

Modelos de Iluminação Global

Prof. Antonio L. Apolinário Jr.
Estagiaria Docente: Rafaela Alcantara

UFBA/IM/DCC/BCC - 2018s.1

Roteiro

- Iluminação Global
- Princípio do Acompanhamento de Raios
 - Modelos de Rasterização
 - *Object Order*
 - *Image Order*
- *Ray Casting*
 - Calculo de interseção de raio contra primitivas
 - Calculo da Normal
- *Ray Tracing Recursivo*
- Técnicas de Aceleração

Leitura de referencia

- Capítulo 12

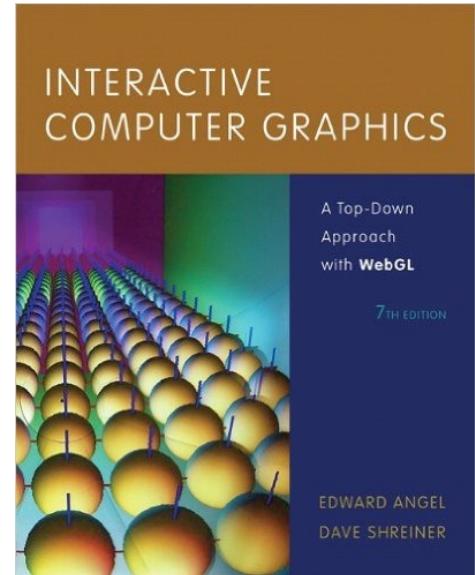
Interactive Computer Graphics - A top-down approach with WebGL

7th Edition

Angel, Edward

Shreiner, Dave

Addison-Wesley. 2014.



Leitura de referencia

- Capítulo 15 (seção 4)

Computer Graphics : Principles and Practice Third Edition in C

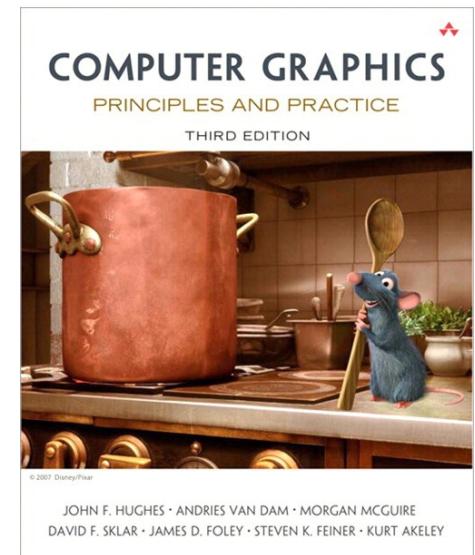
3rd Edition

John F. Hughes / Andries van Dam

Morgan McGuire / David F. Sklar

James D. Foley / Steven K. Feiner

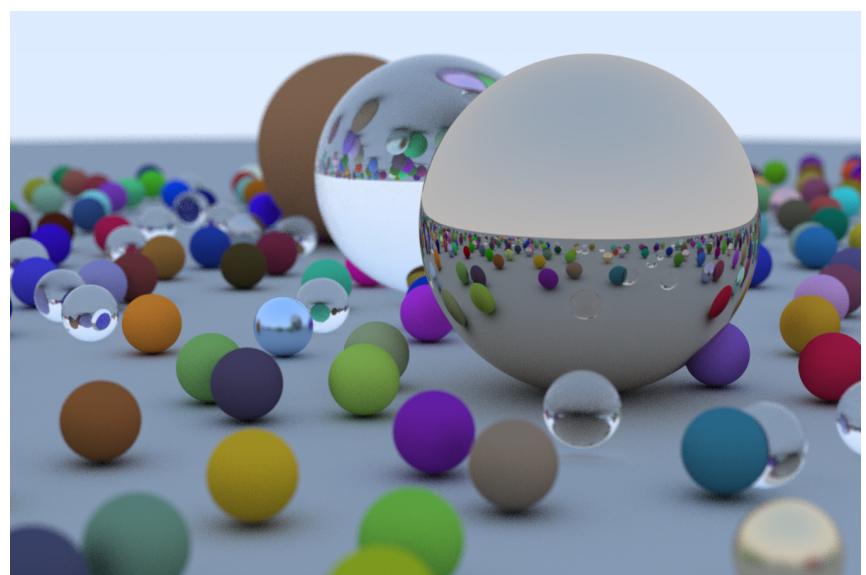
Addison-Wesley. 2013.



Modelos de Iluminação Global

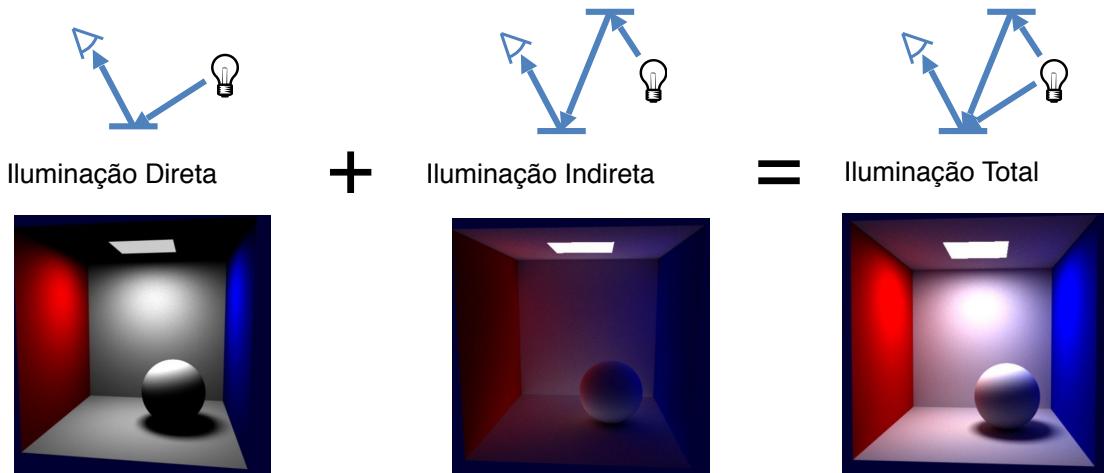
Algoritmos de Reflexão Locais

- Não são capazes de representar de forma direta:
 - Sombras
 - Penumbra
 - Reflexão indireta
 - Entre outros...



Iluminação Global

- Leva em conta a iluminação direta e indireta de uma cena



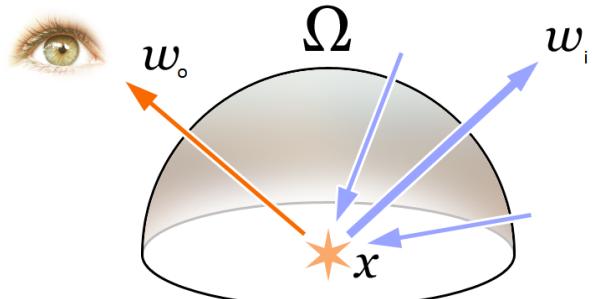
Iluminação Global

- Equação de Renderização [Kajiya86]

$$L_o(\mathbf{x}, \omega_o, \lambda, t) = L_e(\mathbf{x}, \omega_o, \lambda, t) + \int_{\Omega} f_r(\mathbf{x}, \omega_i, \omega_o, \lambda, t) L_i(\mathbf{x}, \omega_i, \lambda, t) (\omega_i \cdot \mathbf{n}) d\omega_i$$

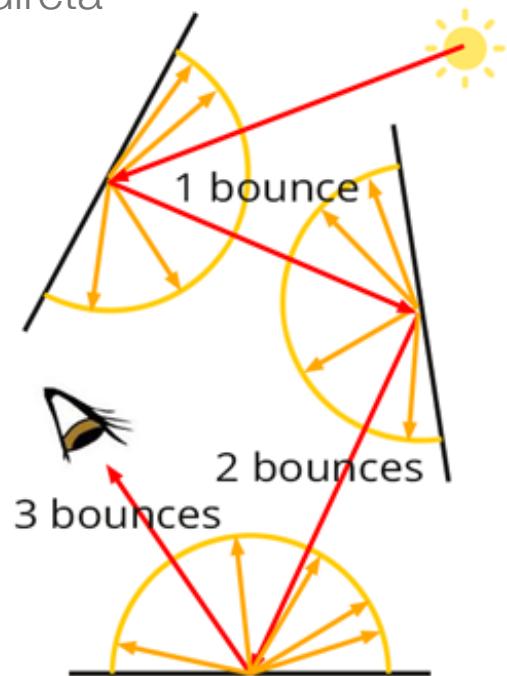
onde

- λ => comprimento de onda da luz
- t => temp
- \mathbf{x} => localização no espaço
- \mathbf{n} => vetor normal
- ω_o => direção de saída da luz
- ω_i => direção negativa da luz de chegada
- $L_o(\mathbf{x}, \omega_o, \lambda, t)$ => radiância total
- $L_e(\mathbf{x}, \omega_o, \lambda, t)$ => radiância emitida
- Ω => hemisfério unitário em torno de \mathbf{n} contendo todos os valores possíveis de ω_i
- $f_r(\mathbf{x}, \omega_i, \omega_o, \lambda, t)$ => BRDF (*bidirectional reflectance distribution function*),
- $L_i(\mathbf{x}, \omega_i, \lambda, t)$ => radiância chegando em \mathbf{x} na direção ω_i
- $\omega_i \cdot \mathbf{n}$ => atenuação da irradiância de chegada em função do seu ângulo de inclinação em relação a normal



