Grafika Komputerowa

Radiosity (Metoda Energetyczna)

Alexander Denisjuk

denisjuk@pjwstk.edu.pl

Polsko-Japońska Wyższa Szkoła Technik Komputerowych

zamiejscowy ośrodek dydaktyczny w Gdańsku

ul. Brzegi 55

80-045 Gdańsk

Radiosity (Metoda Energetyczna)

Najnowsza wersja tego dokumentu dostępna jest pod adresem

`http://users.pjwstk.edu.pl/~denisjuk/`

Radiosity

Podział na płaty

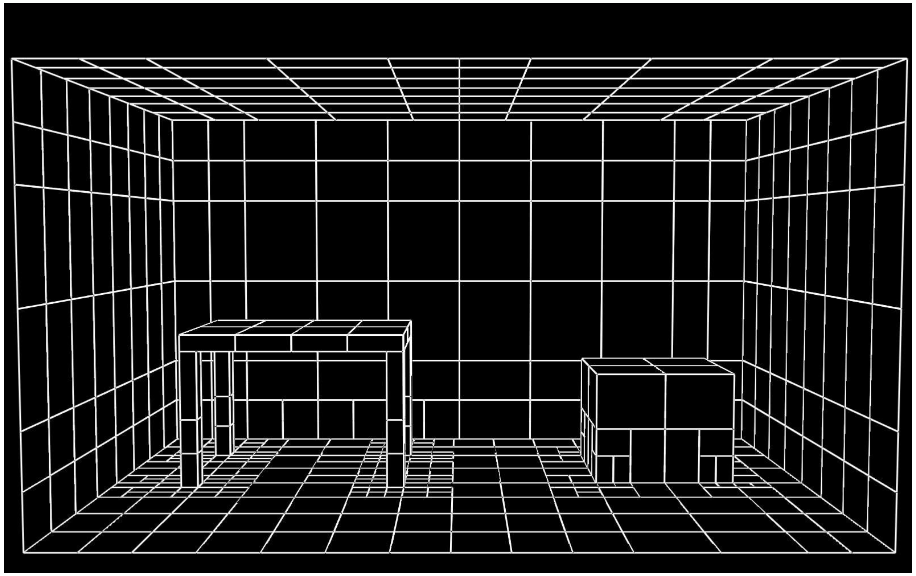

Figure XI.1: The patches used to render the radiosity scene of figures XI.2 and XI.3. See color plate C.13.

Cieniowanie płaskie

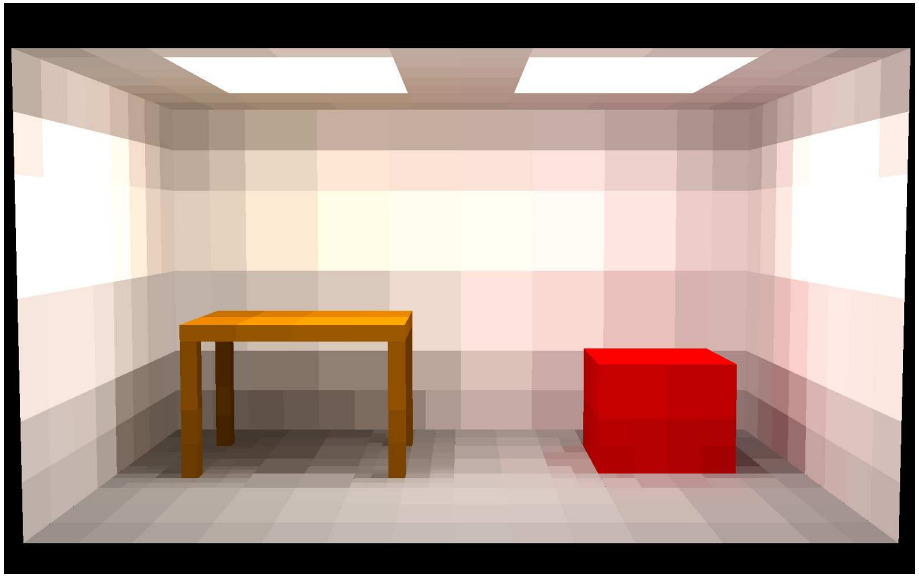

Figure XI.2: A radiosity rendered figure, with flat shading. It is evident that this image is based on the patches shown in figure XI.1. See color plate C.14.

Cieniowanie z interpolacją

Figure XI.3: A radiosity rendered figure, with smooth shading of illumination. The red color of the box is reflected onto the nearby walls, giving them a slight reddish hue. This is based on the patches shown in figure XI.1. See color plate C.15.

