

Das Radiosity-Verfahren

Nachteil der lokalen Beleuchtungsverfahren und des Raytracing:

pauschaler Term für ambiente Beleuchtung

- ambientes Licht ist aber insbes. in Innenräumen wichtig:
durchschnittlich ca. 30 % des Lichts in einer Szene stammen nicht unmittelbar von einer Lichtquelle, sondern wurden einmal oder mehrfach an Oberflächen reflektiert
(z.T. sogar Spitzenwerte bis zu 80 %)

⇒ ein einziger konstanter Term für diesen Beleuchtungsanteil ist unzureichend für realitätsnahe Modelle

notwendig: physikalisches Modell für die Entstehung von ambientem (diffusem) Licht

- physikalisches Grundprinzip: Energieerhaltung
- Analogie: Wärmeübertragung

"Radiosity-Verfahren":

entwickelt 1983 von Goral, Greenberg et al.

- globales Beleuchtungsverfahren *für ideal diffus reflektierende Oberflächen*
- realitätsnah, denn diffuse Reflektion ist in der Praxis häufig, spiegelnde Flächen sind eher die Ausnahme
- Grundprinzip: *Gleichgewicht* zwischen zugeführter Strahlungsenergie von Lichtquellen und absorbierte Strahlungsenergie durch alle Oberflächen

in dem Verfahren wird die spezifische Beleuchtungsstärke jeder Fläche berechnet

⇒ die Ergebnisse können im Experiment nachgeprüft und bestätigt werden

- ◆ Vor allem für die Lichtverteilung **in** Gebäuden ist das Radiosity-Verfahren wie kein anderes Beleuchtungsverfahren geeignet.
- ◆ Ein großer Vorteil der **diffusen Reflexion** ist, daß die Leuchtdichte (Intensität) einer Fläche unabhängig von der Beobachtungsrichtung ist.
- ◆ Wurde eine Szene mit dem Radiosity-Verfahren berechnet, so ist es möglich, die Szene aus allen Richtungen zu betrachten, ja sogar durch die Szene hindurchzugehen, ohne sie neu berechnen zu müssen. → wird auch die hohe Rechenzeit für das Verfahren akzeptabel.

Herleitung der Radiosity-Gleichung:

Es wird eine *Bilanzierung des Lichtaustausches* zwischen den Lichtquellen und Objekten durchgeführt.

Dabei werden *Lichtquellen und beleuchtete Objekte* formal *gleich* behandelt.

Insbes.: Lichtquellen sind auch 3D-Objekte ⇒ keine Abstraktion zu Punktlichtquellen (i. Ggs. zum klassischen Raytracing)

Die Rechnung kann für die *Leuchtdichte* (photometrisch) oder für die entsprechende radiometrische Größe, die *Strahldichte* L , durchgeführt werden (vgl. Anfang von Kap. 11: physikalische Grundlagen). L : Energie pro Raumwinkel und pro Fläche.

Ausgangspunkt: **Reflektionsgleichung**.

Trifft in einem Punkt \mathbf{x} Licht aus mehreren Richtungen ein, so beeinflussen sich die einzelnen Reflektionen nicht, sondern können linear überlagert werden. Durch Integration über alle Einfallrichtungen erhält man für die **reflektierte Strahldichte**

$$L_r(\mathbf{x}, \vec{\omega}_r) = \int_{\Omega_i} \rho(\mathbf{x}, \vec{\omega}_r, \vec{\omega}_i) L_i \cos(\mathbf{x}, \vec{\omega}_i) \cos \theta_i d\vec{\omega}_i$$

Index i : einfallende Strahlung;

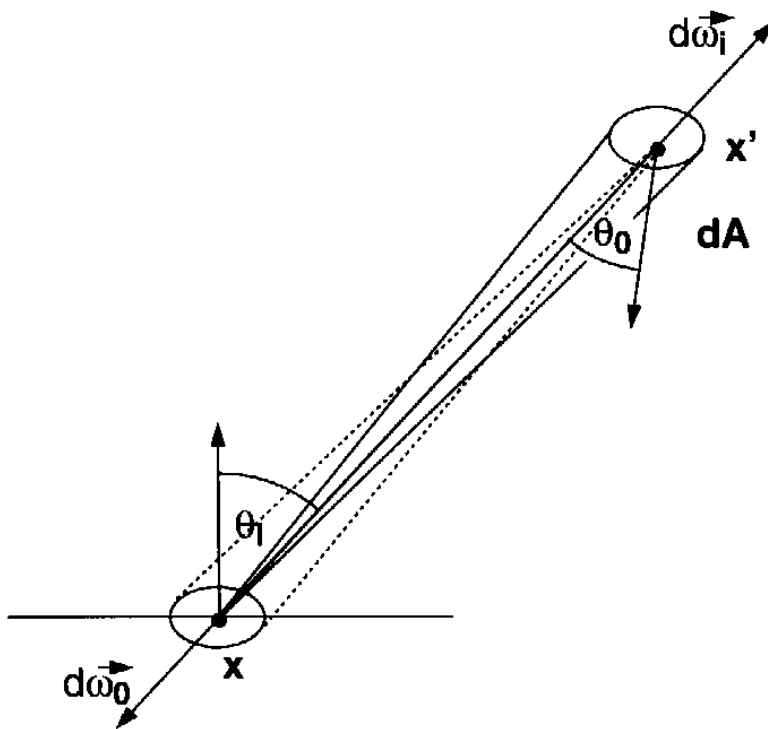
r : reflektierte Strahlung

$\rho(\mathbf{x}, \vec{\omega}_r, \vec{\omega}_i)$ spektraler Reflexionsfaktor (BRDF)

darin ist θ_i der Winkel zwischen Lichteinfallsrichtung und Flächennormale im Punkt \mathbf{x} , ω_r und ω_i sind die Reflektions- bzw. die Einfallsrichtung. Integriert wird über alle möglichen räumlichen Einfallsrichtungen, von denen \mathbf{x} Licht erhalten kann (im allg. Fall eine Vollkugel um \mathbf{x}).

Anstatt über alle Einfallsrichtungen zu integrieren, kann man nun auch *über alle Flächen der Szene*, die Licht aussenden oder reflektieren, integrieren.

Dazu: Grundgleichung des Strahlungsaustausches zwischen 2 Flächen einsetzen (vgl. Anfang von Kap. 10):
(\mathbf{x}' ist ein Punkt auf der zweiten Fläche)



Es ergibt sich die Beleuchtungsgleichung (*rendering equation*) nach Kajiya:

$$L_r(\mathbf{x}, \vec{\omega}_r) = L_e(\mathbf{x}, \vec{\omega}_r) + \int_A \rho(\mathbf{x}, \vec{\omega}_r, \vec{\omega}_i) \cdot L(\mathbf{x}', \vec{\omega}_0) H(\mathbf{x}, \mathbf{x}') \frac{\cos \theta_i \cos \theta_0}{|\mathbf{x} - \mathbf{x}'|^2} dA.$$

- Hier wurde als erster Summand noch ein Emissionsterm L_e eingefügt für den Fall, dass \mathbf{x} selbstleuchtend ist (formale Gleichbehandlung von Lichtquellen und Szenenobjekten!).
- $H(\mathbf{x}, \mathbf{x}')$ ist der *Sichtbarkeitsterm* (= 1, falls \mathbf{x}' von \mathbf{x} sichtbar, sonst 0).

