der Substrukturierung und Anzahl der Sample-Strahlen), welche die Berechnungszeiten und die Lösungsqualität mitunter erheblich beeinflussen. Ziel der durchgeführten Leistungsmessungen war es, die Auswirkungen dieser Einflußgrößen auf das Skalierbarkeitsverhalten der parallelen Radiosity-Berechnungen zu untersuchen. Bei sämtlichen durchgeführten Berechnungen wurde die Octree-Methode in Kombination mit Begrenzungsvolumen von Objekten und dem Shadow-Cache verwendet, um die Anzahl der Schnittpunkttests während der Formfaktorberechnungen zu reduzieren (siehe 5.3.1.3). Die verwendeten Methoden zur Formfaktorberechnung und die Substrukturierungskriterien bei der adaptiven Verfeinerung von Patches ermöglichen es, daß die Objektflächen der Szenen während der initialen Netzgenerierung in deutlich weniger Patches unterteilt werden müssen (und somit größer sind), als es bei der in Abschnitt 5.2 vorgestellten datenparallelen Methode notwendig war, um eine ausreichende Lösungsqualität zu erzielen.

Die Messungen wurden auf dem *hpcLine*-Parallelrechner von Fujitsu/Siemens durchgeführt. Dieses System verfügt über 96 Doppelprozessor-Knoten (2 PentiumII-Prozessoren mit 450

MHz Taktfrequenz), wobei jeder Knoten über 512 MByte Hauptspeicher verfügt. Das System ermöglicht es, zwischen zwei verschiedenen Kommunikationsnetzwerken unterschiedlicher Leistungsfähigkeit auszuwählen. Bei den in diesem Abschnitt präsentierten Meßergebnissen wurde Fast-Ethernet zur Kommunikation zwischen den Knoten benutzt. Der Parallelrechner wird unter dem Betriebssystem Linux (RedHat 6.2) betrieben, wodurch es möglich ist eine Standard-Version der PVM-Library zur Interprozeß-Kommunikation zu verwenden.

Zunächst wurden Messungen mit einer sehr einfachen Testszene durchgeführt. Hierbei wurde mit verschiedenen Anzahlen von Prozessoren jeweils eine zu 99% konvergierte Radiosity-Lösung berechnet. Dieses bedeutet, daß bei Abbruch der Berechnungen in der Szene nur noch 1% der nicht verschossenen spezifischen Ausstrahlung enthalten ist, die zu Berechnungsbeginn von den primären Lichtquellen in die Umgebung abgestrahlt wurde. Die Objektoberflächen der Szene *Büro* wurden initial in 1289 Patches unterteilt und lediglich eines dieser Patches ist eine primäre Lichtquelle. Die Szene stellt einen einfachen Büroraum dar, der mit einem Schreibtisch, einem Regal, zwei Tischen und einem Sessel möbliert ist. Für eine Szene mit so geringer Komplexität wie die Szene *Büro* kann mit dem datenparallelen Progressive-Refinement-Verfahren aus Abschnitt 5.2 eine Verkürzung der Berechnungszeiten auf maximal acht Prozessoren erzielt werden. Hierbei liegt die Effizienz der parallelen Berechnungen deutlich unter 40%. Die Ursache hierfür ist eine schlechte statische Lastverteilung. Bei sehr einfachen Szenen ist es sehr schwer, eine gleichmäßige geometrische Partitionierung der Szene in Teilszenen durchzuführen, so daß die Teilszenen annähernd die gleiche Anzahl von Patches enthalten. Dieses stellt jedoch bei der beschriebenen gleichmäßigen Verteilung benachbarter Patches auf verschiedene Prozessoren kein Problem dar. Somit sind auch effiziente parallele Berechnungen für sehr einfache Szenen auf deutlich mehr als acht Prozessoren möglich.

Zunächst sind parallele Berechnungen basierend auf der Methode von Wallace u.a. [WKEH89] (siehe Abschnitt 5.3.1.1) zur Bestimmung von Punkt-zu-Patch Formfaktoren durchgeführt worden. Die Sichtbarkeit der Shooting-Patches wurden bei der Bestimmung von Punkt-zu-Patch Formfaktoren jeweils bzgl. vier Sample-Punkten überprüft. Als Auswahlstrategie für die Shooting-Patches der lokalen Iterationen wurden sowohl die synchrone als auch die asynchrone Strategie verwendet (siehe 5.3.2.4). Es wurden Meßreihen durchgeführt, bei denen Patch-Hierarchien mit maximaler Tiefe 4 und maximaler Tiefe 6 erzeugt wurden. Die jeweils erzielten Berechnungszeiten sind Tabelle 5.23 in und Tabelle 5.24 angegeben.

Testszene <i>Büro</i>						
Synchrone Berechnungen mit 4 Sample-Punkten auf jedem Shooting-Patch						
	Maximale Tiefe einer Patch-Hierarchie beträgt 4			Maximale Tiefe einer Patch-Hierarchie beträgt 6		
Proz.	Zeit (h/m/s)	Speed-Up	Effizienz	Zeit (h/m/s)	Speed-Up	Effizienz
1	39m 21s	1,0	100,0	1h 25m 18s	1,0	100,0
2	22m 25s	1,8	87,8	46m 35s	1,8	91,6
4	12m 22s	3,2	79,6	25m 58s	3,3	82,1
8	7m 37s	5,2	64,6	15m 10s	5,6	70,3
12	5m 21s	7,4	61,3	10m 46s	7,9	66,0
16	4m 35s	8,6	53,7	9m 26s	9,0	56,5
24	3m 44s	10,5	43,9	7m 42s	11,1	46,2
32	-	-	-	6m 25s	13,3	41,5

Tabelle 5.23: Synchrone parallele Radiosity-Berechnungen auf Basis der Formfaktorapproximation nach Wallace u.a. mit unterschiedlichen adaptiven Substrukturierungstiefen der Patch-Hierarchien.

Testszene Büro						
Asynchrone Berechnungen mit 4 Sample-Punkten auf jedem Shooting-Patch						
Proz.	Maximale Tiefe einer Patch-Hierarchie beträgt 4			Maximale Tiefe einer Patch-Hierarchie beträgt 6		
	Zeit (h/m/s)	Speed-Up	Effizienz	Zeit (h/m/s)	Speed-Up	Effizienz
1	39m 21s	1,0	100,0	1h 25m 18s	1,0	100,00
2	21m 48s	1,8	90,3	46m 35s	1,8	91,6
4	11m 56s	3,3	82,4	25m 15s	3,4	84,5
8	6m 50s	5,8	72,0	14m 07s	6,0	75,5
12	4m 36s	8,6	71,3	9m 43s	8,8	73,2
16	3m 59s	9,9	61,7	7m 51s	10,9	67,9
24	3m 12s	12,3	51,2	6m 20s	13,5	56,1
32	-	-	-	5m 16s	16,2	50,6

Tabelle 5.24: Asynchrone parallele Radiosity-Berechnungen auf Basis der Formfaktorapproximation nach Wallace u.a. mit unterschiedlichen adaptiven Substrukturierungstiefen der Patch-Hierarchien.

An den Berechnungszeiten ist deutlich zu sehen, daß die adaptive Unterteilung von Patches einen erheblichen Einfluß auf die sequentiellen und parallelen Berechnungszeiten hat. Wenn die maximal zulässige Tiefe der Patch-Hierarchien bei den Testmessungen mit der Szene *Büro* um zwei vergrößert wird, bedeutet dieses eine Verdoppelung der Berechnungszeiten. Aufgrund der Erhöhung der lokalen Berechnungskomplexität auf den Prozessoren werden die parallelen Berechnungen für tiefere Patch-Hierarchien auf einer größeren Anzahl von Prozessoren effizienter ausgeführt.

Der bei den parallelen Berechnungen erzielte Speed-Up ist aus Gründen der Übersicht und der besseren Vergleichbarkeit in Abb. 5.46 graphisch dargestellt.

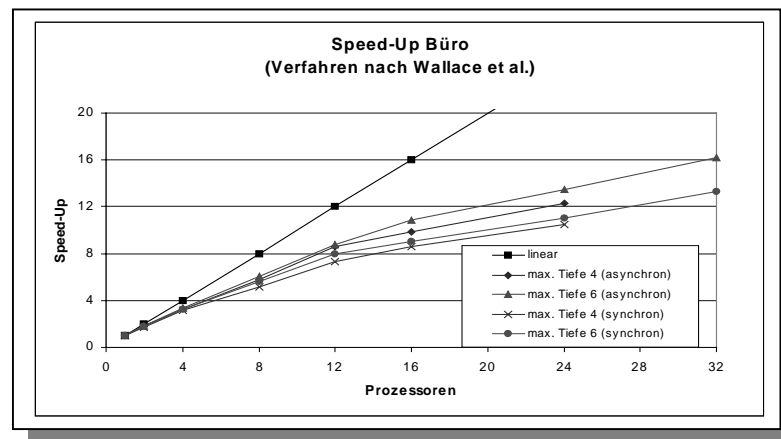


Abb. 5.46: Gegenüberstellung des erzielten Speed-Ups für unterschiedliche Auswahlstrategien von Shooting-Patches und unterschiedlichen Substrukturierungstiefen.

Der Abbildung kann entnommen werden, daß selbst für eine sehr einfache Szene ein Speed-Up noch auf bis zu 32 Prozessoren erzielt werden kann, wenn die lokale Berechnungskomplexität aufgrund der adaptiven Verfeinerung von Patches ausreichend groß ist. In diesem Fall ist die berechnete Radiosity-Lösung eine gute Approximation der tatsächlichen Beleuchtungsfunktionen der Objektflächen. Weiterhin ist deutlich ersichtlich, daß die asynchrone Auswahl von Shooting-Patches zu einer wesentlichen Verbesserung der Effizienz der parallelen Berechnungen führt. Die in Abb. 5.46 dargestellten Kurven weisen einen sehr gleichmä-

ßigen Verlauf auf und unterliegen nicht solchen Schwankungen, wie sie bei den Testmessungen für das in Abschnitt 5.2 beschriebene datenparallele Progressive-Refinement-Verfahren zu beobachten waren. Der Grund hierfür ist in der deutlich besseren statischen Lastverteilung zu suchen, die durch eine zufällige Verteilung der Patches und der Lichtquellen auf die Prozessoren erreicht wird.

Testszene Büro						
32 Sample-Punkte auf jedem Shooting-Patch und Empfänger-Element Maximale Tiefe einer Patch-Hierarchie beträgt 4						
Proz.	Asynchrone Berechnungen			Synchrone Berechnungen		
	Zeit (h/m/s)	Speed-Up	Effizienz	Zeit (h/m/s)	Speed-Up	Effizienz
1	32m 38s	1,0	100,0	32m 38s	1,0	100,0
2	16m 46s	1,9	96,8	17m 02s	1,9	95,3
4	8m 43s	3,7	93,1	9m 09s	3,6	88,7
8	4m 38s	7,0	87,6	5m 25s	6,0	74,9
16	2m 40s	12,2	76,1	3m 25s	9,5	59,4
24	1m 58s	16,5	68,8	2m 48s	11,6	48,3
32	1m 42s	19,1	59,7	2m 33s	12,7	39,8
40	1m 38s	19,9	49,7	2m 34s	12,7	31,6
48	1m 32s	21,2	44,1	2m 28s	12,3	25,7

Tabelle 5.25: Asynchrone und synchrone parallele Radiosity-Berechnungen auf Basis des Monte-Carlo-Verfahrens zur Bestimmung von Patch-zu-Patch Formfaktoren.

Es wurden ebenfalls die Zeiten für die parallele Berechnung einer Radiosity-Lösung auf Basis des Monte-Carlo-Verfahrens zur Bestimmung von Patch-zu-Patch Formfaktoren ermittelt. Hierbei sind sowohl auf den Shooting-Patches als auch auf den Empfänger-Elementen jeweils 32 Sample-Punkte zufällig plazierte worden. Diese große Anzahl von Sample-Punkten gewährleistet, daß die Approximation der Formfaktoren hinreichend genau ist, um ein bzgl. der subjektiven visuellen Qualität der Radiosity-Lösung gutes Ergebnis zu erzielen. In Tabelle 5.25 werden die Berechnungszeiten bei einer synchronen Auswahl von Shooting-Patches den Berechnungszeiten bei einer asynchronen Auswahl von lokalen Shooting-Patches gegenübergestellt. Eine Verbesserung der Berechnungszeiten bei den parallelen Berechnungen konnte auf bis zu 48 Prozessoren erzielt werden. Die sich aus den in Tabelle 5.25 angegebene Bezeiten ergebenden Speed-Ups sind in Abb. 5.47 graphisch gegenübergestellt.

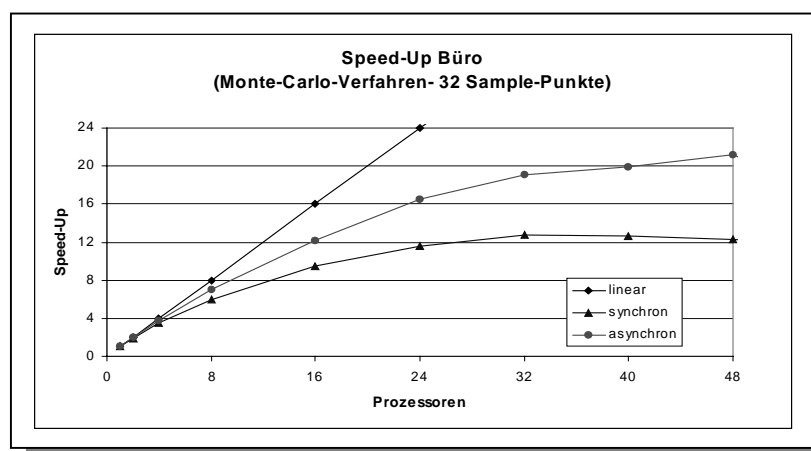


Abb. 5.47: Gegenüberstellung des Speed-Ups der parallelen Berechnungen auf Basis des Monte-Carlo-Verfahrens zur Formfaktorbestimmung bei synchroner und asynchroner Auswahl von Shooting-Patches.

Wie der Abbildung zu entnehmen ist, sind auch im Fall der Formfaktorapproximation auf Basis des Monte-Carlo-Verfahrens die asynchronen Berechnungen deutlich effizienter als die synchron ausgeführten Berechnungen. Die Kurvenverläufe unterliegen ebenfalls keinen deutlich sichtbaren Schwankungen.

Es ist offensichtlich, daß bei der Verwendung der Methode von Wallace u.a. zu Approximation von Formfaktoren die parallelen Berechnungen schlechter skalieren als bei der Formfaktorbestimmung auf Basis des Monte-Carlo-Verfahrens. Die Ursache hierfür liegt in den grundsätzlich unterschiedlichen Ansätzen dieser Verfahren begründet. Da die Methode von Wallace u.a. Eckpunkt-Radiosities berechnet, werden die Formfaktoren zwischen einem Shooting-Patch und den Eckpunkten einer gemeinsamen Kante von zwei benachbarten Patches doppelt berechnet, wenn die beiden Patches zwei verschiedenen Prozessoren zugeordnet sind. Dieses ist aufgrund der zufälligen Verteilung der Patches des Radiosity-Netzes auf die Prozessoren sehr wahrscheinlich. Wenn vier Patches an einen Eckpunkt angrenzen, wird ein identischer Formfaktor im schlimmsten Fall vierfach berechnet (durch vier verschiedene Prozessoren). Dieses muß geschehen, damit die Prozessoren unabhängig voneinander einen Substrukturierungstest durchführen können, der überprüft, ob die Radiosities entlang der Kanten eines Patches deutlich voneinander abweichen (siehe 5.3.1.4). Das Monte-Carlo-Verfahren berechnet Patch-zu-Patch Formfaktoren. Die Prozessoren können hierbei vollständig unabhängig voneinander Formfaktoren bestimmen, ohne daß Informationen auf verschiedenen Prozessoren mehrfach berechnet werden. Aus diesem Grund skalieren die parallelen Berechnungen im allgemeinen besser, wenn Patch-zu-Patch Formfaktoren mit dem Monte-Carlo-Verfahren bestimmt werden. Es ist jedoch zu beachten, daß die visuelle Qualität von Bildern, die auf Basis der durch das Verfahren von Wallace u.a. direkt berechneten Eckpunkt-Radiosities erzeugt werden, bei einer vergleichbaren Anzahl von verfolgten Sample-Strahlen (vergleichbarer lokaler Berechnungskomplexität) deutlich besser ist.

Aufgrund der Beobachtungen, die bei den Testmessungen für die einfache Szene *Büro* gemacht werden konnten, wurde bei allen weiteren Messungen die asynchrone Auswahlstrategie für Shooting-Patches verwendet und eine moderate maximale Substrukturierungstiefe (i.a. Tiefe 4) vorgegeben, um die Berechnungszeiten für sequentielle Referenzlösungen in einem akzeptablen Zeitrahmen zu halten.

Testszene Konferenzraum			
Asynchrone Berechnungen mit 1 Sample-Punkt auf jedem Shooting-Patch			
Proz.	Zeit (h/m/s)	Speed-Up	Effizienz
1	3h 33m 06s	1,0	100,0
4	1h 05m 13s	3,3	81,7
8	34m 36s	6,2	77,0
16	18m 22s	11,6	72,5
24	13m 26s	15,7	66,1
32	11m 04s	19,3	60,2
40	9m 40s	22,0	55,1
48	9m 11s	23,2	48,3

Tabelle 5.26: Parallele Radiosity-Berechnungen für die Testszene *Konferenzraum* auf Basis der Methode von Wallace u.a. zur Bestimmung von Formfaktoren.

Eine etwas komplexere Testszene ist durch einen Konferenzraum gegeben, der Bestandteil des Gebäudemodells *Soda Hall (Department of Computer Science - Berkley)* ist. Der Raum

ist mit einem langen Besprechungstisch und einem etwas kleineren Tisch sowie einer größeren Anzahl von Stühlen ausgestattet. Weiterhin sind zwei Objekte in der Szene enthalten, die Pflanzen repräsentieren und eine sehr hohe lokale Komplexität aufweisen. Der Raum wird von vier Deckenlichtquellen beleuchtet und die Objektoberflächen sind initial in 5190 Patches unterteilt worden. Wie schon zuvor bei der Szene *Büro* wird eine 99% konvergierte Radiosity-Lösung berechnet, wobei die Berechnungen mit beiden unterstützten Methoden zur Approximation von Formfaktoren durchgeführt wurden. Die ermittelten Berechnungszeiten sind in Tabelle 5.26 und Tabelle 5.27 zusammengefaßt.

Testszene Konferenzraum			
Asynchrone Berechnungen mit 16 Sample-Punkten auf jedem Shooting-Patch und Empfänger-Element			
Proz.	Zeit (h/m/s)	Speed-Up	Effizienz
1	8h 30m 53s	1,0	100,0
2	4h 34m 21s	1,9	93,1
4	2h 18m 16s	3,7	92,4
10	57m 57s	8,8	88,2
20	30m 41s	16,7	83,3
30	21m 43s	23,5	78,4
40	18m 49s	27,2	67,9
50	16m 09s	31,6	63,3
60	14m 17s	35,8	59,6
70	13m 44s	37,2	53,1
80	12m 44s	40,1	50,2

Tabelle 5.27: Parallele Radiosity-Berechnungen für die Testszene *Konferenzraum* auf Basis des Monte-Carlo-Verfahrens zur Bestimmung von Formfaktoren.

Der bei den parallelen Berechnungen mit der Testszene *Konferenzraum* erzielte Speed-Up ist in Abb. 5.48 graphisch dargestellt.

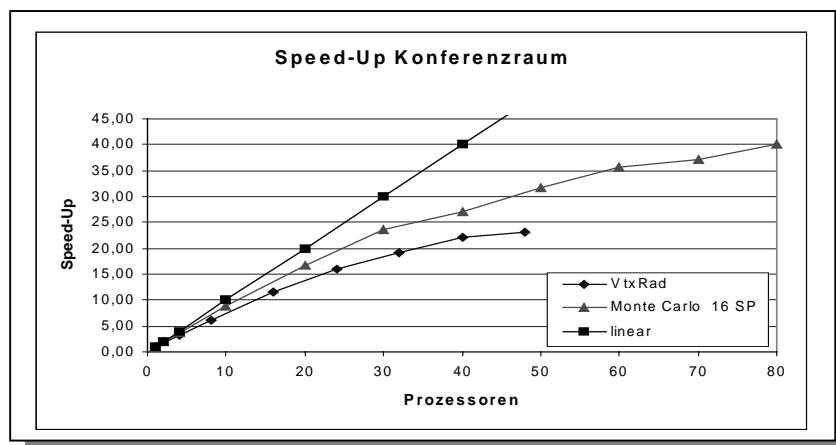


Abb. 5.48: Speed-Up für die Testszene *Konferenzraum*.

Es ist deutlich zu sehen, daß in beiden Fällen auf einer größeren Anzahl von Prozessoren eine Reduktion der Berechnungszeiten möglich ist, als es bei der einfachen Testszene *Büro* der Fall war. Bei den Berechnungen mit der Szene *Konferenzraum* werden aufgrund der Eingabeparameter durch beide Methoden zur Approximation eines Formfaktors weniger Sample-

Strahlen generiert als bei den Berechnungen für die erste Testszene. Das die parallelen Berechnungen trotzdem besser skalieren ist dadurch zu erklären, daß die lokale Berechnungskomplexität auf den Prozessoren durch die größere Anzahl der initial zugewiesenen Patches vergrößert wird. Somit wird die Verringerung der lokalen Berechnungskomplexität amortisiert, die durch Reduktion der Anzahl der generierten Sample-Strahlen entsteht. Dieses läßt den Schluß zu, daß die parallelen Berechnungen mit steigender Komplexität der Szenen bei gleicher Qualität der Formfaktorapproximation besser skalieren. Dieses wird auch durch die Meßergebnisse belegt, die für komplexere Szenen ermittelt wurden und weiter unten präsentiert werden.

Eine Testszene, die schon zur Untersuchung der Skalierbarkeitseigenschaften des datenparallelen Progressive-Refinement-Verfahrens in 5.2 verwendet wurde, ist die Szene *Townhouse*. Für die hier durchgeführten parallelen Radiosity-Berechnungen wurden die Objektoberflächen initial in 29957 Patches unterteilt. Die Szene wird durch 19 primäre Lichtquellenpatches beleuchtet. Es wurde jeweils eine zu 99% konvergierte Radiosity-Lösung bestimmt. Die initial erzeugten Patches wurden im Verlauf der Berechnungen durch adaptive Verfeinerung in 324990 Elemente auf der untersten Ebene der Patch-Hierarchien verfeinert.

Mit Hilfe der Testmessungen auf Grundlage der Eingabeszene *Townhouse* sollen die Auswirkungen demonstriert werden, die eine Vergrößerung der lokalen Berechnungskomplexität durch variieren der Anzahl von Sample-Strahlen auf die Skalierbarkeit des parallelen Verfahrens hat. Zu diesem Zweck wurden für beide unterstützten Methoden zur Formfaktorapproximation mehrere Meßreihen mit unterschiedlichen Anzahlen von Sample-Strahlen durchgeführt.

Wenn während der parallelen Berechnungen eine Radiosity-Lösung auf Basis der Berechnung von Punkt-zu-Patch Formfaktoren nach der Methode von Wallace u.a. bestimmt wird, kann eine erhebliche Verkürzung der Berechnungszeit auf bis zu 96 Prozessoren erzielt werden (siehe Tabelle 5.28).

Testszene <i>Townhouse</i>									
Asynchrone Berechnungen mit verschiedenen Anzahlen von Sample-Punkten auf jedem Shooting-Patch									
Proz.	1 Sample-Punkt			4 Sample-Punkte			6 Sample-Punkte		
	Zeit (h/m/s)	Sp.-Up	Eff.	Zeit (d/h/m/s)	Sp.-Up	Eff.	Zeit (d/h/m/s)	Sp.-Up	Eff.
1	13h 49m 01s	1,0	100,0	2d 20h 28m 20s	1,0	100,0	3d 19h 19m 17s	1,0	100,0
2	8h 02m 16s	1,7	85,9	1d 15h 31m 23s	1,7	86,6	2d 02h 29m 45s	1,8	90,4
4	4h 13m 21s	3,3	81,8	20h 21m 22s	3,4	84,1	1d 02h 24m 17s	3,5	86,5
8	2h 13m 29s	6,2	77,6	10h 46m 04s	6,4	79,5	14h 01m 18s	6,5	81,4
16	1h 11m 08s	11,7	72,8	5h 25m 33s	12,6	78,9	7h 22m 34s	12,4	77,4
20	56m 52s	14,6	72,9	4h 38m 54s	14,7	73,6	5h 54m 01s	15,5	77,4
24	48m 19s	17,2	71,5	3h 49m 04s	17,9	74,7	5h 01m 16s	18,2	75,8
32	39m 31s	21,0	65,6	3h 02m 58s	22,5	70,2	3h 48m 30s	24,0	74,9
40	34m 13s	24,2	60,6	2h 19m 26s	29,5	73,7	3h 06m 40s	29,4	73,4
48	28m 54s	28,7	59,8	2h 03m 04s	33,4	69,6	2h 38m 06s	34,7	72,2
56	26m 20s	31,5	56,2	1h 46m 30s	38,6	68,9	2h 18m 53s	39,5	70,5
64	25m 49s	32,1	50,2	1h 42m 55s	39,9	62,4	2h 14m 16s	40,8	63,8
72	23m 19s	35,6	49,4	1h 32m 53s	44,2	61,4	2h 00m 29s	45,5	63,2
80	21m 47s	38,1	47,6	1h 24m 16s	48,8	60,9	1h 48m 04s	50,7	63,4
88	22m 52s	36,3	41,2	1h 17m 45s	52,8	60,1	1h 42m 41s	53,4	60,7
96	21m 29s	38,6	40,2	1h 16m 11s	53,9	56,2	1h 34m 22s	58,1	60,5

Tabelle 5.28: Parallele Radiosity-Berechnungen für die Testszene *Townhouse* auf Basis der Methode von Wallace u.a. zur Bestimmung von Punkt-zu-Patch Formfaktoren.

