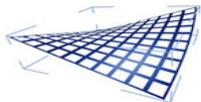

Iluminação e FotoRealismo: Ray Tracing Clássico

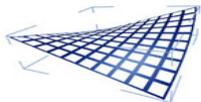
Luís Paulo Peixoto dos Santos

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



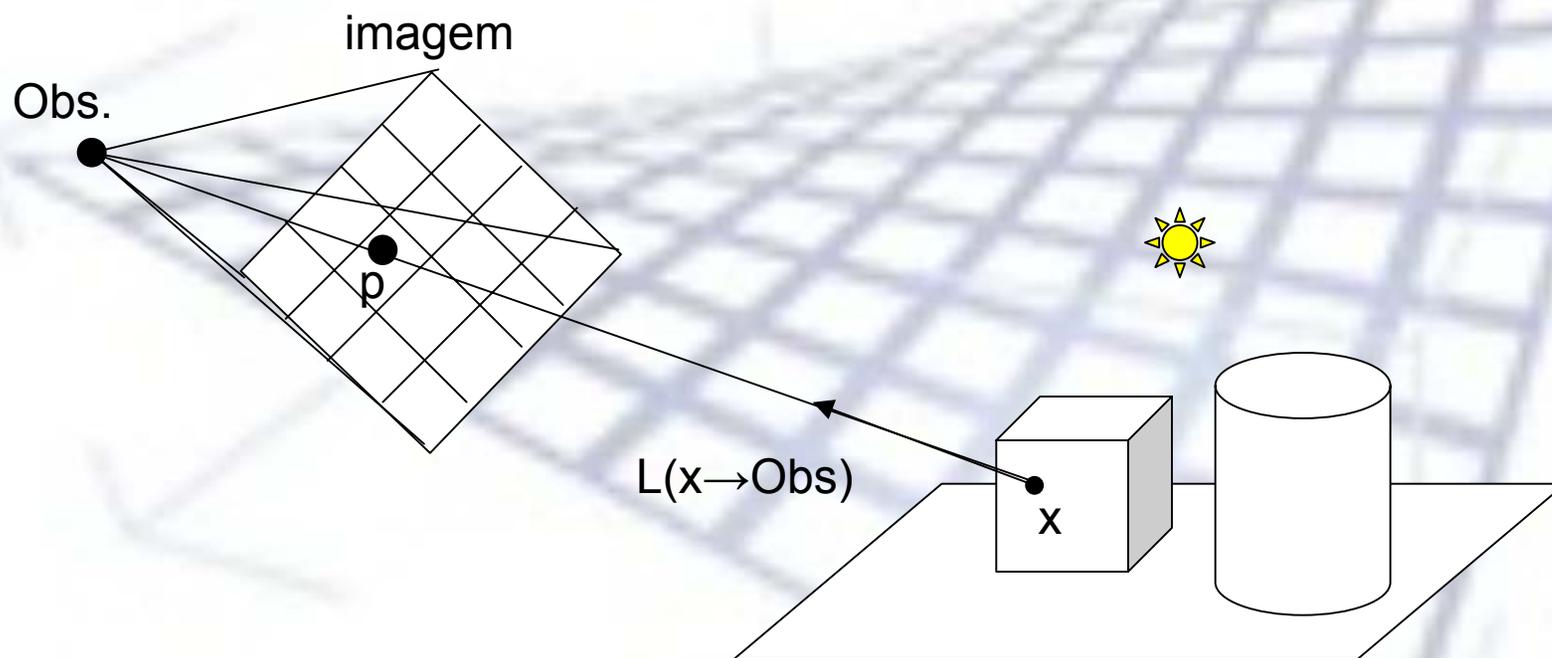
Introdução

- *Ray tracing* foi introduzido na Computação Gráfica em 1980 por Whitted
- Algoritmo de iluminação global recursivo, dependente da posição do observador, baseado na “recolha” de radiância a partir do observador na direcção das fontes de luz
- Ideal para simular fenómenos especulares (reflexão, transmissão)

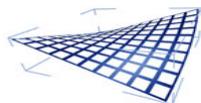


Configuração

- Orientado ao pixel: cálculo da radiância para cada ponto p do plano da imagem



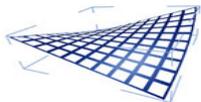
$$L(p \rightarrow Obs) = L(x \rightarrow Obs)$$



Ray Tracing - Algoritmo

```
// ciclo principal
computeImage (viewPoint) {
  para cada pixel p {
    raio = GerarRaio (viewPoint, p, PRIMARIO)
    radiance[p] = rad (raio)
  }
}

rad (raio) {
  objecto, x = trace (raio)
  computeRad (x, raio, objecto)
}
```

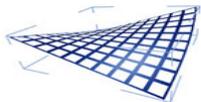


Ray Tracing - Algoritmo

```
// intersecção mais próxima da origem do raio

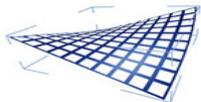
trace (raio) {           // p = r (origem, direcção)
  tmin = Max_dist
  Para todos os objectos da cena {
    x = intersect (raio, objecto)
    dist = distancia (raio.origem, x)

    if (dist < tmin) {
      tmin = dist
      p = x
      obj = objecto    }
  }
  return (obj, p)
}
```



Complexidade

- Os raios primários detectam quais os objectos que são projectados no plano da imagem
- O cálculo de $r(x, \Psi)$ é a operação que consome mais tempo de cálculo neste algoritmo
- Na abordagem apresentada a sua complexidade é linear com o número de objectos: $O(N_{obj})$
- São várias as técnicas de aceleração desenvolvidas para diminuir o tempo associado ao cálculo de $r(x, \Psi)$

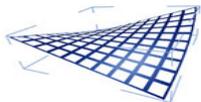


Ray Tracing – Direcções a considerar

- Uma vez encontrado o ponto x , que projecta directamente no pixel p , é necessário proceder à sua iluminação (*shading*), isto é, ao cálculo de

$$L(x \rightarrow \Theta) = L_e(x \rightarrow \Theta) + \int_{\Omega_s} f_r(x, \Psi \leftrightarrow \Theta) L(x \leftarrow \Psi) \cos(\vec{N}_x, \Psi) \partial\omega_\Psi$$

- Sabendo que não é possível considerar todas as direcções da hemisfera Ω_s , quais deverão ser escolhidas para incluir no cálculo deste radiância?



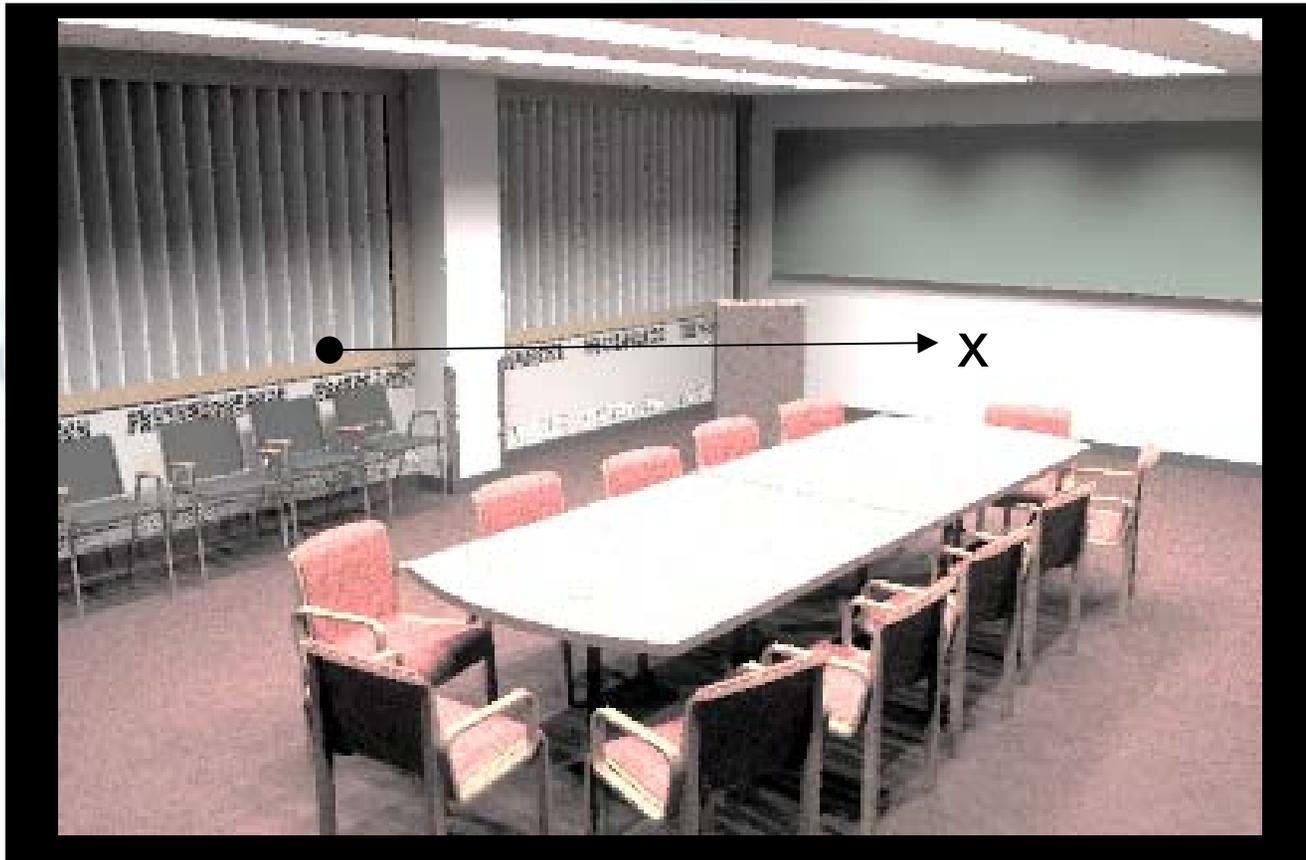
Ray Tracing – Direcções a considerar

- Devem ser escolhidas as direcções que, com maior probabilidade, maiores contribuições dão para o valor final de $L(x \rightarrow \Theta)$

$$L(x \rightarrow \Theta) = L_e(x \rightarrow \Theta) + \int_{\Omega_s} f_r(x, \Psi \leftrightarrow \Theta) L(x \leftarrow \Psi) \cos(\vec{N}_x, \Psi) \partial\omega_\Psi$$

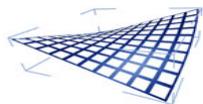
- Quais são essas direcções?
 - Direcções Ψ que maximizam a BRDF, isto é, aquelas para as quais é maior a radiância reflectida na direcção Θ
 - Direcções Ψ que maximizam $L(x \leftarrow \Psi)$, isto é, direcções de maior radiância incidente
 - Direcções Ψ que maximizam o cosseno

Ray Tracing – Direcções a considerar



Modelo por:
Anat Grynberg
Greg Ward

Ponto x determinado por um raio primário. Quais as direcções a considerar para fazer o *shading* de x , isto é, calcular $L(x \rightarrow \Theta)$?



