

Iluminação Global

André Tavares da Silva

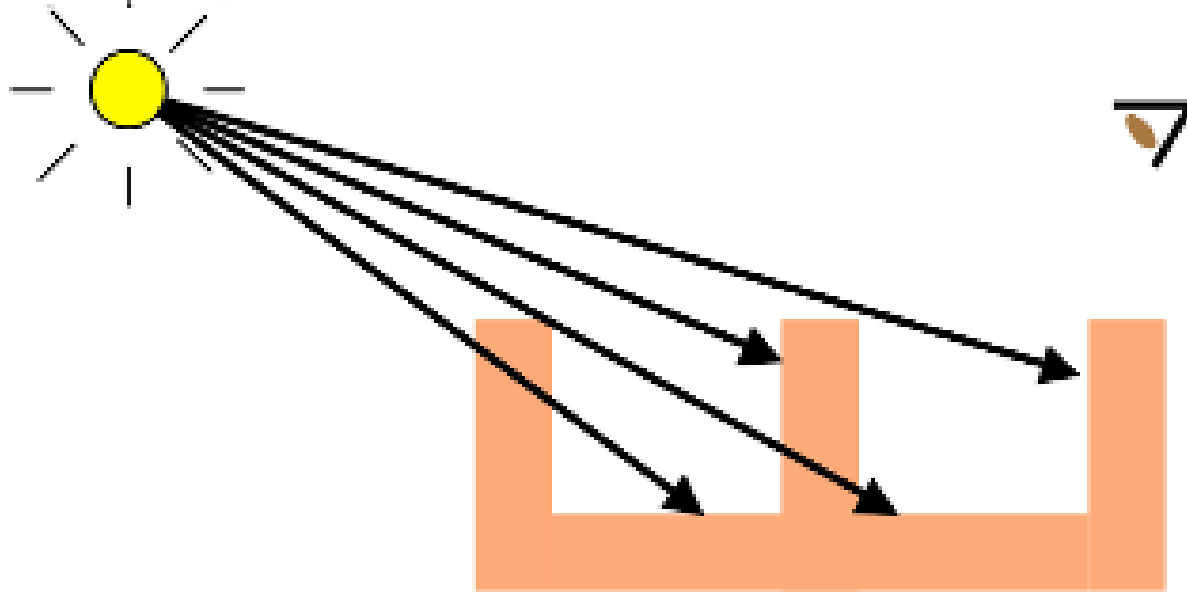
andre.silva@udesc.br

Modelos de Iluminação

- Descrevem como a luz
 - Interage com os materiais
 - É transportada na cena (*light transport*)
 - Atinge o observador
- Categorias
 - Modelos de Iluminação Locais
 - Modelos de Iluminação Globais

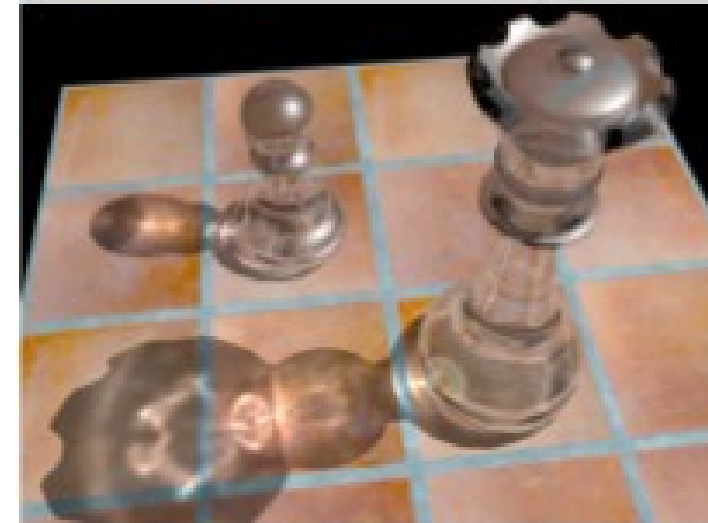
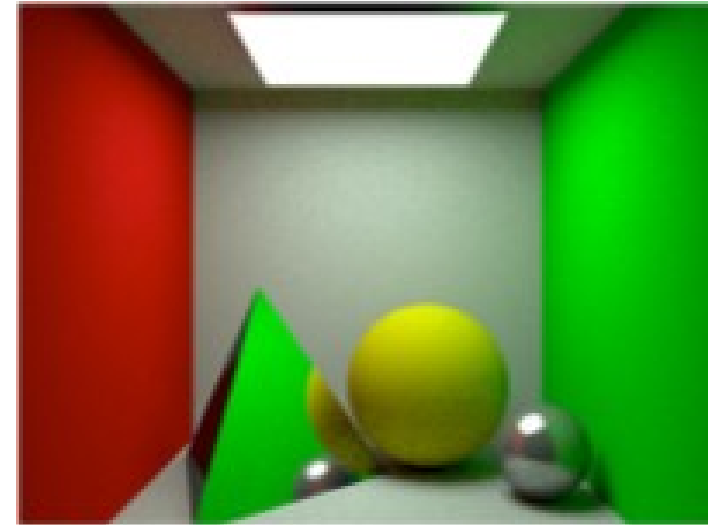
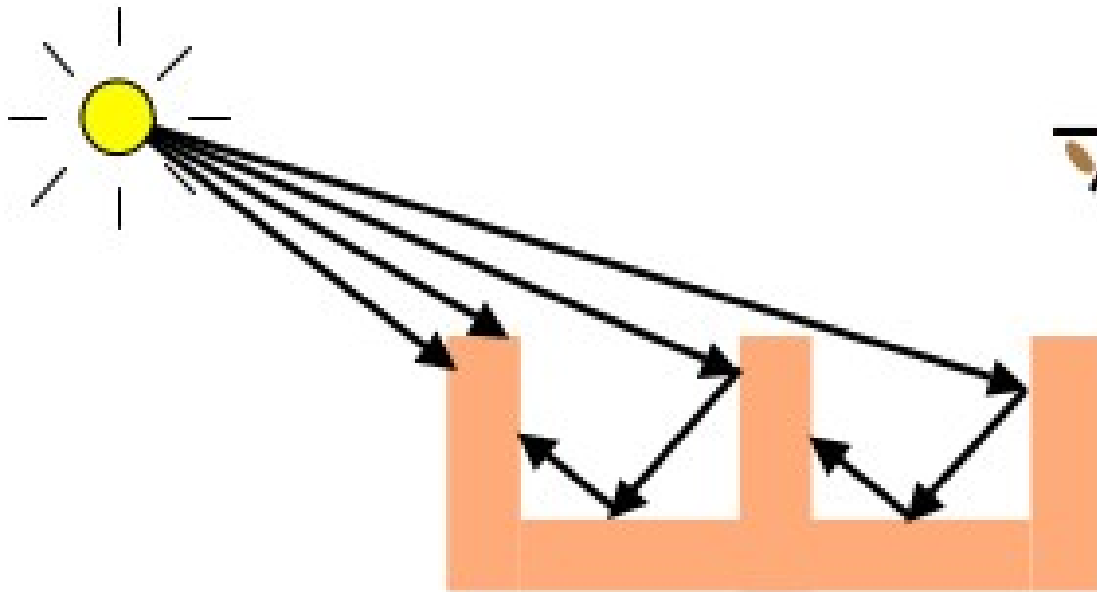
Modelos de Iluminação Locais

- Não consideram inter-reflexões
- Rápidos para cálculo
- Não são fisicamente corretos
- Em geral, baixo realismo



Modelos de Iluminação Globais

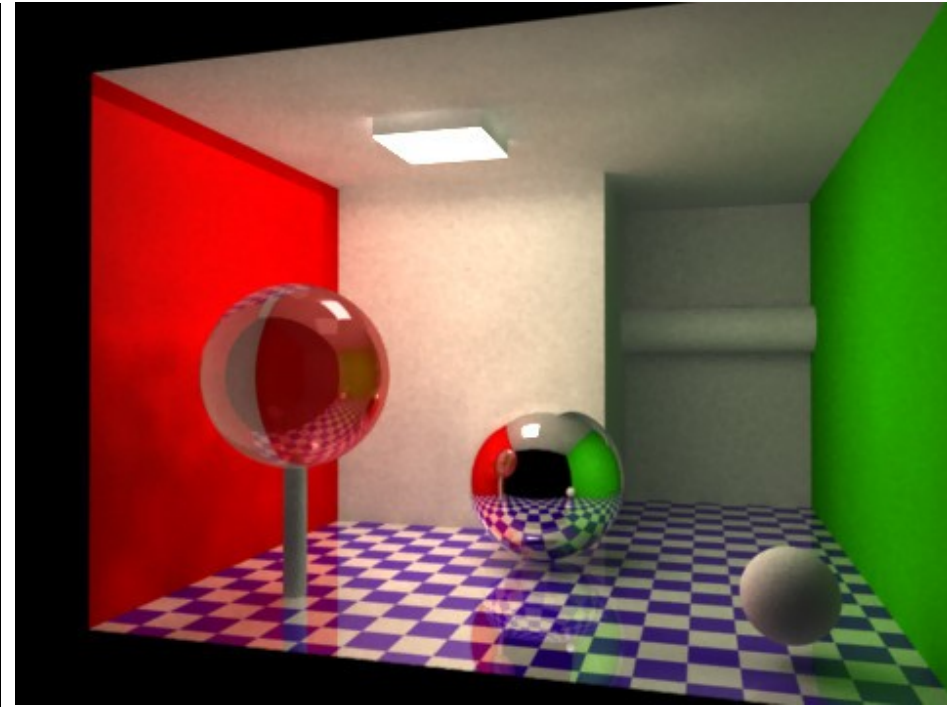
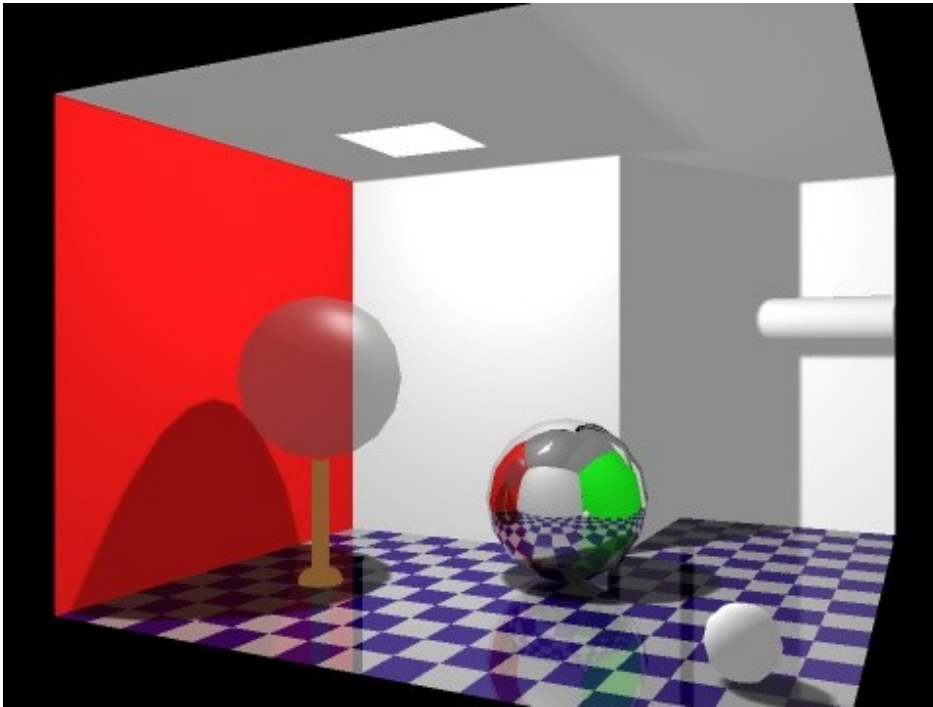
- Toda a cena é considerada
- Consideram inter-reflexões
- Maior custo computacional
- Chave para rendering realista