Wie Abb. 5.49 zu entnehmen ist, werden die Skalierbarkeitseigenschaften des Verfahrens deutlich besser, wenn die lokale Berechnungskomplexität durch Erhöhen der Anzahl zu generierender Sample-Strahlen vergrößert wird. So kann z.B. das Speed-Up bei parallelen Berechnungen auf 96 Prozessoren von 38,6 auf 58,1 verbessert werden, wenn statt einem Sample-Punkt auf jedem Shooting-Patch sechs Sample-Punkte platziert werden, um die Approximationsgenauigkeit der Formfaktoren zu steigern. Dieses entspricht einer Verbesserung der Effizienz um 20%.

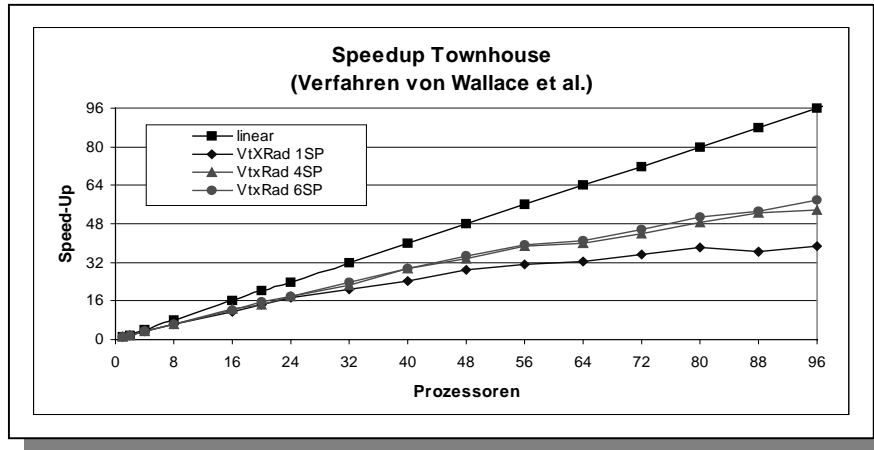


Abb. 5.49: Speed-Up der parallelen Radiosity-Berechnungen auf Basis der Methode von Wallace u.a. für die Testszene *Townhouse* mit variabler Anzahl von Sample-Strahlen.

In Tabelle 5.29 sind die Berechnungszeiten zusammengefasst, die benötigt wurden, wenn das Monte-Carlo-Verfahren zu Bestimmung von Patch-zu-Patch Formfaktoren verwendet wird.

Testszene <i>Townhouse</i>									
Asynchrone Berechnungen mit verschiedenen Anzahlen von Sample-Punkt auf jedem Shooting-Patch und Empfänger-Element									
Proz.	16 Sample-Punkte			24 Sample-Punkte			32 Sample-Punkte		
	Zeit (d/h/m/s)	Sp.-Up	Eff.	Zeit (d/h/m/s)	Sp.-Up	Eff.	Zeit (d/h/m/s)	Sp.-Up	Eff.
1	2d 4h 41m 46s	1,0	100,0	3d 13h 50m 29s	1,0	100,0	5d 00h 20m 14s	1,0	100,0
2	1d 4h 54m 27s	1,8	91,2	1d 22h 07m 18s	1,9	93,1	2d 16h 23m 05s	1,9	93,5
4	14h 19m 07s	3,7	92,0	23h 04m 52s	3,7	93,0	1d 07h 47m 14s	3,8	94,6
8	7h 11m 16s	7,3	91,6	11h 33m 11s	7,4	92,9	16h 29m 59s	7,3	91,2
16	3h 45m 58s	14,0	87,5	5h 55m 39s	14,5	90,5	8h 27m 28s	14,2	88,9
20	3h 06m 16s	17,0	84,9	4h 49m 49s	17,8	88,9	6h 41m 16s	18,0	90,0
24	2h 33m 50s	20,6	85,6	3h 59m 48s	21,5	89,5	5h 41m 54s	21,1	88,0
32	1h 58m 10s	26,8	83,6	2h 38m 44s	28,8	90,1	4h 18m 38s	27,9	87,2
40	1h 36m 49s	32,7	81,6	2h 29m 45s	34,4	86,0	3h 26m 34s	35,0	87,4
48	1h 26m 51s	36,4	75,8	2h 09m 24s	39,8	83,0	2h 55m 29s	41,1	85,7
56	1h 16m 03s	41,6	74,2	1h 52m 53s	45,6	81,5	2h 34m 59s	46,6	84,0
64	1h 06m 36s	47,5	74,2	1h 42m 48s	50,1	78,3	2h 24m 39s	49,9	78,0
72	1h 01m 15s	51,6	71,7	1h 35m 12s	54,1	75,1	2h 04m 43s	57,9	80,4
80	56m 34s	55,9	69,9	1h 24m 36s	60,9	76,1	2h 00m 01s	60,2	75,2
88	54m 53s	57,6	65,5	1h 23m 11s	61,9	70,4	1h 49m 11s	66,1	75,2
96	52m 51s	59,8	62,3	1h 20m 54s	63,7	66,3	1h 41m 21s	71,2	74,2
104	52m 15s	60,5	58,2	1h 13m 08s	70,4	67,7	1h 37m 38s	74,0	71,1
112	-	-	-	1h 08m 43s	75,0	66,9	1h 32m 40s	78,0	69,6

Tabelle 5.29: Parallele Radiosity-Berechnungen für die Testszene *Townhouse* auf Basis des Monte-Carlo-Verfahrens zur Bestimmung von Patch-zu-Patch Formfaktoren.

Eine erhebliche Reduktion der Berechnungszeiten wurde auf bis zu 112 Prozessoren erzielt. Wiederum skalieren die parallelen Berechnungen deutlich besser, wenn die Approximationsgenauigkeit der Formfaktoren erhöht wird, indem eine größere Anzahl von Sample-Punkten auf Shooting-Patches und Empfängererelementen plazierte werden. Wenn 32 Sample-Strahlen pro zu berechnenden Formfaktor erzeugt werden, wird selbst auf 112 Prozessoren noch eine Effizienz von annähernd 70% erzielt.

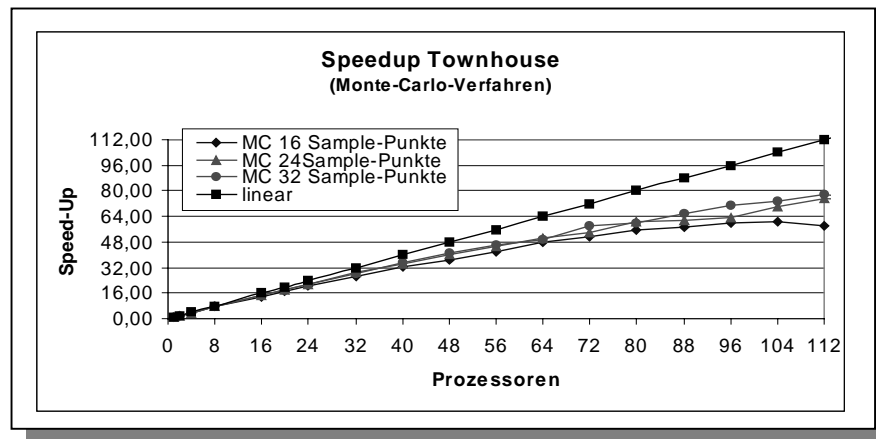


Abb. 5.50: Speed-Up der parallelen Radiosity-Berechnungen auf Basis des Monte-Carlo-Verfahrens zur Bestimmung von Formfaktoren für die Testszene *Townhouse* mit variabler Anzahl von Sample-Strahlen.

Die in diesem Abschnitt präsentierten Meßergebnisse demonstrieren deutlich die guten Skalierbarkeitseigenschaften des neuen parallelen Progressive-Refinement-Verfahrens. Aufgrund der statischen Lastverteilung durch zufällige Platzierung von Patches auf die Prozessoren und dem geringen Kommunikations-Overhead des Verfahrens können effiziente parallele Radiosity-Berechnungen selbst für Eingabeszenen von geringer Komplexität noch auf bis zu 40 Prozessoren ausgeführt werden. Für komplexe Eingabeszenen können die Berechnungszeiten für eine qualitativ gute Radiosity-Lösung noch auf mehr als 100 Prozessoren deutlich verkürzt werden.

Wie die Meßergebnisse deutlich aufzeigen, existieren mehrere Einflußfaktoren, die sowohl die Effizienz der parallelen Berechnungen als auch die Qualität der Radiosity-Lösung beeinflussen. Diese Einflußfaktoren sind

- die maximale Substrukturierungstiefe, die bei der adaptiven Unterteilung von Patches zugelassen wird
- und die Anzahl der Sample-Strahlen, die zur Überprüfung der relativen Sichtbarkeit von Sample-Punkten bei der Formfaktorberechnung generiert werden.

Für das in 5.3 eingeführte parallele Progressive-Refinement-Verfahrens gilt generell, daß die Effizienz der parallelen Berechnungen besser wird, wenn eine hohe Lösungsqualität gefordert, d.h. die Anzahl der generierten Sample-Strahlen erhöht und die Substrukturierungstiefe vergrößert werden muß. Der Grund hierfür ist, daß die lokale Berechnungskomplexität auf den Prozessoren des Parallelrechners vergrößert wird, ohne daß der Kommunikationsaufwand bei den parallelen Berechnungen signifikant zunimmt. Die lokale Berechnungskomplexität wird ebenfalls vergrößert, wenn komplexe Eingabeszenen verwendet werden. In diesem Fall ist auch die initiale Last (Anzahl der Patches), die einem Prozessor zugewiesen wird, größer als bei kleineren Szenen. Hierdurch vergrößert sich indirekt die Anzahl der generierten

Sample-Strahlen und die Anzahl der durchgeführten adaptiven Verfeinerungen von Patches. Somit skaliert das parallele Verfahren insbesondere für komplexe Eingabeszenen sehr gut.

Eine weitere erhebliche Einflußgröße auf die Effizienz der parallelen Progressive-Refinement-Berechnungen ist die Auswahlstrategie der Shooting-Patches. Die asynchrone Auswahlstrategie hat sich hierbei als deutlich effizienter erwiesen als die synchrone Auswahl von Shooting-Patches. Dieses gilt insbesondere für große Anzahlen von Prozessoren, da die Wartezeiten der Prozessoren aufgrund schwankender Iterationszeiten auf den Prozessoren sehr groß werden und ein zusätzlicher Mehraufwand für die Synchronisation der Berechnungsprozesse entsteht.

Die getroffenen Aussagen bzgl. der Einflußfaktoren auf die Effizienz der parallelen Berechnungen gelten für beide unterstützte Verfahren zur Bestimmung von Formfaktoren bei den parallelen Berechnungen verwendet wird. Im allgemeinen ist jedoch die Effizienz der parallelen Berechnungen auf Basis des Monte-Carlo-Verfahrens zur Formfaktorapproximation gerade für eine größere Anzahl von Prozessoren besser als die parallelen Berechnungen von Punkt-zu-Patch Formfaktoren.

Der gewählte datenparallele Ansatz ermöglicht die Berechnung von Radiosity-Lösungen für extrem komplexe Eingabeszenen. In diesem Abschnitt wurden Testmessungen nur für Szenen und Parametereinstellungen durchgeführt, bei denen sich der sequentielle Berechnungsaufwand noch in einem vertretbaren Rahmen bewegt. Für komplexere Eingabeszenen ist die sequentielle Berechnung einer Referenzlösung nicht in annehmbarer Zeit möglich und der lokale Speicher eines Prozessors reicht nicht aus, um die adaptiv verfeinerte Radiosity-Lösung im Hauptspeicher aufzunehmen. In diesem Fall verfälschen die Zeiten, die zur Auslagerung von Daten auf ein externes Speichermedium benötigt werden (Swapping), die Berechnungszeiten erheblich.

5.3.3 Parallele Radiosity-Berechnung in dynamischen Szenen

Das in 5.3.2 beschriebene parallele Progressive-Refinement-Verfahren kann in seiner bisher beschriebenen Form lediglich eine Radiosity-Lösung für statische Eingabeszenen berechnen. Sobald eine Modifikation der Szene vorgenommen wird, muß der vollständige Lösungsprozeß erneut durchgeführt werden. In Kapitel 4 wurde gezeigt, daß ein Progressive-Refinement-Verfahren in ein inkrementelles Radiosity-Verfahren überführt werden kann. Das inkrementelle Radiosity-Verfahren ermöglicht eine schnelle Berechnung von vollständig konvergier-ten Radiosity-Lösungen in dynamisch veränderbaren Umgebungen, indem eine lokale Korrektur der bis zum Zeitpunkt der Modifikation berechneten Lösung vorgenommen wird.

Ein datenparalleles inkrementelles Radiosity-Verfahren, das auf der Methode von Chen [Chen90] basiert, wurde in Abschnitt 5.2.7 vorgestellt. Die Berechnungszeiten für eine aktualisierte Radiosity-Lösung konnten durch die parallelen Berechnungen erheblich verkürzt werden (siehe 5.2.7.2). Wenn einzelne Modifikationen der Szene vorgenommen werden, die nur lokal beschränkte Auswirkungen auf die globale Beleuchtung der gesamten Szene haben (z.B. verschieben eines Objektes innerhalb einer Teilszene), führt eine geometrische Partitionierung der Szene in zusammenhängende Regionen (Teilszenen) für größere Anzahlen von Prozessoren zu einer geringen Effizienz der parallelen Berechnungen. Die Ursache hierfür ist, daß zuvor berechnete Radiosity-Werte lediglich in lokal begrenzten Regionen korrigiert werden müssen. Somit ist an den inkrementellen Berechnungen im Anschluß an eine Modifikation im allgemeinen nur ein kleiner Teil der insgesamt eingesetzten Prozessoren beteiligt. Dieses Problem kann umgangen werden, indem eine statische Lastverteilung vorgenommen wird, die auf

der gleichmäßigen Verteilung der Patches einer Szene auf die Prozessoren basiert. Die Patches, für die im Anschluß an eine Modifikation der Szene Korrekturwerte berechnet werden müssen, sind selbst bei lokal begrenzten Auswirkungen der Interaktion auf die globale Beleuchtung mit großer Wahrscheinlichkeit auf verschiedene Prozessoren verteilt. Die dynamische Verteilung der Last ist hierdurch während der parallelen Berechnung einer inkrementell korrigierten Radiosity-Lösung wesentlich besser als bei einer geometrischen Partitionierung der Eingabeszene.

Das in 5.3.2 beschriebene parallele Progressive-Refinement-Verfahren ist derart konzipiert worden, daß es vorzugsweise in Kombination mit einer statischen Lastverteilung zu verwenden ist, die auf einer Verteilung benachbarter Patches des Radiosity-Netzes auf verschiedene Prozessoren basiert. Dieses Verfahren kann sehr einfach in ein inkrementelles Radiosity-Verfahren zur effizienten parallelen Berechnung von Radiosity-Lösungen in dynamisch veränderbaren Szenen umgewandelt werden. Die grundlegende Vorgehensweise bei den parallelen Berechnungen bleibt hierbei unverändert. Insbesondere die statische Lastverteilung sowie die Kommunikationsmechanismen (asynchrone vs. synchrone verteilte Auswahl von Shooting-Patches) bleiben unverändert.

Die Prozessoren haben die Aufgabe korrigierte Energieverteilungen für die jeweils lokal gespeicherten Patch-Hierarchien gemäß der in 4.3 beschriebenen inkrementellen Methode im Anschluß an Modifikationen der Szenenbeschreibung zu berechnen. Um eine inkrementelle Korrektur einer Radiosity-Lösung durchzuführen, werden durch die Prozessoren analog zu den parallelen Berechnungen in statischen Szenen lokale und externe Shooting-Patches bearbeitet. Hierbei sind einige Veränderungen des parallelen Verfahrens notwendig, um Radiosity-Lösungen in dynamisch veränderbaren Umgebungen zu berechnen. Diese Veränderungen betreffen die Auswahl von lokalen Shooting-Patches, die lokale Anpassung von Energiewerten nach der Modifikation von Oberflächeneigenschaften, die Berechnung von inkrementellen Formfaktoren, die Octree-Traversierung (und der entsprechenden Datenstruktur) sowie die Ende-Erkennung.

5.3.3.1 Anpassung des Radiosity- und des Intersection-Netzes

Während der parallelen Radiosity-Berechnungen werden zwei verschiedene Szenenrepräsentationen verwendet, ein Intersection-Netz und ein Radiosity-Netz (siehe 5.3.1.3). Bei allen Interaktionen, welche die Geometrie der Szene betreffen, müssen somit beide Repräsentationen der Szene aktualisiert werden. Für jedes neue Objekt wird sowohl eine Repräsentation des Objektes in Form von kleineren dreieckigen oder viereckigen Flächenelementen (*Radiosity-Objekt*) als auch eine zusätzliche Repräsentation in Form von großen Polygonen (*Intersection-Objekt*) generiert. Nach dem Einfügen eines Objektes in die Szene wird eine gleichmäßige Verteilung der Patches des zugehörigen Radiosity-Objektes auf die Prozessoren bestimmt. Gemäß dieser Verteilung werden die Patches des Radiosity-Objektes den Prozessoren mit Hilfe von Nachrichten übermittelt. Die Empfängerprozessoren fügen die zugeteilten Patches in die lokale Datenstruktur zur Speicherung des Radiosity-Netzes ein. Das entsprechende Intersection-Objekt des eingefügten Objektes wird allen Prozessoren mittels einer Broadcasting-Nachricht bekannt gemacht. Jeder Prozessor fügt das empfangene Intersection-Objekt nach der Berechnung eines Begrenzungskörpers in das lokal gespeicherte Intersection-Netz ein. Bei Verwendung der Octree-Methode zur Beschleunigung der Schnittpunkttests (siehe 5.3.1.3) muß das Intersection-Objekt jedes neu eingefügte Objekt in die Octree-Datenstruktur übernommen werden. Somit enthält die Octree-Datenstruktur alle Versionen der Szene seit dem Start der Berechnungen bis zum aktuellen Zeitpunkt. Dieses ist notwendig, um korrekte in-

krementelle Formfaktoren bzgl. aller vorgenommenen Modifikationen der ursprünglichen Szenengeometrie berechnen zu können.

Wenn ein Objekt der Szene gelöscht wird, dürfen die entsprechenden Repräsentationen des Objektes weder aus dem Radiosity-Netz noch aus dem Intersection-Netz entfernt werden, da diese Daten noch zur Durchführung von Korrekturberechnungen benötigt werden. Für ein gelöscht Objekt wird lediglich der Löszeitpunkt vermerkt. Dieser wird durch die Anzahl von Szenenmodifikationen spezifiziert, die seit dem Start der Berechnungen bis zum Löschen des Objektes ausgeführt wurden. Mit Hilfe dieser Information und der Geometry-Queue (siehe 4.3.4) können während der Berechnung von inkrementellen Formfaktoren verschiedene Versionen der Szene rekonstruiert werden. Eine lokale Geometry-Queue wird analog zu dem in 5.2.7 beschriebenen parallelen inkrementellen Radiosity-Verfahren durch jeden Prozessor verwaltet.

Bei der Veränderung von Objekteigenschaften besteht keine Notwendigkeit das Intersection-Netz zu aktualisieren. Es wird lediglich eine Broadcasting-Nachricht an alle Prozessoren verschickt, welche die eindeutige Objektnummer des veränderten Objektes und eine Beschreibung der Veränderung (Farbe, spezifische Ausstrahlung) enthält. Alle Prozessoren überprüfen, ob sie Patches dieses Objektes lokal speichern. Ist dieses der Fall, werden die Materialeigenschaften des entsprechenden Objektes in der Datenstruktur des Radiosity-Netzes angepaßt. Unmittelbar im Anschluß daran werden die notwendigen inkrementellen Korrekturen gemäß 4.3.1 zunächst lokal ausgeführt und dann die lokalen Progressive-Refinement-Berechnungen fortgesetzt. Eine Verteilung der berechneten Korrekturwerte innerhalb der gesamten Szene erfolgt automatisch im weiteren Verlauf der Berechnungen, wodurch auch die indirekten Beleuchtungseffekte an die veränderten Gegebenheiten angepaßt werden.

5.3.3.2 Auswahl von Shooting-Patches

Die parallele Bestimmung von Shooting-Patches während der inkrementellen Berechnungen wird gemäß der in 5.3.2.4 beschriebenen synchronen oder asynchronen Methode durchgeführt. Die jeweilige lokale Auswahl von Shooting-Patches erfolgt hierbei gemäß der in 4.3.5 beschriebenen Vorgehensweise. Bei dem Austausch von Shooting-Patches zwischen den Prozessoren wird in jeder Nachricht mitgeteilt, ob es sich bei der auszuführenden Iteration um eine *Redistribution*- oder eine *Propagation-Iteration* handelt. Eine Propagation-Iteration ist eine herkömmliche Iteration des Progressive-Refinement-Verfahrens, die grundsätzlich bzgl. der vollständigen aktuellen Szene ausgeführt wird. Bei einer Redistribution-Iteration handelt es sich um eine Iteration, in der eine Umverteilung der Energie vorgenommen wird, die schon zuvor von dem entsprechenden Shooting-Patch bzgl. einer älteren Version der Szene verteilt wurde. Je nach Art der Szenenmodifikation müssen bei einer Redistribution-Iteration inkrementelle Formfaktoren berechnet werden (siehe 4.3.2), sobald ein dynamisches Objekt (d.h. eingefügtes oder gelöscht Objekt) die relative Sichtbarkeit zwischen einem Shooting-Patch und einem Empfänger-Element oder Eckpunkt beeinträchtigt.

5.3.3.3 Berechnung inkrementeller Formfaktoren

Die in 5.2.7 vorgestellte parallele Methode verwendet verschiedene Hemicubes zur Bestimmung von inkrementellen Formfaktoren, wobei der statische und der dynamische Teil der Szene auf verschiedene Hemicubes projiziert wird. Inkrementelle Sichtbarkeitsbeziehungen werden durch einen Vergleich der Inhalte der Hemicube-Pixel im Anschluß an die Projektionen ermittelt. Die hier beschriebene Methode basiert auf vollkommen unterschiedlichen Ansätzen zur Approximation von Formfaktoren, bei denen eine Strahlverfolgungstechnik ver-

wendet wird, um relative Sichtbarkeitsbeziehungen auszuwerten (siehe 5.3.1.3 und 5.3.2.5). Hierbei können räumliche Kohärenzeigenschaften ausgenutzt werden, um die Anzahl der zu bestimmenden inkrementellen Formfaktoren zu reduzieren [Chen90].

Ein inkrementeller Formfaktor muß nur dann berechnet werden, wenn das Empfänger-Patch ganz oder teilweise innerhalb des Schattenvolumens liegt, das auf Basis der Position des Shooting-Patch und des dynamischen Objektes der aktuellen Redistribution-Iteration bestimmt wird (siehe 4.3.2). In diesem Fall muß eine Korrektur der Energiemenge vorgenommen werden, die in einer vorangegangenen Propagation-Iteration von dem Shooting-Patch an das Empfänger-Patch abgegeben wurde, da sich aufgrund der Szenenmodifikation veränderte relative Sichtbarkeitsverhältnisse zwischen Shooting-Patch und den Oberflächen des statischen Teils der Szene ergeben haben.

Der zentrale Aspekt bei den verwendeten Methoden zur Bestimmung von Formfaktoren ist die Auswertung von relativen Sichtbarkeitsbeziehungen zwischen Sample-Punkten auf dem jeweiligen Shooting-Patch und einem Sample-Punkt auf einem Empfänger-Patch bzw. einem Eckpunkt des Radiosity-Netzes. Der endgültige Formfaktor setzt sich aus einer Summe von einfacheren Formfaktoren zusammen, die zwischen Sample-Punkten auf den Shooting- und den Empfänger-Patches ausgewertet werden (3.1.2.5 und 3.1.2.6). Bei der Approximation von Formfaktoren in dynamisch veränderbaren Szenen muß nun eine durch die Szenenmodifikation verursachte Veränderung der Sichtbarkeitsbeziehungen zwischen einem Paar von Sample-Punkten erkannt und durch Bestimmung inkrementeller Formfaktoren berücksichtigt werden. Für einen Sample-Strahl können nur drei verschiedene Fälle eintreten:

- Der Sample-Strahl schneidet auf der Strecke zwischen den entsprechenden Sample-Punkten weder ein statisches noch das dynamische Objekt. Die Sample-Punkte sind somit auch in der dynamisch veränderten Szene gegenseitig sichtbar.
- Der Sample-Strahl schneidet auf der Strecke zwischen den entsprechenden Sample-Punkten das dynamische und mindestens ein statisches Objekt. Die Sample-Punkte sind somit auch in der dynamisch veränderten Szene gegenseitig nicht sichtbar.
- Der Sample-Strahl schneidet auf der Strecke zwischen den entsprechenden Sample-Punkten ausschließlich das dynamische Objekt. Bei einer Einfüge-Operation sind die Sample-Punkte somit in der dynamisch veränderten Szene nicht mehr gegenseitig sichtbar, während sie bei einer Löschoption in der dynamisch veränderten Szene gegenseitig sichtbar werden.

In den ersten beiden Fällen verursacht das dynamische Objekt keine Veränderung der relativen Sichtbarkeit zwischen den beiden Sample-Punkten in der dynamisch veränderten Szene. Lediglich in dem letzten Fall hat die Szenenmodifikation Auswirkungen auf die relative Sichtbarkeit der entsprechenden Sample-Punkte. Je nach Art der Modifikation, muß der inkrementelle Formfaktor um den Anteil, den der einfache Punkt-zu-Punkt Formfaktor zwischen den beiden Sample-Punkten an dem endgültigen Formfaktor hat (siehe 3.1.2.5 und 3.1.2.6), dekrementiert (einfügen eines Objektes) oder inkrementiert (löschen eines Objektes) werden.

