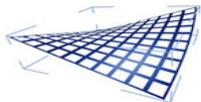

Iluminação e FotoRealismo: Radiosidade

Luís Paulo Peixoto dos Santos

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



Premissas

- Todas as interacções dos objectos com a luz são difusas

$$L(x \rightarrow \Theta) = L(x), \forall \Theta \in \Omega$$

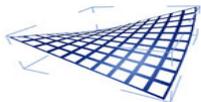
- Podemos então quantificar a radiosidade (W/m^2) em vez da radiância

$$B(x) = \int_{\Omega} L(x) \cos(\vec{N}_x, \Theta) \partial_{w_{\Theta}} = L(x) \int_{\Omega} \cos(\vec{N}_x, \Theta) \partial_{w_{\Theta}} = \pi L(x)$$

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

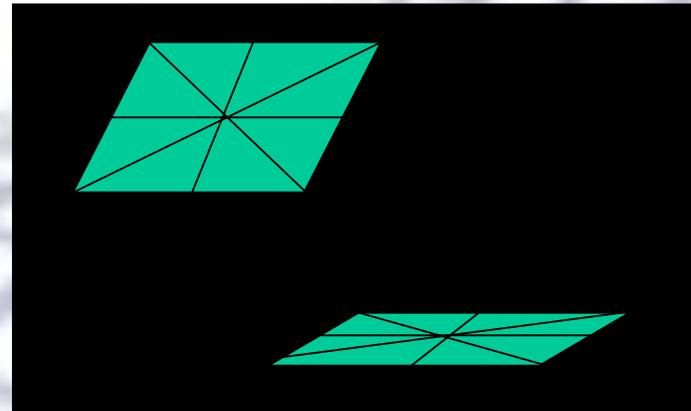
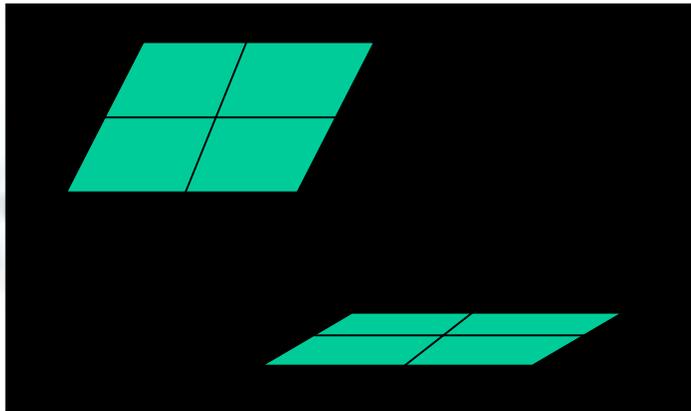
$$f_r(x, \Theta \leftrightarrow \Psi) = f_r(x) = \frac{\rho_d(x)}{\pi}$$

A BRDF $\rho_d(x)$ é o coeficiente de reflexão difuso (dependente de λ)

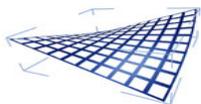


Premissas

- A geometria é subdividida em *patches*, geralmente poligonais



- A qualidade final depende da granularidade da subdivisão em *patches*
- Esta pode ser decidida antes da execução ou refinada durante a mesma



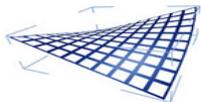
Premissas

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

$$\rho(x) = \rho_i, \forall x \in P_i$$

- A radiosidade é constante ao longo da área de cada *patch* P_i

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



Formulação Matemática

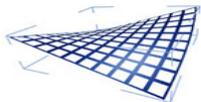
$$B_i = B_{e_i} + \rho_i \sum_j B_j \frac{A_j}{A_i} F_{ji}$$

- F_{ji} é designado por *form-factor* e indica a fracção da radiosidade B_j reflectida por P_j que chega a P_i . Este valor relaciona-se apenas com a geometria da cena e exhibe a seguinte propriedade:

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

logo:

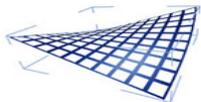
$$B_i = B_{e_i} + \rho_i \sum_j B_j F_{ij}$$



Form factors - Complexidade

$$F_{ij} = \frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{V(x_i, x_j) \cos(\theta_i) \cos(\theta_j)}{\pi r^2} \hat{\partial}_{A_j} \hat{\partial}_{A_i}$$