Illuminação Global

- Considerar apenas a reflexão indireta



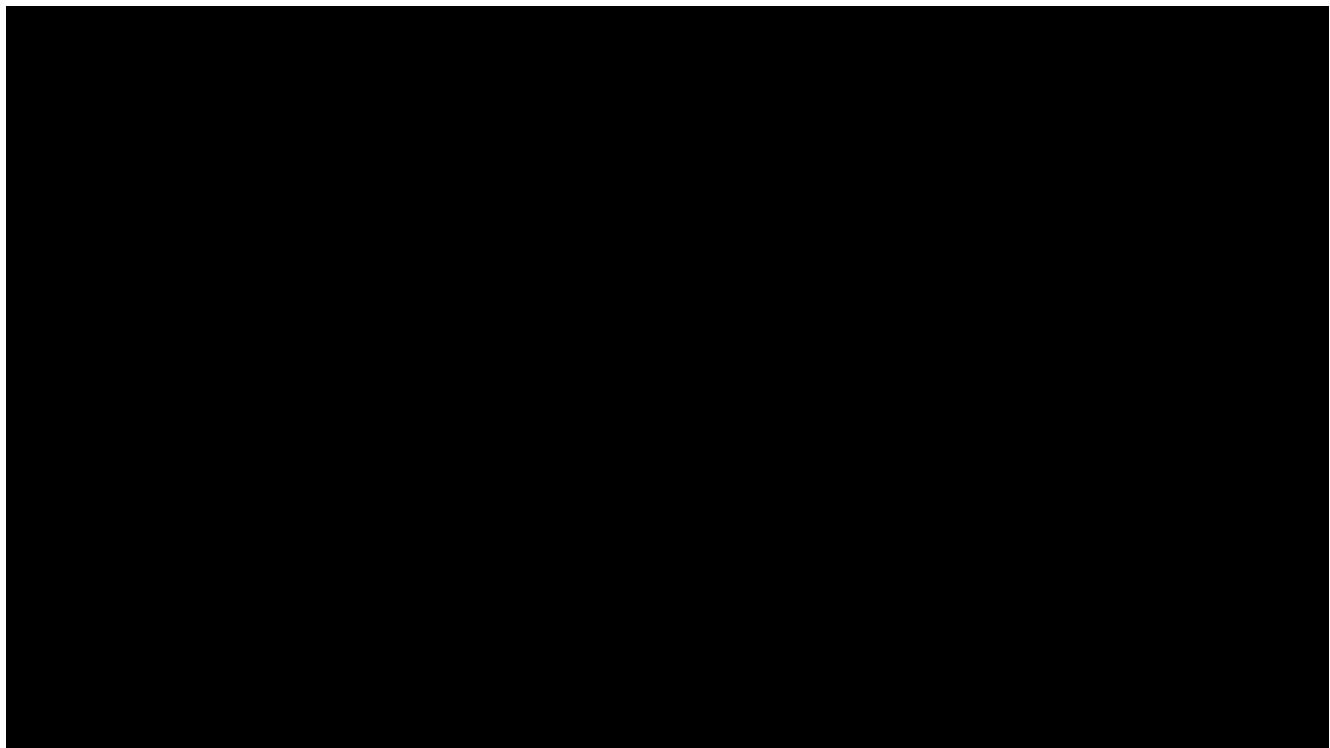
© www.scratchapixel.com

Ray Tracing

- Reflexão indireta
 - Acompanhamento de raios



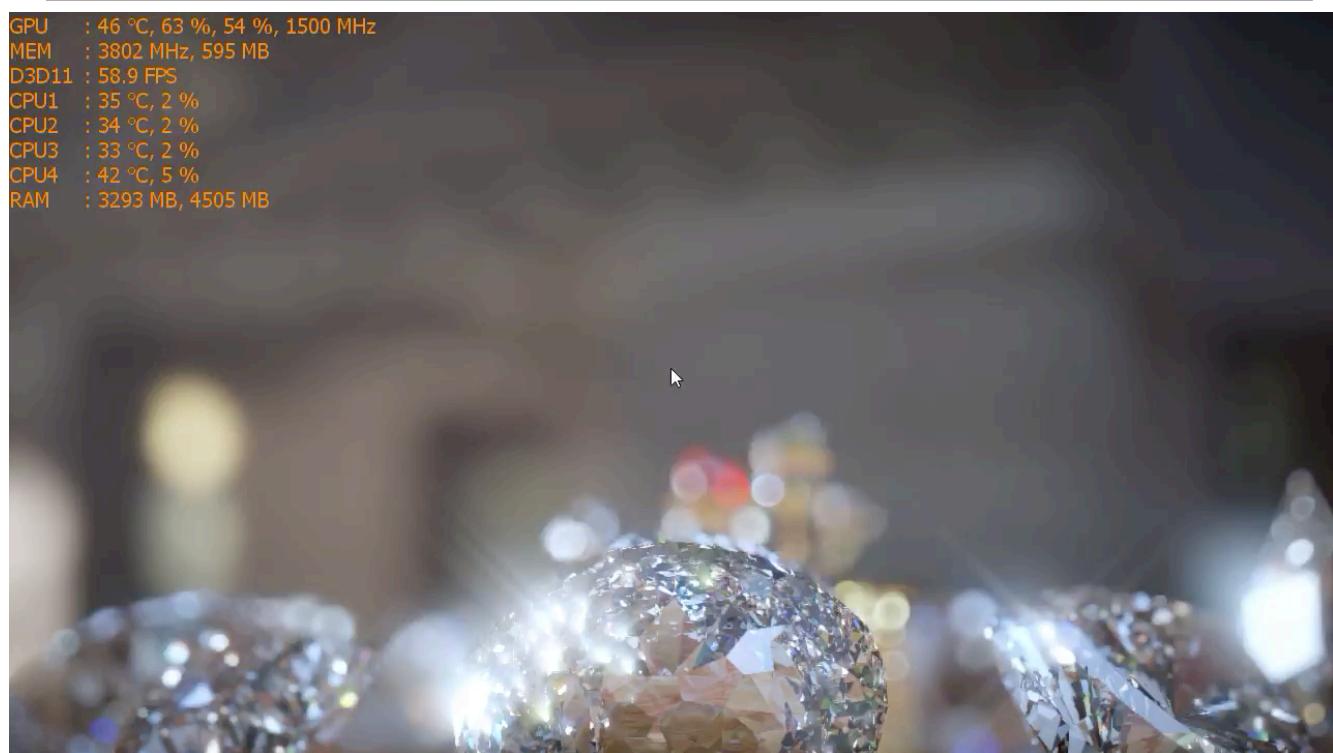
Ray Tracing



Luxo Jr. 1986 directed by John Lasseter. [SIGGRAPH](#) 1986.

Ray Tracing

GPU : 46 °C, 63 %, 54 %, 1500 MHz
MEM : 3802 MHz, 595 MB
D3D11 : 58.9 FPS
CPU1 : 35 °C, 2 %
CPU2 : 34 °C, 2 %
CPU3 : 33 °C, 2 %
CPU4 : 42 °C, 5 %
RAM : 3293 MB, 4505 MB

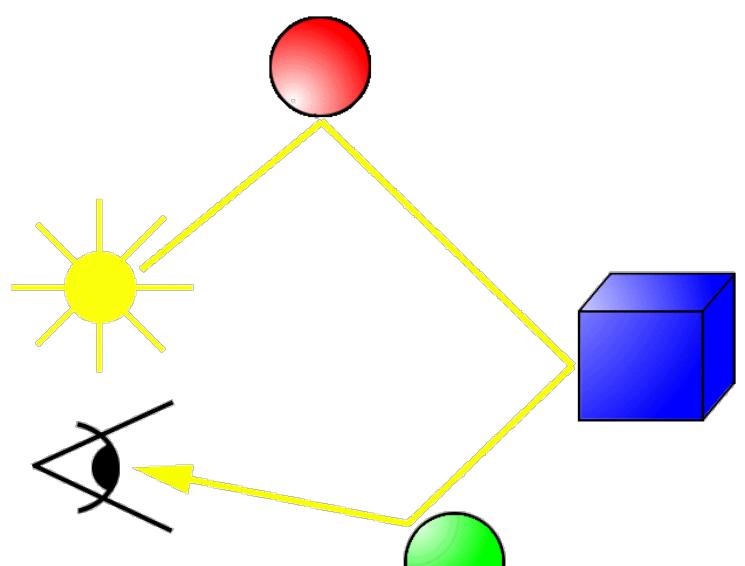


RigidGems (2014) http://www.rigidgems.sakura.ne.jp/index_en.html

Princípio do Acompanhamento de Raios

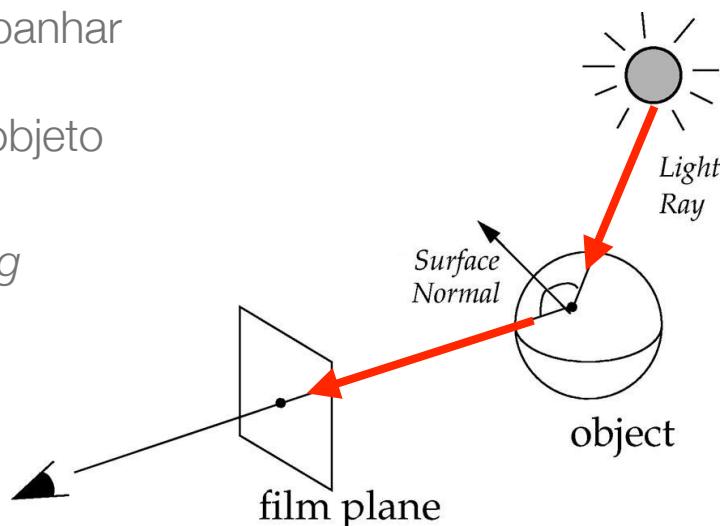
Acompanhamento de Raios (*Ray Tracing*)

- Combina um modelo de iluminação com determinação de visibilidade
- Simula efeitos de iluminação global tais como
 - Sombras
 - Reflexão especular e refração recursivas
- Acompanha vários caminhos da luz



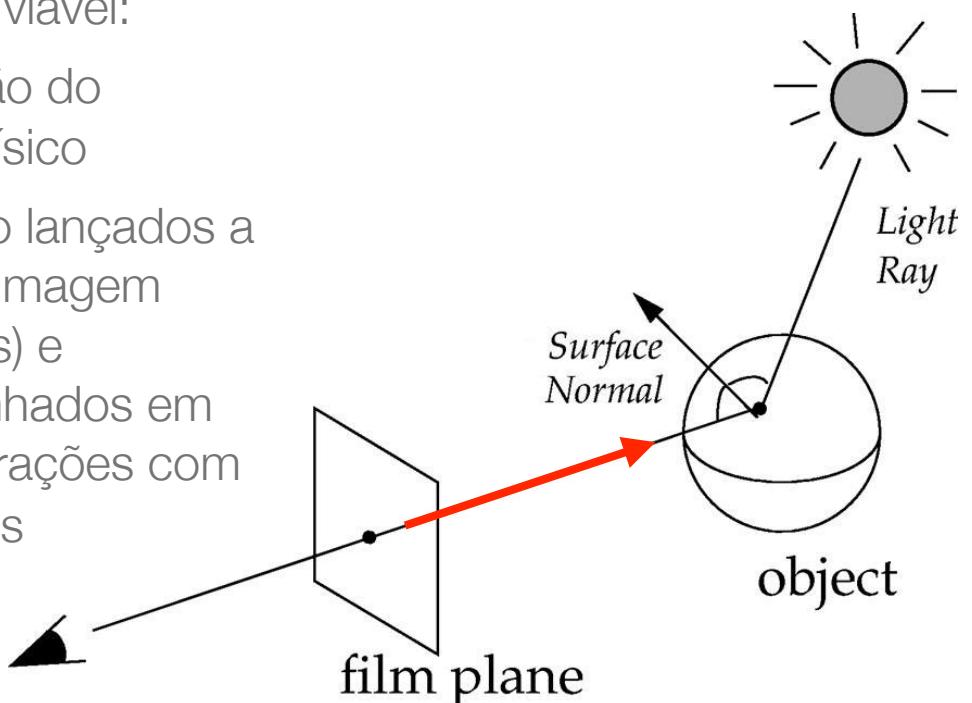
Ray Tracing

- Simulação do fenômeno físico:
 - Objetos são visíveis pela interação de raios que emanam da fonte de luz, refletem nos objetos e são captados por um sensor óptico.
 - Algoritmo deve acompanhar esses raios , calcular sua interação com o objeto (*Phong*) e propagá-la.
 - *Forward Ray Tracing*
 - Solução inviável...



Ray Tracing

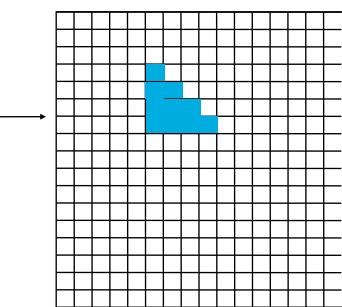
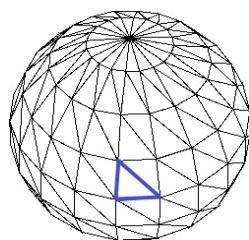
- Solução mais viável:
 - Aproximação do fenômeno físico
 - Raios são lançados a partir da imagem (amostras) e acompanhados em suas interações com os objetos



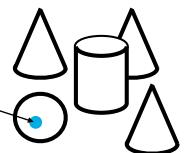
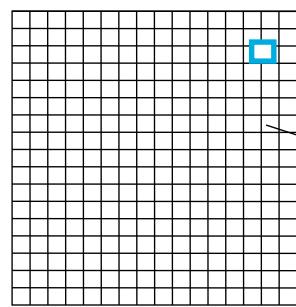
Ray Tracing x Rasterização

- Rasterização é uma técnica baseada no objetos (*object-order*)
- Ray tracing é uma técnica baseada na imagem (*image-order*)

Para cada triângulo do objeto...

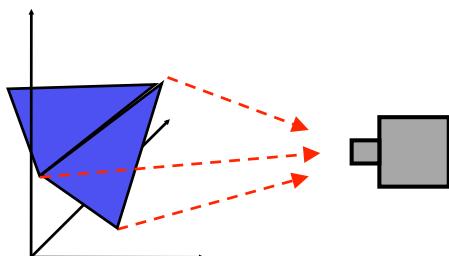


Para cada amostra da imagem...

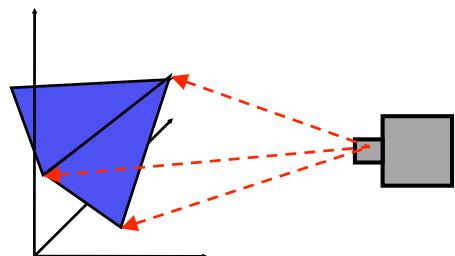


Ray Tracing x Rasterização

- Rasterização:
 - Baseado projeção da geometria do objeto para o sistema da câmera
- Ray tracing:
 - Baseado na projeção de elementos da imagem nos objetos



Rasterization:
Projection geometry forward



Ray Tracing:
Project image samples backward

Ray Tracing x Rasterização

- Algoritmo Rasterização:

Para cada objeto na cena

Para cada face do objeto

Informar a geometria dos vértices

Rasterizar a face
Aplicando *Phong*

- Algoritmo *Ray Tracing*:

Para cada amostra da imagem

Determinar os objetos atingidos pelo raio lançado da amostra

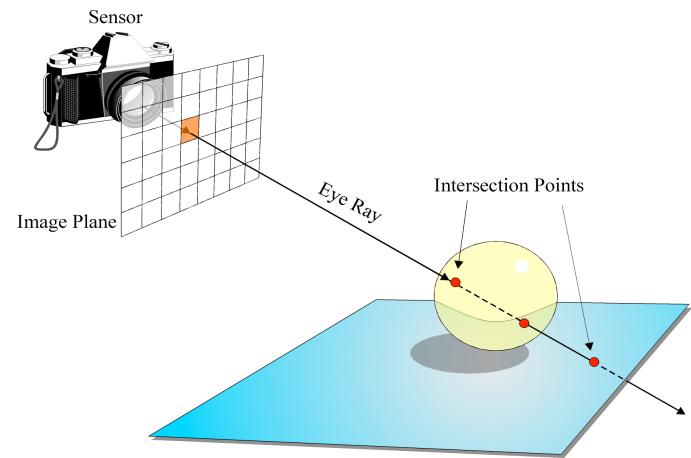
Pintar a amostra com a cor do objeto naquele ponto

Aplicar *Phong*

Ray Casting

Ray Casting

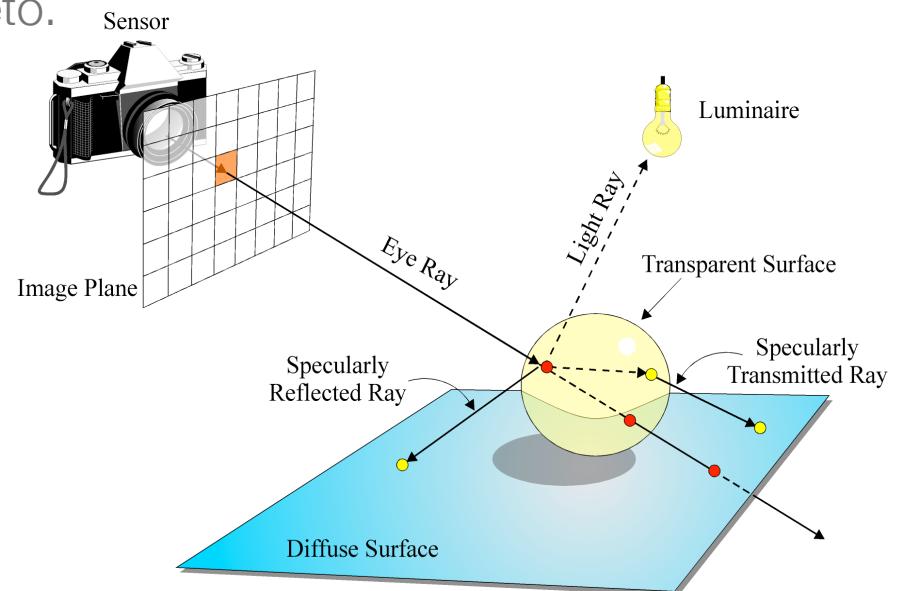
- Acompanhamento de um único raio de luz para cada pixel
- Identificado um ponto de intercessão, a iluminação é calculada aplicando-se *Phong*



Appel A. (1968) **Some techniques for machine rendering of solids**. AFIPS Conference Proc. 32 pp.37-45.

