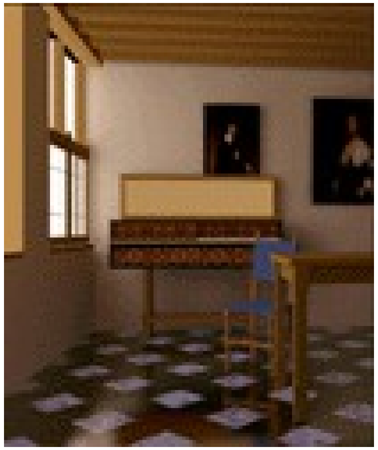
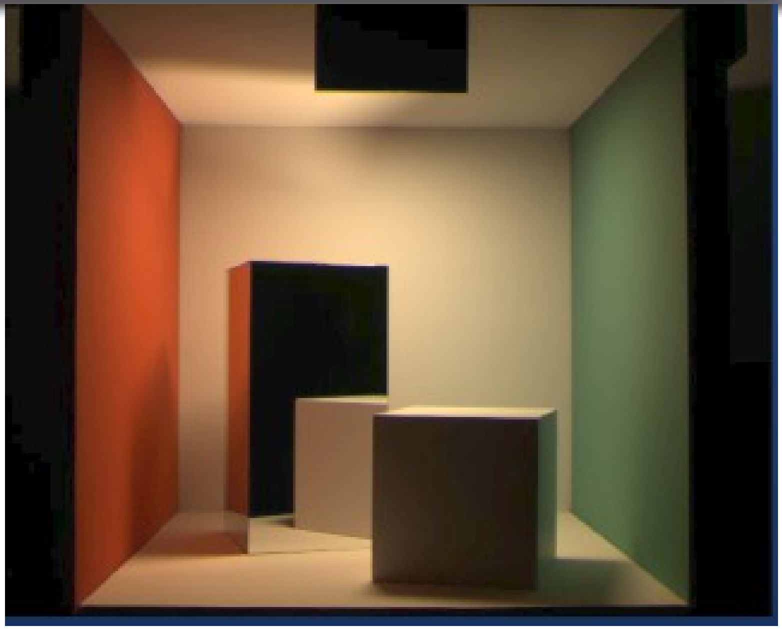
# Sevalna metoda
## (Radiosity)

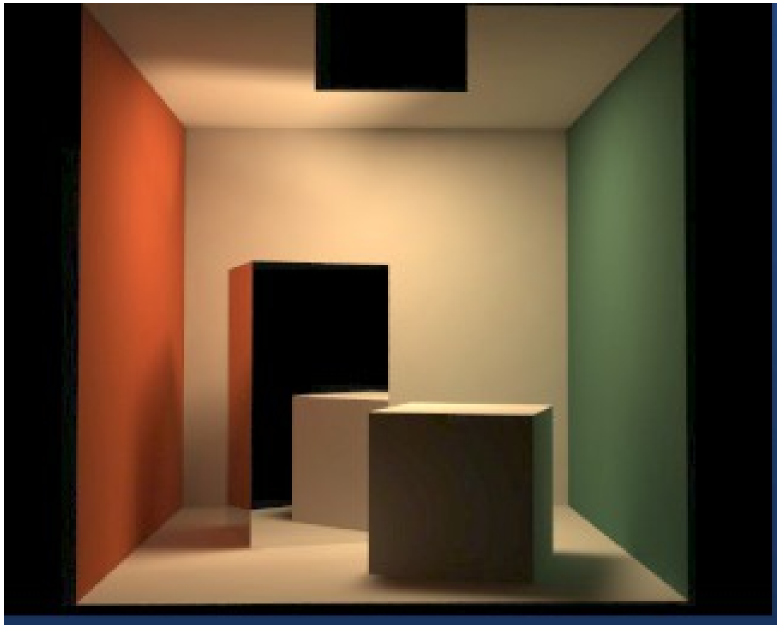
# Sevalna metoda (radiosity)

- Sledenje žarka modelira zrcalne odboje in lome skozi prosojne predmete, toda še vedno ne upošteva difuzne svetlobe
- Sevanje je delež energije, ki se od neke površine odda ali odbije
- S shranjevanjem svetlobne energije v neki količini, lahko efektu sevanja sledimo