Exemplo

Local (OpenGL)

Global



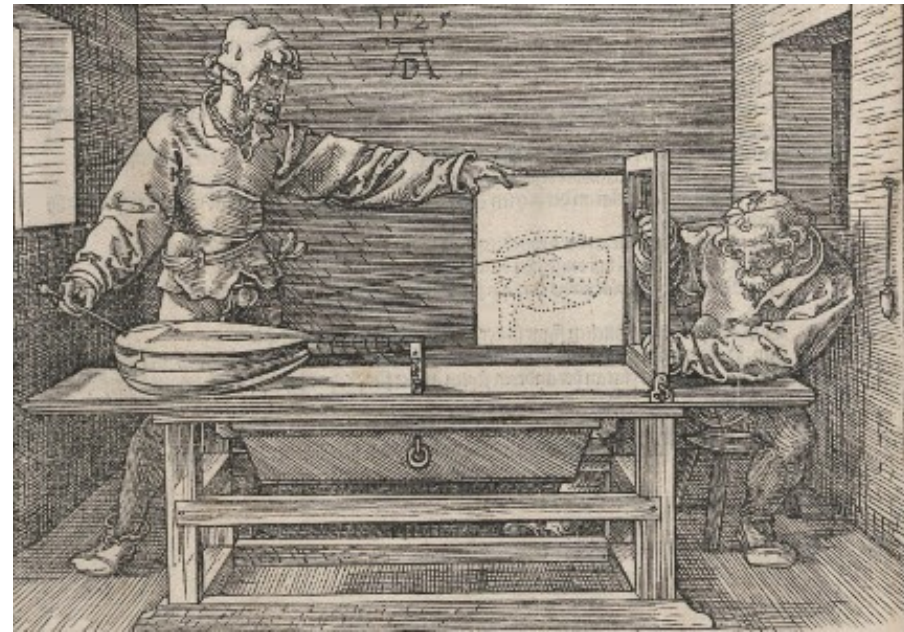
http://www.winosi.onlinehome.de/Gallery_t14_03.htm

Modelo de Iluminação Utópico

- Poderíamos ter um modelo em computador simulando toda a interação luminosa entre as fontes de luz e objetos;
- Para cada fonte de luz, traçar raios em todas as infinitas direções rebatendo em todos os objetos;
- A cada raio, traçar outros infinitos para todas as direções possíveis.

Modelos de Iluminação Global

- Ao invés de se emitir um número infinito de raios da fonte de luz para o objeto e depois para o observador, são emitidos um número finito de raios do observador através dos pontos de amostragem para os objetos e depois para a fonte de luz (ou outro objeto).
- Generalização do que foi definido por Albrecht Dürer para estudo de projeção perspectiva para pinturas.



Ray Casting

(primeira ideia de raios)

Para cada pixel da tela

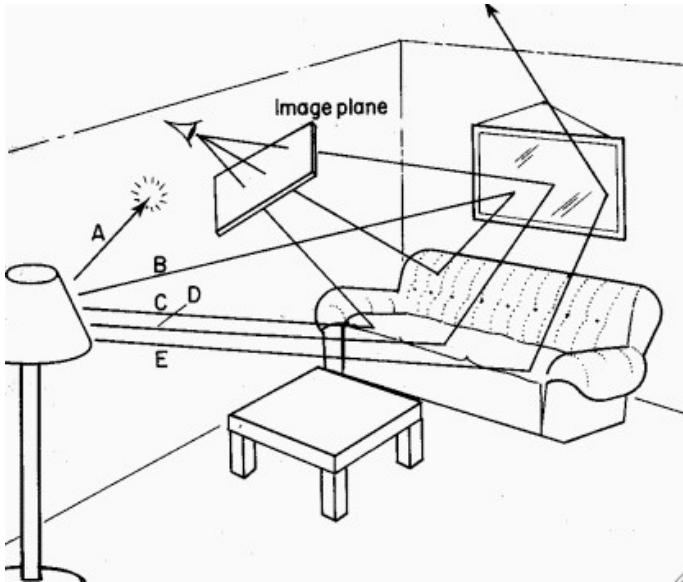
Construa um raio a partir do olho

Para cada objeto na cena

Encontre a intersecção com o raio

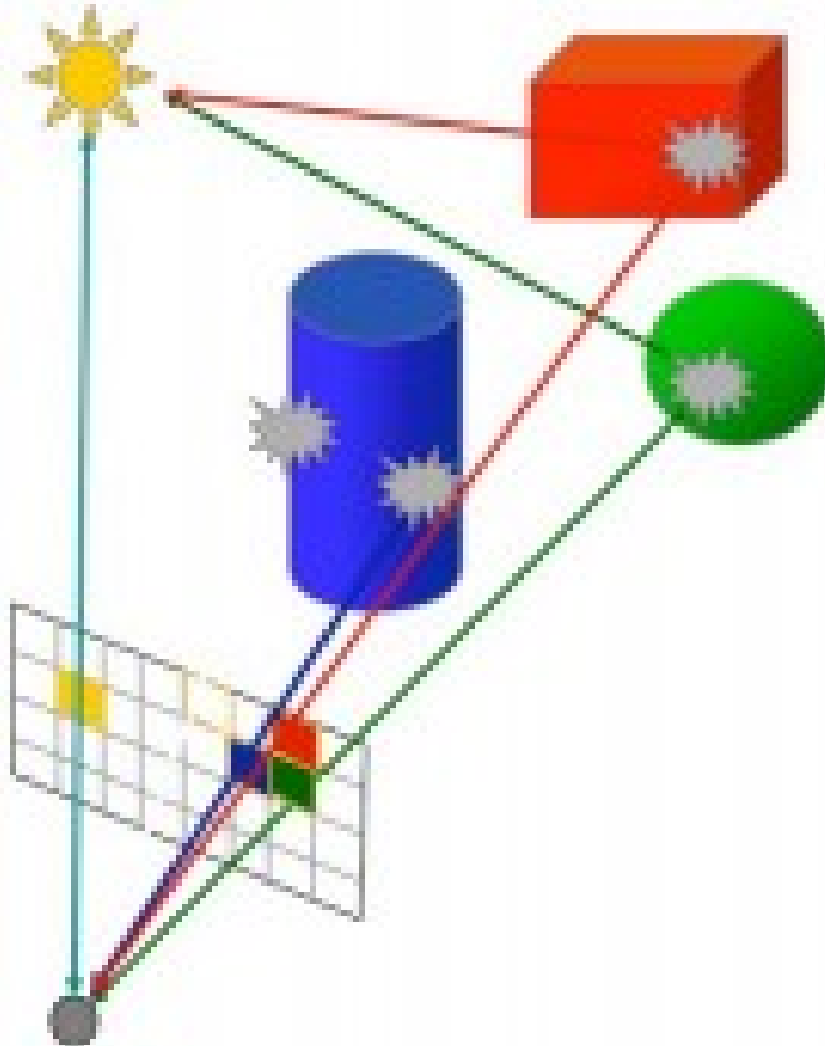
Mantenha se for a mais próxima

Calcule a iluminação neste ponto

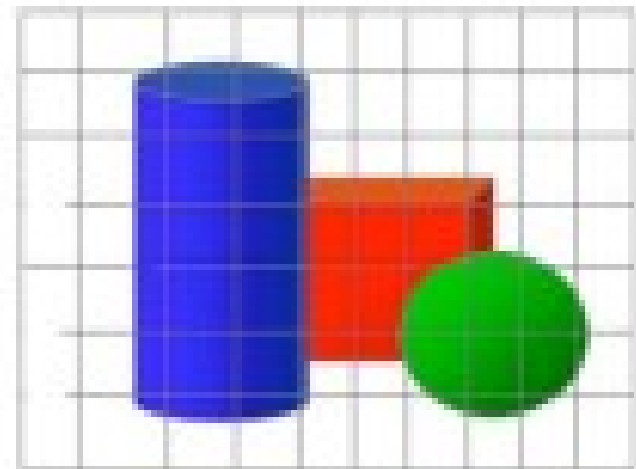


Arthur Appel. Some Techniques for
Shading Machine Renderings of Solids
AFIPS Spring Joint Computer Conf,
p. 37-45, **1968**.

Ray Casting



direct illumination



Ray Tracing (recursivo)

- Gere raios primários, que vão da posição de observação aos pontos de amostragem:
 - Encontre objeto mais próximo do observador ao longo do raio, isto é, ache a primeira interseção entre o raio e um objeto da cena
 - Use modelo de iluminação para determinar luz no elemento de superfície mais próximo
 - Gere **raios secundários** que se originam no objeto interceptado e assim sucessivamente

Sem Ray Tracing



Ray Tracing



PIXAR ANIMATION STUDIOS

Toy Story 3 | Special Image | Pixar Creative Services
generated from element: light_comp_nomo_nodof_film
tt340_8pub2.special16.118.tif - 2010:02:03 13:51:22 - (1920 x 1080)

Ray Tracing



Ray Tracing



Ray Tracing

- Gere raios primários, que vão da posição de observação aos pontos de amostragem:
 - Encontre objeto mais próximo do observador ao longo do raio, isto é, ache a primeira interseção entre o raio e um objeto da cena
 - Use modelo de iluminação para determinar luz no elemento de superfície mais próximo
 - Gere **raios secundários** que se originam no objeto interceptado e assim sucessivamente

