ILUMINAÇÃO E SOMBREAMENTO

Adair Santa Catarina
Curso de Ciência da Computação
Unioeste – Campus de Cascavel – PR

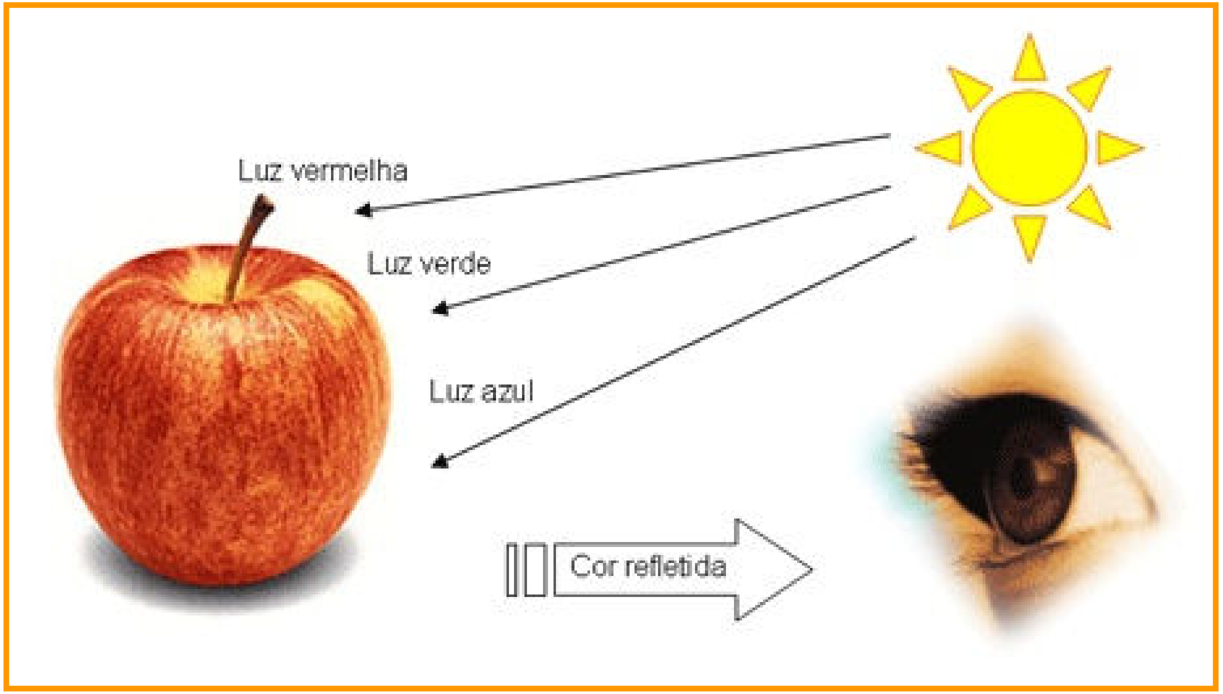
Jan/2021

Importância da Iluminação em CG

A posição, orientação, características da luz e seus efeitos sobre as superfícies dos objetos são essenciais para obtenção do aspecto realístico da cena sintética.

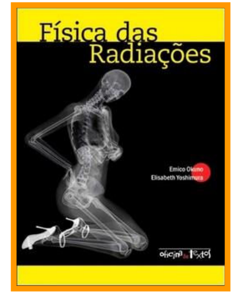
Importância da Iluminação em CG

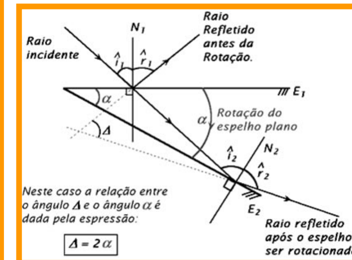
A cor que percebemos de um objeto é resultado da reflexão dos diversos estímulos luminosos que interagem com o material que o compõe.



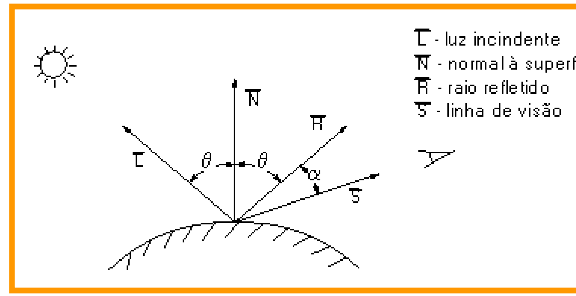
Ótica + Física das Radiações

As regras da ótica e a física das radiações explicariam a interação da luz nos objetos.



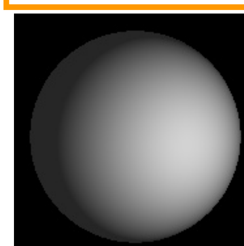


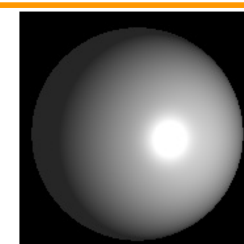
Na prática, devido à complexidade ou inexistência de modelos completos, utilizam-se modelos simplificados.



Algumas destas simplificações não têm fundamentação teórica, mas produzem resultados satisfatórios.

$$I_s = I_l \cdot W(i,\lambda) \cdot \cos^n \alpha$$





Luz Ambiente

Corresponde à cor intrínseca do objeto, fruto da múltiplas reflexões da luz nas muitas superfícies presentes no ambiente.

$$I_a = I_{la} \cdot K_a$$

I_a = Intensidade ambiente;

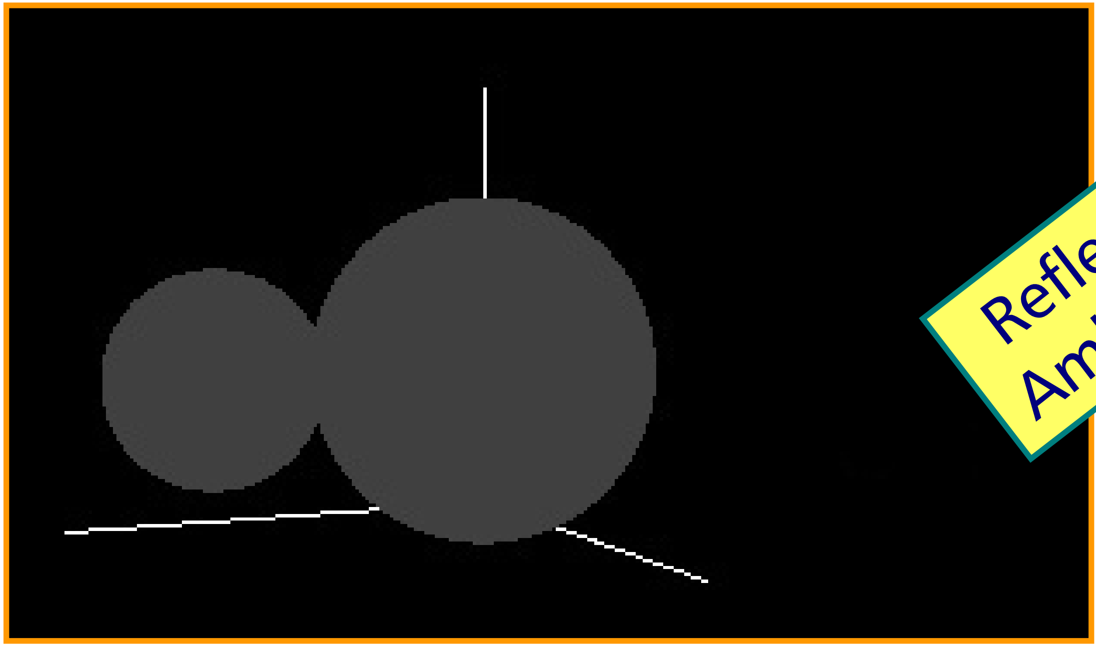
I_{la} = Intensidade da luz ambiente;

K_a =Coeficiente de reflexão ambiente ($0 \leq K_a \leq 1$).

Luz Ambiente

Pode-se imaginar como um modelo onde não há fonte de luz externa; um mundo irreal onde cada objeto possui luz própria.

O coeficiente de reflexão ambiente (Ka) é empírico e não corresponde diretamente a qualquer propriedade física real dos materiais.



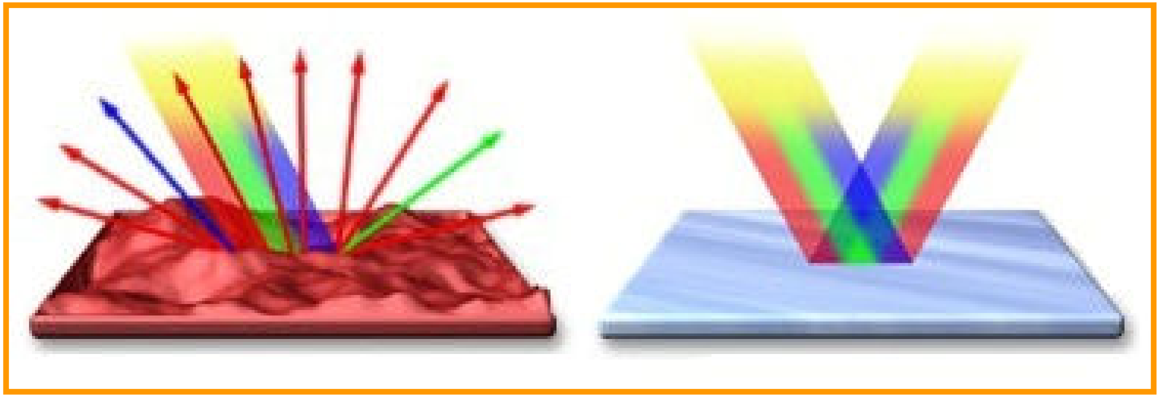
Reflexão Ambiente

Reflexão Difusa x Reflexão Especular

Uma superfície é Lambertina (reflexão difusa) se é capaz de refletir a luz igualmente em todas as direções, como as superfícies foscas.