# Koncept sevalne metode

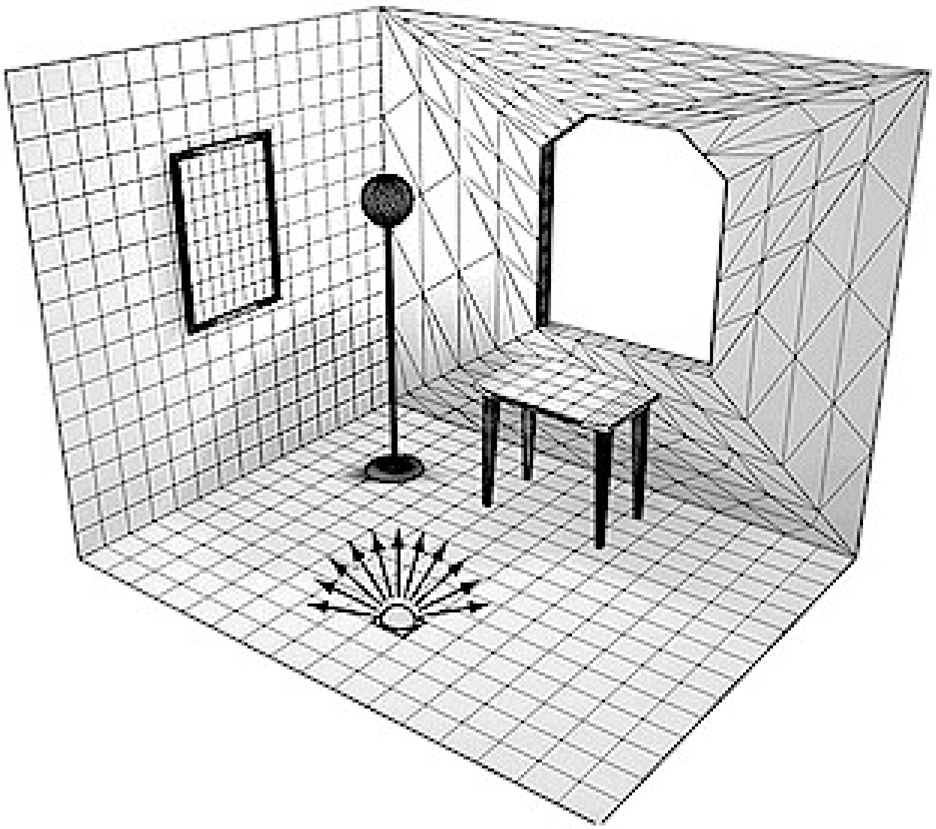




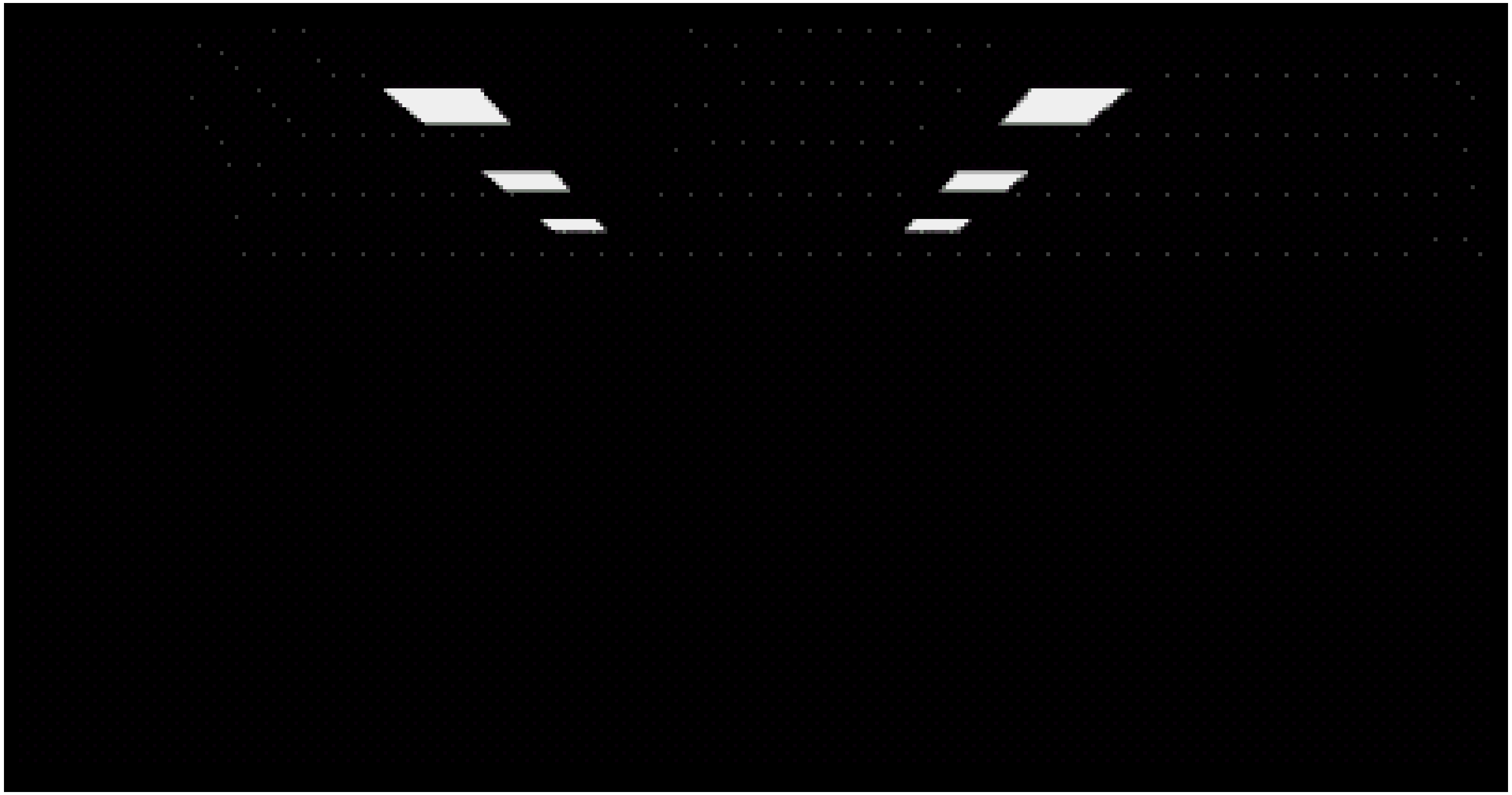
- Sevanje posamezne površine je odvisno od sevanja vseh ostalih površin
  - Globalno osvetlitev obravnavamo kot linearni sistem
  - Potrebna je konstantna BRDF, tj. Bi-directional Reflectance Distribution Function (difuzna svetloba)
  - Enačbo upodabljanja rešujemo kot problem matrik
- Postopek
  - Razdeli površine na mrežo ploskev (v nadaljevanju elementi)
  - Izračunaj snovne faktorje površin
  - Izračunaj sevanje
  - Prikaži ploskve

# Algoritem

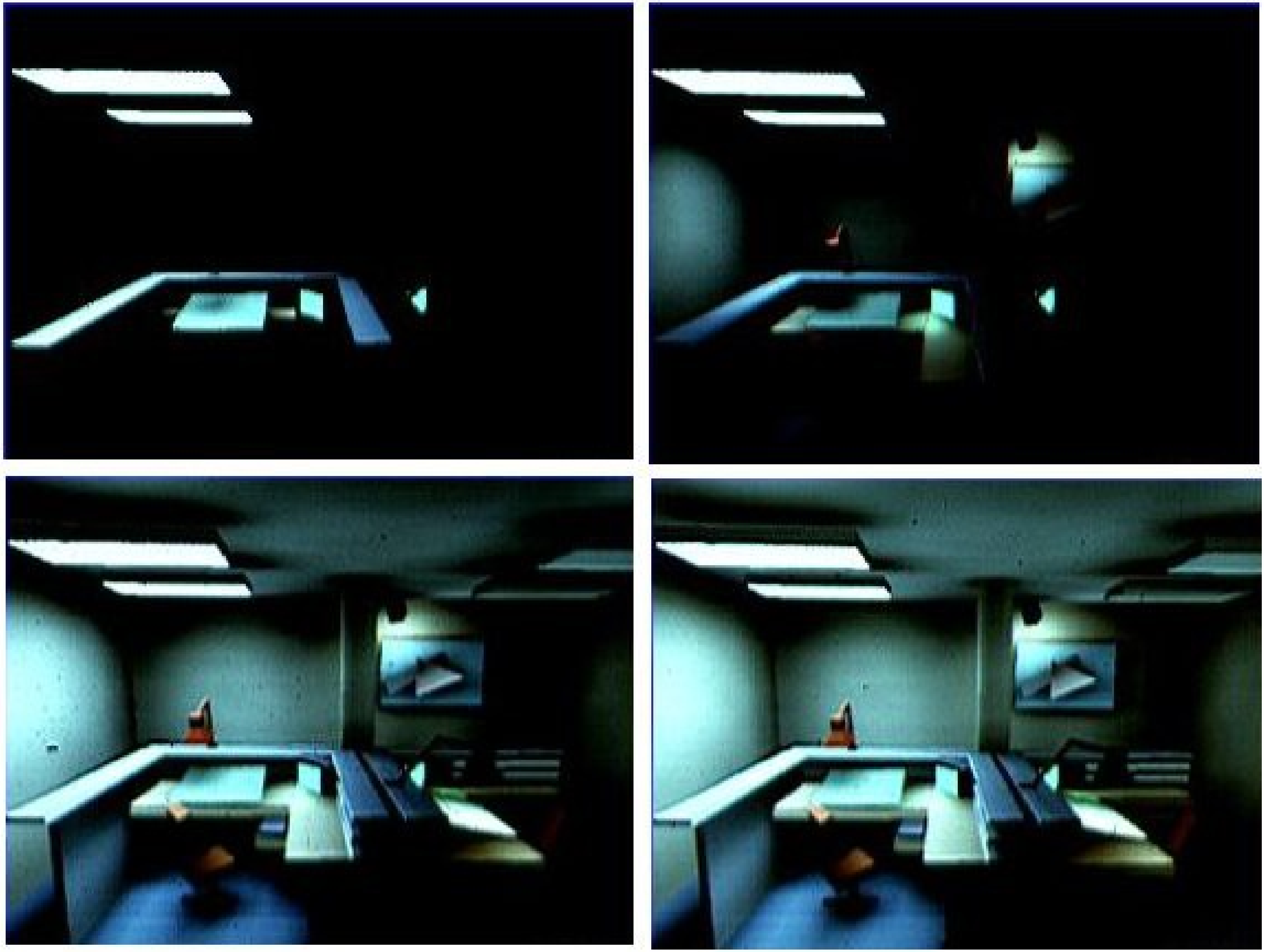
To dosežemo tako, da najprej razdelimo originalne ploskve na mrežo manjših ploskev, recimo jim elementi. Med postopkom nato računamo, kakšen prispevek svetlobe seva vsak posamezen element drugim elementom. Za vsak element mreže pomnimo seštevek takih sevalnih vrednosti.