Es ist zu beachten, daß bei der Strahlverfolgung vor jedem Schnittpunkttest mit einem Objekt des Intersection-Netzes überprüft werden muß, ob das Objekt Teil der Szene ist, bzgl. der die Energie des aktuellen Shooting-Patches verteilt bzw. umverteilt werden muß. Dieses ist notwendig, da in den verwendeten Datenstrukturen zur Beschleunigung der Schnittpunkttests die

Objekte aller Versionen der Szene seit Beginn der Berechnungen enthalten sind. Ein solcher Test ist sehr einfach zu realisieren, indem der Einfüge- und der Löszeitpunkt eines Objektes mit dem Geometry-Queue-Zähler des aktuellen Shooting-Patches verglichen wird. Nur wenn der Geometry-Queue-Zähler in dem durch Einfüge- und Löszeitpunkt des Objektes definierten Intervall liegt, wird ein Schnittpunkttest mit dem Objekt durchgeführt.

Im Gegensatz zu der in 5.2.7 beschriebenen Methode wird durch die Berechnung von inkrementellen Formfaktoren kein zusätzlicher Kommunikationsaufwand erzeugt, da die Auswertung der relativen Sichtbarkeitsbeziehungen zwischen Sample-Punkten durch Verwendung des Intersection-Netzes vollständig ohne Kommunikation erfolgt.

5.3.3.4 Adaptive Verfeinerung von Patches

Ebenso wie bei der Berechnung einer Radiosity-Lösung für eine statische Szene wird auch bei den inkrementellen Radiosity-Berechnungen eine adaptive Verfeinerung von Patches vorgenommen. Ein gewisses Problem bei den inkrementellen Berechnungen besteht darin, adaptive Verfeinerungen von Patches rückgängig zu machen. Wenn sich die Beleuchtungsfunktionen von Patches aufgrund einer Modifikation der Szene verändern, kann es durchaus sein, daß in gewissen Regionen der Szene keine adaptive Substrukturierung mehr notwendig ist. Dieses ist zum Beispiel der Fall, wenn ein Objekt aus der Szene entfernt wird, welches Schatten auf andere Objekte wirft. Um eine fehlerfreie Korrektur von Radiosity-Werten innerhalb von Patch-Hierarchien zu ermöglichen (und somit auch die Rücknahme von adaptiven Verfeinerungen), muß während einer Redistribution-Iteration die gleiche Entscheidungsgrundlage bei den Substrukturierungstests gegeben sein, wie bei der entsprechenden früher durchgeführten Propagation-Iteration. Diese ist nur dann gewährleistet, wenn alle Substrukturierungstests sowohl in Propagation- als auch in Repropagation-Iterationen auf Basis der in der jeweiligen Iteration empfangenen Energie durchgeführt werden und nicht auf Basis der gesamten bisher im Berechnungsverlauf empfangenen Energie. Dadurch wird es möglich bei asynchronen parallelen Berechnungen die negative Energie korrekt innerhalb der Patch-Hierarchien zu verteilen, so daß Radiosity-Werte in einer Redistribution-Iteration exakt auf den Ebenen korrigiert werden, auf denen sie in der entsprechenden Propagation-Iteration zuvor verteilt wurden.

5.3.3.5 Ende-Erkennung

Wenn Eigenschaften von Objekten (z.B. Farbe, spezifische Ausstrahlung) modifiziert werden, verändert sich aufgrund der unmittelbar auszuführenden Korrekturberechnungen (siehe 4.3.1) die Menge der lokal auf den Prozessoren vorhandenen Restenergie, die noch zu verteilen ist, bevor eine globale Konvergenz erreicht ist. Insbesondere wenn der Szene Energie entzogen wird (Lichtquelle aus) oder wenn ihr Energie hinzugefügt wird (Lichtquelle an) ändert sich die initial in der Szene enthaltene Energiemenge. Um eine korrekte Ende-Erkennung für parallele inkrementelle Radiosity-Berechnungen im Anschluß an derartige Szenenmodifikationen zu ermöglichen, ist es notwendig, Informationen über die jeweils lokal existierenden Mengen von Restenergie zwischen den Prozessoren auszutauschen und die initial in der Szene enthaltene Energie auf allen Prozessoren gemäß der Modifikation zu korrigieren. Unter dieser Voraussetzung kann die in 5.3.2.6 beschriebene Methode zur Ende-Erkennung von parallelen Radiosity-Berechnungen in statischen Szenen auch im Fall dynamisch veränderbaren Szenen verwendet werden.

5.3.3.6 Performanz der parallelen inkrementellen Berechnungen

Um die Performanz des vorgestellten parallelen Progressive-Refinement-Verfahrens zur inkrementellen Berechnung einer Radiosity-Lösung in dynamisch veränderbaren Umgebungen zu untersuchen, wurden Meßreihen auf dem *hpcLine*-System von Fujitsu/Siemens (siehe 5.3.2.7) mit zwei verschiedenen Szenen unterschiedlicher Komplexität durchgeführt. Die Testmessungen basieren auf dem gleichen eingeschränkten Anwendungsszenario, wie es schon bei den in Abschnitt 5.2.7.2 präsentierten Messungen verwendet wurde. Bei allen Berechnungen wurde Shaft-Culling in Kombination mit Begrenzungsvolumen für Objekte und dem Shadow-Cache durchgeführt, um die Anzahl der Schnittpunkttests während der Bestimmung von Formfaktoren zu reduzieren (siehe 5.3.1.3). Es wurden ausschließlich parallele Berechnungen von Punkt-zu-Patch Formfaktoren nach Wallace u.a. [WKEH89] (siehe Abschnitt 5.3.1.1) durchgeführt, wobei die Sichtbarkeit der Shooting-Patches immer bzgl. einem Sample-Punkt überprüft wurde. Aus den Ergebnissen, die in 5.3.2.7 präsentiert wurden, kann geschlossen werden, daß die Skalierbarkeitseigenschaften der inkrementell durchgeführten Korrekturberechnung im allgemeinen besser werden, wenn eine größere Anzahl von Sample-Strahlen generiert wird. In diesem Fall wird die lokale Berechnungskomplexität der parallel zu lösenden Teilprobleme größer, ohne das der Kommunikations-Overhead signifikant zunimmt.

Zunächst wurden Messungen auf Grundlage der schon aus Abschnitt 5.3.2.7 bekannten Szene *Büro* durchgeführt. Innerhalb dieser sehr einfachen Szene wurden verschiedene Modifikationen vorgenommen und die zur inkrementellen Korrektur einer zuvor bestimmten Radiosity-Lösung (*Redistribution*) benötigten Berechnungszeiten in dem eingeschränkten Anwendungsszenario ermittelt. Zusätzlich wurden die Zeiten ermittelt, die benötigt werden, um eine vollständige Neuberechnung der Radiosity-Lösung (*Repropagation*) durchzuführen. Es wurde sowohl bei den inkrementellen Korrekturberechnungen als auch bei der vollständigen Neuberechnung eine zu 99% konvergierte Radiosity-Lösung bestimmt. Die Berechnungszeiten für die inkrementellen Korrekturberechnungen und die entsprechenden vollständigen Neuberechnungen sind in Tabelle 5.30 zusammengefaßt.

Testszene <i>Büro</i>								
Berechnungszeiten für verschiedene Operationen (Sekunden)								
Proz.	<i>Tisch entfernen</i>		<i>Bodenfarbe ändern</i>		<i>Tisch einfügen</i>		<i>Lichtquelle ausschalten</i>	
	inkr.	Reprop.	inkr.	Reprop.	inkr.	Prop.	inkr.	Reprop.
1	342	914	497	815	347	1053	701	1057
2	193	486	273	436	183	600	364	588
4	99	255	154	236	98	306	200	304
8	60	159	81	133	55	169	120	170
12	44	111	60	99	43	127	83	130
16	36	95	49	88	37	100	72	108
20	34	87	42	72	36	91	60	89
24	-	-	41	65	32	83	56	78
28	-	-	38	61	33	81	55	77
32	-	-	34	57	30	85	50	76

Tabelle 5.30: Gegenüberstellung von inkrementellen Berechnungszeiten einer korrigierten Radiosity-Lösung und den Berechnungszeiten, die zur vollständigen Neuberechnung der Radiosity-Lösung im Anschluß an eine Modifikation der Testszene *Büro* benötigt werden.

In der Testszene Büro wurden Veränderungen vorgenommen, welche die Geometrie der ursprünglichen Szene (Tisch entfernen, Tisch einfügen) und Objekteigenschaften (Bodenfarbe ändern, Lichtquelle ausschalten) betreffen. Diese Veränderungen haben unterschiedliche Auswirkungen auf die globale Beleuchtung in der Szene. Insbesondere die Auswirkungen einer veränderten Geometrie sind lokal sehr begrenzt und somit müssen nur wenige inkrementelle Formfaktoren bestimmt werden. Dieses ist der Grund dafür, daß die inkrementellen Korrekturberechnungen deutlich schneller sind als eine vollständige Neuberechnung der Radiosity-Lösung. Die Veränderungen der Objekteigenschaften haben aufgrund der geringen Komplexität der Testszene Büro große Auswirkungen auf die globale Beleuchtung. Die Zeiten zur inkrementellen Korrektur der Radiosity-Lösung sind hier ebenfalls deutlich kürzer als die Zeiten, die zur vollständigen Neuberechnung einer Radiosity-Lösung für die veränderte Szene benötigt werden. Durch die parallelen inkrementellen Berechnungen können die Aktualisierungszeiten einer Radiosity-Lösung im Anschluß an eine Veränderung in der Szene noch erheblich verkürzt werden.

Um einen besseren Vergleich der Zeiten für Redistribution- und Repropagation-Berechnungen zu ermöglichen, werden diese in Abb. 5.51 graphisch gegenübergestellt.

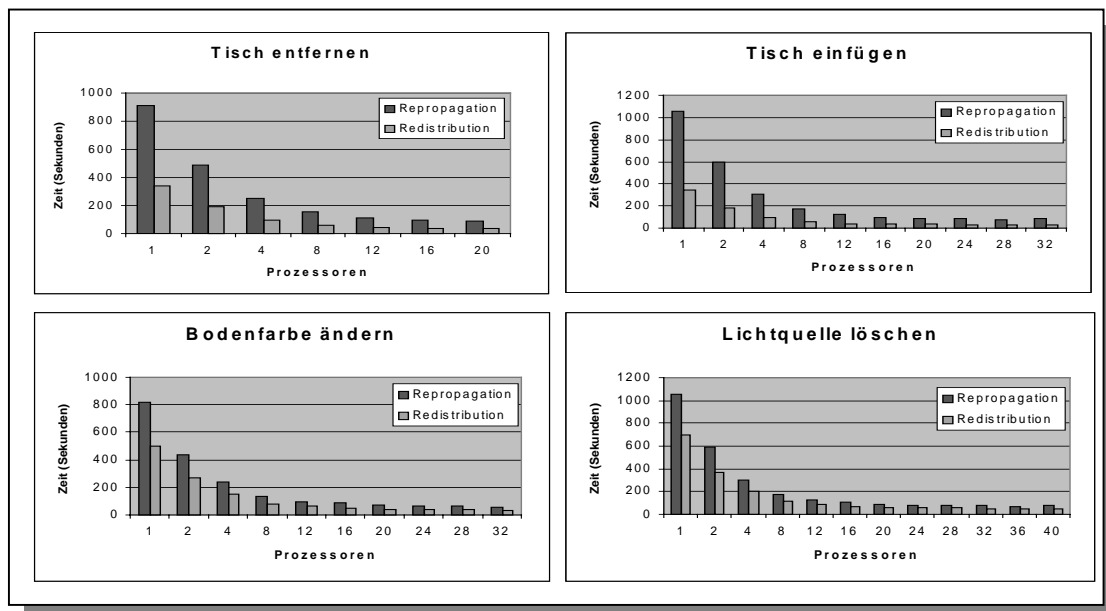


Abb. 5.51: Graphische Darstellung der parallelen inkrementellen Berechnungszeiten und der Zeiten zur Berechnung einer vollständigen Neuberechnung einer Radiosity-Lösung im Anschluß an verschiedene Modifikationen der Szene Büro.

Interessant ist nun die Fragestellung, wie gut die parallelen inkrementellen Korrekturberechnungen im Vergleich zu den parallelen Neuberechnungen der Radiosity-Lösungen skalieren. Aufschluß hierüber gibt Abb. 5.52, wo der sich aus den ermittelten Berechnungszeiten ergebende Speed-Up visualisiert ist. Anhand der Diagramme ist deutlich zu erkennen, daß der Speed-Up der parallelen Redistribution-Berechnungen und der Speed-Up der parallelen Repropagation-Berechnungen ähnlich ist und keine deutlichen Abweichungen zu erkennen sind. Die inkrementellen Korrekturberechnungen können selbst für eine Szene mit geringer Komplexität durch die zufällige Verteilung der Patches auf die Prozessoren und die dadurch bedingte gute statische Lastverteilung noch auf einer größeren Anzahl von Prozessoren effizient ausgeführt werden. Mit dem datenparallelen Verfahren, das in Abschnitt 5.2.7 eingeführt wurde, konnten selbst für eine Szene mit großer Komplexität Redistribution-Berechnungen

nur auf bis zu 16 Prozessoren durchgeführt werden, wobei die Effizienz deutlich unter 40% lag.

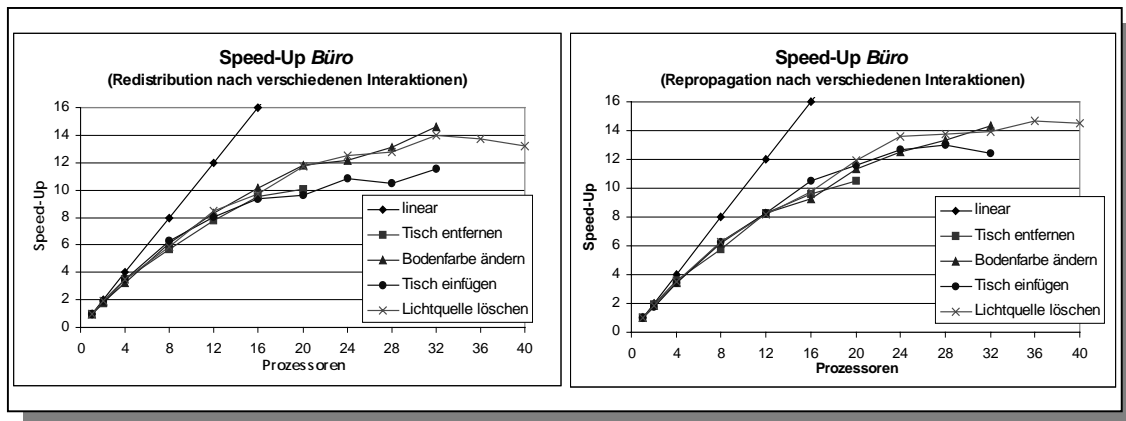


Abb. 5.52: Speed-Up der parallel durchgeführten inkrementellen Radiosity-Berechnungen (links) und der entsprechenden parallelen Neuberechnungen einer Radiosity-Lösung (rechts).

Ein komplexeres Modell ist durch die ebenfalls aus Abschnitt 5.3.2.7 bekannten Szene *Konferenzraum* gegeben. Auch in dieser Szene wurden verschiedene Veränderungen von Materialeigenschaften und der Geometrie vorgenommen und die Berechnungszeiten der inkrementellen Korrekturberechnungen in dem eingeschränkten Anwendungsszenario ermittelt. Zusätzlich wurden die Zeiten ermittelt, die benötigt wurden, um eine vollständige Neuberechnung der Radiosity-Lösung durchzuführen. Es wurde sowohl bei den inkrementellen Korrekturberechnungen als auch bei der vollständigen Neuberechnung eine zu 99% konvergierte Radiosity-Lösung bestimmt. Die Berechnungszeiten für die inkrementellen Korrekturberechnungen und die entsprechenden vollständigen Neuberechnungen sind in Tabelle 5.31 zusammengefaßt.

Testszene Konferenzraum								
Berechnungszeiten für verschiedene Operationen (Sekunden)								
Proz.	Stühle entfernen		Objekt einfügen		Objektfarbe ändern		Eine Lichtquelle löschen	
	inkr.	Reprop.	inkr.	Reprop.	inkr.	Reprop.	inkr.	Reprop.
1	1271	4122	1123	5022	1799	4682	3198	5050
2	679	2355	595	2856	974	2599	1800	2834
4	346	1253	317	1482	490	1384	938	1453
6	247	837	220	1042	366	956	663	1044
8	193	671	173	780	273	743	500	788
12	137	462	119	577	191	518	349	552
16	111	370	98	451	156	407	277	450
20	95	323	81	365	150	373	236	379
24	89	300	78	337	123	324	205	349
28	79	269	72	320	104	289	202	302
32	79	247	66	280	101	261	184	302
36	-	-	66	293	99	263	169	277
40	-	-	-	-	94	251	162	281
44	-	-	-	-	85	242	155	258

Tabelle 5.31: Gegenüberstellung von inkrementellen Berechnungszeiten einer korrigierten Radiosity-Lösung und den Berechnungszeiten, die zur vollständigen Neuberechnung der Ra-

diosity-Lösung im Anschluß an eine Modifikation der Testszene *Konferenzraum* benötigt werden.

Wie schon bei den Berechnungen auf Basis der Szene *Büro*, kann auch für die Szene *Konferenzraum* eine erhebliche Zeitersparnis durch die inkrementellen Korrekturberechnungen erreicht werden. Aufgrund der größeren Komplexität dieser Szene weisen die Auswirkungen der Veränderungen der Szene auf die globale Beleuchtung eine größere Lokalität auf. Aus diesem Grund kann durch Redistribution-Berechnungen eine deutlich größere Zeitersparnis im Vergleich zu einer vollständigen Neuberechnung erreicht werden, als es bei der einfacheren Testszene der Fall ist.

Um einen besseren Vergleich der Zeiten für die Repropagation- und die Redistribution-Berechnungen zu ermöglichen, werden die Unterschiede in Abb. 5.53 graphisch dargestellt.

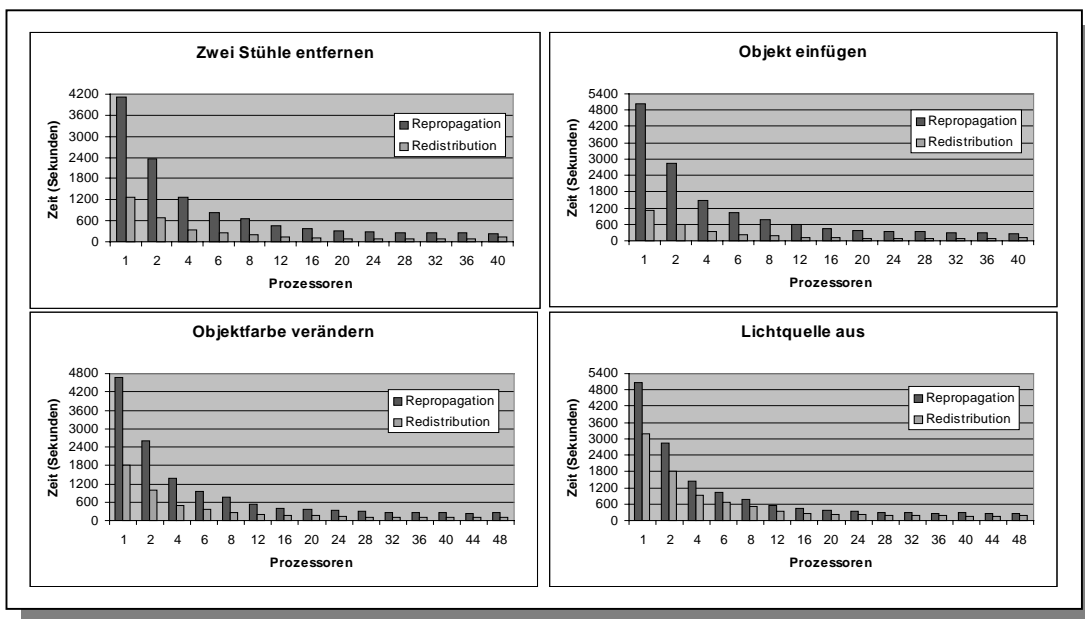


Abb. 5.53: Graphische Darstellung der parallelen inkrementellen Berechnungszeiten und der Zeiten zur Berechnung einer vollständigen Neuberechnung einer Radiosity-Lösung im Anschluß an verschiedene Modifikationen der Szene *Konferenzraum*.

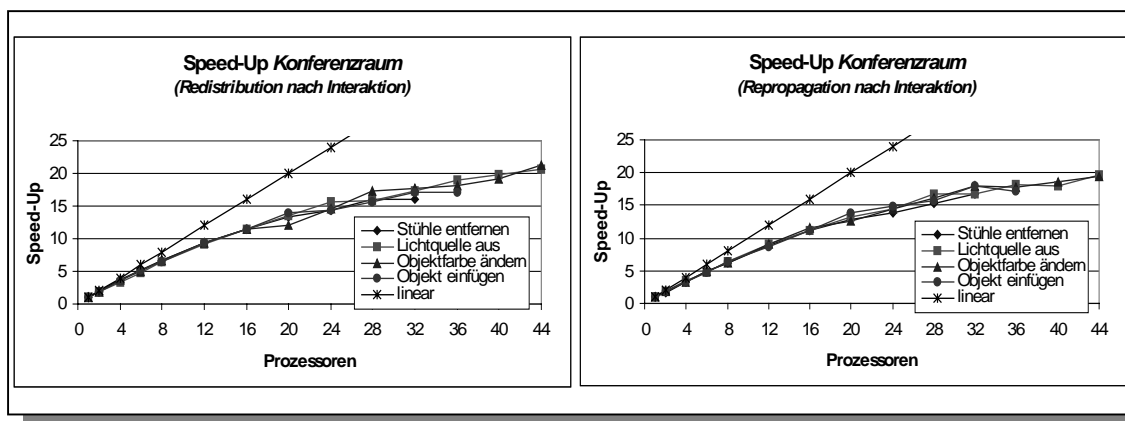


Abb. 5.54: Speed-Up der parallel durchgeführten inkrementellen Radiosity-Berechnungen (links) und der entsprechenden parallelen Neuberechnungen einer Radiosity-Lösung (rechts).

Aufgrund der größeren Komplexität der Szene können effiziente parallele Redistribution-Berechnungen auf einer größeren Anzahl von Prozessoren durchgeführt werden. Die Übereinstimmung zwischen den Skalierbarkeitseigenschaften der parallelen Redistribution-Berechnungen und der parallelen Repropagation-Berechnungen nehmen hierbei mit wachsender Komplexität der Szene noch zu (vgl. Abb. 5.52 und Abb. 5.54).

Die durchgeführten Testmessungen demonstrieren, daß Redistribution-Berechnungen für eine zuvor berechnete Radiosity-Lösung im Anschluß an Veränderungen der Szene auf einer großen Anzahl von Prozessoren effizient durchgeführt werden können. Die Berechnungszeiten zur Anpassung einer Radiosity-Lösung sind hierbei erheblich kürzer als die Zeiten, die zur vollständigen Neuberechnung einer Radiosity-Lösung nötig sind. Die Lokalität von Veränderungen der Szene ist offensichtlich in sehr komplexen Szenen besonders stark ausgeprägt, insbesondere wenn die Szene ein Gebäude mit mehreren getrennten Räumen darstellt. In diesem Fall werden die Berechnungszeiten der inkrementellen Korrekturberechnungen im Vergleich zu einer vollständigen Neuberechnung der Radiosity-Lösung noch deutlich kürzer ausfallen.

Die Skalierbarkeitseigenschaften von parallelen Berechnungen in dynamisch veränderbaren Szenen entsprechen denen von parallelen Neuberechnungen einer Radiosity-Lösung im Anschluß an eine Modifikation der Szene. Aus diesem Grund gelten alle in Abschnitt 5.3.2.7 gemachten Aussagen über die Skalierbarkeit des parallelen Progressive-Refinement-Verfahrens für statische Szenen auch für die inkrementellen Korrekturberechnungen in dynamisch veränderbaren Szenen. Die in diesem Abschnitt präsentierten Meßergebnisse wurden auf Basis der Berechnung von Eckpunkt-Radiositäten nach Wallace u.a. ermittelt. Hierbei ist jeweils nur ein Sample-Punkt pro Shooting-Patch verwendet worden. Wenn eine größere Anzahl von Sample-Punkten pro Shooting-Patch generiert wird, verbessert sich die Effizienz der parallelen Berechnungen und größere Anzahlen von Prozessoren können bei den Korrekturberechnungen verwendet werden.

Wie schon bei dem in Abschnitt 5.2.7 beschriebenen Verfahren zur parallelen Berechnung von Radiosity-Lösungen in dynamisch veränderbaren Szenen, korrigiert auch das neue parallele Verfahren die direkten Auswirkungen von Änderungen der Szene auf die globale Beleuchtung wesentlich schneller, als es die in dem eingeschränkten Anwendungsszenario ermittelten Berechnungszeiten vermuten lassen. Bei parallel durchgeführten inkrementellen Korrekturberechnungen werden die wesentlichen Auswirkungen auf die globale Beleuchtung schon im kleinen Sekundenbereich angepaßt und die veränderte Beleuchtungssituation kann bei der Bildgenerierung berücksichtigt werden. Somit stellt das vorgestellte parallele Verfahren einen Schritt in Richtung des Modellierungsparadigmas *Rendering while Modelling* dar.