Vereinfachende Annahme: Alle Flächen sind Lambertsche Reflektoren (ideal-diffuse Reflektion, alle Abstrahlrichtungen mit gleicher reflektierter Leuchtdichte)

⇒ die BRDF ρ ist unabhängig von ω_r und ω_i und kann durch den Reflektionsgrad r (genauer: durch r/π) ersetzt und vor das Integral gezogen werden.

Für die Radiosity $B(\mathbf{x})$ im Punkt \mathbf{x} ergibt sich:

$$B(\mathbf{x}) = E(\mathbf{x}) + r \int_A B(\mathbf{x}') H(\mathbf{x}, \mathbf{x}') \frac{\cos \theta_i \cos \theta_o}{|\mathbf{x} - \mathbf{x}'|^2} dA$$

$$B(\mathbf{x}) = E(\mathbf{x}) + r \int_A B(\mathbf{x}') G(\mathbf{x}, \mathbf{x}') dA$$

$$\text{mit } G(\mathbf{x}, \mathbf{x}') = H(\mathbf{x}, \mathbf{x}') \frac{\cos \theta_i \cos \theta_o}{|\mathbf{x} - \mathbf{x}'|^2} \quad \text{Geometrieterm}$$

worin $E(\mathbf{x})$ die Emission des Punktes \mathbf{x} ist (vgl. Encarnaço et al.; Krmker).

Dies ist eine *Integralgleichung* (genauer: eine Fredholmsche Integralgleichung der zweiten Art) – die (zu errechnende) Radiosity B kommt auf der linken Seite und hinter dem Integralzeichen vor!

Die Gleichung (genauer: das System aller entspr. Gleichungen, wenn \mathbf{x} alle Punkte der Szene durchluft) beschreibt folg.

Sachverhalt:

- die Szene ist ein energetisch geschlossenes System
- die von den Lichtquellen ausgesandte Energie entspricht der Summe (dem Integral) der von den Objekten absorbierten Energie
- lokal (in jedem Punkt \mathbf{x}) ist die einfallende Lichtenergie gleich gro wie die Gesamtheit der absorbierten und reflektierten Lichtenergie
- die Radiosity $B(\mathbf{x})$ gibt die von \mathbf{x} abgegebene Lichtenergie (bzw. ihr photometr. Gegenstck) pro Flcheneinheit an

In die Radiosity-Gleichung gehen ein:

- Materialeigenschaften (Reflektionsgrad r , Emissivität E)
- Lagebeziehungen (Sichtbarkeitsterm H , Abstände $|\mathbf{x}-\mathbf{x}'|$, Winkel), zusammengefasst im Geometrieterm $G(\mathbf{x}, \mathbf{x}')$.

Allgemeines Lösungsverfahren für Integralgleichungen dieses Typs:

Finite-Elemente-Verfahren

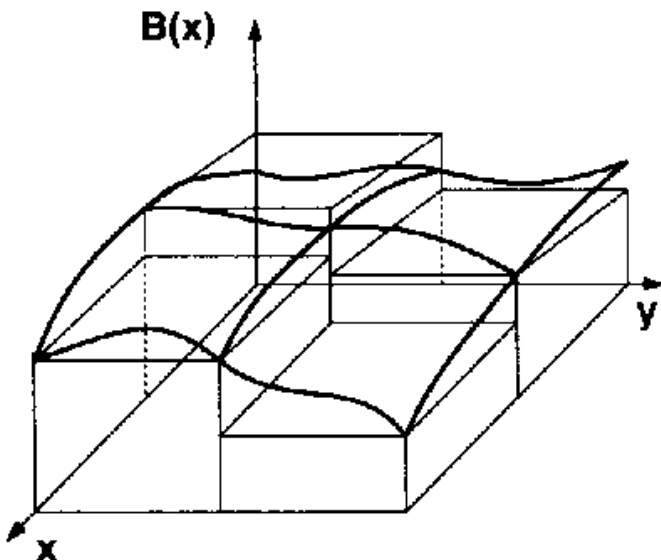
- der Definitionsbereich der Funktion B (die Flächen der Szene) wird in einfache Teilgebiete (finite Elemente) – meist Dreiecke oder Rechtecke – zerlegt,
- als Ansatz für B wählt man eine Linearkombination von Basisfunktionen N , die nur auf endlich vielen Elementen von 0 verschieden sind (vgl. Basisfunktionen der Splines!):

$$B(\mathbf{x}) \approx \hat{B} = \sum_{i=1}^n B_i N_i(\mathbf{x}),$$

die einfachsten Basisfunktionen sind auf einem Element (Dreieck oder Rechteck) konstant 1 und außerhalb 0

⇒

Approximation einer Radiosityfunktion durch konstante Elemente:



aus dieser Approximation ergibt sich die "klassische Radiosity-Gleichung", die die obige Integralgleichung zu einer Summe vereinfacht.

Vereinfachende Annahmen:

- Emission für jede Fläche homogen
- Reflektionsgrad für jede Fläche konstant
- die Flächen (finiten Elemente) werden durchnummeriert mit Index i , ihre Flächeninhalte A_i sind bekannt

Durch Umformung der obigen Integralgleichung (s. Encarnação et al. II, S. 160-163) ergibt sich:

$$E_i = \sum_{j=1}^n B_j \left[\delta_{ij} - r_i \frac{1}{A_i} \int_{A_i} \int_{A_j} G(\mathbf{x}, \mathbf{x}') dA_i dA_j \right] \quad \forall i = 1, \dots, n.$$

Mit den Abkürzungen

$$F_{ij} = \frac{1}{A_i} \int_{A_i} \int_{A_j} G(\mathbf{x}, \mathbf{x}') dA_i dA_j, \quad i, j = 1, \dots, n$$

erhalten wir

$$E_i = \sum_{j=1}^n B_j [\delta_{ij} - r_i F_{ij}] \quad \forall i = 1, \dots, n.$$

F_{ij} : Formfaktor

= Anteil der von der Fläche i abgegebenen Strahlungsleistung, der auf der Fläche j ankommt
abhängig von den Lagebeziehungen in der Szene (Geometrie), *nicht* vom Beobachterstandpunkt oder der Beleuchtung!

- = der zentrale Begriff im Radiosity-Verfahren
- waren bereits vor Einführung dieses Verfahrens in der Computergrafik bekannt aus der Wärmetechnik (vgl. Siegel 1981).
- Reziprozitätseigenschaft: $A_i F_{ij} = A_j F_{ji}$
- die Summe aller F_{ij} für festes i ist ≤ 1 .
- Anzahl der Formfaktoren wächst quadratisch mit Anzahl der Flächen.

- Formfaktorbestimmung: aufwändigster Arbeitsschritt im Radiosity-Verfahren. Effizienzsteigernde Ansätze dazu siehe unten.