# Demonstracija sevalne metode



# Demonstracija sevalne metode



# Osnove

Definition:

- The *radiosity* of a surface is the rate at which energy leaves the surface
  - Radiosity = rate at which the surface *emits* energy + rate at which the surface *reflects* energy

Simplifying assumptions

- Environment is closed
- All surfaces have *Lambertian* reflectance
- Surface patches emit and reflect light uniformly over their entire surface

# Sevalna metoda (radiosity)

For each surface *i*:

$$B_i = E_i + \rho_i \sum B_j F_{ji} (A_j / A_i)$$

where

$B_i$, $B_j$ = radiosity of patch *i, j*

$A_i$, $A_j$ = area of patch *i, j*

$E_i$ = energy/area/time emitted by *i*

$\rho_i$ = reflectivity of patch *i*

$F_{ji}$ = *Form factor* from *j* to *i*

# Faktorji oblike (Form Factors)

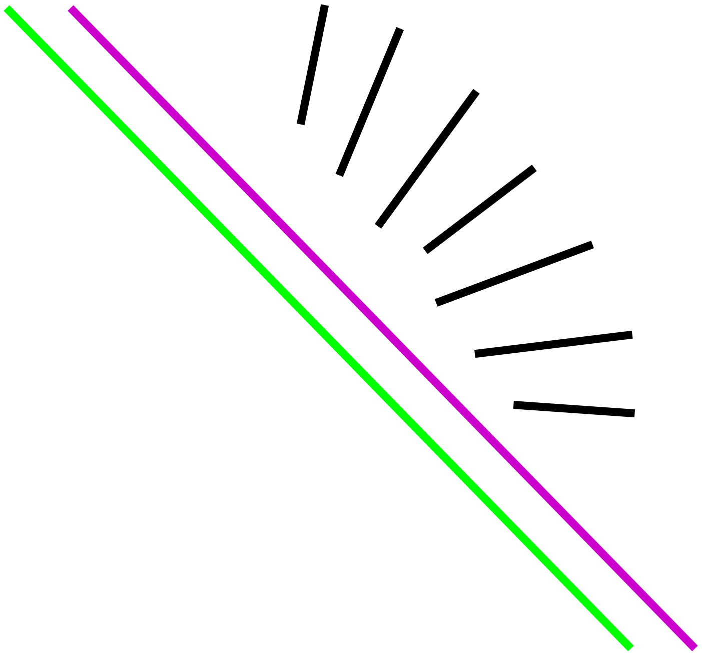
*Form factor*: fraction of energy leaving the entirety of patch *i* that arrives at patch *j*, accounting for:

- The shape of both patches
- The relative orientation of both patches
- Occlusion by other patches

