# Iluminação e FotoRealismo:
## Radiosidade

Luís Paulo Peixoto dos Santos

http://gec.di.uminho.pt/mcgav/ifr

# Premissas

- Todas as interacções da luz com os objectos são difusas

$$L(x \to \Theta) = L(x), \forall_{\Theta \in \Omega_s}$$

- Expressa em termos de radiosidade (W/m$^2$)

$$B(x) = \int_{\Omega_s} L(x)\cos(\vec{N}_x, \Theta)\partial_{\omega_\Theta} = L(x)\int_{\Omega_s} \cos(N_x, \Theta)\partial_{\omega_\Theta} = \pi L(x)$$

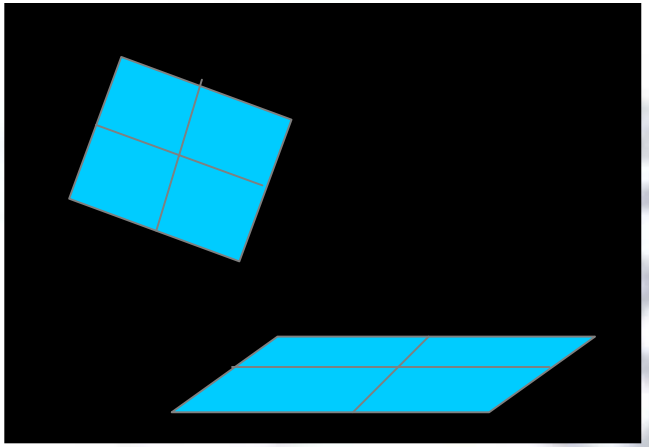
- A BRDF $f_r(x,\Theta\leftrightarrow\Psi)$ é independente das direcções

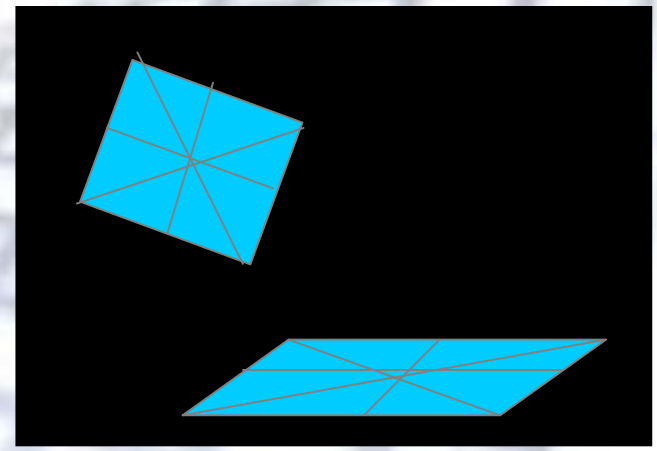
$$f_r(x, \Theta \leftrightarrow \Psi) = f_r(x) = \frac{\rho_d(x)}{\pi}$$

onde $\rho_d(x)$ é o coeficiente de reflexão difuso (dependente de $\lambda$)

# Premissas

- A geometria é subdividida em *patches* Pi (geralmente poligonais)





- A qualidade da solução final depende da granularidade da subdivisão em *patches*
- Esta pode ser decidida antes da execução (e.g. radiosidade hierárquica) ou refinada em tempo de execução

# Premissas

- O coeficiente de reflexão difuso é constante para cada *patch*

$$\rho(x) = \rho_i, \forall_{x \in P_i}$$

- A radiosidade é constante para cada *patch*

$$B_i = \frac{1}{A_i} \int_{A_i} B(x) \partial_{A_i}$$

# Formulação matemática

- A equação de rendering  cada ponto *x* de um *patch* i é :

$$B_i(x) = B_{ei}(x) + \int_A \frac{\rho(x)}{\pi} B(y) V(x,y) G(x,y) \partial_{A_y}$$

- Integrando para a área de $P_i$ ($A_i$) e assumindo que B e ρ são constantes para $P_i$

$$B_i A_i = B_{ei} A_i + \rho_d \frac{1}{A_i} \int_{A_i} \int_A B(y) \frac{V(x,y) G(x,y)}{\pi} \partial_{A_y} \partial_{A_i}$$

- Convertendo o integral da área de todos os outros *patches* num somatório das várias áreas dos *patches*  $P_j$

$$B_i A_i = B_{ei} A_i + \rho_d \sum_j \frac{1}{A_i} \int_{A_i} \int_{A_j} B(y) \frac{V(x,y) G(x,y)}{\pi} \partial_{A_j} \partial_{A_i}$$

# Formulação matemática

- Assumindo que a radiosidade é constante ao longo de cada *patch* $P_j$

$$B_i A_i = B_{ei} A_i + \rho_d \sum_j B_j A_j \frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{V(x,y) G(x,y)}{\pi} \partial_{A_j} \partial_{A_i}$$

- Dividindo dos 2 lados por $A_i$

$$B_i = B_{ei} + \rho_d \sum_j B_j \frac{A_j}{A_i} \int_{A_i} \int_{A_j} \frac{1}{A_i} \frac{V(x,y) G(x,y)}{\pi} \partial_{A_j} \partial_{A_i}$$

Obtemos a equação da radiosidade

$$B_i = B_{ei} + \rho_d \sum_j B_j \frac{A_j}{A_i} F_{ji}$$

# Formulação matemática

$$B_i = B_{ei} + \rho_d \sum_j B_j \frac{A_j}{A_i} F_{ji}$$

$F_{ji}$ é designado por form-factor e indica a fracção de radiosidade emitida por $P_j$ que incide em $P_i$. Os *form factors* exibem a seguinte propriedade:

$$F_{ji} A_j = F_{ij} A_i \Leftrightarrow F_{ij} = \frac{F_{ji} A_j}{A_i}$$

logo:

$$B_i = B_{ei} + \rho_d \sum_j B_j F_{ij}$$

# Form-Factors - complexidade

$$F_{ij} = \frac{1}{A_i}\int_{A_i}\int_{A_j}\frac{V(x_i, y_j)G(x_i, y_j)}{\pi}\partial_{Aj}\partial_{A_i}$$

$$F_{ij} = \frac{1}{A_i}\int_{A_i}\int_{A_j}\frac{V(x_i, y_j)\cos(\theta_i)\cos(\theta_j)}{\pi r^2}\partial_{Aj}\partial_{A_i}$$

O *form factor* é um integral duplo de difícil resolução analítica.