Die obige Gleichung für E_i wird umgeformt
 \Rightarrow in Analogie zur Integralformel jetzt als *Summe*:

Radiosity-Gleichungssystem ($i = 1, 2, \dots, n$ durchläuft die Flächen der Szene)

$$B_i = E_i + r_i \sum_{j=1}^n B_j F_{ij}, \quad \forall i = 1, \dots, n. \quad (3.33)$$

Die Gleichung wird noch übersichtlicher, wenn man in die Matrixschreibweise übergeht:

$$\vec{B} = \vec{E} + \mathbf{rF} \vec{B} \Rightarrow (\mathbf{1} - \mathbf{rF}) \vec{B} = \vec{E},$$

wobei nun \vec{E} und \vec{B} Vektoren der Länge n und \mathbf{F} eine $n \times n$ -Matrix ist. Auch \mathbf{r} ist eine $n \times n$ -Matrix, allerdings in Diagonalgestalt. Ausgeschrieben lautet das Gleichungssystem:

$$\begin{pmatrix} 1 & -r_1 F_{12} & \cdots & -r_1 F_{1n} \\ -r_2 F_{21} & 1 & \cdots & -r_2 F_{2n} \\ \vdots & \vdots & & \vdots \\ -r_n F_{n1} & -r_n F_{n2} & \cdots & 1 \end{pmatrix} \begin{pmatrix} B_1 \\ B_2 \\ \vdots \\ B_n \end{pmatrix} = \begin{pmatrix} E_1 \\ E_2 \\ \vdots \\ E_n \end{pmatrix}, \quad (3.34)$$

dies ist ein *lineares Gleichungssystem* mit den Unbekannten B_i (Radiosities = spezifische Ausstrahlungen der einzelnen Flächen der Szene).

(Im letzten Umformungsschritt wurde $F_{ii} = 0$ angenommen, d.h. keine Fläche strahlt Licht auf sich selbst ab: plausibel jedenfalls für konvexe oder ebene Flächen.)

Lösbarkeit (und effektive Lösbarkeit) des linearen Gleichungssystems:

- abhängig von den Eigenschaften der Koeffizientenmatrix $K = \mathbf{I} - \mathbf{rF}$
- diese ist hier *diagonaldominant*, d.h.

$$\sum_{j=1, j \neq i}^n |k_{ij}| < |k_{ii}|.$$

- \Rightarrow sie ist auch regulär \Rightarrow es ex. eindeutige Lösung.

Lösungsverfahren:

- Standardverfahren, d.h. Gaußsches Eliminationsverfahren: Rechenaufwand $O(n^3) \Rightarrow$ Aufwand zu groß für real vorkommende Szenen (in der Regel mehrere tausend Flächenelemente)
- \Rightarrow Verwendung iterativer Verfahren, die Eigenschaften der Matrix K ausnutzen:

- Jacobi-Iteration
- Gauß-Seidel-Verfahren
- Southwell-Verfahren
(siehe Encarnação et al. II, S. 165-168;
Deussen 2001: Skript zur Computergrafik III)

- *Progressive Refinement.*

Initialisierung der Radiosities B_i mit den Emissionstermen E_i

"unverteilte Radiosity" ΔB_i wird ebenfalls mit E_i initialisiert

Iterationsschritt:

unverteilte Radiosity *einer* Fläche j wird auf alle anderen verteilt:
Aktualisierung der Radiosities und der unverteilter Radiosities gemäß

$$B_i = B_i + \Delta B_j r_i F_{ij}$$
$$\Delta B_i = \Delta B_i + \Delta B_j r_i F_{ij}$$

anschließend wird die "unverteilte Radiosity" ΔB_j der Fläche j auf 0 gesetzt.

(Beachte: dazu braucht man jeweils nur die Formfaktoren für ein festes j !)

Wahl der Fläche j (\Rightarrow Geschwindigkeit der Konvergenz des Verfahrens):

man nimmt jeweils *die Fläche mit der größten unverteilter Strahlungsleistung*

(\Rightarrow am Anfang wird das Licht der stärksten Lichtquelle verteilt)

die "unverteilter Radiosities" konvergieren im Laufe des Verfahrens gegen 0 und nähern sich immer mehr aneinander an

Abbruchkriterium für die Iteration: Summe der unverteilter Radiosities unterschreitet Schwellenwert

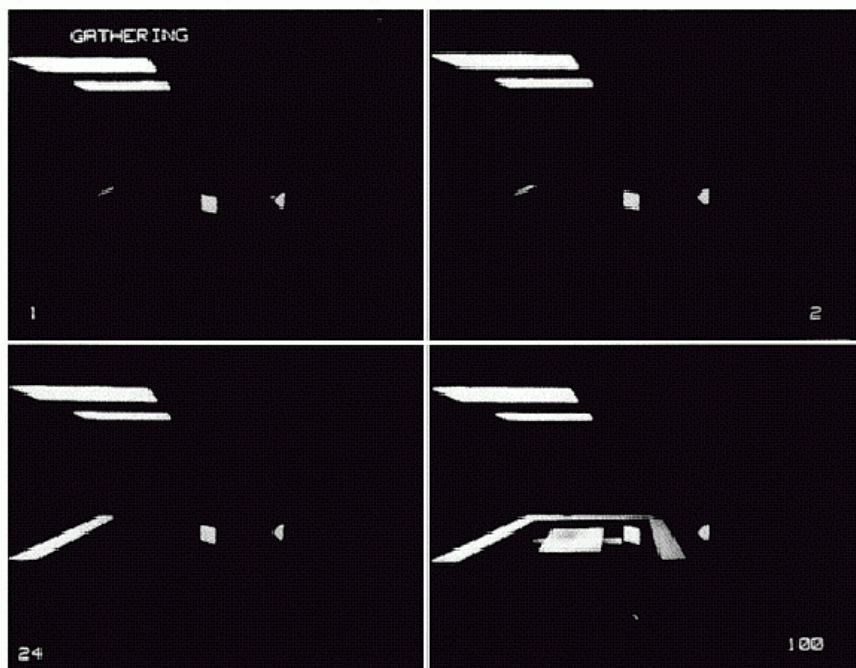
Vorteil des Verfahrens: in jedem Iterationsschritt reicht die Kenntnis einer Zeile der Formfaktormatrix

⇒ da bereits nach wenigen Iterationsschritten gute Ergebnisse erzielt werden, liegt der Gesamt-Rechenaufwand weit unter dem für "simultane" Lösungsansätze für das Gleichungssystem wie z.B. Gauß-Seidel-Verfahren

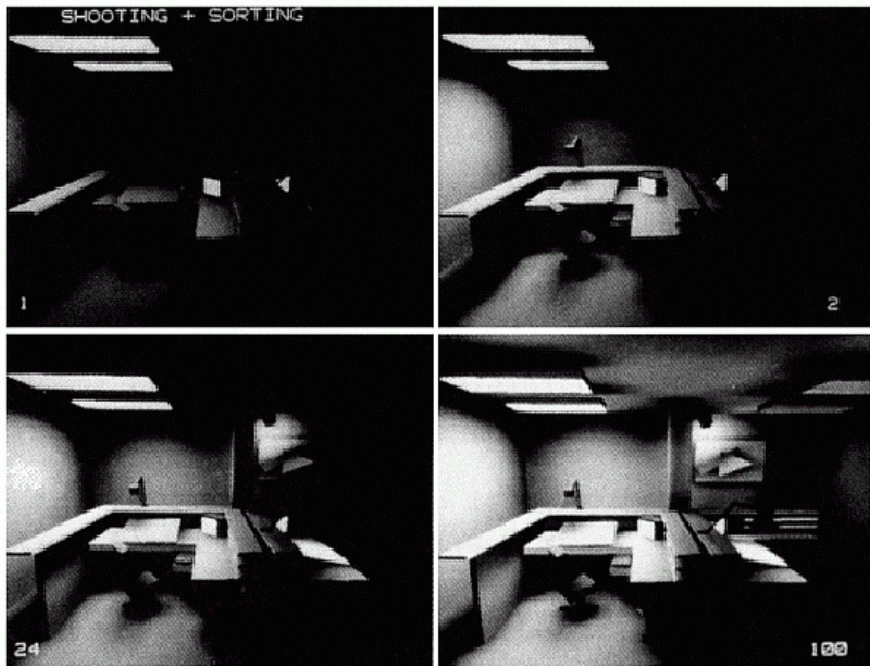
außerdem: es wird nur der Speicherplatz für eine Zeile von Formfaktoren benötigt

