# Teil 9: Radiosity

## Simulation globaler Lichtverteilung



## Radiosity

Lösen globaler diffuser Beleuchtung in abgeschlossenem System

Gut geeignet für Innenraumsimulationen

Algorithmik und Ergebnisbilder doch sehr verschieden zu Ray Tracing

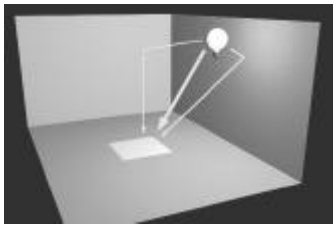


## Prinzip

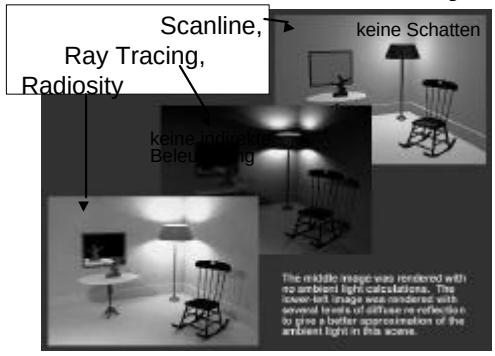
Lichquellen & Objektoberflächen werden als flächenhafte Strahler betrachtet, die

- reflektierte
- emittierte

Energie in die Umgebung abstrahlen.



## Scanline / RT / Radiosity






## Radiosity Beleuchtungsmodell (1)

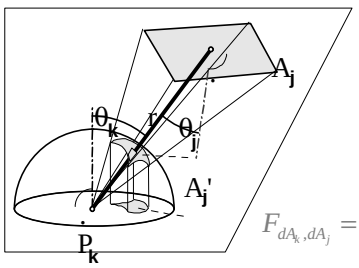
$$B_k = E_k + \rho_k \sum_{j \neq k} F_{kj} B_j$$

| | |
|---|---|
| $B_k$ | Radiosity von patch k |
| $E_k$ | Eigenemmission von patch k |
| $\Sigma F_{kj} B_j$ | Beitrag von den anderen patches |
| $F_{kj}$ | Form-Faktor, Beitrag v. $B_j$ zu $B_k$ |
| $\rho_k$ | Reflexionsfaktor von patch k |



## Radiosity Beleuchtungsmodell (2)

**Form-Faktor $F_{kj}$: Beitrag von patch j zu patch k**

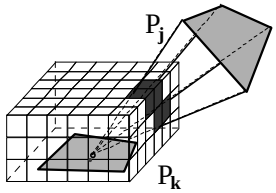



$$F_{dA_k, dA_j} = \frac{\cos\phi_j \cos\phi_k dA_j}{\pi r^2}$$

## Radiosity Beleuchtungsmodell (3)

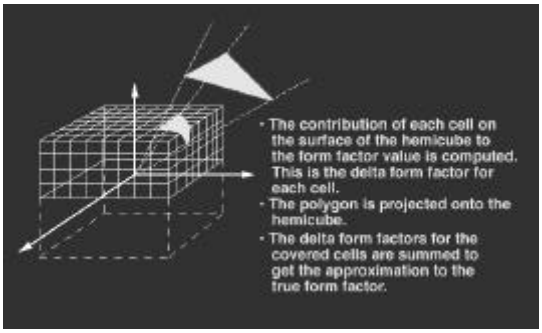
### Form-Faktor Berechnung

- teuerster Schritt bei Radiosity
- numerisch (Monte Carlo Methode)
- hemicube Ansatz






## Form-Faktor – Hemicube Ansatz






## Radiosity Beleuchtungsmodell (4)

### Form-Faktor Eigenschaften

- Gesamtenergie = konstant $\sum_{j=1}^{n} F_{kj} = 1$
- gleichförmige Reflexion $A_k F_{kj} = A_j F_{jk}$
- keine Selbstbeleuchtung $F_{kk} = 0$