Superfícies brilhantes como metal polido ou uma maçã apresentam reflexão especular, pois, de acordo com a direção da observação, pode-se ver um brilho ou ponto de luz concentrada.

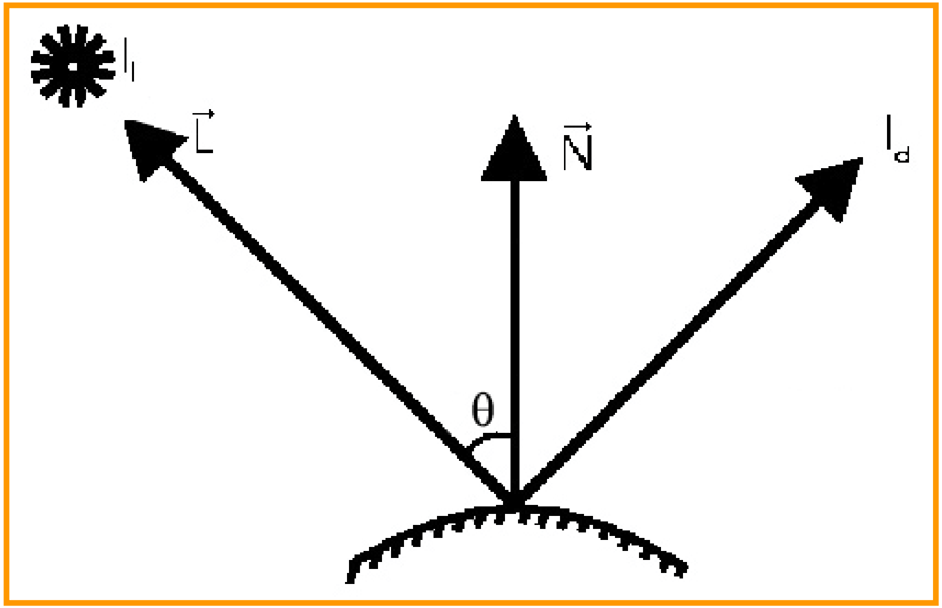
Difusa



Especular

Reflexão Difusa

A intensidade da reflexão difusa (I_d) é inversamente proporcional ao ângulo ($0 \leq \theta \leq \pi/2$) formado entre o vetor na direção do feixe de luz incidente (***L***) e o vetor normal à superfície (***N***).



$$I_d = I_l \cdot K_d \cdot \cos\theta$$

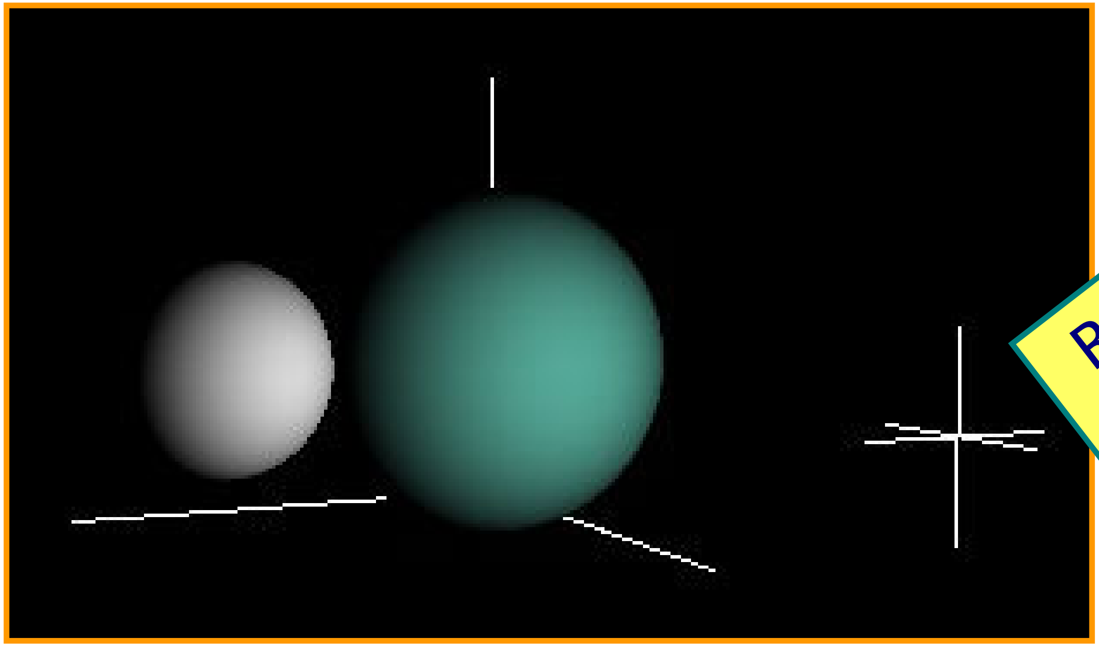
$$\cos\theta = \hat{N} \cdot \hat{L}$$

$$I_d = I_l \cdot K_d \cdot \left(\hat{N} \cdot \hat{L}\right)$$

I_l = Intensidade da fonte luminosa; K_d =Coeficiente de reflexão difusa ($0 \leq K_d \leq 1$).

Reflexão Difusa

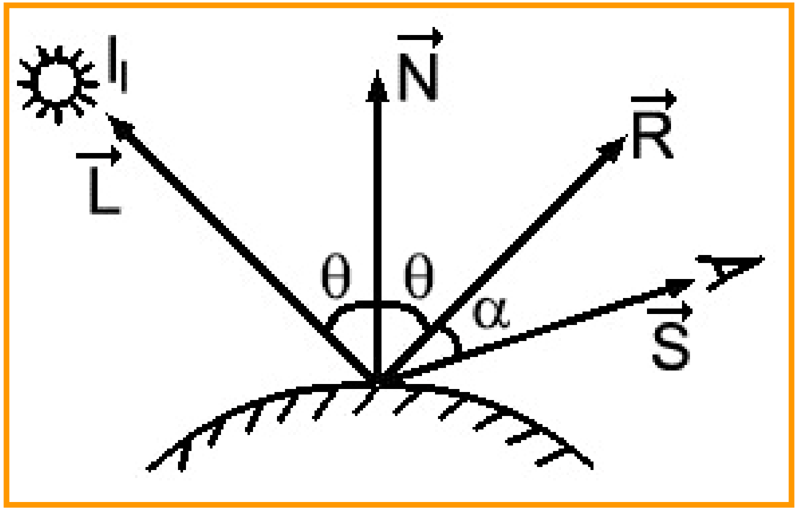
Inversamente proporcional ao ângulo de incidência do feixe de luz porque à medida que θ aumenta, o valor de $\cos(\theta)$ diminui.



Reflexão Difusa

Reflexão Especular

A reflexão especular é resultado da reflexão total ou quase total da luz incidente em uma região concentrada ao redor do ângulo de reflexão especular ($0 \leq \alpha \leq \pi/2$).



Bui Tuong Phong:

$$I_s = I_l \cdot K_s \cdot \cos^n \alpha$$

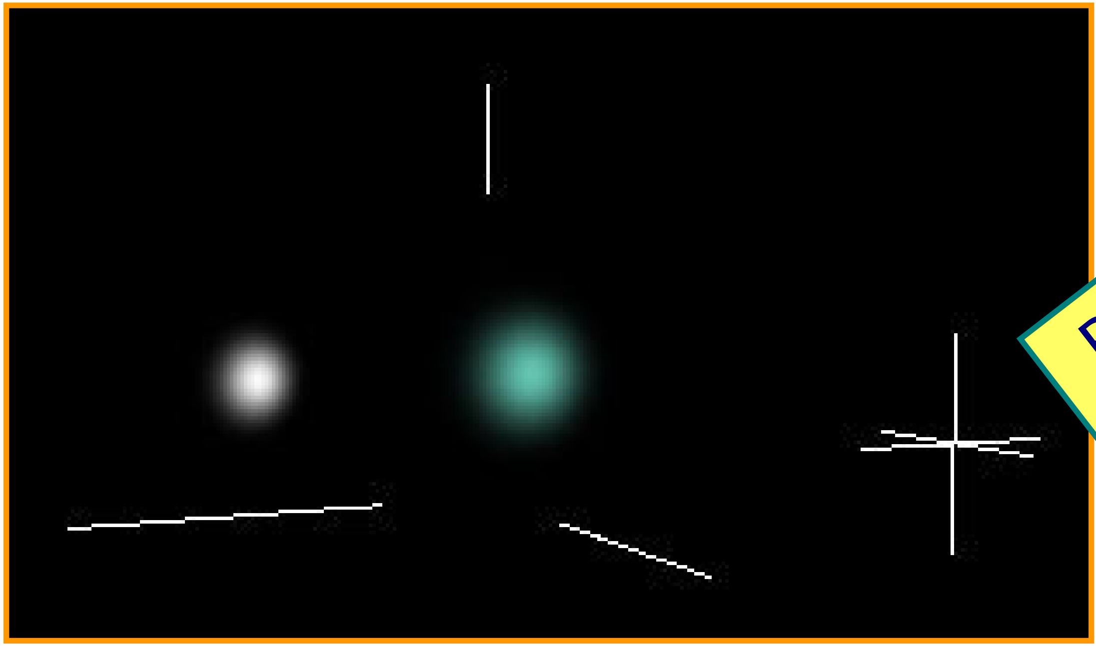
$$\cos\alpha = \hat{R} \cdot \hat{S}$$

$$\hat{R} = \left(2\hat{L} \cdot \hat{N}\right) \cdot \hat{N} - \hat{L}$$

R = Vetor reflexão; ***S*** = Vetor direção de observação;
n = aproximação da distribuição espacial da luz refletida especularmente;
K_s = Coeficiente de reflexão especular ($0 \leq K_s \leq 1$).

Reflexão Especular

Valores grandes de ***n*** caracterizam distribuições espectrais de metais e outras superfícies especulares. Valores pequenos de ***n*** caracterizam superfícies não-metálicas ou opacas.

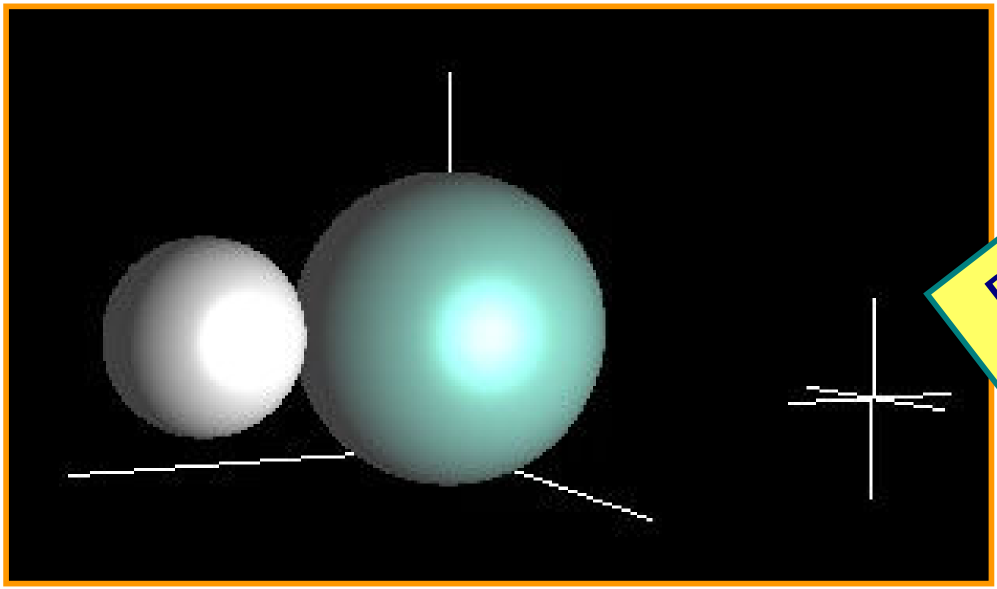


Reflexão Especular

Função de Iluminação

Combinando-se o efeitos dos três modelos de reflexão define-se a **função de iluminação**, para uma única fonte de luz.

$$I_t = I_{la} K_a + I_l \left(K_d \left(\hat{N} \cdot \hat{L} \right) + K_s \left(\hat{R} \cdot \hat{S} \right)^n \right)$$



Reflexão Total

Atenuação das Fontes Luminosas

Na função de iluminação anterior objetos a diferentes distâncias da fonte luminosa recebem a mesma intensidade luminosa, o que não está de acordo com a realidade.

Fator de atenuação

$$I_t = I_{la}K_a + f_{att} \cdot I_l\left(K_d\left(\hat{N}\cdot\hat{L}\right) + K_s\left(\hat{R}\cdot\hat{S}\right)^n\right)$$

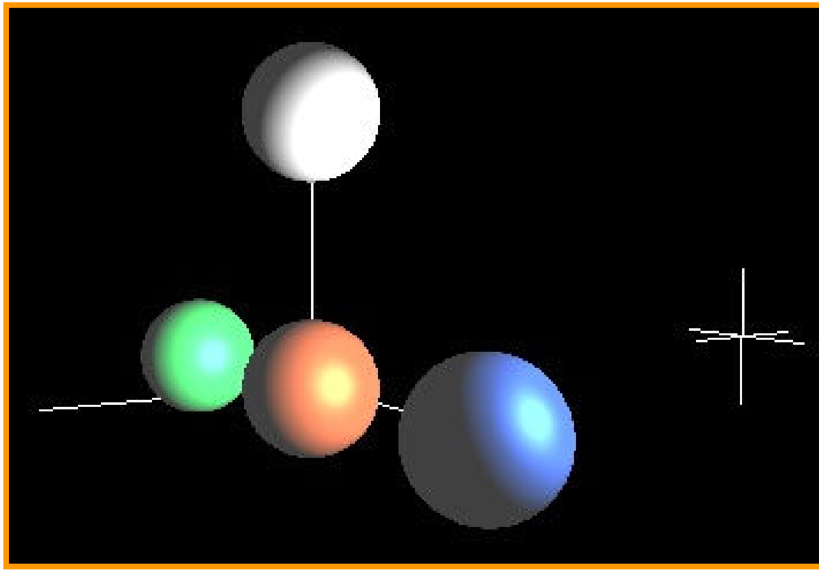
$$f_{att} = \min\left(\frac{1}{d_L^2}, 1\right)$$

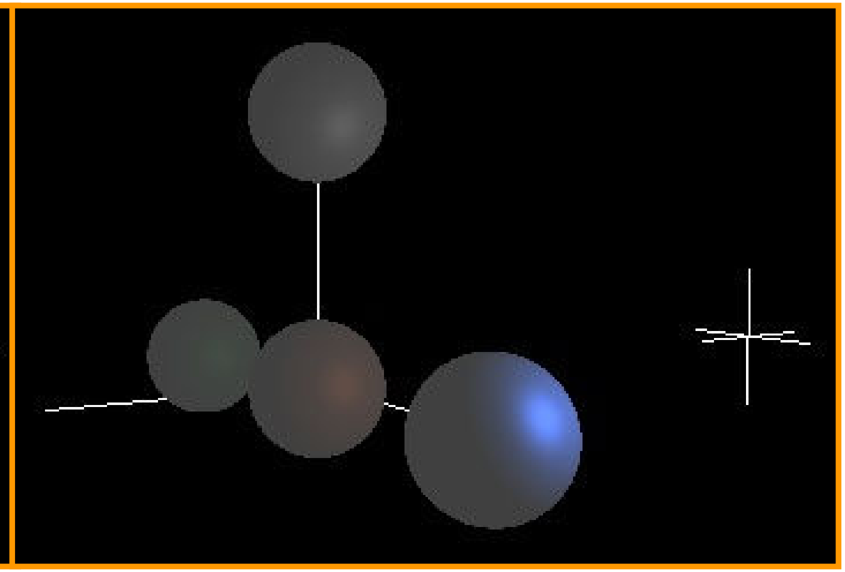
$$f_{att} = \min\left(\frac{1}{c_1 + c_2 d_L + c_3 d_L^2}, 1\right)$$

d_L^2 = distância entre o objeto e a fonte luminosa.

c_1, c_2 e c_3 = constantes arbitradas associadas com a fonte luminosa.

Efeito da Atenuação





Sem atenuação

Com atenuação ($1/d_L^2$)

Considerações sobre Cores

A função de iluminação estudada até o momento está definida apenas para iluminação monocromática.

Como incorporar cores na função de iluminação?

Definem-se os parâmetros relacionados às fontes luminosas e aos materiais que compõem os objetos como tuplas no espaço de cores, geralmente o RGB.