- O *form factor* é um integral duplo de difícil resolução analítica
- Indica a fracção de radiosidade emitida pelo *patch* i que incide no *patch* j
- O seu valor depende da geometria: visibilidade mútua, distância, área e orientação relativa dos *patches*
- O cálculo dos *form factors* tem uma complexidade $O(n^2)$, podendo rapidamente tornar-se a componente mais dispendiosa do algoritmo

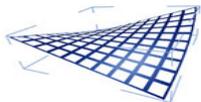


Sistema de Equações

$$B_i = B_{e_i} + \rho_i \sum_j B_j F_{ij}$$

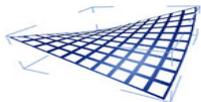
Num ambiente fechado existe uma solução para cada B_i , que pode ser obtida resolvendo um sistema de n equações com n^2 *form factors*:

$$\begin{bmatrix} 1 - \rho_1 F_{11} & -\rho_1 F_{12} & \cdot & \cdot & -\rho_1 F_{1n} \\ -\rho_2 F_{21} & 1 - \rho_2 F_{22} & \cdot & \cdot & -\rho_2 F_{2n} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ -\rho_n F_{n1} & -\rho_n F_{n2} & \cdot & \cdot & 1 - \rho_n F_{nn} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ \cdot \\ \cdot \\ B_n \end{bmatrix} = \begin{bmatrix} B_{e_1} \\ B_{e_2} \\ \cdot \\ \cdot \\ B_{e_n} \end{bmatrix}$$



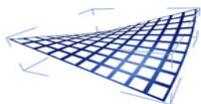
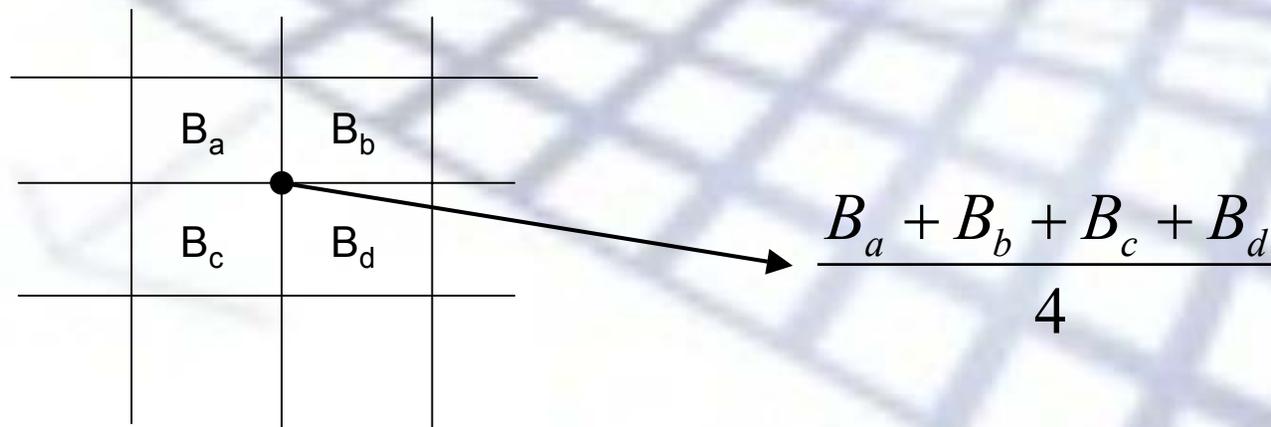
Propriedades do sistema de equações

- 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 radiação sobre si mesmo
- Em ambientes complexos a maioria dos F_{ij} são 0, pois os *patches* i e j não são mutuamente visíveis
- B_i , B_{ei} e ρ_i são dependentes do comprimento de onda; implica resolver 3 sistemas para o espaço RGB
- F_{ij} num meio participativo também depende do comprimento de onda; aspecto geralmente ignorado devido à complexidade dos cálculos



Rendering

- As radiosidades são calculadas 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 dos vértices sejam conhecidas. Sintetiza-se assim uma imagem a partir de **qualquer** ponto de vista.