We'll return later to the calculation of form factors

# Faktorji oblike (form factors)

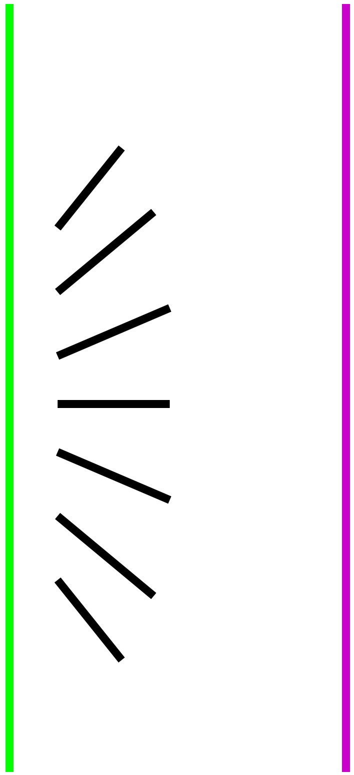
Nekaj primerov…



Faktor oblike:
približno 100%

# Faktorji oblike

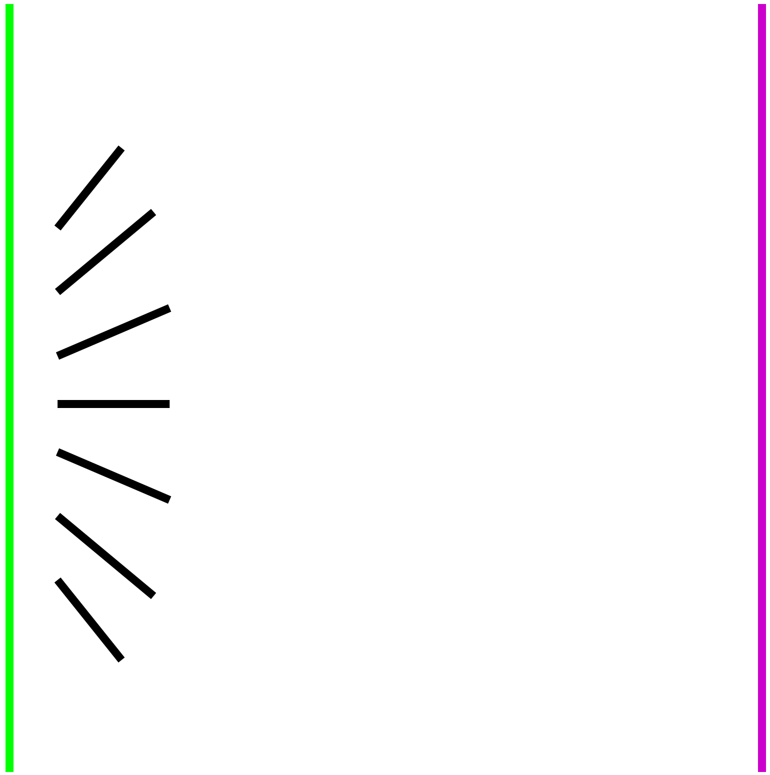
Nekaj primerov…



Faktor oblike:
približno 50%

# Faktorji oblike

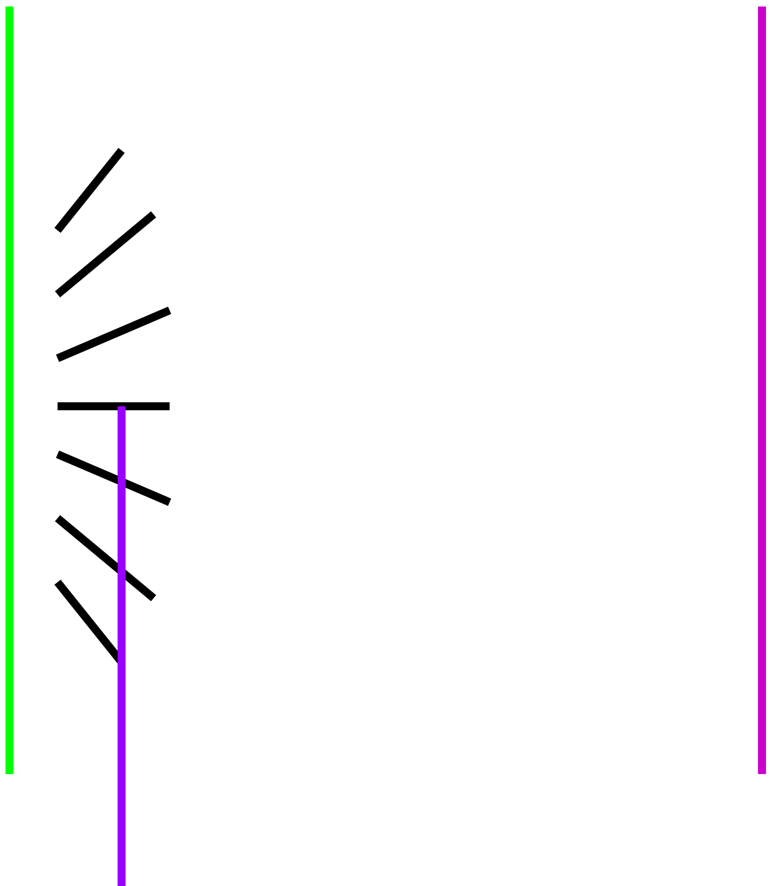
Nekaj primerov…



Faktor oblike:
Približno 10%

# Faktorji oblike

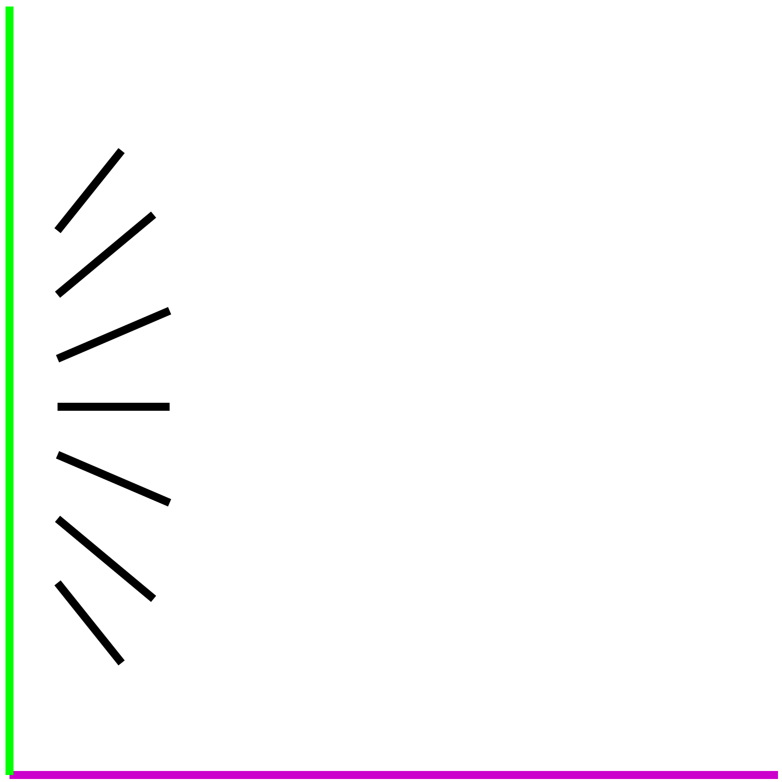
Nekaj primerov…



Faktor oblike:
Približno 5%

# Faktorji oblike

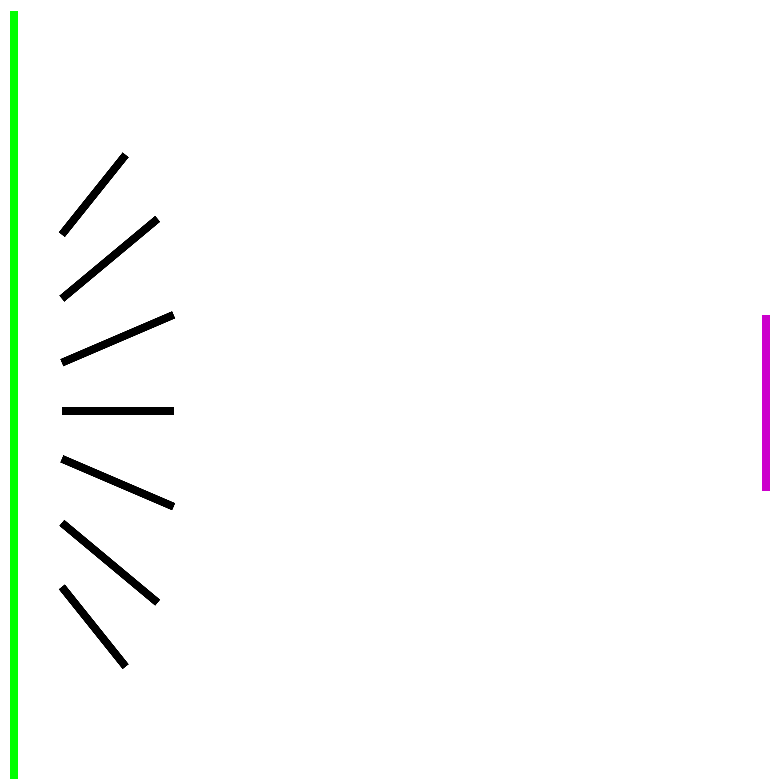
Nekaj primerov…



Faktor oblike:
približno 30%

# Faktorji oblike

Nekaj primerov…



Faktor oblike:
približno 2%

# Faktorji oblike

In diffuse environments, form factors obey a simple reciprocity relationship:

$$A_i F_{ij} = A_j F_{ji}$$

Which simplifies our equation:

$$B_i = E_i + \rho_i \sum B_j F_{ij}$$

Rearranging to:

$$B_i - \rho_i \sum B_j F_{ij} = E_i$$

# Faktorji oblike

So…light exchange between all patches becomes a matrix:

$$\begin{bmatrix} 1-\rho_1 F_{11} & -\rho_1 F_{12} & \cdots & -\rho_1 F_{1n} \\ -\rho_2 F_{21} & 1-\rho_2 F_{22} & \cdots & -\rho_2 F_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ -\rho_n F_{n1} & -\rho_n F_{n2} & \cdots & 1-\rho_n F_{nn} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_n \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_n \end{bmatrix}$$

