Lösen der Matrizengleichung

$\longrightarrow$ letztes Kapitel: Berechnung der Formfaktoren $\mathbf{F}$

$\longrightarrow$ außerdem: $\mathbf{B}$: zu berechnende Strahlung, $\mathbf{E}$: gegebenes Eigenleuchten

nun:

Wie löst man $\mathbf{K}\,\mathbf{B} = \mathbf{E}$

Idee:

$$\mathbf{K}\,\mathbf{B} = \mathbf{E} \quad \longrightarrow \quad (\mathbf{I} - \mathbf{P}\,\mathbf{F})\,\mathbf{B} = \mathbf{E}$$

mit: $\mathbf{I}$: Einheitsmatrix, $\mathbf{P}\,\mathbf{F}$ Matrix aus Formfaktoren und Reflektivität

Zur Erinnerung:

$$\mathbf{K}: \begin{bmatrix} 1-\rho_1 F_{1,1} & -\rho_1 F_{1,2} & \cdots & -\rho_1 F_{1,n} \\ -\rho_2 F_{2,1} & 1-\rho_2 F_{2,2} & \cdots & -\rho_2 F_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ -\rho_n F_{n,1} & \cdots & \cdots & 1-\rho_n F_{n,n} \end{bmatrix}$$

$$\mathbf{I}: \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & 1 \end{bmatrix} \quad \mathbf{PF}: \begin{bmatrix} -\rho_1 F_{1,1} & -\rho_1 F_{1,2} & \cdots & -\rho_1 F_{1,n} \\ -\rho_2 F_{2,1} & -\rho_2 F_{2,2} & \cdots & -\rho_2 F_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ -\rho_n F_{n,1} & \cdots & \cdots & -\rho_n F_{n,n} \end{bmatrix}$$

Eigenschaften von K

- Größe $n \times n$ bei n Basisfunktionen ($\approx$ Knoten)
- nicht dünn besetzt
  $\longrightarrow$ Werte sind Null, wenn ρ Null, oder $F_{ij} = 0$
  (d.h die Elemente sehen sich nicht)
- nicht symmetrisch
  $\longrightarrow$ Umwandlung in symmetrische Matrix:
  jede Zeile i mit A_i multiplizieren (es gilt $F_{ij}A_i = F_{ji}A_j$)
  $\longrightarrow$ Symmetrie wird im folgenden nicht explizit benutzt
- Diagonaldominant:

$$\sum_{\substack{j=1 \\ j \neq i}}^{n} |K_{ij}| \leq |K_{ii}|, \qquad \forall i$$

$\longrightarrow$ Diagonaldominanz ist die Voraussetzung, daß iterative Lösungsverfahren anwendbar sind

- Spektralradius bis Eins
  $\longrightarrow$ Spektralradius: Norm einer Matrix
  $\longrightarrow$ bestimmt die Größe des größten Eigenwertes
  $\longrightarrow$ Maß für Konvergenzgeschwindigkeit iterativer Verfahren
  $\longrightarrow$ hat $\mathbf{P}\,\mathbf{F}$ eine Norm kleiner Eins, so ist $\mathbf{K} = \mathbf{I} - \mathbf{P}\,\mathbf{F}$ invertierbar und die Neumannfolge von Multiplikationen mit $\mathbf{P}\,\mathbf{F}$ konvergiert

$$\mathbf{K}^{-1} = [\mathbf{I} - \mathbf{P}\,\mathbf{F}]^{-1} = \sum_{a=0}^{\infty} (\mathbf{P}\,\mathbf{F})^a, \qquad ||\mathbf{P}\,\mathbf{F}|| < 1$$

- gut konditioniert
  $\longrightarrow$ kleine Änderungen der Eingabewerte erzeugen kleine Änderungen der Ergebniswerte

Lösungsverfahren

Direkte (Gauß-) Lösung:

$\rightarrow O(n^3)$ $\Rightarrow$ nicht praktikabel

$\rightarrow$ außerdem numerisch schwierig

Iterative Verfahren

Verfahren (zu Lösen $\mathbf{K}\,\mathbf{B} = \mathbf{E}$):

Bilde: $\mathbf{r}^{(0)} = \mathbf{K}\,\mathbf{B}^{(0)} - \mathbf{E}$

hierbei $\mathbf{B}^{(0)}$: Startwert, $\mathbf{r}^{(0)}$: Restevektor

Entwickle: $\mathbf{B}^{(1)}$ aus $\mathbf{r}^{(0)}$ usw.