5.3.4 Datenparallele Berechnung von erweiterten Formfaktoren

Das Ziel von Verfahren zur Simulation globaler Beleuchtungseffekte ist es, alle möglichen Pfade zu berücksichtigen, die ein Photon nach dessen Emission durch eine Lichtquelle innerhalb einer Umgebung zurücklegen kann, bis es von einer Oberfläche absorbiert wird. Wie in 2.1.3.2 gezeigt wurde, lassen sich diese sogenannten Lichttransportpfade mit Hilfe von regulären Ausdrücken klassifizieren. Die bisher in dieser Arbeit vorgestellten parallelen Verfahren unterstützen lediglich die Simulation von Teilmengen der Menge $L(D/S)^*E$ aller möglichen Lichttransportpfade:

- LD^*E (Radiosity – 5.2 und 5.3.2)
- LD^*S^*E (einfache Zwei-Phasen-Methode – 5.2.8 und 5.3.5)

Das in 5.3.2 vorgestellte parallele Progressive-Refinement-Verfahren ist dahingehend erweitert worden, daß die berechnete blickpunktunabhängige Radiosity-Lösung wesentlich komplexere Transportpfade aus der Menge $LD^*(SD^+)^*D^*E$ berücksichtigt. Dieses trägt zu einer deutlichen Steigerung des Realitätsgrades der generierten Bilder bei und es lassen sich Beleuchtungseffekte berücksichtigen, die durch konventionelle Radiosity- und Raytracing-Methoden nicht zu simulieren sind (z.B. die indirekte Beleuchtung von Objekten mit Hilfe eines Spiegels).

In Abb. 5.55 ist ein Beispiel für eine indirekte Beleuchtung eines Objektes über eine spiegelnde Oberfläche angegeben. In dem linken Bild ist die dem Spiegel zugewandte Seite des Würfels auf der Schreibtischoberfläche nicht beleuchtet, während er im rechten Bild aufgrund der Bestimmung von erweiterten Formfaktoren durch spiegelnd reflektierte Energie einer Deckenlichtquelle beleuchtet wird.

An dieser Stelle wird davon ausgegangen, daß die Objekte der Szene das Licht ideal diffus und ideal spiegelnd reflektieren. Es ist jedoch zu beachten, daß durch das grundlegende Konzept die Berücksichtigung von beliebigen bidirektionalen Reflexions-Verteilungsfunktionen (2.1.3.1) möglich ist. Insbesondere Brechungen des Lichtes durch transparente Objekte können in zukünftigen Implementierungen des parallelen Verfahrens integriert werden.

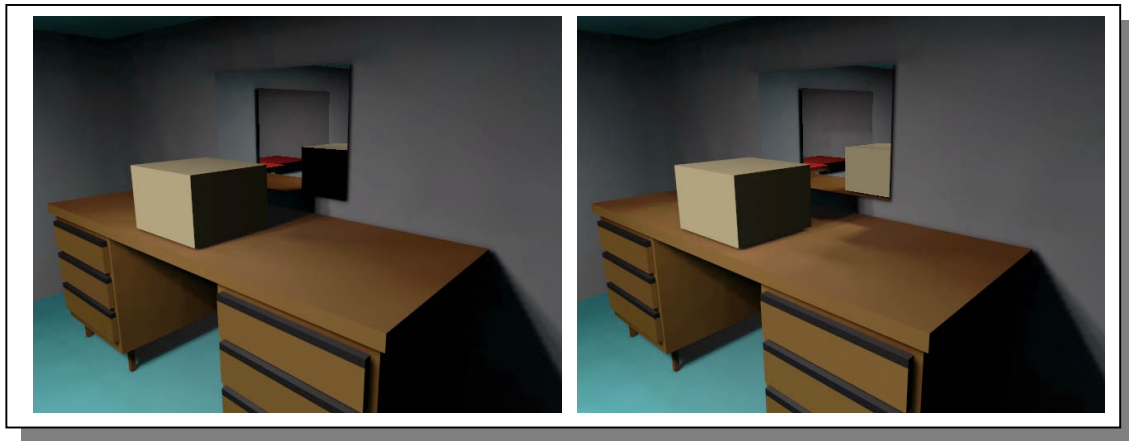


Abb. 5.55: Einfache Zwei-Phasen-Visualisierung einer Radiosity-Lösung mit einfachen Formfaktoren (links) und mit erweiterten Formfaktoren (rechts).

Die Erweiterung des parallelen Progressive-Refinement-Verfahrens zur Berechnung von erweiterten Formfaktoren besteht aus der Durchführung von zwei Berechnungsphasen pro Shooting-Iteration. In der ersten Berechnungsphase wird gemäß dem bisher beschriebenen Verfahren der direkte Energietransfer zwischen einem Shooting-Patch und den übrigen Patches der Szene oder deren Eckpunkten bestimmt. Im Anschluß daran wird in einer zusätzlichen Berechnungsphase der indirekte Transfer von Energie des Shooting-Patches über spiegelnde Flächen approximiert. Dieses geschieht auf Basis der Berechnung von erweiterten Formfaktoren (siehe 3.5.2) mit Hilfe von Informationen, die in der ersten Berechnungsphase in einer speziellen Datenstruktur gespeichert wurden.

Während der ersten Phase einer Progressive-Refinement-Iteration wird für jedes Empfänger-Patch überprüft, ob es einfallendes Licht spiegelnd reflektiert. In diesem Fall werden die Sample-Strahlen zwischen Shooting-Patch und Empfänger-Patch jeweils in ideale Spiegel-

richtung reflektiert. Für jeden reflektierten Strahl wird mit Hilfe des Intersection-Netzes die Objektoberfläche bestimmt, auf die der Strahl als nächstes auftrifft. Die gefundenen Objektoberflächen sowie die jeweils zugehörige Spiegelfläche (oder auch Transferfläche) werden in einer als *Reflection-Buffer* bezeichneten Datenstruktur vermerkt. Es ist zu beachten, daß jede Oberfläche eines Objektes des Intersection-Netzes eindeutig durch ein Tupel bestehend aus Objektoberfläche und Objekt Nummer bestimmt ist. Bei der Erweiterung des Reflection-Buffers wird darauf geachtet, daß jedes Paar bestehend aus Transferfläche und indirekte Empfänger Oberfläche nur einmal in dem Reflection-Buffer vermerkt wird. Nach Beendigung der ersten Berechnungsphase einer Iteration sind in dem entsprechenden Reflection-Buffer somit Verweise auf alle Objektoberflächen gespeichert, deren Patches potentiell Energie von dem Shooting-Patch dieser Iteration indirekt über eine bestimmte Transferfläche (z.B. Spiegel) empfangen können. Zwischen diesen Patches und dem aktuellen Shooting-Patch werden in der zweiten Berechnungsphase der Iteration erweiterte Formfaktoren bestimmt, um die indirekt empfangene Energiemenge zu bestimmen. Die gesamte Energiemenge, die ein Patch empfangen kann, setzt sich in jeder Iteration somit aus der Summe von direkt und indirekt empfangener Energie zusammen.

Um eine korrekte datenparallele Berechnung von erweiterten Formfaktoren in den einzelnen Iterationen zu ermöglichen, muß ein Prozessor in der zweiten Berechnungsphase einer Iteration über einen Reflection-Buffer verfügen, der die identischen Informationen enthält, wie der entsprechende sequentiell erzeugte Reflection-Buffer dieser Iteration. Wenn bei datenparallelen Berechnungen lokale Reflection-Buffer durch die Prozessoren aufgebaut werden, berücksichtigen diese nach Beendigung der ersten Berechnungsphase einer Iteration jeweils nur einen Teil der in der gesamten Szene auftretenden spiegelnden Reflexionen. Der vollständige Reflection-Buffer kann durch Vereinigung der durch die Prozessoren berechneten lokalen Reflection-Buffer des gleichen Shooting-Patches erfolgen. Nach der Vereinigungsoperation steht jedem Prozessor ein Reflection-Buffer zur Verfügung, der dem entsprechenden sequentiell erzeugten Reflection-Buffer entspricht. Die Vereinigungsoperation basiert hierbei auf einem synchronen oder asynchronen Austausch der lokalen Reflection-Buffer zwischen den Prozessoren. Im Fall eines synchronen Austausches der Reflection-Buffer wird die zweite Berechnungsphase (Berechnung von erweiterten Formfaktoren) nach der Vereinigung der lokalen Reflection-Buffer zu einem vollständigen Reflection-Buffer unmittelbar im Anschluß an die erste Berechnungsphase einer Iteration ausgeführt. Bei einem asynchronen Austausch von lokal berechneten Reflection-Buffern werden diese unmittelbar nach der ersten Berechnungsphase bearbeitet. Die fehlenden erweiterten Formfaktoren bzgl. des gleichen Shooting-Patches werden jeweils auf Basis eines empfangenen unvollständigen Teil-Reflection-Buffern berechnet. Es ist zu beachten, daß für ein bestimmtes Shooting-Patch sämtliche erweiterten Formfaktoren erst dann berechnet wurden, wenn alle Teil-Reflection-Buffer von jedem Prozessor empfangen und bearbeitet worden sind. Jeder Prozessor muß hierbei garantieren, daß kein indirekter Energietransfer mehrfach ausgeführt werden kann. Dieses ist notwendig, da Teil-Reflection-Buffer die bzgl. eines asynchron ausgewählten Shooting-Patches auf verschiedenen Prozessoren erzeugt wurden, identische Oberflächenpaare enthalten können. Bei synchronen datenparallelen Berechnungen können während der Vereinigung der Teil-Reflection-Buffer doppelte Einträge sehr einfach eliminiert werden. Bei den asynchronen Berechnungen ist dieses nur mit Hilfe einer Liste von Reflection-Buffern möglich, in der empfangene Teil-Reflection-Buffer vereinigt und verwaltet werden, um jeweils die Oberflächenpaare in einem empfangenen Teil-Reflection-Buffer zu bestimmen, die bei der Bestimmung von erweiterten Formfaktoren bisher noch nicht berücksichtigt wurden. Da die in den Reflection-Buffern gespeicherten Informationen nur ein geringes Datenvolumen aufweisen, ist der Kommunikationsaufwand bei der parallelen Berechnung von erweiterten Formfaktoren gering.

Die in der zweiten Berechnungsphase einer Iteration verwendete Methode zur Approximation von erweiterten Formfaktoren kann als direkte Erweiterung der zur Bestimmung von herkömmlichen Formfaktoren verwendeten Methoden (siehe 5.3.1.1) betrachtet werden. Hierbei wird auf die in 3.5.2 beschriebene Idee zurückgegriffen, bei der ein Spiegel ein virtuelles Fenster darstellt, welches den Blick in eine Szene ermöglicht, die an der Spiegelebene gespiegelt wurde. Die Einträge des Reflection-Buffer werden sukzessive abgearbeitet, wobei zunächst die auf dem Shooting-Patch platzierten Sample-Punkte an der in dem jeweiligen Eintrag spezifizierten Spiegeloberfläche gespiegelt werden.

Zwischen Sample-Punkten auf den Patches bzw. den Eckpunkten einer potentiellen Empfängeroberfläche werden Sample-Strahlen in Richtung der gespiegelten Sample-Punkte des Shooting-Patches generiert (siehe Abb. 5.56). Anschließend wird überprüft, ob diese Strahlen die entsprechende Spiegelfläche (virtuelles Fenster) treffen. Über alle Strahlenbahnen, welche die Spiegelfläche schneiden, kann potentiell Energie von dem Shooting-Patch zu dem zugehörigen Sample-Punkt auf dem Empfänger-Patch transportiert werden. Bei der Bestimmung eines erweiterten Formfaktors werden zwei Sample-Punkte als gegenseitig indirekt sichtbar betrachtet, wenn der entsprechende reflektierte Sample-Strahl weder auf der Wegstrecke von dem Shooting-Patch zum Spiegel noch auf der Strecke vom Spiegel zu den Sample-Punkten auf dem Empfänger-Patch durch ein anderes Objekt der Szene abgeblockt wird.

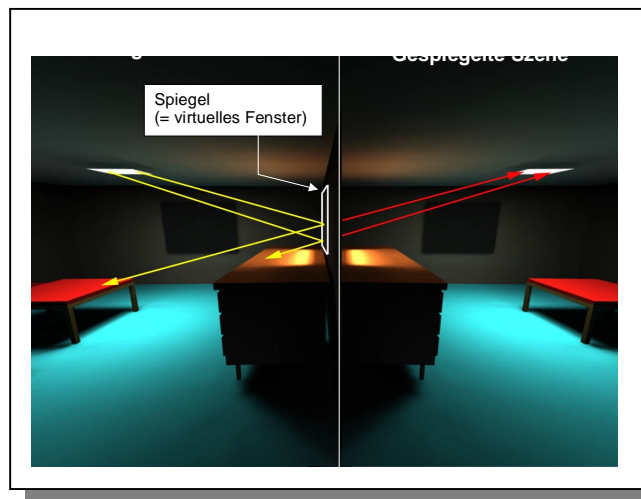


Abb. 5.56: Berechnung von erweiterten Formfaktoren.

Wenn zwei Sample-Punkte gegenseitig indirekt sichtbar sind, wird das entsprechende Punktpaar bei der Approximation des erweiterten Formfaktors berücksichtigt. Dieser ergibt sich analog zu den in 3.1.2.5 und 3.1.2.6 beschriebenen Methoden aus der Kombination der berechneten Formfaktoren zwischen den Sample-Punkten. Bei dem Energietransfer auf Basis der erweiterten Formfaktoren muß zusätzlich noch berücksichtigt werden, daß die indirekt empfangene Energie noch gemäß der spiegelnden Reflexionskoeffizienten der Transferoberfläche abgeschwächt werden muß. Dieses geschieht durch Multiplikation des Reflexionskoeffizienten der Transferoberfläche mit dem berechneten erweiterten Formfaktor.

Um zu verdeutlichen, daß die parallele Berechnung von erweiterten Formfaktoren sehr effizient durchgeführt werden kann, wurden in die Testszene Büro zwei perfekte Spiegel eingefügt, welche die Schreibtischoberfläche und den Boden indirekt durch spiegelnde Reflexion beleuchten. In dieser Testszene wurden Radiosity-Lösungen mit erweiterten Formfaktoren und

Radiosity-Lösungen mit den konventionellen Formfaktoren mit verschiedenen Anzahlen von Prozessoren bestimmt. Hierbei wurden die Radiosity-Werte an den Eckpunkten des Radiosity-Netzes bestimmt. Die adaptive Substrukturierungstiefe wurde variiert, um die Auswirkungen der Veränderung der lokalen Berechnungskomplexität auf die Berechnungszeiten und die Skalierbarkeit der parallelen Berechnungen zu verdeutlichen. Die Meßergebnisse sind in Tabelle 5.32 zusammengefaßt.

Es ist deutlich zu erkennen, daß die Berechnungszeiten einer zu 99% konvergierten Radiosity-Lösung bei der Verwendung von erweiterten Formfaktoren erheblich länger sind, als wenn Radiosity-Lösungen auf Basis der konventionellen Formfaktoren berechnet werden. Dieses ist durch den größeren Berechnungsaufwand zu begründen, der bei der Berechnung von erweiterten Formfaktoren durch zusätzliche Sichtbarkeitsüberprüfungen entsteht. Durch die Verwendung von mehreren Prozessoren können die Berechnungszeiten jedoch deutlich reduziert werden.

Szene Büro				
Berechnungszeiten für eine Lösung mit normalen und mit erweiterten Formfaktoren				
Prozessoren	max. Tiefe 4		max. Tiefe 6	
	normal	erweitert	normal	erweitert
1	41m 14s	56m 11s	1h 46m 57s	2h 36m 15s
2	23m 00s	31m 37s	58m 06s	1h 25m 35s
4	12m 56s	17m 15s	31m 19s	44m 41s
8	7m 14s	9m 56s	18m 36s	27m 00s
16	4m 41s	6m 02s	9m 47s	15m 12s
24	3m 16s	4m 59s	7m 49s	12m 15s
32	2m 33s	3m 51s	6m 09s	9m 28s
40	2m 18s	3m 21s	5m 11s	7m 54s
48	2m 10s	3m 48s	5m 10s	7m 43s

Tabelle 5.32: Berechnungszeiten für die parallele Berechnung einer Radiosity-Lösung auf Basis von erweiterten und von normalen Formfaktoren.

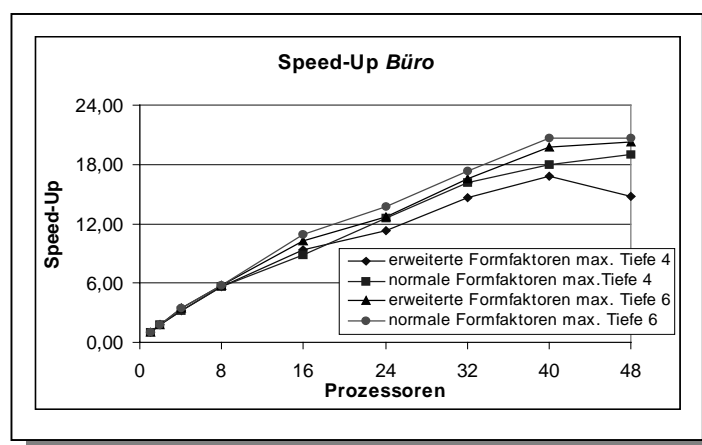


Abb. 5.57: Speed-Up für die parallele Berechnung einer Radiosity-Lösung auf Basis von erweiterten und von normalen Formfaktoren.

Wenn der erzielte Speed-Up gegenübergestellt wird (siehe Abb. 5.57), ist deutlich zu erkennen, daß die parallelen Berechnungen besser skalieren, wenn eine Radiosity-Lösung basierend auf den konventionellen Formfaktoren berechnet wird. Die Ursache hierfür ist, daß bei der Berechnung von erweiterten Formfaktoren ein zusätzliche Kommunikations-Overhead entsteht, wenn die Reflection-Buffer zwischen den Prozessoren ausgetauscht werden. Je größer jedoch der lokale Berechnungsaufwand der Prozessoren wird (z.B. durch tiefere Substrukturierung der Patches), fällt dieser Kommunikations-Overhead weniger ins Gewicht und die Skalierbarkeit der parallelen Berechnung einer Radiosity-Lösung auf Basis von erweiterten Formfaktoren verbessert sich deutlich.

5.3.5 Blickpunktabhängige Spiegelungen

Das in 5.3 bisher vorgestellte datenparallele Progressive-Refinement-Verfahren simuliert Beleuchtungseffekte, die unabhängig von dem Standpunkt eines Betrachters sind. In realen Umgebungen sind jedoch sehr häufig Objekte mit Materialeigenschaften anzutreffen, die Spiegelungen der Umgebung auf den Objektoberflächen verursachen. Diese Spiegelungen sind abhängig von dem Standpunkt und der Blickrichtung eines Betrachters innerhalb der Szene. In Abschnitt 5.2.8 wurde ein paralleles Verfahren vorgestellt, das in einer Berechnungsphase im Anschluß an eine zuvor durchgeführte Radiosity-Berechnung die Generierung von Einzelbildern und Animationen ermöglicht. Durch diese einfache Zwei-Phasen-Methode werden bei der synthetischen Bildgenerierung sämtliche Lichttransportpfade aus der Menge LD^*S^*E berücksichtigt und die entsprechenden Beleuchtungseffekte visualisiert.

Das parallele Verfahren aus Abschnitt 5.2.8 kann ebenfalls in direkter Kombination mit der in 5.3.4 beschriebenen erweiterten Radiosity-Methode verwendet werden, wodurch Lichttransportpfade der Menge $LD^*(SD^+)^*D^*S^*E$ simuliert werden. Eine solche Kombination der beiden parallelen Verfahren ist jedoch aus verschiedenen Gründen als problematisch zu betrachten. Da bei diesem Ansatz die Bildgenerierung nicht datenparallel erfolgt, ist die Größe der Szenen durch den zur Verfügung stehenden Hauptspeicher der Prozessoren limitiert. Weiterhin muß die vollständige Radiosity-Lösung vor der Bildgenerierung berechnet und in eine Datei geschrieben werden. Diese Datei wird von allen während der Raytracing-Phase verwendeten Prozessoren eingelesen. Der damit verbundene Mehraufwand reduziert die Effizienz des parallelen Bildgenerierungsprozesses.

Die Lösung dieser Probleme besteht in einer Erweiterung des datenparallelen Radiosity-Verfahrens, so daß die Bildgenerierung direkt in die parallelen Progressive-Refinement-Berechnungen integriert wird. Zur Erzeugung von Einzelbildern, die Spiegelungen berücksichtigen, werden datenparallele Raytracing-Berechnungen auf Basis der verteilt gespeicherten Zwischenergebnisse der Progressive-Refinement-Methode durchgeführt, wobei die Kamerapositionen und Blickrichtungen sowie Bildauflösung durch den Benutzer frei definiert werden können.

5.3.5.1 Grundprinzip der integrierten einfachen Zwei-Phasen-Methode

Gemäß der durch einen Anwender spezifizierten Kameradaten wird innerhalb der Szene eine virtuelle Bildebene generiert. Diese wird gemäß der Auflösung der virtuellen Bildebene in Bildpunkte (*Pixel*) diskretisiert. Die Ermittlung der Pixel-Intensitäten (bzw. Farben) erfolgt durch eine rekursive Strahlverfolgung [Whi80][Glass89]. Von der definierten Kameraposition wird ein Strahl durch jedes Pixel der virtuellen Bildebene gesendet (*Sehstrahl*). Diese Sehstrahlen werden auf Schnittpunkte mit den Objekten der Szene überprüft (siehe Abb. 5.58).

Wenn ein Strahl auf kein Objekt der Szene trifft, wird dem jeweiligen Pixel eine vordefinierte Hintergrundfarbe zugeordnet. Trifft ein Sehstrahl auf ein Objekt, wird mit Hilfe eines einfachen lokalen Beleuchtungsmodells (siehe hierzu Kapitel 2 und [Glass89]) ein Helligkeitswert für den getroffenen Oberflächenpunkt berechnet, der von der gegebenen Kameraposition aus betrachtet an dem entsprechenden Bildpunkt wahrgenommen werden kann. Das verwendete lokale Beleuchtungsmodell bestimmt für jeden getroffenen Oberflächenpunkt einen diffus und einen spiegelnd reflektierten Helligkeitsanteil. Der diffuse Anteil wird jeweils auf Basis der während der Progressive-Refinement-Berechnungen bestimmten Radiosity-Werte berechnet. Hierzu muß für einen Oberflächenpunkt der durch einen Sehstrahl geschnitten wird, daß Element in den Patch-Hierarchien des Radiosity-Netzes identifiziert werden, welches den Oberflächenpunkt enthält. Der in Richtung Kamera diffus reflektierte Helligkeitsanteil ergibt sich durch bilineare Interpolation der entsprechenden Eckpunkt-Radiosities eines getroffenen Hierarchie-Elementes.

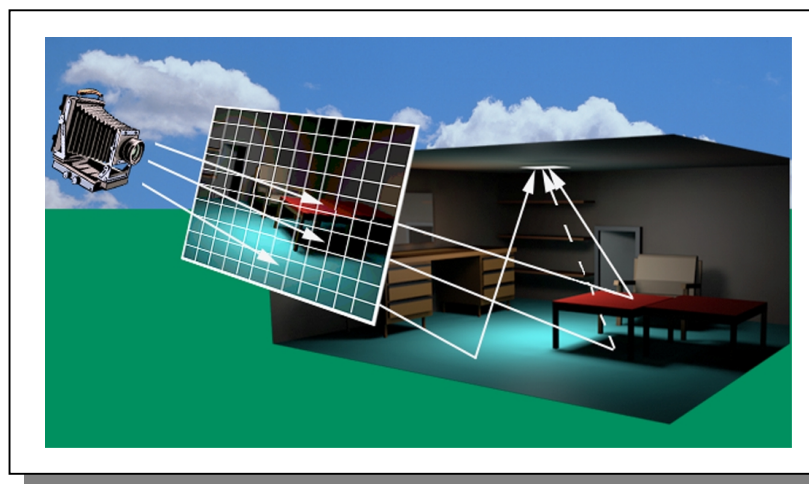


Abb. 5.58: Ermittlung von Helligkeitswerten (Intensitäten) für die Bildelemente einer virtuellen Bildebene.

Wenn die Materialeigenschaften eines getroffenen Oberflächenpunktes eine spiegelnde Reflexion des Lichtes zulassen, wird der reflektierte Sehstrahl (*Reflexionsstrahl*) rekursiv weiterverfolgt, bis eine maximale Rekursionstiefe erreicht oder kein Objekt der Szene getroffen wird. Für jeden auf der Strahlenbahn ermittelten Schnittpunkt wird analog ein diffus und ein spiegelnd reflektierter Helligkeitsanteil ermittelt. Der an einem Pixel der virtuellen Bildebene wahrgenommene Helligkeitswert setzt sich dann als gewichtete Summe der entlang der rekursiv verfolgten Strahlenbahn ermittelten Helligkeitswerte zusammen. Mit Hilfe eines Farbmodells (z.B. RGB-Modell) kann aus den Helligkeitwerten eine Pixelfarbe bestimmt werden.

Bei einem konventionellen rekursiven Raytracing-Verfahren werden für einen gefundenen Schnittpunkt zwischen einem Sehstrahl und einem Objekt der Szene Strahlen in Richtung sämtlicher Lichtquellen der Szene generiert (*Schattenfühler*). Mit Hilfe der Schattenfühler wird überprüft, ob der entsprechende Oberflächenpunkt durch die einzelnen Lichtquellen direkt beleuchtet wird oder ob verdeckende Objekte innerhalb der Szene existieren. Da die direkte Beleuchtung der Szene durch die Lichtquellen schon während der Radiosity-Berechnungen berücksichtigt wurde, besteht nicht die Notwendigkeit, während der Raytracing-Phase Schattenfühler zur Bestimmung der direkten Beleuchtungsanteile zu erzeugen. Hierdurch wird ein erheblicher Teil des Berechnungsaufwandes der Raytracing-Methode eingespart und die harten Schattengrenzen, die üblicherweise durch Raytracing entstehen, werden durch realistische weiche Radiosity-Schatten ersetzt.