*What do the various terms mean?*

# Faktorji oblike

$$
\begin{bmatrix}
1-\rho_1 F_{11} & -\rho_1 F_{12} & \dots & -\rho_1 F_{1n} & B_1 & E_1 \\
-\rho_2 F_{21} & 1-\rho_2 F_{22} & \dots & -\rho_2 F_{2n} & B_2 & E_2 \\
. & . & \dots & . & . & . \\
. & . & \dots & . & . & . \\
. & . & \dots & . & . & . \\
-p_n F_{n1} & -\rho_n F_{n2} & \dots & 1-\rho_n F_{nn} & B_n & E_n
\end{bmatrix}
\begin{bmatrix} \\ \end{bmatrix}
=
\begin{bmatrix} \\ \end{bmatrix}
$$

Note: $E_i$ values zero except at emitters

Note: $F_{ii}$ is zero for convex or planar patches

Note: sum of form factors in any row = 1 (*Why?*)

Note: *n* equations, *n* unknowns!

# Sevalna metoda

Now “just” need to solve the matrix!

- W&W: matrix is “diagonally dominant”
- Thus Guass-Siedel must converge (*what’s that?*)

End result: radiosities for all patches

Solve RGB radiosities separately, color each patch, and render!

Caveat: for rendering, we actually color vertices, not patches (see F&vD p 795)

# Sevalna metoda

Q: *How many form factors must be computed?*

A: O($n^2$)

Q: *What primarily limits the accuracy of the solution?*

A: The number of patches

# Faktorji oblike

Calculating form factors is hard

- Analytic form factor between two polygons in general case: open problem till last few years

Q: *So how might we go about it?*

Hint: Clearly form factors are related to visibility: how much of patch *j* can patch *i* "see"?

# Faktorji oblike: polkocke

*Hemicube algorithm*: Think Z-buffer

- Render the model onto a *hemicube* as seen from the center of patch i
- Store item IDs instead of color
- Use Z-buffer to resolve visibility
- See W&W p 278

Q: *Why hemicube, not hemisphere?*

# Faktorji oblike: polkocke

Advantages of hemicubes

- Solves shape, size, orientation, and occlusion problems in one framework
- Can use hardware Z-buffers to speed up form factor determination (*How?*)

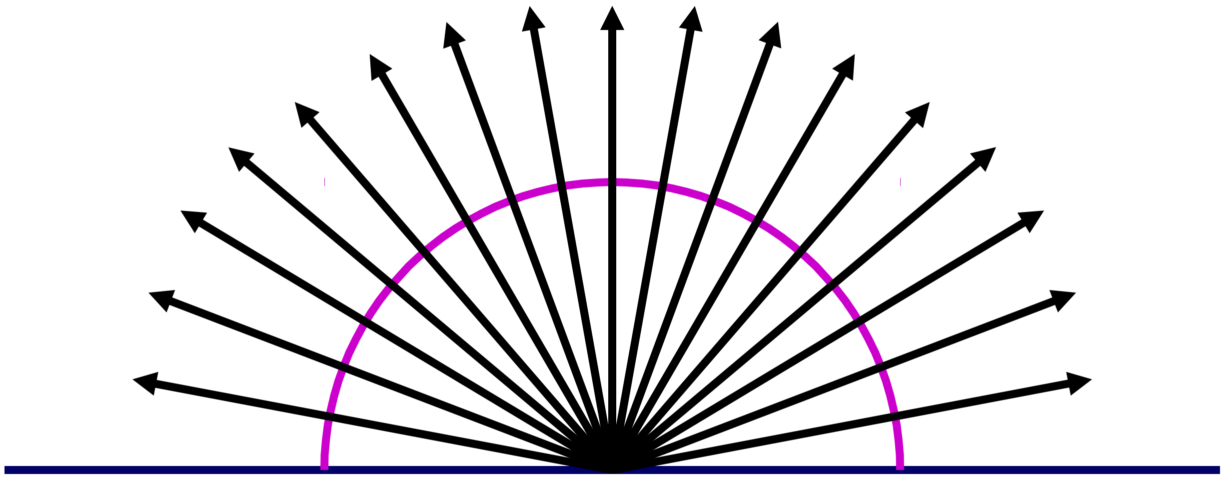
# Faktorji oblike: polkocke

Q: *What are some disadvantages of hemicubes?*

- Aliasing!  Low resolution buffer can't capture actual polygon contributions very exactly
  - Causes "banding" near lights (plate 41)
- Actual form factor is over area of patch; hemicube samples visibility at only center point on patch (*So?*)

# Faktorji oblike: Zlivanje žarkov (Ray Casting)

- Idea: shoot rays from center of patch in hemispherical pattern



# Zlivanje žarkov (Ray Casting)

Advantages:

- Hemisphere better approximation than hemicube
  - More even sampling reduces aliasing
- Don’t need to keep item buffer
- Slightly simpler to calculate coverage

# Zlivanje žarkov (Ray Casting)

Disadvantages:

- Regular sampling still invites aliasing
- Visibility at patch center still isn't quite the same as form factor
- Ray tracing is generally slower than Z-buffer-like hemicube algorithms
  - Depends on scene, though
  - Q: *What kind of scene might ray tracing actually be faster on?*

# Faktorji oblike

Source-to-vertex form factors

- Calculating form factors at the patch vertices helps address some problems:

```
for every patch vertex
    for every source patch
        sample source evenly with rays
        visibility = % rays that hit
```

- Q: *What are the problems with this approach?*

# Faktorji oblike

Summary of form factor computation

- Analytical:
  - Expensive or impossible (in general case)
- Hemicube
  - Fast, especially using graphics hardware
  - Not very accurate; aliasing problems
- Ray casting
  - Conceptually cleaner than hemicube
  - Usually slower; aliasing still possible