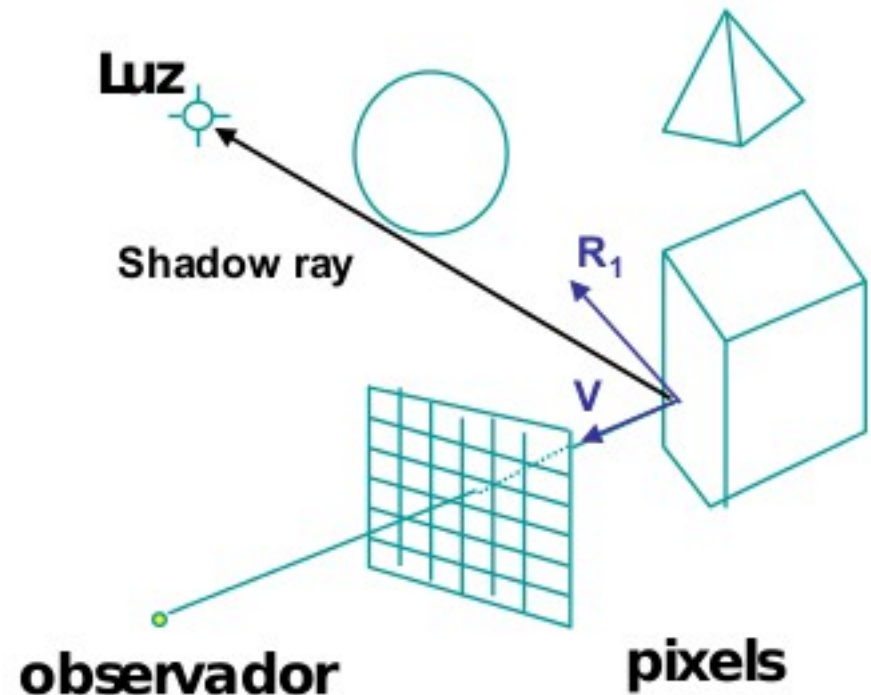
Ray Tracing

R_1 é o vector de reflexão
máxima:

$$R_1 = V - 2(V \cdot N)N$$

Intensidade Luminosa Inicial:

$$I = k_a I_a + k_d (N \cdot L)$$



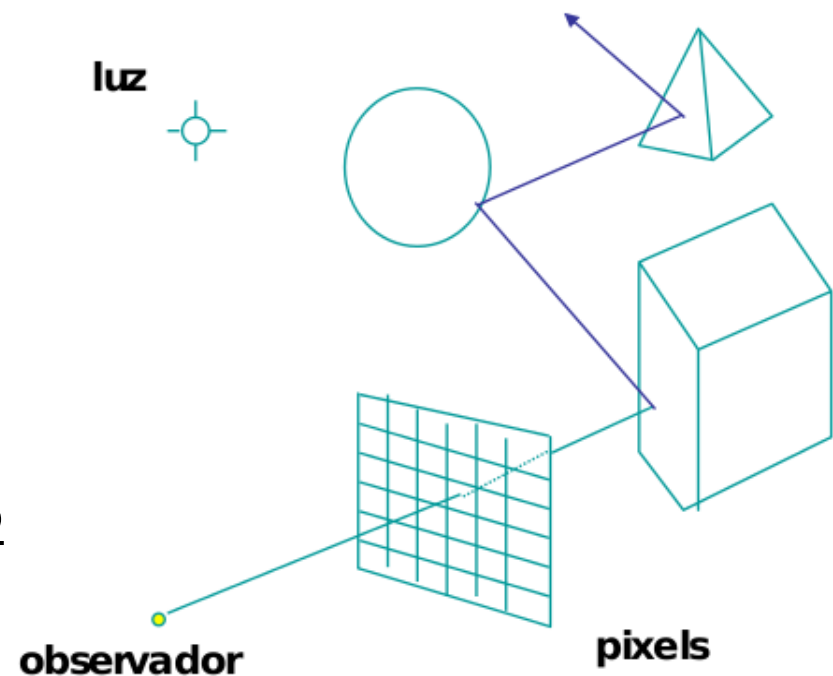
Ray Tracing - reflexão

Intensidade luminosa agora é:

$$\mathbf{I} = k_a I_a + k_d (\mathbf{N} \cdot \mathbf{L}) + \mathbf{k}_r \mathbf{I}_r$$

\mathbf{I}_r é calculada **recursivamente**

\mathbf{k}_r é um coeficiente de Reflexão



Em cada intercepção é necessário determinar qual o objeto mais próximo.
Pode ser considerada uma atenuação devido à distância da face à luz.

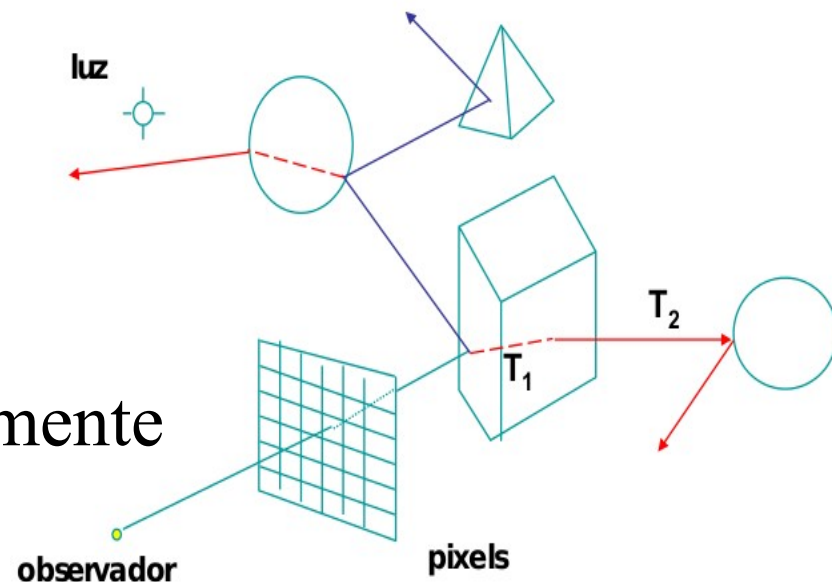
Ray Tracing - refração

Se os objetos forem **translúcidos** é necessário considerar os raios transmitidos para o interior (e de volta para o exterior) do objeto.

Intensidade luminosa fica:

$$\mathbf{I} = k_a \mathbf{I}_a + k_d (\mathbf{N} \cdot \mathbf{L}) + k_r \mathbf{I}_r + \mathbf{k}_t \mathbf{I}_t$$

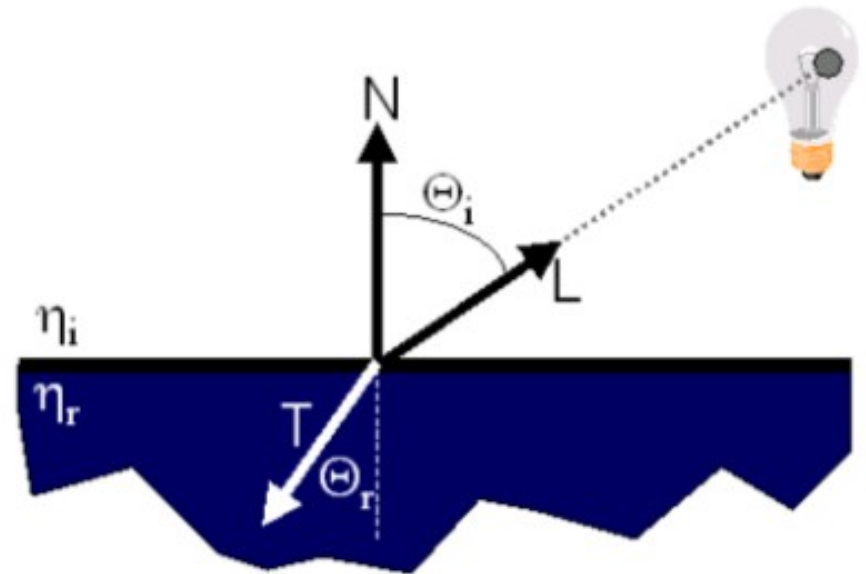
\mathbf{I}_r e \mathbf{I}_t são calculadas recursivamente



Ray Tracing - refração

- Para superfícies finas, podemos ignorar a mudança de direção, assumindo que a luz viaja em linha reta através da superfície;
- Para objetos sólidos, aplica-se a Lei de Snell:

$$\eta_r \sin \Theta_r = \eta_i \sin \Theta_i$$



$$T = \left(\frac{\eta_i}{\eta_r} \cos \Theta_i - \cos \Theta_r \right) N - \frac{\eta_i}{\eta_r} L$$

Ray Tracing

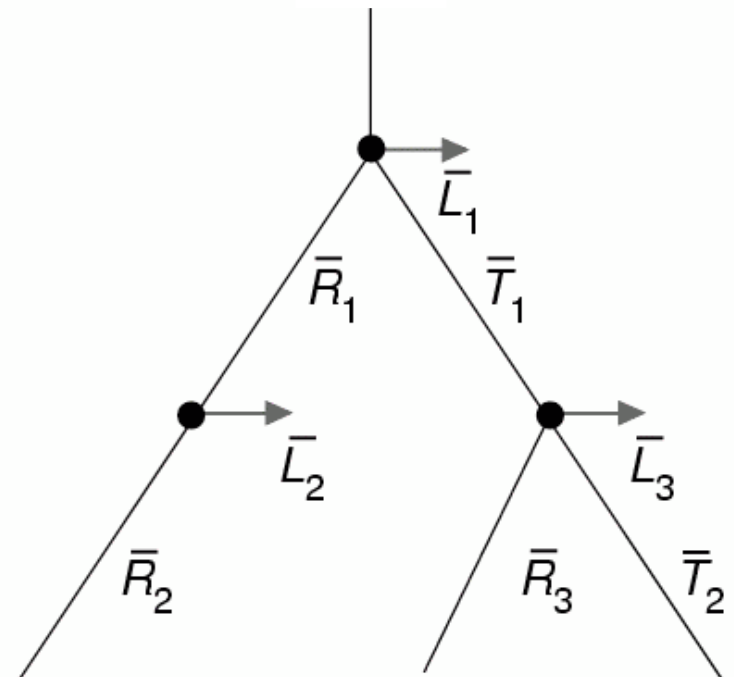


Árvore de Raios

Para cada pixel constrói-se uma **árvore de intersecções**. A cor final do pixel é determinado percorrendo-se a árvore das folhas para a raiz e calculando as contribuições de cada ramo de acordo com o modelo de reflexão.

Em objetos opacos não existe o raio transmitido.