Indica a quantidade de radiosidade emitida pelo *patch* j que incide no *patch* i.

O seu valor depende da geometria: visibilidade mútua, distância, orientação e área dos *patches*.

O cálculo dos *form-factors* tem uma complexidade $O(n^2)$, podendo rapidamente tornar-se a componente mais dispendiosa do algoritmo.

# Sistema de equações

$$B_i = B_{ei} + \rho_d \sum_j B_j F_{ij}$$

Num ambiente fechado existe uma solução para cada $B_i$ que pode ser obtida resolvendo um sistema de *n* equações com $n^2$ form-factors

$$\begin{bmatrix} 1-\rho_1 F_{11} & -\rho_1 F_{12} & \cdot & \cdot & -\rho_1 F_{1n} \\ -\rho_2 F_{21} & 1-\rho_2 F_{22} & \cdot & \cdot & -\rho_2 F_{2n} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ -\rho_n F_{n1} & -\rho_n F_{n2} & \cdot & \cdot & 1-\rho_n F_{nn} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ \cdot \\ \cdot \\ B_n \end{bmatrix} = \begin{bmatrix} B_{e1} \\ B_{e2} \\ \cdot \\ \cdot \\ B_{en} \end{bmatrix}$$

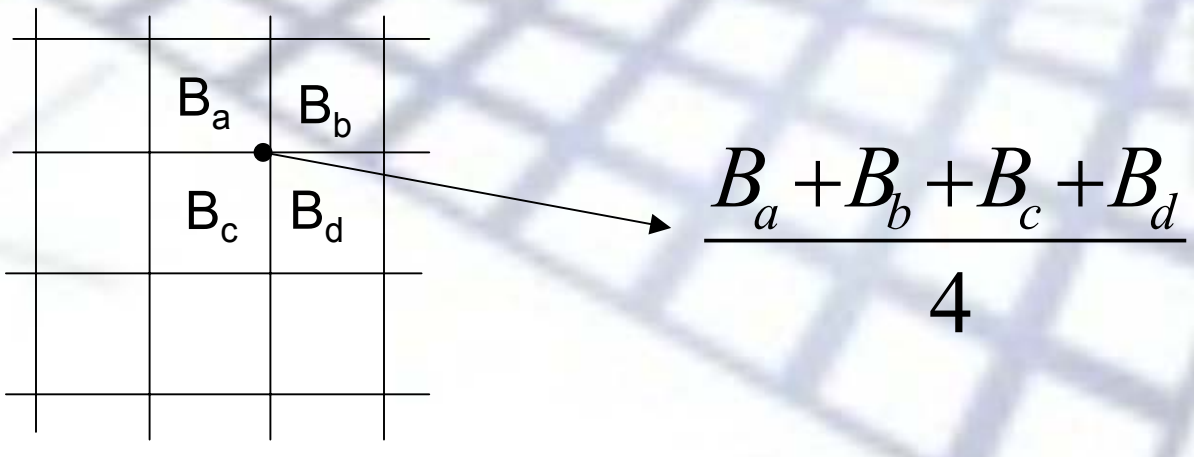
# Propriedades do sistema de equações

- Os valores $B_{ei}$ são diferentes de 0 apenas para os *patches* correspondentes a fontes de luz
- $F_{ii}$ é igual a 0 para *patches* planares ou convexos.
  O *patch* não emite radiosidade sobre si mesmo.
- Em ambientes complexos um grande número de $F_{ij}$ serão 0 pois os *patches* $P_i$ e $P_j$ não são mutuamente visíveis
- A matriz de *form-factors* é uma matriz dispersa que pode ser resolvida numericamente usando o método de Gauss-Seidel
- $B_i$, $B_{ei}$ e $\rho_i$ são dependentes do comprimento de onda, implicando a resolução de 3 sistemas.
- Os form-factors num meio participativo também são dependentes do comprimento de onda. Este aspecto é geralmente ignorado devido à complexidade do cálculo.

# Rendering

- As radiosidades são calculados no espaço dos objectos.

- O *hardware* gráfico pode ser usado para interpolar a radiosidade ao longo de cada patch (Gouraud *shading*) desde que as radiosidades nos vértices sejam conhecidas. Sintetiza-se assimuma imagem a partir de **qualquer** ponto de vista.



# Resolução do sistema - *gathering*

- O método de Gauss-Seidel computa aproximações $B_i^t$ a partir da aproximação $B_i^{t-1}$. A resolução é feita linha a linha calculando um $B_i^t$ em cada iteração.
- $B_i^0 = B_{ei}$, sendo 0 para todos os *patches* não auto-emissores.

$$\begin{bmatrix} B_1^t \\ \cdot \\ B_i^t \\ \cdot \\ B_n^t \end{bmatrix} = \begin{bmatrix} B_{e1} \\ \cdot \\ B_{ei} \\ \cdot \\ B_{en} \end{bmatrix} + \begin{bmatrix} \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \rho_i F_{i1} & \rho_i F_{i2} & \cdot & \cdot & \rho_i F_{in} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} B_1^{t-1} \\ \cdot \\ B_i^{t-1} \\ \cdot \\ B_n^{t-1} \end{bmatrix}$$

- Implica o cálculo inicial de todos os $n^2$ *form-factors.*
- Cada *patch* Pi recolhe (*gathers*) radiosidade de todos os outros *patches*.
- Cada avaliação da matriz calcula apenas um $B_i^t$

# Radiosidade Progressiva

- A principal desvantagem do método anterior é que cada iteração aproxima a iluminação de apenas um *patch*
- Implica também o cálculo inicial de todos os $n^2$ *form-factors*
- O objectivo da radiosidade progressiva é conseguir rapidamente uma aproximação à radiosidade de todos os *patches*
- Esta solução é depois refinada com mais iterações
- O refinamento progressivo é conseguido reordenando a forma como a radiosidade é calculada

# Radiosidade Progressiva

- Em lugar de calcular a aproximação $B_i^t$ devida a todos os patches $P_j$ com radiosidade $B_j^{t-1}$ vamos calcular a radiosidade $B_j^t$ de todos os *patches* $P_j$ devida à radiosidade $B_i^{t-1}$ do *patch* $P_i$

$$B_j \text{ devida a } B_i = \rho_j B_i F_{ji} = \rho_j B_i F_{ij} \frac{A_i}{A_j}$$

- Em cada iteração podemos calcular a variação de radiosidade $\Delta B_j$ para todos os $P_j$ devida ao patch $P_i$.