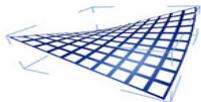
Ray Tracing – Direcções a considerar



Radiâncias incidentes em x ao longo de toda a hemisfera Ω_s .

Parece claro que a radiância incidente é maximizada na direcção das fontes de luz.

Conclusão: as direcções da hemisfera correspondentes a fontes de luz devem ser amostradas, pois a radiância incidente em x ao longo destas direcções é, com grande probabilidade, elevada.



Iluminação Directa e Indirecta

- A radiância reflectida pode ser dividida em 2 integrais, um correspondente a iluminação directa oriunda das fontes de luz e outro correspondente à iluminação indirecta, oriunda de todas as outras direcções.

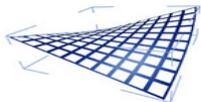
$$L(x \leftarrow \Psi) = L(r(x, \Psi) \rightarrow -\Psi)$$

$$L(r(x, \Psi) \rightarrow -\Psi) = L_e(r(x, \Psi) \rightarrow -\Psi) + L_r(r(x, \Psi) \rightarrow -\Psi)$$

$$L_r(x \rightarrow \Theta) = \int_{\Omega_s} f_r(x, \Theta \leftrightarrow \Psi) L_e(r(x, \Psi) \rightarrow -\Psi) \cos(\vec{N}_x, \Psi) \partial \omega_\Psi$$

$$+ \int_{\Omega_s} f_r(x, \Theta \leftrightarrow \Psi) L_r(r(x, \Psi) \rightarrow -\Psi) \cos(\vec{N}_x, \Psi) \partial \omega_\Psi$$

$$L_r(x \rightarrow \Theta) = L_{directa}(x \rightarrow \Theta) + L_{indirecta}(x \rightarrow \Theta)$$



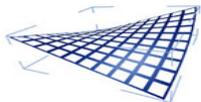
Iluminação Directa

- A iluminação directa é mais conveniente se expressa em termos de integração sobre a área.

Sejam:

- $y = r(x, \Psi)$
- L o vector que define a direcção de x para y , isto é, o vector na direcção da fonte de luz

$$L_{directa}(x \rightarrow \Theta) = \int_{A_{fontes}} f_r(x, \Theta \leftrightarrow \vec{L}) L_e(y \rightarrow -\vec{L}) V(x, y) G(x, y) \partial A_y$$



Iluminação Directa

- A maioria dos ray tracers clássicos permite aproximar as fontes de luz como fontes ideais: pontuais e que emitem com igual intensidade em todas as direcções.
- A BRDF é representada pelo modelo de iluminação local. Este deve ser convenientemente escolhido, pois um modelo fisicamente incorrecto levará a resultados incorrectos

Whitted sugere que o modelo de iluminação local seja o de Lambert, com um coeficiente de reflexão difusa directa para cada um dos canais: kd_R , kd_G , kd_B .

O modelo de Phong permite adicionar um reflexo especular à iluminação directa, mas não é fisicamente correcto.

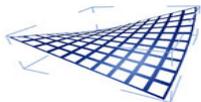
Iluminação Directa

- Lambert para / fontes de luz:

$$L_{directa,\lambda}(x \rightarrow \Theta) = \sum_l L_{e,\lambda}(y_l \rightarrow -\vec{L}_l) V(x, y_l) \frac{\cos(\vec{N}_x, \vec{L}_l)}{r_{x,y_l}^2} k_{d,\lambda}$$

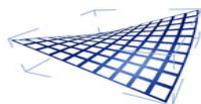
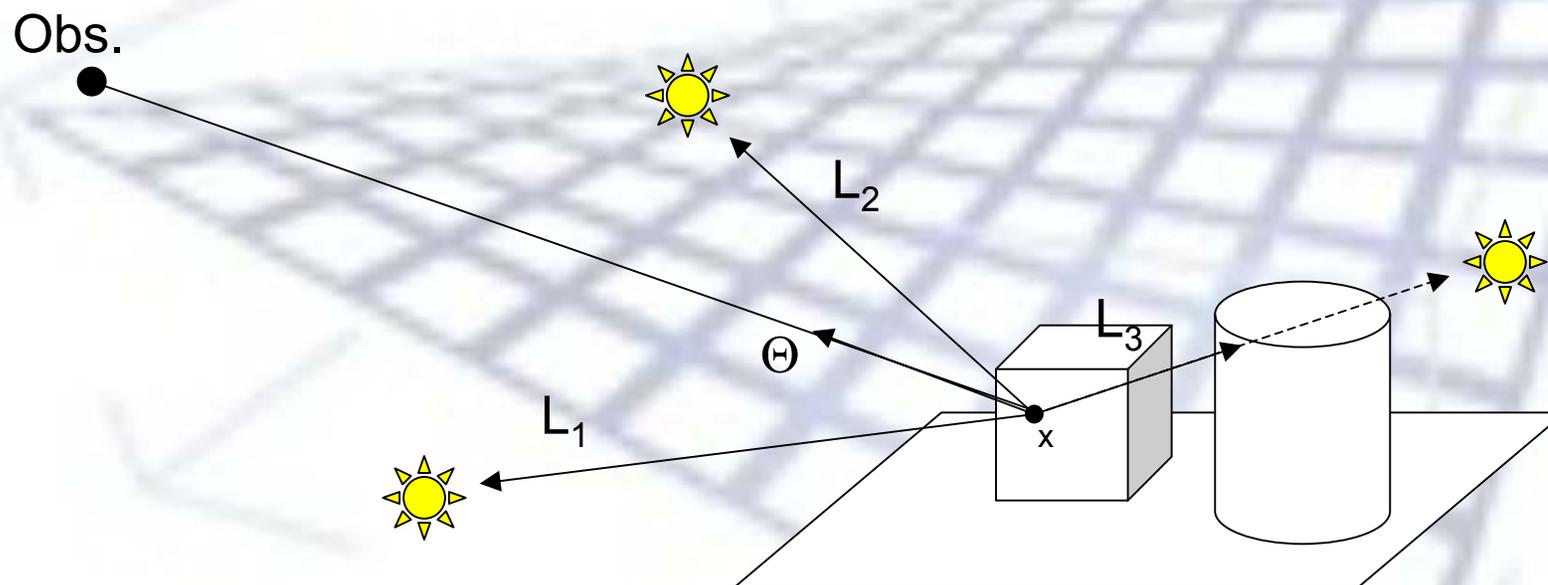
- Phong para / fontes de luz:

$$L_{directa,\lambda}(x \rightarrow \Theta) = \sum_l L_{e,\lambda}(y_l \rightarrow -\vec{L}_l) V(x, y_l) \frac{\cos(\vec{N}_x, \vec{L}_l)}{r_{x,y_l}^2} (k_{d,\lambda} + k_s \cos^{n_s}(\Theta, \vec{R}))$$



Shadow Rays

- Para cada intersecção um *ray tracer* deve enviar um raio (shadow ray) para cada fonte de luz. Se o ponto de intersecção e a fonte de luz forem mutuamente visíveis então procede-se ao *shading*

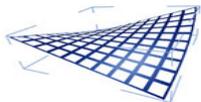


Ray Tracing - algoritmo

```
computeRad (x, raio, objecto) {
    radiance = directIllum (x, raio.dir, objecto)
    return (radiance)
}

directIllum (x, dir, objecto) {
    rad = 0;
    para cada fonte de luz l {
        raio = GerarRaio (x, l, SHADOW)
        if (visibilidade (raio, l))
            rad += shade (x, dir, l, objecto)
    }
    return (rad)
}

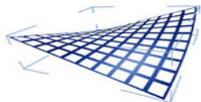
// shade() depende do modelo de iluminação local
```