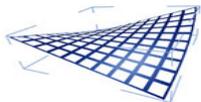


Resolução do sistema - *gathering*

- 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.
- $B_i^0 = B_{e_i}$, sendo 0 para todos os *patches* não auto-emissores

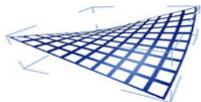
$$\begin{bmatrix} B_1^t \\ \cdot \\ \overline{B_i^t} \\ \cdot \\ B_n^t \end{bmatrix} = \begin{bmatrix} B_{e_1} \\ \cdot \\ \overline{B_{e_i}} \\ \cdot \\ B_{e_n} \end{bmatrix} + \begin{bmatrix} \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \overline{-\rho_i F_{i1}} & \overline{-\rho_i F_{i2}} & \cdot & \cdot & \overline{-\rho_i F_{in}} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} B_1^{t-1} \\ \cdot \\ B_i^{t-1} \\ \cdot \\ B_n^{t-1} \end{bmatrix}$$

- Cada *patch* P_i recolhe (*gathers*) radiossidade de todos os outros *patches*
- A resolução de cada linha da matriz calcula apenas um B_i^t
- Uma iteração percorre todas as linhas; calcula novo B_i^t para todos os *patches*
- Uma iteração implica o cálculo de todos os n^2 *form factors*



Radiosidade Progressiva

- A principal desvantagem do método anterior é que a resolução de uma linha da matriz aproxima a iluminação de apenas um *patch*
- Uma iteração aproxima a iluminação de todos os *patches* mas implica o cálculo de n^2 *form factors*
- O objectivo da radiosidade progressiva é conseguir rapidamente uma aproximação à iluminação de todos os *patches*
- Esta aproximação é refinada com mais iterações
- O refinamento progressivo é conseguido reordenando a forma como a radiosidade é calculada

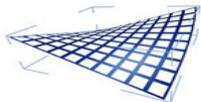


Radiosidade Progressiva - *shooting*

- A radiosidade clássica (*gathering*) calcula, para cada linha, a radiosidade de um patch i , B_i^t , devida à radiosidade de todos os *patches* B_j^{t-1}
- A radiosidade progressiva (*shooting*) calcula, para cada coluna, a radiosidade B_j^t de todos os *patches* devida à radiosidade B_i^{t-1} do *patch* i

$$\Delta B_j = B_j \text{ devida a } B_i = \rho_j B_i F_{ji}$$

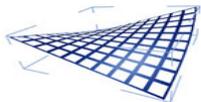
- $B_j^t = B_j^{t-1} + \Delta B_j$
- Para cada coluna da matriz apenas é necessário calcular n *form factors* para o *patch* P_i cuja radiosidade está a ser propagada (*shooting*)



Radiosidade Progressiva

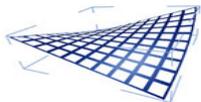
$$\begin{bmatrix} B_1^t \\ B_2^t \\ \cdot \\ \cdot \\ B_n^t \end{bmatrix} = \begin{bmatrix} B_1^{t-1} \\ B_2^{t-1} \\ \cdot \\ \cdot \\ B_n^{t-1} \end{bmatrix} + \begin{bmatrix} \cdot \\ \cdot \\ \frac{B_i^{t-1}}{\cdot} \\ \cdot \\ \cdot \end{bmatrix} \begin{bmatrix} \cdot \cdot \rho_1 F_{1i} \cdot \cdot \\ \cdot \cdot \rho_2 F_{2i} \cdot \cdot \\ \cdot \cdot \cdot \cdot \cdot \cdot \\ \cdot \cdot \cdot \cdot \cdot \cdot \\ \cdot \cdot \rho_n F_{ni} \cdot \cdot \end{bmatrix}$$

$$B_j^t = B_j^{t-1} + \underbrace{B_i^{t-1} \rho_j F_{ji}}_{\Delta B_j}$$

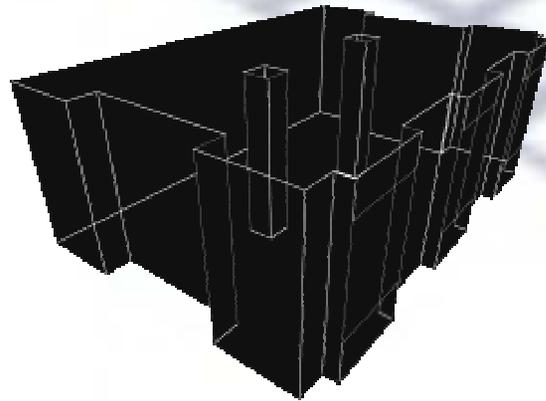
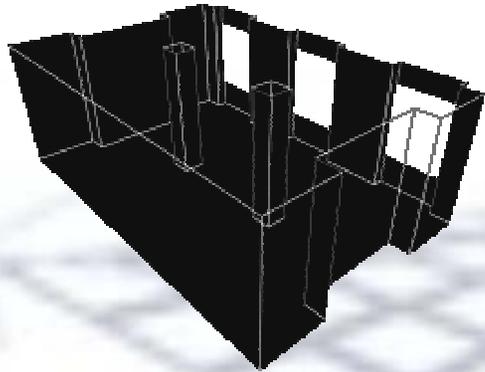