5.3.5.2 Parallele Bilderzeugung

Die integrierte Bildgenerierungsphase basiert auf einer parallelen Berechnung der Pixel-Intensitäten bzw. der entsprechenden Farben. Die parallelen Berechnungen werden gemäß einer statischen Aufteilung des Bildraumes durchgeführt, wobei die einzelnen Intensitäten der Pixel zeilenweise durch verschiedene Prozessoren ermittelt werden. Den einzelnen Prozessoren sind aufgrund der verteilten Speicherung des Radiosity-Netzes und der datenparallelen Radiosity-Berechnungen nur Teile der vollständigen Radiosity-Lösung bzw. von Zwischenlösungen lokal bekannt. Zur Bestimmung der diffusen Intensitätsanteile für gefundene Schnittpunkte von Sehstrahlen oder Reflexionsstrahlen mit den Objekten der Szene muß, anders als bei der in 5.2.8 vorgestellten Methode, ein Informationsaustausch zwischen den Prozessoren erfolgen. Da die Patches der Radiosity-Netzes aus Gründen der besseren statischen Verteilung der Last für die datenparallele Progressive-Refinement-Methode zufällig auf die Prozessoren verteilt wurden, kann keine der herkömmlichen Ansätze für datenparallele Raytracing-Berechnungen verwendet werden. Diese beruhen zumeist auf einer geometrischen Partitionierung der Szene in zusammenhängende Teilregionen. Der hier vorgestellte Ansatz zur datenparallelen Bildgenerierung durch eine modifizierte Raytracing-Methode basiert auf dem Konzept des *Pixelstatus-Speichers*.

Der Pixelstatus-Speicher ist eine spezielle Datenstruktur, in der bestimmte Informationen für jedes Pixel der virtuellen Bildebene gespeichert sind. Diese Informationen werden benötigt, um die Berechnung der endgültigen Pixel-Intensitäten (Farben) unter Berücksichtigung der korrekten diffus und spiegelnd reflektierten Helligkeitsanteile an Schnittpunkten bei einer verteilten Speicherung des Radiosity-Netzes zu ermöglichen. Da die parallele Bildgenerierung gemäß einer statischen Aufteilung des Bildraumes auf die Prozessoren erfolgt, ist der Pixelstatus-Speicher kein zusammenhängendes Speichersegment für die gesamte Bildebene. Für jeden der definierten Teilbereiche des Bildes (hier Bildzeilen) wird ein eigenständiger Pixelstatus-Speicher erzeugt und im Berechnungsverlauf aktualisiert. Die Pixelstatus-Speicher der einzelnen Bildzeilen werden zwischen den Prozessoren ausgetauscht. Hierbei besteht die grundsätzliche Idee darin, für ein bestimmtes Pixel die diffus reflektierten Helligkeitsanteile der Schnittpunkte eines Sehstrahles und der durch spiegelnde Reflexion verursachten Reflexionsstrahlen mit den Objekten einer Szene in dem entsprechenden Eintrag eines Pixelstatus-Speichers auf den verschiedenen Prozessoren einzusammeln. Sobald die Anteile sämtlicher Schnittpunkte bekannt sind, kann die endgültige Pixel-Intensität bestimmt werden.

Ein Bildbereich, für den ein (Teil-) Pixelstatus-Speicher generiert wurde, ist eindeutig in der Datenstruktur des Pixelstatus-Speichers definiert. Dieses ist notwendig, um abschließend die berechneten Pixel-Intensitäten an die korrekten Positionen in den Bildspeicher zu kopieren. Weiterhin werden für jedes Bildelement (Pixel) innerhalb des Pixelstatus-Speichers die folgenden Informationen gespeichert:

- Die Pixel-Intensität, die durch die Prozessoren im Berechnungsverlauf aktualisiert wird.
- Eine Liste mit global eindeutigen Verweisen auf Objektoberflächen des Intersection-Netzes, die durch den Sehstrahl des Pixels und die erzeugten Reflexionsstrahlen auf rekursiv verfolgten Strahlenbahn getroffen werden. Die Objektverweise werden in dieser Liste in der Reihenfolge gespeichert, in der sie auf der Strahlenbahn durch den Sehstrahl bzw. die erzeugten Reflexionsstrahlen getroffen werden.
- Eine Liste mit Distanzen. Diese Werte geben die Distanz von dem Ursprung eines Seh- bzw. Reflexionsstrahles bis zum nächsten Schnittpunkt mit einer Objektoberfläche der

Szene auf dem entsprechenden Pfadabschnitt der rekursiv verfolgten Strahlenbahn an. Mit Hilfe dieser Information und der Richtung des Sehstrahls läßt sich der vollständige Strahlenpfad innerhalb der Szene bis zur maximalen Rekursionstiefe reproduzieren, ohne daß Schnittpunktberechnungen durchgeführt werden müssen. Hierbei müssen lediglich die Richtungsänderungen des Strahls gemäß der Materialeigenschaften des getroffenen Objektes berücksichtigt werden. Dieses ist ein wesentlicher Aspekt der datenparallelen Berechnungen, da die Strahlenbahnen mehrfach nachvollzogen werden müssen, um ggf. fehlende diffuse Helligkeitsanteile ermitteln zu können. Die Schnittpunkte werden nicht explizit in dem Pixelstatus-Speicher abgelegt, da dieses bei der Kommunikation zwischen den Prozessoren zu erheblich größeren auszutauschenden Datenmengen führen würde. Die Rekonstruktion der Schnittpunkte mit Hilfe der gespeicherten Distanzwerte erfordert hingegen nur einige wenige zusätzliche arithmetische Operationen pro Schnittpunkt.

- Eine Liste mit booleschen Werten, die für alle gefundenen Schnittpunkte auf der Strahlenbahn des Sehstrahles und der reflektierten Strahlen angeben, ob der entsprechende diffus reflektierte Helligkeitsanteil eines Schnittpunktes in der aktuell berechneten Pixel-Intensität schon enthalten ist. Diese Liste wird von den Prozessoren im Verlauf der parallelen Bestimmung der endgültigen Pixel-Intensität aktualisiert.

Die parallele Bildgenerierung wird in zwei separate Berechnungsabschnitte unterteilt. In dem ersten Berechnungsabschnitt berechnen die einzelnen Prozessoren für jedes Pixel der zugeordneten Bildbereiche die Schnittpunkte der Sehstrahlen und der jeweils erzeugten Reflexionsstrahlen mit den Oberflächen der Szene und speichern Verweise auf die getroffenen Oberflächen sowie die ermittelten Distanzwerte in lokalen Pixelstatus-Speichern. In dem zweiten Berechnungsabschnitt tauschen die Prozessoren die Pixelstatus-Speicher gemäß eines festen Kommunikationsschemas aus und berechnen die verteilt gespeicherten diffus reflektierten Helligkeitsanteile der zuvor bestimmten Schnittpunkte.

Da jeder Prozessor über eine vollständige geometrische Repräsentation der Szene in Form des Intersection-Netzes verfügt, können die Schnittpunkte auf den rekursiv verfolgten Strahlenbahnen in dem ersten Berechnungsabschnitt jeweils lokal ohne Kommunikation zwischen den Prozessoren bestimmt werden. Wie zuvor beschrieben, werden für die Pixel der Teilbereiche der virtuellen Bildebene Sehstrahlen generiert, die zur Bestimmung der an den entsprechenden Bildpositionen sichtbaren Objekte benötigt werden. Die Bestimmung der sichtbaren Objekte erfolgt hierbei auf Basis des Intersection-Netzes. Die globalen Objektnummern dieser Objekte werden dazu verwendet, die entsprechenden lokalen Objekte im Radiosity-Netz zu identifizieren. Die auf einem Prozessor gespeicherten Patches eines getroffenen Objektes werden auf einen Schnittpunkt mit dem primären Strahl überprüft. Sollte ein Schnittpunkt mit einem lokal gespeicherten Patch des Objektes existieren, wird der Radiosity-Wert an der Stelle des Schnittpunktes durch bilineare Interpolation der Eckpunkt-Radiosities des getroffenen Patches bestimmt. Andernfalls wird in dem entsprechenden Eintrag des entsprechenden Pixelstatus-Speichers vermerkt, daß bisher kein diffus reflektierter Helligkeitsanteil für den Schnittpunkt berechnet werden konnte. Die rekursive Strahlenverfolgung wird wie zuvor beschrieben fortgesetzt, um die getroffenen Objekte nach jeder spiegelnden Reflexion oder Refraktion basierend auf dem Intersection-Netz zu identifizieren. Nach der Identifizierung eines getroffenen Objektes wird jeweils versucht, für den Schnittpunkt einen diffus reflektierten Intensitätsanteil basierend auf den lokal verfügbaren Radiosity-Informationen zu ermitteln. Diese Intensitäts-Information wird bei Verfügbarkeit direkt auf die im Pixelstatus-Speicher verwaltete Pixel-Intensität mit dem Reflexionskoeffizienten gewichtet addiert. Wenn die In-

formation lokal nicht verfügbar ist, wird dieses in dem Pixelstatus-Speicher vermerkt (boolsche Information).

Aufgrund der verteilten Berechnung und Speicherung der Radiosity-Informationen (Aufteilung des Radiosity-Netzes) kann jedoch der Fall auftreten, daß nicht alle Intensitäts-Werte für die gefundenen Schnittpunkte der Seh- oder Reflexionsstrahlen mit den Objekten des Intersection-Netzes lokal ermittelt werden konnten. Dieses ist an den entsprechenden Stellen im Pixelstatus-Speicher vermerkt. Die Ermittlung der fehlenden diffusen Intensitäts-Werte erfolgt im zweiten Berechnungsabschnitt. Die Prozessoren kommunizieren gemäß einer virtuellen Ring-Topologie. Über diesen virtuellen Ring verschickt jeder Prozessor seine berechneten bzw. aktualisierten Pixelstatus-Speicher an seinen Nachfolgerprozessor auf dem virtuellen Ring (siehe Abb. 5.59).

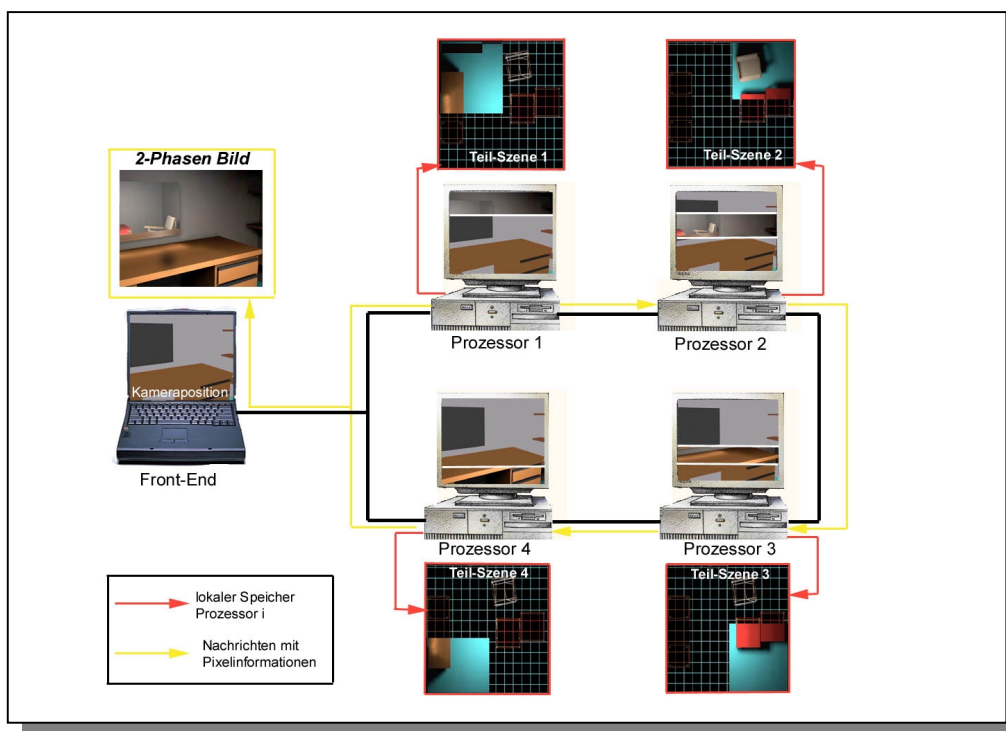


Abb. 5.59: Parallele Bildgenerierung auf Basis einer verteilt gespeicherten Radiosity-Lösung unter Berücksichtigung von spiegelnden Objekten.

Nach erhalten eines Pixelstatus-Speichers versucht jeder Prozessor, die fehlenden diffusen Intensitätsanteile der gefundenen Schnittpunkte basierend auf den lokal verfügbaren Radiosity-Informationen zu bestimmen. Hierbei ist zu beachten, daß keine erneute Berechnung der auf einem Strahlenpfad getroffenen Objekte erfolgen muß, da dieses schon im ersten Berechnungsabschnitt geschehen ist und die globalen Objektnummern von getroffenen Objekten vollständig in den Einträgen der Pixelstatus-Speicher abgelegt wurden. Ein Prozessor durchsucht den empfangenen Pixelstatus-Speicher nach Objekten, die einen Schnittpunkt mit einem Seh- oder Reflexionsstrahl besitzen, für welchen bisher kein diffuser Intensitätsanteil ermittelt werden konnte. Die lokal gespeicherten Patches dieses Objektes werden auf einen Schnittpunkt mit dem entsprechenden (rekonstruierten) Strahl überprüft. Wenn ein bisher fehlender Intensitätswert verfügbar ist, wird die entsprechende Information im Pixelstatus-Speicher angepaßt. Spätestens wenn ein Pixelstatus-Speicher den gesamten virtuellen Ring passiert hat

und alle Prozessoren ihre lokal verfügbaren Informationen in den Pixelstatus-Speicher eingetragen haben, sind alle diffus reflektierten Helligkeitsanteile der gespeicherten Schnittpunkte berechnet und die endgültigen Pixel-Intensitäten bestimmt worden. Somit sind bei n Prozessoren $n-1$ Austauschoperationen auf dem virtuellen Ring notwendig. Im Anschluß an die $n-1$ -te Austausch- und Aktualisierungsoperation der Pixelstatus-Speicher können die Prozessoren die Intensitäts-Werte der Pixel des jeweils verfügbaren Pixelstatus-Speichers mit Hilfe der *Gamma-Korrektur* [FvDFH90] in Farbwerte transformieren und den entsprechenden Bildbereich an die Visualisierungskomponente übermitteln. Hierbei enthält die Ergebnismeldung Informationen über die Position des Bildbereiches innerhalb des virtuellen Bildschirms, sowie eine Menge von Farbwerten für aufeinanderfolgende (zeilenweise) Pixel.

Eine wesentliche Rolle bei der Generierung von qualitativ hochwertigen Bildern spielt das sogenannte *Anti-Aliasing*. Aufgrund der Diskretisierung der Bildebene in eine endliche Anzahl von diskreten Bildpunkten (Pixel) ist die Darstellung von schräg verlaufenden Kanten (Linien) ein großes Problem. Hier tritt das sogenannte *Aliasing* auf, das durch ein regelmäßiges Abtastungsmuster (gitterförmige Diskretisierung der Bildebene) verursacht wird. Je nach Bildauflösung sind mehr oder weniger starke sogenannte Treppenstufen-Effekte an diesen Kanten zu beobachten. Dieses beeinträchtigt den Realitätsgrad der Bilder erheblich. Zur Reduzierung dieser unschönen Effekte wurden in der Vergangenheit verschiedene *Anti-Aliasing* Techniken eingeführt (siehe [Glas89]). Das Grundprinzip dieser Techniken basiert darauf, daß nicht nur ein Sehstrahl pro Bildelement (Pixel) generiert wird, sondern mehrere Sehstrahlen durch die endliche Fläche des Bildelements gesendet werden und die endgültige Pixel-Intensität als Durchschnittswert der ermittelten Intensitäten gesetzt wird. Das hier vorgestellte Programm unterstützt zwei verschiedene Methoden zur Verteilung der Sehstrahlen auf der Fläche eines Pixels. Bei einem statischen *Anti-Aliasing* werden maximal fünf Sehstrahlen pro Pixel generiert und rekursiv innerhalb der Szene verfolgt. Die generierten Sehstrahlen schneiden den Mittelpunkt sowie die vier Eckpunkte des entsprechenden Pixels. Bei einem zufälligen *Anti-Aliasing* wird eine fest vorgegebene Anzahl von Sehstrahlen pro Pixel erzeugt und rekursiv verfolgt.

Es ist zu beachten, daß sich bei der Durchführung von einem statischen oder zufälligen *Anti-Aliasing* die Menge der in den Pixelstatus-Speichern abgelegten Informationen linear mit der Anzahl der erzeugten Sehstrahlen pro Pixel vergrößert. Somit wird auch die während der parallelen Berechnung auszutauschende Datenmenge mitunter erheblich vergrößert.

Aufgrund der gewählten statischen Lastverteilungsstrategie und der datenparallelen Berechnungen können sich Wartezeiten der Prozessoren ergeben, die eine geringere Effizienz der parallelen Bildgenerierung verursachen. Im Idealfall bekommt jeder Prozessor die gleiche Anzahl von Bildzeilen zur Berechnung zugeordnet. Ein Ungleichgewicht der statischen Last entsteht, wenn die Anzahl der Bildzeilen nicht ohne Rest durch die Anzahl der Prozessoren geteilt werden kann. Dann bekommen einige Prozessoren mehr Zeilen zugewiesen als andere. Weiterhin kann Lastungleichgewicht durch deutlich abweichende Materialeigenschaften der auf den Strahlenbahnen getroffenen Objekte entstehen. Aufgrund der Materialeigenschaften der getroffenen Objekte ergeben sich Strahlenbahnen unterschiedlicher Länge, wodurch der erste Berechnungsabschnitt auf den Prozessoren unterschiedlich lang dauert. Zusätzlich kann die Nachberechnung der fehlenden diffus reflektierten Helligkeitsanteile der in den Pixelstatus-Speichern abgelegten Schnittpunkte zwischen den Prozessoren stark variieren. Abhilfe kann hier zukünftig nur eine dynamische Lastverteilungsstrategie analog zu der in 5.2.8 vorgestellten Methode schaffen.

Der wesentliche Vorteil der beschriebenen Methode zur Bildgenerierung unter Berücksichtigung von Spiegelungen innerhalb der Szene besteht darin, daß die Methode direkt in die datenparallele Progressive-Refinemente-Methode integriert ist. Hierdurch ist es möglich, sowohl Zwischenergebnisse als auch die endgültige Lösung der Radiosity-Berechnungen zu visualisieren, ohne daß hierzu die Daten explizit auf ein externes Speichermedium geschrieben werden und von dort für die parallele Bildgenerierung wieder einzulesen. Weiterhin ist aufgrund der datenparallelen Berechnungen die Größe der Szenen nicht mehr durch den zur Verfügung stehenden Hauptspeicher eines Prozessors limitiert.

Kapitel 6

Systemintegration

Auf Basis der in Kapitel 5 eingeführten parallelen Verfahren ist ein interaktives System zur Bildgenerierung realisiert worden. Dieses System ermöglicht die Durchführung von Online-Begehungen (*Walkthrough*) von virtuellen Umgebungen, für welche mit Hilfe der integrierten datenparallelen Progressive-Refinement-Methoden eine globale Beleuchtungssimulation durchgeführt wurde. Entlang von frei definierbaren Kamerapfaden kann im Anschluß an eine Begehung die parallele Berechnung von qualitativ hochwertigen Einzelbildern sowie Animationen erfolgen, wobei die im letzten Kapitel beschriebene Zwei-Phasen-Methode (Abschnitt 5.2.8) verwendet wird.

Innerhalb dieses Kapitels wird ein kurzer Überblick über die Architektur des Bildgenerierungssystems sowie dessen Funktionalität gegeben. Weiterhin wird auf eine wesentliche Erweiterung des Systems eingegangen, welche es Anwendern erlaubt, die parallelen Verfahren als Dienstleistung über IP-Netzwerke (Internet) zu nutzen.

6.1 Architektur und Komponenten des Bildgenerierungssystems

Das Bildgenerierungssystem ist in Form einer modularen Client/Server-Architektur realisiert worden [SL97]. Die Architektur des Systems ist in Abb. 6.1 skizziert.

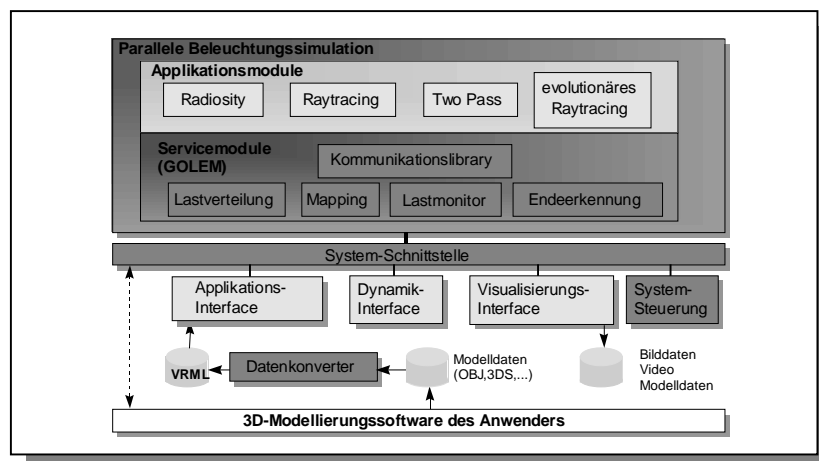


Abb. 6.1: Architektur des Bildgenerierungssystems.

Bei den Komponenten des Bildgenerierungssystems wird unterschieden zwischen *Client-Komponenten*, die auf Workstations in einem lokalen Netzwerk gestartet werden und der auf

einem Parallelrechner ausführbaren *Server-Komponente*, welche Berechnungsaufträge von Clients bearbeitet und die Ergebnisse den entsprechenden Visualisierungs-Clients übergibt. Bei diesen Berechnungsaufträgen handelt es sich entweder um eine blickpunktunabhängige Radiosity-Berechnung oder um die Erzeugung von Einzelbildern und Animationen mit Hilfe von Raytracing bzw. einer Zwei-Phasen-Methode. Über die Systemsteuerung kann ein Anwender zwischen den verschiedenen unterstützten globalen Beleuchtungsverfahren auswählen und sowohl die entsprechenden Client-Komponenten auf den Frontend-Systemen starten, als auch die Server-Komponente initialisieren und die parallelen Berechnungen starten.

Das System stellt eine integrierte Gesamtlösung zur Bildgenerierung unter Berücksichtigung der globalen Beleuchtungseffekte in komplexen Szenen dar. Verschiedene parallele Verfahren zur Beleuchtungssimulation wie Radiosity, Standard-Raytracing, evolutionäres Raytracing, Zwei-Phasen-Methode und Photontracing wurden in die Server-Komponente integriert. Dadurch wird eine Simulation der wesentlichen, durch spiegelnde und diffuse Reflexion entstehenden Beleuchtungseffekte ermöglicht.

Radiosity-Berechnungen werden unabhängig von einem Betrachterstandpunkt durchgeführt. Somit sind die Simulationsergebnisse für eine virtuelle Begehung einer Szene (*Walkthrough*) geeignet und es können hierbei Beleuchtungseffekte visualisiert werden, die durch diffuse Interobjektreflexionen entstehen. Auf Anfrage des Visualisierungs-Clients werden Zwischenergebnisse der Radiosity-Berechnung von der Server-Komponente übermittelt, so daß die Beleuchtungswerte der Objektoberflächen während der Begehung der Szene aktualisiert werden. Durch diese Vorgehensweise wird die Qualität der visualisierten Radiosity-Lösungen ständig verbessert. Der Benutzer kann während der parallelen Radiosity-Berechnungen mit der Szene interagieren, wobei die globale Beleuchtung inkrementell angepaßt und die veränderte Beleuchtungssituation mit geringer Verzögerung angezeigt wird. Weiterhin ist es möglich, einen Animationspfad zu definieren, um anschließend eine Animation auf Basis der einfachen Zwei-Phasen-Methode (siehe 5.2.8) zu erzeugen. Das System ist aufgrund seiner Konzeption für die Nutzung von parallelen Systemen mit verteiltem Speicher und den effizienten Parallelisierungen der Beleuchtungsverfahren bezüglich der Rechenleistung sehr gut skalierbar.

Der Austausch von Modelldaten zwischen der Modellierungssoftware eines Anwenders und dem hier beschriebenen Bildgenerierungssystem wird auf Basis von Standard-Dateiformaten realisiert (z.B. 3DS, OBJ oder VRML), die mit Hilfe eines Datenkonverters in ein internes Datenformat umgewandelt werden. Das interne Datenformat kann anschließend von einem Applikations-Interface interpretiert werden. Eine direkte Anbindung der Modellierungssoftware als Client ist über die offene Systemschnittstelle möglich. Die Voraussetzung hierfür ist jedoch die Verfügbarkeit des Quellcodes der Modellierungssoftware.

Das Bildgenerierungssystem ist in die parallele Entwicklungs- und Anwendungsumgebung GOLEM eingebettet. GOLEM wurde im Rahmen des BMBF-Projektes PARAGRAPH mit dem Ziel entwickelt, eine von der Hardwarearchitektur eines Parallelrechnersystems unabhängige Applikationsentwicklung zu unterstützen und diese Applikationen anschließend ohne tiefgehendes Expertenwissen im Bereich des parallelen Rechnens über eine graphische Benutzerschnittstelle bedienen zu können.

Bei der Entwicklung und der Ausführung von parallelen Applikationen treten immer wieder Standardaufgaben auf, wie z.B.

- das Starten und die Initialisierung einer parallelen Applikation,

- die Abbildung eines Prozeßgraphen auf eine Prozessortopologie (Einbettung),
- Interprozeßkommunikation auf einem Parallelrechner
- Kommunikation zwischen interaktiven Benutzerschnittstellen auf Frontend-Rechnern und Berechnungsprozessen auf Parallelrechnersystemen,
- verteilte Ende-Erkennung,
- Monitoring und
- dynamische Lastverteilung.

Diese Standardfunktionalitäten werden durch GOLEM in Form von Funktions-Bibliotheken (*Libraries*) und Benutzungsschnittstellen unterstützt. Diese können zur Vereinfachung der Entwicklung von beliebigen parallelen Applikationen verwendet werden. Aufgrund der Nutzung von Kommunikationsstandards wie PVM und TCP/IP-Sockets ist die Unterstützung einer großen Anzahl von Rechnersystemen und Betriebssystemen gewährleistet, wodurch sich die Portierung auf neue Parallelrechnersysteme im allgemeinen sehr schnell und einfach realisieren läßt.