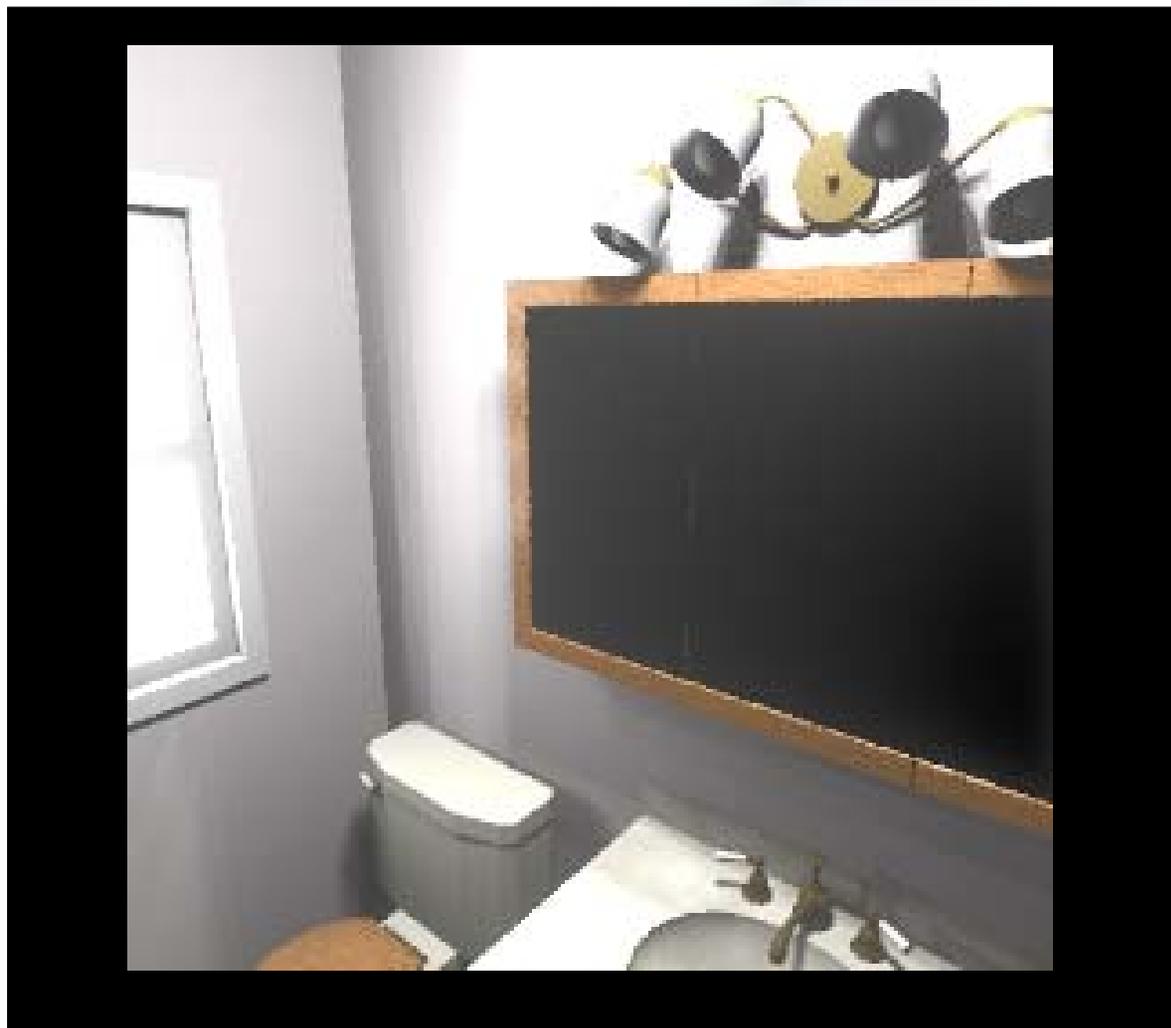
Ray Tracing - Algoritmo

```
// visibilidade da fonte de luz

visibilidade (raio,l) { // V(x,y)
  tmin = distancia (raio.origem,l)
  Para todos os objectos da cena {
    p = intersect (raio, objecto)
    dist = distancia (raio.origem, p)
    if (dist < tmin)
      return (0)
  }
  return 1
}
```



Ray Tracing – Iluminação Directa



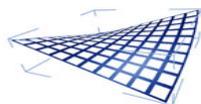
Iluminação Directa

Toda a radiância tem origem na janela.

O espelho não reflecte qualquer radiância.

Os metais das torneiras e candeeiros parecem difusos, pois o modelo de iluminação local não inclui *highlights*.

Modelo por:
Greg Ward



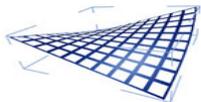
Iluminação Indirecta

$$L_{indirecta}(x \rightarrow \Theta) = \int_{\Omega_s} f_r(x, \Theta \leftrightarrow \Psi) L_r(r(x, \Psi) \rightarrow -\Psi) \cos(\vec{N}_x, \Psi) \partial\omega_\Psi$$

Quais as direcções da hemisfera a considerar para a iluminação indirecta? Tipicamente aquelas que correspondem aos máximos da BRDF.

Os ray tracers convencionais processam as direcções especulares:

- reflexão especular
- transmissão especular



Direcções Reflexão e Transmissão

- Reflexão especular

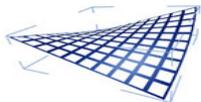
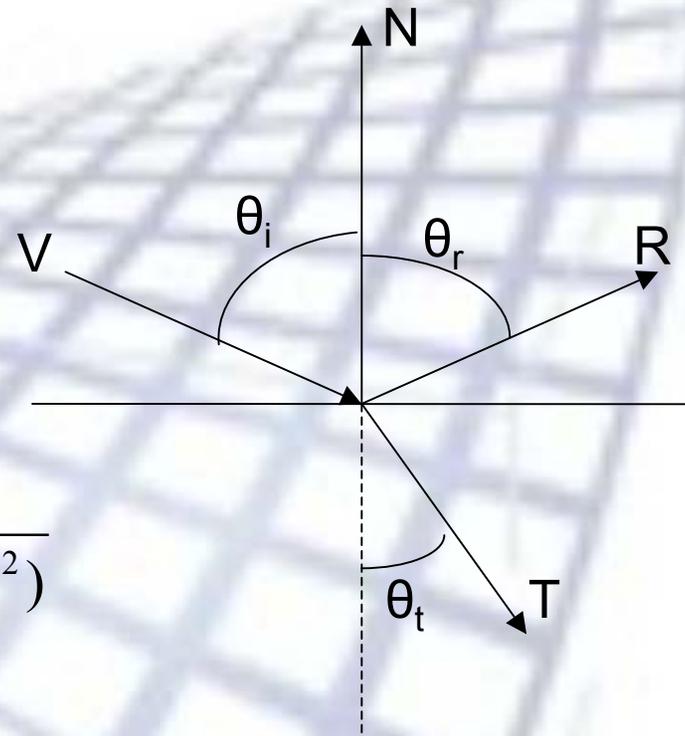
$$\vec{R} = 2\vec{N}(\vec{N} \cdot \vec{V}) - \vec{V}$$

- Transmissão especular
(lei de Snell)

$$\eta = \frac{\eta_t}{\eta_i} = \frac{\sin \theta_i}{\sin \theta_t}$$

$$\cos \theta_t = \sqrt{1 - \sin^2 \theta_t} = \sqrt{1 - \eta^{-2}(1 - (\vec{V} \cdot \vec{N})^2)}$$

$$\vec{T} = \frac{\vec{V}}{\eta} - (\cos \theta_t - \frac{(\vec{V} \cdot \vec{N})}{\eta})\vec{N}$$



Iluminação Indirecta

- Nos algoritmos de ray tracing clássico

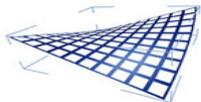
$$L_{indirecta}(x \rightarrow \Theta) = \int_{\Omega_s} f_r(x, \Theta \leftrightarrow \Psi) L_r(r(x, \Psi) \rightarrow -\Psi) \cos(\vec{N}_x, \Psi) \partial\omega_\Psi$$

É aproximada apenas pelas direcções \vec{R}_g e \vec{T}_g .

A BRDF é aproximada pelas constantes:

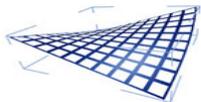
- $k_{sg}(\lambda)$ – coeficiente de reflexão especular global
- $k_{tg}(\lambda)$ – coeficiente de transmissão especular global

$$L_{indirecta,\lambda}(x \rightarrow \Theta) = k_{sg}(\lambda) L_{r,\lambda}(r(x, \vec{R}_g) \rightarrow -\vec{R}_g) \cos(\vec{N}_x, \vec{R}_g) \\ + k_{tg}(\lambda) L_{r,\lambda}(r(x, \vec{T}_g) \rightarrow -\vec{T}_g) \cos(\vec{N}_x, \vec{T}_g)$$