- $B_j^{t+1} = B_j^t + \Delta B_j$

- Para cada iteração apenas é necessário calcular n form-factors $F_{ij}$ para o *patch* i cuja radiosidade $B_i^t$ está a ser propagada (*shooting*)

# Radiosidade progressiva

$$\begin{bmatrix} B_1^{t+1} \\ B_2^{t+1} \\ \cdot \\ \cdot \\ B_n^{t+1} \end{bmatrix} = \begin{bmatrix} B_1^{t} \\ B_2^{t} \\ \cdot \\ \cdot \\ B_n^{t} \end{bmatrix} + \begin{bmatrix} \cdot \\ \cdot \\ \overline{B_i^t} \\ \cdot \\ \cdot \end{bmatrix} \begin{bmatrix} \cdot & \cdot & \rho_1 F_{1i} & \cdot & \cdot \\ \cdot & \cdot & \rho_{21} F_{2i} & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \rho_n F_{ni} & \cdot & \cdot \end{bmatrix}$$

$$B_j^{t+1} = B_j^t + \underbrace{B_i^t \sum_j (\rho_j F_{ji})}_{\Delta B_j} = B_j^t + \underbrace{B_i^t \sum_j (\rho_j F_{ij} \frac{A_i}{A_j})}_{\Delta B_j}$$

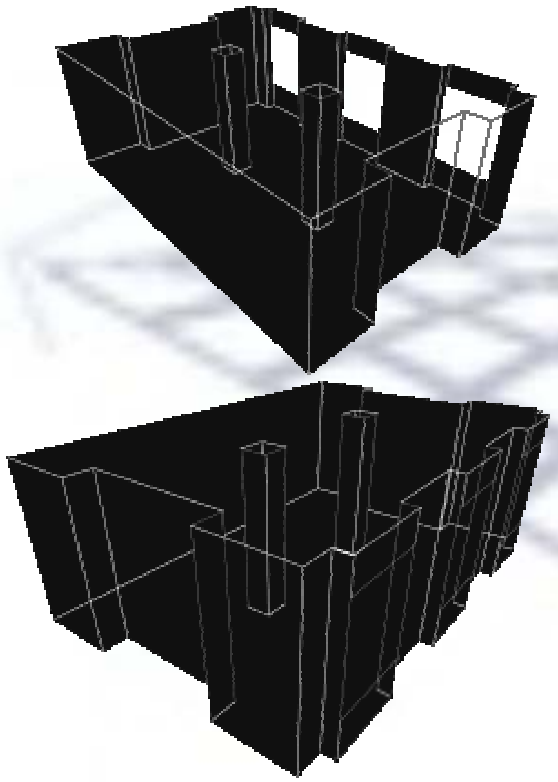
# Radiosidade Progressiva

- Inicialmente $B_i^0 = B_{ei}^0$. Qual deve ser a ordem pela qual se seleccionam os *patches* $P_i$ que vão propagar a sua radiosidade?

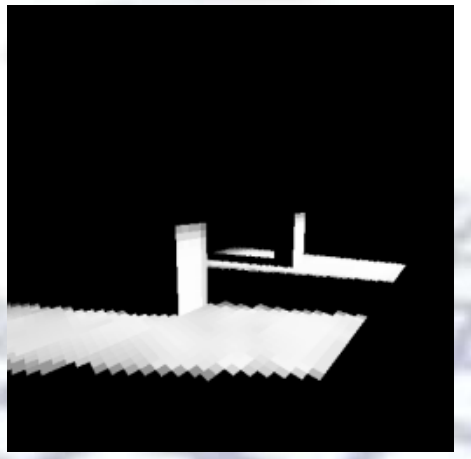
- Estes devem ser ordenados por $B_i^t$. O *patch* $P_i$ com maior radiosidade $B_i^t$ deve ser o primeiro a propagar a sua radiosidade para que a solução convirja o mais rapidamente possível.

- O processo para quando os $\Delta B_j$ forem menores que um determinado limite.