Podział na płaty

1. równomiernie oświetlone
2. wystarczająco małe
3. zużycie pamięci, obliczalna skompikowalność: $O(n^2)$

Rownanie energetyczne

- Wybrano płaty $P_1, \ldots, P_n$ o polach poweirzchni $A_1, \ldots, A_n$.
- B_i jest średnim światłem promieniowanym przez płat P_i.
- $B_i = E_i + R_i \cdot B_i^{\mathrm{in}}$

Współczynnik sprzężenia optycznego

- $F_{i,j}$ — ilość światła, przekazywanego z płata P_i.
- $F_{i,i} = 0$.

Rownanie energetyczne

- Całkowite światło emitowane z P_j: $A_j B_j$.
- Całkowite światło padające na P_i: $A_i B_i^{\mathrm{in}}$.
- $A_i B_i^{\mathrm{in}} = \sum_{j=1}^{n} F_{j,i} A_j B_j$.
- Równanie wzajemności: $A_i F_{i,j} = A_j F_{j,i}$.
- $B_i^{\mathrm{in}} = \sum_{j=1}^{n} F_{i,j} B_j$.
- Równanie radiosity: $B_i = E_i + R_i \sum_{j=1}^{n} F_{i,j} B_j$.

Radiosity equation

- $B = E + MB$.
- $B = \begin{pmatrix} B_1 \\ \vdots \\ B_n \end{pmatrix}, E = \begin{pmatrix} E_1 \\ \vdots \\ E_n \end{pmatrix}$.
- $M = \big(R_i F_{i,j}\big)$.
- $(I - M)B = E \Rightarrow B = (I - M)^{-1}E$.

Algorytm radiosity

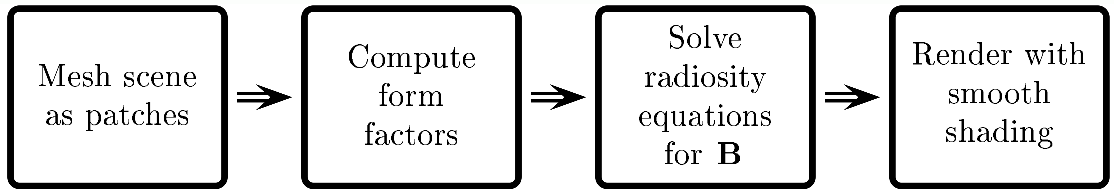

Figure XI.4: The four stages of the radiosity algorithm

Obliczenie współczynników sprzężenia optycznego

- $F_{i,j} \in [0,1]$, $\sum_{j=1}^{n} F_{i,j} = 1$.
- $F_{1,2} \approx 0$, $F_{2,1} \approx 1$.

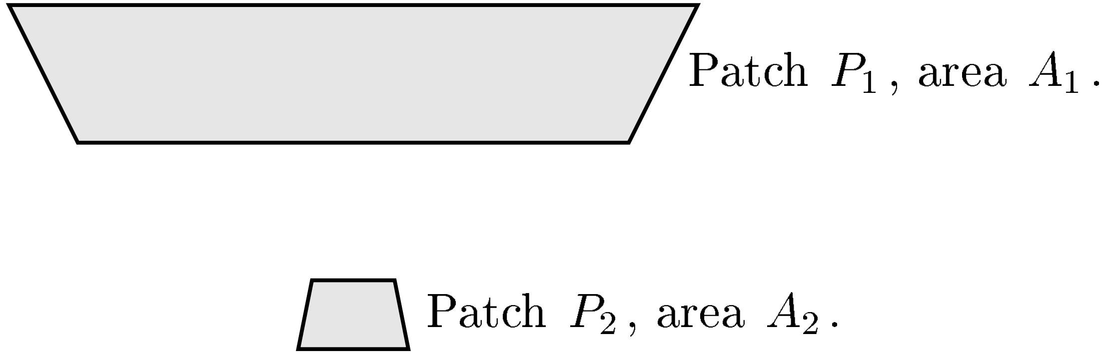

Figure XI.5: A large patch and small patch.

Obliczenie współczynników sprzężenia optycznego

- $\varphi_i, \varphi_j < 90^\circ$.
- pole zrzutu P_j na sferę jednostkową w P_i: $\cos\varphi_j A_j / d^2$.
- część pola zrzutu P_j: $\frac{(\cos\varphi_j) A_j}{2\pi d^2}$.