$\rightarrow$ ist $\mathbf{r}^{(\mathbf{i})} = 0$, so ist eine Lösung erreicht.

Jacobi-Iteration

$\longrightarrow$ für jede Zeile gilt:

$$\sum_j K_{ij} B_j = E_i$$

oder, wenn Element K_{ii} herausgezogen wird:

$$K_{ii} B_i = E_i - \sum_{\substack{j \\ j \neq i}} K_{ij} B_j$$

Division durch K_{ii} ergibt (plus iterative Schreibweise):

$$B_i^{(k+1)} = \frac{E_i}{K_{ii}} - \sum_{\substack{j \\ j \neq i}} \frac{K_{ij} B_j^{(k)}}{K_{ii}}$$

über den Restevektor r_i^k ausgedrückt:

$$B_i^{(k+1)} = B_i^{(k)} + \frac{r_i^{(k)}}{K_{ii}}$$

$\longrightarrow$ Jacobi-Iteration: wähle beliebige Zeile und relaxiere

$\longrightarrow$ pro Zeile: O(n)

$\longrightarrow$ viele Zeilenveränderungen notwendig

Gauss-Seidel-Verfahren

$\longrightarrow$ Variation der Jacobi-Iteration

$\longrightarrow$ hier: die neuesten Versionen der B_i werden gleich verwendet

Behandlung einer Zeile (Relaxierung eines r_i):

$$B_i^{(k+1)} = E_i - \sum_{j=1}^{i-1} K_{ij} \frac{B_j^{(k+1)}}{K_{ii}} - \sum_{j=i+1}^{n} K_{ij} \frac{B_j^{(k)}}{K_{ii}}$$

$\longrightarrow$ Aufwand für Update eines B_j: $O(n)$

$\Rightarrow$ ein Iterationsschritt: $O(n^2)$

Algorithmus:

für alle i: $B_i =$ Startwert

while (keine Konvergenz) {

für alle i:

$$B_i = E_i - \sum_{\substack{j=1 \\ j \neq i}}^{n} \frac{B_j K_{ij}}{K_{ii}}$$

}

Konvergenzkriterien:

$$||\mathbf{r}||_\infty < \varepsilon \quad \text{oder} \quad ||\mathbf{B}^{(k+1)} - \mathbf{B}^{(k)}||_\infty < \varepsilon$$

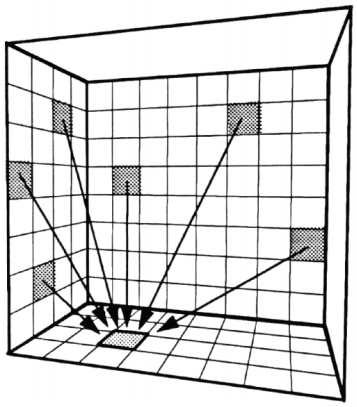
wobei ε festgelegte Fehlerschranke

es gilt ($K_{ii} = 1$, wenn Elemente sich nicht selbst "sehen"):

$$B_i = E_i - \sum_{\substack{j=1 \\ j \neq i}}^{n} \frac{B_j K_{ij}}{K_{ii}} = E_i + \sum_{\substack{j=1 \\ j \neq i}}^{n} \underbrace{\rho_i B_j K_{ij}}_{\Delta B_i}$$