# Sistem sevalne metode

- Razveljavimo žarečo verzijo enačbe upodabljanja

$$L(\mathbf{x},\omega) = L_e(\mathbf{x},\omega) + \int f_r(\mathbf{x},\omega,\omega')G(\mathbf{x},\mathbf{x}')L(\mathbf{x}',\omega')dA'$$

$$G(\mathbf{x},\mathbf{x}') = \frac{\cos\theta\cos\theta'}{\|\mathbf{x}-\mathbf{x}'\|^2}V(\mathbf{x},\mathbf{x}')$$

- Iztekajoča verzija enačbe upodabljanja

$$\Phi_o = \Phi_e + \int f_r(\mathbf{x},\omega,\omega')\Phi_i \frac{\cos\theta\cos\theta'}{r^2}dA'$$

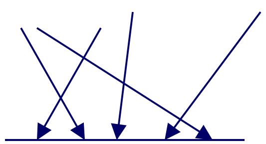
- Iztekanje = Sevanje X Površina

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

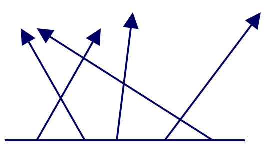
- Difuzni odboj je konstanten

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

- $F_{ij}$ je *faktor snovi*



$\Phi_i$



$\Phi_o$

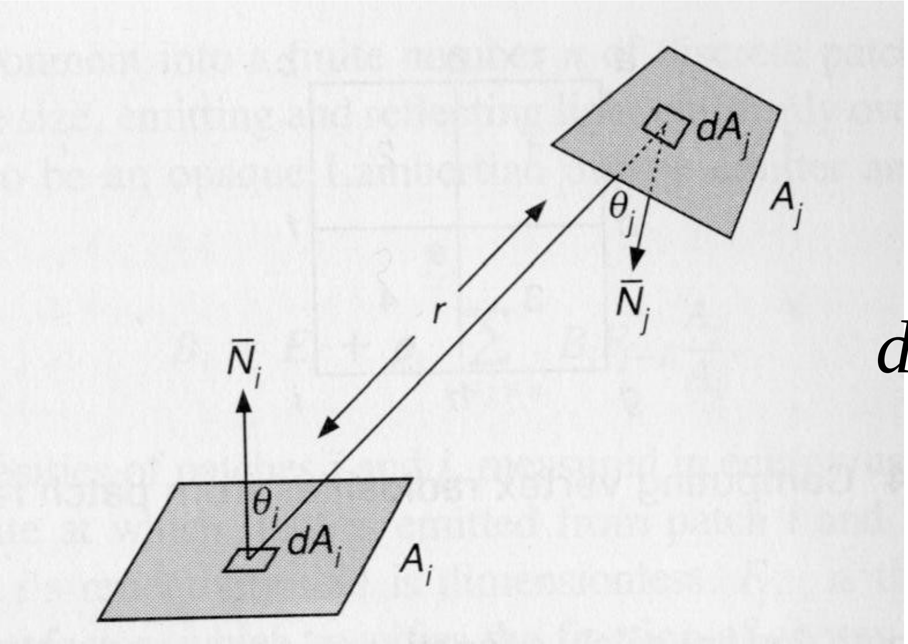
# Sevalna metoda (radiosity)

$$B_i = \varepsilon_i + \rho_i \sum_{1 \le j \le n} B_j F_{j,i} \frac{A_j}{A_i}$$

- Sevanje elementa ‘i’ je
- $\varepsilon_i$ je delež svetlobe, ki se jo odda
- $\rho_i$ je odbojnost i-tega elementa
- $F_{j,i}$ je faktor snovi
  - ulomek elementa j, ki doseže element i je določen z orientacijo obeh elementov in z ovirami
- Uporabi $A_j/A_i$ (površina elementa j / površina elementa i) da določiš enote za vso svetlobo oddano od j glede na prejeto na enoto površine i-ja

# Faktorji oblike

- Izračunaj nxn matriko faktorjev snovi da shraniš odnose med sevanjem posameznega svetlobnega elementa in vsemi ostalimi