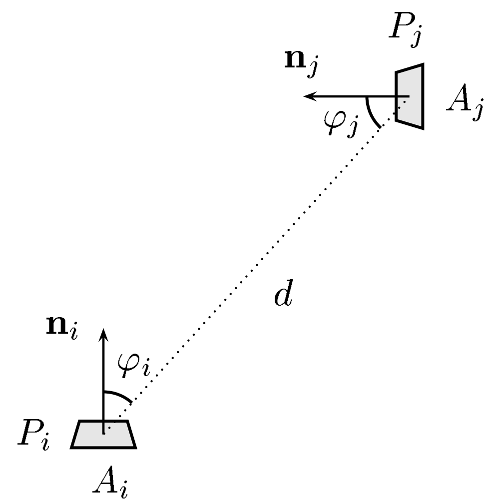

Figure XI.6: Two infinitesimally small patches P_i and P_j with areas A_i and A_j. The normals to the patches are $\mathbf{n}_i$ and $\mathbf{n}_j$.

Obliczenie współczynników sprzężenia optycznego

- $F_{i,j} \sim \cos\varphi_i \frac{(\cos\varphi_j)A_j}{2\pi d^2}$.
- Całkowita ilość emitowanego światła jest 1.
- $\int_{S_i+} \cos\varphi\, dA = \int_0^{\pi/2} \cos\varphi(2\pi\sin\varphi)\, d\varphi = \pi$.
- $F_{i,j} = \cos\varphi_i \frac{(\cos\varphi_j)A_j}{\pi d^2} \Rightarrow F_{i,j}A_i = F_{j,i}A_j$.

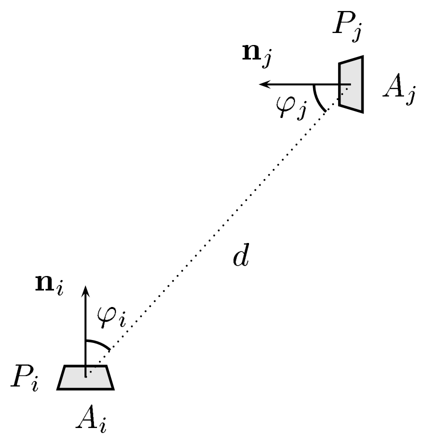

Figure XI.6: Two infinitesimally small patches P_i and P_j with areas A_i and A_j. The normals to the patches are $\mathbf{n}_i$ and $\mathbf{n}_j$.

Ograniczenia

- Płaty są małe w porównaniu do g.
- Nie uwzględnia się widoczność płat.

Widoczność płat. Ray tracing

- $F_{i,j} = V_{i,j} \cos\varphi_i \frac{(\cos\varphi_j)A_j}{\pi d^2}$.
- $V_{i,j}$ — współczynnik widoczności.
- $V_{i,j} = V_{j,i}$.
- jittering (fluktacje).

Współczynniki sprzężenia optycznego. Hemicube method

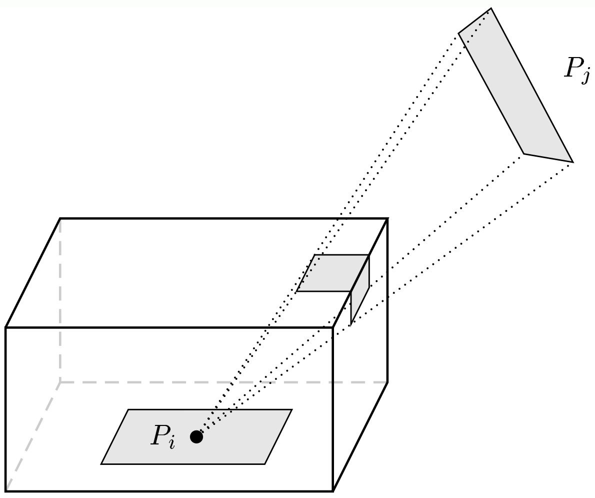

Figure XI.7: Projection onto a hemicube.

- P_j zastępiony przez projekcję.
- Z-buffor. Sprzętowa akceleracja.
- $F_{i,j} = \sum_{\text{pixele } P_j}$ część światłaP_i.

Hemicube method

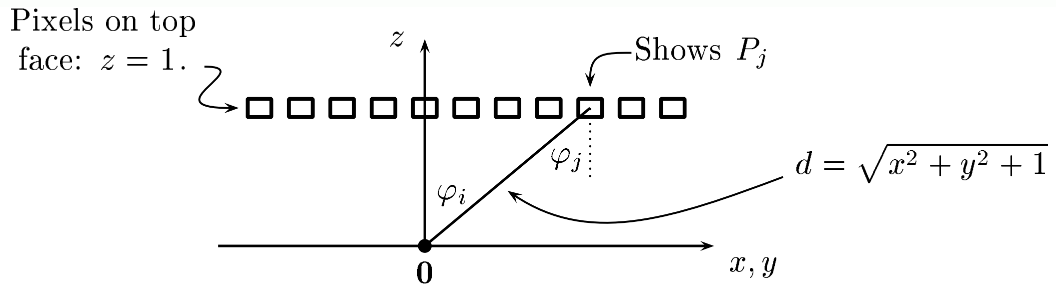

Figure XI.8: A row of pixels along the top of the hemicube. One pixel shows patch P_j. The origin is placed at the center of patch P_i. The top of the cube is the $z = 1$ plane.