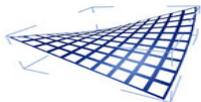
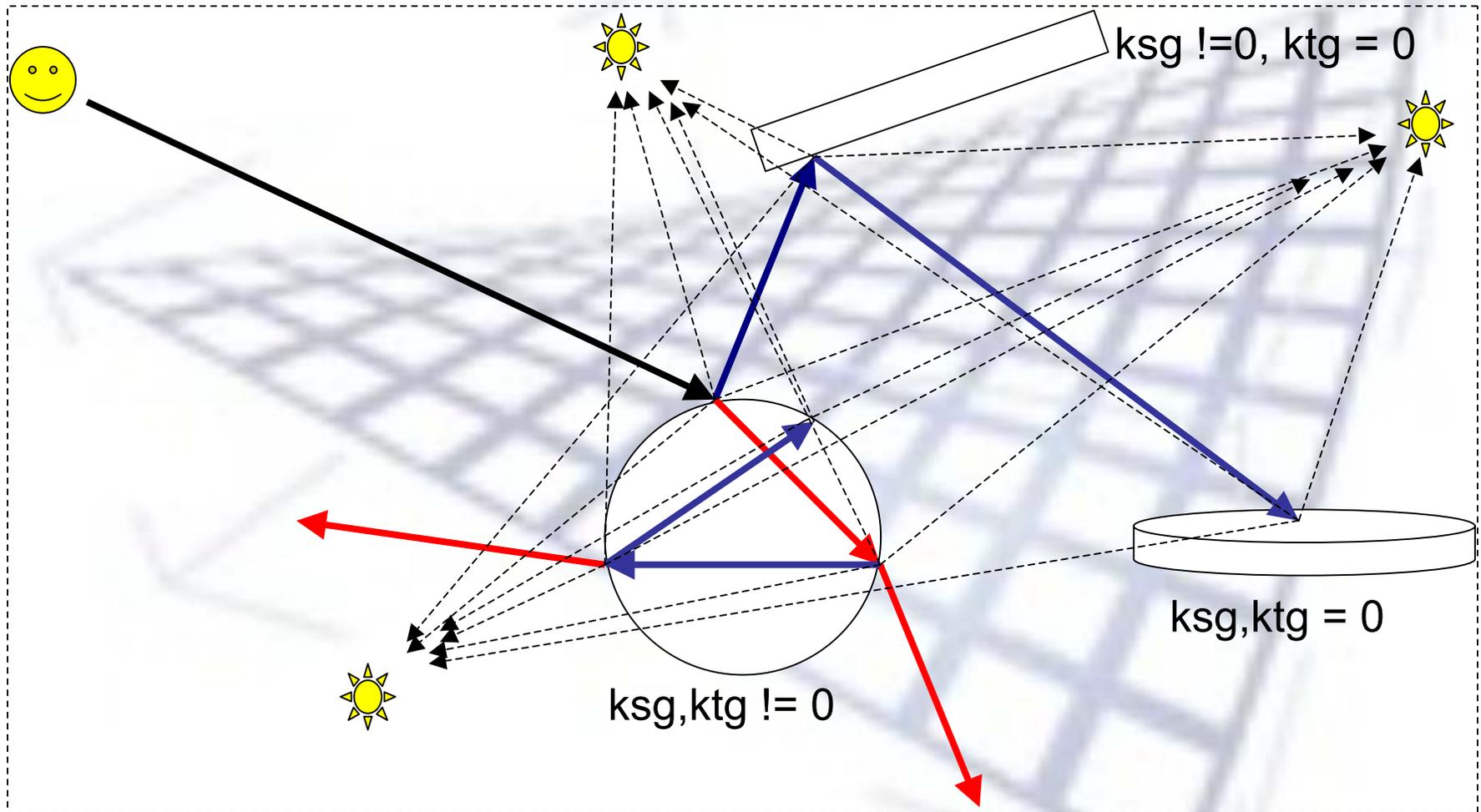


Raios Secundários

- Para calcular a radiância incidente em x ao longo de cada uma das direcções R_g e T_g devem ser enviados raios secundários ao longo de cada uma destas direcções.
- O processamento dos raios secundários é em tudo equivalente aos raios primários, fazendo do *ray tracing* um algoritmo recursivo que gera uma árvore de raios.
- O processo de geração de novos raios deve terminar se o material não for especular ou quando a árvore atingir uma profundidade limite.

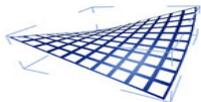
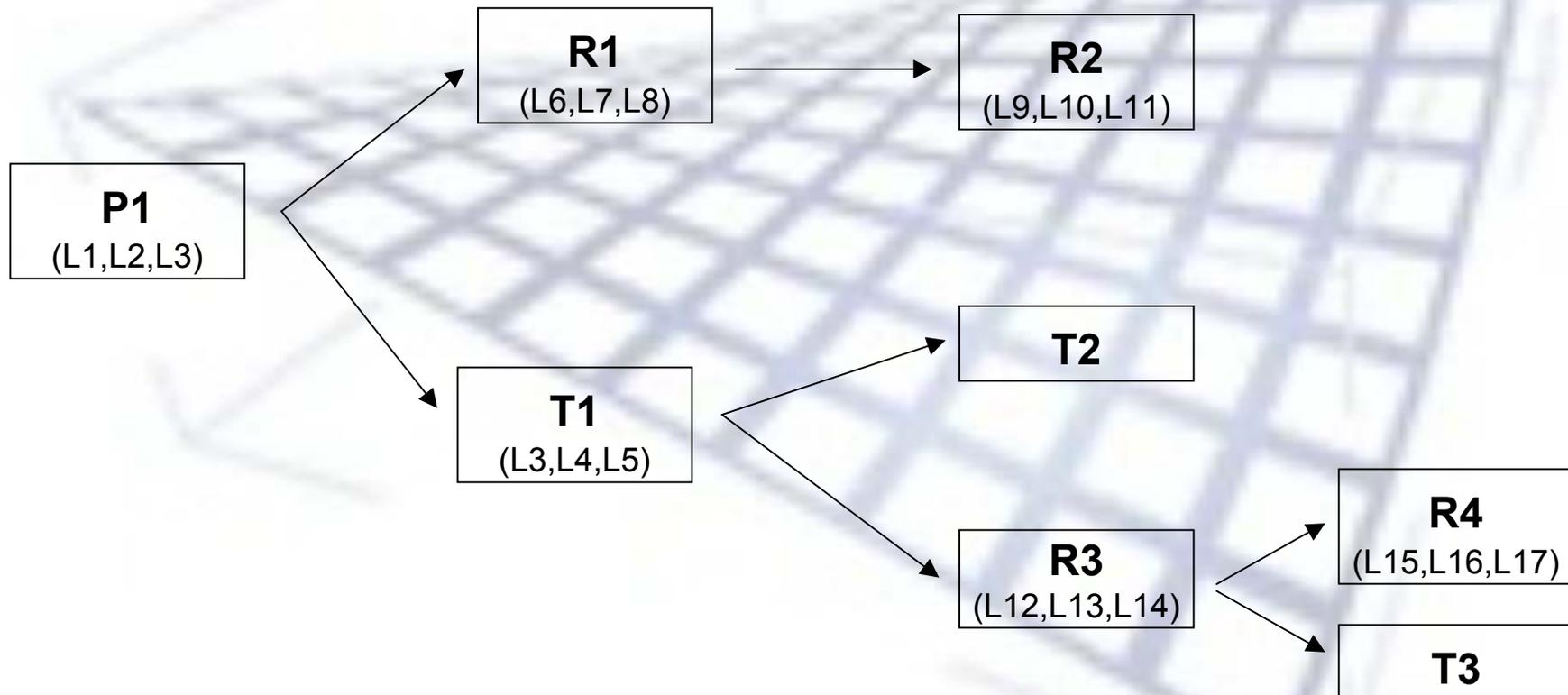


Raios Secundários



Raios Secundários

- Árvore de raios para esquema anterior



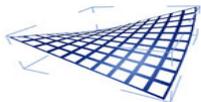
Ray Tracing - Algoritmo

```
computeRad (x, raio, objecto) {
    rad = directIllum (x, raio.dir, objecto)

    if (depth < MAX_DEPTH) {
        if (ksg > 0) { // reflexão especular
            raioR = GerarRaio (x, Rg, REFLEXAO)
            objR, p = trace (raioR)
            rad += ksg*cos(N,raioR)*computeRad (p,raioR,objR) }

        if (ktg > 0) { // transmissão especular
            raioT = GerarRaio (x, Tg, TRANSMISSAO)
            objT, p = trace (raioT)
            rad += ktg*cos(N,raioT)*computeRad (p,raioT,objT) }

    } return (rad) }
```



Iluminação Directa e Indirecta



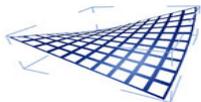
Iluminação Directa e Especular

Toda a radiância tem origem na janela.

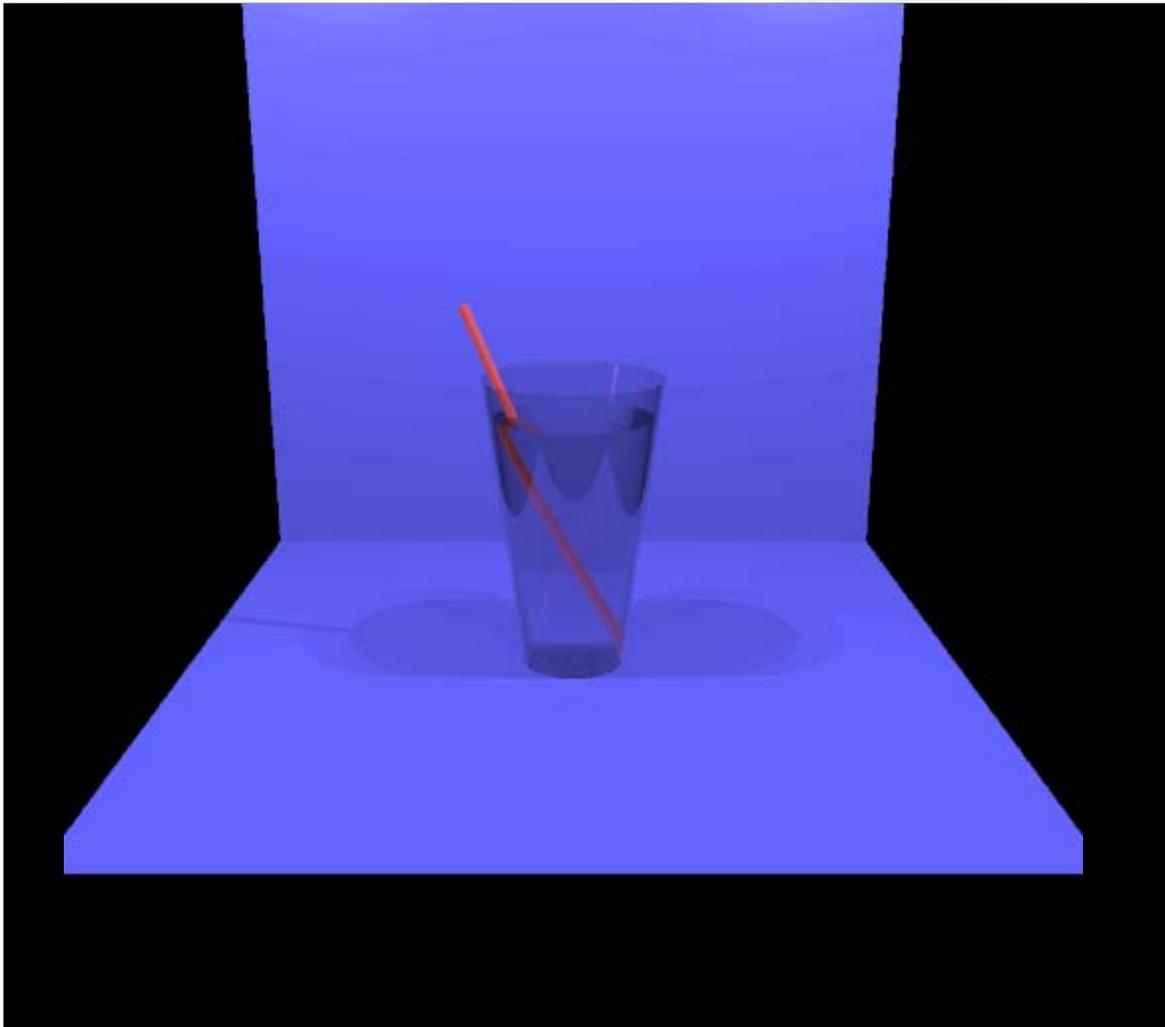
O espelho reflecte objectos que não projectam directamente no plano de imagem..