Durch den modularen Aufbau des Bildgenerierungssystems, offene Systemschnittstellen und einer von der Architektur eines Parallelrechnersystems unabhängigen Implementierung auf Basis von GOLEM lassen sich Erweiterungen des Systems auf einfache Art und Weise durchführen. Bei den Erweiterungen kann es sich sowohl um die Integration neuer Verfahren zur parallelen Beleuchtungssimulation auf Seiten des Servers, als auch um neue Benutzungsschnittstellen in Form von Clients handeln. Somit kann das Bildgenerierungssystem schnell an individuelle Anforderungen von neuen Anwendern und an deren Hardwareressourcen angepaßt werden.

6.2 Beleuchtungssimulation und Bildgenerierung als Dienstleistung in IP-Netzwerken

In vielen Bereichen der Medienproduktion, der Architektur und dem Produktdesign wird dem Einsatz von computergenerierten Animationen, QuickTime-VR und anderen VR-Anwendungen ein hoher Stellenwert eingeräumt. In den letzten Jahren wurde die Komplexität der virtuellen Szenarios deutlich umfangreicher und die Anforderungen an den Realitätsgrad der synthetisch erzeugten Bilder und Animationen stiegen an. Aus diesem Grund sind die Modellierungs-, Animations- und Schattierungs- Fähigkeiten von 3D-Animationssoftware erheblich verbessert worden. Dieses hat jedoch zur Folge, daß die erforderliche Rechenleistung für komplexe Animationssequenzen eine neue Größenordnung annimmt und Bildfolgen somit durch herkömmliche Hardware nicht mehr in einer akzeptablen Zeitspanne erzeugt werden können.

Dienstleister im Bereich der Erstellung von Computeranimationen (Visualisierungsbüros, Werbeagenturen, Designer, etc.) verwenden in den meisten Fällen kleine Workstations und PCs. Mit deren Hilfe sind sie in der Lage, die den Animationen zugrundeliegenden Szenenbeschreibungen zu erstellen und Einzelbilder mit hoher Qualität unter großem Zeitaufwand zu generieren. Die Erzeugung einer Animationssequenz von mehreren Sekunden mit 25 hochauflösenden Bildern pro Sekunde ist mit einer solchen Hardwareausstattung eine extrem zeitaufwendige Aufgabe, die unter Umständen mehrere Tage oder sogar Wochen in Anspruch nehmen kann. Während dieser Zeit ist die Verfügbarkeit von Rechnerkapazitäten für die Bearbeitung von weiteren Produktionen stark eingeschränkt. Um die neuen Möglichkeiten der Modellierungs- und Bildgenerierungs-Softwar optimal zu nutzen, sind die Unternehmen dazu gezwungen, immer leistungsfähigere Hardware zu beschaffen.

Ein wesentlicher Nachteil von performanten Parallelrechnern ist der Preis. Dieses führt dazu, daß der Anwenderkreis solcher HPC-Systeme im allgemeinen auf große Unternehmen und Forschungseinrichtungen beschränkt ist, da Investitionen in dieser Größenordnung für kleine mittelständische Unternehmen nicht tragbar sind.

Eine Trennung von technischer Infrastruktur und kreativer Anwendung würde es den Dienstleistern der Medienindustrie ermöglichen, sich auf die laufenden Produktionen zu konzentrieren, anstatt auf technische Aspekte. Eine solche Trennung ist durch die Verfügbarkeit des globalen Internets als Dienstleistung (*Rendering-Service*) realisierbar, wobei eine auflösungs- und formatunabhängige Bildgenerierung auf Hochleistungsrechnern (Parallelrechner, Mainframes, eng gekoppelte Workstation-Cluster) durchgeführt wird, die mit spezieller paralleler Rendering-Software ausgestattet sind. Diese Art von Dienstleistung ist durch die Bereitstellung eines parallelen *Rendering-Servers* möglich, der die zur Verfügung stehende Rechenkapazität vernetzter Hoch- und Höchstleistungsrechner nutzt, um termingerecht qualitativ hochwertige Bilder, Animationen oder beleuchtete VR-Modelle zu berechnen. Um eine breite Akzeptanz einer solchen Dienstleistung in den Bereichen der Medienproduktion, Architekturvisualisierung und Design zu erreichen, müssen durch den Rendering-Server gewisse Kriterien erfüllt werden:

- Durch den Rendering-Server müssen verschiedene Bildgenerierungsverfahren verfügbar gemacht werden, die den Qualitätsanforderungen von professionellen Anwendern genügen. Die bekanntesten derzeit eingesetzten Verfahren zur photorealistischen Bildgenerierung sind durch einen enormen Bedarf an Rechenzeit und Speicherplatz gekennzeichnet. Somit können diese Verfahren bei vollem Leistungsumfang und komplexen Szenen nur auf Hardwareplattformen der höheren Preisklasse effizient eingesetzt werden. Das komplette Leistungsspektrum solcher Verfahren kann mit Hilfe eines parallelen Rendering-Servers, welcher Zugriff auf eine Menge von dezentralen Hochleistungsrechnern hat, einem breiten Anwenderfeld kostengünstig und einfach zur Verfügung gestellt werden.
- Es ist davon auszugehen, daß die potentiellen Benutzer des parallelen Rendering-Servers unterschiedliche Modellierungssoftware zur Erzeugung von Modelldaten einsetzen und verschiedene Ausgabeformate für die Weiterverarbeitung benötigen. Aus diesem Grund muß ein Rendering-Server mit einer flexiblen und einfach zu erweiternden Schnittstelle zur Verarbeitung der verschiedenen Ein- und Ausgabeformate sowie effizienten Verfahren zur Datenkompression ausgestattet sein.
- Medienproduktionen sind in den meisten Fällen an vorgegebene Fertigstellungstermine gebunden. Die Vorteile der Parallelverarbeitung können durch einen parallelen Rendering-Server bei der Generierung photorealistischer Computeranimationen ausgenutzt werden, um eine fristgerechte Fertigstellung zu garantieren. Ein auf Raytracing basierendes Bildgenerierungsverfahren läßt sich effizient parallelisieren, wodurch Einzelbilder einer Animation auf einem Parallelrechner deutlich schneller generiert werden können, als auf sequentiellen Rechnern. Der Berechnungsvorgang einer kompletten Animation läßt sich noch weiter beschleunigen, indem durch den Rendering-Server die Animationserzeugung in Teilaufgaben zerlegt wird, die auf verschiedene, durch Breitbandnetze verbundene, heterogene Hochleistungsrechner verteilt werden. Hierbei können die Teilaufgaben mit gewissen Prioritäten versehen und gemäß dieser Prioritäten auf die verfügbaren Ressourcen verteilt werden.

- Eine einfache Nutzbarkeit des Rendering-Services wird einen sehr großen Einfluß auf dessen Akzeptanz durch den Anwender haben. Mit Hilfe des WWW kann einem breiten Anwenderkreis ein Zugriff auf die Dienstleistung ermöglicht werden, indem eine Client-Infrastruktur zur Fernsteuerung der Rendering-Operationen geschaffen wird.

Ein *Remote-Rendering-System*, welches die effizienten parallelen Methoden zur globalen Beleuchtungssimulation und photorealistische Bildgenerierung als E-Commerce-Dienstleistung über das Internet zugänglich macht, wurde auf Basis des in 6.1 beschriebenen Client/Server-Systems entwickelt. Das System ist im Rahmen des EU-Forschungsprojektes HiPEC (High Performance Visualization System supporting Networked Electronic Commerce Applications) für eine spezielle E-Commerce-Applikation aus dem Anwendungsbereich der Produktkonfiguration und Produktvisualisierung realisiert worden [SL98]. Dieses Dienstleistungsszenario wurde in dem BMBF-Projekt HiQOS auf weitere Anwendungsfelder ausgedehnt [SP00]. Die Nutzung dieser Dienstleistung ermöglicht einer großen Anzahl von Anwendern aus den unterschiedlichsten Bereichen einen Zugriff auf leistungsfähige parallele Verfahren und die entsprechenden Hardwareressourcen.

6.2.1 Architektur des Remote-Rendering-Systemes

Das hier vorgestellte Remote-Rendering-System ermöglicht sowohl die vom Standpunkt eines Betrachters unabhängige Simulation globaler Beleuchtungseffekte in virtuellen Umgebungen auf Basis einer datenparallelen Radiosity-Methode, als auch eine schnelle parallele Generierung von Einzelbildern und Animationssequenzen durch ein paralleles Raytracing-Verfahren oder durch eine Zwei-Phasen-Methode.

Das System hat eine hierarchische Architektur. Die wesentlichen Komponenten sind das Client-System, das HPC-Frontend und das HPC-Backend (siehe Abb. 6.2).

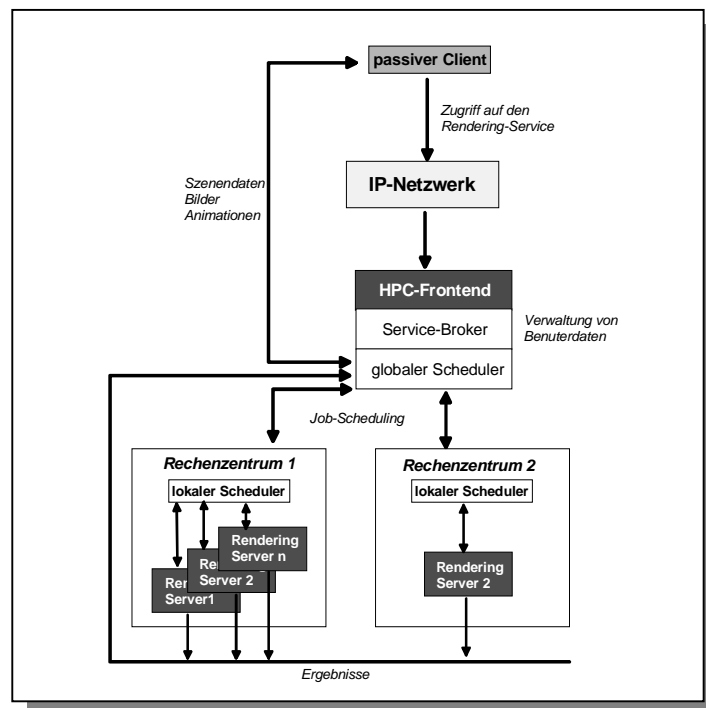


Abb. 6.2: Architektur des *Remote-Rendering-Systems*.

6.2.1.1 Clients

In der derzeitigen Implementierung sind Client-Systeme passiv. Dieses bedeutet, daß sich ein Kunde der Rendering-Dienstleistung explizit auf einer Internet-Seite des Remote-Rendering-Systems anmeldet, die Auftragsdaten auf einem externen Speichermedium plaziert und anschließend eine Referenz (URL) auf diese Daten sowie Eingabeparameter für den Rendering-Server auf der WWW-Seite des Remote-Rendering-Systems spezifiziert. Zukünftig sind aktive Clients in Form von *Plug-Ins* für Standard-Modellierungssoftware (z.B. 3D-StudioMAX) oder als Java-Applets denkbar. In diesem Fall werden die entsprechenden Auftragsdaten direkt von dem lokalen System eines Anwenders über IP-Netzwerke (d.h. Internet) an das Remote-Rendering-System übertragen.

6.2.1.2 HPC-Frontend

Das HPC-Frontend ist die Schnittstelle zwischen dem Client und dem eigentlichen Rendering-System. Es besteht aus zwei separaten Modulen, dem *Service-Broker* und dem *globalen Scheduler*.

Der Service-Broker ist ein Internet-Server, dessen Aufgabe in der Administration von Anwendern und der Durchführung von betriebswirtschaftlichen Aktivitäten besteht. Der Endanwender meldet sich auf einer Internet-Seite des Rendering-Systems an und spezifiziert mit Hilfe eines speziellen Formulars die Steuerungs-Parameter für die Auftragsbearbeitung sowie die Position der Auftragsdaten (Szenenbeschreibung, Texturen, Kamerapositionen, Materialbeschreibungen). Jedem eingehenden Auftrag wird ein eindeutiger Auftragsbezeichner zugewiesen. Im Anschluß daran wird der Berechnungsauftrag an den globalen Scheduler weitergeleitet.

Der globale Scheduler ist verantwortlich für die Abwicklung von Berechnungsaufträgen. Hierzu werden zunächst die Auftragsdaten von der spezifizierten Netz-Adresse (URL) geladen, die Szenenbeschreibung in ein internes Dateiformat umgewandelt und im Anschluß daran wird der gesamte Auftrag in eine Warteschlange eingefügt. Der globale Scheduler verwaltet Informationen über den Status sämtlicher in den angeschlossenen Rechenzentren verfügbaren Hardwareressourcen. Auf Basis dieser Informationen werden Aufträge aus der Auftragswarteschlange auf freie Ressourcen zur Abwicklung verteilt.

6.2.1.3 HPC-Rendering

Das grundlegende Konzept des Remote-Rendering-Systems erlaubt es, daß die verwendeten Parallelrechner in geographisch getrennten Rechenzentren verteilt installiert sind. Die Rechner können sich sowohl hinsichtlich ihrer Leistungsfähigkeit als auch ihrer Systemarchitektur grundsätzlich unterscheiden. Die Nutzung dieser Hardware-Ressourcen im Rahmen der Rendering-Dienstleistung wird mit Hilfe von zwei Software-Komponenten realisiert, die in jedem Rechenzentrum installiert sind und an die jeweilige Umgebung angepaßt werden müssen. Hierbei handelt es sich um den *lokalen Scheduler* und den *Rendering-Server*.

Die Aufgabe eines lokalen Schedulers besteht darin, die innerhalb eines Rechenzentrums lokal verfügbaren Hardwareressourcen zu kontrollieren und deren Status an den globalen Scheduler zu übermitteln sowie die lokale Bearbeitung von zugeteilten Aufträgen zu initiieren und den Berechnungsverlauf zu überwachen. Um diese Aufgaben zu erfüllen, wird durch den lokalen Scheduler die in den Rechenzentren jeweils installierte Verwaltungs-Software der Parallelrechner verwendet (z.B. *CCS – Computing Center Software* in Paderborn [KR98]).

Der Rendering-Server entspricht dem im 6.1 vorgestellten Client/Server-System zur parallelen globalen Beleuchtungssimulation und Bildgenerierung. Im Kontext des Remote-Rendering-Systems werden hierbei jedoch die interaktiven Client-Systeme zur Steuerung der parallelen Berechnungen in einem Offline-Modus, gemäß der durch den Anwender spezifizierten Parameter, betrieben. Die Server-Komponenten werden jeweils für verschiedene Rechnerarchitekturen angepaßt und übersetzt, so daß ein lokaler Scheduler alle verfügbaren Hardware-Ressourcen eines Rechenzentrums nutzen kann, um die zugeteilten Aufträge zu bearbeiten. Hierbei kann ein lokaler Scheduler zu jedem Zeitpunkt mehrere Rendering-Server starten und den Berechnungsverlauf überwachen.

Um das beschriebene Remote-Rendering-System einer großen Anzahl von Endanwendern zur Verfügung zu stellen, wird die Dienstleistung über das Internet angeboten. Der gravierende Nachteil hierbei ist die geringe Bandbreiten, die den Benutzern zur Verfügung stehen und die dadurch verursachten langen Kommunikationszeiten. Ein wichtiger Aspekt zur Reduktion dieser Zeiten ist die Kompression der Berechnungsergebnisse, da diese den wesentlichen Anteil des Kommunikationsvolumens zwischen Client-Systemen und dem Remote-Rendering-System ausmachen.

Im Fall der Generierung von Einzelbildern und Animationssequenzen können Kompressionsstandards wie JPG und MPEG verwendet werden, um die Datenmenge zu reduzieren, die von dem Rendering-System über das Internet an Kunden transferiert wird. Schwieriger gestaltet sich die Situation, wenn adaptiv verfeinerte Polygonnetze übertragen werden müssen, die während der parallelen Radiosity-Berechnungen erzeugt wurden. Das resultierende Netz ist um ein Vielfaches größer als das ursprüngliche Eingabernetz, da zusätzlich zu den geometrischen Informationen noch Beleuchtungsinformationen für jedes Polygon gespeichert werden.

Im Kontext der hier beschriebenen Dienstleistung werden zwei verschiedene Techniken zur Kompression von Radiosity-Netzen getestet und eingesetzt, um eine schnelle und qualitativ gute Reduktion der Netzkomplexität zu erreichen. Bei der ersten Methode handelt es sich um ein datenparalleles Verfahren, welches die Anzahl der in einem Radiosity-Netz enthaltenen Polygone um 80 bis 90 Prozent reduziert, ohne das durch die Entfernung von Polygonen signifikante sichtbare Fehler entstehen [SR00]. Bei dem zweiten Ansatz werden Radiosity-Texturen für Objektoberflächen berechnet. Die grundlegende Idee besteht darin, Beleuchtungsinformationen in Texturen (Bitmaps) zu speichern und diese auf die Original-Objekte (ohne adaptive Substrukturierung) abzubilden. Anstatt eines adaptiv verfeinerten Radiosity-Netzes müssen in diesem Fall lediglich die Texturen sowie Abbildungsvorschriften für die Texturen an die Endanwender übertragen werden. Diese können die Objekte der ursprünglichen Eingabeszene mit den Radiosity-Texturen belegen. Beide Ansätze tragen erheblich zur Steigerung der Performanz der Rendering-Dienstleistung und somit der Anwenderakzeptanz bei, indem der Kommunikationsaufwand deutlich reduziert wird.

6.2.2 Unterstützte Anwendungsszenarios

Wichtige kommerzielle Anwendungen der photorealistischen Bildgenerierung sind in den Bereichen der Produkt- und Architekturvisualisierung sowie der Animationserzeugung im Bereich Film und Werbung angesiedelt. Spezielle Anwendungen aus diesen Bereichen werden durch das hier beschriebene Remote-Rendering-System unterstützt.

6.2.2.1 Produktkonfiguration und -visualisierung

In einem sehr speziellen Anwendungsszenario wird die interaktive Konfiguration und Visualisierung von Badezimmern mit individuell zusammengestellten Einzelkomponenten ermöglicht [SL98]. Die wesentliche Idee besteht darin, den Verkaufsvorgang zu vereinfachen und den Entscheidungsprozeß der Kunden zu unterstützen.

Innerhalb des Ausstellungsraumes eines Einzelhändlers für qualitativ hochwertige Badezimmerausstattung wird ein PC lokal installiert. Auf diesem PC ist ein Konfigurationswerkzeug installiert, mit dessen Hilfe einzelne Objekte (Badezimmerkomponenten) aus einer Datenbank ausgewählt und in einem virtuellen Raum beliebig plaziert werden können. Die Ausdehnung des Raumes sowie die Platzierung von Türen, Fenstern und Lichtquellen kann hierbei ebenfalls individuell gestaltet werden. Es können sehr schnell verschiedene Ansichten des komponierten Badezimmers erzeugt werden, wobei ein einfaches lokales Beleuchtungsverfahren verwendet wird. Die so erzeugten Bilder ermöglichen dem Betrachter, eine erste konkrete räumliche Vorstellung zu gewinnen. Aufgrund der Verwendung von lokalen Beleuchtungsmodellen wirken diese Bilder jedoch sehr künstlich. Wenn bei der Bildgenerierung Schattenwurf, Reflexionen und Refraktionen des Lichtes berücksichtigt werden, verursacht dieses einen Zeitaufwand, der während eines Beratungsgespräches nicht vertretbar ist.

In diesem Anwendungsszenario wird eine Beschreibung des konfigurierten Badezimmers sowie frei definierte Kamerapositionen über eine ISDN-Verbindung von dem PC an den Rendering-Server übertragen. Die übermittelten Daten dienen als Eingabe für den Rendering-Server, welcher mittels parallelem Raytracing Einzelbilder erzeugt und diese an den Auftraggeber übermittelt. Aufgrund der Qualität der Bilder kann dem Kunden eine exakte Vorstellung von dem fertigen Produkt vermittelt werden (siehe Abb. 6.3).



Abb. 6.3: Raytracing-Bild eines individuell konfigurierten Badezimmers.

6.2.2.2 Architekturvisualisierung

Virtuelle Begehungen von komplexen Gebäudemodellen gewinnen im Architektur- und Bauwesen zunehmend an Bedeutung. In der Planungs- und Durchführungsphase eines Bauvorha-

bens können Computeranimationen von Architekten und Beleuchtungsplanern dazu verwendet werden, ihre Ideen sichtbar zu machen. Im allgemeinen sind Baupläne und abstrakte lichttechnische Berechnungen nicht dazu geeignet, einen plastischen Eindruck von einem Bauvorhaben zu vermitteln. Im Rahmen von Genehmigungs- und Präsentationsverfahren können computergenerierte Bilder und Animationen eingesetzt werden, um ein Bauvorhaben darzustellen. Hierbei wird die Kommunikation zwischen Architekten, Bauträgern, Behörden und Anliegern durch die allgemeine Verständlichkeit von realistisch wirkenden Bildern vereinfacht. Ein virtuelles 3D-Modell ermöglicht schon während der Entwurfs- und Wettbewerbsphase einen guten Einblick in die ersten Konzepte, wodurch Planungsfehler sehr leicht zu entdecken sind. Die Nutzung computergenerierter Bilder zur Unterstützung bei der Beleuchtungsplanung setzt die korrekte Wiedergabe der Beleuchtungsverhältnisse innerhalb real existierender oder virtueller Umgebungen voraus. Dieses bedingt den Einsatz rechenintensiver und physikalisch basierter Bildgenerierungsverfahren, wodurch der Kosten- und Zeitaufwand zur Erzeugung von Animationen um ein Vielfaches gesteigert wird.

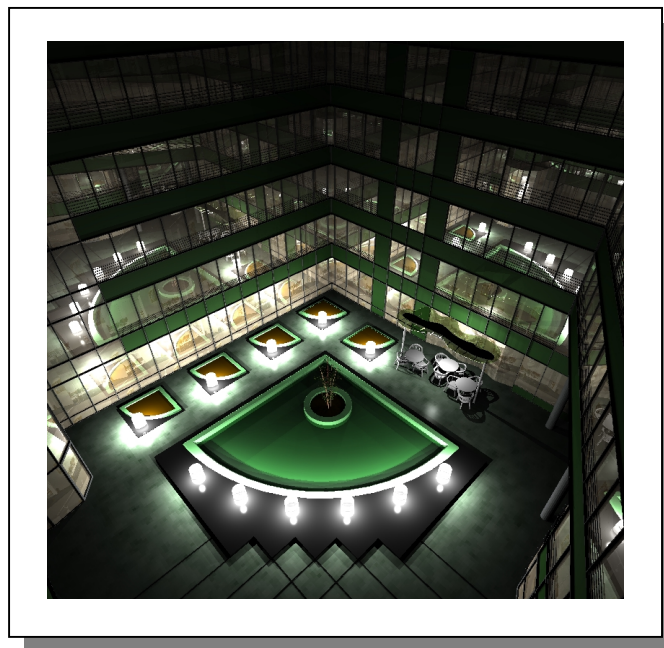


Abb. 6.4: Innenansicht des *Rosenthaler-Hofs* (Zwei-Phasen-Methode-Methode).

Mit Hilfe der in dem Rendering-Server integrierten parallelen Radiosity-Verfahren lassen sich auch für sehr komplexe Szenen globale Beleuchtungssimulationen durchführen oder qualitativ hochwertige Einzelbilder (Abb. 6.4) und längere Animationssequenzen auf Basis einer Zwei-Phase-Methode berechnen (siehe Kapitel 5). Der beschleunigte Bildgenerierungsprozeß ermöglicht den vermehrten Einsatz von computergenerierten Bildern und Animationen auch in Zwischenstadien eines Bauvorhabens.

6.2.2.3 Visuelle Effekte in der Filmindustrie

Eine sehr schnell wachsender Anwendungsbereich der Computergrafik ist die Erzeugung von visuellen Effekten bei der Produktion von Werbe-, TV- und Kinofilmen. Seit die Bilder laufen lernten, gibt es visuelle Effekte (*Visual Effects*) in der Filmindustrie. Hierbei sei nur auf Fritz Langs "*Metropolis*" verwiesen. Waren die Effekte anfangs reine Modelle und eher einfacher Natur, wie beispielsweise die Wunderkerzen als Raketenantrieb bei "*Flash Gordon*", wurden

die Modelle bis Anfang der 70er Jahre immer komplexer und erreichten ihren vorläufigen Höhepunkt in Stanley Kubriks "*Odyssee 2001*" und George Lucas "*Star Wars*".

Mit zunehmender Leistungsfähigkeit der Computer wurden die digitalen Effekte geboren. Das sind Effekte, die nicht mehr durch ein real existierendes Modell erzeugt, sondern vollständig synthetisch im Computer hergestellt werden. Einer der ersten Filme, in dem diese in größerem Umfang verwendet wurden, war "*Tron*". Allerdings waren digitale Effekte anfangs sehr teuer und wurden deshalb nur spärlich eingesetzt. Der Durchbruch kam dann mit Filmen wie "*Terminator 2*" und "*Jurassic Park*" Ende der 80er Jahre, da Hard- und Software leistungsfähiger und kostengünstiger wurden.

Heutzutage sind mit dem Computer erzeugte visuelle Effekte allgegenwärtig, sei es in der Werbung, im Kino oder im Fernsehen. Im professionellen Bereich sind dabei die Anforderungen enorm gestiegen. Zwischen virtuellen und realen Bildern kann auch der Profi nur sehr schwer unterscheiden und Photorealismus wurde zum Standard.