(Nachteil: wird eine Fläche zum wiederholten Mal verteilende Fläche, so werden deren Formfaktoren jedesmal wieder neu berechnet)

Beispiel:



Gauss-Seidel after 1, 2, 24, and 100 Steps



Progressive Refinement after 1, 2, 24, and 100 Steps

noch offen:

Berechnung der Formfaktoren

Darstellung als doppeltes Flächenintegral (aus der Definition):

$$F_{A_i-A_j} = F_{i,j} = \frac{1}{A_i} \int_{A_i} \int_{A_j} H_{ij} \frac{\cos \vartheta_j \cos \vartheta_i}{\pi R^2} dA_j dA_i.$$

analytisch lösbar nur in einfachen Fällen

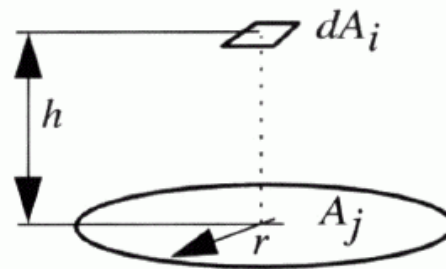
(hier übernommen aus Deussen 2001):

Analytische Methoden

→ für einfache Elemente (Element-Element) und für differentielle Flächen

→ vollständige Sichtbarkeit wird vorausgesetzt

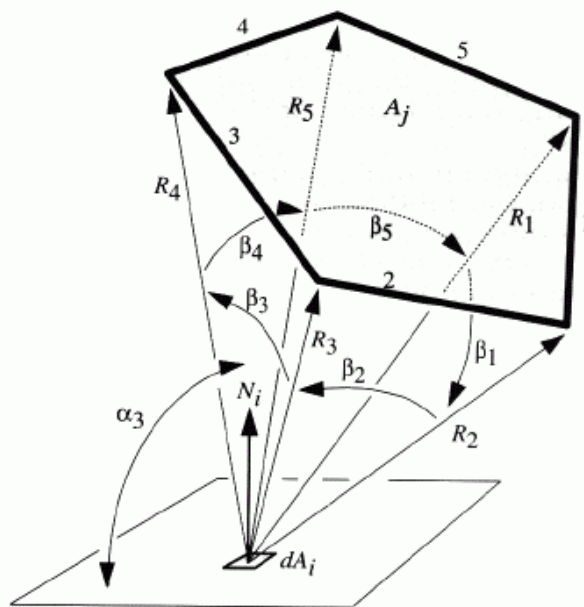
1. Differentielle Fläche auf Kreis



$$F_{ij} = \frac{r^2}{h^2 + r^2}$$

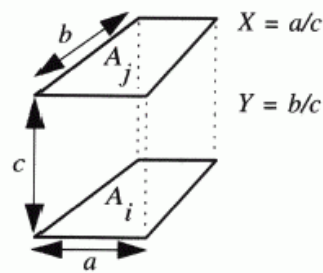
aus: Cohen/Wallace: Radiosity and Realistic Image Synthesis

2. Differentielle Fläche auf Polygon (kommt häufig vor)



$$F_{dA_i \rightarrow A_j} = \frac{1}{2\pi} \sum_{i=1}^n \beta_i \cos \alpha_i$$

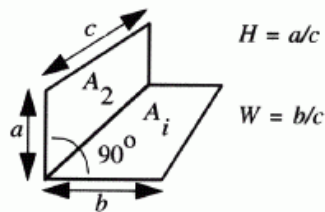
3. Formfaktor zwischen parallelen Rechtecken



aus: Cohen/Wallace: Radiosity and Realistic Image Synthesis

$$F_{ij} = \frac{2}{\pi XY} \left\{ \ln \left[\frac{(1 + X^2)(1 + Y^2)}{1 + X^2 + Y^2} \right]^{1/2} + Y \sqrt{1 + X^2} \tan^{-1} \left(\frac{Y}{\sqrt{1 + X^2}} \right) - X \tan^{-1} X - Y \tan^{-1} Y \right\}$$

4. Formfaktor zwischen senkrechten Rechtecken



aus: Cohen/Wallace: Radiosity and Realistic Image Synthesis

$$F_{ij} = \frac{1}{\pi W} \left\{ W \tan^{-1} \frac{1}{W} + H \tan^{-1} \frac{1}{H} - \sqrt{H^2 + W^2} \tan^{-1} \frac{1}{\sqrt{H^2 + W^2}} + \frac{1}{4} \ln \left[\frac{(1+W^2)(1+H^2)}{1+W^2+H^2} \left(\frac{W^2(1+W^2+H^2)}{(1+W^2)(W^2+H^2)} \right)^{W^2} \left(\frac{H^2(1+H^2+H^2)}{(1+H^2)(W^2+H^2)} \right)^{H^2} \right] \right\}$$