O ramo da árvore termina quando o raio atinge um objeto não refletor ou atinge uma determinada profundidade pré-estabelecida.



```
void RT( Point3D start, Point3D end, int depth, RGB *color) {
    if ( depth > MAXDEPTH ) *color = BLACK;
    else { /* verifica se raio intersecta algum objeto. */
        if ( rayHit ( start, end, &hitObject, &hitPoint )){ // retorna o mais
próximo se possível
            /* contribuicao local - se luz atinge ou não o ponto */
            shade( hitObject, hitPoint, localColor );

            /* calcula direções de reflexão e transmissão */
            calcReflection( hitObject, hitPoint, &reflectDirection);
            calcTrans( hitObject, hitPoint, &transmDirection );

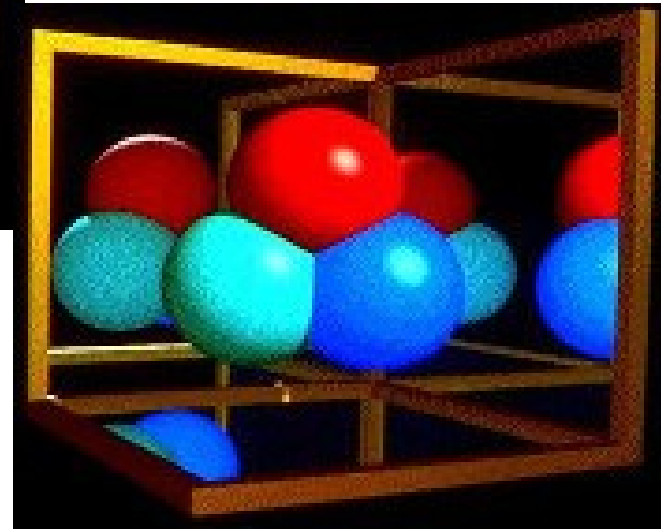
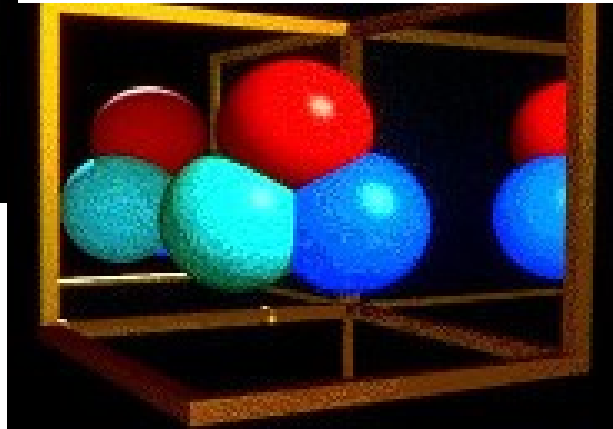
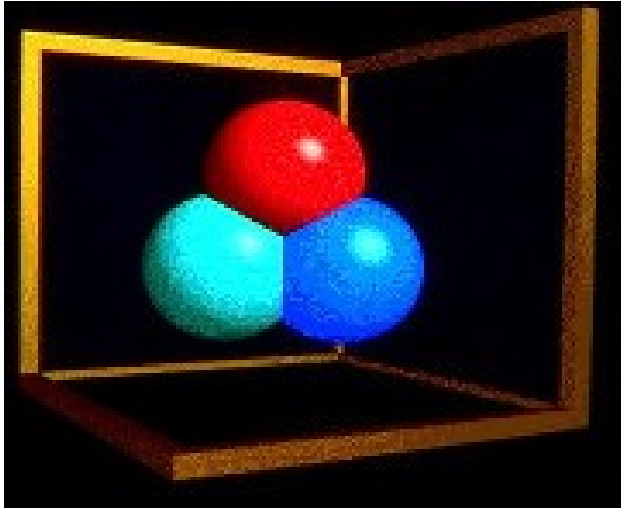
            /* Chamadas recursivas - reflexão e transmissão*/
            RT ( hitPoint, reflectDirection, depth+1, &reflectedColor );
            RT ( hitPoint, transmDirection, depth+1, &transmColor );

            /* combina cores */

            combineColor(hitObject,localColour,reflectedColour,transmColour,color);
        }
        else *color = BLACK;
    }
}
```

Obs.: falta incluir vários objetos e fontes de luz (baseado no material de Marcelo Walter)

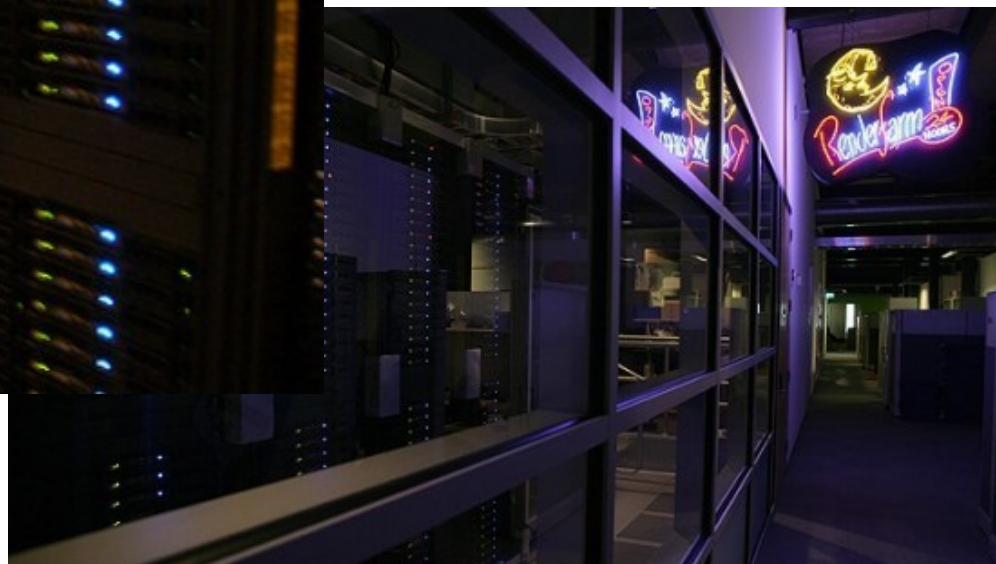
Múltiplos níveis de reflexão



Ray Tracing

- Vantagens:
 - sombras, reflexões e refrações são facilmente incorporadas
 - simula razoavelmente bem os efeitos especulares
- Desvantagens:
 - o custo de cálculo das intersecções é elevado
 - não simula bem os efeitos de iluminação difusa

Render Farm



Big Hero 6 \approx 1,1 bilhões de horas
55.000 cores em 4 localizações

Ray Tracing

- A fim de reduzir o número de intersecções para melhorar a performance pode-se aplicar algumas otimizações.
- Elas são feitas em duas áreas:
 - Diminuição do número de raios a processar.
 - Diminuição do número de intersecções a testar

Diminuição do número de raios

- ***Item Buffers*** - determinam-se quais as áreas da tela onde se situam os objetos (*Z-Buffer* e outros pré-processamentos)
- ***Adaptive Tree-Depth Control*** - não é necessário levar todos os ramos da árvore de *shading* à sua profundidade máxima (importância de um raio luminoso sobre o pixel a que pertence, diminui a cada reflexão ou transmissão)
- ***Light-Buffers*** - a cada fonte de luz associam-se listas com os objetos que a rodeiam (em cada direção e por ordem de afastamento); ao processar uma intersecção, são gerados raios por reflexão, por transmissão e para as fontes de luz; estes últimos, uma vez definida a sua direção, vêm limitadas as hipóteses de intersecção com os objetos que se encontram na lista respectiva.

Diminuição do número de intersecções

- **Volumes Envolventes** - antes de efetuar o teste de intersecção de um raio com um objeto, tenta-se a sua intersecção com um volume simples (caixa, esfera,...) envolvente do objeto. Este teste prévio é muito rápido e exclui imediatamente muitos testes de intersecção mais complexos.
- **Organização Hierárquica dos Volumes Envolventes** - a utilização de volumes envolventes de outros volumes envolventes permite economizar muitos testes de intersecção: se um raio não intersecta um volume, então também não intersecta os volumes nele contidos.

Diminuição do número de intersecções

- **Divisão Espacial em Grades Tridimensionais** - cada célula resultante desta divisão conhece os objetos que contém, total ou parcialmente. De acordo com a posição e a direção do raio em questão, só determinadas células são visitadas e, deste modo, só os objetos nelas contidos são testados. Dado que a ordem de progressão nas células é definida pelo sentido do raio, a primeira célula onde se detecte uma intersecção termina o processo de visita do raio às células.

Ray Tracing

- Vantagens:
 - sombras, reflexões e refrações são facilmente incorporadas
 - simula razoavelmente bem os efeitos especulares
- Desvantagens:
 - o custo de cálculo das intersecções é elevado
 - não simula bem os efeitos de iluminação difusa

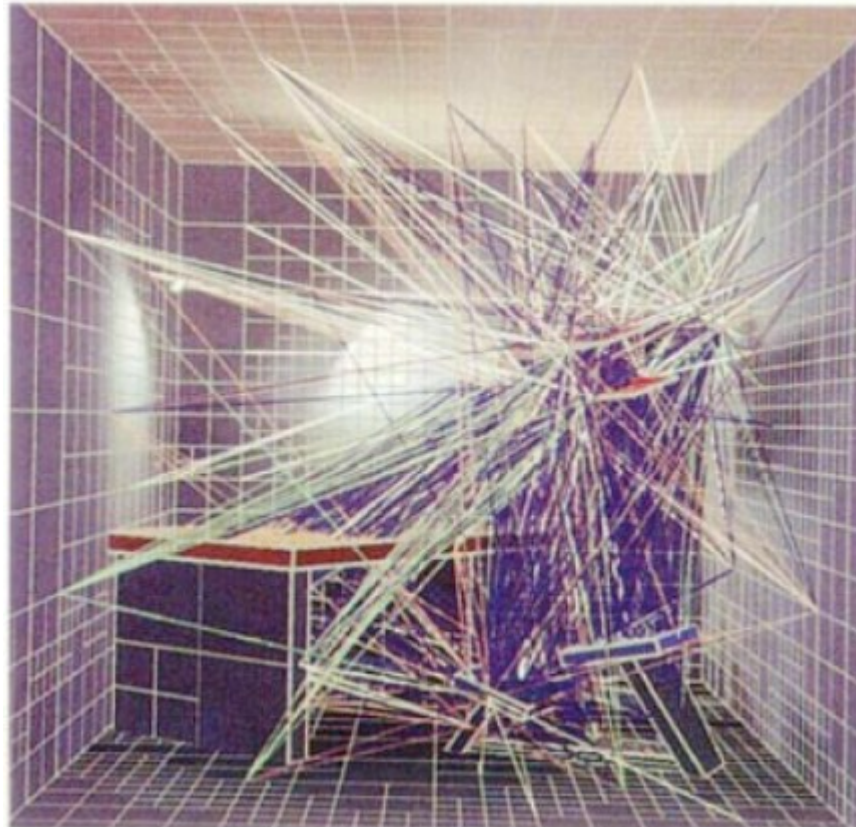
Iluminação Indireta

- Luz recebida diretamente e indiretamente através de inter-reflexões no ambiente;
- *Color bleeding*: a cor do objeto é transmitida