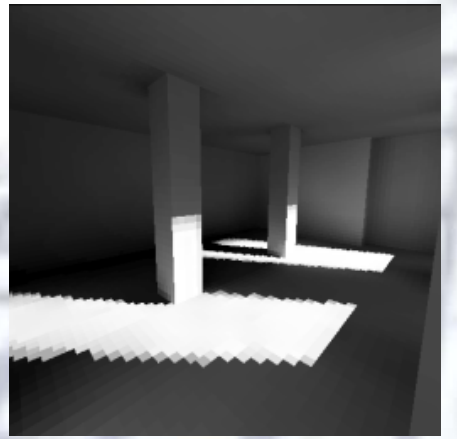
# Radiosidade Progressiva



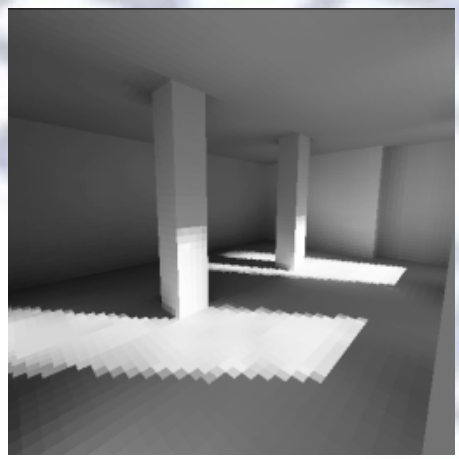
Geometria da cena



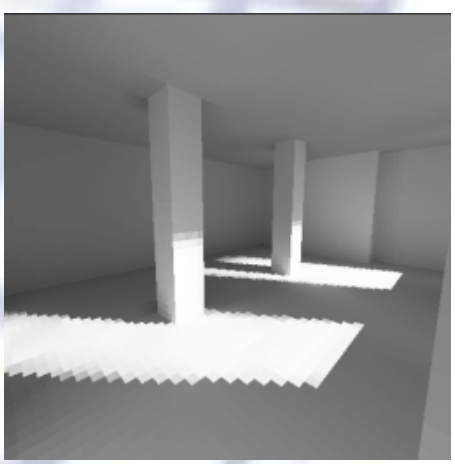
1 iteração



2 iterações



3 iterações



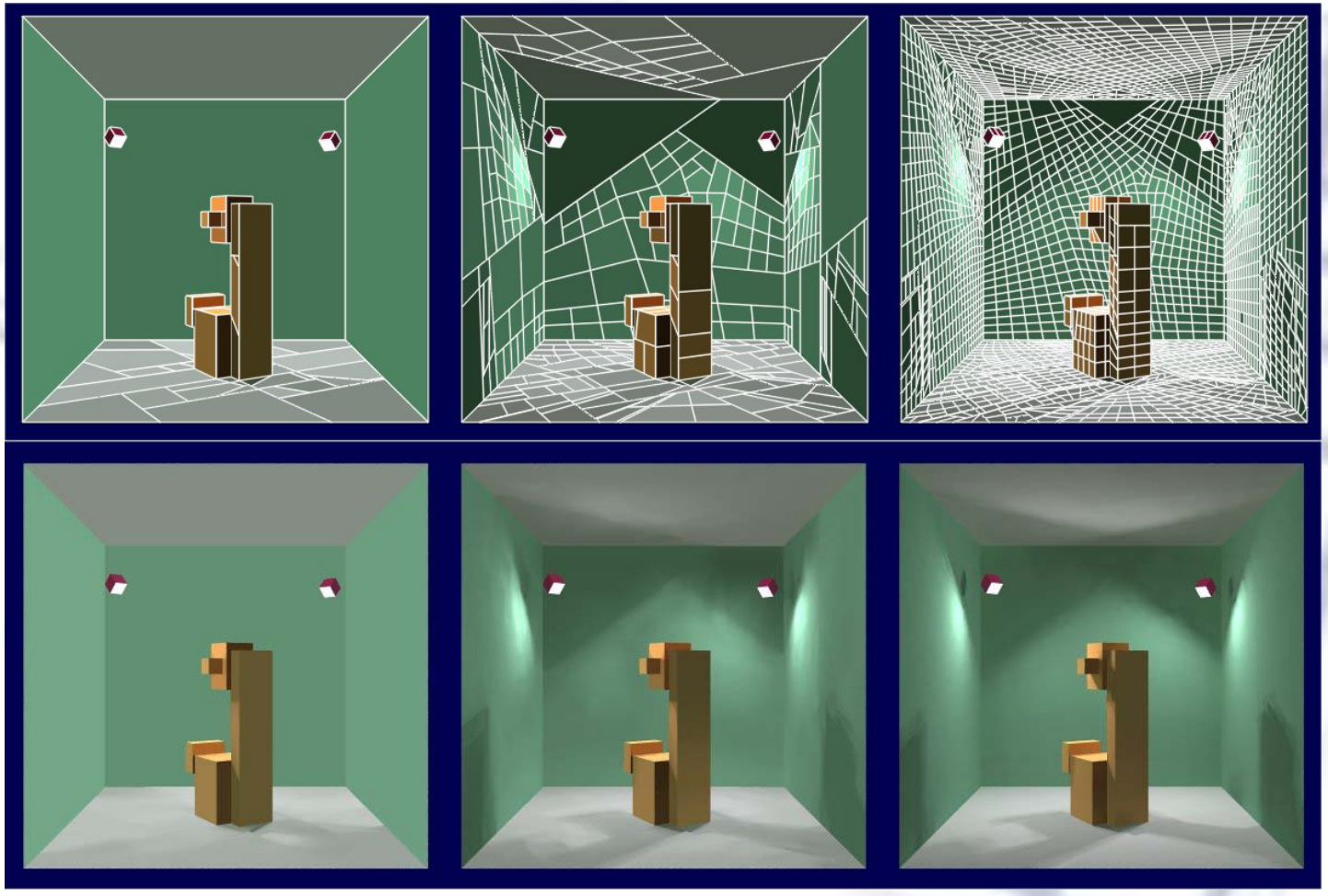
16 iterações

# Radiosidade Progressiva

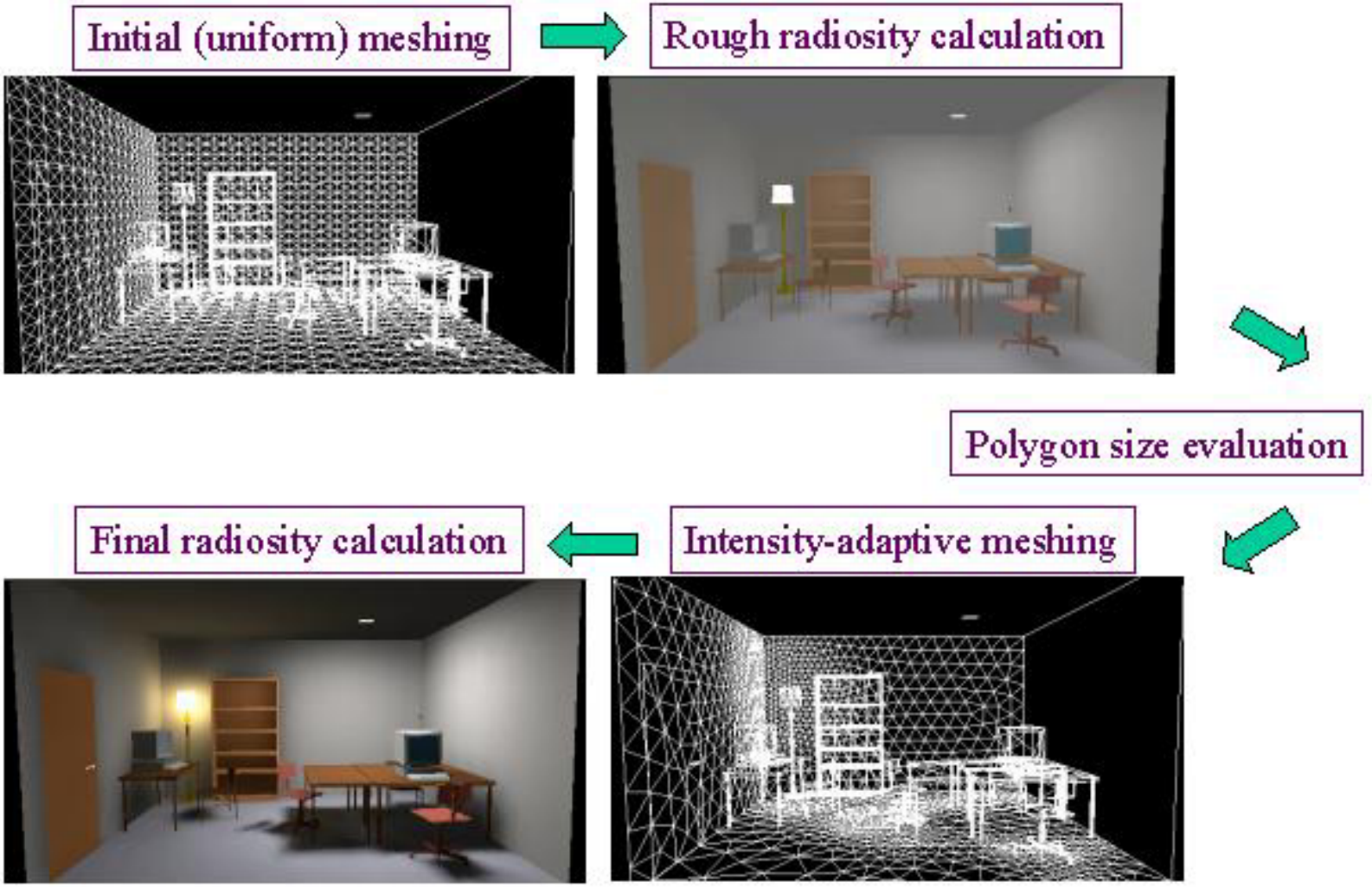
# Refinamento da grelha de *patches*

- A qualidade da solução depende da granularidade da grelha de *patches*
- Em zonas de alto gradiente (e.g. limites das sombras) o carácter discreto dos *patches* pode ser muito visível