Ray Casting

- Vários fenômenos podem ocorrer quando um raio encontra um objeto.



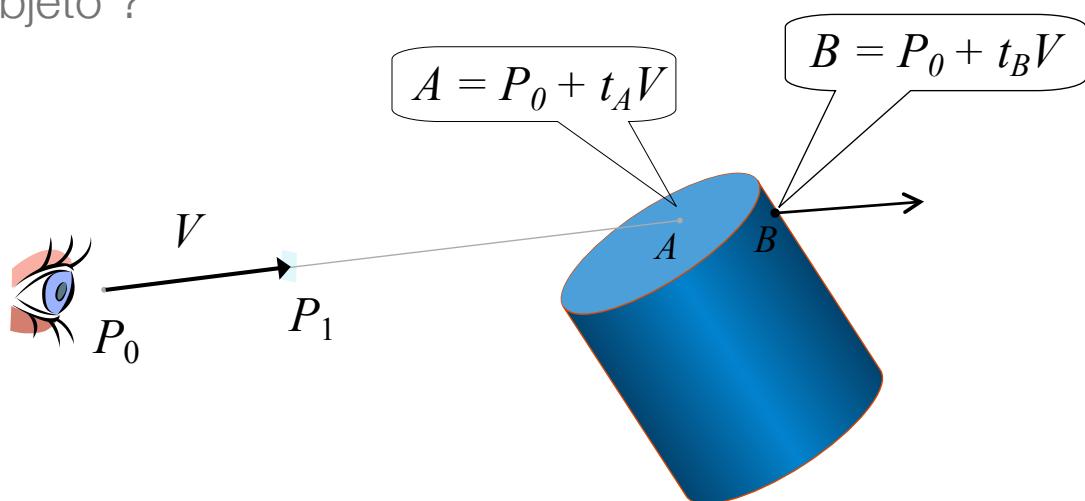
Appel A. (1968) **Some techniques for machine rendering of solids**. AFIPS Conference Proc. 32 pp.37-45.

Ray Casting

- Como identificar qual objeto é atingido por um raio?
 - Calcular a interseção do raio com todos os objetos da cena
 - Ordenar as interseções
 - Tomar a mais próxima do observador.
- Problema:
 - Custo do calculo da interseção pode ser alto !!

Interseção Raio / Objeto

- Raio é modelado como uma reta em forma paramétrica:
 - $\mathbf{R}(t) = \mathbf{P}_0 + t(\mathbf{P}_1 - \mathbf{P}_0) = \mathbf{P}_0 + t\mathbf{V}$
- Para quais valores do parâmetro t a reta intercepta o objeto ?



Interseção Raio / Objeto Implícito

- Raio é modelado em forma paramétrica:
 - $\mathbf{R(t)} = [\mathbf{Rx(t)} \; \mathbf{Ry(t)} \; \mathbf{Rz(t)}]\mathbf{T}$
- Logo, os pontos de interseção satisfazem
 - $f(\mathbf{Rx(t)}, \mathbf{Ry(t)}, \mathbf{Rz(t)}) = 0$
- Basta resolver a equação para determinar o(s) valor(es) de t que a satisfazem

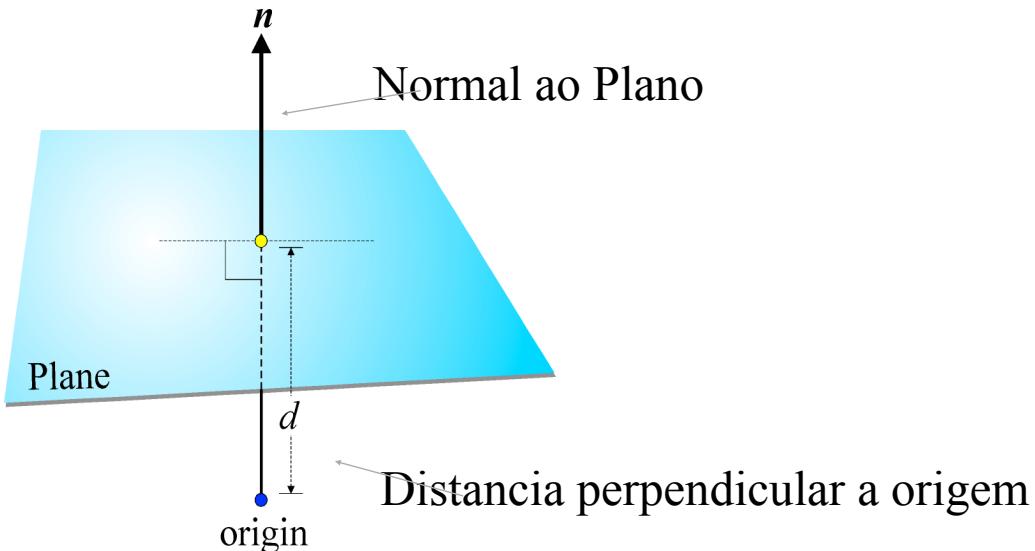
Interseção com Planos

- Plano em forma implícita
$$Ax + By + Cz + D = 0$$
- Se queremos um plano que passa por um ponto \mathbf{Q} e tem normal \mathbf{N} podemos escrever
$$(\mathbf{P} - \mathbf{Q}) \cdot \mathbf{N} = 0$$
- Resolução da forma habitual

Interseção Raio / Plano

- Um plano é definido implicitamente por:

$$\vec{v} \times \vec{n}_P = d_p \quad \text{or} \quad n_x x + n_y y + n_z z - d_p = 0$$



Interseção Raio / Plano

- Substituindo a equação do plano no raio e resolver uma equação em t :

$$n_x(O_x + t d_x) + n_y(O_y + t d_y) + n_z(O_z + t d_z) - d_p = 0$$

$$\Rightarrow t = -\frac{n_x O_x + n_y O_y + n_z O_z - d_p}{n_x d_x + n_y d_y + n_z d_z} = -\frac{\vec{n}_P \times \vec{O}_r}{\vec{n}_P \times \vec{d}_r}$$

$$\vec{n}_P \times \vec{d}_r = 0$$

- Caso o raio seja paralelo ao plano não há interseção

Interseção Raio / Quádrica

- Esfera de raio 1 centrada na origem:

$$x^2 + y^2 + z^2 - 1 = 0$$

- Raio parametrizado como:

$$[V_x t + P_x \ V_y t + P_y \ V_z t + P_z]^T$$

- Logo,

$$(V_x t + P_x)^2 + (V_y t + P_y)^2 + (V_z t + P_z)^2 - 1 = 0$$

- ou

$$at^2 + bt + c = 0$$

- onde:

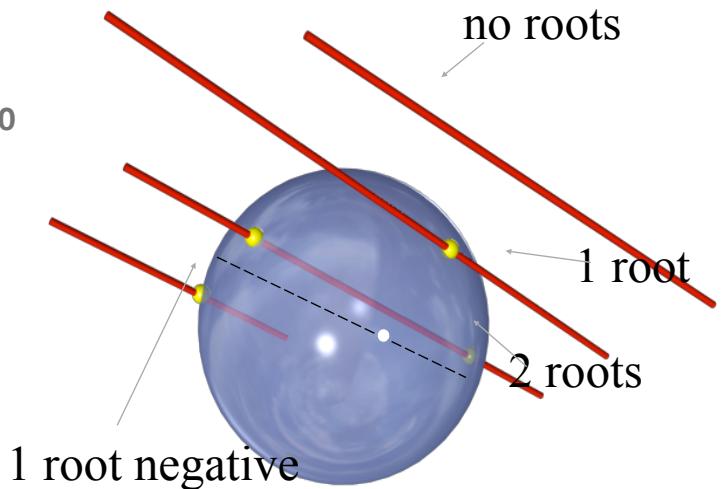
$$a = V_x^2 + V_y^2 + V_z^2$$

$$b = 2 * (V_x P_x + V_y P_y + V_z P_z)$$

$$c = P_x^2 + P_y^2 + P_z^2 - 1$$

- Seja $\Delta = b^2 - 4ac$, então

$$t = \frac{-b \pm \sqrt{\Delta}}{2a}$$

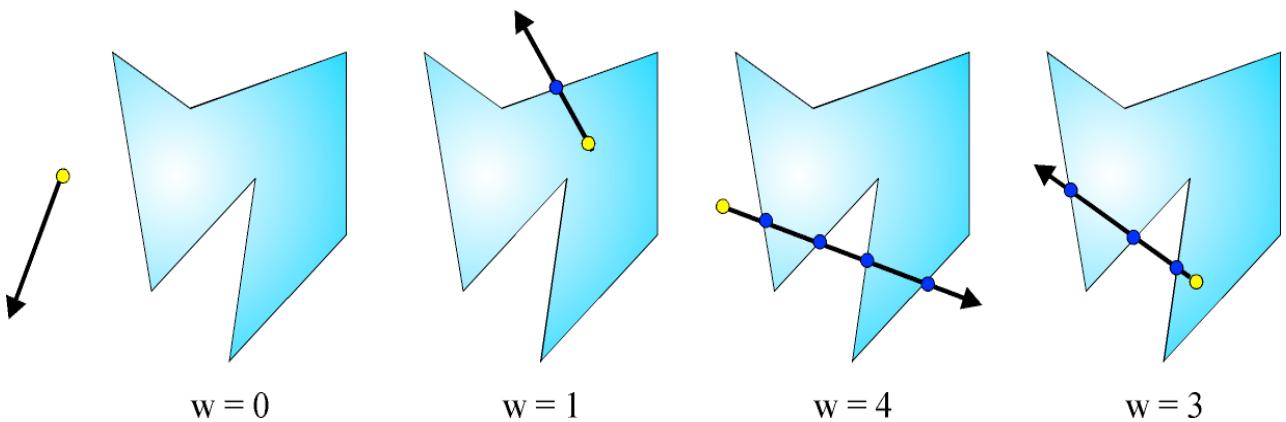


Interseção Raio / Polígono

- Calcular a interseção raio contra o plano do polígono
 - Dado o ponto de interseção, determinar se este está no interior do polígono

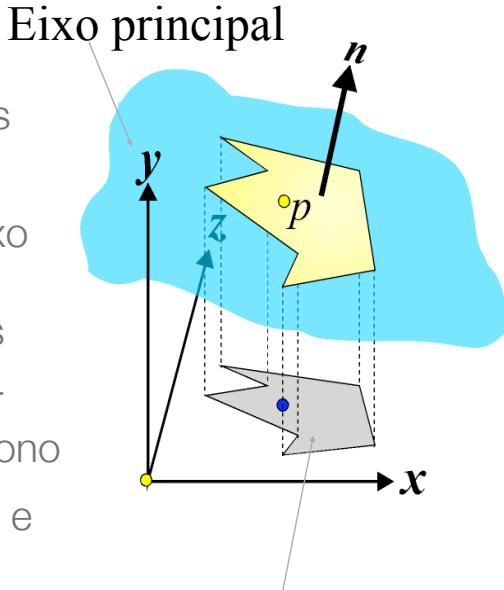
Interseção Raio / Polígono

- Teorema de Jordan:
 - Construir um raio qualquer cuja origem é o ponto de interseção
 - Contar o numero de arestas do polígono que cruzam o raio (w)
 - Se w é ímpar então o ponto é interior



Interseção Raio / Polígono

- Problema essencialmente 2D:
 - Depois de calcular o ponto de interseção raio / plano podemos reduzir o problema para 2D:
 - Projetando paralelamente ao eixo principal do polígono no plano formado pelos outros dois eixos
 - Eixo principal é dado pela maior coordenada da normal do polígono
 - A projeção preserva a topologia e simplifica os cálculos



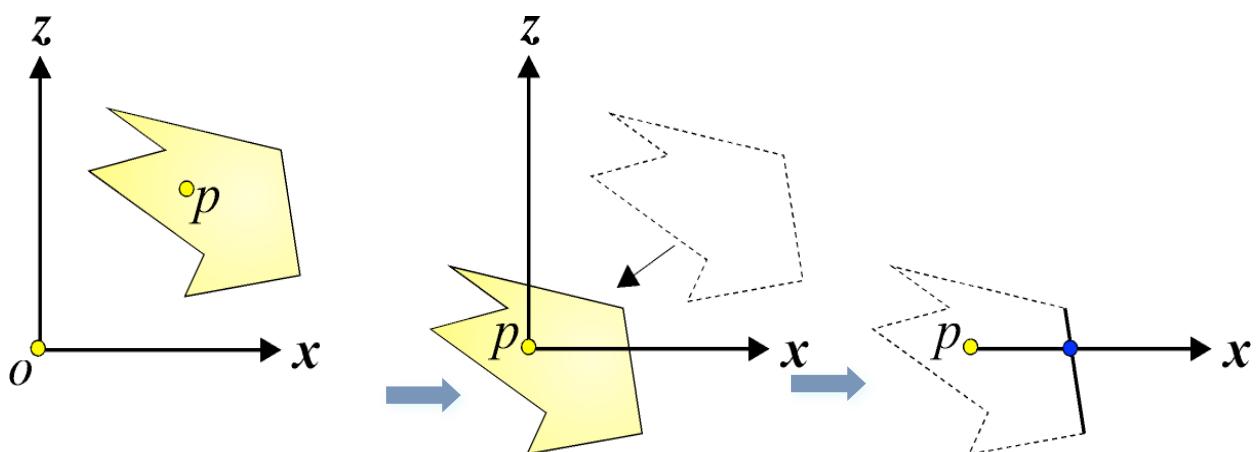
Projeção do polígono e ponto de interseção

Interseção Raio / Polígono

- A projeção é constante para cada polígono
 - Pode ser pré-computada juntamente com o eixo principal.
- Uma vez projetados, o numero de cruzamentos pode ser determinado:
 - Transladar o ponto de interseção para a origem
 - Aplicando a mesma translação ao polígono.
 - Utilizar um dos eixos como direção
 - Determinar o numero de cruzamentos ao longo do eixo positivo.