$I_{la} = (I_{laR}, I_{laG}, I_{laB})$ → Luz ambiente

$I_l = (I_{lR}, I_{lG}, I_{lB})$ → Luzes pontuais

$K_a = (K_{aR}, K_{aG}, K_{aB})$

$K_d = (K_{dR}, K_{dG}, K_{dB})$ → Materiais

$K_s = (K_{sR}, K_{sG}, K_{sB}, n)$

Função de Iluminação (Cores e Luzes)

Uma função de iluminação é escrita para cada canal do sistema de cores utilizado. No caso do sistema RGB, tem-se:

$$I_{tR} = K_{aR} I_{laR} + \sum_{i=1}^{n} f_{att}(d_i) \cdot I_{lRi} \left[K_{dR} \left(\hat{N} \cdot \hat{L}_i \right) + K_{sR} \left(\hat{R}_i \cdot \hat{S} \right)^n \right]$$

$$I_{tG} = K_{aG} I_{laG} + \sum_{i=1}^{n} f_{att}(d_i) \cdot I_{lGi} \left[K_{dG} \left(\hat{N} \cdot \hat{L}_i \right) + K_{sG} \left(\hat{R}_i \cdot \hat{S} \right)^n \right]$$

$$I_{tB} = K_{aB} I_{laB} + \sum_{i=1}^{n} f_{att}(d_i) \cdot I_{lBi} \left[K_{dB} \left(\hat{N} \cdot \hat{L}_i \right) + K_{sB} \left(\hat{R}_i \cdot \hat{S} \right)^n \right]$$

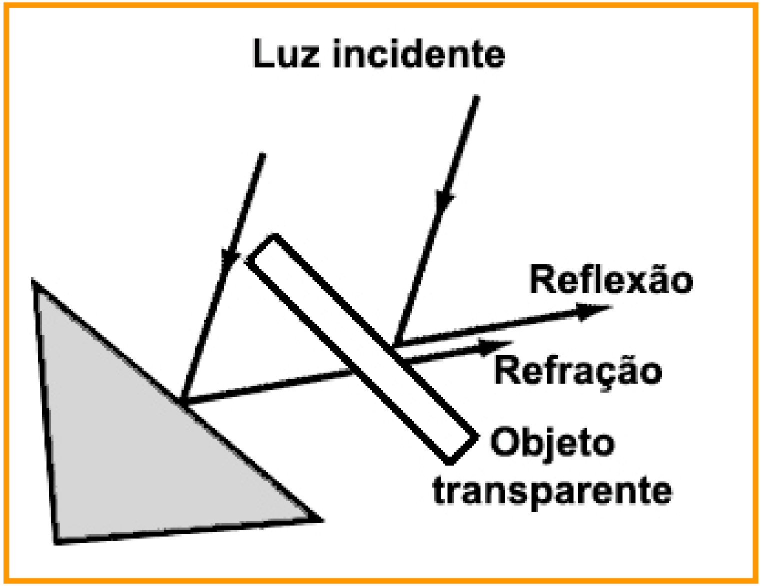
Lembrando que:

$$0 \leq \theta \leq \frac{\pi}{2} \Rightarrow \left(\hat{N} \cdot \hat{L}_i \right) \geq 0)$$

$$0 \leq \alpha \leq \frac{\pi}{2} \Rightarrow \left(\hat{R}_i \cdot \hat{S} \right) \geq 0)$$

Transparência

Superfícies transparentes geralmente refletem e refratam partes da luz incidente. A porção refratada depende do grau de transparência das superfícies e se há fontes de luz ou outras superfícies iluminadas atrás delas.



Transparência

Modifica-se a função de iluminação para computar a quantidade de luz refletida em objetos que estão atrás da superfície transparente.

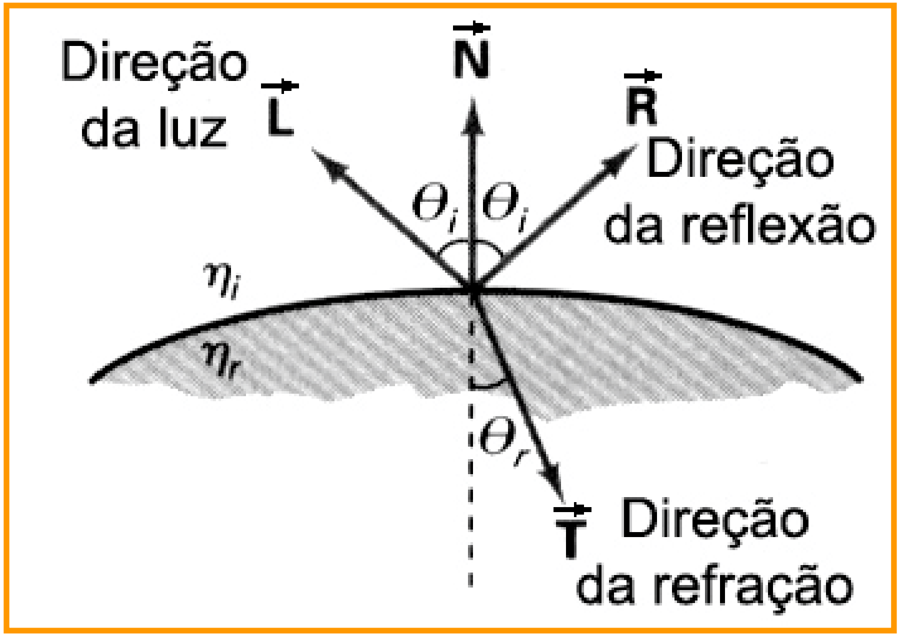
Superfícies transparentes ← reflexão difusa e especular.

Efeito difuso → importante pois a luz refratada embaça a imagem do objeto de fundo.

A refração difusa é computada decrementando a intensidade da luz refratada e aumentando a intensidade luminosa numa porção limitada da superfície. Isso é computacionalmente caro e, por isso, muitos modelos consideram apenas o efeito especular.

Transparência Realista

Considera a mudança na direção dos raios luminosos ocasionados pela refração da luz.



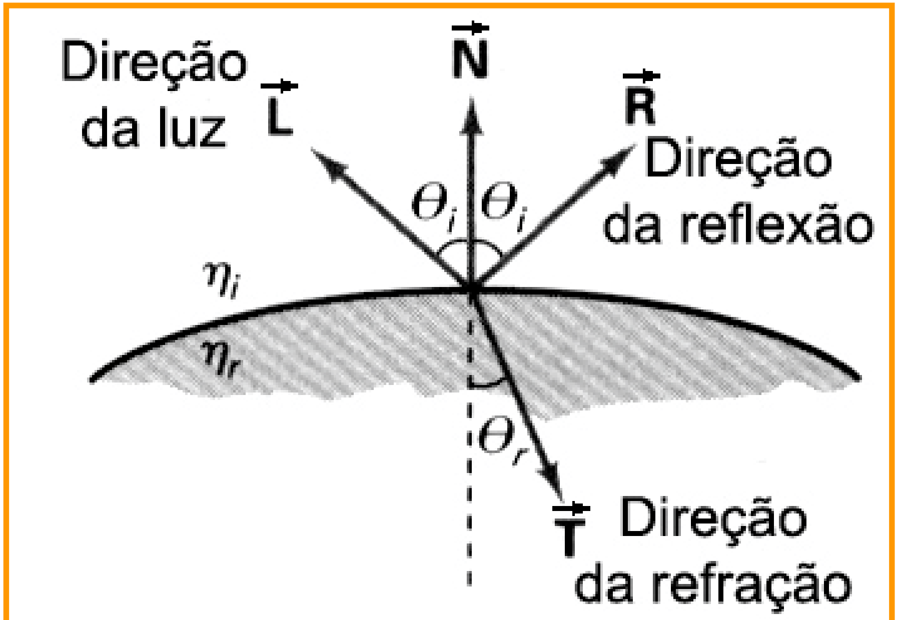
Lei de Snell

$$sen\,\theta_r = \frac{\eta_i}{\eta_r} sen\,\theta_i$$

η = índice de refração
θ = ângulos de incidência/refração
T = vetor de transmissão

Transparência Realista

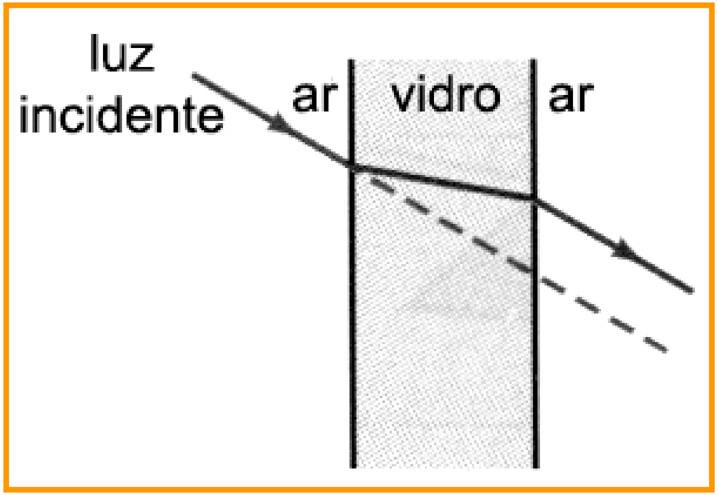
A direção de transmissão (***T***) é calculada por:



$$\hat{T} = \left(\frac{\eta_i}{\eta_r} \cos\theta_i - \cos\theta_r \right) \hat{N} - \frac{\eta_i}{\eta_r} \hat{L}$$

Transparência Realista

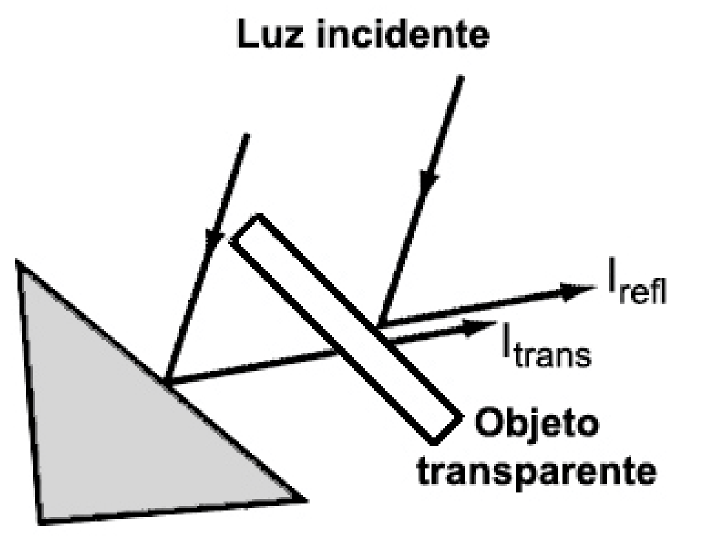
Na prática, a refração desvia o raio transmitido para um caminho paralelo ao raio incidente, permitindo simplificações no processo.