Os metais das torneiras e candeeiros adquirem um aspecto especular

Modelo por:
Greg Ward



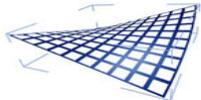
Iluminação Directa e Indirecta



Iluminação Directa e Especular

As fontes de luz estão suspensas à esquerda e à direita da cena

A imagem da palha é refractada devido aos diferentes índices de refractividade do vidro e da água.



Reflexões Difusas

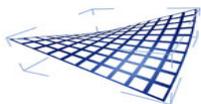
$$L(x \rightarrow \Theta) = L_e(x \rightarrow \Theta) + L_{directa}(x \rightarrow \Theta) + L_{indirecta}(x \rightarrow \Theta)$$

$$L_{directa,\lambda}(x \rightarrow \Theta) = \sum_l L_{e,\lambda}(y_l \rightarrow -\vec{L}_l) V(x, y_l) \frac{\cos(\vec{N}_x, \vec{L}_l)}{r_{x,y_l}^2} (k_{d,\lambda} + k_s \cos^{n_s}(\Theta, \vec{R}))$$

$$L_{indirecta,\lambda}(x \rightarrow \Theta) = k_{sg}(\lambda) L_{r,\lambda}(r(x, \vec{R}_g) \rightarrow -\vec{R}_g) \cos(\vec{N}_x, \vec{R}_g) \\ + k_{tg}(\lambda) L_{r,\lambda}(r(x, \vec{T}_g) \rightarrow -\vec{T}_g) \cos(\vec{N}_x, \vec{T}_g)$$

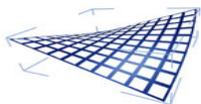
Apenas as direcções de reflexão e transmissão especulares são incluídas no *ray tracing* clássico!

Como considerar as restantes direcções da hemisfera Ω_s ?



Reflexões Difusas

- As restantes direcções correspondem a interreflexões difusas.
- Uma vez que estas não têm um carácter direccionado, são necessários centenas de raios para aproximar este valor em cada ponto de intersecção
- A maior parte dos *ray tracers* aproximam estas interreflexões usando um termo designado por luz ambiente
- Este é considerado constante e não direccionado em toda a cena, resultando numa iluminação “plana”, fisicamente incorrecta.
- Alguns *ray tracers* aproximam as interreflexões difusas, usando métodos como: a cache de irradiância, radiosidade instantânea, etc.



Reflexões Difusas

- A irradiância ambiente é dada por

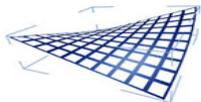
$$E_{amb,\lambda}$$

- O termo de reflexão ambiente para cada material é

$$k_{amb}(\lambda)$$

- Resultando numa reflexão na direcção Θ

$$L_{amb,\lambda}(x \rightarrow \Theta) = k_{amb}(\lambda) * E_{amb,\lambda}$$



Reflexões Difusas

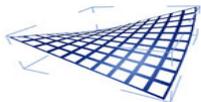


Iluminação Directa,
Especular e Ambiente

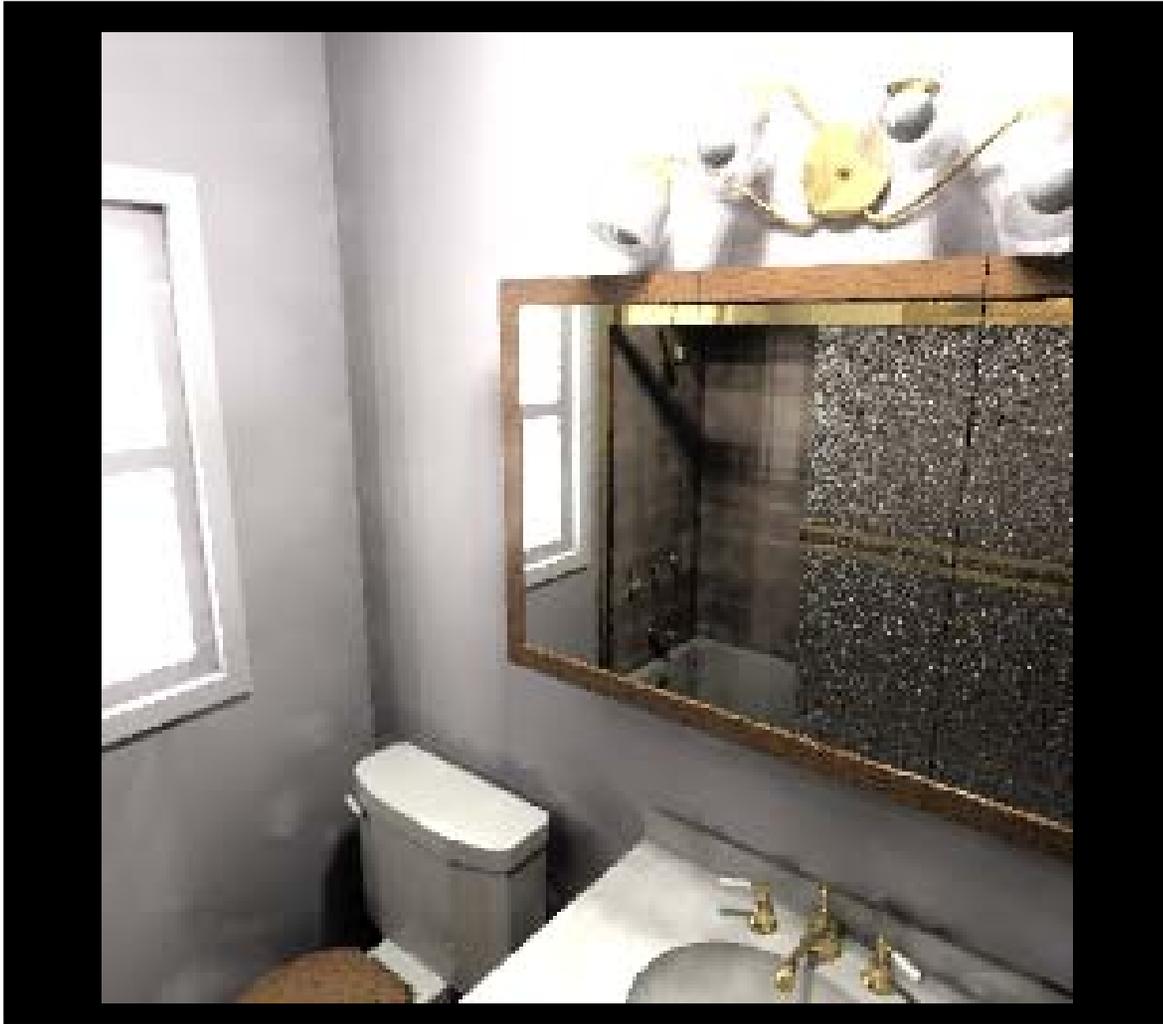
A imagem apresenta-se mais clara que sem iluminação ambiente, em particular, nas sombras projectadas pelo depósito do autoclismo, espelho e candeeiro.

A luz ambiente é constante em toda a cena.

Modelo por:
Greg Ward



Reflexões Difusas

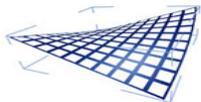


Iluminação Directa,
Especular e
Interreflexões Difusas

As interreflexões difusas
são calculadas usando a
cache de irradiância.

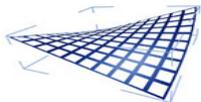
A radiância reflectida
pela parede é agora
variável, notando-se
especialmente na
graduação das sombras.