Radiosidade Progressiva

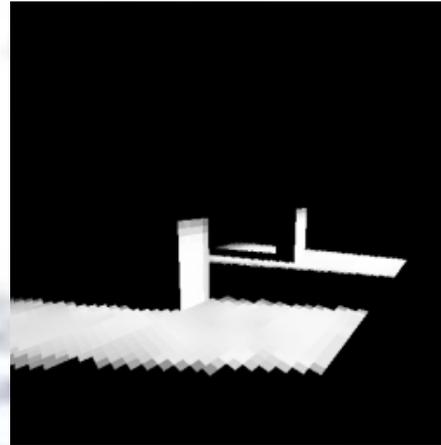
- Inicialmente $B_i^0 = B_{ei}$.
- Qual deve ser a ordem pela qual se seleccionam os *patches* P_i que vão propagar a sua radiosidade?
- Estes devem ser ordenados por B_i^t . O *patch* P_i com maior radiosidade no instante t deve ser o primeiro a propagar a sua radiosidade para que a solução convirja o mais rapidamente possível.
- O processo para quando os ΔB_j forem menores que um determinado limite.



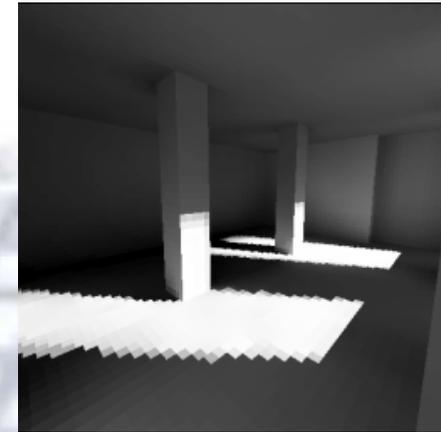
Radiosidade Progressiva