$\longrightarrow$ ΔB_i: Anteil von Element j zur Strahlung von Element i

$\longrightarrow$ physikalisch betrachtet: Sammele Licht von allen anderen Flächen und addiere zum eigenen Leuchten



aus: Cohen/Wallace: Radiosity and Realistic Image Synthesis

Southwell-Iteration

$\longrightarrow$ Variante des Gauß-Seidel Verfahrens

$\longrightarrow$ wähle jeweils Zeile mit maximalem Fehler $|r_i|$

for (jedes i) $\{$
    $B_i = 0$;
    $r_i = E_i$;
$\}$

while (keine Konvergenz) $\{$
    i = Index mit $|r_i|$ *= max;*
    $B_i = B_i + r_i / K_{ii}$; /* üblicher Term */
    temp $= r_i$;
    für jedes j: $r_j = r_j - K_{ji}/K_{ii} *$ *temp;* /* update Fehler */
$\}$

Warum dieses Updating:

$\longrightarrow$ sei $\Delta\mathbf{B}^{(p)}$ als Veränderung des Strahlungsvektors bekannt

$$\mathbf{B}^{(p+1)} = \mathbf{B}^{(p)} + \Delta\mathbf{B}^{(p)}$$

$\longrightarrow$ dann gilt:

$$\mathbf{r}^{(p+1)} = \underbrace{\mathbf{E} - \mathbf{K}(\mathbf{B}^{(p)}}_{\mathbf{r}^{(p)}} + \Delta\mathbf{B}^{(p)}) = \mathbf{r}^{(p)} - \mathbf{K}\Delta\mathbf{B}^{(p)}$$

$\longrightarrow$ nur ein B_i wurde verändert: in $\Delta\mathbf{B}^{(p)}$ alle Einträge Null bis auf ΔB_i

also:

$$r_j^{(p+1)} = r_j^{(p)} - \frac{K_{ji}}{K_{ii}}\, r_i^{(p)} \qquad = r_j^{(p)} + \rho_i\, F_{ji}\, r_i^{(p)}, \quad \forall j$$

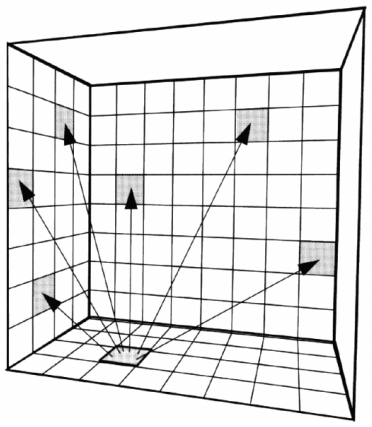
Was geschieht eigentlich mit $r_j^{(p)}$?

$$r_j^{(p+1)} = r_j^{(p)} + \rho_i \, F_{ji} \, r_i^{(p)} \quad = r_j^{(p)} + \rho_i \, F_{ij} \frac{A_i}{A_j} \, r_i^{(p)}, \quad \forall j$$

oder:

$$r_j^{(p+1)} A_j = r_j^{(p)} A_i + \rho_i \, F_{ij} \, r_i^{(p)} \, A_i, \quad \forall j$$

$\longrightarrow$ physikalisch betrachtet: noch unverbrauchte Energie $(r_i^{(p)} \times A_i)$ wird von Fläche i auf Flächen j verteilt:



aus: Cohen/Wallace: Radiosity and Realistic Image Synthesis

Progressive Refinement

$\longrightarrow$ Abwandlung der Southwell-Iteration

$\longrightarrow$ Ziel: nach jeder Iteration Feedback über gesamte Beleuchtung

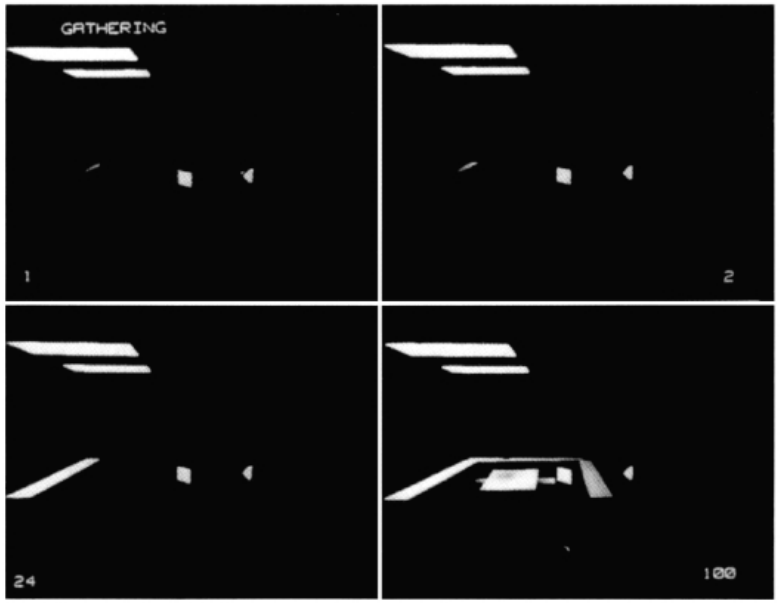
$\longrightarrow$ Bestimme für jedes Element B_i und ΔB_i (Teil von B_i, der pro Iteration auf andere Elemente verteilt werden soll)

$\longrightarrow$ pro Iteration: Element mit maximaler Restenergie ($\Delta B_i * A_i$) wird ausgewählt und ΔB_i wird auf die Elemente verteilt

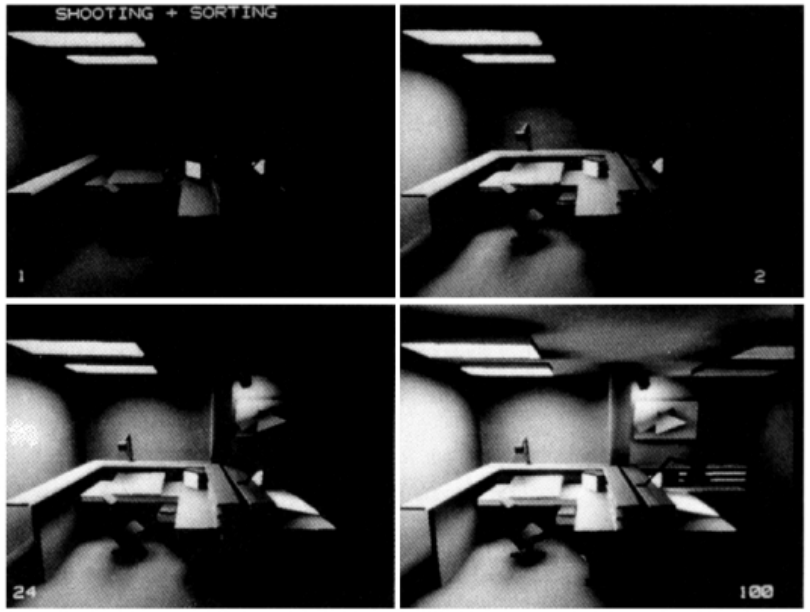
Algorithmus

for (jedes i) $\{$
$B_i = E_i$;
$\Delta B_i = E_i$;
$\}$ *while (keine Konvergenz)* $\{$
i = Element mit maximalem $\Delta B_i * A_i$;
für jedes j: $\{$
$\Delta rad = \Delta B_i * \rho_j F_{ji}$;
$\Delta B_j = \Delta B_j + \Delta rad$;
$B_j = B_j + \Delta rad$;
$\}$ $\Delta B_i = 0$;
Stelle Bild mit B_j *als Intensitäten von Element* j *dar*
$\}$

Vergleich Gauss-Seidel und progressive Refinement



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



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

aus: Cohen/Wallace: Radiosity and Realistic Image Synthesis

Addition ambienter Energie

$\longrightarrow$ ambienter Term wird hinzugefügt

$\longrightarrow$ grobe Approximation des reflektierten Lichts, das momentan noch nicht in Berechnung einbezogen wurde

$$\overline{\Delta B} = \frac{\sum r_i A_i}{\sum A_i} \quad \text{(gewichtete Summe)}$$