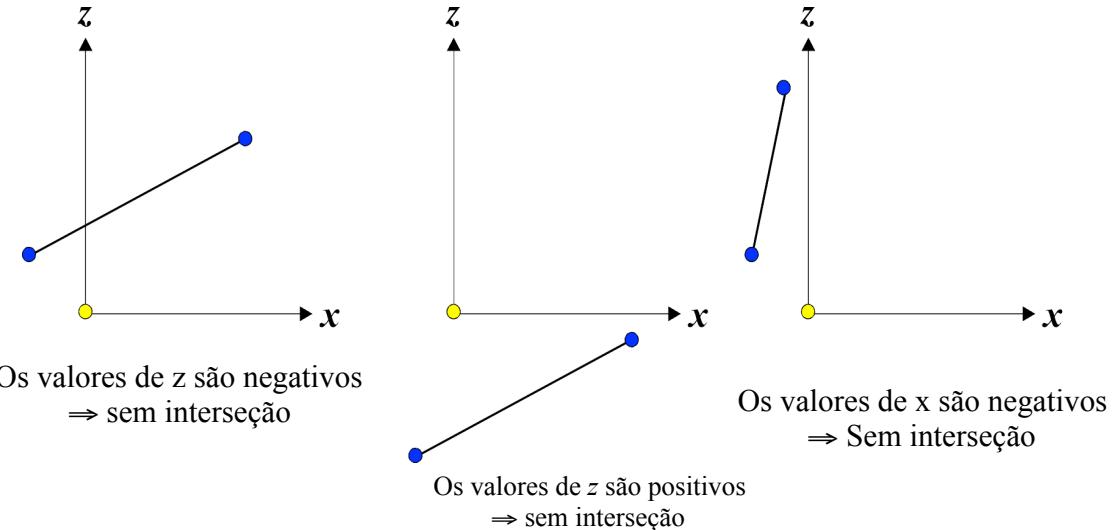
Interseção Raio / Polígono

- Exemplo:



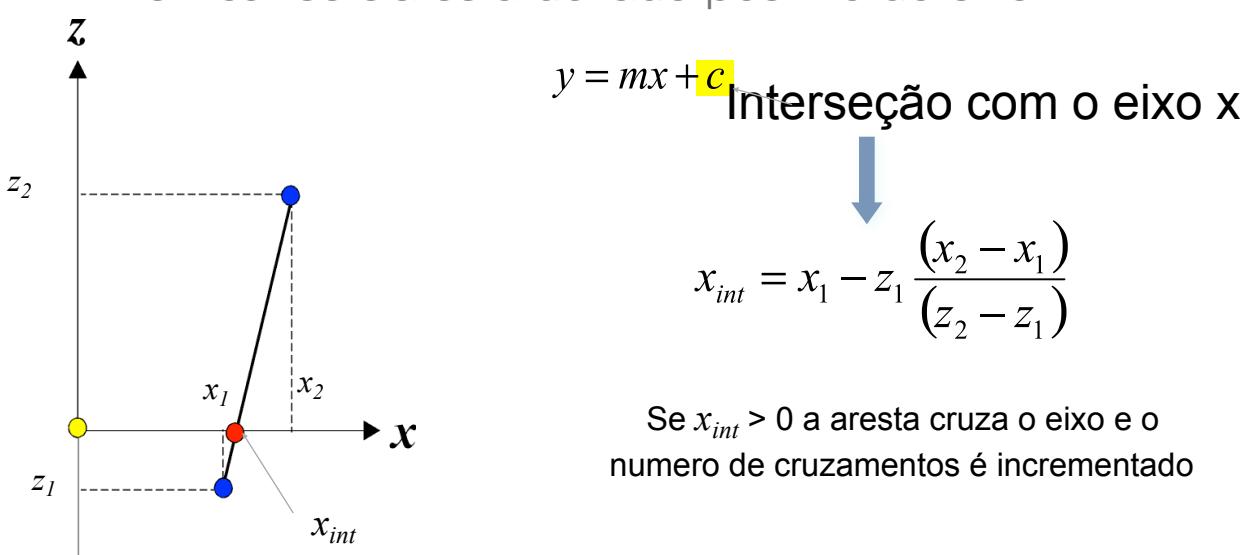
Interseção Raio / Polígono

- O processo pode ser acelerado determinando quais arestas interceptam o eixo
 - 3 casos simples de rejeição trivial:



Interseção Raio / Polígono

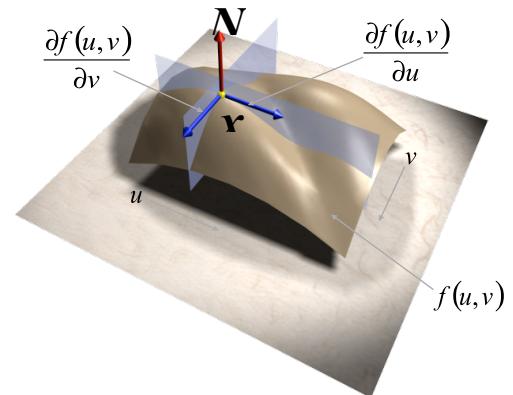
- Caso não haja rejeição trivial a interseção com o eixo deve ser calculada
 - Verificar se ela está do lado positivo do eixo:



Vetor Normal

- Em Superfícies Explicitas:
- Dada uma superfície explícita definindo a superfície S com parâmetros u e v , podemos encontrar a normal no ponto x :
- Determinando as tangentes as curvas em u e v no ponto, ou seja:

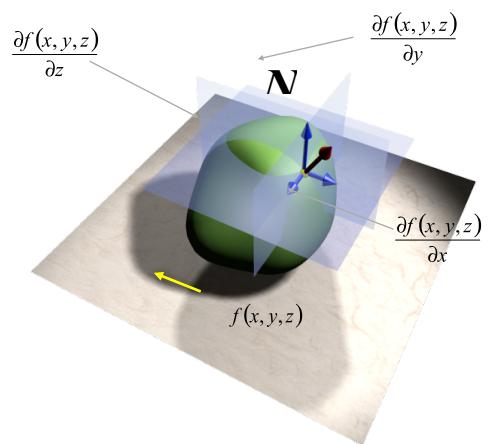
$$N = \frac{\frac{\partial f}{\partial v} \times \frac{\partial f}{\partial u}}{\left| \frac{\partial f}{\partial v} \times \frac{\partial f}{\partial u} \right|}$$



Vetor Normal

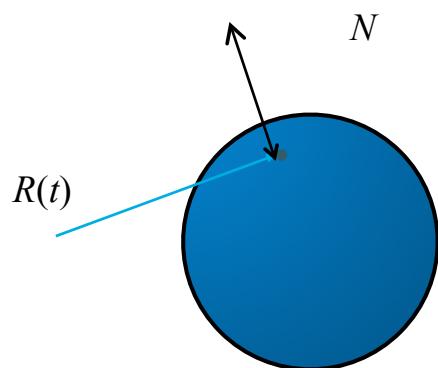
- Em Superfícies Implícitas:
 - A função $f(x, y, z)$ representa um campo de densidade, portanto o gradiente desse campo pode ser tomado como a normal no ponto, ou seja:

$$\begin{aligned} N &= i \frac{\partial f}{\partial x} + j \frac{\partial f}{\partial y} + k \frac{\partial f}{\partial z} \\ &= \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} & \frac{\partial f}{\partial z} \end{bmatrix} \\ &= \nabla f \end{aligned}$$



Vetor Normal

- Exemplo:
- Normal na esfera é dada pelo gradiente no ponto de interseção

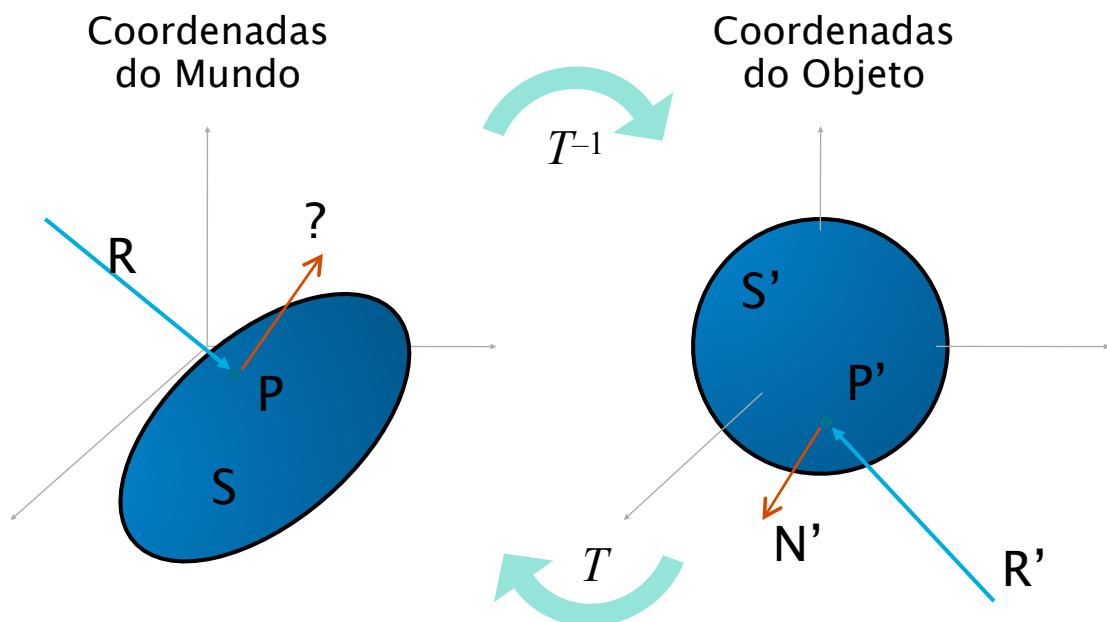


$$N = \begin{bmatrix} \partial f / \partial x \\ \partial f / \partial y \\ \partial f / \partial z \end{bmatrix} = \begin{bmatrix} 2x \\ 2y \\ 2z \end{bmatrix}$$

Interseção com Objetos Transformados

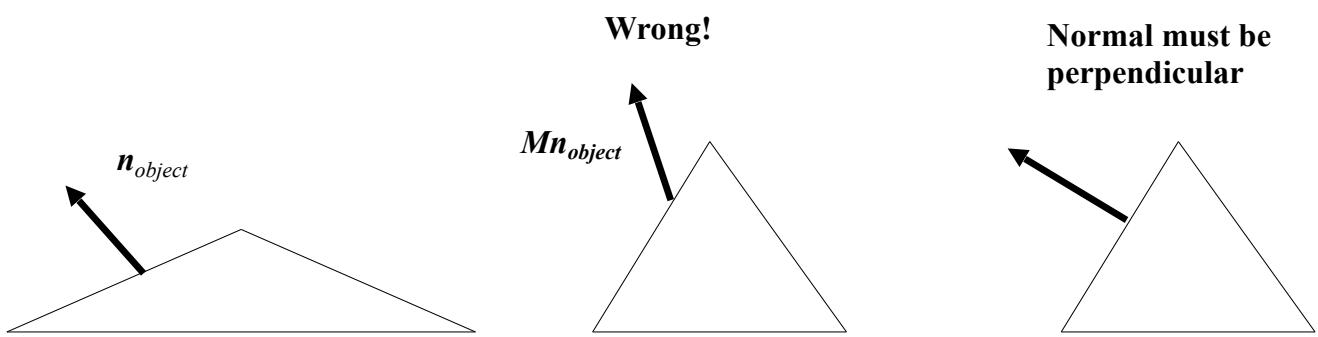
- As rotinas de interseção normalmente lidam com objetos primitivos de tamanho, posição e orientação fixas (ex.: esfera de raio unitário na origem)
 - Para obter objetos genéricos, usa-se transformações lineares afim
 - Para computar a interseção de um raio R com um objeto transformado $S = T S'$:
 - Leva-se o raio para o sistema de coordenadas da primitiva: $R' = T^{-1} R$
 - Computa-se o ponto P' resultante da interseção $R' \times S'$
 - O ponto de interseção é trazido de volta ao sistema de coordenadas do mundo: $P = T P'$

Interseção com Objetos Transformados



Transformando Normais

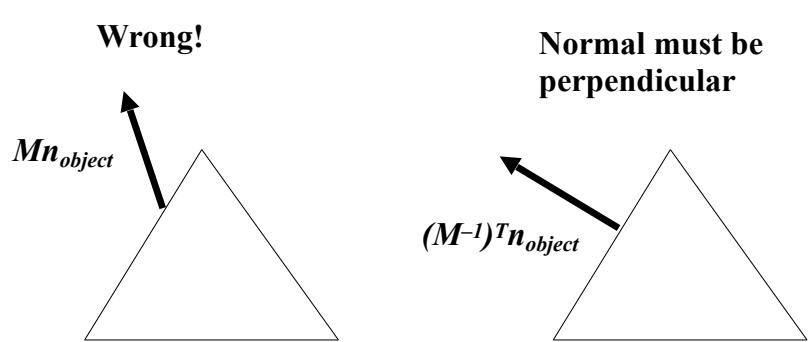
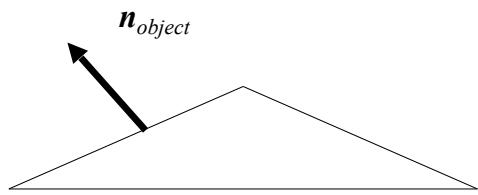
- Ao contrário do que nossa intuição indica,
 - $\mathcal{N}' \neq T\mathcal{N}$
- Por quê?