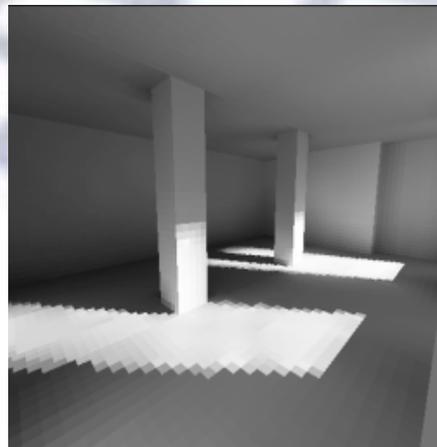
Geometria da cena



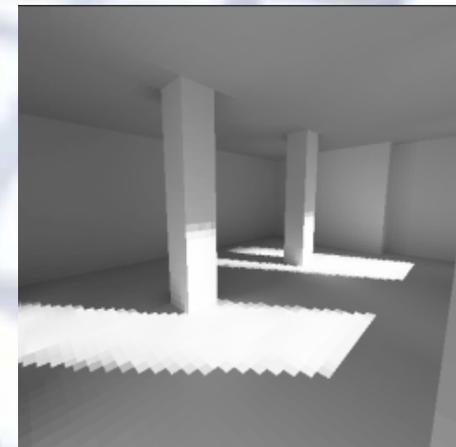
1 iteração



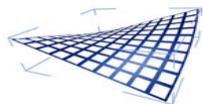
2 iterações



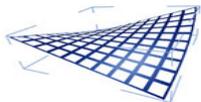
3 iterações



16 iterações

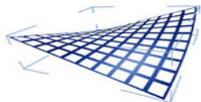


Radiosidade Progressiva

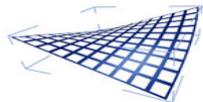
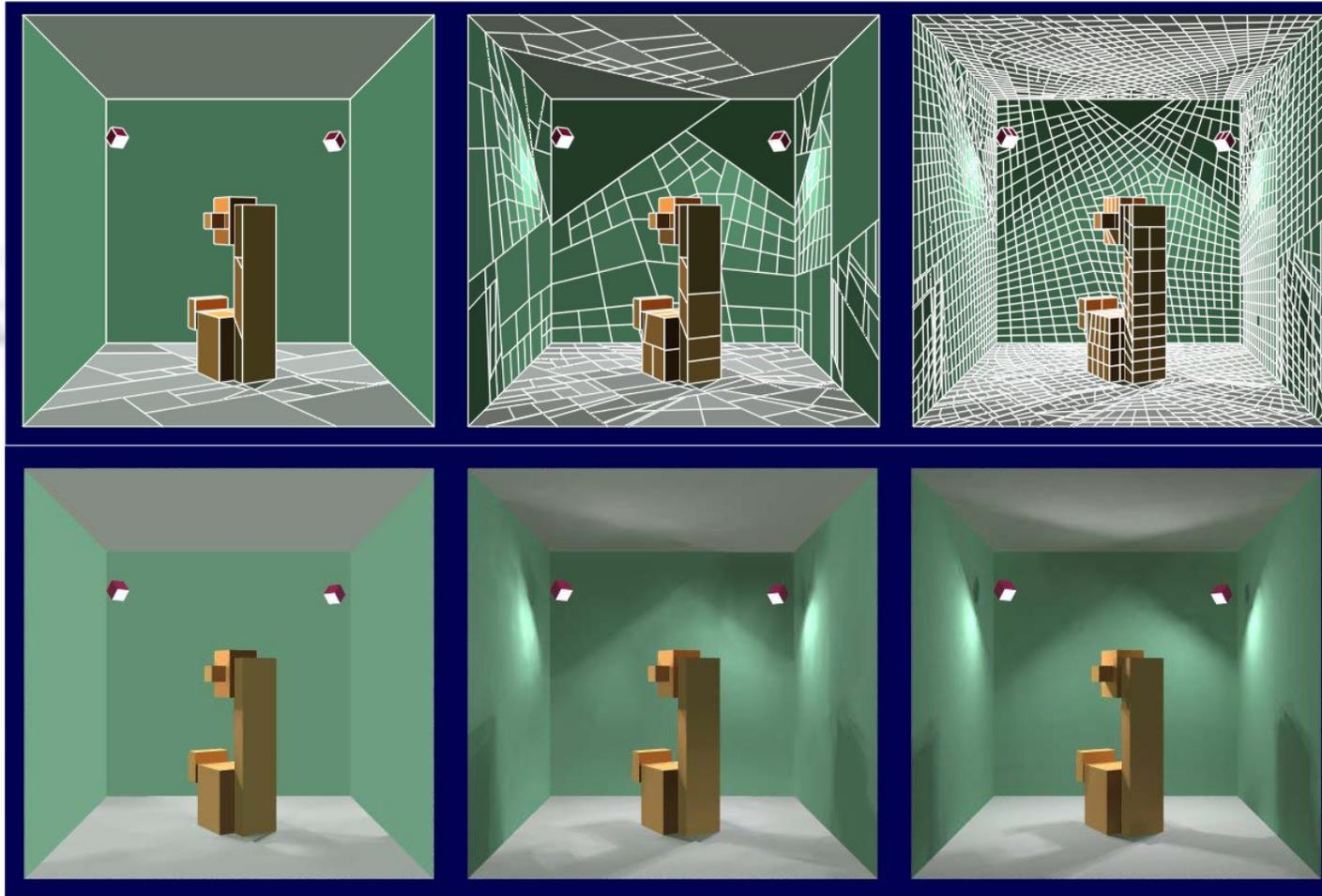


Refinamento da grelha de *patches*

- 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
- ou refinada durante a execução do próprio algoritmo, e.g., *discontinuity meshing*