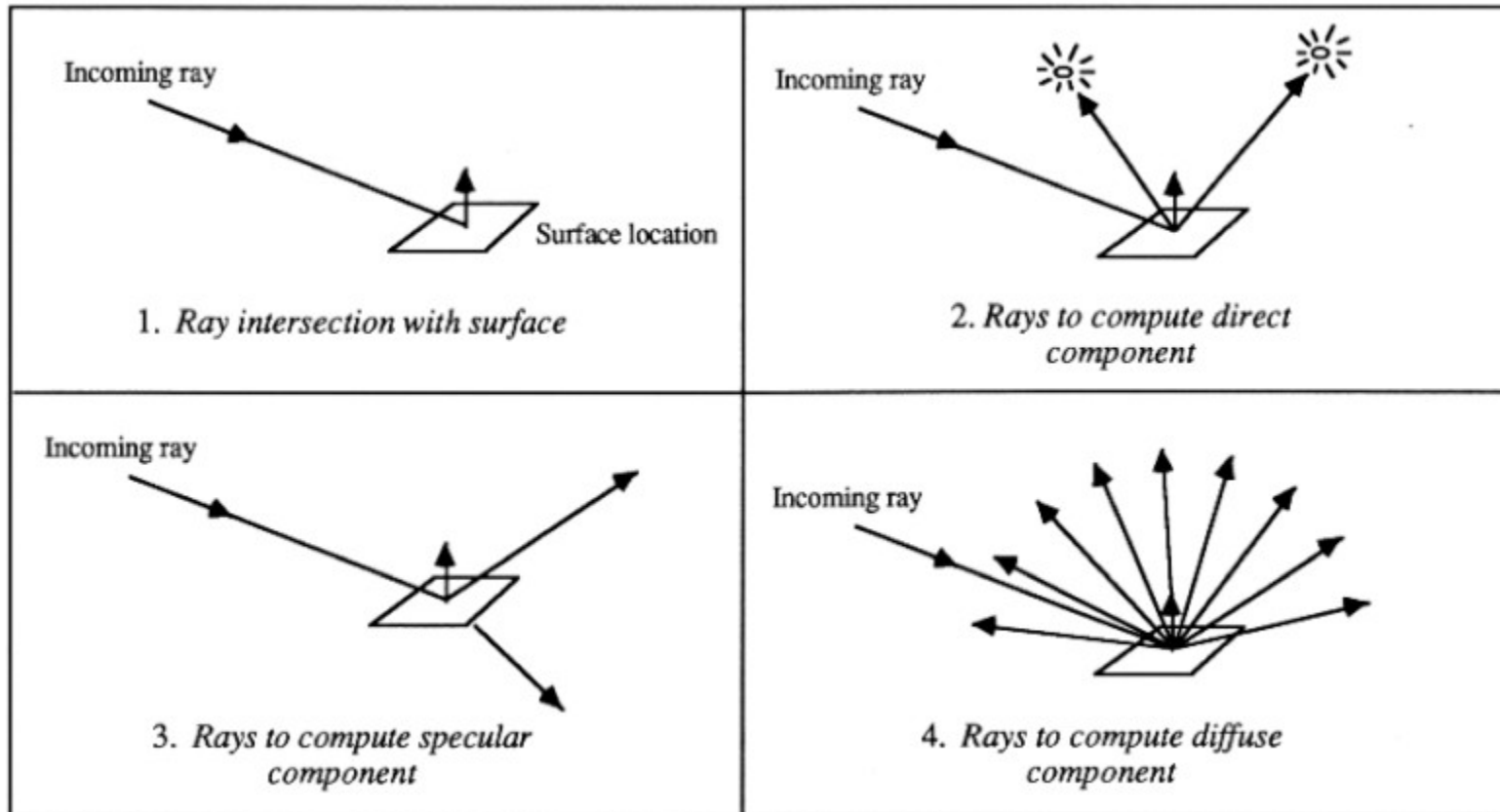


Parcela Difusa em Ray Tracing

- Como aproximar as inter-reflexões difusas?
- Para cada raio, disparar um número N de raios difusos no ambiente em direções aleatórias ao redor da normal.



Parcela Difusa em Ray Tracing



A Ray Tracing Solution to Diffuse Interreflection by
Greg Ward, Francis Rubinstein and Robert Clear SIGGRAPH 1988

Parcela Difusa em Ray Tracing



rt normal



1 nível



7 níveis

Radiosidade

- Ray tracing modela a reflexão especular e transparência refrativa, mas ainda usa uma componente ambiente nos efeitos de luz
- Radiosidade é a taxa à qual a energia é emitida ou refletida
- Estes efeitos de radiosidade podem ser traçados pela superfície pela conservação da energia da luz num volume

Radiosidade

- O algoritmo é **independente do ponto de observação**. O algoritmo só realiza o cálculo de iluminação e trabalha no espaço de objeto.
- É complementado por um algoritmo de cálculo de visibilidade para a produção da imagem final.
- Fases do processamento:
 1. Modela as interações entre objetos e fontes de luz, sem considerar a posição do observador.
 2. Cria a imagem considerando o observador, efetuando cálculo de visibilidade (*z-buffer*, por exemplo) e sombreamento de polígonos (Gouraud).

Radiosidade

- Nos modelos anteriores (Ray Tracing e modelos de iluminação local), as fontes de luz foram tratadas de forma diferente das superfícies que iluminam. Já os métodos de radiosidade consideram que todas as superfícies podem (auto-)emitir luz. Assim, as fontes de luz são modeladas como superfícies normais, com uma dada área.
- O método assume que os processos de emissão e reflexão são difusos ideais (superfícies lambertianas). Necessita das faces discretizadas em *patches* de forma a garantir que na área correspondente a um *patch* a radiosidade se mantém constante.

Patches



Radiosidade

- A radiosidade é definida como a energia expelida (\mathbf{B}_i) por unidade de área de um *patch* (\mathbf{A}_i), sendo composta por:

$$\mathbf{B}_i \mathbf{A}_i = \mathbf{E}_i \mathbf{A}_i + \rho_i \sum_j (\mathbf{F}_{j-i} \mathbf{B}_j \mathbf{A}_j)$$

\mathbf{B}_i - radiosidade, energia expelida do *patch* em Watt/m²

\mathbf{E}_i - emissão, luz auto-emitida pelo *patch* \mathbf{i}

ρ_i - reflectividade,

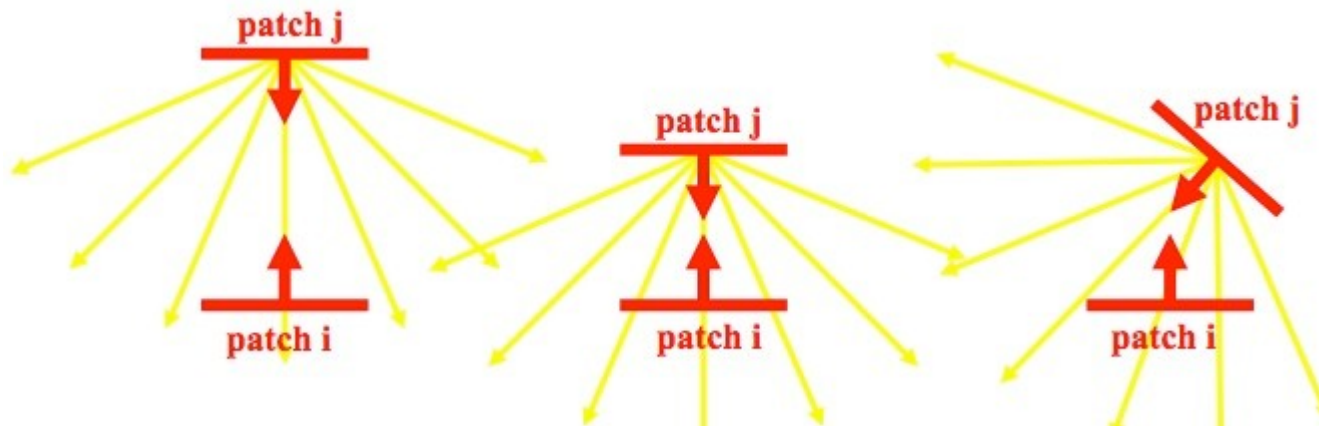
percentagem da energia incidente reflectida pelo *patch* \mathbf{i}

\mathbf{F}_{j-i} - fator de forma,

percentagem de energia que abandona o *patch* \mathbf{j} e atinge \mathbf{i}

Radiosidade

- Os **fatores de forma** (FF) representam geometricamente como um *patch* “enxerga” os outros.
- A complexidade do método de radiosidade está no cálculo dos fatores de forma ($\sim 90\%$).
- Leva em consideração forma, orientação e visibilidade entre os *patches*.



Radiosidade

- Em ambientes difusos, existe a seguinte relação de reciprocidade entre fatores de forma: $\mathbf{A}_i \cdot \mathbf{F}_{i-j} = \mathbf{A}_j \cdot \mathbf{F}_{j-i}$
- Que aplicada na expressão anterior da radiosidade resulta em:
 $\mathbf{B}_i - \rho_i \sum_j \mathbf{B}_j \mathbf{F}_{i-j} = \mathbf{E}_i$
- Assim, a interação de luz entre *patches* pode ser representada por um sistema de equações lineares:

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

$$\mathbf{M} \cdot \mathbf{B} = \mathbf{E}$$

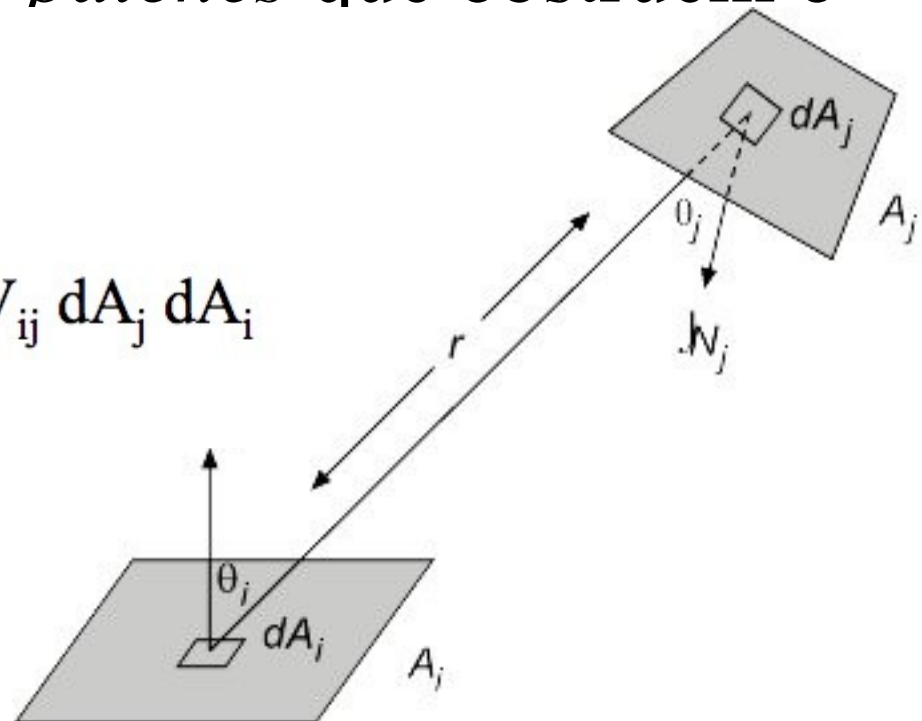
Radiosidade

- Problema: **M é enorme** (número de *patches*) e acarreta problemas de armazenamento.
- Solução: Utilizar métodos numéricos
- Eliminação Gaussiana muito cara computacionalmente ($O(n^3)$)
- Melhores métodos iterativos
 - “Chute inicial” para a solução
 - Calcula uma melhor aproximação a cada iteração
 - Iteração de Gauss-Seidel, Jacobi ou Southwell

Fator de Forma

- O fator de forma representa a fração de energia de um determinado *patch* que atinge outro, tomando em consideração a forma e orientação relativa de ambos bem como outros *patches* que obstruem o caminho.