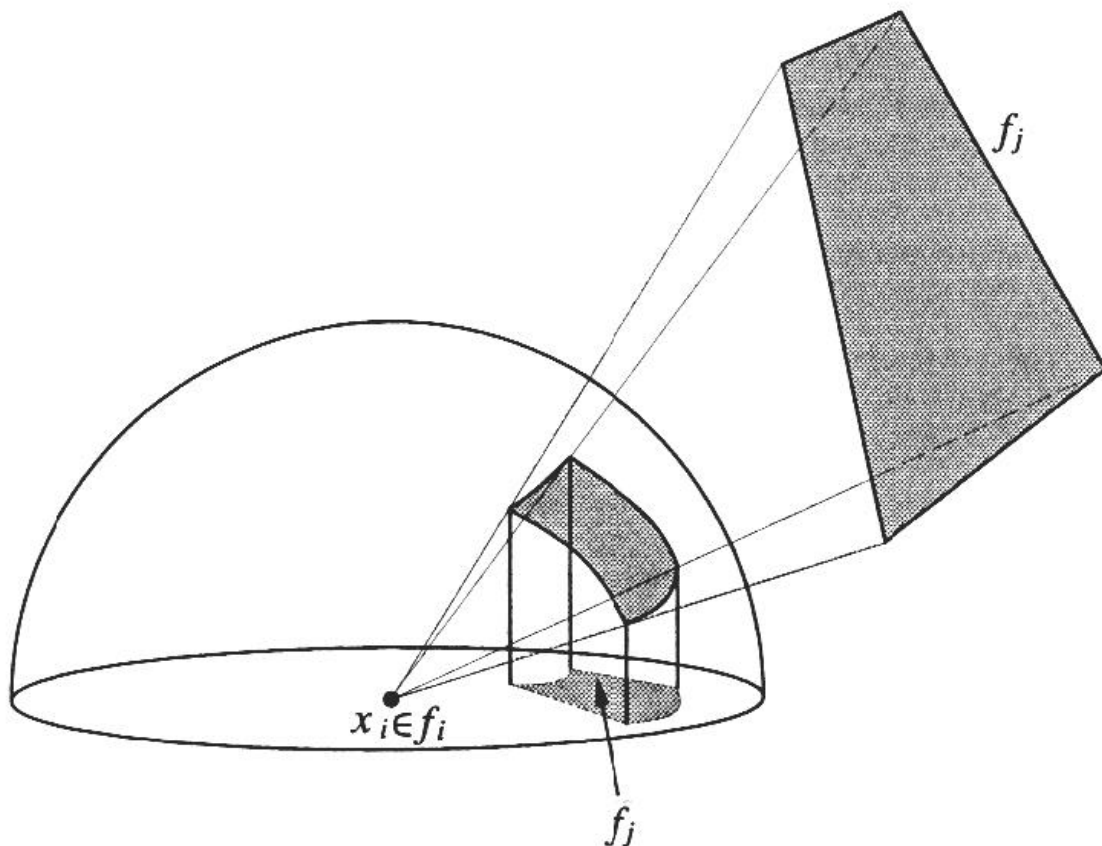
Nusselts Analogon:

Schon 1928 löste Wilhelm Nusselt das Integral der Formfaktorgleichung auf geometrische Weise.

Um das betrachtete Flächenelement wird eine Einheits-Halbkugel gelegt.

Ein Teil des Integranden ($\cos \vartheta_j dA_j / R^2$) entspricht dem infinitesimalen Raumwinkel, und dieser der Projektion des Flächenstücks dA_j auf die Einheitshalbkugel.

Der noch verbleibende Term im Integranden entspricht (bis auf den Faktor π) der Parallelprojektion dieses infinitesimalen Halbkugelausschnitts auf die Grundfläche:



Wenn man berücksichtigt, dass π die Fläche des Einheitskreises ist, so wird deutlich, dass der Formfaktor zwischen dA_j und A_j (Zeichnung: x_i, f_j) der Anteil ist, der vom zweimalig projizierten Flächenstück in der Basisfläche der Halbkugel eingenommen wird.

Für die allgemeine Berechnung von Formfaktoren wird nun die Oberfläche der Halbkugel in kleine Segmente aufgeteilt. Für jedes Segment wird der Formfaktor nach obigem Projektionsverfahren berechnet (→ "Delta-Formfaktoren").

Nach Abspeichern dieser Faktoren wird jede Fläche der Szene auf die Halbkugel projiziert, wobei festgestellt wird, welche Segmente auf der Halbkugel von der Projektion der Fläche bedeckt werden.

Der gesamte Formfaktor der Fläche ist die Summe der Delta-Formfaktoren der von der Projektion bedeckten Segmente.

→ "geniales" Verfahren, aber leider nicht praxistauglich:

- Probleme bei Einteilung der Halbkugel in gleichgroße Segmente
- Probleme bei der Scan-Konvertierung der Szene auf die Halbkugel

aber: Verfahren wurde weiterentwickelt zum ersten brauchbaren Formfaktor-Berechnungsverfahren für die Praxis, dem *Hemi-Cube-Verfahren* (Cohen & Greenberg 1985)

Idee des Hemi-Cube:

anstelle der Halbkugel Rückgriff auf Flächen, für die in der grafischen Datenverarbeitung große Erfahrung existiert: rechteckige Rasterflächen

⇒ Halbkugel wird ersetzt durch *Halbwürfel* (Hemi-Cube)!

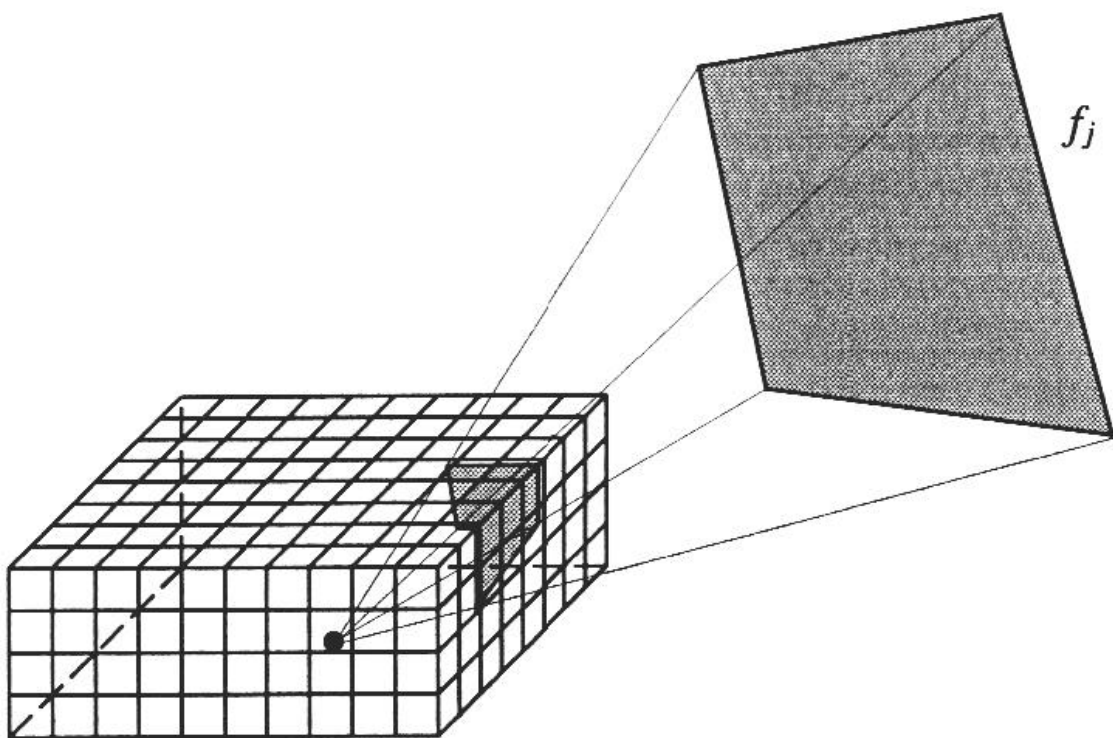
Jede der 5 Seiten wird mit einem rechteckigen Raster überzogen (*Hemi-Cube-Pixel*)

⇒ Projektion der Szene auf eine Seite des Hemi-Cube stellt prinzipiell das gleiche Problem dar wie die Rasterisierung einer 3D-Szene für ein übliches Display

⇒ übliche Scankonvertierungsverfahren anwendbar (vgl. Kap. 5)

- Jedem Hemi-Cube-Pixel wird ein Deltaformfaktor zugeordnet, der den Beitrag zum gesamten Formfaktor beschreibt
- den Formfaktor einer Fläche erhält man, indem man die Deltaformfaktoren derjenigen Hemi-Cube-Pixel, die durch die Projektion der Fläche bedeckt werden, aufsummiert
- je feiner die Unterteilung des Hemi-Cube, desto genauer die Formfaktorberechnung
- durch Verwendung des z-Buffer-Algorithmus ist auch die Einbeziehung der Verdeckung von Flächen (Sichtbarkeitsterm H) möglich