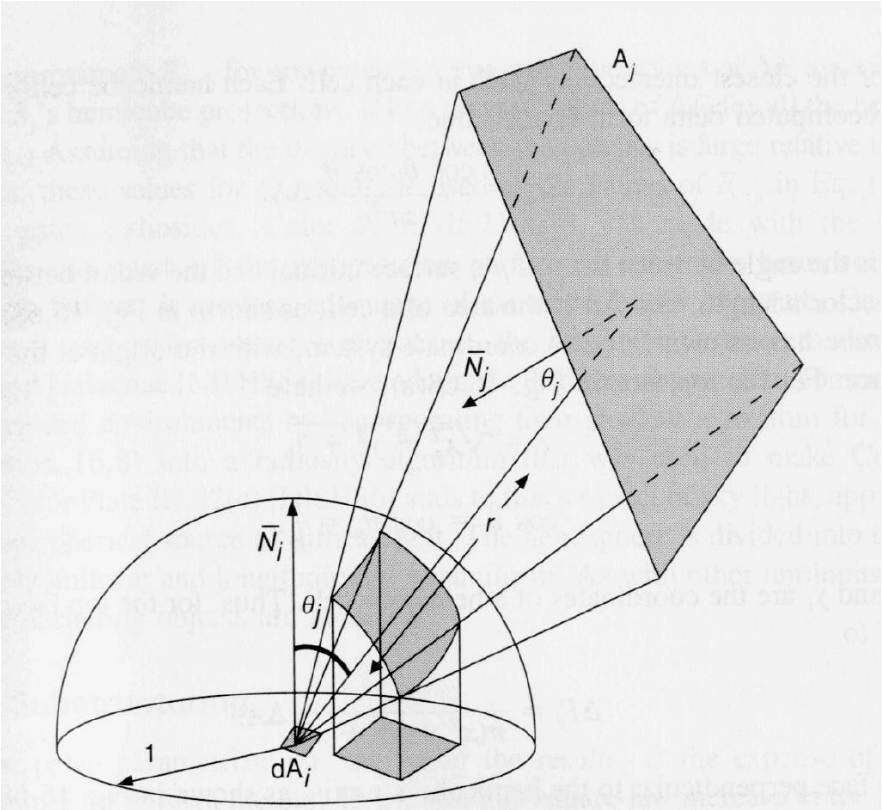

$$dF_{di,dj} = \frac{\cos\theta_i \cos\theta_j}{\pi r^2} H_{ij} dA_j$$

# Faktor oblike – projekcije krogle

Modeliranje faktorja oblike s projekcijami krogle

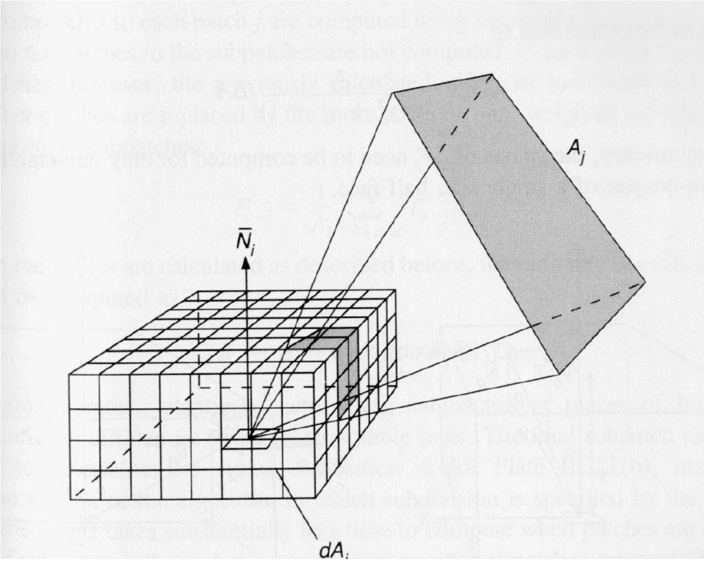
- poligon $A_j$ projeciramo na enoto poloble s središčem (in tangentno na) $A_i$
  - Prispevek $\cos\theta_j / r^2$
- Projeciramo to projekcijo ploskev poloble
  - Prispevek $\cos\theta_i$
- To površino delimo s kroga
  - Prispevek $\pi$

$$dF_{di,dj} = \frac{\cos\theta_i \cos\theta_j}{\pi r^2} H_{ij} dA_j$$



# Faktor oblike – polkocka

- Polkocka omogoča hitrejše računanje
  - Analitična rešitev polkrogle je draga
  - Uporabimo pravokotni približek, polkocko
  - Kosinusi za vrh in stranice se poenostavijo
  - Dimenzije 50 – 200 kvadratov so dobre



# Reševanje za vse elemente

En element definiran z: $$B_i = \varepsilon_i + \rho_i \sum_{1 \le j \le n} B_j F_{j,i} \frac{A_j}{A_i}$$

Simetrija: $A_iF_{i,j} = A_jF_{j,i}$

Zato: $$B_i = \varepsilon_i + \rho_i \sum_{1 \le j \le n} B_j F_{i,j}$$

In: $$B_i - \rho_i \sum_{1 \le j \le n} B_j F_{i,j} = \varepsilon_i$$

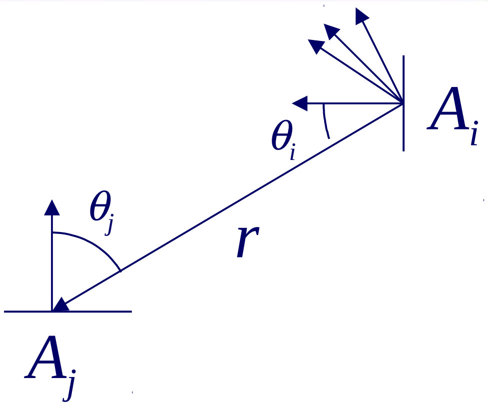
Uporabi algebro matrik za izračun $B_i$-jev:

$$\begin{bmatrix} 1 - \rho_1 F_{1-1} & -\rho_1 F_{1-2} & \cdots & -\rho_1 F_{1-n} \\ -\rho_2 F_{2-1} & 1 - \rho_2 F_{2-2} & \cdots & -\rho_2 F_{2-n} \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ -\rho_n F_{n-1} & -\rho_n F_{n-2} & \cdots & 1 - \rho_n F_{n-n} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ \cdot \\ \cdot \\ \cdot \\ B_n \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ \cdot \\ \cdot \\ \cdot \\ E_n \end{bmatrix}.$$

# Sevalna metoda (radiosity)

- Sevanje je računsko kompleksno
  - Naredi doktorsko disertacijo o izboljšavi...
- Oddana svetloba in točka pogleda se lahko spremenita
- Koti svetlobe in položaji objektov se ne morejo spremeniti
  - Računanje faktorjev snovi je drago
- Zrcalni odboji se ne modelirajo

# Faktor oblike




$F_{dAidAj}$ – delež skupne jakosti svetlobe, ki zapusti difuzno površino elementa *i* (imenovalec) in doseže difuzno površino elementa *j* (števec)

$$F_{dAidAj} = \frac{I_i \cos\theta_i \cos\theta_j dA_i dA_j}{I_i dA_i \pi r^2} = \frac{\cos\theta_i \cos\theta_j dA_j}{\pi r^2}$$

$F_{dAiAj}$ – delež skupne jakosti svetlobe, ki zapusti difuzno povšino elementa *i* in v celoti doseže element *j*

$$F_{dAiAj} = \int_{A_j} \frac{\cos\theta_i \cos\theta_j}{\pi r^2} dA_j$$

$F_{AiAj} = F_{ij}$ – delež skupne jakosti svetlobe, ki zapusti element *i* in doseže element *j*

$$F_{ij} = \frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{\cos\theta_i \cos\theta_j}{\pi r^2} dA_j dA_i$$

**Lepe lastnosti faktorjev oblike**

$F_{ii} = 0$ ko so ploskve ravne

$F_{ij} = 0$ če so ploskve zaprte

$A_i F_{ij} = A_j F_{ji}$

$F_{ij}$ je brez dimenzij

$$F_{ij} = \frac{1}{A_i} \int_{A_i} \int_{\Omega} \frac{\cos\theta_i}{\pi} d\omega_j dA_i$$

Ko $r^2 >> A_j$ …

$$F_{ij} \approx \int_{A_j} \frac{\cos\theta_i \cos\theta_j}{\pi r^2} dA_j$$

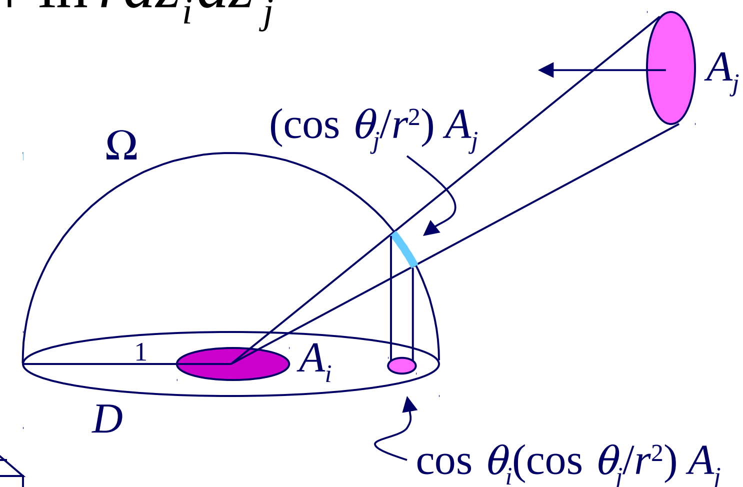
# Računanje faktorjev oblike

## Stokesov teorem [Lambert 1760, Goral *et al.* S84]

$$F_{ij} = \frac{1}{2\pi A_i} \oint_{\partial A_i} \oint_{\partial A_j} \ln r\,dx_i dx_j + \ln r\,dy_i dy_j + \ln r\,dz_i dz_j$$