- A grelha de *patches* pode ser definida de forma estática, antes da execução do algoritmo (e.g. radiosidade hierárquica [Hanrahan et al.91])
- ou refinada durante a execução do próprio algoritmo, e.g., *discontinuity meshing*

# Refinamento da grelha de *patches*



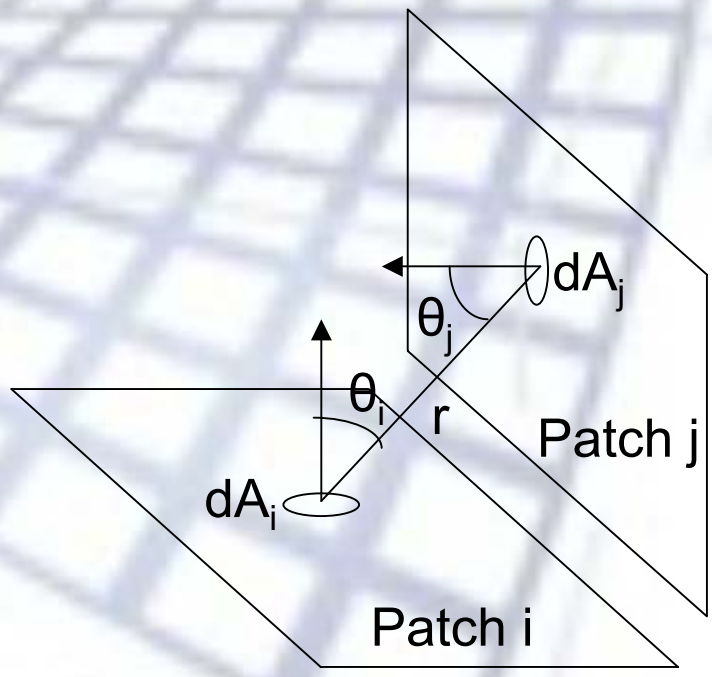
# Refinamento da grelha de *patches*



# Cálculo dos *form-factors*

$$F_{ij} = \frac{1}{A_i}\int_{A_i}\int_{A_j} V(x_i, y_j)\frac{\cos(\theta_i)\cos(\theta_j)}{\pi r^2}\partial_{Aj}\partial_{A_i}$$