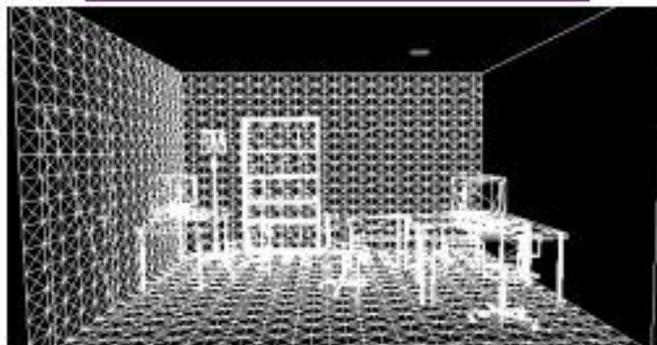


Refinamento da grelha de *patches*



Refinamento da grelha de *patches*

Initial (uniform) meshing



Rough radiosity calculation

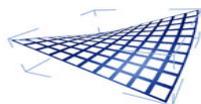
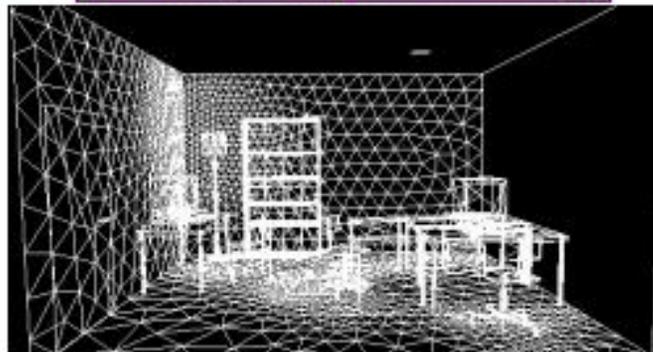


Polygon size evaluation

Final radiosity calculation



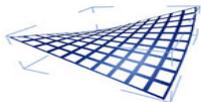
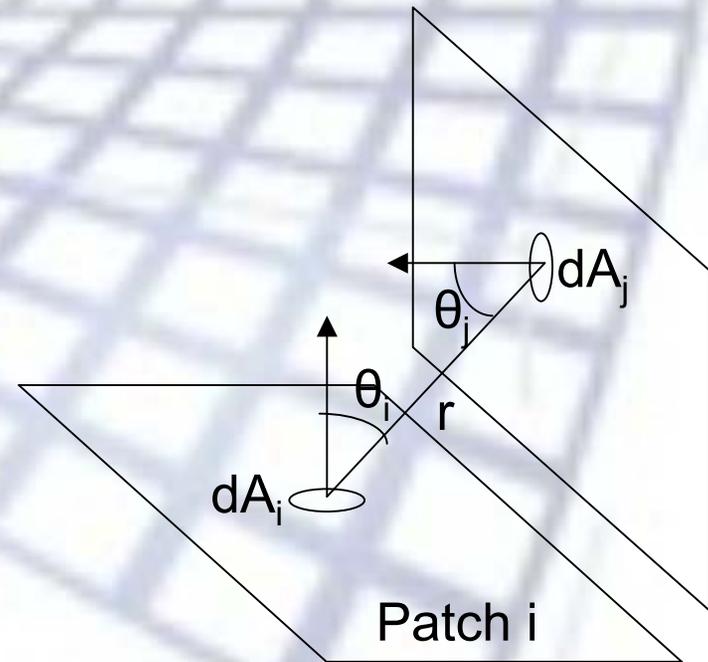
Intensity-adaptive meshing



Cálculo dos *form-factors*

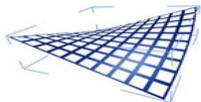
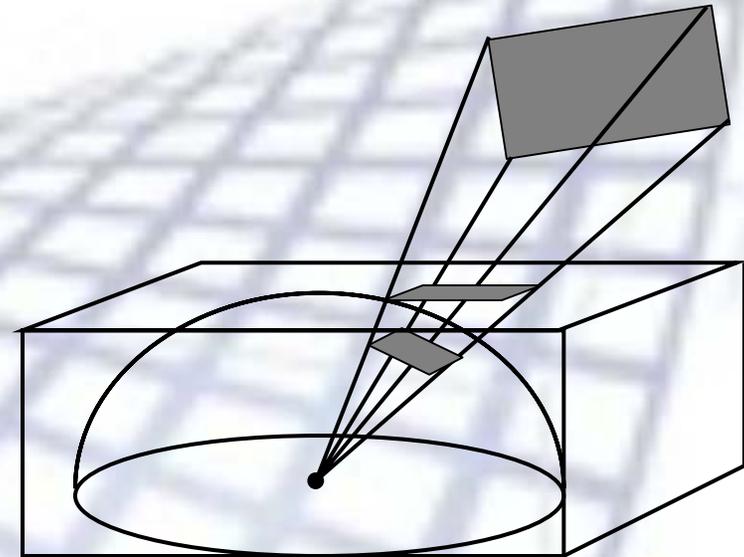
$$F_{ij} = \frac{1}{A_i} \int_{A_i} \int_{A_j} V(x_i, y_j) \frac{\cos(\theta_i) \cos(\theta_j)}{\pi r^2} dA_j dA_i$$

- F_{ij} representa a fracção da radiosidade emitida por P_j que incide em P_i .
- O cálculo analítico dos *form-factors* é dispendioso, recorrendo-se a soluções geométricas.