## Radiosity Beleuchtungsmodell (5)

**Radiosity Gleichung**

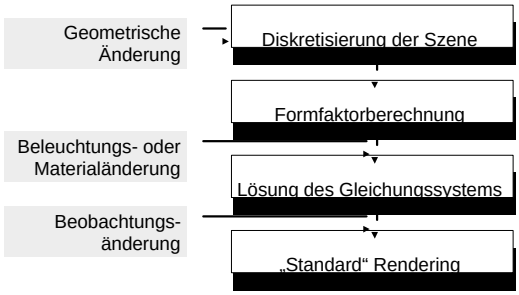
$$B_k = E_k + \rho_k \sum_{j \neq k} F_{kj} B_j$$

$$B_k - \rho_k \sum_{j \neq k} F_{kj} B_j = E_k$$

$$\begin{bmatrix} 1 - R_1F_{11} & -R_1F_{12} & \dots & -R_1F_{1n} \\ -R_2F_{21} & 1 - R_2F_{22} & \dots & -R_2F_{2n} \\ \vdots & & & \vdots \\ -R_nF_{n1} & -R_nF_{n2} & \dots & 1 - R_nF_{nn} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_n \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_n \end{bmatrix}$$



## Algorithmus
**(Radiosity)**

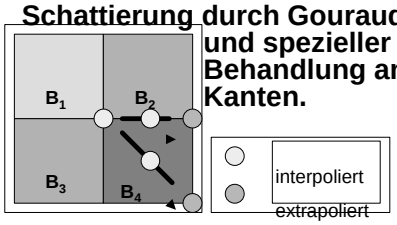





## Darstellung

**$B_i$‘s legen die Darstellungsfarbe fest**

**Schattierung durch Gouraud-Shading und spezieller Behandlung an den Kanten.**



## Radiosity Beleuchtungsmodell (6)

**Lösung der Radiosity Gleichung**

- Gauss-Elimination
- LU Faktorisierung

$$\begin{bmatrix} 1-R_1F_{11} & -R_1F_{12} & \dots & -R_1F_{1n} \\ -R_2F_{21} & 1-R_2F_{22} & \dots & -R_2F_{2n} \\ \vdots & \vdots & & \vdots \\ -R_nF_{n1} & -R_nF_{n2} & \dots & 1-R_nF_{nn} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_n \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_n \end{bmatrix}$$

**Zeit- und Platz-aufwendig**

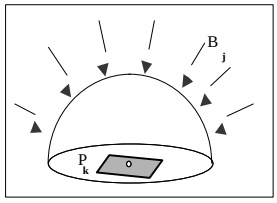


## Radiosity Beleuchtungsmodell (7)

**Lösen der Radiosity Gleichung**

- Gauss-Seidel Iteration

$$B_k^{i+1} = E_k + \rho_k \sum_{j \neq k} F_{kj} B_j^{i}$$




Gathering



## Progressive Refinement

**Generelle Methode:**

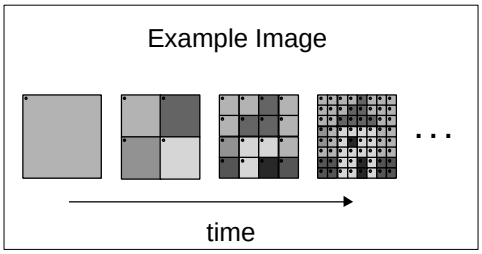
- Zuerst eine Näherungslösung bestimmen
- Näherungslösung sukzessive verfeinern

**Ansätze:**

- Ray Tracing: Verfeinerung der räumlichen Auflösung
- Radiosity: Verfeinerung der radiometrischen Auflösung

## Progressive Refinement

**(Ray Tracing)**



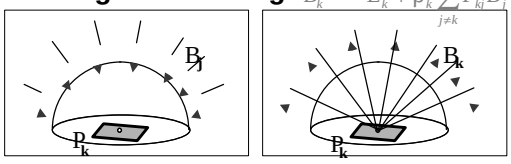


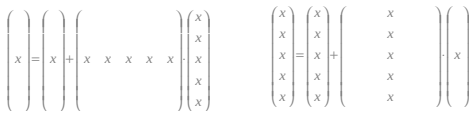

## Progressive Gathering




## Radiosity Beleuchtungsmodell (8)

**Gathering** vs. **Shooting** $B_k^{i+1} = E_k + \rho_k \sum_{j \neq k} F_{kj} B_j^{\ i}$



## Shooting – Algorithmus

1. Patch mit größter Energie auswählen
2. Eine Formfaktor-Spalte berechnen
3. Radiosity der umgebenden Patches aktualisieren
4. Solange bis Konvergenz