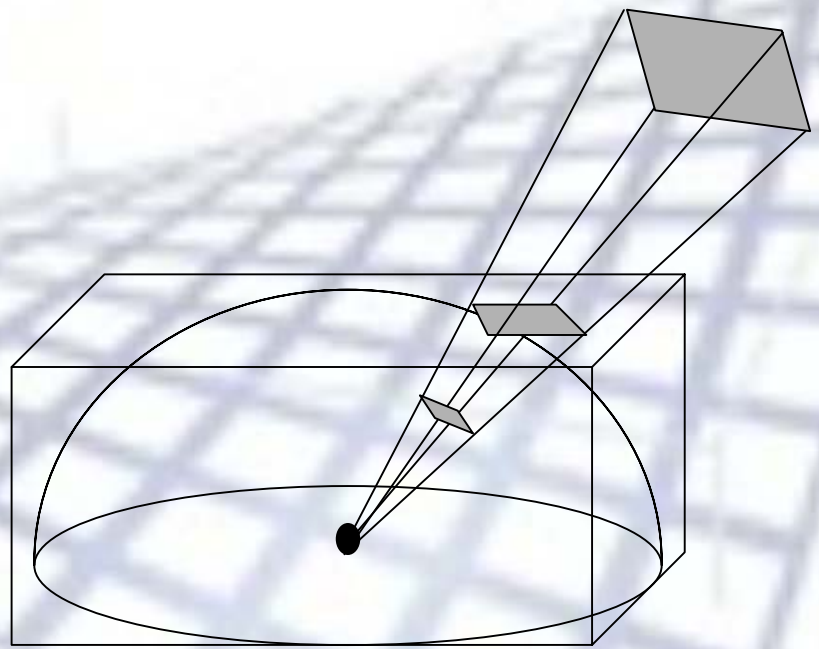
- Fij representa a fracção da radiosidade emitida por Pj que incide em Pi.
- O cálculo analítico dos *form-factors* é dispendioso, recorrendo-se a soluções geométricas.



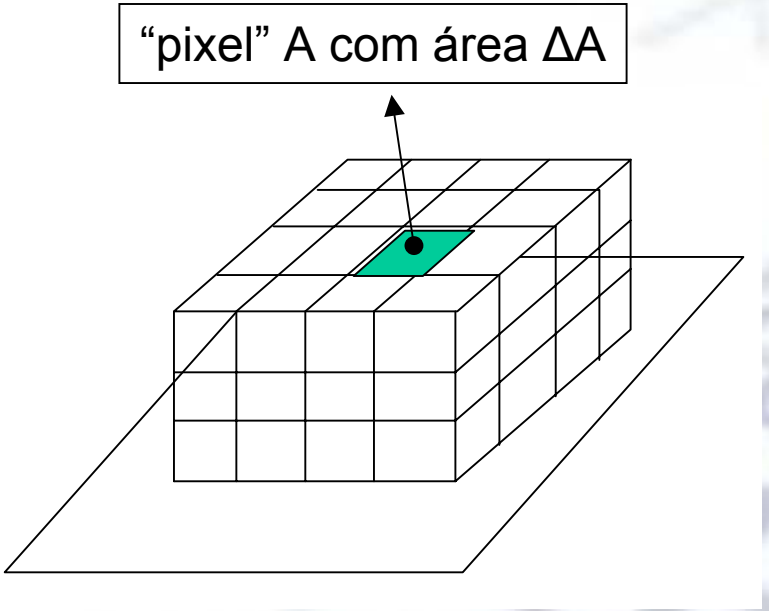
# *Form-Factors:* HemiCubo

**Analogia de Nusselt** - O polígono, a sua projecção no hemicubo e a sua projecção na hemisfera têm todos o mesmo *form-factor.*

Esta analogia permite usar projecções dos *patches* num hemicubo para calcular *form factors*

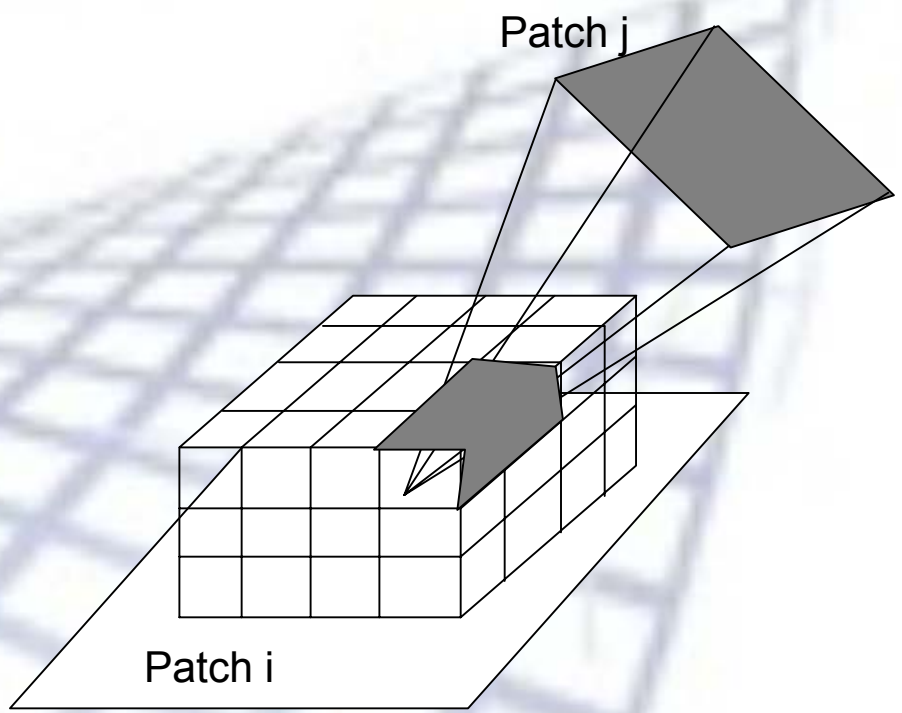


# *Form-Factors:* Hemicubo




A área ΔA é igual para todos os elementos do hemicubo. $\Delta F_q$ para cada elemento é dada por:

$$\Delta F_q = \frac{\cos\theta_i \cos\theta_j}{\pi r^2} \Delta A$$




$$F_{ij} = \sum_q \Delta F_q$$

Para todos os $F_q$ onde $P_j$ projecta

# *Form-Factors:* Hemicubo

- Para cada *patch* i todos os *patches* j devem ser projectados no hemicubo.
- Vários *patches* j projectam no mesmo “pixel” do hemicubo
- O algoritmo deve armazenar o ID do *patch* que projecta no “pixel” e a distância entre $P_i$ e $P_j$ (à semelhança de um Z-buffer)
- Para cada “pixel” apenas o *patch* $P_j$ mais próximo interessa, pois todos os outros estarão ocludidos por este.