Modelo por:
Greg Ward



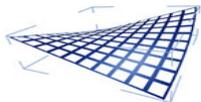
Ray Tracing Clássico – Limitações

- Caminhos (*light paths*) $L[D][S]^*E$
- sharpness
- Aliasing
- Fontes de luz
- Tempo de execução

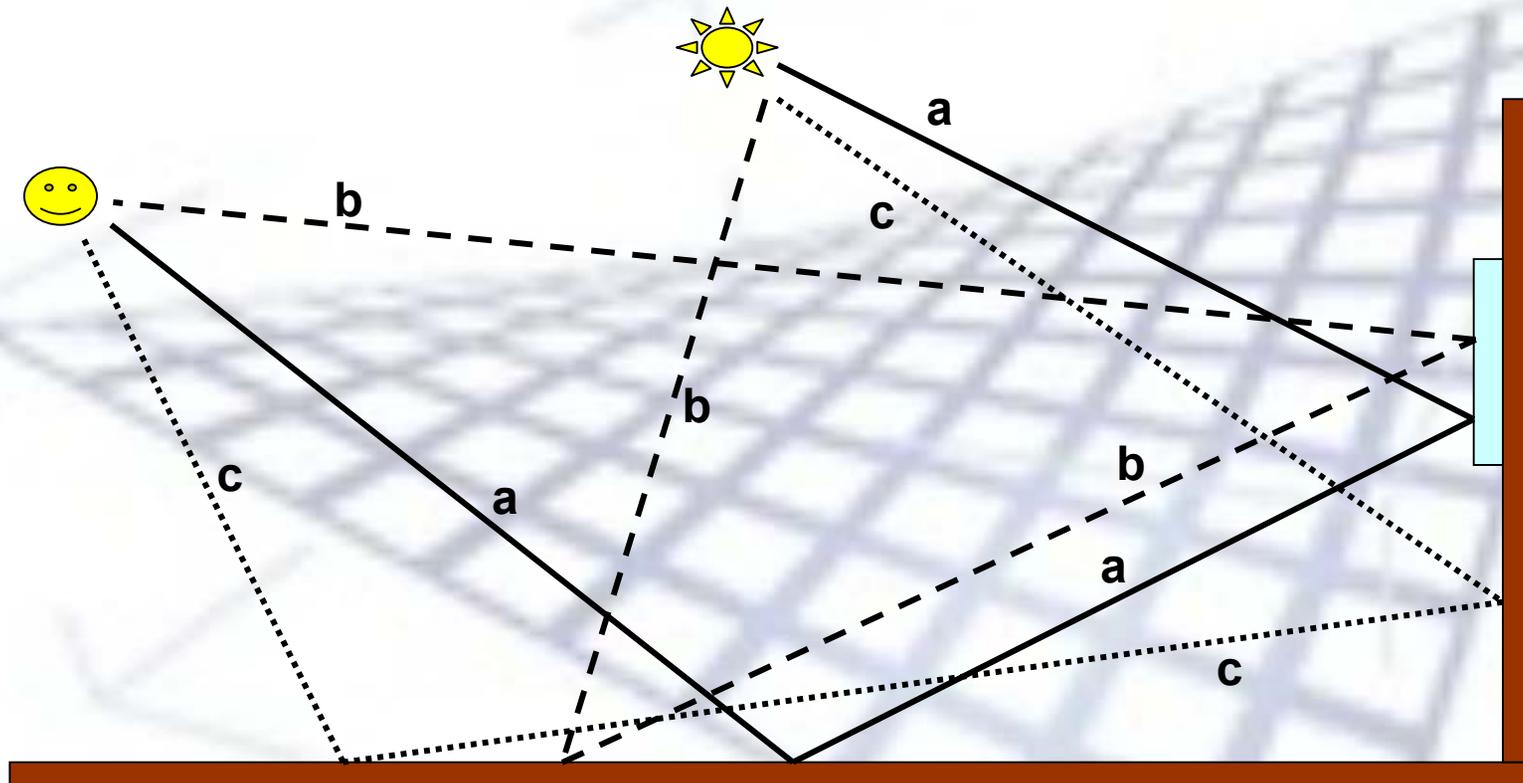


Light Paths

- Sendo o transporte de luz aproximado pela óptica geométrica, podemos conceber que cada fóton percorre um caminho (*light path*) desde a fonte de luz até ao seu destino final
- Este caminho é composto por segmentos de recta direccionais, representando os extremos interacções do fóton com um objecto.
- A interacção pode ser difusa (D) ou especular (S)
- O conjunto de interacções é representado por uma string com origem na fonte de luz (L) e a terminar no ponto onde o fóton é absorvido ou no observador (E)



Light Paths



Material Difuso



Material Especular



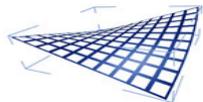
a - LSDE



b - LDSE

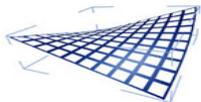


c - LDDE



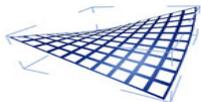
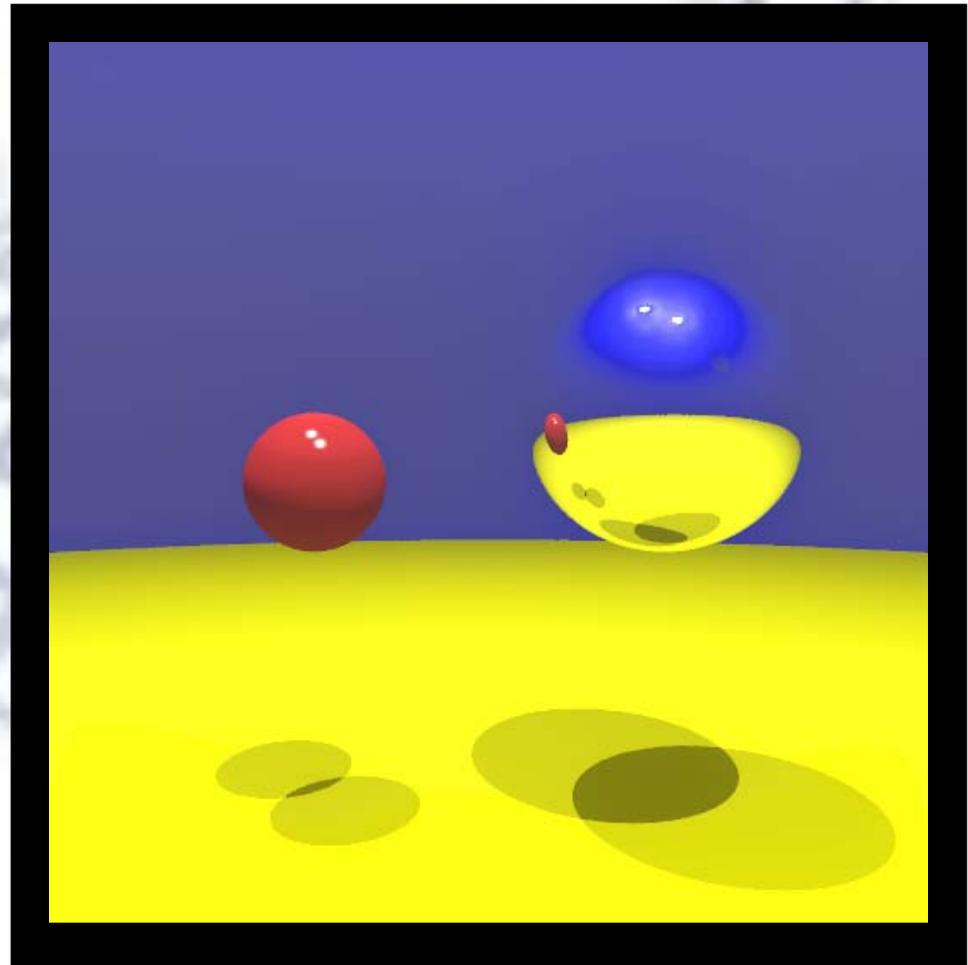
Light Paths

- A árvore de raios gerada começa no observador e termina sempre que encontrar um material difuso.
- Os caminhos simulados em *ray tracing* são sempre do tipo $L[D][S^*]E$
- Os caminhos 'a' e 'c' do acetato anterior não são simulados
- O caminho 'a' resultaria na projecção pelo espelho do reflexo da fonte de luz no chão difuso. O *ray tracing* clássico não modela este fenómeno.



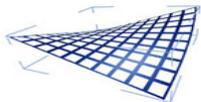
Sharpness

- O carácter direccional preciso dos raios resulta em imagens em que os objectos e seus reflexos têm contornos muito bem definidos, as sombras são extremamente precisas;
- Na realidade os contornos das formas visíveis não são usualmente tão bem definidos e as sombras têm graduações (umbra, penumbra)



Aliasing

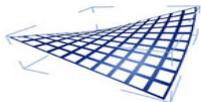
- Defeitos na imagem causados pela baixa frequência de amostragem relativamente às altas frequências presentes no mundo
- Os objectos do mundo virtual, assim como as suas sombras, reflexos, etc., podem projectar-se na imagem com frequências arbitrariamente altas, dependendo do seu tamanho, distância ao observador, detalhe, etc.
- A frequência de amostragem do *ray tracer* é a frequência espacial dos raios, que pode ser muito inferior à que seria necessária para capturar todos os detalhes do mundo



Aliasing

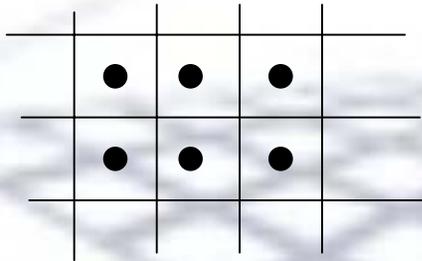
Situações típicas de *aliasing*:

- Características pequenas ou distantes (objectos, sombras, fontes de luz, reflexos, etc.) que não são intersectados por nenhum raio. Uma pequena alteração na posição do observador pode resultar no seu aparecimento súbito (*popping*)
- Arestas incorrectas (*jagged edges*) – devido à natureza regular da grelha de pixels arestas oblíquas relativamente ao ecrã aparecem com um aspecto quadriculado, em lugar de manterem o seu aspecto curvo ou rectilíneo.
- Texturas com grande detalhe ou padrões regulares de alta frequência aparecem *aliased* devido à baixa frequência de amostragem

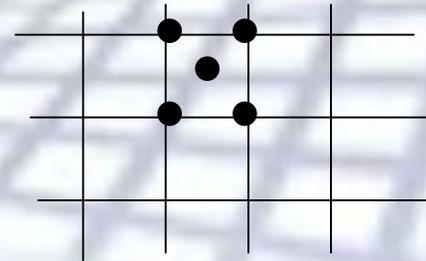


Aliasing – sobre amostragem

- Sobre-amostragem corresponde ao processo de redução dos efeitos de *aliasing* aumentando a frequência de amostragem (resolução dos raios primários) e calculando a média de várias amostras.