Simplificação: ignorar a mudança na direção do raio transmitido, ou seja, os meios possuem iguais índices de refração. Adequado para superfícies finas.

Transparência Realista

Para uma superfície transparente combina-se a intensidade transmitida (I_{trans}) com a intensidade refletida (I_{refl}).



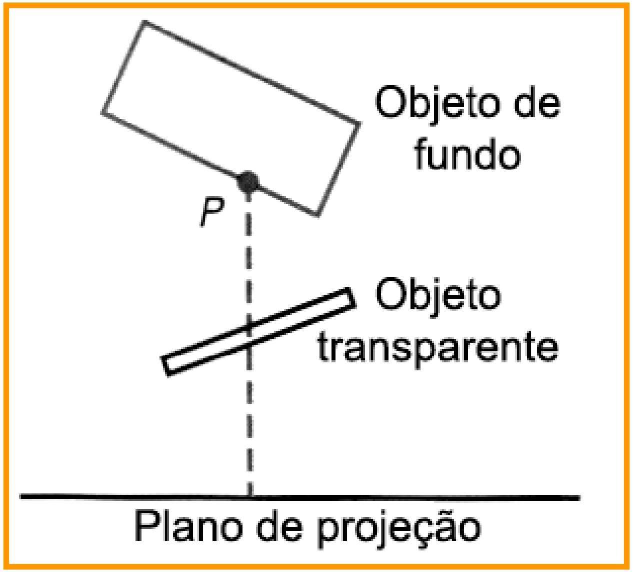
$$I = (1 - K_t) I_{refl} + K_t I_{trans}$$

K_t = coeficiente de transparência
($0 \leq K_t \leq 1$)

Objetos altamente transparentes possuem K_t próximos de 1; objetos opacos têm K_t próximos de 0.

Transparência com *Z-Buffer* Modificado

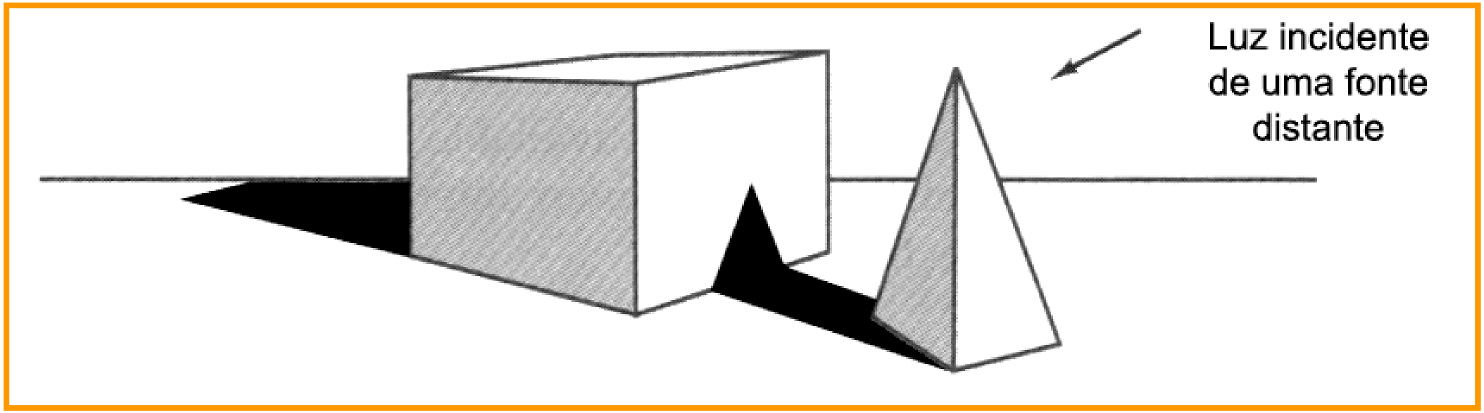
Processa-se inicialmente as superfícies visíveis dos objetos opacos, atribuindo a intensidade da reflexão ao *buffer* da imagem.



Compara-se a profundidade dos objetos transparentes visíveis com o buffer de profundidade. Se essa for menor combina-se intensidade de reflexão com a intensidade da superfície opaca previamente armazenada no *buffer*.

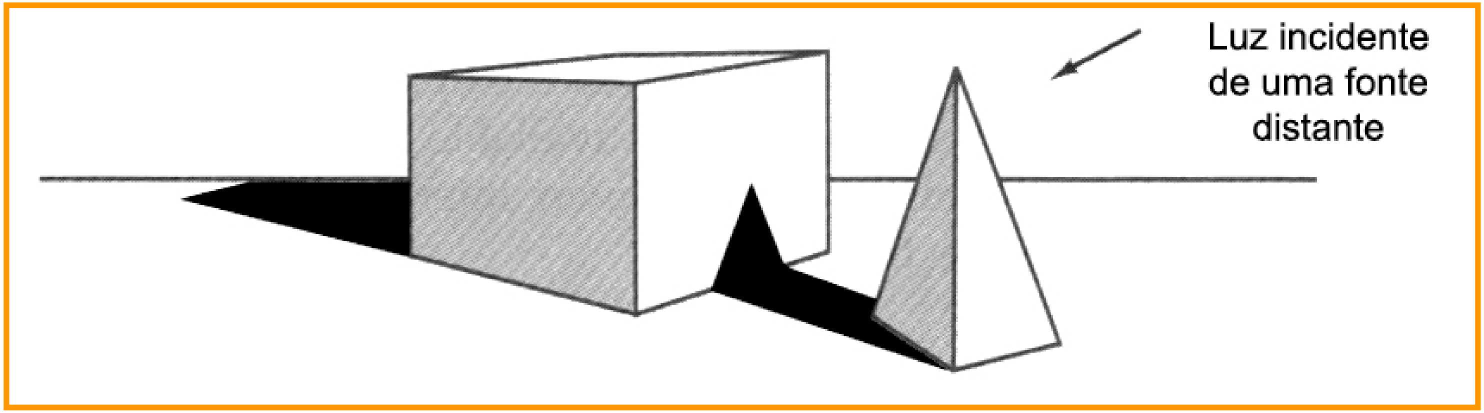
Sombras

Utilizam-se os métodos de ocultação de superfícies para localizar as áreas onde as fontes de luz produzem sombras.



Posicionando-se o observador na fonte de luz determinam-se quais partes das superfícies não são visíveis a partir da posição da luz (áreas de sombra).

Sombras



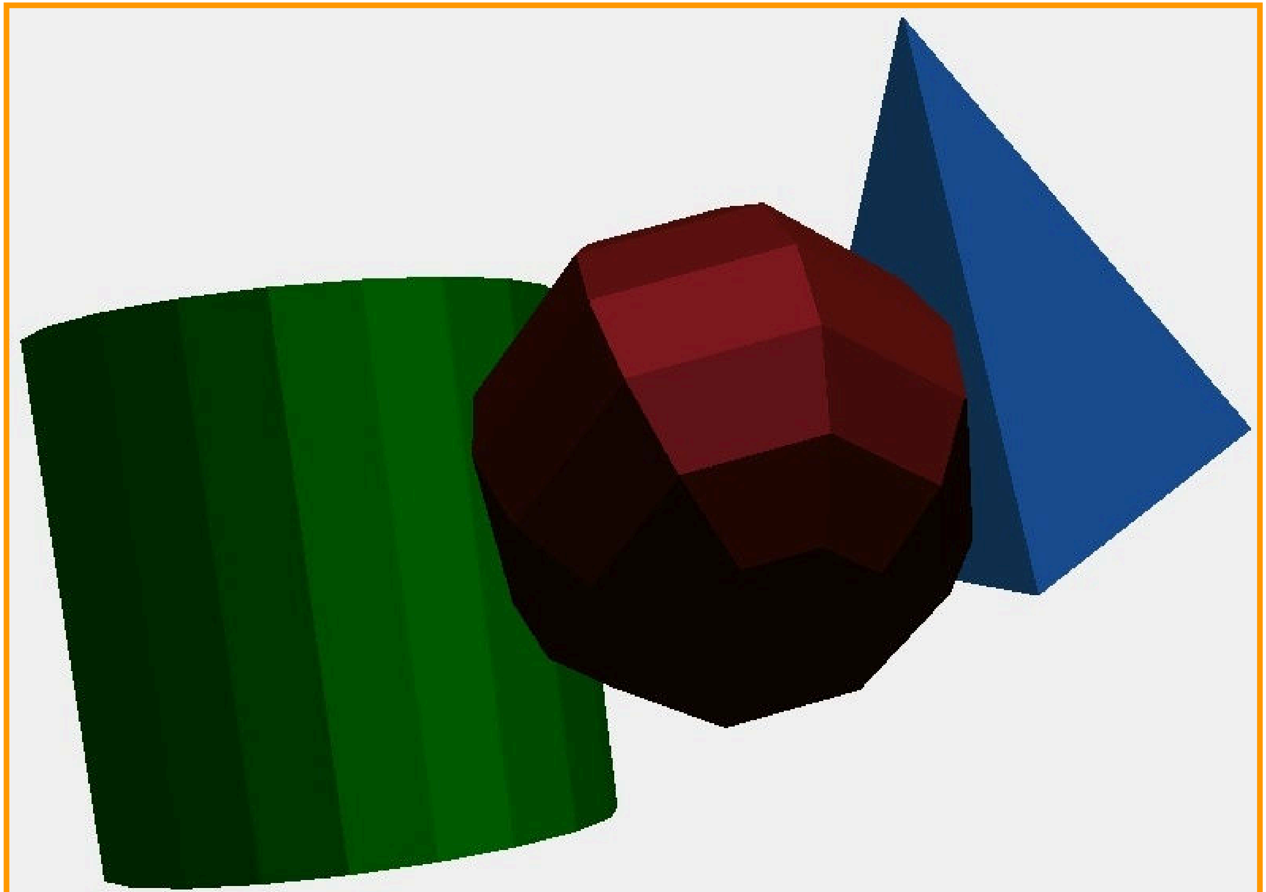
As áreas de sombra são tratadas como superfícies e armazenadas na lista de superfícies.