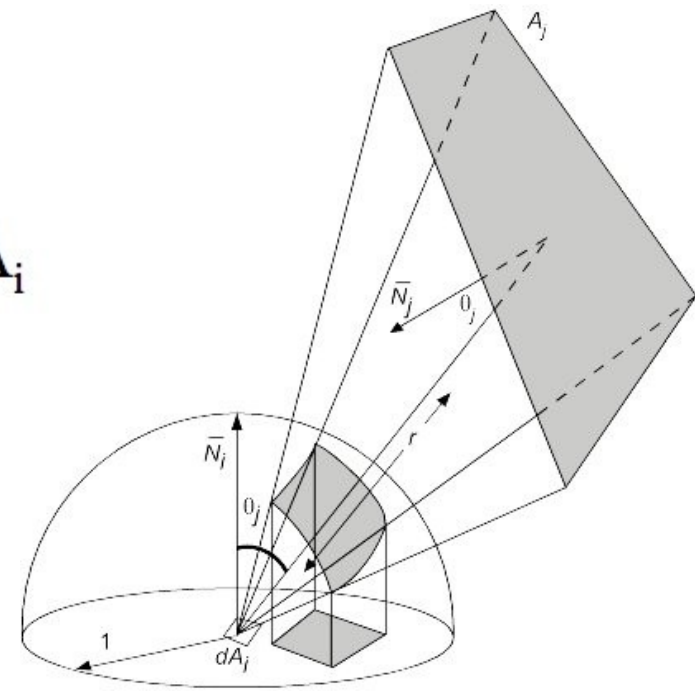
$$F_{ij} = \frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{\cos \theta_i \cos \theta_j}{\pi r^2} V_{ij} dA_j dA_i$$



Fator de Forma

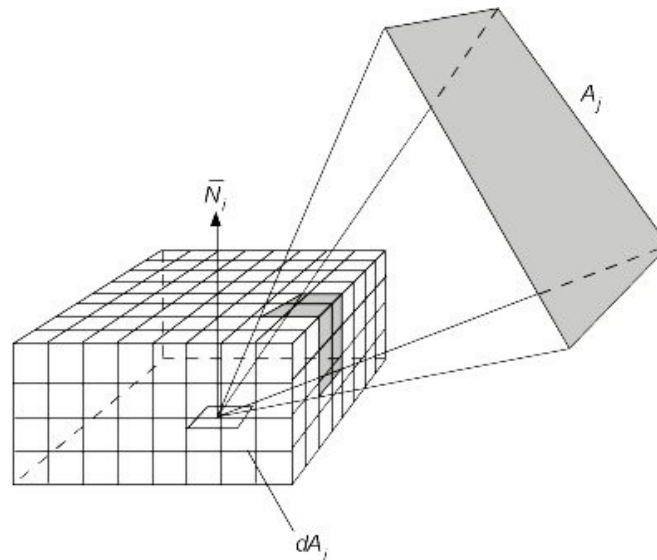
- O cálculo do Fator de Forma F_{i-j} corresponde a projetar as partes de A_j visíveis de dA_i num hemisfério centrado em dA_i , projetando depois de forma ortográfica na base do hemisfério e dividindo pela área do círculo. (Analogia de Nusselt)

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



Fator de Forma - simplificação

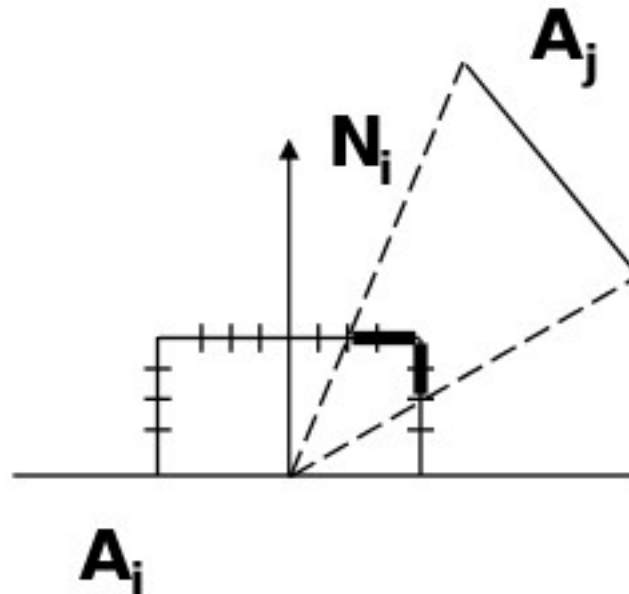
- **Método do Hemicubo:** simplificação de Cohen e Greenberg:
 - Em vez de usar a projeção num hemisfério, projeta na parte superior de um cubo centrado em dA_i , sendo a parte superior do cubo paralela com a superfície.
 - Cada face do hemicubo é dividida num conjunto de células quadradas de igual dimensão (ex: 50 por 50)



Método do Hemicubo

- São calculados fatores de forma elementares para cada célula do hemicubo, F_q para o quadrado q .
- O fator de forma F_{i-j} é então obtido somando todas as contribuições dos quadrados cobertos pelo *patch* j .

$$F_{i-j} = \sum F_q$$



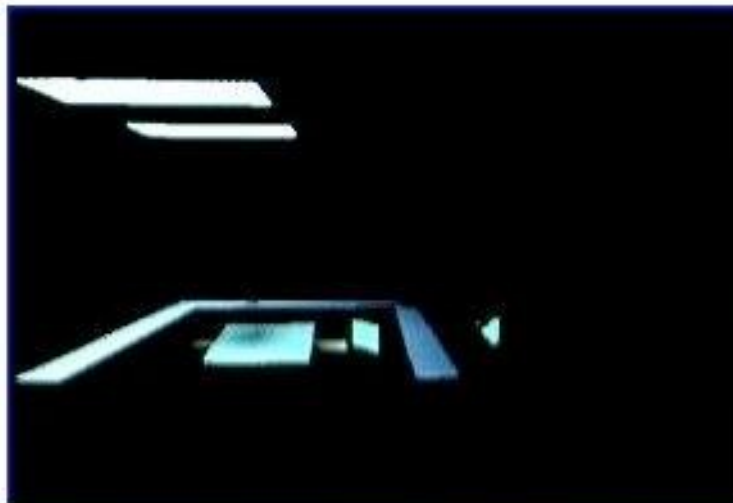
Progressive Refinement Radiosity (Refinamento Progressivo)

- Para resolução do sistema de equações lineares
- Usa métodos iterativos com convergência para a solução final
- Aproveitamento dos resultados intermédios como sendo "provisórios"
- Imagem é apresentada desde o início dos cálculos
- Qualidade dos resultados vai melhorando com o tempo de processamento

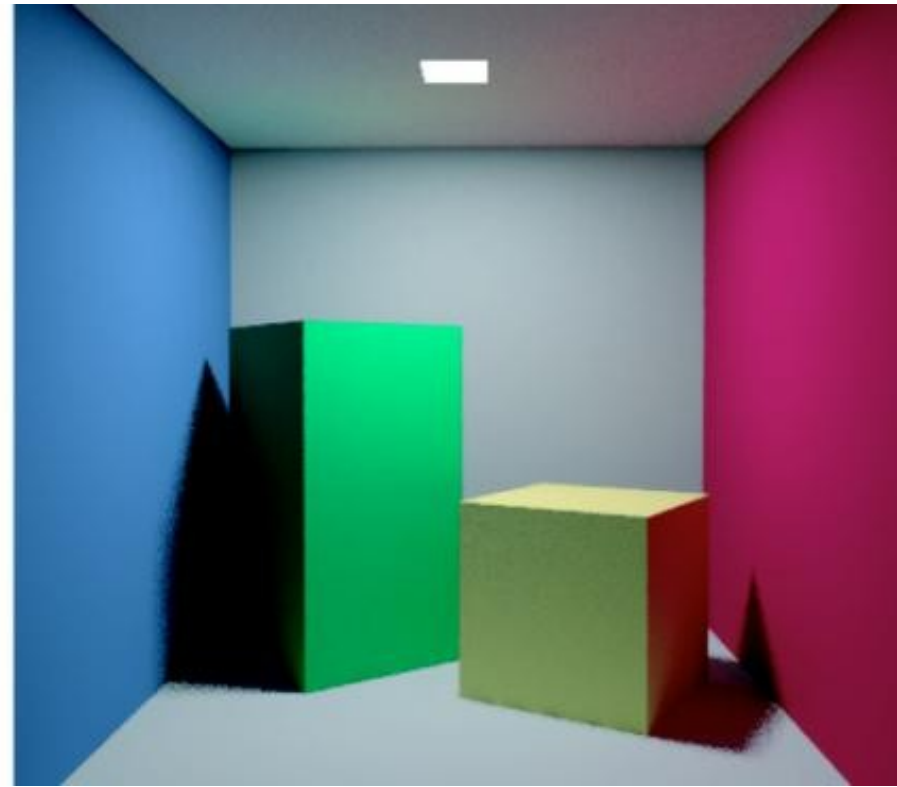
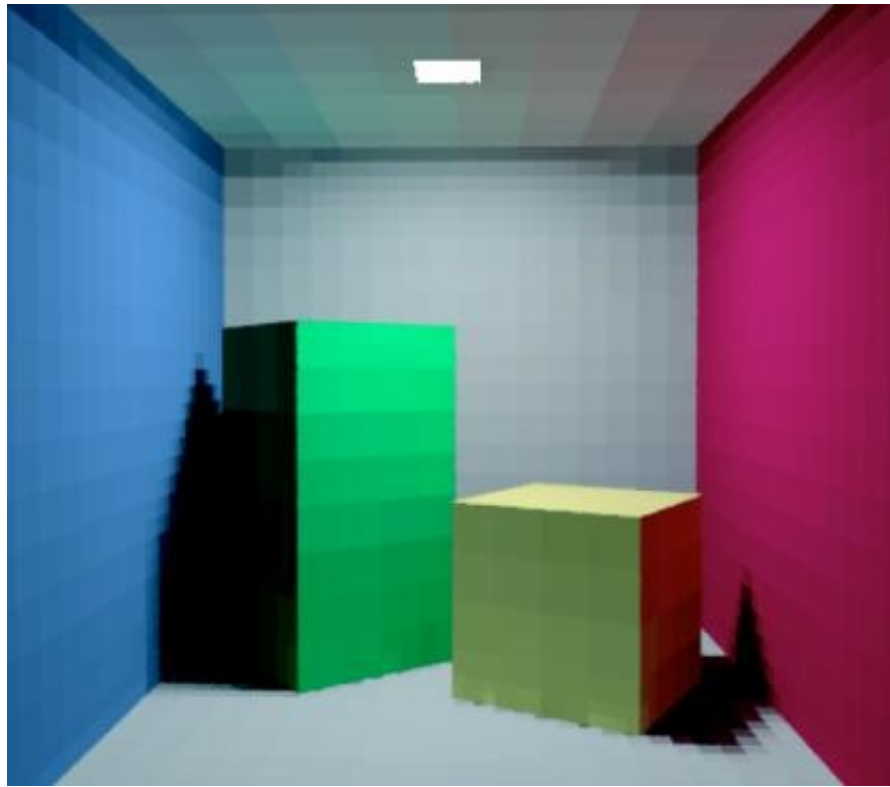
Refinamento Progressivo

- Algoritmo:
 - Um *patch* é escolhido por vez para disparar luz e calcula-se como esta luz se espalha na cena
 - Dispara-se primeiro os *patches* que influenciam mais a cena (emissores de luz, energia*área do *patch*)
 - Inicialmente $\mathbf{B}_i = \mathbf{E}_i$, para todo *patch*