⇒ erstmals wird das Radiosity-Verfahren damit für komplexe Szenen praktisch anwendbar



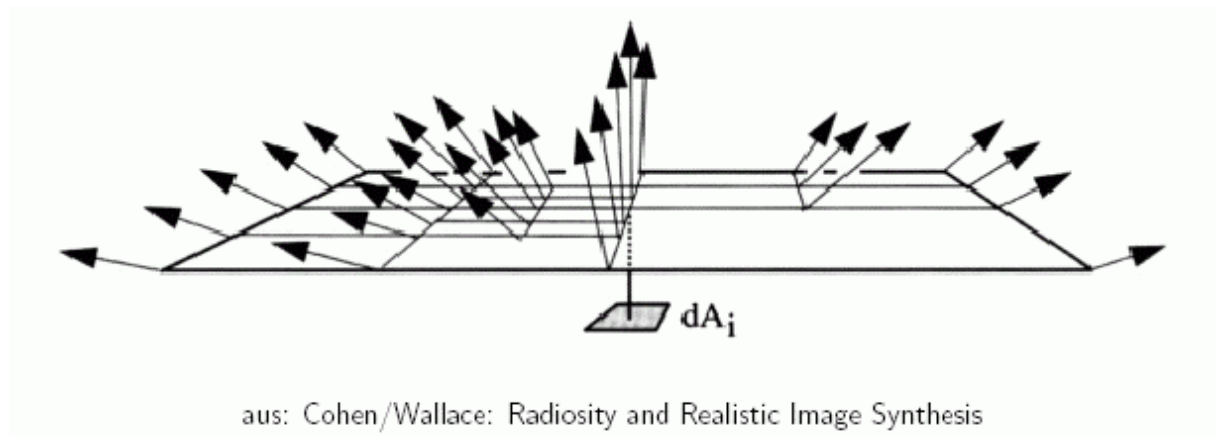
mittels eines Hemi-Cube werden alle Formfaktoren eines Flächenelements zu allen anderen Flächen bestimmt
 ⇒ ideal für Progressive Refinement: ein Hemi-Cube reicht aus, um die Strahlung der hellsten Fläche an alle anderen zu verteilen

Berechnung der Delta-Formfaktoren und Genauigkeitsbetrachtungen: siehe Encarnação et al. II, S. 175 ff.

Modifikationen des Hemi-Cube-Verfahrens:

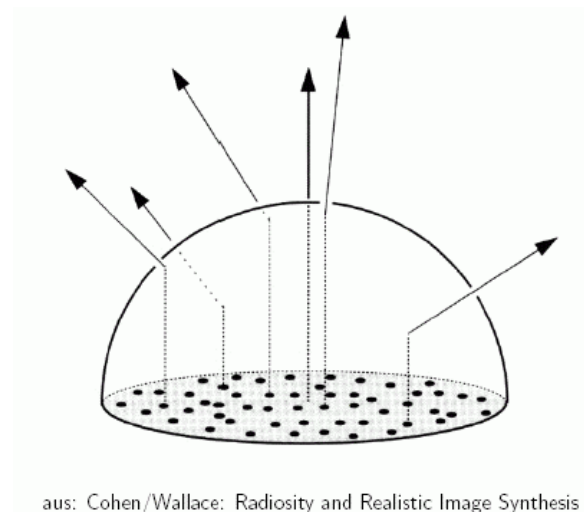
Sillions Verfahren:

- verwendet statt Halbwürfel nur noch eine Fläche (single plane method)



- ersetzt z-Buffer durch Warnock-Algorithmus
- \Rightarrow flexibler, geringere Aliasing-Probleme
- Nachteil: Elemente am Horizont werden übergangen (nicht so schlimm, sie tragen meist wenig zur Gesamthelligkeit bei)

Monte-Carlo-Raytracing mit reverser Nusselt-Analogie:



- flexibel, wichtige Teile werden genauer gesampelt

Vorgehensweise bei Veränderung der Szene:

1. Veränderung der Beleuchtung

- Koeffizientenmatrix des Gleichungssystems (Formfaktoren, Reflektivitäten) bleibt unverändert
- aber neue rechte Seite E
- Gleichungssystem muss neu gelöst werden
- sind die Änderungen von E gering, ist die alte Lösung B eine gute Ausgangslösung für die Iteration
- weitere Möglichkeit: Koeffizientenmatrix invertieren, dann Lösung für beliebige rechte Seite schnell verfügbar.
- in vielen Situationen sinnvoll: separate Berechnung der Szene für jede Lichtquelle einzeln, skalierte Addition der Ergebnisse

2. Veränderung der Oberflächeneigenschaften (Reflektivitäten)

- Koeffizientenmatrix ändert sich, Formfaktoren müssen aber nicht neu berechnet werden
- kleine Änderungen: verwende altes B als Startlösung
- bei wenigen geänderten Flächen: unverteilte Energie verteilen

$$\Delta B_j = \frac{\rho_j^{neu} - \rho_j^{alt}}{\rho_j^{alt}} B_j$$

3. Veränderung der Geometrie

- Formfaktoren ändern sich (und damit die Matrix)
- schon eine Änderung kann gesamte Matrix verändern!
- für einige einfache Veränderungen Vereinfachungen möglich
- *Idee für sich bewegendes Objekt*: berechne Volumen, durch welches das Objekt sich bewegt. Während Formfaktor-berechnung markiert man die Faktoren, deren Flächen durch das Volumen in der wechselseitigen Sichtbarkeit verändert werden – bei Bewegung dann: nur markierte Formfaktoren neu berechnen!

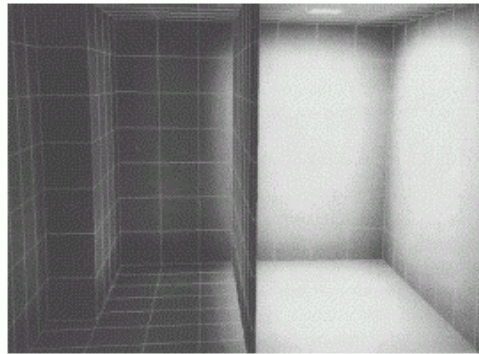
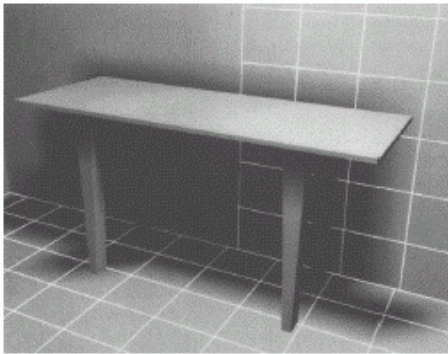
Farben im Radiosity-Verfahren

- Da die Reflektivitäten und Emissivitäten wellenlängen-abhängig sind, muss eigentlich das Gleichungssystem für jede Wellenlänge separat gelöst werden
- meist beschränkt man sich auf Lösungen für die 3 Grundfarben R, G, B