As áreas de sombra são invariantes à posição do observador, mas não o são em relação às luzes.

Estas superfícies recebem apenas iluminação ambiente, que pode ser combinada com texturas.

Sombreamento (*Shading*)

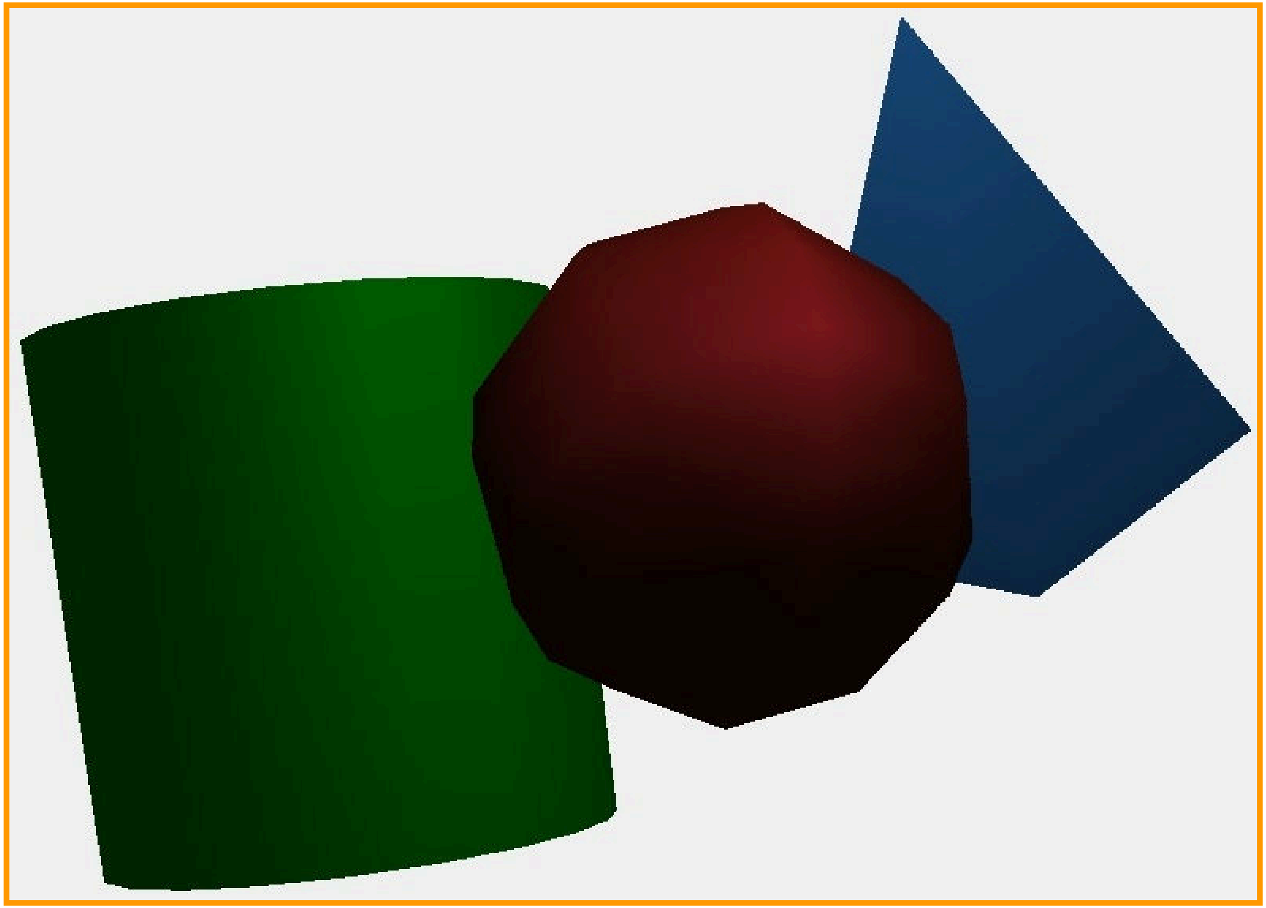
Sombreamento é o processo de colorização dos objetos.



Constante

Sombreamento (*Shading*)

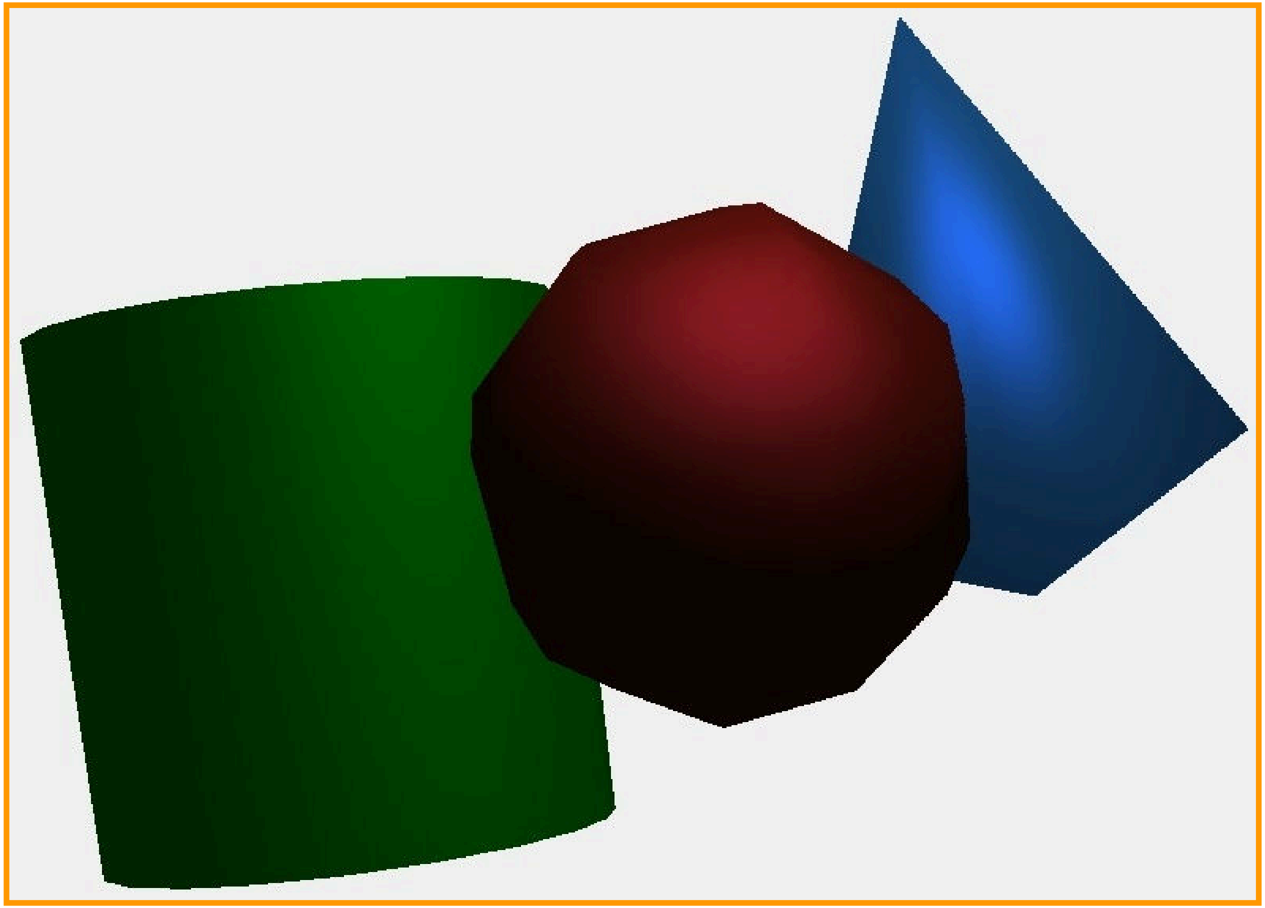
Sombreamento é o processo de colorização dos objetos.