Transformando Normais

- Se a transformação não envolve deformação, isto é, é composta apenas de transformações rígidas e escalas uniformes, ela pode ser aplicada também à normal
- Para transformações afim genéricas, entretanto,

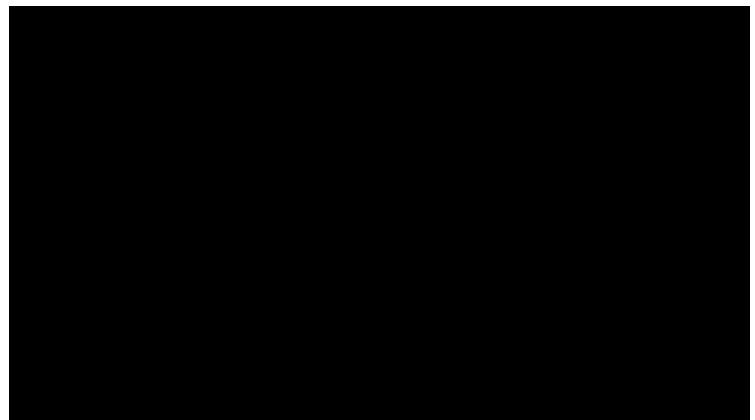
- $\mathbf{N}' = (\mathbf{T}^{-1})^T \mathbf{N}$



Ray Tracing Recursivo

Ray Tracing Recursivo

- Introduzido por Whitted(1980):
 - VAX-11/780. 480x640x9 bpp => 74 minutos

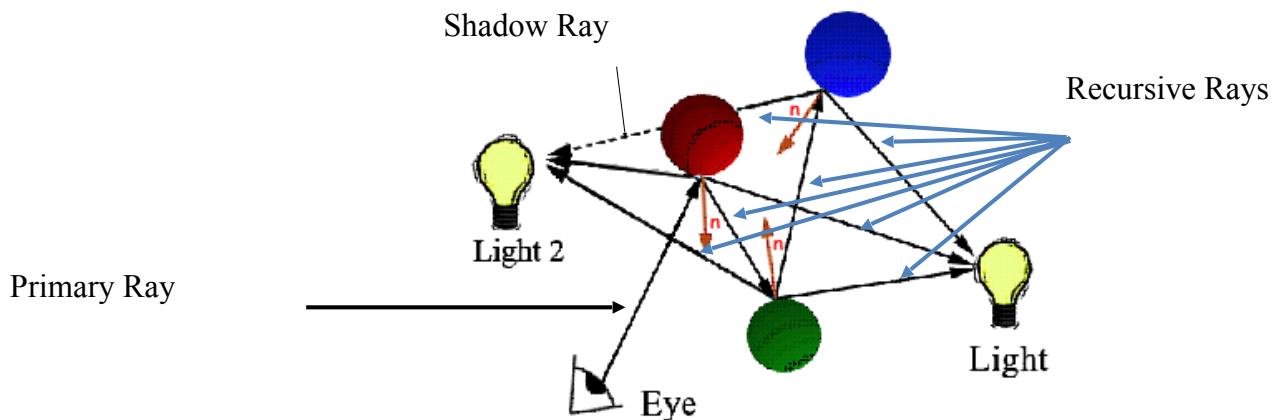


Whitted, T. (1980). **An improved illumination model for shaded display**. Commun. ACM 23, 6 (Jun. 1980), 343-349.

Ray Tracing Recursivo

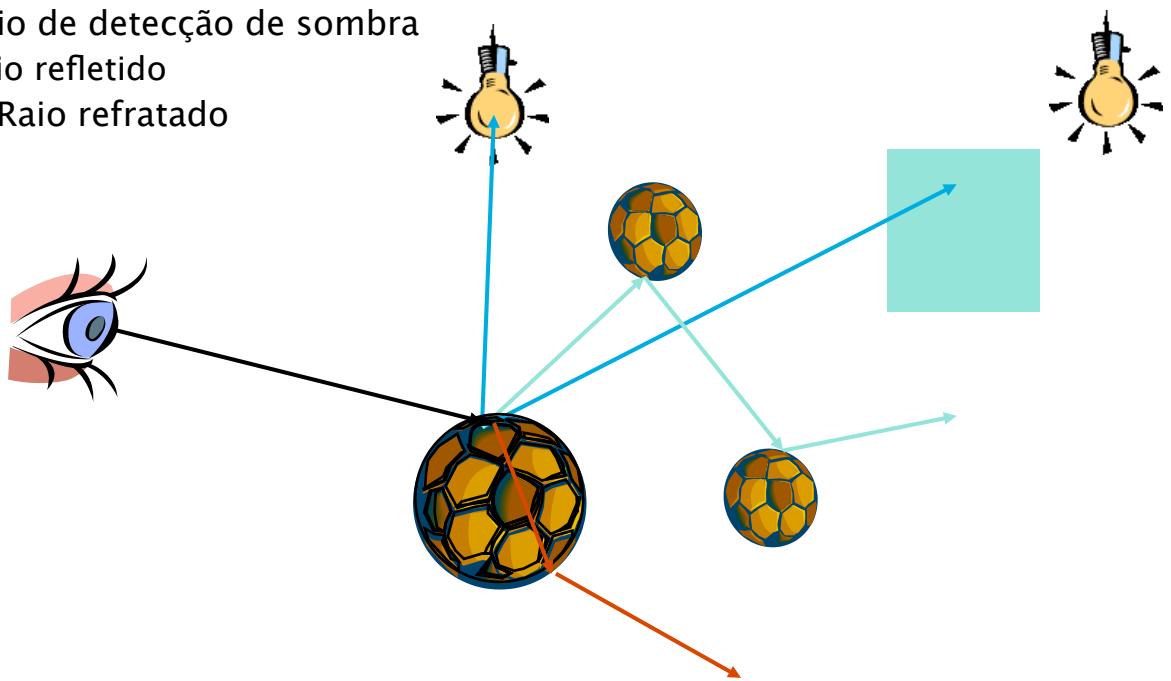
- Baseado no uso do modelo de *Phong* acrescido de outros componentes
 - Gerados pelo acompanhamento recursivo dos raios de luz

$$I_{\lambda} = \underbrace{L_{a\lambda} k_a O_{a\lambda}}_{ambient} + \sum_{lights} f_{att} L_{p\lambda} [\underbrace{k_d O_{d\lambda} (\vec{n} \cdot \vec{l})}_{diffuse} + \underbrace{k_s O_{s\lambda} (\vec{r} \cdot \vec{v})^n}_{specular}] + \underbrace{k_s O_{s\lambda} I_{r\lambda}}_{reflected} + \underbrace{k_t O_{t\lambda} I_{t\lambda}}_{transmitted} + \underbrace{\text{recursive rays}}_{recursive rays}$$



Ray Tracing Recursivo

- Raio de visibilidade
- Raio de detecção de sombra
- Raio refletido
- Raio refratado

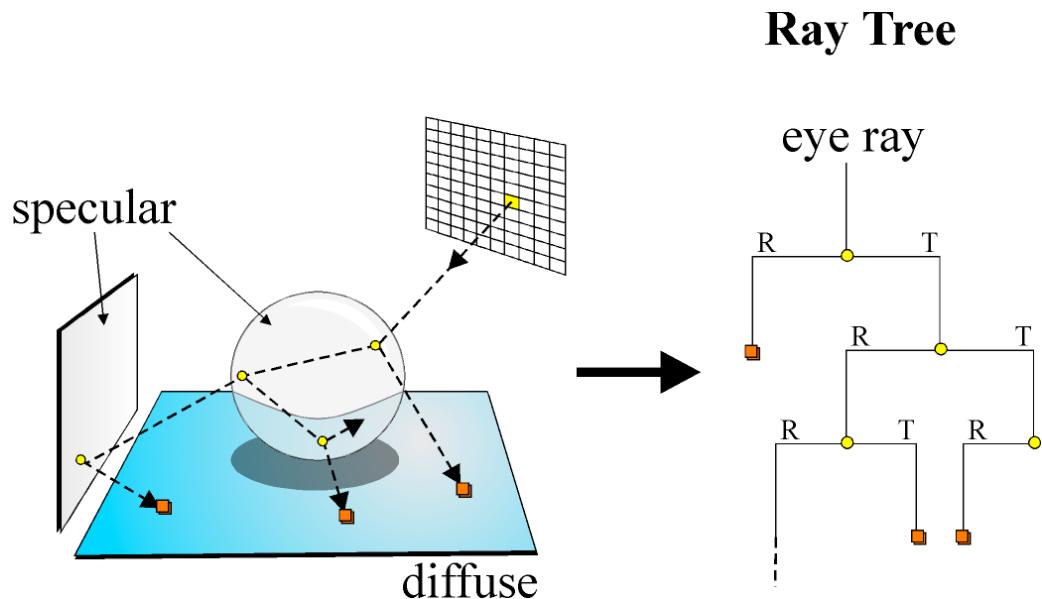


Ray Tracing Recursivo

- Para cada pixel da imagem
 - Calcular raio que passa pelo pixel e pelo olho
 - Determinar objeto atingido pelo raio
 - Ponto de interseção
 - Normal
 - Propriedades de material
 - Propriedades de textura
 - Computar contribuição da iluminação ambiente
 - Para cada fonte de luz, determinar visibilidade (raios de detecção de sombra)
 - Se fonte visível, somar contribuição reflexão difusa
 - Se limite de recursão não foi atingido
 - Somar contribuição reflexão especular acompanhando raio refletido
 - Somar contribuição de transmissão acompanhando raio refratado

Ray Tracing Recursivo

- O processo pode ser visto como a geração de uma árvore de raios



Ray Tracing Recursivo

- Algoritmos:

```
for each pixel in viewport {
    determine eye ray for pixel
    intersection = trace(ray, objects)
    colour = shade(ray, intersection)
}
```

```
Intersection trace(ray, objects) {
    for each object in scene
        intersect(ray, object)
    sort intersections
    return closest intersection
}
```

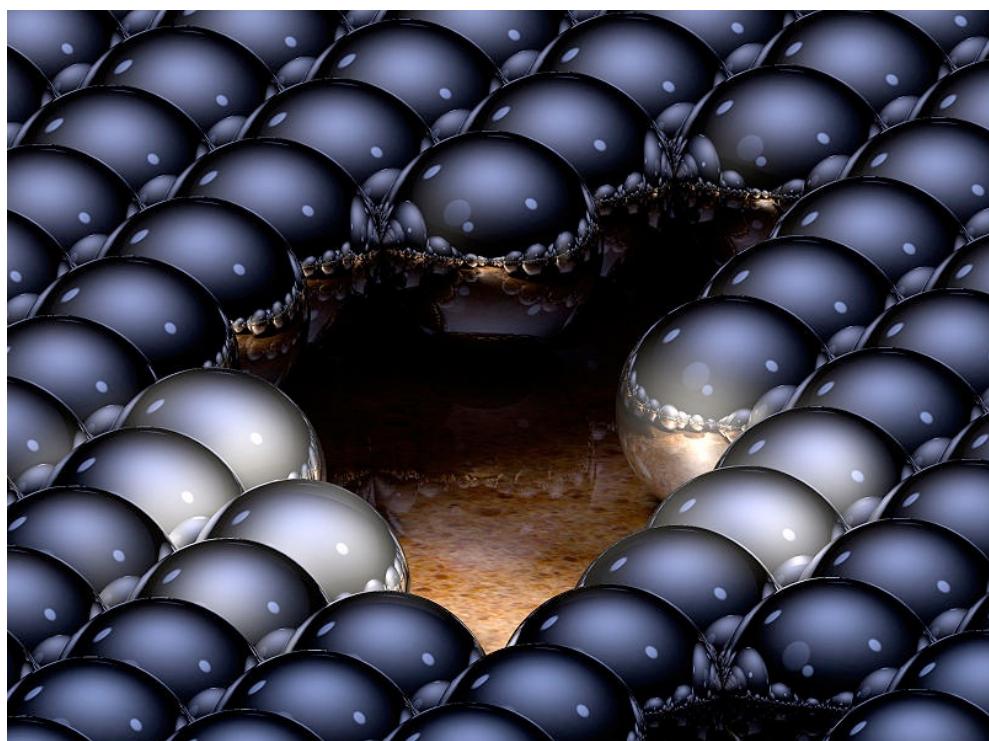
Ray Tracing Recursivo

```
colour shade(ray, intersection) {
    if no intersection
        return background colour

    for each light source
        if(visible)
            colour += Phong contribution

    if(recursion level < maxlevel) {
        ray = reflected ray
        intersection = trace(ray, objects)
        colour += ρrefl*shade(ray, intersection)
        ray = transmitted ray
        intersection = trace(ray, objects)
        colour += ρtrans* shade(ray, intersection)
    }
    return colour
}
```