# Form-Factors: Ray Tracing

- Em vez de usar um hemicubo pode ser usada uma hemisfera de raio 1 centrada em Pi, com a superfície dividida em áreas elementares.

- Cada área elementar corresponde aos ângulos (Δφ, Δθ) com

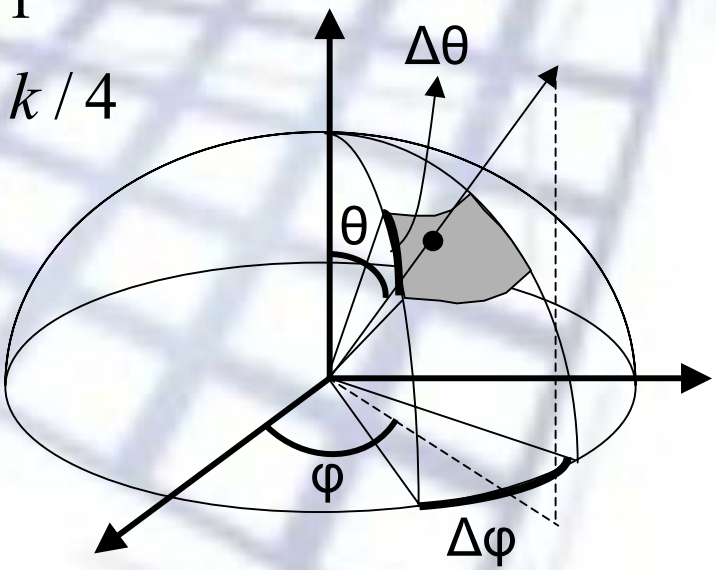
$$\Delta\varphi = \Delta\theta = 2\pi / k$$

$$\varphi \in [0, 2\pi], \varphi = l(\Delta\varphi), l = 0,1,\ldots, k-1$$

$$\theta \in [0, \pi/2], \theta = m(\Delta\theta), m = 0,1,\ldots, k/4$$

$$\Delta A = \Delta\varphi \Delta\theta \sin\theta$$

$$\Delta F_{lm} = \frac{2\pi}{k^2} \sin(2\theta)$$



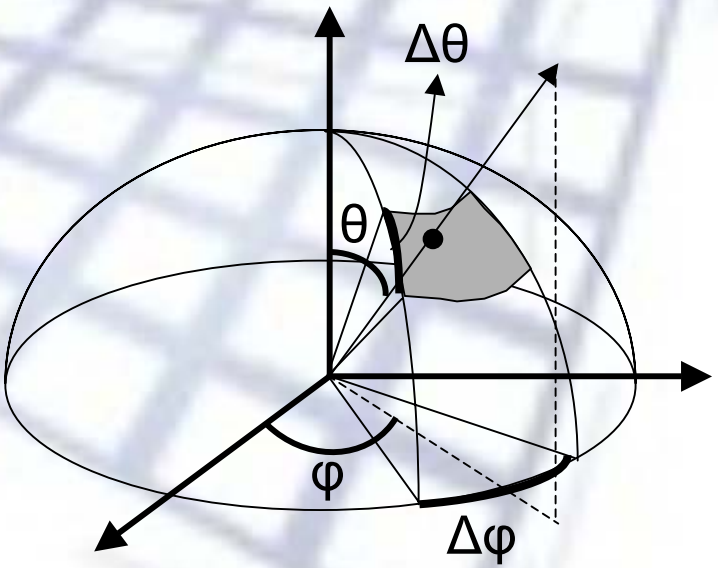
# *Form-Factors: Ray Tracing*

- Cada área elementar (I,m) dá origem a um raio que determina qual o patch $P_j$ visível ao longo daquela direcção

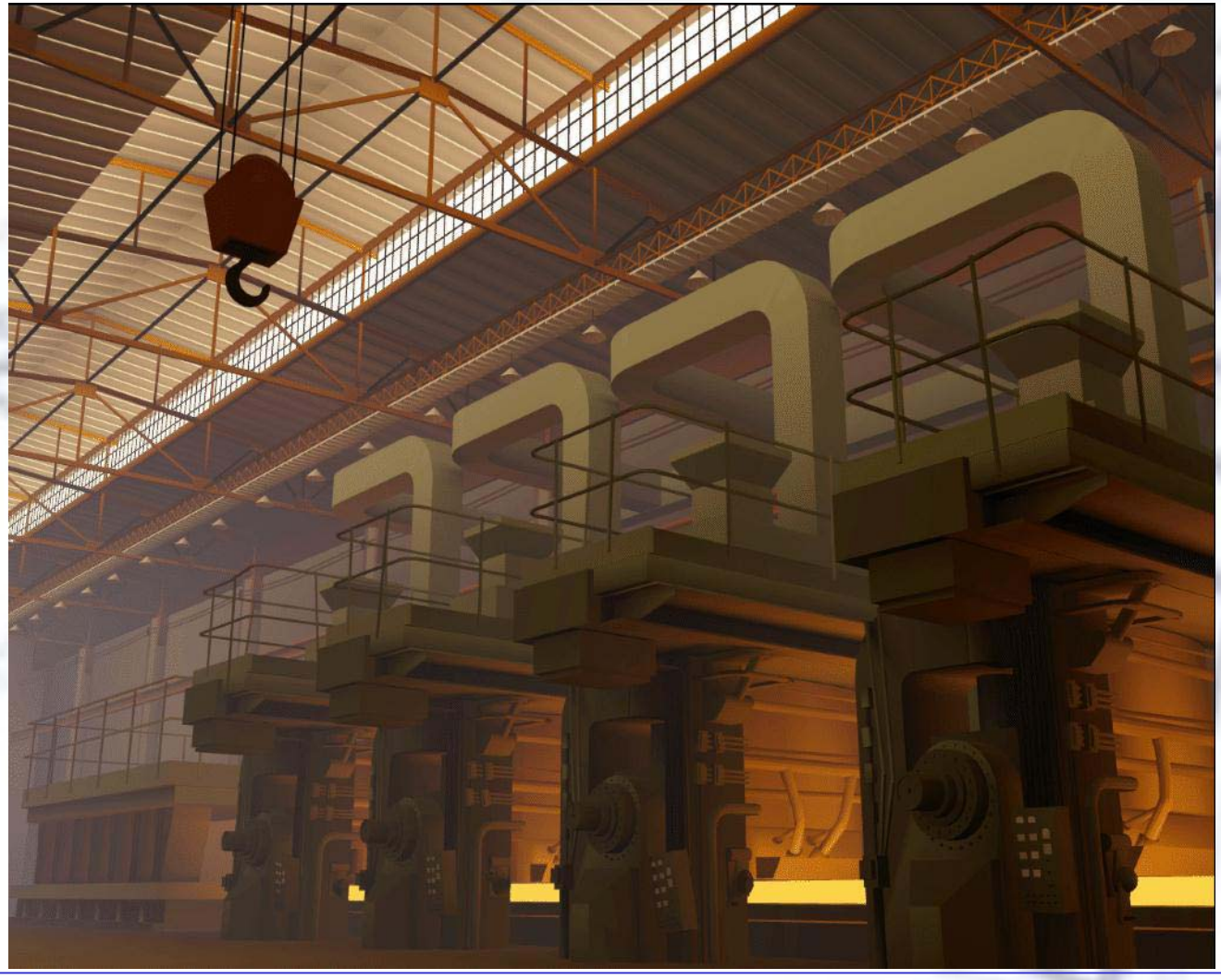
- O *form-factor* Fij é dado pela soma dos $\Delta F_{lm}$