Gouraud

Sombreamento (*Shading*)

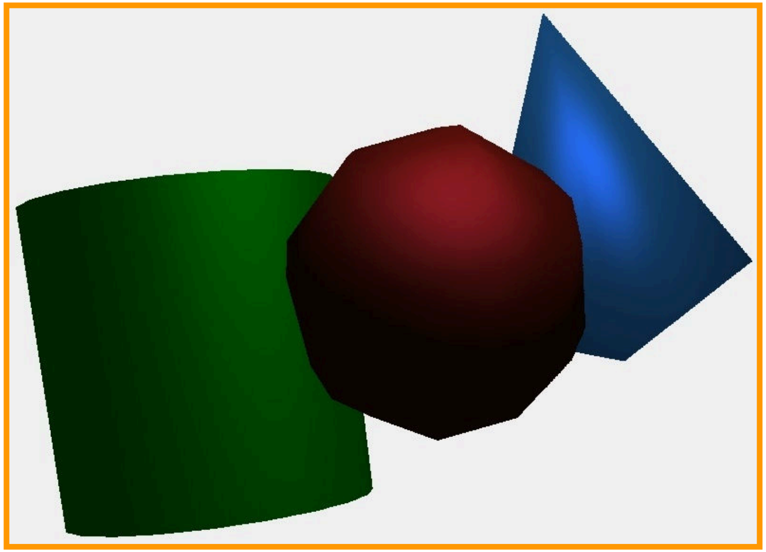
Sombreamento é o processo de colorização dos objetos.



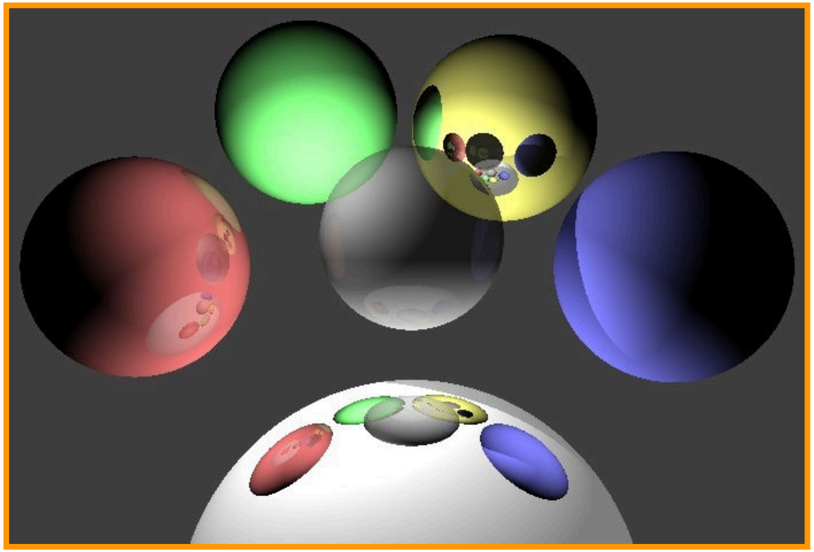
Phong

Sombreamento (*Shading*)

Estes 3 modelos de sombreamento são chamados de algoritmos de sombreamento local, pois as cores são determinadas exclusivamente em função da iluminação e das propriedades de cada objeto.



Sombreamento Local



Sombreamento Global

Sombreamento Constante (*Flat shading*)

É o modelo mais simples de sombreamento.

Aplica-se a função de iluminação apenas uma vez em cada face plana dos objetos, geralmente no centróide da face.

Este único valor de iluminação é utilizado para preencher toda o polígono correspondente à face.

Aproximação só equivale à realidade se:

- Fonte de luz no infinito;
- Observador no infinito;
- A face representa mesmo um objeto plano e não uma aproximação de uma superfície curva.

Sombreamento Gouraud

Desenvolvido por Henri Gouraud (1971)

Consiste no cálculo da iluminação em amostras da superfície de um polígono, seguido de uma interpolação linear.

Adequado para se utilizar em conjunto com algoritmos de determinação de visibilidade que empregam *scan lines*, como o *z-buffer*.

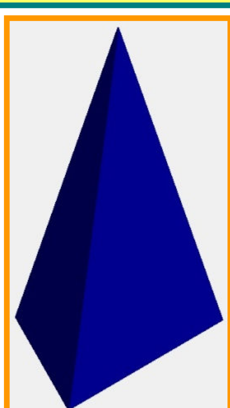
Assim como *z-buffer*, pode ser implementado utilizando aritmética incremental.

Sombreamento Gouraud

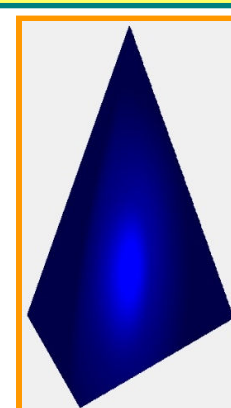
A sombreamento Gouraud apresenta limitações:

Quando se utiliza projeção perspectiva ou aproximações planares de superfícies curvas, os resultados do sombreamento não são realistas.

O sombreamento Gouraud é incapaz de apresentar pontos de reflexão acentuada (efeito especular) situados no interior da superfícies dos polígonos.


Flat

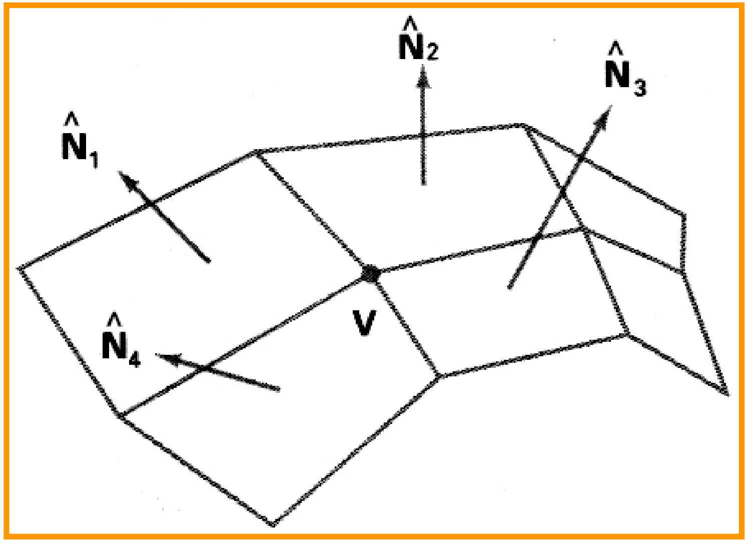

Gouraud


Phong

Sombreamento Gouraud – Algoritmo

1) Determinar o vetor normal unitário médio em cada vértice do objeto;
2) Aplicar a função de iluminação calculando a intensidade total em cada vértice do objeto;
3) Interpolar linearmente as intensidades dos vértices para o restante de cada face do objeto.

Vetor normal unitário médio em cada vértice.

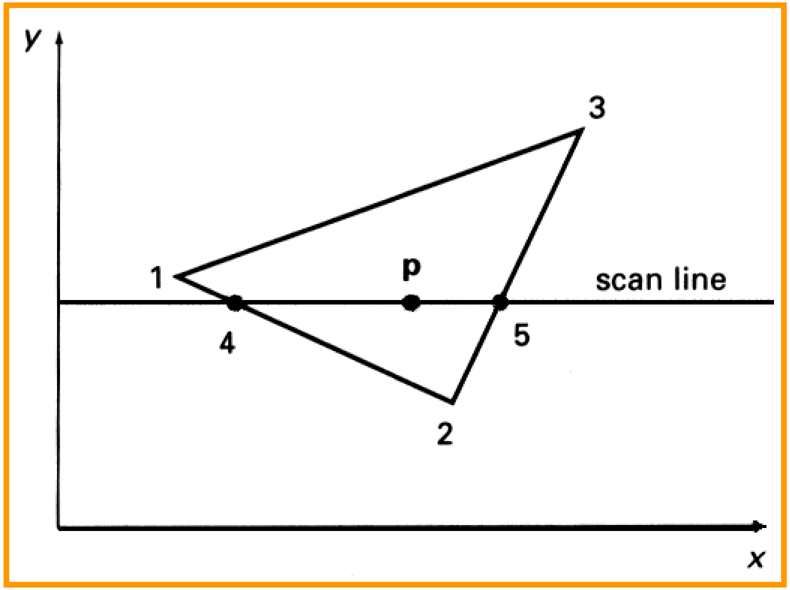


$$\hat{N}_V = \frac{\sum_{k=1}^{n} \hat{N}_k}{\left| \sum_{k=1}^{n} \hat{N}_k \right|}$$

Sombreamento Gouraud – Algoritmo

Obtidos os vetores normais unitários médios calcula-se a intensidade de iluminação total nos vértices do objeto.