Die Berechnung von Animationssequenzen mit großen Bildauflösungen ist heutzutage ohne die Nutzung von Parallelverarbeitung undenkbar. Der zeitintensivste Teil ist hierbei die blickpunktunabhängige Simulation der globalen Beleuchtung in computergenerierten virtuellen Umgebungen (*Virtual Sets*). Die berechnete Lösung dient als Grundbeleuchtung der endgültigen Animation, die in einem aus mehreren Phasen bestehenden Produktionsprozeß entsteht. Die Qualität der globalen Beleuchtungssimulation ist hierbei ein kritischer Aspekt, aber die Budgets und Produktionszeiten für die Erzeugung einer Animation sind eng begrenzt. Aus diesem Grund wird die Beleuchtungsberechnung oftmals vereinfacht. Die resultierenden synthetischen Bilder sind zwar immer noch beeindruckend, obwohl die wiedergegebenen Beleuchtungseffekte oftmals weit davon entfernt sind, als physikalisch korrekt bezeichnet werden zu können. Nur einige wenige Produktionsfirmen können sich die Anschaffung von Hochleistungsrechnern und großen Workstation-Clustern leisten, um komplexe, oftmals eigenständig entwickelte, parallele Berechnungsverfahren einzusetzen.

Das Anwendungsszenario des hier beschriebenen Remote-Rendering-Systems im Bereich der Erzeugung von visuellen Effekten für Filmproduktionen besteht aus der vom Blickpunkt eines Betrachters unabhängigen Simulation der globalen Beleuchtung in komplexen virtuellen Umgebungen. Im Gegensatz zu den anderen zuvor beschriebenen Anwendungsszenarien ist das Ergebnis eines Berechnungsauftrages kein einzelnes Bild oder Animation, sondern eine dreidimensionale Beschreibung der Umgebung, welche zusätzlich zu den Geometrie- und Materialdaten noch Informationen über die Beleuchtung der Objektoberflächen enthält (Radiosity-Lösung). Diese erweiterte Szenenbeschreibung wird von den Produktionsfirmen als Grundlage eines mehrstufigen Produktionsprozesses verwendet, in welchem zusätzliche Effekte zu der berechneten globalen Beleuchtungssituation hinzugefügt werden, bis die endgültige Animationssequenz fertiggestellt ist.

Ein Beispiel für eine diffus beleuchtete virtuelle Umgebung ist in Abb. 6.5 angegeben. Dort wurde in einem Raumschiffmodell die globale Beleuchtung simuliert, die durch direkte Beleuchtung und diffuse Reflexion des Lichtes entsteht. In weiteren Produktionsphasen werden die Oberflächen mit zusätzlichen Texturen belegt und atmosphärische Effekte sowie Spiegelungen eingefügt. Diese Schritte sind jedoch von der Berechnungskomplexität um ein vielfaches geringer, als die globale Beleuchtungssimulation und können somit auf den lokalen Rechnern der Produktionsfirma ausgeführt werden.

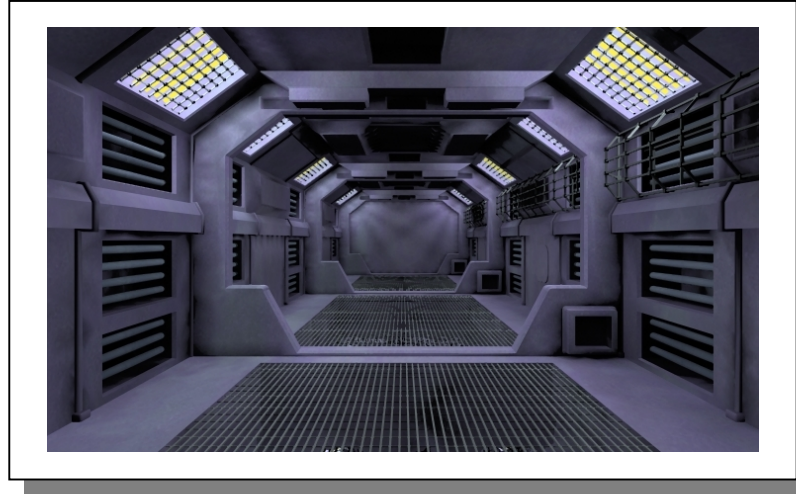


Abb. 6.5: Bildgenerierung auf Basis einer Radiosity-Lösung für ein Raumschiff-Segment.

6.2.3 Bewertung des Rendering-Services

Die Performanz und Qualität des Remote-Rendering-Services wurde zusammen mit industriellen Endanwendern aus den beschriebenen Anwendungsgebieten analysiert [SL98] [SP00].

Die vorgenommenen Tests haben belegt, daß die Rendering-Dienstleistung für eine Auftragskomplexität (Szenengröße, Animationslänge, Bildauflösungen, etc.), wie sie üblicherweise in den verschiedenen Anwendungsgebieten auftreten, sowie die Zeiten, die zur Zuteilung eines Auftrages auf die entsprechenden Hardware-Ressourcen benötigt werden, einen zu vernachlässigenden Mehraufwand durch Kommunikation zwischen Client-Systemen und dem Remote-Rendering-System verursachen.

Die in dem Rendering-Server integrierten parallelen Verfahren ermöglichen eine dramatische Steigerung der Geschwindigkeit und der Qualität bei der Simulation der globalen Beleuchtung mit Hilfe der datenparallelen Radiosity-Methode und bei der Generierung von qualitativ hochwertigen Einzelbildern und Animationen (im Vergleich zu den bisher durch die Endanwender eingesetzten sequentiellen Verfahren). Untersuchungen bzgl. der Skalierbarkeitseigenschaften der parallelen Verfahren wurden in Kapitel 5 vorgenommen.

Kapitel 7

Zusammenfassung der erzielten Ergebnisse

In diesem Kapitel wird zunächst eine kurze Zusammenfassung der wichtigsten Ergebnisse der vorliegenden Arbeit gegeben. Das Hauptaugenmerk der Betrachtungen, die bisher im Verlauf der Arbeit gemacht wurden, hat auf der Performanz der eingeführten parallelen Verfahren gelegen. Im Kontext der Bildgenerierung auf Basis von globalen Beleuchtungsverfahren ist dieses jedoch nur ein wichtiger Teilaspekt. Die Qualität der generierten Bilder steht bei den meisten Anwendungen im Vordergrund. Aus diesem Grund wird in Abschnitt 7.2 die Qualität der vorgestellten globalen Beleuchtungsverfahren anhand von repräsentativen Visualisierungen komplexer Gebäudemodelle demonstriert.

7.1 Zusammenfassung

In dieser Arbeit wurde verdeutlicht, daß durch eine Simulation der globalen Beleuchtung in virtuellen Umgebungen der Realitätsgrad von computergenerierten Bildern erheblich gesteigert werden kann. Existierende sequentielle Verfahren, welche die Wechselwirkungen der Beleuchtung von Objektoberflächen innerhalb einer Szene bei der Bildgenerierung berücksichtigen, sind gekennzeichnet durch eine große Daten- und Berechnungskomplexität. Die Parallelisierung dieser Verfahren stellt eine Möglichkeit dar, die Berechnungszeiten deutlich zu verkürzen. Durch die Entwicklung von effizienten datenparallelen Verfahren wird es ermöglicht, die Vorteile von Rechnersystemen mit verteiltem Speicher zu nutzen, um eine photorealistische Bildgenerierung unter Berücksichtigung globaler Beleuchtungseffekte auch in komplexen Szenen durchzuführen.

Das Hauptergebnis dieser Arbeit ist die Entwicklung von zwei neuen effizienten datenparallelen Progressive-Refinement-Verfahren zur Approximation der globalen Beleuchtung in hochgradig komplexen Modellen und die Integration dieser Verfahren in ein interaktives Bildgenerierungssystem. Die beiden Verfahren basieren auf vollständig unterschiedlichen Methoden zur Bestimmung von Formfaktoren sowie verschiedenen datenparallelen Ansätzen zur Berechnung von konvergierten Radiosity-Lösungen. Diese Unterschiede haben deutliche Auswirkungen auf die Effizienz der parallelen Berechnungen und die Qualität der Radiosity-Lösungen.

Die beiden parallelen Progressive-Refinement-Verfahren wurden dahingehend erweitert, daß Radiosity-Berechnungen in dynamisch veränderbaren Szenen durchgeführt werden können. Interaktive Manipulationen der Geometrie und der Materialeigenschaften von Objekten der Szene werden während der Beleuchtungssimulation berücksichtigt, indem inkrementell eine korrigierte Radiosity-Lösung bestimmt wird. Aufgrund der parallelen Berechnung einer korrigierten Lösung werden sehr kurze Anpassungszeiten erzielt. Die Auswirkungen von Sze-

nenmodifikationen auf die globale Beleuchtung können visuell kontrolliert werden, ohne das lange Wartezeiten entstehen, wie es bei einer kompletten Neuberechnung einer Lösung der Fall ist. Somit stellen die vorgestellten parallelen Verfahren einen Schritt in Richtung des Modellierungsparadigmas *Rendering while Modeling* dar, welches als wesentliche Verbesserung des bisher gültigen Paradigmas *Modeling then Rendering* zu betrachten ist. Dem Verfasser dieser Arbeit sind keine weiteren parallelen Verfahren bekannt, welche Radiosity-Berechnungen in dynamisch veränderbaren Szenen unterstützen.

Mit Hilfe von Verfahren zur Simulation globaler Beleuchtungseffekte soll ein möglichst großer Teil der Pfade berücksichtigt werden, die ein Photon nach dessen Emission durch eine Lichtquelle innerhalb eine Umgebung zurücklegen kann, bevor es von einer Oberfläche absorbiert wird. Auf diesen *Lichttransportpfaden* tragen die Photonen bei jeder Richtungsänderung, die durch Reflexion und Refraktion verursacht wird, zur direkten oder indirekten Beleuchtung der jeweiligen Objektoberflächen bei. Die im Rahmen dieser Arbeit vorgestellten parallelen Progressive-Refinement-Verfahren ermöglichen es, bei der globalen Beleuchtungssimulation sehr komplexe Lichttransportpfade zu simulieren. Durch die Berechnung von erweiterten Formfaktoren und der Durchführung einer Zwei-Phasen-Simulation werden erheblich mehr Beleuchtungseffekte berücksichtigt, als dieses mit der klassischen Radiosity-Methode möglich ist. Bei der Bildgenerierung können Beleuchtungseffekte wiedergeben werden, die durch diffuse Reflexionen zwischen den Objekten der Szene und spiegelnde Reflexion sowie transparente Objekte entstehen. Dieses trägt zu einer deutlichen Steigerung des Realitätsgrades der Bilder bei, wobei die Berechnungszeiten durch die parallelen Berechnungen auf einer großen Anzahl von Prozessoren deutlich verkürzt werden können.

Die entwickelten parallelen Methoden wurden in ein Bildgenerierungssystem integriert, welches die Vorteile von Computersystemen mit spezieller Grafikhardware und Parallelrechnern mit verteiltem Speicher nutzt, um Anwendern eine interaktive Begehung (*Walkthrough*) korrekt beleuchteter virtueller Umgebungen zu ermöglichen. Das System wurde in Form einer modularen Client/Server-Architektur konzipiert. Hierbei werden Client-Komponenten auf Computern mit spezieller Hardware gestartet, die eine beschleunigte Durchführung von Walkthrough-Animationen auf der Basis von berechneten Radiosity-Lösungen unterstützt. Die eigentliche Radiosity-Simulation wird als Server-Komponente auf einem Parallelrechner mit verteiltem Speicher datenparallel ausgeführt. Der Anwender kann zu jedem Zeitpunkt der globalen Beleuchtungssimulation Zwischenergebnisse der Berechnungen zur Visualisierung anfordern oder Veränderungen der Szene an die Server-Komponente übermitteln. Bei der Implementierung der parallelen Radiosity-Methode wurde aus Gründen der Portabilität PVM (*Parallel Virtual Machine*) verwendet. Dieses ermöglicht die Nutzung einer großen Anzahl von Rechnersystemen für die parallele Beleuchtungssimulation. Das System hat einen hochgradig modularen Aufbau, ist einfach um zusätzliche Komponenten zu erweitern und kann in kurzer Zeit an die Bedürfnisse von Endanwendern individuell angepaßt werden.

Ein wesentlicher Nachteil von leistungsfähigen Parallelrechnern ist deren Preis. Dieses führt dazu, daß der Anwenderkreis solcher HPC-Systeme im allgemeinen auf die Großindustrie und Forschungseinrichtungen beschränkt ist. Da es sich bei der Zielgruppe des in dieser Arbeit beschriebenen Systems um kleine Architektur- und Visualisierungsbüros handelt, ist davon auszugehen, daß auf Seiten der Anwender keine leistungsfähigen parallelen Systeme lokal vorhanden sind. Aus diesem Grund wurde ein *Remote-Rendering-System* entwickelt, das die effizienten parallelen Methoden zur globalen Beleuchtungssimulation als Electronic-Commerce Dienstleistung über das Internet zugänglich macht. Im Rahmen des EU-Forschungsprojektes HiPEC (*High Performance Visualization System supporting Networked Electronic Commerce Applications*) wurde dieses Szenario erfolgreich in einer speziellen E-

Commerce Applikation getestet. Dieses Szenario wird in dem BMBF-Projekt HiQOS auf weitere Anwendungsfelder ausgedehnt, die über den Bereich der Architekturvisualisierung hinausgehen. Die grundlegende Architektur des Remote-Rendering-Systems, dessen Komponenten sowie die typischen Anwendungsszenarios sind in dieser Arbeit beschrieben worden, um die praktische Relevanz der in dieser Arbeit behandelten Aspekte zu verdeutlichen.

7.2 Repräsentative Visualisierungen

Neben der Berechnungsgeschwindigkeit steht vor allen Dingen die Qualität der generierten Bilder im Vordergrund. Die Qualität der in Kapitel 5 beschriebenen Verfahren wird im folgenden anhand von verschiedenen repräsentativen Architekturszenen demonstriert, die als Grundlage für eine Simulation der Lichtenergieverteilung durch die vorgestellten parallelen Radiosity-Verfahren (bzw. Mehrphasen-Verfahren) verwendet wurden. Es werden jeweils Informationen über die speziellen Eigenschaften der Szene und die parallelen Berechnungen angegeben.

7.2.1 Konferenzraum

In Abb. 7.1 ist eine Ansicht eines Konferenzraumes wiedergegeben, der durch Licht beleuchtet wird, das durch die Seitenfenster einfällt. Für das einfallende Licht wurde eine niedrige Farbtemperatur angenommen, um eine Beleuchtungsstimmung zu simulieren, wie sie am Abend bei Sonnenuntergang und einem niedrigen Sonnenstand gegeben ist. Dieses Modell wurde auch im Rahmen einiger Testmessungen für die parallelen Radiosity-Berechnungen in Kapitel 5 verwendet. Dort erfolgte die Beleuchtung der Szene jedoch durch Deckenlichtquellen.

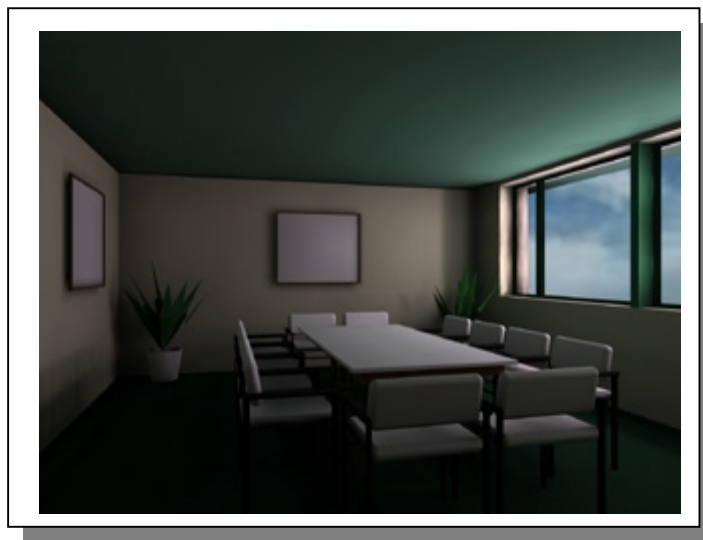


Abb. 7.1: Abendbeleuchtung eines Konferenzraums des Soda-Hall Komplexes.

7.2.2 Wohnzimmer

Bei dem Modell, für welches in Abb. 7.2 eine Zwei-Phasen-Visualisierung der globalen Beleuchtung wiedergegeben ist, handelt es sich um ein primär durch Deckenfluter und eine kleine Tischlampe beleuchtetes Wohnzimmer. Somit wird der Hauptteil der globalen Beleuchtung

durch indirekte Beleuchtung verursacht, indem das von Deckenflutern abgegebene Licht durch die Decke des Wohnzimmers diffus in die Szene reflektiert wird. Das Modell wurde von der *IEZ AG* im Rahmen des BMBF-Projektes HiQos zur Evaluierung der globalen Beleuchtungsverfahren des Remote-Rendering-Systems (siehe Abschnitt 6.2) modelliert.



Abb. 7.2: Beleuchtung eines Wohnzimmers durch Deckenfluter.

7.2.3 Rosenthaler-Hof

Das Bauvorhaben *Rosenthaler-Hof* im Zentrum von Berlin wurde als Präsentationsszene im Rahmen des BMBF-Projektes PARAGRAPH ausgewählt. Es besteht aus dem ehemaligen Kaufhaus *Wertheim*, das saniert und zu einem Gebäudekomplex mit Wohnungen, Büros und Geschäften umgebaut wird. Das Modell wurde von der Firma *SystemConnect* aus Berlin modelliert. In Abb. 7.3 sind zwei Ansichten innerhalb des überdachten Innenhofes der Bürogebäude dargestellt, wobei die Beleuchtungsverhältnisse bei Nacht wiedergegeben sind.

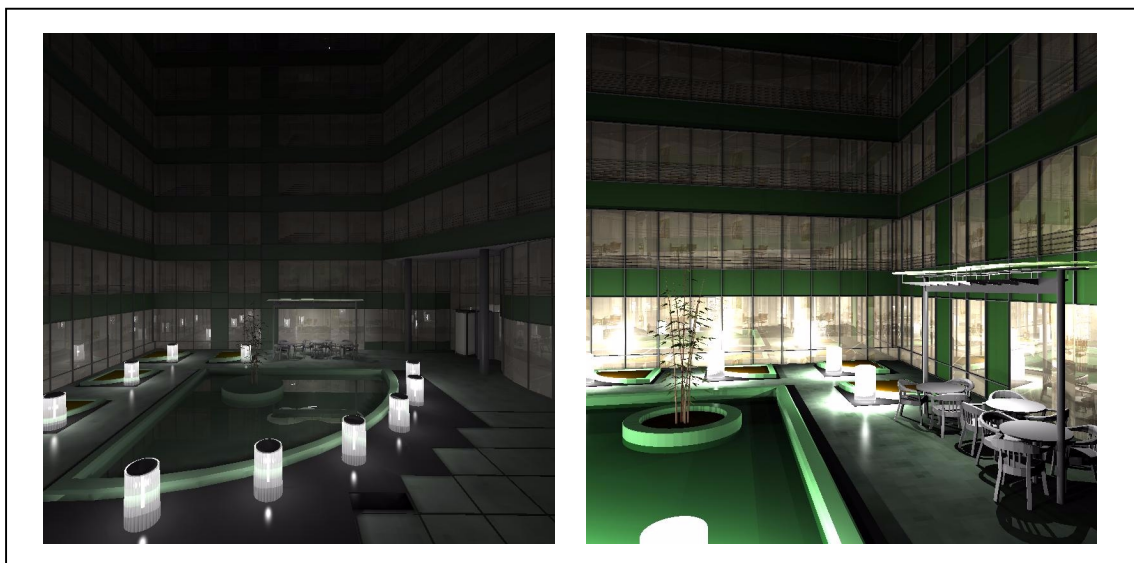


Abb. 7.3: Zwei-Phasen-Visualisierung der Präsentationsszene *Rosenthaler Hof*.

7.2.4 Universitätsgebäude in der Fürstenallee

Durch das *Heinz Nixdorf Institut* (HNI) wurde ein CAD-Modell der Zweigstelle der Universität-GH Paderborn in der Fürstenallee erstellt. Für dieses Modell ist eine Beleuchtungssimulation mit dem entwickelten parallelen Mehrphasen-Verfahren durchgeführt worden. Bei diesem Modell handelt es sich, mit initial ca. 250.000 Patches innerhalb des Radiosity-Netzes, um eines der komplexesten Szenarios, das durch das parallele Radiosity-Verfahren verarbeitet wurde. Aufgrund der adaptiven Verfeinerung während der Radiosity-Berechnungen entsteht ein Radiosity-Lösung mit ca. 1,8 Mio. Patches. In Abb. 7.4 sind die verschiedenen Beleuchtungssituationen des Lichthofes bei Tag und bei Nacht wiedergegeben.

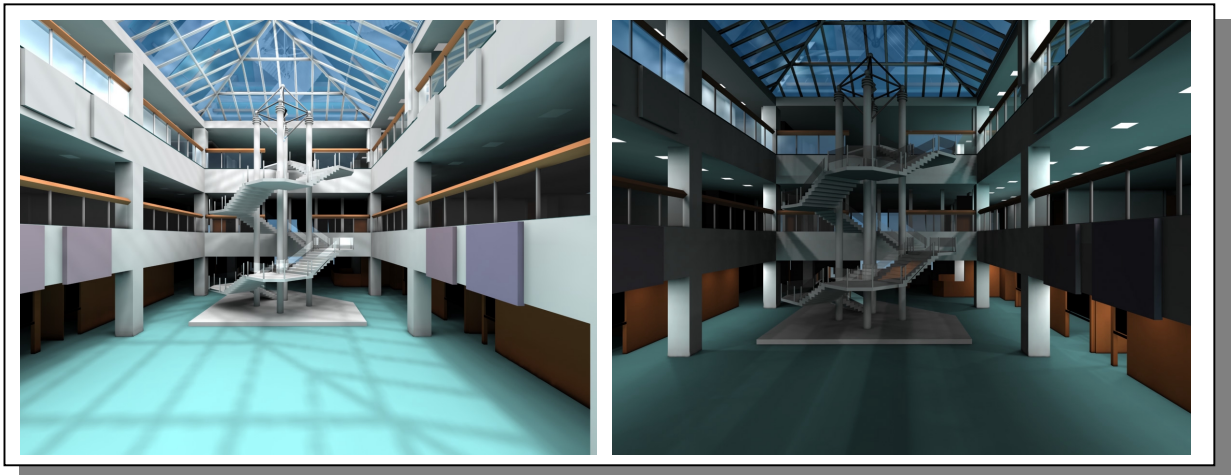


Abb. 7.4: Der Lichthof der *Fürstenallee 11* bei Tag (links) und bei Nacht (rechts).

7.2.5 Sudanesischer Löwentempel

Für das Software-Theater im Heinz Nixdorf Museumsforum wurde eine Tempelanlage im Sudan am Oberlauf des Nils mit umfangreichen Texturinformationen modelliert.



Abb. 7.5: Verschiedene Ansichten innerhalb des *Sudanesischen Löwentempels*.

Dieses Modell wurde in Zusammenarbeit der Sudanarchäologischen Gesellschaft zu Berlin (SAG), dem Seminar für Sudanarchäologie und Ägyptologie der Humboldtuniversität in Berlin und des Fraunhofer-Instituts für Arbeitswesen und Organisation (IAO), Stuttgart, erstellt. Es repräsentiert das Innere des Löwentempels bei Musawwarat es Sufra und wird durch Sonnenlicht, das durch die geöffnete Tür einfällt, beleuchtet. Um einen realistischen Eindruck und realistische Atmosphäre zu schaffen ist eine Simulation der globalen Beleuchtung in dieser Umgebung unbedingt notwendig. Zwei verschiedene Innenansichten der Tempelanlage sind in Abb. 7.5 wiedergegeben. Die globale Beleuchtung wurde in diesem Fall mit dem parallelen Progressive-Refinement-Verfahren berechnet, das in Abschnitt 5.3 vorgestellt wurde.

7.2.6 Kinokomplex

Bei dem Modell *Kinokomplex* handelt es sich um einen vollständigen Komplex, bestehend aus mehreren Vorführräumen und der Eintrittshalle. Der Kinokomplex wurde im Rahmen einer Diplomarbeit im Bereich Architektur an der Fachhochschule Lippe (Abteilung Detmold) modelliert. Mit Hilfe des in Abschnitt 5.2 vorgestellten datenparallelen Radiosity-Verfahrens wurde die Beleuchtungssituation simuliert, die in einem Kino dadurch entsteht, daß der Zuschauerbereich hauptsächlich durch die Kinoleinwand indirekt beleuchtet wird.

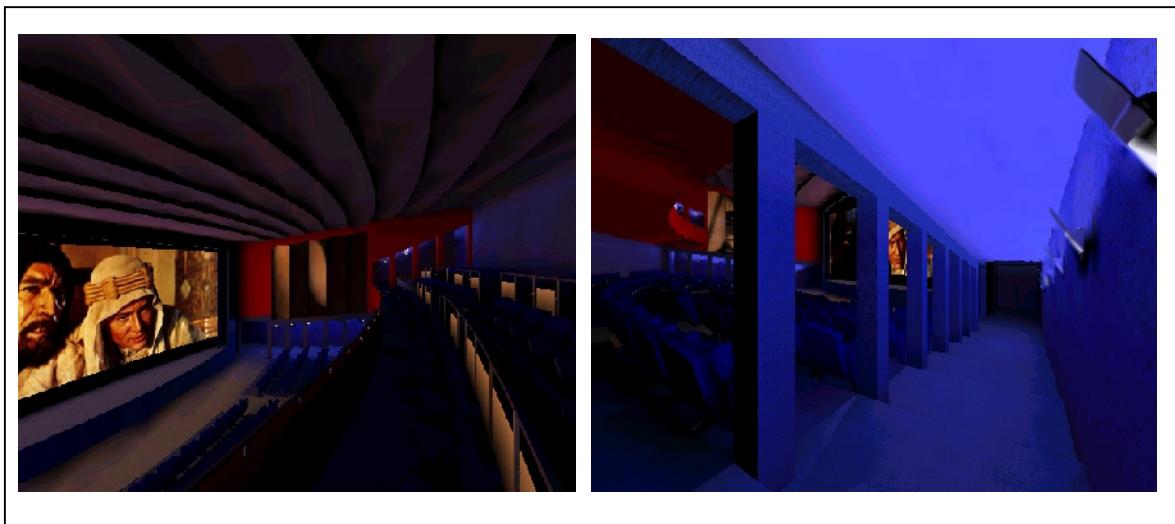


Abb. 7.6: Großer Kinosaal des Kinokomplexes (Radiosity-Visualisierung)

7.2.7 Dom zu Wetzlar

Von der Firma *UPASTART!* wurde im Rahmen des BMBF-Projektes HiQos das Modell des *Doms zu Wetzlar* zur Verfügung gestellt, um die Qualität des in 5.3 vorgestellten parallelen Radiosity-Verfahrens zu evaluieren. Dieses Modell beschreibt den Dom, wie er im frühen Mittelalter ausgesehen hat. Das Modell wurde ursprünglich dazu verwendet, die baugeschichtliche Entwicklung des Doms zu Wetzlar in Form von Computeranimation in einem Dokumentationsvideo zu veranschaulichen. In Abb. 7.7 ist zwei Innenansichten des Doms zu Wetzlar angegeben. Die Szene wird durch Licht beleuchtet, das durch die Fensteröffnungen in das Innere des Kirchengebäudes fällt. Der größte Teil der Beleuchtung der Szene wird durch diffus reflektiertes Licht verursacht.