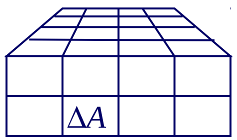
## Nusseltov analog

$$F_{ij} = \mathrm{proj}_D(\mathrm{proj}_W(A_j))/\mathrm{Area}(D)$$



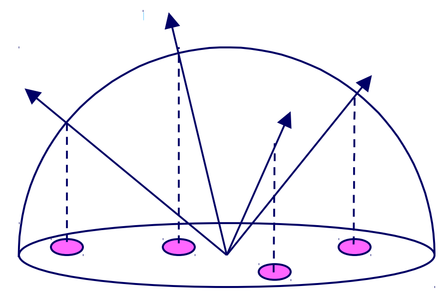

## Polkocka

$$\Delta F_{dAiAj} = \frac{\cos\phi_i \cos\phi_j}{\pi r^2}\Delta A$$




## Monte-Carlo metanje žarkov

- Uniformni vzorčni disk
- $F_{ij}$ = (# žarkov ki zadanejo $A_j$) / (# vseh žarkov)

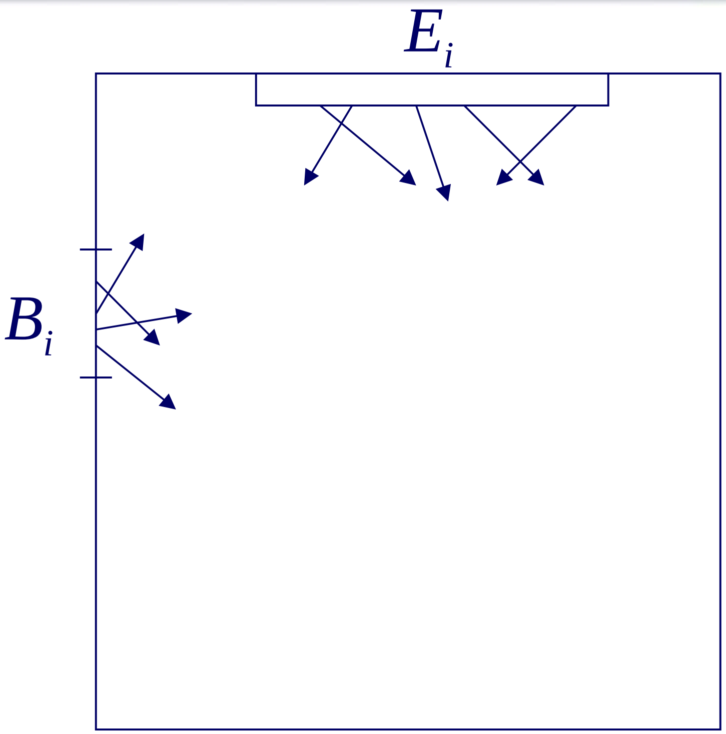


# Matrično sevanje

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

$$A_i F_{ij} = A_j F_{ji}$$

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




$$B_i - \rho_i \sum_{j=1}^{n} F_{ij} B_j = E_i$$

$$R = \begin{bmatrix} \rho_1 & & & \\ & \rho_2 & & \\ & & \ddots & \\ & & & \rho_n \end{bmatrix}$$

$$(I - RF)B = E$$

$$\begin{bmatrix} 1-\rho_1 F_{11} & -\rho_1 F_{12} & \cdots & -\rho_1 F_{1n} \\ -\rho_2 F_{21} & 1-\rho_2 F_{22} & \cdots & -\rho_2 F_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ -\rho_n F_{n1} & -\rho_n F_{n2} & \cdots & 1-\rho_n F_{nn} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_n \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_n \end{bmatrix}$$

# Zbiranje podatkov

$$\begin{bmatrix} 1 & -\rho_1 F_{12} & \cdots & -\rho_1 F_{1n} \\ -\rho_2 F_{21} & 1 & \cdots & -\rho_2 F_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ -\rho_n F_{n1} & -\rho_n F_{n2} & \cdots & 1 \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_n \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_n \end{bmatrix}$$

Rešuje se kot linearni sistem $A\mathbf{x} = \mathbf{b}$

$$MB = E$$

Jacobi

- Sevanje = Oddajanje + Odboj ostalih sevanj

$$B_i^{(k+1)} = E_i - \Sigma_{j\neq i}\, M_{ij}\, B_j^{(k)}$$

Gauss-Siedel

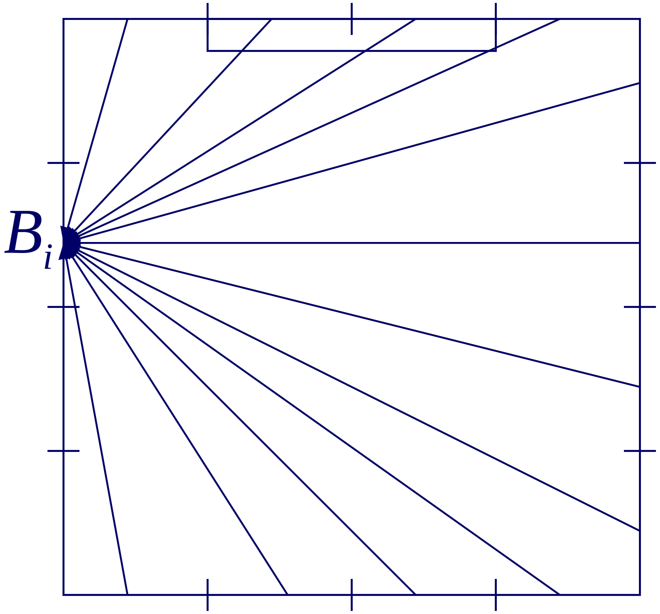
- računa mestoma (tu pa tam)

$$B_i = E_i - \Sigma_{j\neq i}\, M_{ij}\, B_j$$

pretirana omilitev

- Gauss-Siedel je preveč konzervativen

$$B_i^{(k+1)} = 110\%\, B_j^{(k+1)} - 10\%\, B_j^{(k)}$$



# Izris

Progresivno izboljšanje

Porazdeli dodatno sevanje $\Delta B_i$ med ostale elemente *j*

$$B_j^{(k+1)} = B_j^{(k)} + \Sigma \rho_j F_{ji} \, \Delta B_i$$

Dodatno “neizrisano” sevanje je tisto, kar smo dobili v zadnji iteraciji

$$\Delta B_i = B_j^{(k)} - B_j^{(k-1)}$$

Energija se začne pri oddajniku

Porazdeljuj “progresivno” skozi prizor

Lahko upoštevamo ambient, ko prikazujemo prizor in v njem delamo zamenjave tekom napredovanja progresivnega sevanja