Refinamento Progressivo



Radiosidade nos Vértices



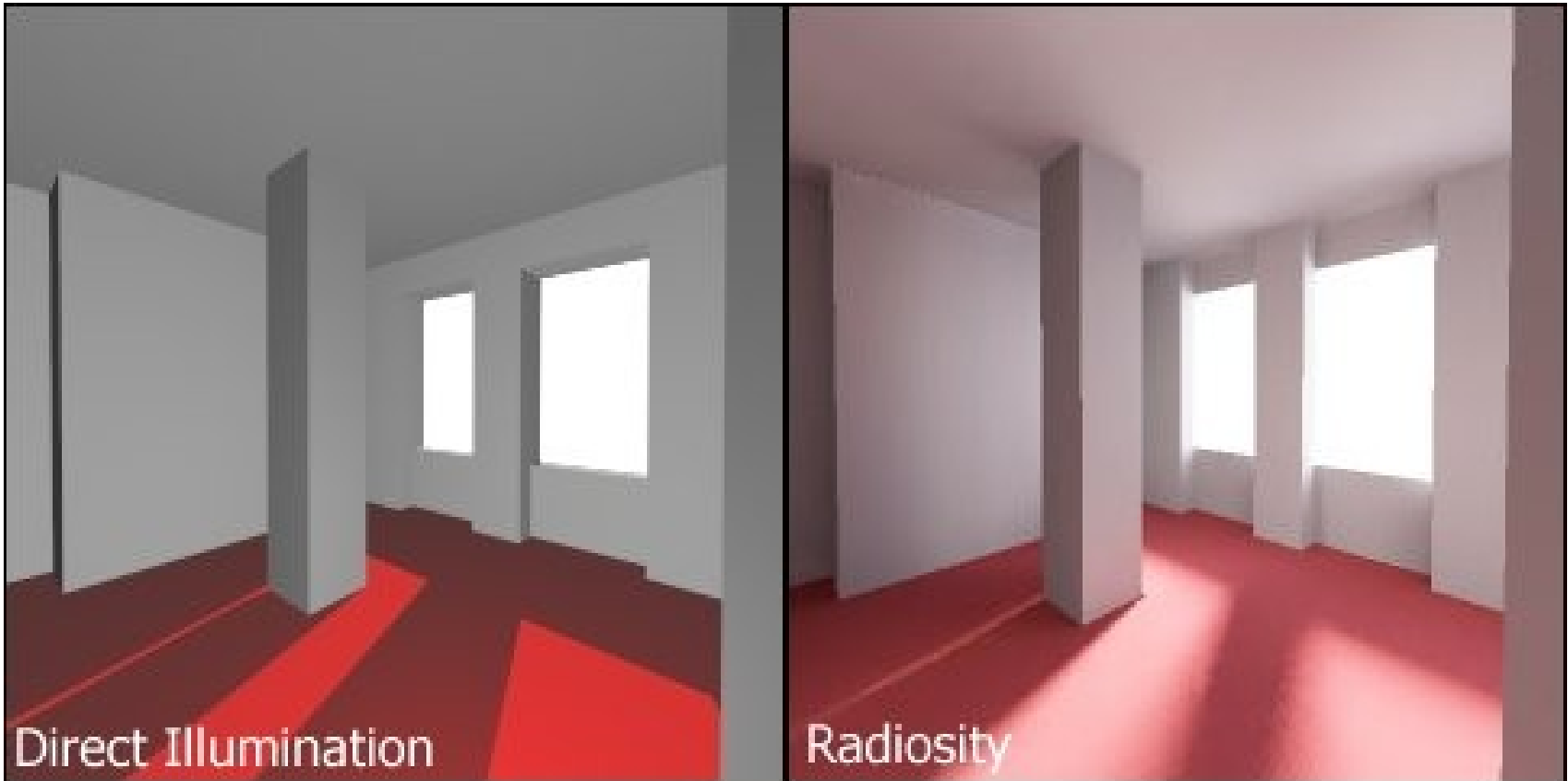
Criação da Imagem

1. Obtém-se a radiação para cada *patch* resolvendo o sistema de equações por eliminação Gaussiana.
2. Definir a posição do observador.
3. Aplicar um algoritmo de visibilidade, por exemplo, *z-buffer*.
4. Calcular a radiação dos vértices de cada polígono.
5. Aplicar a interpolação de cor (Gouraud).

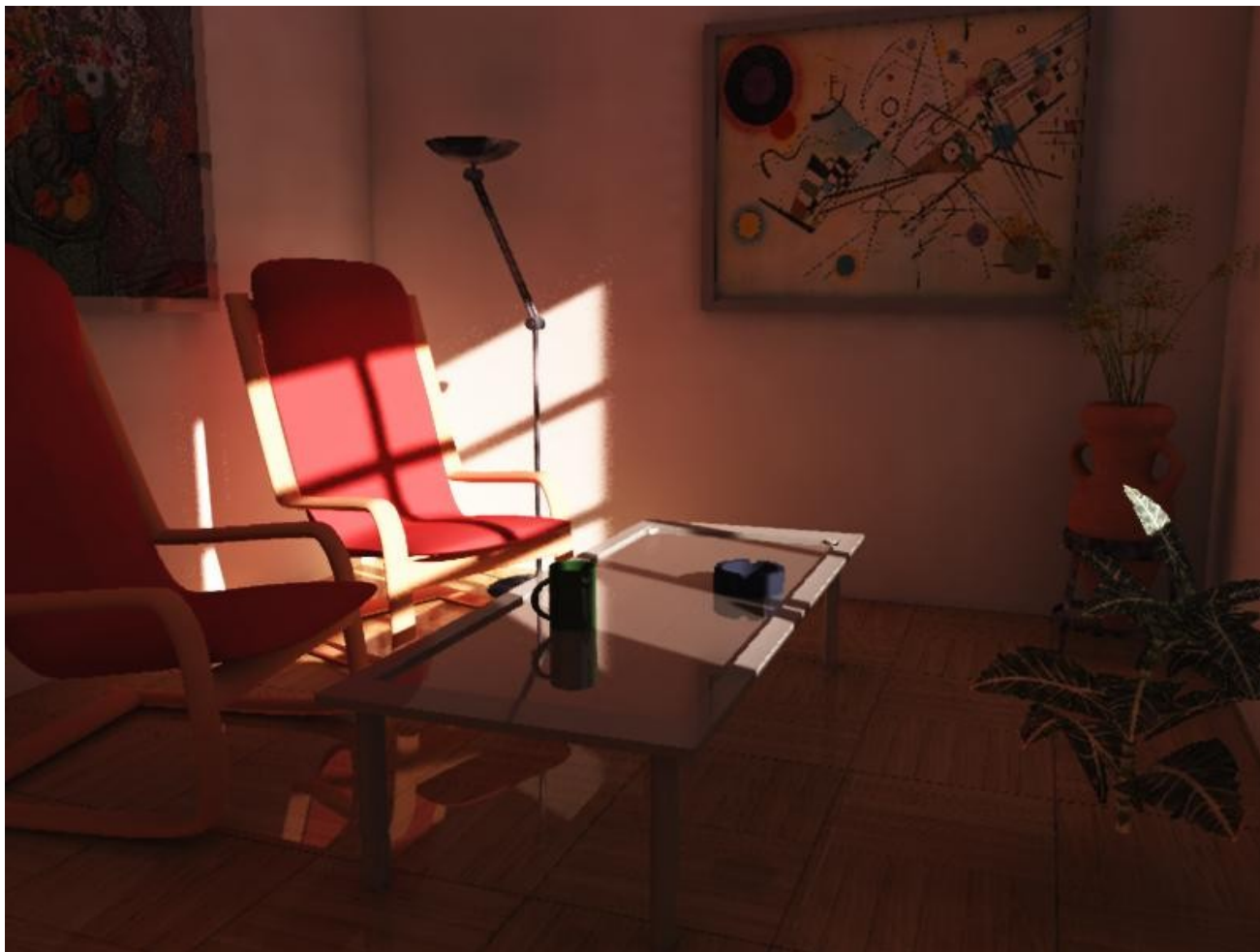
Ray-tracing + Radiosity

- Exploração do que cada um processa melhor:
- Ray Tracing: reflexão especular
- Radiosidade: reflexão difusa

Exemplo



Exemplo



Exemplo



Exemplo



Problema do Ray Tracing

1. Má distribuição da carga de trabalho: o número de raios cresce exponencialmente, e apesar deste aumento, o resultado incremental se torna menos importante para o cálculo de iluminação.
2. Casos complicados:
 - Cáusticas
 - *Color bleeding*

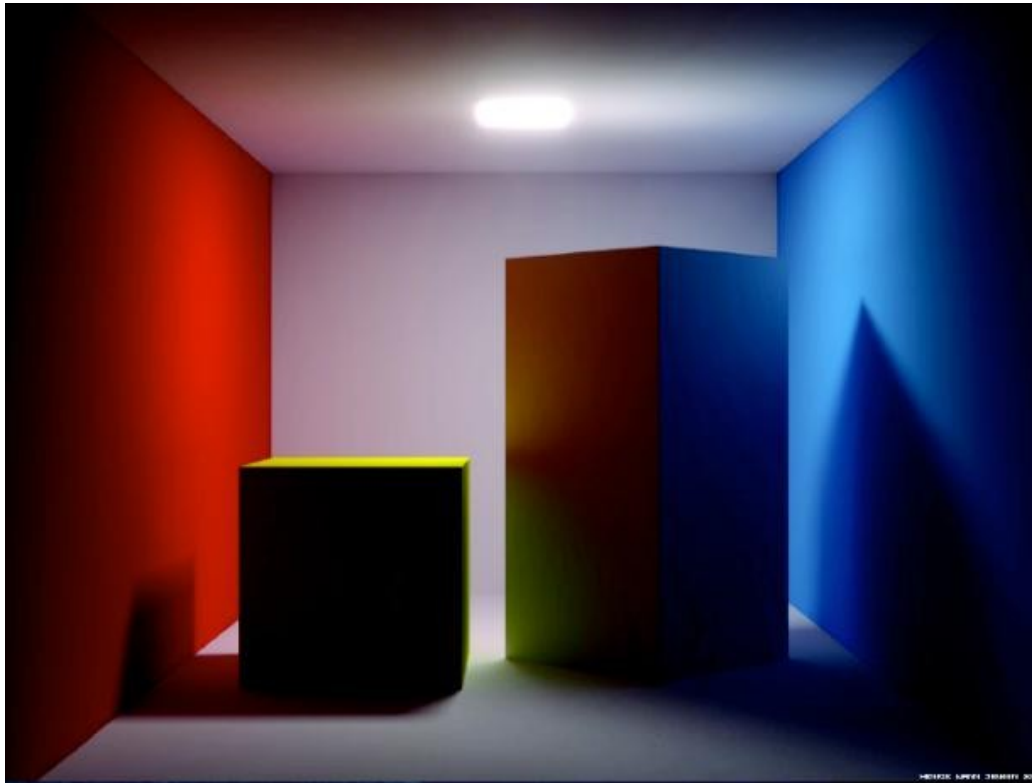
Cáusticas

- Luz foca através de uma superfície especular numa superfície difusa
- **Qual direção** devem ser os raios secundários enviados para detectar as cáusticas?