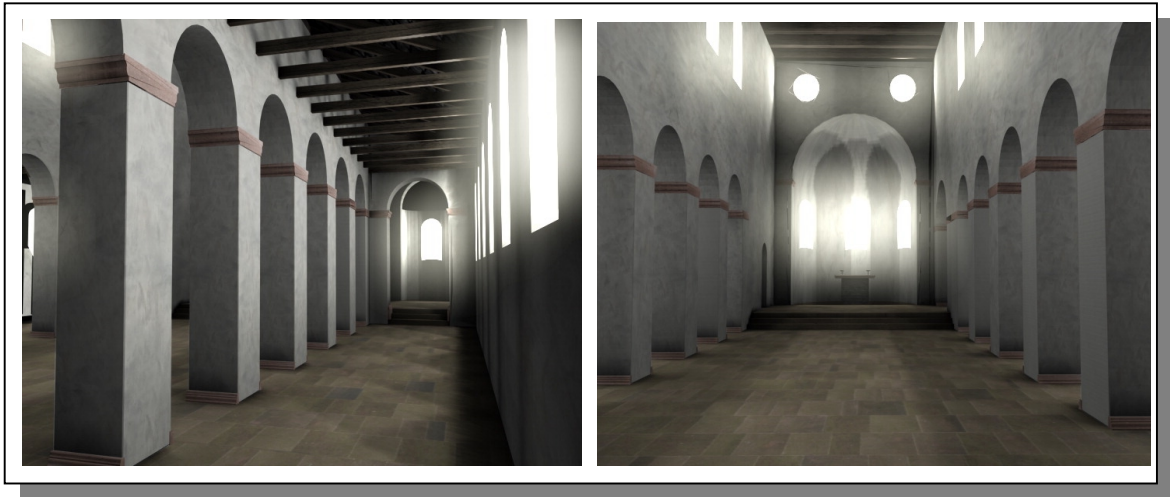


Abb. 7.7: Innenansichten des *Doms zu Wetzlar* in seiner mittelalterlichen Form.

7.2.8 Jagdschloß *Platte* in Wiesbaden

Ebenfalls durch die Firma *Upstart!* wurde ein Modell des Jagdschlusses *Platte* in Wiesbaden erstellt. Bei diesem Modell handelt es sich um das bisher komplexeste Modell, für welches mit Hilfe des parallelen Progressive-Refinement-Verfahrens aus Abschnitt 5.3 eine Radiosity-Lösung bestimmt wurde. Zur Berechnung der Lösung wurden 64 Prozessoren des hpcLine-Systems verwendet, die über fünf Stunden Berechnungszeit benötigten. Der Treppenaufgang, der in Abb. 7.8 dargestellt ist, wird durch eine Öffnung in der Kuppel beleuchtet. Den Objekten der Szene noch keine Farbwerte zugewiesen, da eine Lichtstudie im Vorfeld einer Animationserzeugung durchgeführt werden sollte, um zu identifizieren, wie gut die gesamte Szene ausgeleuchtet wird. Aus diesem Grund sind in Abb. 7.8 lediglich Graustufen-Bilder wiedergegeben, die sehr schön die Licht und Schattenwirkung innerhalb der Szene wiedergeben. Initial bestand das Radiosity-Netz aus ca. 500.000 Patches. Diese Patches wurden im Berechnungsverlauf adaptiv in 3,5 Mio. Patches verfeinert. Die Visualisierungskomponente des interaktiven Bildgenerierungssystems (siehe Abschnitt 6.1) benötigte mehr als 4 GByte Hauptspeicher, um die vollständige Radiosity-Lösung auf der tiefsten Substrukturierungsebene der Patch-Hierarchien anzuzeigen.



Abb. 7.8: Lichtstudie für den Treppenaufgang des Jagdschlusses.

Literaturverzeichnis

- [AK90] Arvo J.R., Kirk D.B., Particle Transport and Image Synthesis, *Computer Graphics (ACM SIGGRAPH 90 Proceedings)*, **24**(4), 63-66, August, 1990.
- [Arv86] Arvo J.R., Backward Ray Tracing, *ACM SIGGRAPH 86 Course Notes – Developments in Ray Tracing*, Vol. 12, August, 1986.
- [Ash94] Ashdown I. Radiosity: A Programmer's Perspective, *John Wiley & Sons Inc*, October, 1994.
- [APV91] Arnaldi B., Pueyo X., Vilaplana J., On the Division of Environments by virtual walls for radiosity computations. *Proc. of the 2nd Eurographics Workshop on Rendering*, 198-205, 1991.
- [APRP96] Arnaldi B., Priol T., Renambot L., Pueyo X. Visibility masks for solving complex radiosity computations on multiprocessors. *Proc. 1st Eurographics Workshop on Parallel Graphics and Visualization*, 219-232, 1996.
- [BFGS86] Bergman L., Fuchs H., Grant E., Spach S., Image Rendering by Adaptive Refinement, *Computer Graphics (ACM SIGGRAPH 86 Proceedings)*, **20**(4), 29-28, August, 1991.
- [BF89] Buckalew C., Fussel D., Illumination Networks: Fast Realistic Rendering with General Reflectance Functions. *Computer Graphics (ACM SIGGRAPH 89 Proceedings)*, Vol. 23, 89-98, 1989.
- [BG95] Bohn C.A., Garmann R., A Parallel Approach to Hierarchical Radiosity, *Proceedings of the Winter School of Computer Graphics and CAD Systems '95*, 26-35, University of West Bohemia, Pilzen, Czech Republic, 1995.
- [Blinn77] Blinn, J.F., Models of light reflection for computer synthesized pictures, *Computer Graphics (ACM SIGGRAPH 77 Proceedings)*, Vol. 11, 31-40, 1977.
- [BL90] Burke A., Leler W., Parallelism and Graphics: An Introduction and Annotated Bibliography, *SIGGRAPH 90 – Course Notes (Parallel Algorithms and Architectures for 3D Image Generation)*, 111-140, August, 1990.
- [BMSW91] Baum D.R., Mann S., Smith K.P., Winget J.M., Making Radiosity Usable: Automatic Preprocessing and Meshing Techniques for the Generation of Accurate Radiosity-Solutions. *Computer Graphics (ACM SIGGRAPH 91 Proceedings)*, **25**(4), 51-60, 1991.
- [Bokh81] Bokhari S.H., One the Mapping Problem, *IEEE Transaction on Computers*, **30**(3), 207 – 214, März, 1981.
- [Boug] Bouguer P., *L'Essais d'Optique*, 1729 Paris.

- [Bres97] Bresink M., Deutschsprachige Terminologie des Radiosity-Verfahrens, *Technischer Bericht der Universität Koblenz 29/97*, 1997.
- [BRW89] Baum D.R., Rushmeier, H.E., Winget J.M. Improving radiosity solutions through the use of analytically determined form factors, *Computer Graphics (ACM SIGGRAPH 89 Proceedings)*, **23**(3), 325-334, 1989.
- [BT75] Bui-Tuong P., Illumination for Computer Generated Pictures, *Communication of the ACM*, **18**(6), 311-317, Juni, 1975.
- [BWCG86] Baum D.R., Wallace J.R., Cohen M.F., Greenberg D.P., The Back-Buffer Algorithm: An Extension of the Radiosity-Method to Dynamic Environments, *The Visual Computer*, **2**(5), 17-28, September, 1986.
- [BW90] Baum D.R., Winget J.M. Real time radiosity through parallel processing and hardware acceleration, *Computer Graphics (1990 Symposium on Interactive 3D Graphics)*, **24**(2), 67-75, 1990.
- [CAO93] Capin T.K., Aykanat C., Oezgüec, Progressive Refinement Radiosity on Ring-Connected Multicomputers, *Proceedings of the 1993 Parallel Rendering Symposium*, 71-76, Oktober, 1993.
- [CCWG88] Cohen M.F., Chen S.E., Wallace J.R., Greenberg D.P. A progressive refinement approach to fast radiosity image generation, *Computer Graphics (ACM SIGGRAPH 88 Proceedings)*, **22**(4), 75-84, 1988.
- [CG85] Cohen M.F., Greenberg D.P. The hemi-cube: A radiosity solution for complex environments, *Computer Graphics (ACM SIGGRAPH 85 Proceedings)*, **19**(3), 31-40, 1985.
- [CGIB86] Cohen M.F., Greenberg D.P., Immel D.S., Brock P.J., An efficient approach for realistic image synthesis, *IEEE Computer Graphics and Applications*, **6**(2), 26-35, 1986.
- [Chen89] Chen S.E., A Progressive Radiosity Method and its Implementation in a Distributed Processing Environment, *M.Sc. thesis*, Program of Computer Graphics, Cornell University, Ithaca, NY, Januar, 1998.
- [Chen90] Chen S.E. Incremental radiosity: An extension of progressive refinement radiosity to an interactive image synthesis system. *Computer Graphics (ACM SIGGRAPH 90 Proceedings)*, Vol. 24, 135-144, 1990.
- [Clark82] Clark J.H., The Geometry Engine: A VLSI Geometry System for Graphics. *Computer Graphics (ACM SIGGRAPH 82 Proceedings)*, **16**(3), 127 - 133, Juli, 1990.
- [CWBV86] Cleary J.G., Wyvill B.M., Birtwistle G.M, Vatti R. Multiprocessor Ray Tracing, *Computer Graphics Forum 5*, North-Holland Publishers, 3-12, 1986.

- [CP91] Chalmers A., Paddon D.J. Parallel processing of progressive refinement radiosity methods. *Proc. of the 2nd Eurographics Workshop on Rendering*, 1991.
- [CR98] Chalmers A., Reinhard E., Parallel and Distributed Photo-Realistic Rendering, *Computer Graphics (ACM SIGGRAPH 98 Course Notes)*, Juli, 1998.
- [CRMT91] Chen S.E., Rushmeier H.E., Miller G., Turner D., A Progressive Multi-Pass Method for Global Illumination, *Computer Graphics (ACM SIGGRAPH 91 Proceedings)*, **25**(4), 165-174, 1991.
- [CT81] Cook, R.L., and Torrance, K.E., A reflectance model for computer graphics, *Computer Graphics (ACM SIGGRAPH 81 Proceedings)*, Vol. 15, . 307-316, 1981.
- [CW93] Cohen M.F., Wallace J.R., Radiosity and Realistic Image Synthesis, *Academic Press*, 1993.
- [DMM95] Diekman R. , Meyer D., Monien B., Parallel Decomposition of Unstructured FEM-Meshes, *Proceedings of IRREGULAR '95*, Springer LNCS 980, 199-215, 1995.
- [DiSo80] Dijkstra E.W., Scholten C.S., Termination Detection for Diffusing Computations, *Transputer Research and Applications (Proceedings NATUG 4)*, 183-193, 1980.
- [DS84] Dippe M., Swensen J., An Adaptive Subdivision Algorithm and Parallel Architecture for Realistic Image Synthesis. *Computer Graphics (ACM SIGGRAPH 84 Proceedings)*, **18**(3), 149-158, Juli, 1984.
- [Enc90] Encarnacao J. , Geometrische Verfahren der graphischen Datenverarbeitung, Berlin, *Springer-Verlag*, 1990
- [ED59] Eckbert E.R.G., Drake R.M, Heat and Mass Transfer, *Mc Graw-Hill*, New York, NY, 1959.
- [Fly72] Flynn M.J., Some Computer Organisations and their Effectiveness, *IEEE Transactions on Computers*, 21(9), 948-960, 1972.
- [FP91] Feda M., Purgathofer W., Progressive Refinement Radiosity on a Transputer Network, *Proceedings of the 2nd Eurographics Workshop on Rendering*, 139-148, Mai, 1991.
- [FPI86] Fujimoto A., Tanaka T., Iwata K., ARTS: Accelerated Ray-Tracing System, *IEEE Computer Graphics and Applications*, **6**(4), 16-26, April, 1986.
- [FvDFH90] Foley J.D., van Dam A., Feiner S.K., Hughes J.F., Computer Graphics, Principles and Practice, *Addison-Wesley*, Massachusetts, 2nd Edition, 1990.
- [Glass84] Glassner A.S., Space Subdivision for fast ray tracing, *IEEE Computer Graphics and Applications*, **4**(10), 15-22, Oktober, 1984.

- [Glass89] Glassner A.S., An Introduction To Ray Tracing, *Academic Press*, 1989.
- [GF85] Glassner A., Fuchs H., Hardware Enhancements For Raster Graphics, *Fundamental Algorithms for Computer Graphics*, Vol. 17, Springer-Verlag, 631-658.
- [Gib98] Gibson S., Efficient Radiosity Simulation using Perceptual Metrics and Parallel Processing, *PhD-Thesis*, University of Manchester, September, 1998.
- [Gou71] Gouraud H., Continuous Shading of Curved Surfaces, *IEEE Transaction on Computers*, **C-20**(6), 623-629, Juni, 1971.
- [GSG90] George D.W., Sillion F., Greenberg D.P., Radiosity Redistribution for Dynamic Environments, *IEEE Computer Graphics and Applications*, **10**(4), 26-34, Juli, 1990.
- [GS87] Goldsmith J., Salomon J., Automatic Creation of Object Hierarchies for Ray Tracing, *IEEE Computer Graphics and Applications*, **7**(5), 14-20, 1987.
- [GTGB84] Goral C.M., Torrance K.E., Greenberg D.P., Bataille B., Modelling the Interaction of Light Between Diffuse Surfaces, *Computer Graphics (ACM SIGGRAPH 84 Proceedings)*, **18**(3), 212-222, Juli, 1984.
- [HSA91] Hanrahan P., Salzman D., Aupperle L., A Rapid Hierarchical Radiosity Algorithm, *Computer Graphics (ACM SIGGRAPH 91 Proceedings)*, **25**(4), 197-206, Juli, 1991.
- [Heck91] Heckbert P.S., Adaptive Radiosity Textures for Bidirectional Ray Tracing, *Computer Graphics (ACM SIGGRAPH 91 Proceedings)*, **24**(4), 145-154, August, 1991.
- [Heiss94] Heiss H.U., Prozessorzuteilung in Parallelrechnern, *BI-Wissenschaftsverlag, Reihe Informatik Band 98*, 1994.
- [Horn75] Hornbeck, R.W., Numerical Methods, *New York: Quantum Publishers*, 1975.
- [Hot54] Hottel H.C., Radiant Heat Transmission, *McGraw-Hill*, New York, NY, 1967.
- [HS67] Hottel H.C., Sarofim A.F., Radiative Transfer. *McGraw-Hill*, New York, NY, 3rd edition, 1954.
- [HF85] Hu M., Foley J.D., Parallel Processing Approaches To Hidden Surface Removal In Image Space, *Computers and Graphics*, **9**(3), 303-317.
- [HS91] Hanrahan P., Salzman D.A., A Rapid Hierarchical Radiosity Algorithm, *Computer Graphics (ACM SIGGRAPH 91 Proceedings)*, **25**(4), 197-206, Juli, 1991.
- [HW91] Haines E.A., Wallace J.R., Shaft Culling for Efficient Ray-Cast Radiosity, 2nd *Eurographics Workshop on Rendering*, Barcelona (Spain), Mai, 1991.

- [IAÖ91] İşler V., Aykanant C., Özgüç B., Subdivision of 3D space based on graph partitioning for parallel ray tracing, *Proceedings of the 2nd Eurographics Workshop on Rendering*, 1991.
- [Kaj86] Kajiya J.T., The Rendering Equation, *Computer Graphics (ACM SIGGRAPH 86 Proceedings)*, **20**(4), 143-150, August, 1986.
- [Kap85] Kaplan M.R., Space Tracing a constant time ray tracer, *Computer Graphics (State of the Art in Image Synthesis SIGGRAPH 85 Course Notes)*, Vol. 11, Juli, 1985.
- [KK86] Kay T.L., Kajiya J.T., Ray Tracing Complex Scenens, *Computer Graphics (ACM SIGGRAPH 86 Proceedings)*, **20**(4), 269-278, August, 1986.
- [KR98] Keller A., Reinefeld A., CCS resource management in networked hpc-systems, *In Proceedings of the Heterogeneous Computing Workshop HPCW 98*, 1998.
- [LKL96] Liang W., King C., Lai F., Adsmith: An efficient object-based distributed shared memory system on PVM, *Proc. Of the 1996 International Symposium on Parallel Architectures, Algorithms and Networks*, 173-179, 1996.
- [LMR92] Lüling R., Monien B., Ramme F., Load Balancing in large networks, *Proceedings of the 3rd IEEE SPDP 92*, 94-101, 1992.
- [Mey90] Meyer G.W., Image synthesis and color vision, *Computer Graphics Techniques Theory and Practice (Hrsg.: Rogers D.F, Earnshaw R.A)*, 45-77, Springer-Verlag, 1990.
- [Mis94] Mistrik R.G. , Rendering the Future, *Lighting Design and Application*, **24**(7), pp. 41-44, Juli, 1994.
- [MRC86] Meyer G.W., Rushmeier H.E. Cohen M.F., Greenberg D.P., An Experimental Evaluation of Computer Graphics Imagery, *ACM Transaction on Graphics*, **5**(1), 30-50, Januar, 1986.
- [MS95] Müller S., Schöffel F., Fast Radiosity Repropagation For Interactive Virtual Environments Using A Shadow-Form-Factor-List, *Photorealistic Rendering Techniques (Hrsg.: Sakas G., Shirly P., Müller S)*, Springer-Verlag, 339-356,1995.
- [NN85] Nishita T., Nakamae E. , Continous Tone Representation of Three Dimansional Objects Taking Account of Shadows and Interreflection, *Computer Graphics (ACM SIGGRAPH 85 Proceedings)*, **19**(3), 23-30, Juli, 1985.
- [NO86] Nemoto K., Omachi T., An Adaptive Subdivision by Sliding Boundary Surfaces for Fast Ray Tracing, *Proceedings of the Conference on Graphics Interfaces*, 43-48, 1986.

- [NOKSO83] Nishimura H., Ohnu H., Kwata T., Shirakawa I., Omura K., LINKS-1: A Parallel Pipelined Multimicrocomputer System For Image Creation, *Proceedings of the 10th Symposium on Computer Architecture*, 387-394, Juni, 1983.
- [Nuss51] Nusselt W., Technische Thermodynamik, Band II, *Theorie der Wärmekraftmaschinen: Sammlung Götschen Bd. 1151*, Berlin, 1951.
- [PB85] Plunket D.J., Bailey M.J. The Vectorization of a Ray Tracing Algorithm for Improved Execution Speed, *IEEE Computer Graphics and Applications*, 52-60, August, 1985.
- [PSV90] Puech C., Sillion F. , Vedel C., Improving Interaction with Radiosity-based Lighting Simulation Programs, *ACM Computer Graphics (Special Issue on 1990 Symposium on Interactive 3D Graphics)*, **24**(2), 51-57, März, 1990.
- [RAPP97] Renambot L., Arnaldi B., Priol T., Pueyo X.. Towards Efficient Parallel Radiosity for DSM-based Parallel Computers Using Virtual Interfaces, *Proceedings of the 3rd Parallel Rendering Symposium (IEEE/ACM/SIGGRAPH)*, Oktober, 1997.
- [RGG90] Recker R.J., George D.W., Greeenber D.W., Acceleration Techniques for Progressive Refinement Radiosity, *Computer Graphics (1990 Symposium on Interactive 3D Graphics)*, **24**(2), 59-66, März, 1990.
- [RW80] Rubin S., Whitted T., A three-dimensional representation for fast rendering of complex scenes, *Computer Graphics (ACM SIGGRAPH 80 Proceedings)*, **14**(3), 110-116, Juli, 1980.
- [Rush86] Rushmeier H.E., Extending the radiosity method to transmitting and specularly reflecting surfaces, *Master's Thesis, Program of Computer Graphics, Cornell University*, 1986.
- [SAWG91] Sillion F.X., Arvo J.R., Westin S.H., Greenberg D.P., A Global Illumination Solution for General Reflectance Distributions, *Computer Graphics (ACM SIGGRAPH 91 Proceedings)*, **25**(4), 187-196, Juli, 1991.
- [Schm94] Schmidt O., Verteilte Energiesimulation in geschlossenen Räumen, *Diplomarbeit an der Universität-GH Paderborn, FB17*, 1994.
- [SH81] Siegel, R., Howell, J.R., Thermal Radiation Heat Transfer, *Whashington, D.C.: Hemisphere Publishing Corp.*, 1981.
- [Š96] Šindlar L., Parallel Radiosity on a Cluster of Workstations, *Proceedings of WSCG '96*, 1996.
- [SL97] Schmidt O., Lange B., Interaktive photorealistische Bildgenerierung durch effiziente parallele Simulation der Lichtenergieverteilung, *Tagungsband der GI-Jahrestagung INFORMATIK 97*, Springer-Verlag, 476-485, 1997.

- [SL98] Schmidt O., Lüling R., HiPEC: High Performance Computing Visualization System supporting Networked Electronic Commerce Applications. *Proc. of Euro-Par'98*, 1149-1152, 1998.
- [Spar63] Sparrow, E.M., A new and simpler formulation for radiative angle factors, *Trans, ASME, Journal of Heat Transfer*, **85**(2), 81-88, 1963.
- [Spar78] Sparrow, E.M., Radiation Heat Transfer, Whashington, D.C.: *Hemisphere Publishing Corp.*, 1978.
- [SP89] Sillion F.X., Puech C., A General Two-Pass Method Integrating Specular and Diffuse Reflection, *Computer Graphics (ACM SIGGRAPH 89 Proceedings)*, **23**(3), 335-344, Juli, 1989.
- [SP94] Sillion F.X., Puech C., Radiosity&Global Illumination, *Morgan Kaufmann Publishers*, 1994.
- [SP00] Schmidt O., Plachetka T., High Performance Computing Rendering System supporting Electronic Commerce Applications in IP-Networks. *Proc. of the Int. Conf. on Parallel and Distributed Processing Techniques and Applications (PDPTA'00)*, Vol. 4, CSREA Press, 1801-1807, 2000.
- [SRRP98] Schmidt O., Rathert J., Reeker L., Plachetka T. Parallel simulation of global illumination. *Proc. of the Int. Conf. on Parallel and Distributed Processing Techniques and Applications (PDPTA'98)*, Vol. 3, CSREA Press, 1289-1296, 1998.
- [SR99] Schmidt O., Reeker L., New Dynamic Load Balancing Strategy for Efficient Data-Parallel Radiosity Calculations. *Proc. of the Int. Conf. on Parallel and Distributed Processing Techniques and Applications (PDPTA'99)*, Vol. 1, CSREA Press, 532-538, 1999.
- [SR00] Schmidt O., Rasch M., Parallel Mesh Simplification. *Proc. of the Int. Conf. on Parallel and Distributed Processing Techniques and Applications (PDPTA'00)*, Vol. 3, CSREA Press, 1361-1367, 2000.
- [SSV95] Stürzlinger W., Schaufler G., Volkert J., Load Balancing for a Parallel Radiosity Algorithm, *Proceedings of the High Performance Computing Symposium 95 (HPCS 95)*, 217-228, Juli, 1995.
- [SSS74] Sutherland I.E., Sproull R.F., Schumacker R.A., A Characterization of Ten Hidden-Surface Algorithmen, *Computing Surveys*, **6**(1), März, 1974.
- [vLiere91] vanLiere R., Divide and Conquer Radiosity, *Proceedings of the 2nd Eurographics Workshop on Rendering*, 191-197, 1991.
- [Wan92] Wanger L., The Effect of Shadow Quality on the Perception of Spatial Relationships in Computer Generated Imagery, *Computer Graphics (ACM Special Issue Proceedings of the 1992 Symposium on Interactive 3D Graphics)*, 39-42, März, 1992.

- [WCG87] Wallace J.R., Cohen M.F., Greenberg D.P., A two-pass solution to the rendering equation: a synthesis of ray tracing and radiosity methods, *Computer Graphics (ACM SIGGRAPH 87 Proceedings)*, **21**(4), .311-320, Juli, 1987.
- [Whi80] Whitted T. , An Improved Illumination Model for Shaded Display , *Communications of the ACM*, **26**(6), 342-349, 1980.
- [WKEH89] Wallace J.R., Kells A.E., Haines E.A., A Ray Tracing Algorithm For Progressive Refinement Radiosity, *Computer Graphics (ACM SIGGRAPH 89 Proceedings)*, **23**(3), 315-324, Juli, 1989.
- [WRC88] Ward G.J., Rubinstein F.M., Clear R.D., A Ray Tracing Solution for Diffuse Interreflection, *Computer Graphics (ACM SIGGRAPH 88 Proceedings)*, **22**(4), 85-92, August, 1988.
- [Zedl93] Radiosity und bewegte Objekte, Seminar Radiosity und Raytracing, *Technische Hochschule Darmstadt, Fachbereich Informatik, Fachgebiet GRIS, WS 92/93.*