- $\cos\varphi_i = \cos\varphi_j = 1/d$.
- $F_{i,j} = \sum_{\text{pixele } P_j} \frac{(1/d)(1/d)\langle\text{Pole pixeli}\rangle}{\pi d^2}$.
- $F_{i,j} = \sum_{\text{pixele } P_j} \frac{\langle\text{Pole pixeli}\rangle}{\pi d^4}$.

Hemicube method

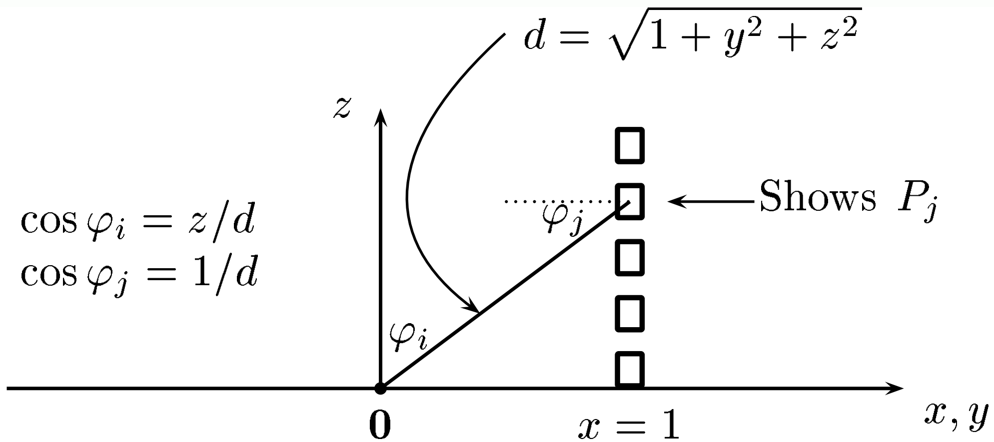

Figure XI.9: A row of pixels along the $x = 1$ side of the hemicube.

- $F_{i,j} = \sum_{\text{pixele } P_j} \frac{z\langle\text{Pole pixeli}\rangle}{\pi d^4}$.

Równanie radiosity. Metody iteracyjne

- $(I - M)B = E$.
- $B = (I - M)^{-1}E$, metody bezpośrednie $O(n^3)$.

Lemat 1. *Niech M będzie macierzą równania radiosity. Wtedy*

1. $m_{i,j} \geqslant 0$,
2. $0 \leqslant \mathrm{MaxRowSum}(M) < 1$,
3. $\mathrm{MaxRowSum}(M^k) < (\mathrm{MaxRowSum}(M))^k$, $k = 1, 2, \ldots,$

gdzie $\mathrm{MaxRowSum}(M) = \max_i \sum_j m_{i,j}$.

Równanie radiosity

Wniosek 2. *Niech* M *będzie macierzą równania radiosity. Wtedy*

1. $I - M$ *jest macierzą odwracalną,*
2. $(I - M)^{-1} = I + M + M^2 + \ldots.$

$$B \approx (I + M + \cdots + M^k)E$$

Metoda Jacobiego

```
// Bnew, B, E są wektorami
// m jest macierzą
    B=E;
    while (not osiągnięta dokładność){
        for (i=1;i<=n;i++){
            Bnew[i]=E[i]+∑_{j=1}^{n} m[i,j]*B[j];
        }
        B=Bnew;
}
```

$$Bnew[i] = E[i] + \sum_{j=1}^{n} m[i,j] * B[j]$$

Metoda Gaussa-Seidla

```
// Bnew, B, E są wektorami
// m jest macierzą
   B=E;
   while (not osiągnięta dokładność){
      for (i=1;i<=n;i++){
         B[i]=E[i]+∑_{j=1}^{n} m[i,j]*B[j];
      }
   }
```

Metoda Strzałów

```
// B, ΔB, E są wektorami, m jest macierzą
B=0;
ΔB=E;
while (not osiągnięta dokładność){
    wybierż j tak, żeby maksymalizować
        ΔB[j]∗A_j;
    B[j]+=ΔB[j];
    for (i=1;i<=n;i++){
        ΔB[i]=ΔB[i]+∑_{j=1}^{n} m[i,j]∗ΔB[j];
    }
    ΔB[j]=0;
    B+=ΔB;
}
```

RRV - Radiosity Renderer and Visualizer

- http://dudka.cz/rrv