Color Bleeding

- Cor de uma superfície difusa refletida em outra superfície difusa
- **Em que direção** emitir os raios secundários?



Path Tracing

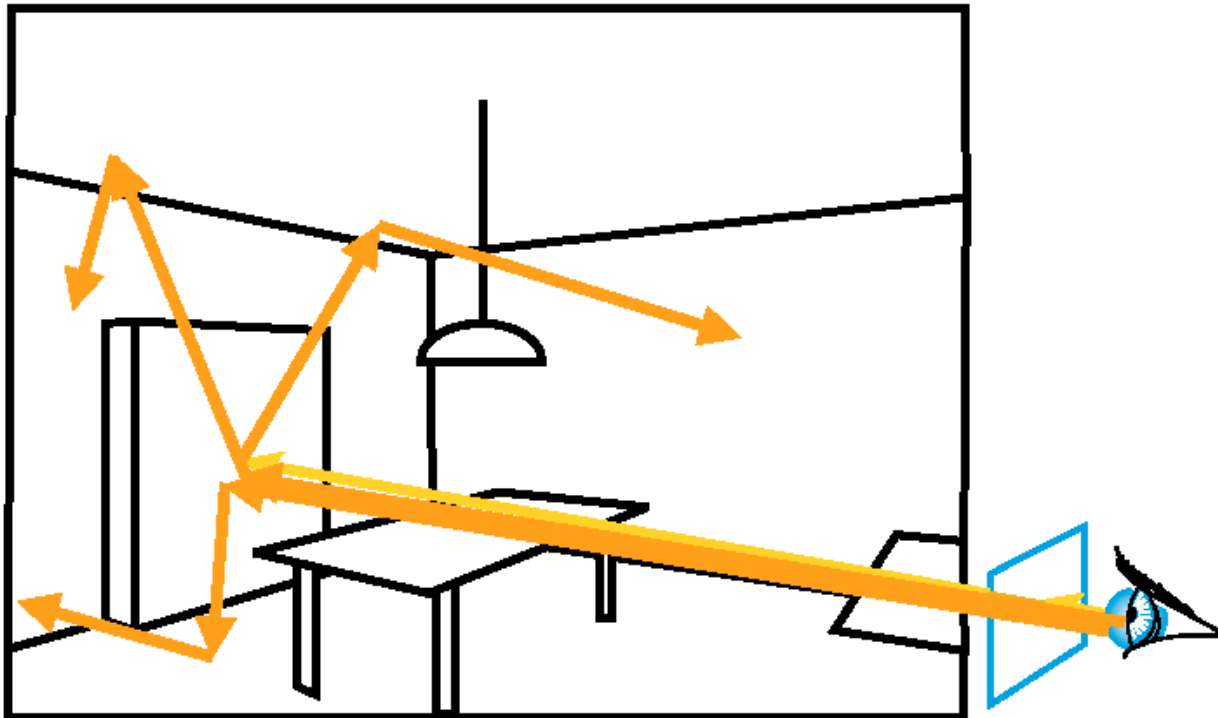
- Uma extensão de Ray Tracing (SIGGRAPH: Kajiya, 86) para tentar solucionar os problemas com cáusticas e *color bleeding*.
- Reflexão difusa gera número **infinito** de raios
 - Selecionar raios aleatoriamente

Monte Carlo Path Tracing

- Path Tracing: a distribuição de luz é amostrada através do envio de raios aleatórios ao longo de todos os caminhos de iluminação possíveis
- Uma abordagem Monte Carlo para o problema de iluminação global
- A média de várias amostras provê uma estimativa da luz total que chega no pixel

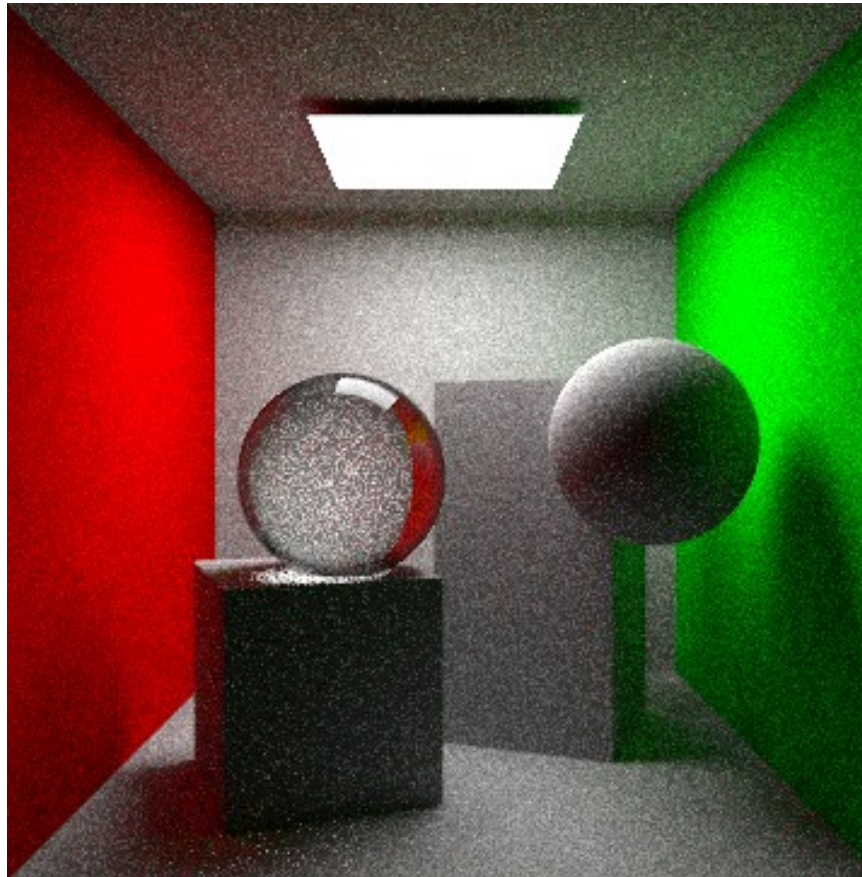
Monte Carlo Path Tracing

- Envia apenas um raio secundário por recursão
- Mas envia muitos raios primários por *pixels*



Monte Carlo Path Tracing

- Poucas amostras, muito ruído



Algoritmo Monte Carlo Path Tracing

- Envia o raio e encontra intersecção
- Do ponto de intersecção enviar
 - Um raio para cada fonte de luz
 - Um raio adicional (critério para este raio mais adiante)
 - Reflexão difusa, OU
 - Reflexão especular, OU
 - Um raio transmitido
- Produz um ray "path"
(em vez de árvore de raios)

Estratégia de Amostragem

- Como escolher a distribuição dos raios aleatórios
- Possibilidades:
 - Amostragem por regiões (*stratified sampling*)
 - Divide as direções possíveis em sub-regiões e envia uma amostra por subregião
 - *Importance sampling*
 - Amostragem de acordo com a BRDF
 - Uniforme
 - Amostragem de acordo com parâmetros do objeto

Escolhendo o tipo de raio

- Como selecionamos qual raio enviar?
- Cada material tem um k_d , k_s , e k_t

Seja $k_{tot} = k_d + k_s + k_t$

Selecionar um número aleatório R no intervalo $(0, k_{tot})$

Se $(R < k_d)$ então dispara difuso

Senão se $(R < k_d + k_s)$ dispara especular

Senão dispara transmitido

Escolhendo o tipo de raio - exemplo

- Sejam $k_d=0.5$, $k_s=0.3$ e $k_t=0.2$
 - Valores aleatórios até 0.5 envia raio difuso
 - Valores aleatórios maiores do 0.5 e menores do que $0.5+0.3$ envia raio especular
 - Valores maiores do que 0.8 envia raio transmitido

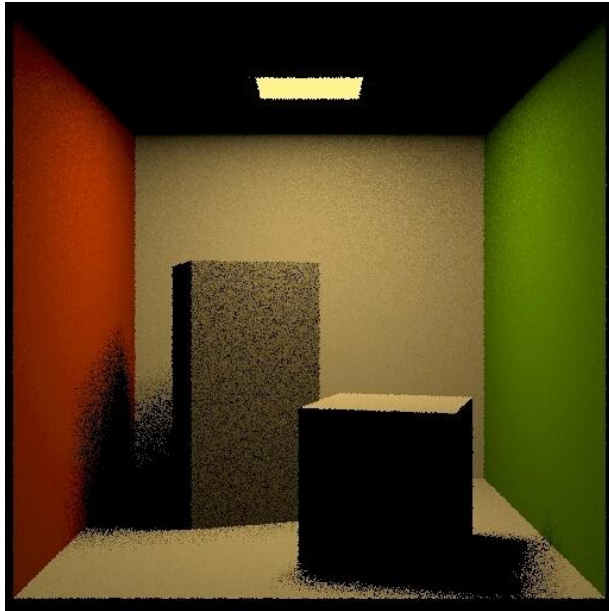
Cálculo da reflexão difusa

- Podemos calcular uma direção aleatória com o seguinte:
- Dados dois números aleatórios ξ_1 em $[0,1]$ e ξ_2 em $[0,1]$, a direção ω_d refletida aleatoriamente é dada por
$$\omega_d = (\theta, \varphi) = (\cos^{-1}(\sqrt{\xi_1}), 2\pi\xi_2)$$

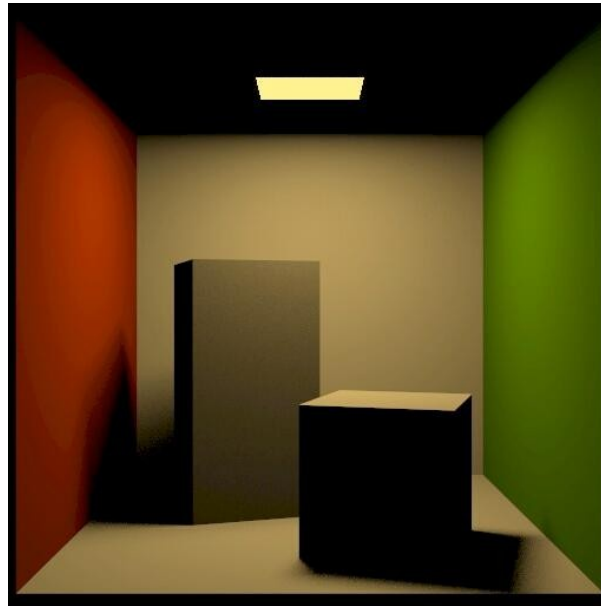
Path Tracing

- Vantagens:
 - Simula iluminação global
 - Não desperdiça tempo em coisas não visíveis
 - Custo constante e proporcional ao número de *paths*
- Problema:
 - Necessário traçar muitos raios para conseguir uma imagem boa
 - Tipicamente 100 – 1000 raios por pixel

Path Tracing



1 path/pixel



100 paths/pixel



200 paths/pixel

Path Tracing