Ray Tracing Recursivo

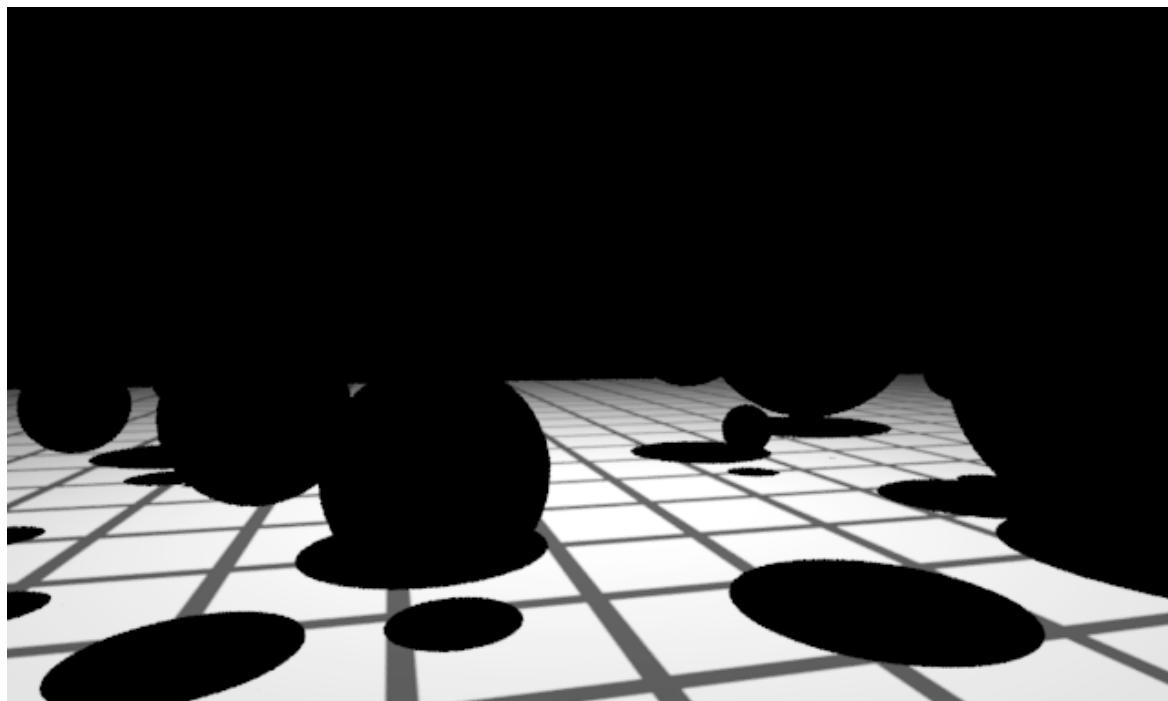


Ray Tracing Recursivo



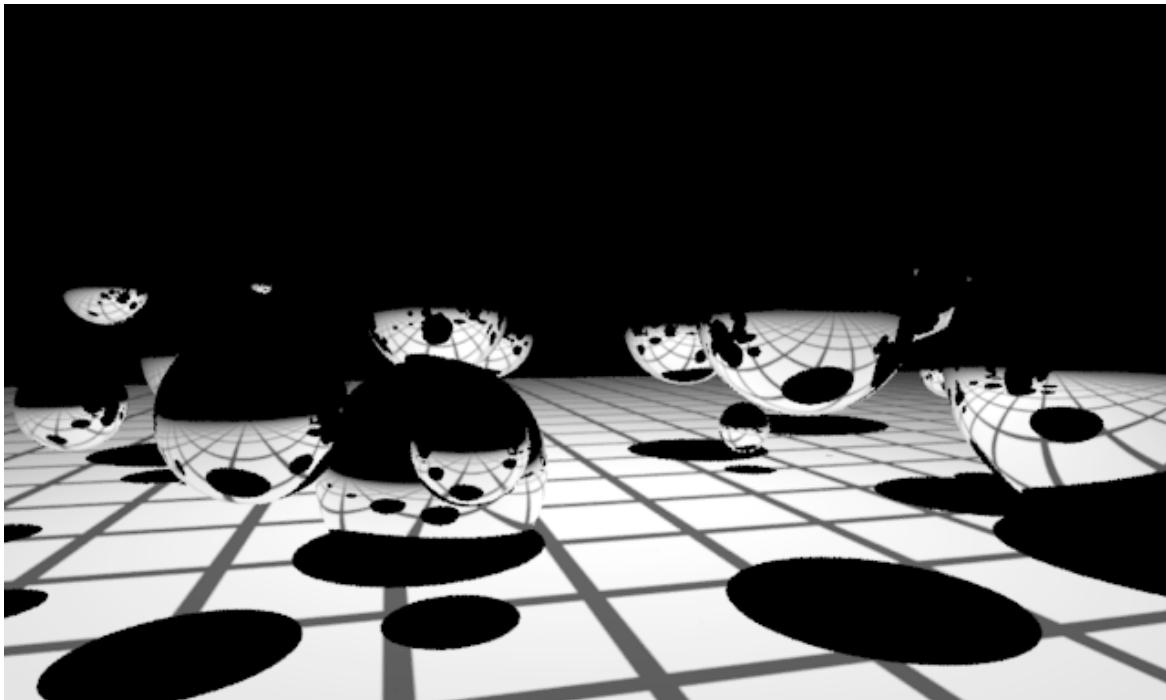
Ray Tracing Recursivo

- Nivel 0



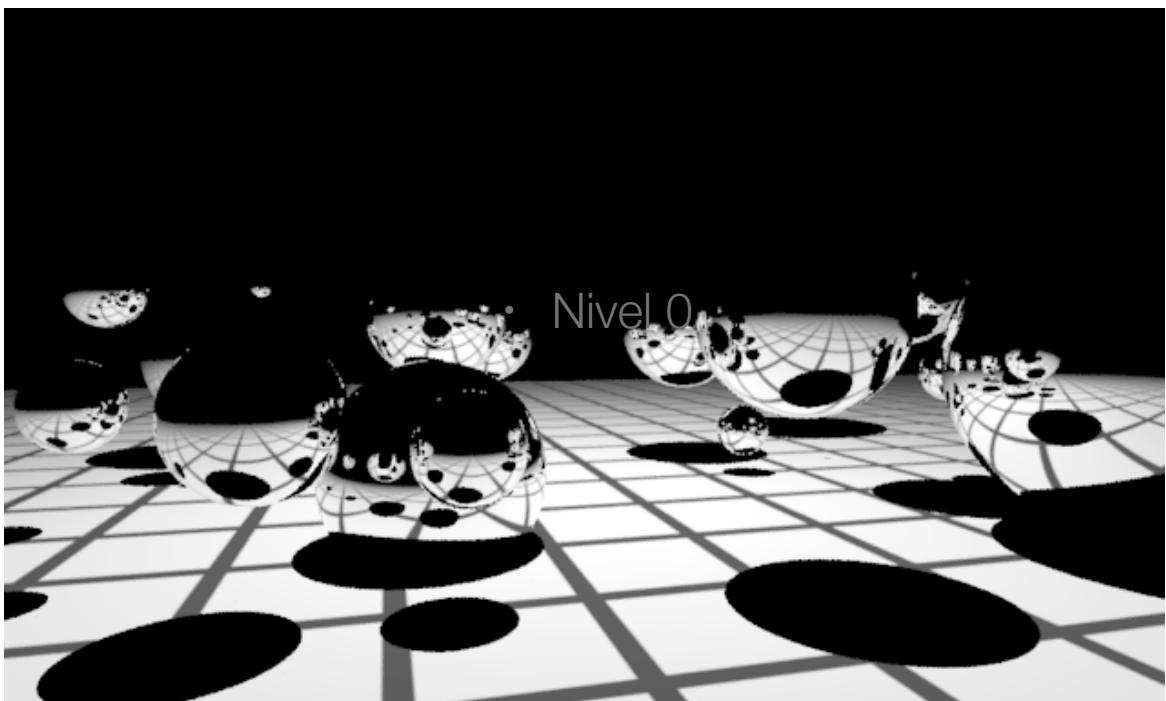
Ray Tracing Recursivo

- Nivel 1



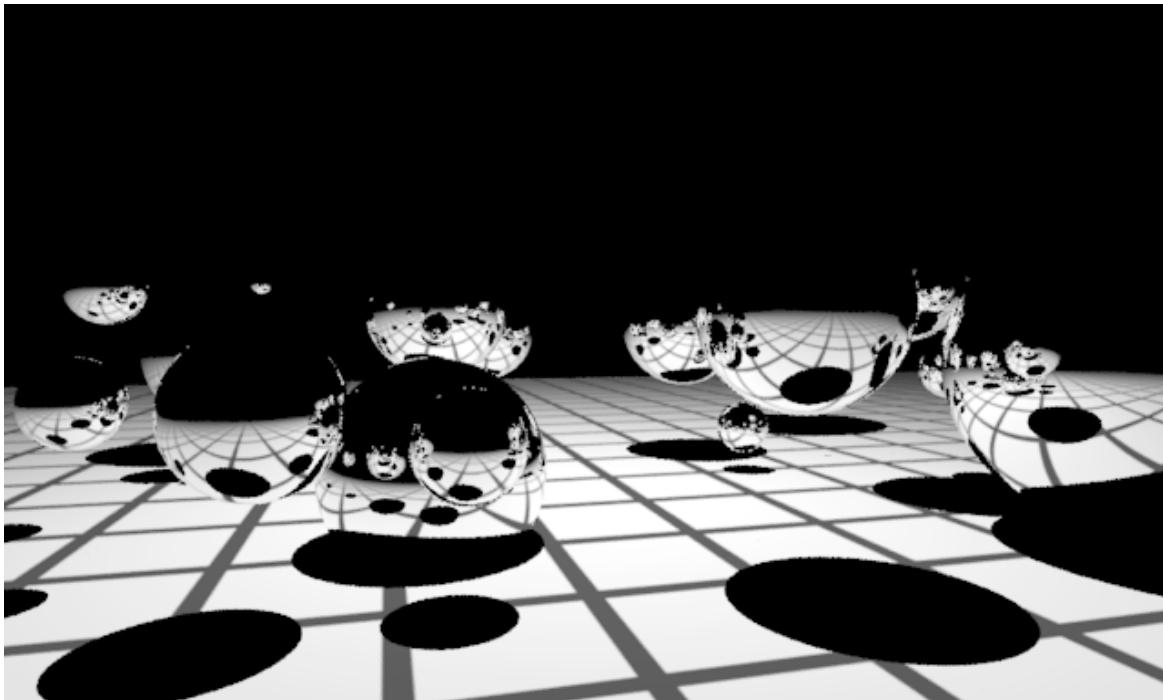
Ray Tracing Recursivo

- Nivel 2



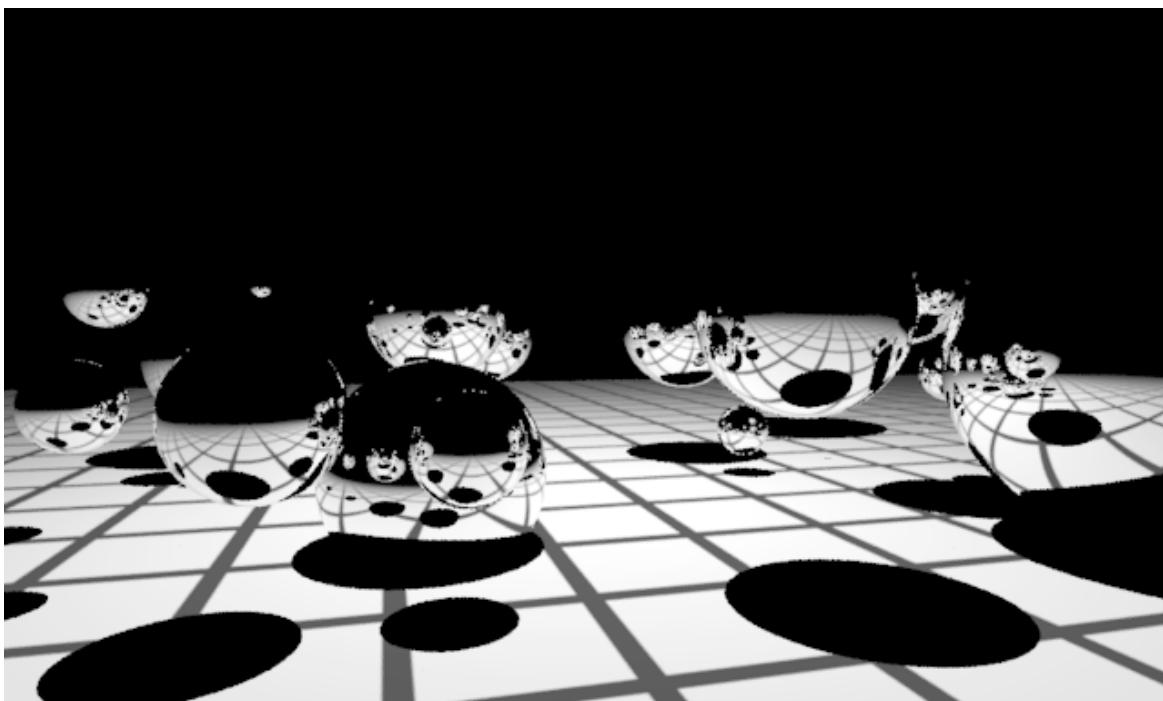
Ray Tracing Recursivo

- Nivel 5



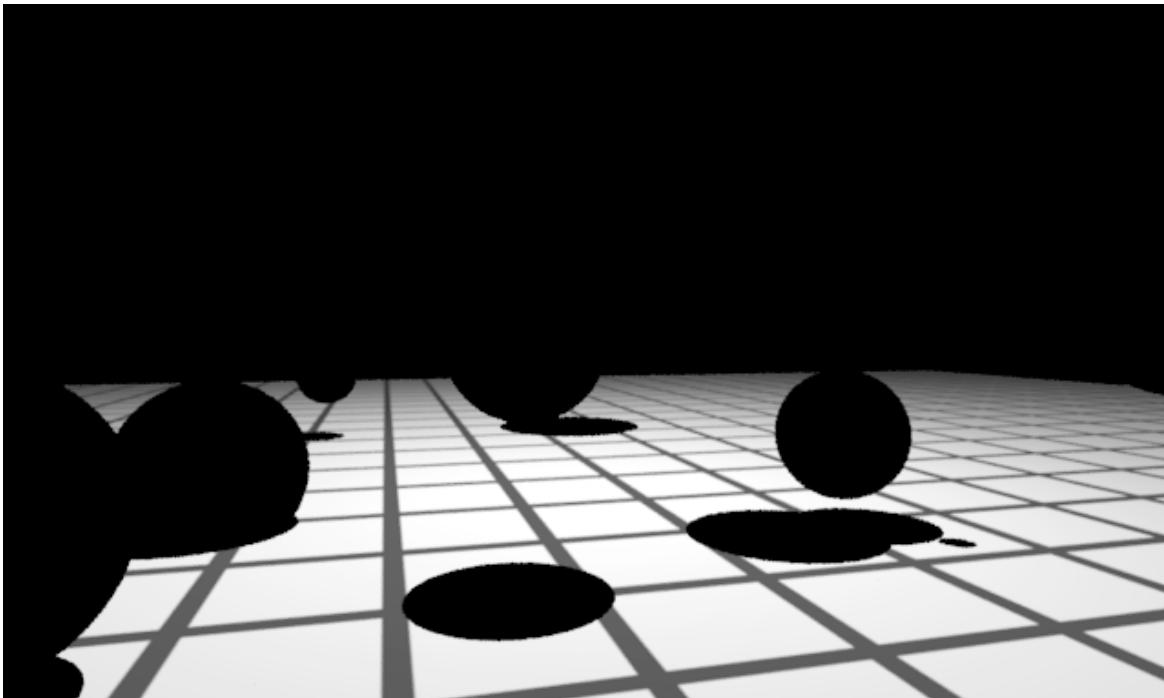
Ray Tracing Recursivo

- Nivel 10



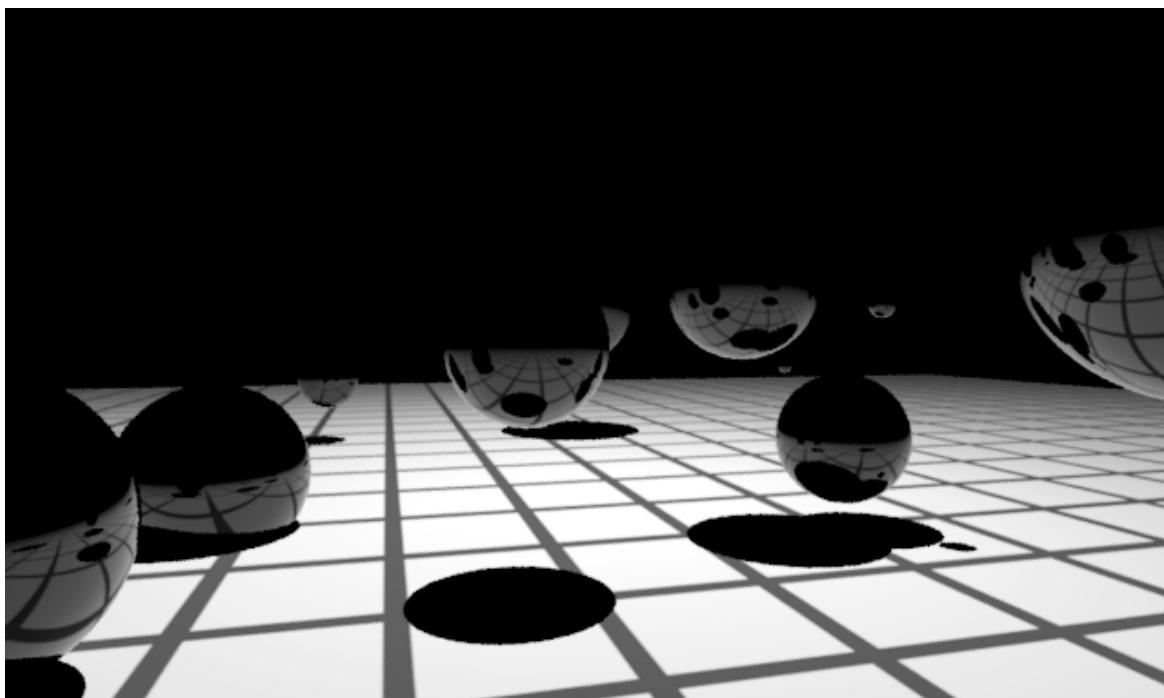
Ray Tracing Recursivo

- Nivel 0



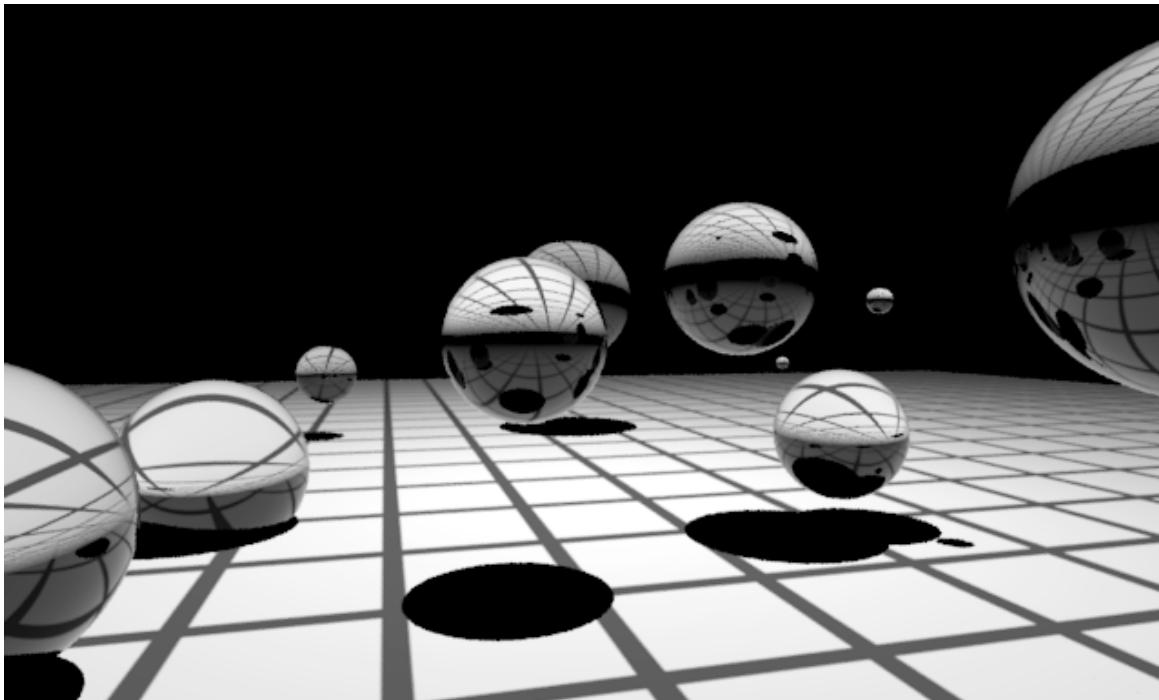
Ray Tracing Recursivo

- Nivel 1



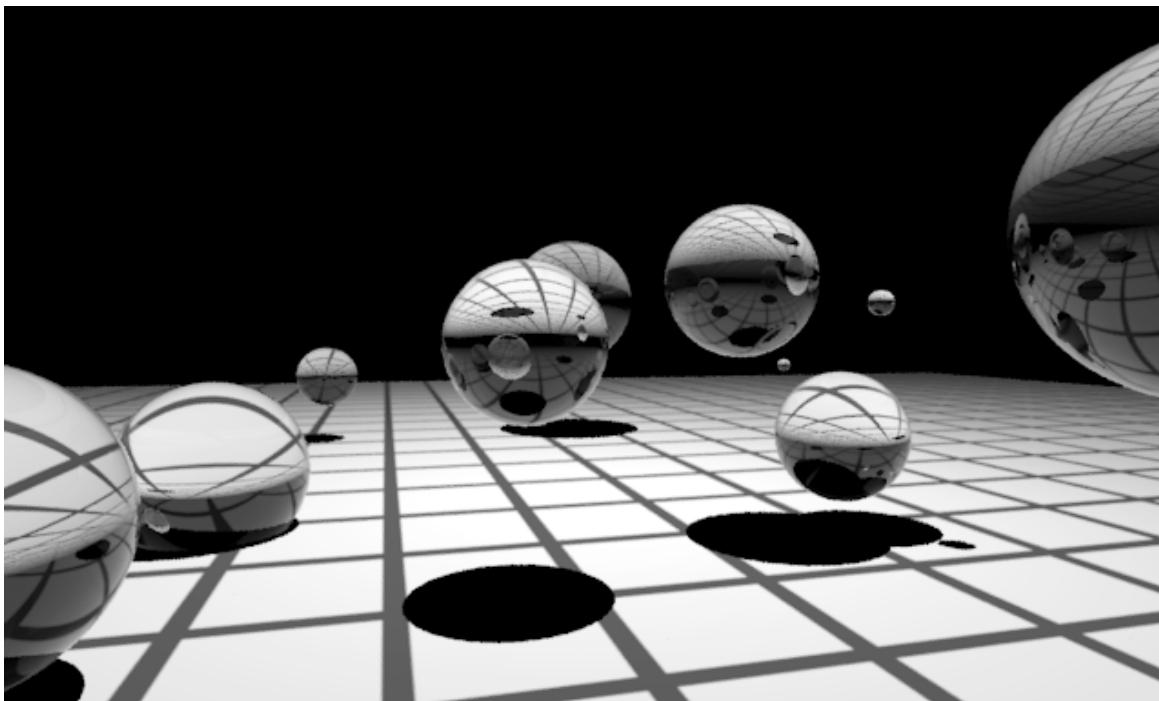
Ray Tracing Recursivo

- Nivel 2



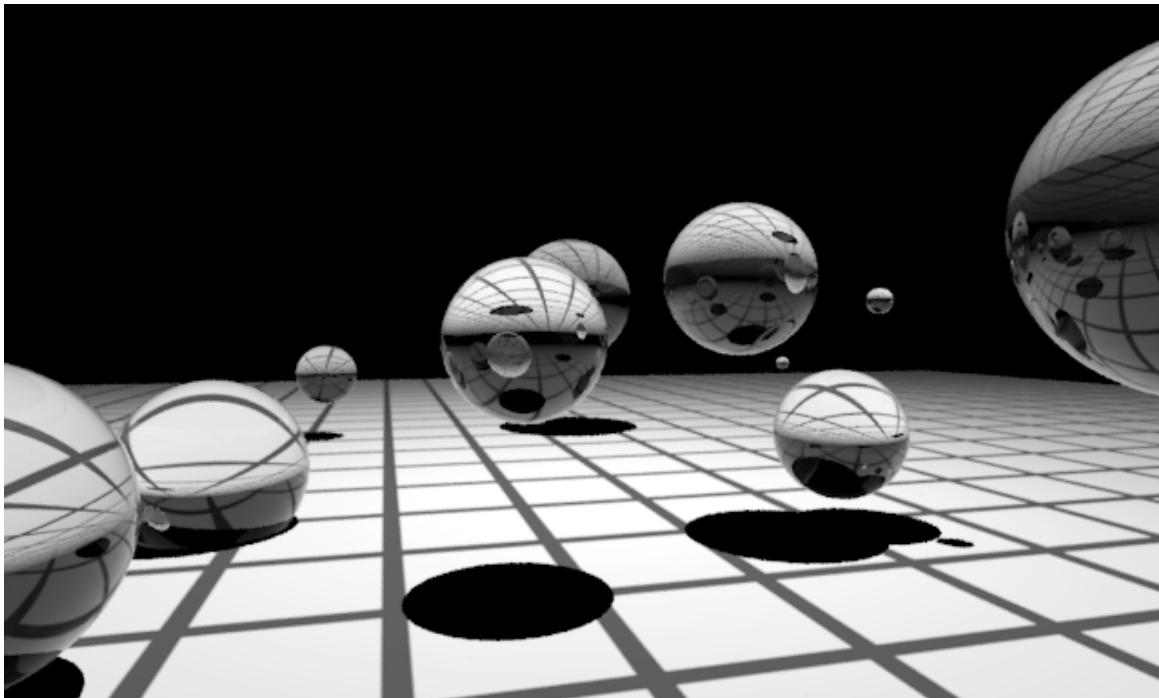