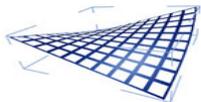


Amostragem central

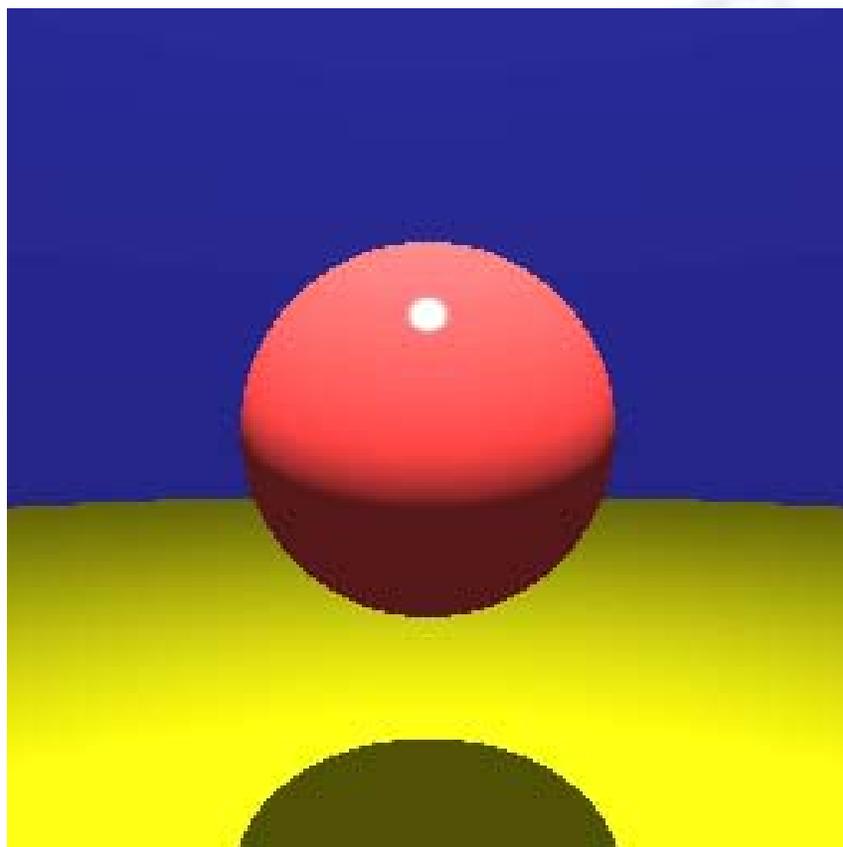


Sobre-Amostragem

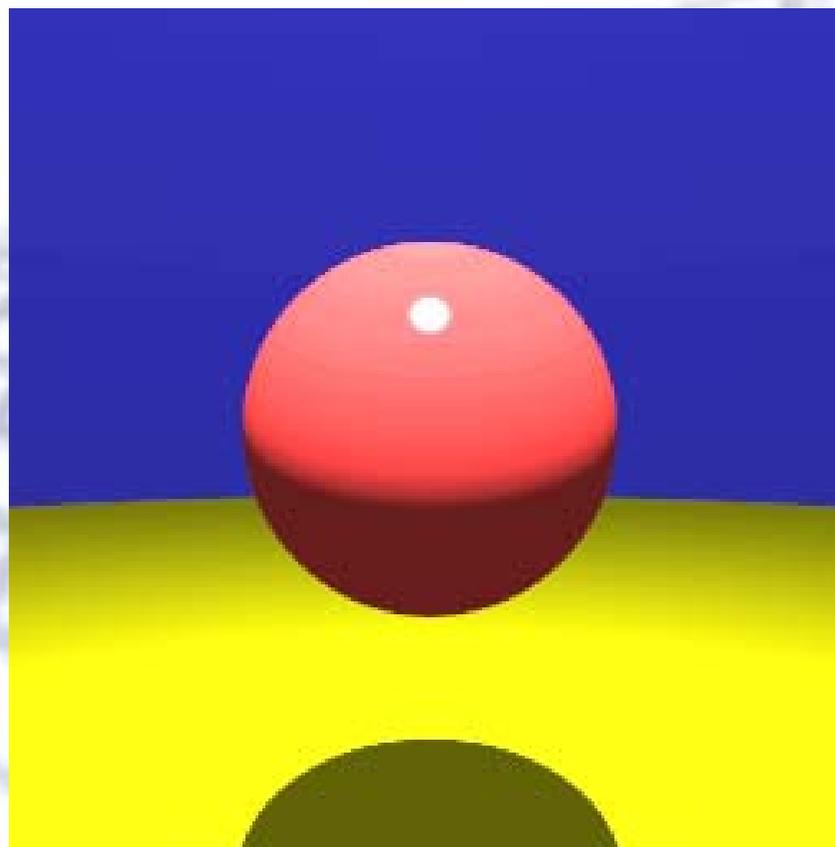
- A sobre-amostragem não elimina o *aliasing*, excepto se a frequência de Nyquist for ultrapassada
- Implica maior número de raios, logo aumento do tempo de *rendering*
- Sobre-amostragem adaptativa – aumentar a frequência de amostragem apenas se a diferença entre 2 pixels vizinhos ultrapassar determinado limite



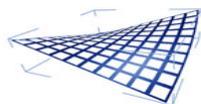
Aliasing – sobre-amostragem



1 raio primário/pixel

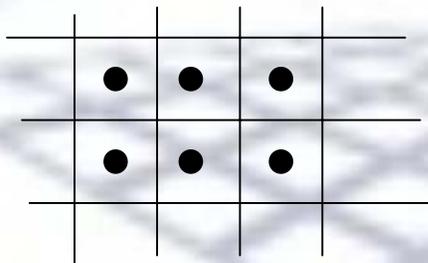


64 raios primários/pixel

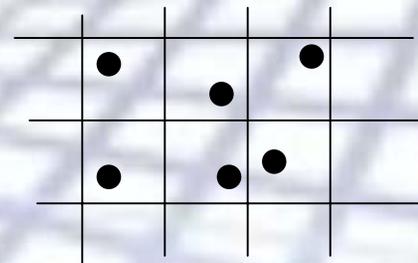


Aliasing - jittering

- Jittering consiste em alterar o padrão regular dos raios primários (frequência de amostragem) perturbando ligeiramente (e aleatoriamente) a direcção destes raios

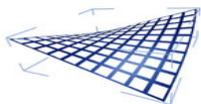


Amostragem regular



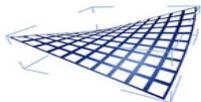
jittering

- O raio que passaria pelo ponto (x,y) do plano de imagem é desviado (*jittered*) para o ponto $(x+\zeta, y+\xi)$, onde ζ, ξ são variáveis estocásticas
- *Jittering* não implica aumentar o número de raios primários, mas resulta em aumento do ruído (variância)

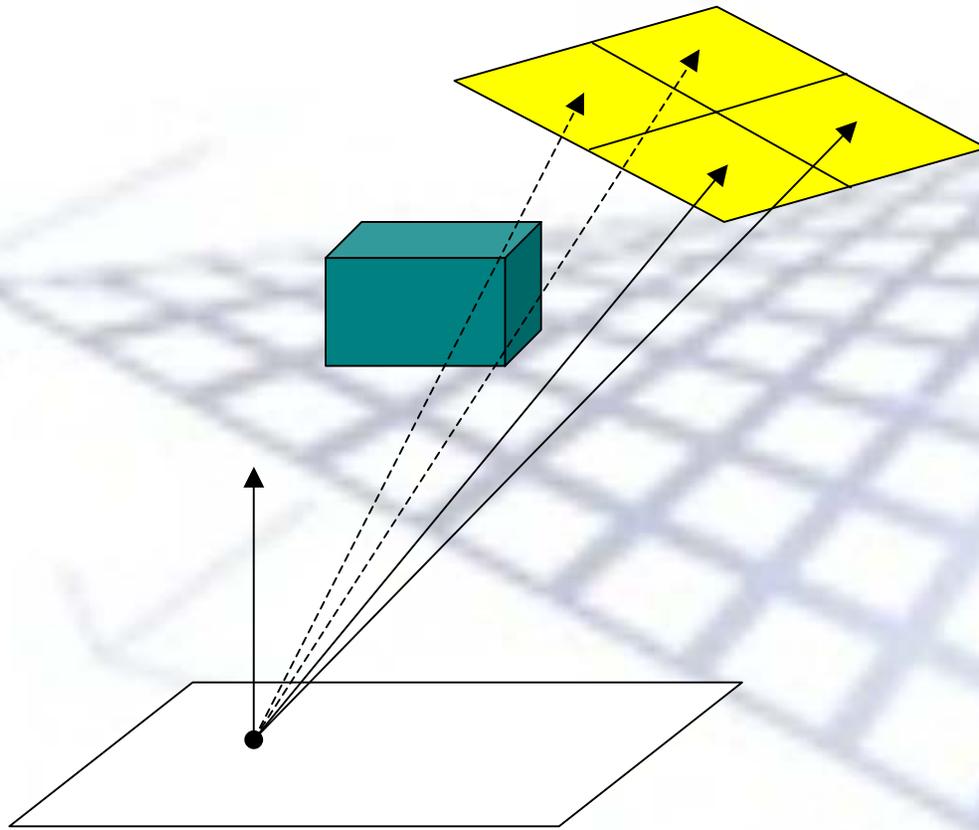


Fontes de Luz

- As fontes de luz reais não são pontuais;
- Têm uma forma geométrica que pode ser incluída no modelo do *ray tracer*
- Cada fonte de luz é amostrada por mais do que um *shadow ray*
- Aumento do número de raios implica aumento do tempo de computação
- Critérios para decidir quantos raios enviar em direcção a cada fonte de luz:
 1. Área da fonte de luz
 2. Potência radiante da fonte de luz
 3. Orientação da fonte de luz relativamente à superfície de interesse
 4. Distância da fonte de luz ao ponto de interesse