$\overline{\Delta B}$: durchschnittliche Energie, die noch nicht verteilt wurde

Teil der Energie, der reflektiert wird:

$$\bar{\rho} = \frac{\sum \rho_i \, A_i}{\sum A_i}$$

Fortgesetzte Reflexion:

$$R_{total} = 1 + \bar{\rho} + \bar{\rho}^2 + \bar{\rho}^3 + ... = \frac{1}{1 - \bar{\rho}}$$

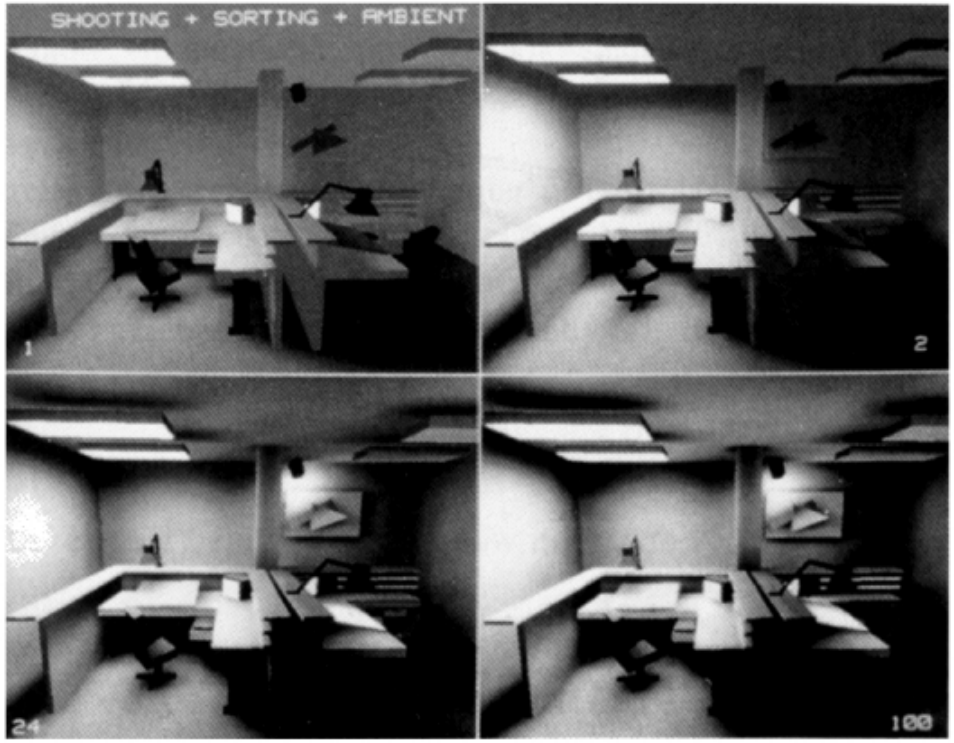
Ambient hinzuzufügende Strahlung:

$$B_{ambient} = \overline{\Delta B} \cdot R_{total}$$

$\longrightarrow$ jedes Element i reflektiert einen Teil ρ_i dieser Strahlung

$$B_i^{display} = B_i + \rho_i \, B_{ambient}$$

Progressive Refinement mit ambienter Energie

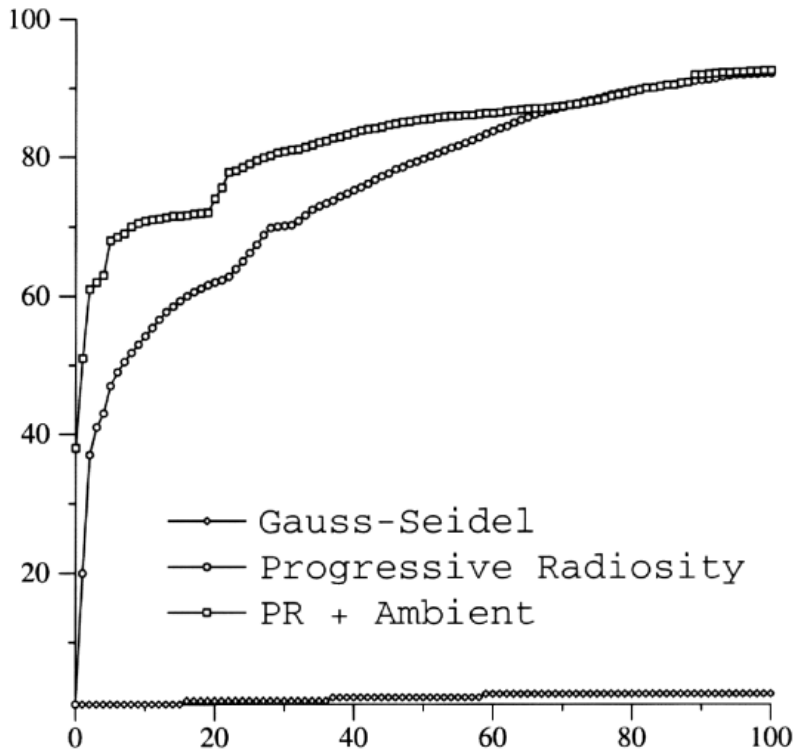


Displayed Image after 1, 2, 24 and 100 Steps

aus: Cohen/Wallace: Radiosity and Realistic Image Synthesis

Vergleich der Verfahren

visuelle Konvergenz der Verfahren:



aus: Cohen/Wallace: Radiosity and Realistic Image Synthesis

Überrelaxation

$\longrightarrow$ Verallgemeinerung des Relaxierungsschrittes über ω:

$$B_i^{(k+1)} = B_i^{(k)} + \omega \frac{r_i^{(k)}}{K_{ii}}$$

$$r_i^{(k+1)} = (1 - \omega)\, r_i^{(k)}$$

Überrelaxation: $\omega > 1$

$\longrightarrow$ schnellere Relaxierung für stabile Systeme

Unterrelaxation: $\omega < 1$

$\longrightarrow$ stabilere Relaxierung für instabile Systeme

Veränderung der Szene

drei Arten der Veränderung:

1. Veränderung der Beleuchtung

   $\longrightarrow$ **K** bleibt unverändert
   $\longrightarrow$ $\mathbf{K}\,\mathbf{B} = \mathbf{E}$ muß neu gelöst werden
   $\longrightarrow$ sind die Änderungen von **E** gering, ist das alte **B** eine gute Ausgangslösung

   $\longrightarrow$ bei vielen Änderungen u.U seperate Berechnung der Szene für alle Lichtquellen, skalierte Addition der Ergebnisse

   $\longrightarrow$ weitere Möglichkeit: **K** invertieren

2. Veränderung der Oberflächeneigenschaften (Reflektivität)

   $\longrightarrow$ **K** ändert sich
   $\longrightarrow$ Formfaktoren müssen aber nicht neu berechnet werden
   $\longrightarrow$ kleine Änderungen: verwende altes **B** als Startlösung
   $\longrightarrow$ bei wenigen geänderten Flächen:
   unverteilte Energie $\Delta B_j A_j$ verteilen:

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

3. Veränderung der Geometrie

   $\longrightarrow$ Formfaktoren und **K** ändern sich
   $\longrightarrow$ potentiell: eine Änderung verändert gesamte Matrix
   $\longrightarrow$ einige Vereinfachungen für einfache Veränderungen

Idee für ein sich bewegendes Objekt:

$\longrightarrow$ Berechne Volumen durch welches das Objekt sich bewegt

$\longrightarrow$ während Formfaktorberechnung: markiere Faktoren, deren Flächen durch das Volumen in der wechselseitigen Sichtbarkeit verändert werden.

$\longrightarrow$ bei Bewegung: nur markierte Formfaktoren werden neu berechnet