Ray Tracing Recursivo

- Nivel 5



Ray Tracing Recursivo

- Nivel 10



Ray Tracing Recursivo

- Nivel 10



Técnicas de Aceleração

Técnicas de Aceleração

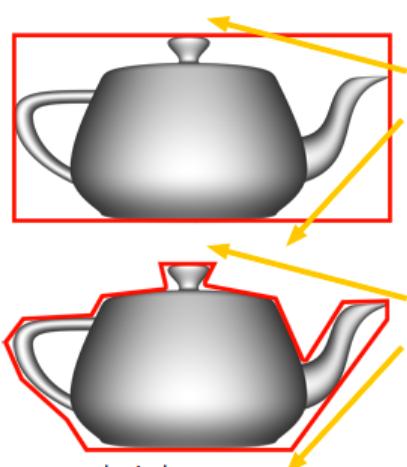
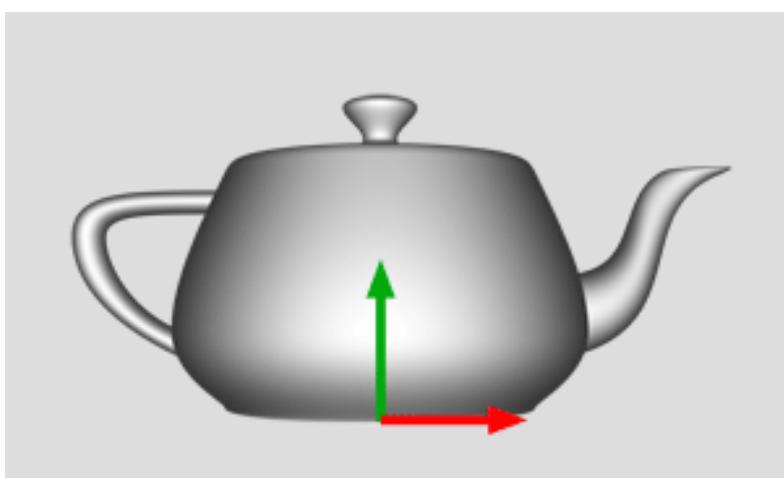
- O que torna o algoritmo *Ray Tracing* computacionalmente custoso?
 - Número de interseções entre raios e objetos
 - Custo de cada cálculo de interseção

Técnicas de Aceleração

- Reduzir o número de interseções
 - Hierarquias de volumes limitantes
 - Subdivisão espacial
- Aceleração dos cálculos de interseções
 - Volumes limitantes
 - Especialmente superfícies paramétricas
- Determinar mais rapidamente o primeiro objeto interceptado (visibilidade)
 - Usar hardware gráfico (*z-buffer*)