## Progressive Refinement Radiosity (1)

**Shooting**

- select brightest patch k and distribute it's radiosity $B_k$

$$B_k = E_k + \rho_k \sum_{j \neq k} F_{kj} B_j \Rightarrow \begin{array}{l} B_{k \text{ due to } B_j} = \rho_k F_{kj} B_j \\ B_{j \text{ due to } B_k} = \rho_j F_{jk} B_k \end{array}$$

$$\Downarrow$$

$$B_{j \text{ due to } B_k} = \rho_j F_{kj} \frac{A_k}{A_j} B_k \Leftarrow A_k F_{kj} = A_j F_{jk}$$



## Progressive Refinement Radiosity (2)

init $B_j := 0$, init $\Delta B_j :=$ emissions
for each patch k {
  do hemicube, calc. form factors $F_{kj}$
  for each patch j {
    $\Delta rad := \rho_j * F_{kj} * \Delta B_k * A_k / A_j$
    $\Delta B_j := \Delta B_j + \Delta rad$
    $B_j := B_j + \Delta rad$
  }
  $\Delta B_k := 0$
}



## Progressive Shooting



## Gathering vs. Shooting

**Gathering:**

- Gleichungssystem lösen (Gauss-Seidel)
- Start: $B_i$'s = 0, außer Lichtquellen
- Änderung einer Fläche pro Iteration

**Shooting:**

- Hellste Fläche selektieren
- Energie auf alle Flächen aufteilen
- Änderung aller Flächen pro Iteration



## Radiosity - Eigenschaften

+ Beliebige Flächenlichtquellen
+ Globale diffuse Beleuchtung
+ Schatten und Halbschatteneffekte
+ Unabhängig vom Betrachterstandpunkt
– Keine Spiegelreflexion und Transparenz
– Szene muß aus Polygonen bestehen

## Radiosity – Beispiele



## Radiosity – Beispiele



## Radiosity – Beispiele

## Radiosity – Beispiele



## Radiosity – Beispiele



## Radiosity – Beispiele

## Ray Tracing vs. Radiosity



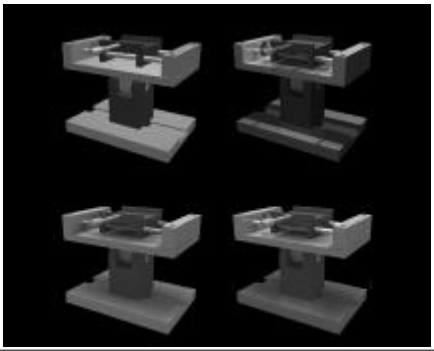
## Ray Tracing vs. Radiosity

| ***Ray Tracing*** | ***Kriterium*** | ***Radiosity*** |
|---|---|---|
| *abhängig* | *Beobachtungs-richtung* | *unabhängig* |
| *spiegelnd* | *geeignet für Refle-xionsart* | *diffus* |
| *beliebig* | *Szenenbeschrei-bung* | *B-Rep* |
| *jedes Bild* | *hoher Berech-nungsaufwand* | *jede Szene* |



## Ray Tracing / Radiosity

**Rad**

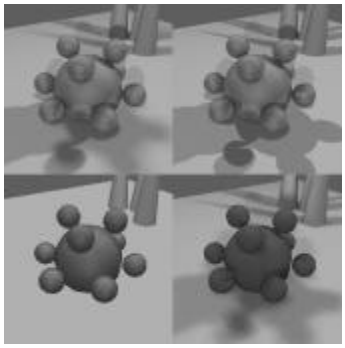


**RT**

Σ

## Ray Tracing / Radiosity

dRT



RT

Rad



## Radiosity Images



## Radiosity Images

## Radiosity Images




## Radiosity Images




## Radiosity Images


## Radiosity Images