$$F_{ij} = \sum_{q} \Delta F_{lm}$$

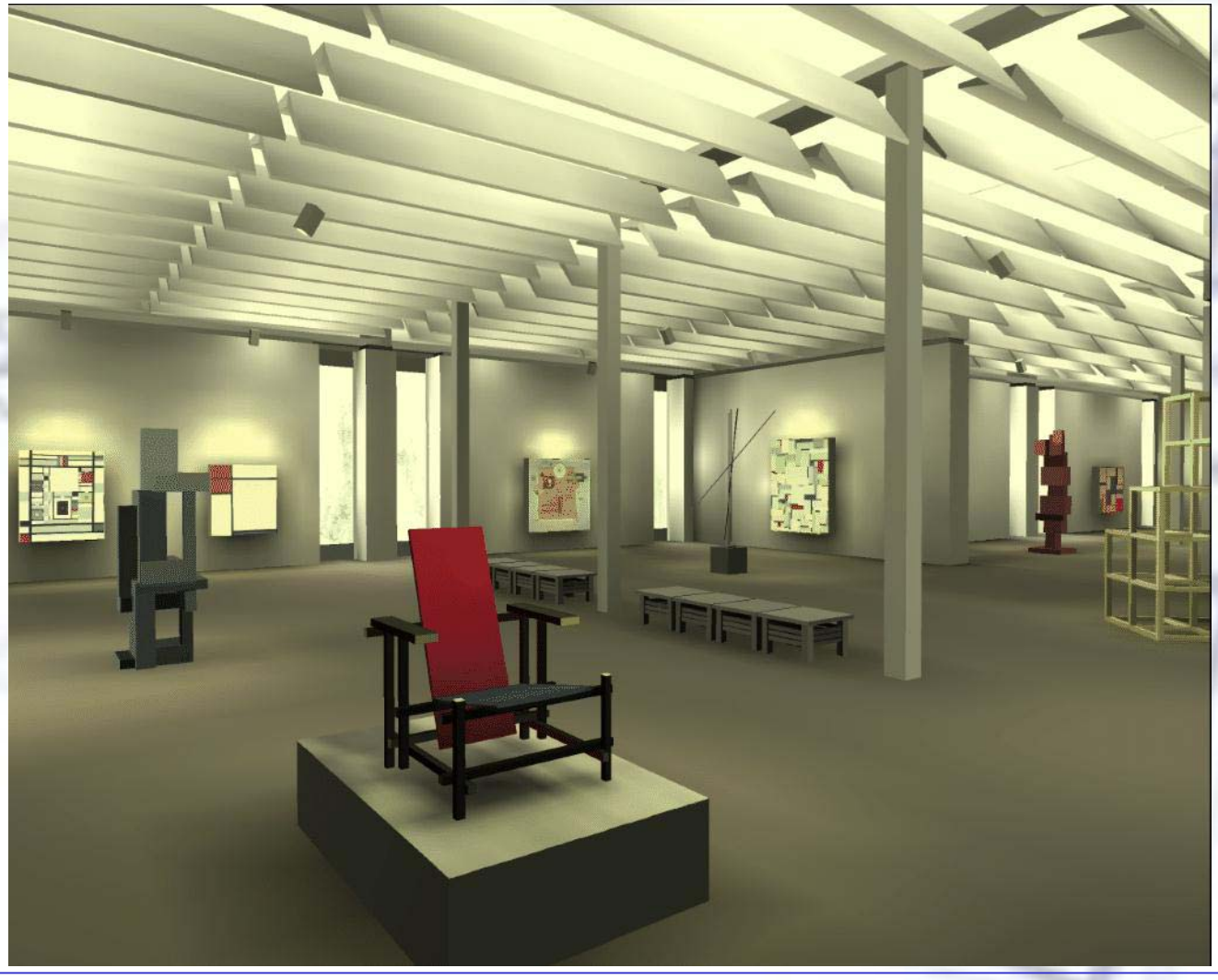
- A opção pelo algoritmo do hemicubo ou pelo *ray tracing* depende do desempenho dos algoritmos de projecção e *ray tracing* utilizados.



# Radiosidade: Imagens



# Radiosidade: Imagens



# Radiosidade: Imagens

'*color bleeding*' – a parede branca assume um tom avermelhado nas zonas fortemente iluminadas devido às interreflexões difusas.
Este efeito é de difícil modelação com *ray tracing*.