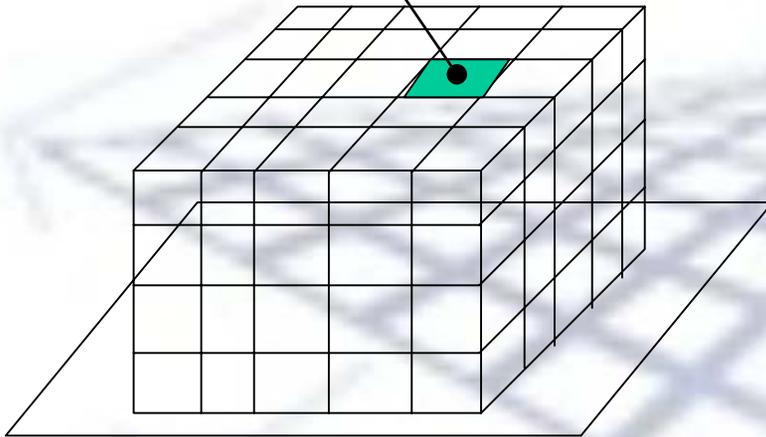
Form factors - HemiCubo

- **Analogia de Nusselt –**
O polígono, as suas projecções no hemicubo e na hemisfera têm os mesmos form factors
- Esta analogia permite usar projecções dos *patches* no hemicubo para calcular *form factors*



Form factors - HemiCubo

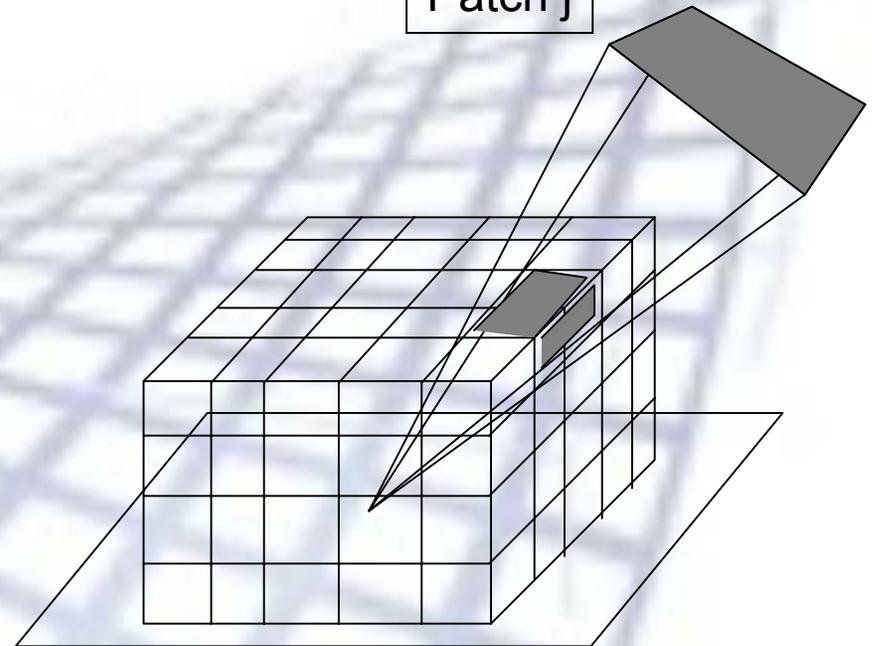
Pixel com área ΔA



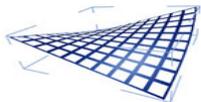
A área ΔA é igual para todos os elementos (pixels) do hemicubo. ΔF_q para cada elemento é dado por:

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

Patch j

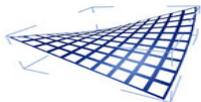


$$F_{ij} = \sum_q \Delta F_q$$



Form factors - HemiCubo

- Para cada *patch* i todos os *patches* j devem ser projectados no hemicubo
- Vários *patches* 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 do Z-buffer)
- Para cada pixel apenas o *patch* P_j mais próximo interessa, pois todos os outros estarão ocluídos por este.