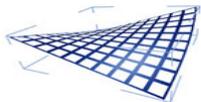
Fontes de Luz



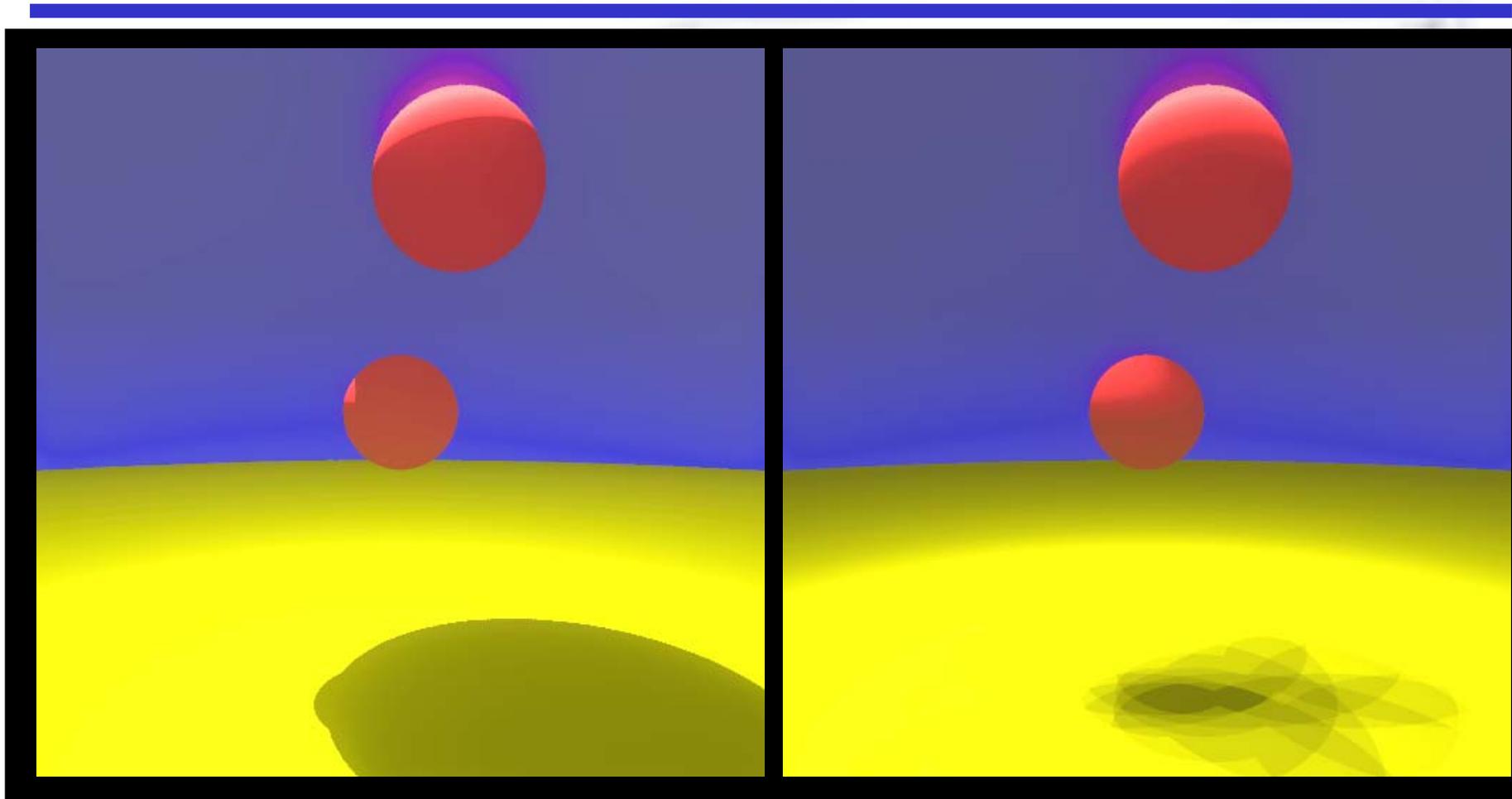
A fonte de luz é dividida em sub-áreas.

A cada sub-área corresponde um *shadow ray*.

O aumento do número de *shadow rays* aumenta o tempo de *rendering*.

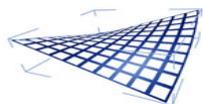


Fontes de Luz



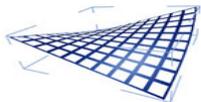
Fonte de luz pontual – 152105 raios

Fonte de luz poligonal – 1688849 raios



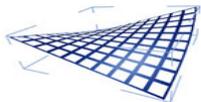
Tempo de execução

- Os longos tempos de execução do *ray tracing* devem-se essencialmente ao cálculo de intersecções entre raios e objectos, necessárias para calcular a visibilidade.
- Soluções alternativas:
 1. Reduzir o número de raios para reduzir o número de intersecções
 2. Reduzir o custo do cálculo das intersecções



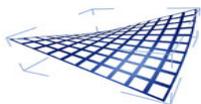
Tempo de execução – número de raios

- Redução dos raios primários –
 - Amostragem de um menor número de raios primários, calculando os restantes por interpolação: pode resultar no aumento do *aliasing*
 - Adopção de técnicas adaptativas – variar a frequência de amostragem em função da diferença entre amostras vizinhas
- Redução dos raios secundários –
 - diminuição da profundidade da árvore de raios
 - envio de novos raios apenas se a sua contribuição esperada for maior que um determinado limite (função dos coeficientes e da profundidade na árvore)
- Redução dos *shadow rays* –
 - Seleccionar criteriosamente as fontes de luz a amostrar para cada ponto de interesse (função da contribuição esperada)



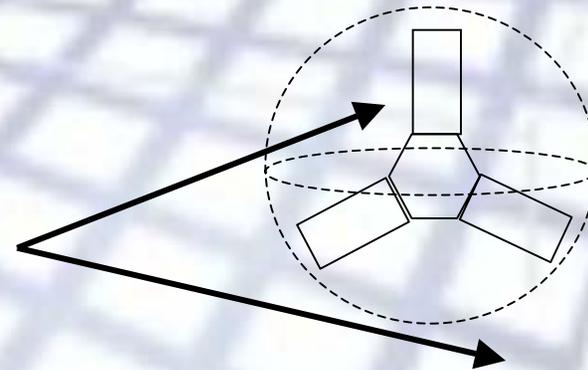
Tempo de execução – custo das intersecções

- A intersecção de cada raio por **todos** os objectos é claramente redundante: um raio atravessa apenas um subconjunto do espaço dos objectos
- A intersecção de um raio (recta) com objectos que possam ser expressos por expressões analíticas (planos, esferas) é mais rápida que a intersecção com formas complexas como um polígono

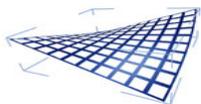


Tempo de execução – bounding volumes

- Objectos complexos podem ser encapsulados em objectos simples (esferas, por exemplo)
- O raio é testado com o objecto simples: apenas se este for intersectado, será então testado com os objectos complexos contidos dentro do *bounding volume*

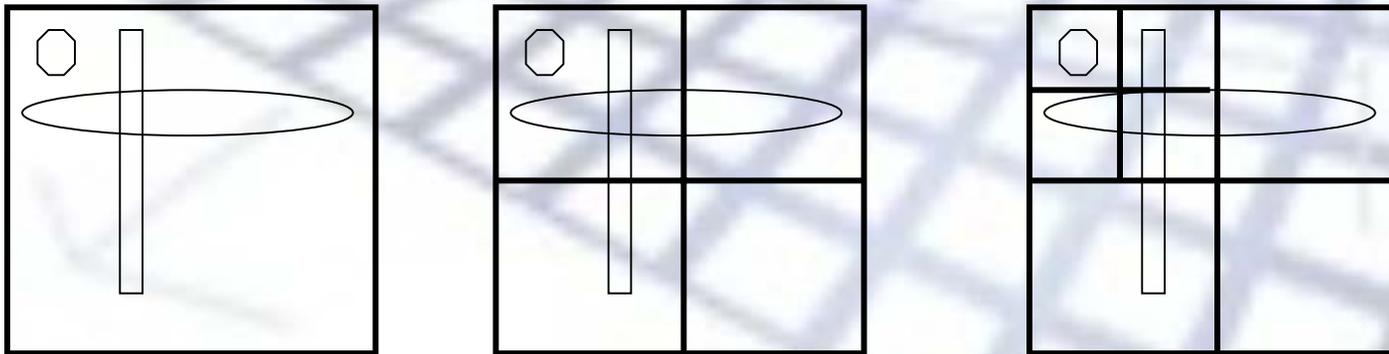


- Este processo pode ser continuado hierarquicamente, com *bounding volumes* dentro de *bounding volumes*.

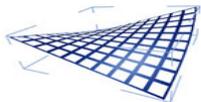


Tempo de execução – ordenação espacial

- O volume (cubo) ocupado pelos objectos pode ser recursivamente dividido em sub-volumes (*voxels*)
- Cada volume é dividido em 8 sub-volumes
- O processo de divisão termina quando o número de objectos dentro de um *voxel* for menor que um determinado limite ou quando a profundidade da árvore de volumes exceder um limite



- No espaço 3D esta sub-divisão resulta numa estrutura conhecida como *octree*.



Tempo de execução – ordenação espacial

- Um raio é intersectado com o volume de nível mais elevado e vai descendo na *octree* apenas para os volumes que intersecta.
- Uma vantagem da *octree* é que conhecendo a origem do raio, conhece-se exactamente quais e porque ordem são visitados os *voxels*.
- A travessia pode terminar assim que se intersectar um objecto, pois este será o mais próximo da origem.