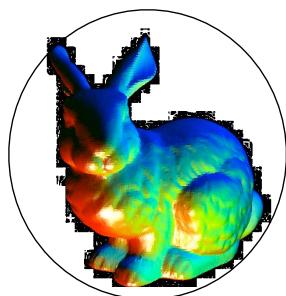
Volumes Limitantes

- Objetos complexos são envolvidos em objetos simples cujo cálculo de interseção é + fácil
 - Serve para eliminar objetos que não interceptam raio

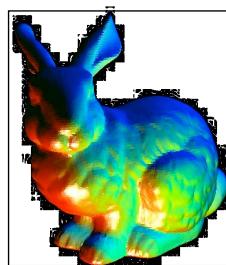


Volumes Limitantes

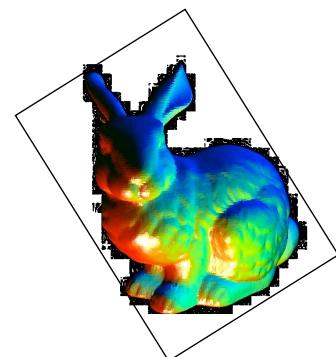
- O tipo de volume limitante pode se ajustar melhor ao objeto



Esfera
+Rápida, mas
nouco iusta



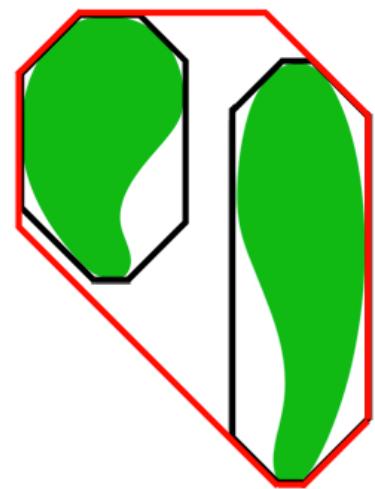
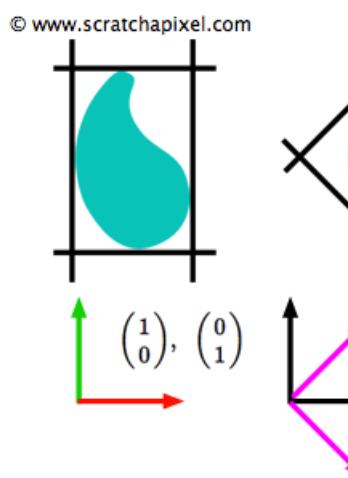
*Caixa Alinhada
com Eixos:*
Rápido, iusto



Caixa Orientada
Rápida, +justa

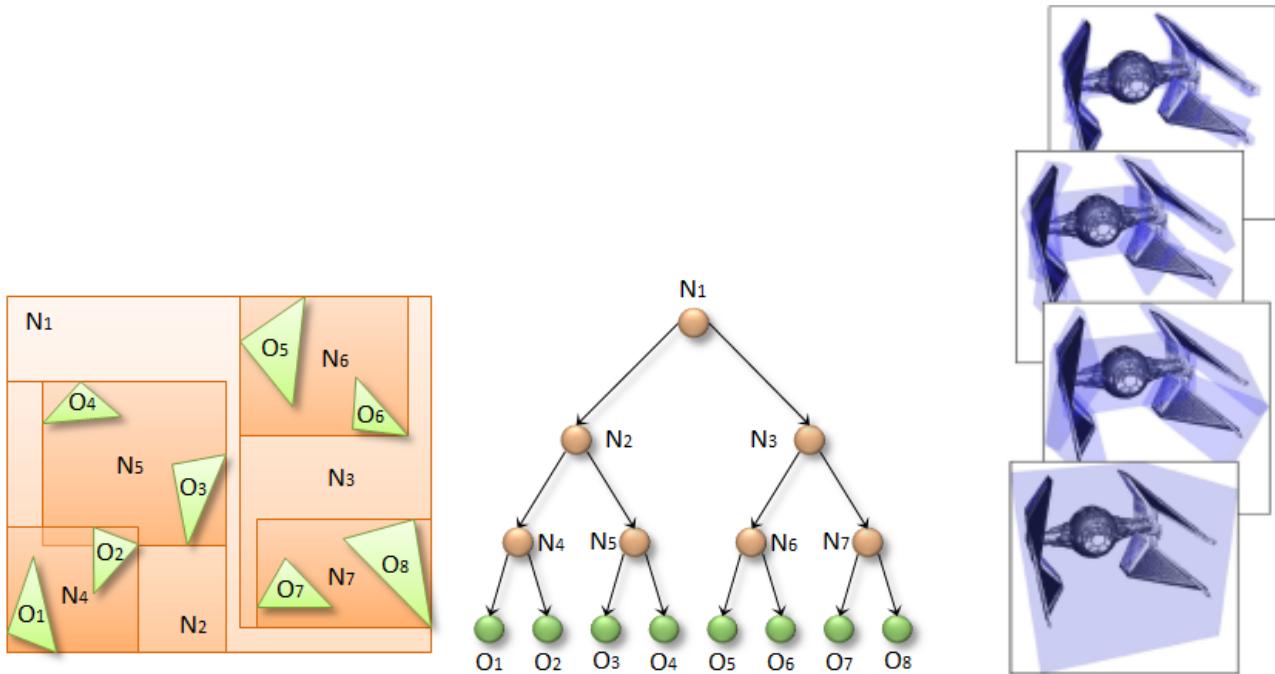
Volumes Limitantes

- Quanto mais justo o volume mais complexo seu calculo



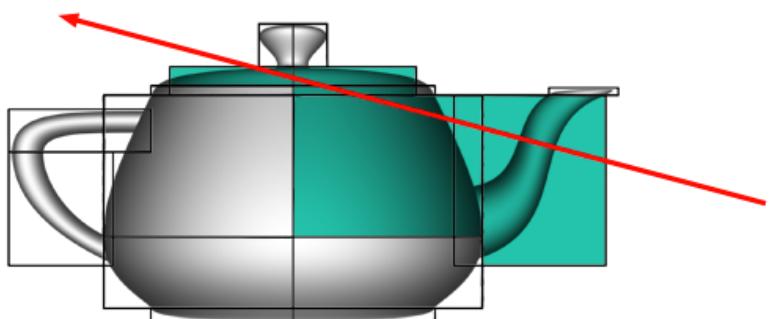
Hierarquias de Volumes

- Podem ser aplicadas aos próprios objetos ou a cenas

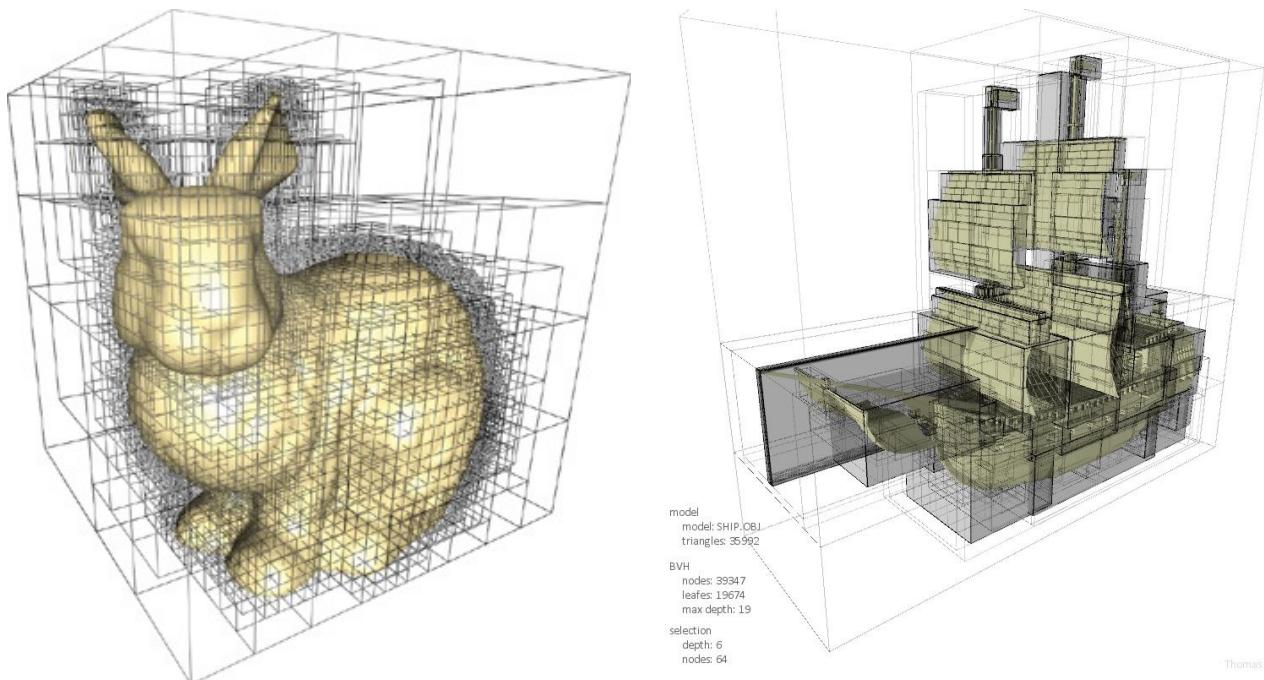


Hierarquias de Volumes

- Aplicadas aos objetos:
 - Raios são recursivamente testados com relação aos nós da hierarquia
 - Nós + próximos do olho são testados primeiro
 - Nós vazios são trivialmente rejeitados

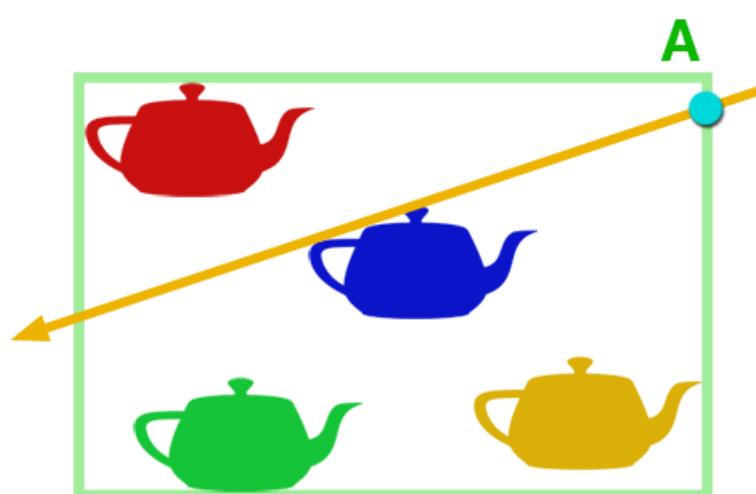


Hierarquias de Volumes



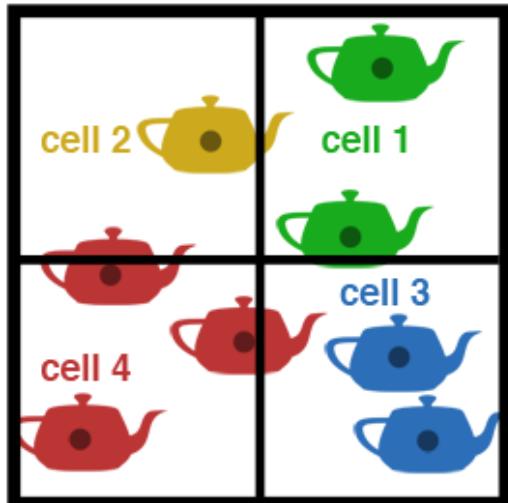
Hierarquias de Volumes

- Aplicadas a cena

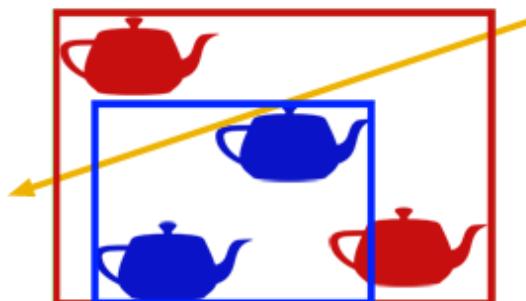


Hierarquias de Volumes

- Hierarquias podem ser regulares ou adaptativas



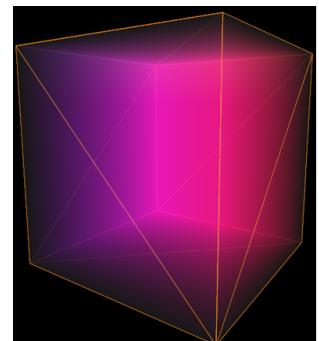
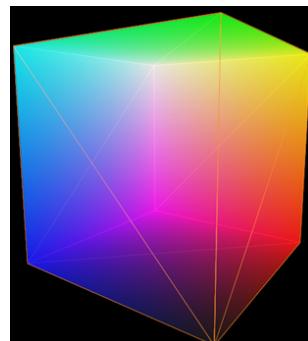
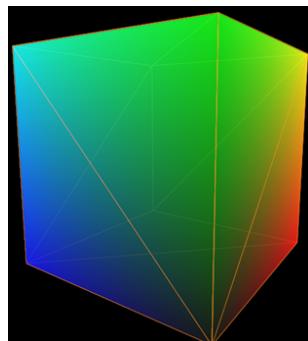
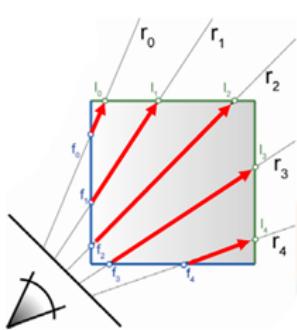
© www.scratchapixel.com



© www.scratchapixel.com

Usando Hardware para determinar Visibilidade

- Hardware é geralmente mais rápido que software
 - Desenha-se os objetos usando acelerador gráfico
- O valor de profundidade armazenado no *z-buffer* pode ser usado para determinar as coordenadas do mundo dos objetos mais próximos
- Um teste exato é feito apenas sobre esses objetos



A seguir...
Avaliação e Seminários