Form factors: ray tracing

- Em vez do hemicubo pode ser usada a hemisfera unitária centrada em Pi, com a superfície dividida em áreas elementares
- Cada área elementar corresponde aos ângulos $(\Delta\varphi, \Delta\theta)$

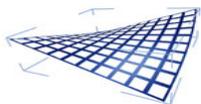
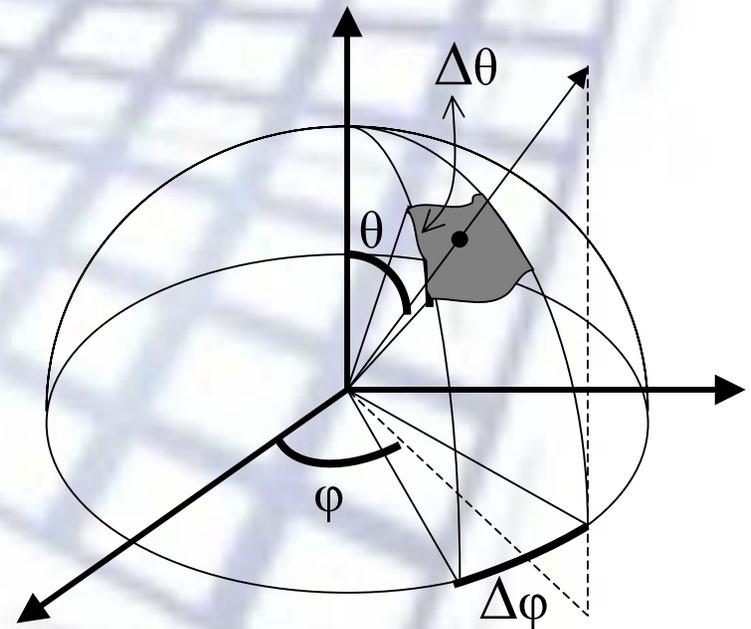
$$\Delta\varphi = \Delta\theta = 2\pi / k$$

$$\varphi \in [0, 2\pi[, \varphi = l * \Delta\varphi, l = 0, 1, \dots, k - 1$$

$$\theta \in [0, \pi / 2[, \theta = m * \Delta\theta, m = 0, 1, \dots, k / 4$$

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

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

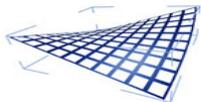
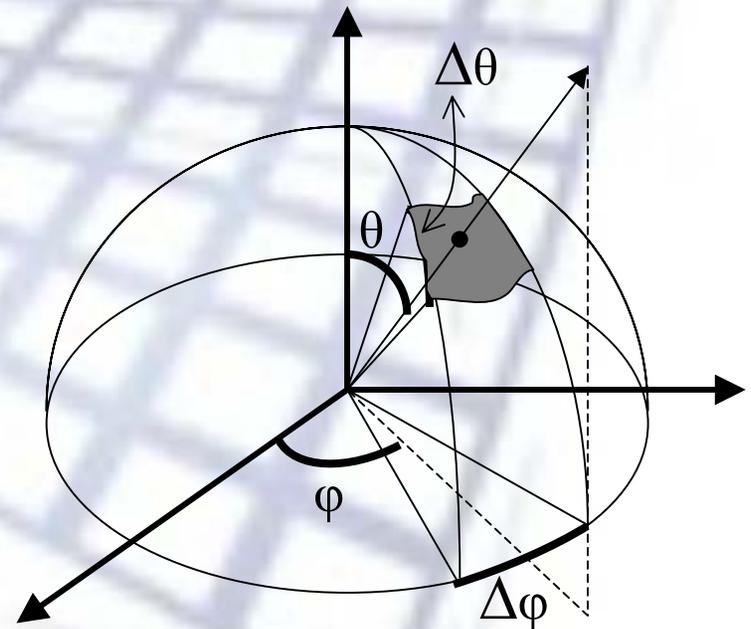


Form factors: ray tracing

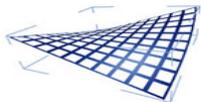
- Cada área elementar dá origem a um raio que determina qual o *patch* visível ao longo daquela direcção

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

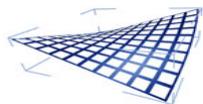
- A opção pelo hemicubo ou pelo ray tracing depende do desempenho dos algoritmos de projecção e de intersecção utilizados



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*.