Problemfälle bei Radiosity

Diskontinuitäten: abrupte Sichtbarkeits-Veränderungen an Kanten werden vom Finite-Elemente-Verfahren oft nicht korrekt umgesetzt

Lichtleck (Light-leaks), wenn über eine Diskontinuität interpoliert wird

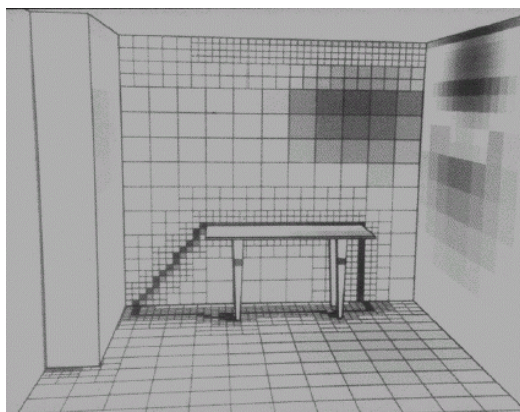


Fehler bei Diskontinuitäten nullter Ordnung, Lichtlecks

aus: Cohen/Wallace: Radiosity and Realistic Image Synthesis

Abhilfe:

verfeinerte Unterteilung der Flächen (feineres Elemente-Gitter)
adaptive Verfeinerung des Gitters an kritischen Stellen



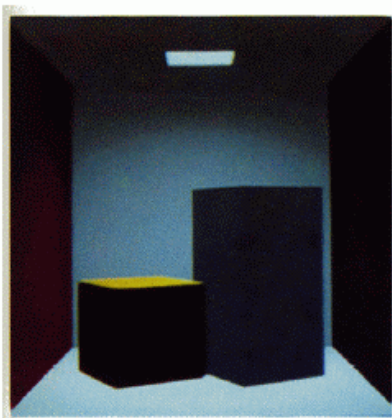
Bewertung des Radiosity-Verfahrens

- als erstes Verfahren der grafischen Datenverarbeitung bietet es die Möglichkeit, indirekte Beleuchtung und die Wirkung von flächenhaften diffusen Lichtquellen effizient zu berechnen
- ⇒ insbes. bei Innenräumen: wesentlich gesteigerter Realismus
- physikalisch basiert
- wichtigstes Verfahren zur Beleuchtungsplanung von Gebäuden
- großer Vorteil: Unabhängigkeit von den Beobachtungs-Parametern ⇒ virtuelles Durchwandern berechneter Szenen in Echtzeit möglich

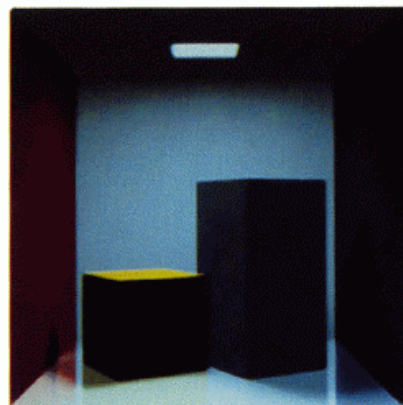
Nachteile:

- in Rechenzeit und Speicherbedarf aufwändiger als Raytracing, kompliziertere Datenstrukturen und Algorithmen
- Artefakte (Aliasing, Lichtlecks) erfordern Vorsichtsmaßnahmen
- Beschränkung auf rein diffuse Lichtausbreitung (kann aber überwunden werden durch Kombination von Radiosity mit Raytracing: "Zwei-Schritt-Verfahren")

Beispiele für Vergleich der Verfahren (aus Deussen 2001):

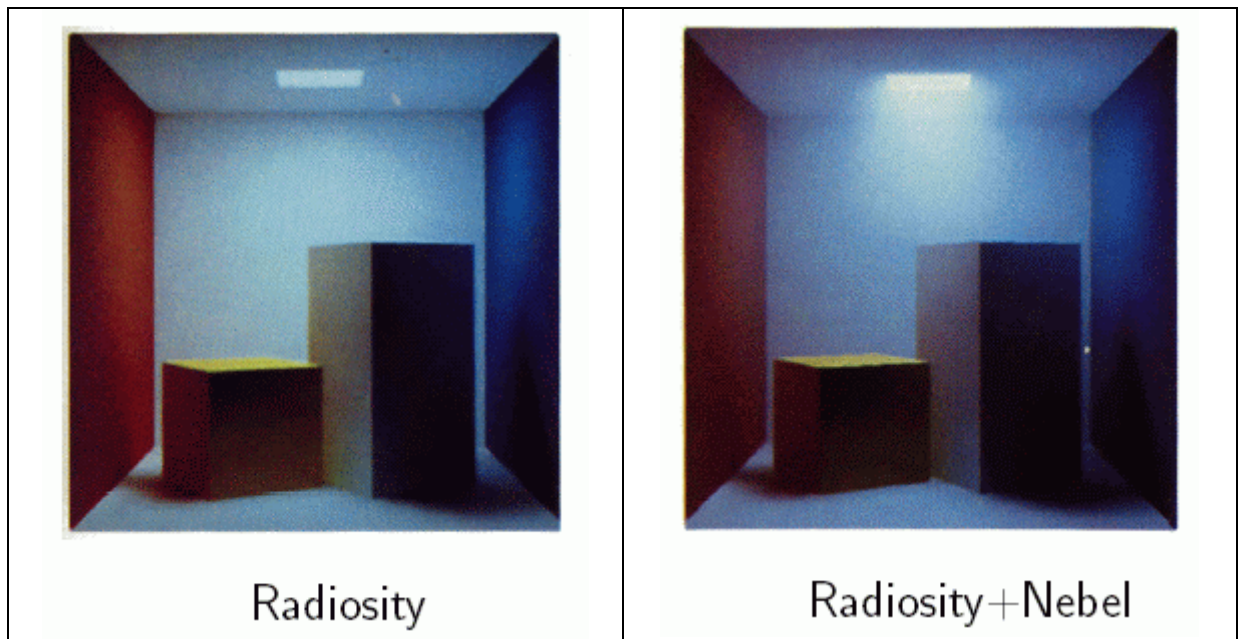


Lokales Modell



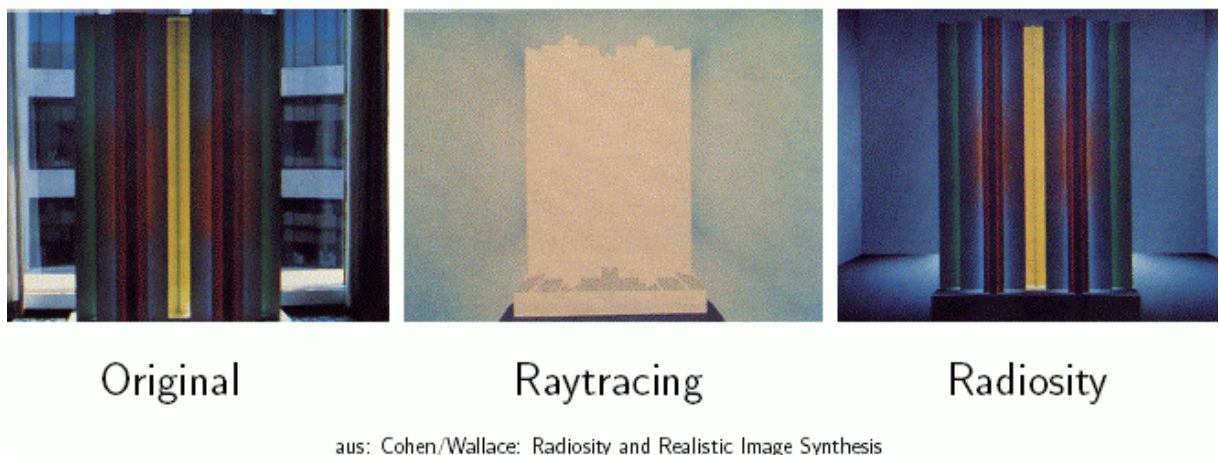
Raytracing

aus: Cohen/Wallace: Radiosity and Realistic Image Synthesis



Überlegenheit des Radiosity-Verfahrens in bestimmten Szenen:

Kunstwerk von John Ferren: Vorderseite weißer Stoff
 Farbe durch Beleuchtung von hinten mit farbigen Stoffen
 Diffus-Diffuse Interaktion



Ansätze zur weiteren Verbesserung des Verfahrens

(a) Kombination mit Raytracing

Einfaches, getrenntes Durchrechnen von Raytracing und Radiosity und anschließendes Addieren der Intensitätswerte aus beiden Verfahren reicht nicht aus:

man benötigt Modellierung des Lichttransports

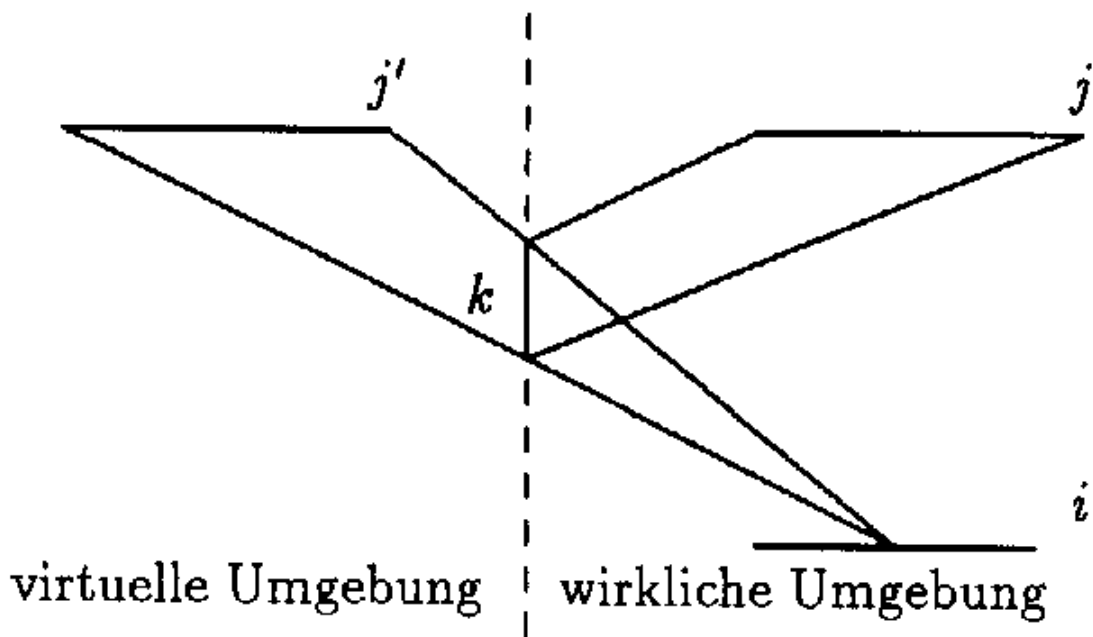
- diffus → diffus
- diffus → spiegelnd
- spiegelnd → diffus
- spiegelnd → spiegelnd

Ansatz: 2 Durchläufe

1. Durchlauf: modifiziertes Radiosity-Verfahren, "→ diffus"
2. Durchlauf: Raytracing-Verfahren mit Rückgriff auf Ergebnisse des 1. Durchlaufs, "→ spiegelnd".

Modifikationen für Radiosity für den 1. Durchlauf:

- Einbeziehung diffuser Transmission dazu Berechnung zusätzlicher "Rückwärts-Formfaktoren" (durch Hemi-Cube auf der Rückseite jedes durchsichtigen Flächenelements) – Zweck: Wiedergabe der diffusen Weitergabe von Licht durch (teilweise) transparente Flächen
- Einbeziehung des Lichteinfalls auf diffus reflektierende Flächen von spiegelnden Flächen:
für 2 diffus reflektierende Flächen i und j , die voneinander Licht über ein spiegelndes Flächenelement k erhalten können, wird ein zusätzlicher Formfaktor berechnet, der den über k laufenden Lichttransport beschreibt. Berechnung dieser Formfaktoren über eine durch k def. virtuelle Umgebung, die bei der Berechnung der Formfaktoren für i ein zusätzliches Flächenelement j' bereitstellt \Rightarrow der indirekte Lichttransport über k wird als direkter Lichttransport zwischen j' und i im Formfaktor $F_{ij'}$ beschrieben (s. Abb.).



Variante des Raytracing für den 2. Durchlauf:

- *beam tracing* (Pyramidenstümpfe statt Einzelstrahlen, siehe Raytracing-Kapitel) mit Rasterisierung der Pyramiden-Grundfläche, Gewichtung der Gitterzellen (Zentrum am stärksten)
- sichtbare Flächenelemente werden für jeden verallgemeinerten Suchstrahl durch Variante des z-Buffer-Verfahrens ermittelt
- Intensitätswert eines Flächenelements enthält i.allg. diffuse und spiegelnde Komponente
- diffuse Komponente entnimmt man dem 1. Durchlauf
- spiegelnde Komponente wird durch die Rekursion des Raytracing ermittelt
- dabei Verminderung der Auflösung des Pyramiden-Pixelgitters mit fortschreitender Rekursionstiefe

Nachteile: höherer Speicherplatz- und Rechenzeitbedarf.

(b) Beschleunigung durch Parallelisierung

dazu bietet sich *Progressive Refinement*-Methode an:

- von n Prozessoren dient einer als Server, einer als Display-Prozessor, $n - 2$ als Clients
- jeder Prozessor erhält komplette Beschreibung der Szene
- nur der Server speichert die Radiositywerte der Flächenelemente ab
- Das Verfahren besteht aus "Runden", die solange durchgeführt werden, bis Konvergenz festgestellt wird
- Zu Beginn jeder Runde bestimmt der Server die $n - 2$ Flächenelemente mit der größten noch nicht abgegebenen Strahlungsenergie und schickt jedem Client eines davon
- jeder Client berechnet die Formfaktoren von dem ihm zugewiesenen Element i zu allen anderen Elementen (eine Zeile der Formfaktormatrix) und schickt diese an den Server
- der Server berechnet daraus neue Radiositywerte für die Elemente durch Verteilen der auf i angesammelten Radiosity
- Server sendet regelmäßig update-Information (aktuelle Radiosity-Werte) an den Display-Prozessor

Effizienz des Verfahrens nimmt mit wachsender Zahl von Clients ab – Optimum suchen (szenenabhängig).