Depois interpola-se a intensidade luminosa.



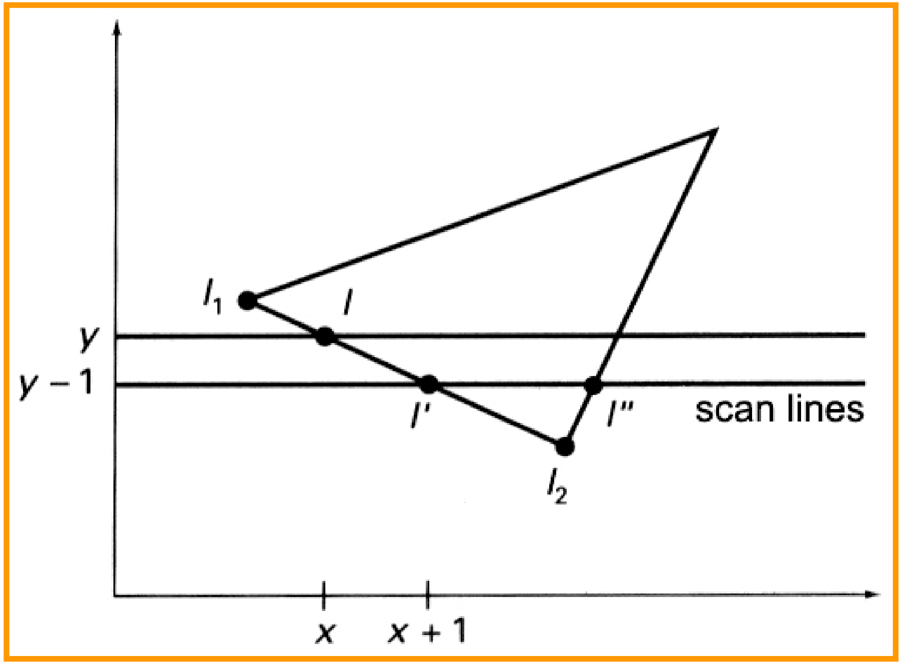
$$I_4 = \frac{y_4 - y_2}{y_1 - y_2} I_1 + \frac{y_1 - y_4}{y_1 - y_2} I_2$$

$$I_5 = \frac{y_5 - y_2}{y_3 - y_2} I_3 + \frac{y_3 - y_5}{y_3 - y_2} I_2$$

$$I_p = \frac{x_5 - x_p}{x_5 - x_4} I_4 + \frac{x_p - x_4}{x_5 - x_4} I_5$$

Interpolação Incremental

Cálculos incrementais são utilizados para interpolar a iluminação entre e dentro das *scan lines*.



Na *scan line* (y):

$$I = \frac{y - y_2}{y_1 - y_2} I_1 + \frac{y_1 - y}{y_1 - y_2} I_2$$

Na *scan line* (y – 1):

$$I' = I + \frac{I_2 - I_1}{y_1 - y_2}$$

Dentro das *scan lines*:

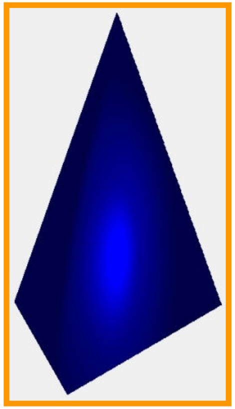
$$I_p = I + \frac{I'' - I'}{x_{I''} - x_{I'}}$$

Sombreamento Phong

Desenvolvido por Bui Tuong Phong (1973).

Phong propôs a interpolação dos vetores normais, com o cálculo posterior da função de iluminação.

O modelo Phong considera adequadamente a variação do ângulo de incidência do feixe de luz, possibilitando a apresentação de pontos com reflexão acentuada no interior das faces do objeto.



Phong

Assim como *z-buffer* e o sombreamento Gouraud, a interpolação dos vetores normais pode utilizar aritmética incremental.

Sombreamento Phong – Algoritmo

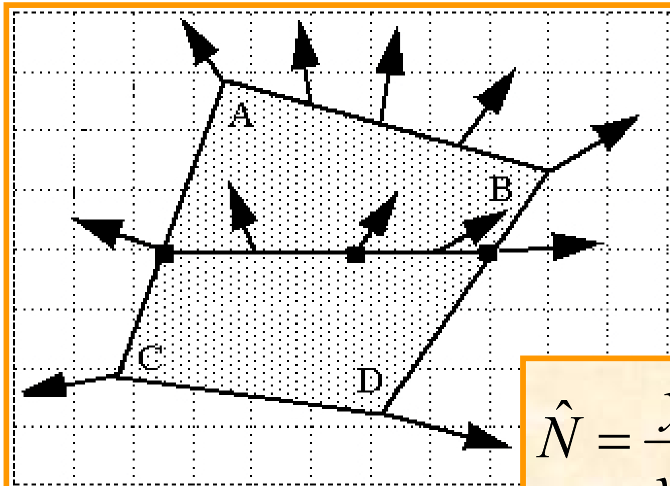
1) Determinar o vetor normal unitário médio em cada vértice do objeto;
2) Interpolar linearmente o vetor normal unitário médio dos vértices ao longo da superfície de cada face;
3) Aplicar a função de iluminação ao longo de cada *scan line* para calcular a intensidade de cada pixel projetado.

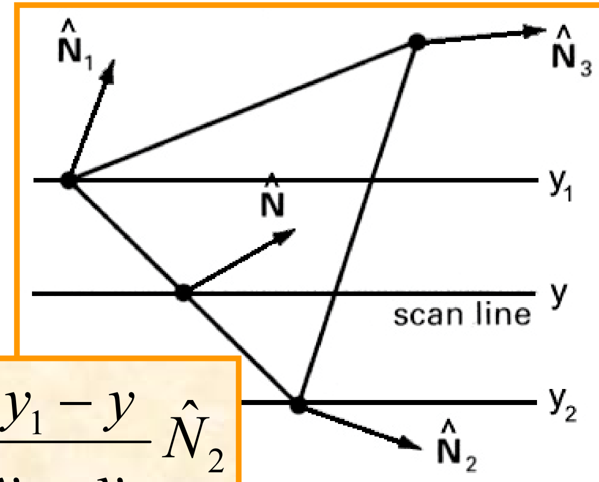
O sombreamento Phong é de 6 a 7 vezes mais lento que o sombreamento Gouraud.

Bishop e Weimer (1986) propuseram o algoritmo Fast Phong, 2 vezes mais lento que Gouraud.

Sombreamento Phong – Algoritmo

Interpolação dos vetores normais ao longo e dentro das arestas:



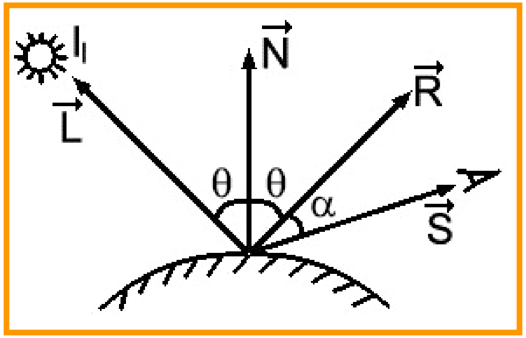


$$\hat{N} = \frac{y - y_2}{y_1 - y_2}\hat{N}_1 + \frac{y_1 - y}{y_1 - y_2}\hat{N}_2$$

Os vetores normais podem ser interpolados usando aritmética incremental. Em cada pixel ao longo da *scan line* é aplicada a função de iluminação.

Sombreamento Phong Simplificado

No cálculo da função de iluminação, no sombreamento Phong, não apenas o vetor normal deve ser interpolado ao longo e dentro das arestas.



Os vetores ***L***, ***R*** e ***S*** também devem ser recalculados, pois cada pixel corresponde uma coordenada na superfície da face.

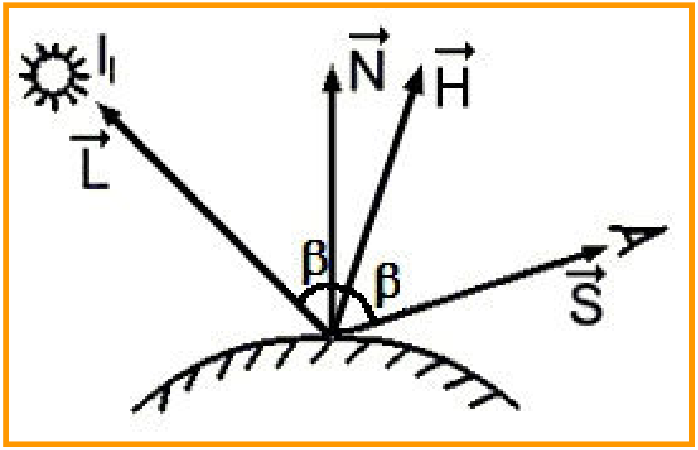
Para evitar tantos cálculos faz-se uma simplificação no sombreamento Phong, sem comprometer significativamente o efeito especular calculado na função de iluminação.

Sombreamento Phong Simplificado

A simplificação consiste em alterar o termo especular da função de iluminação.

$$I_t = I_{la}K_a + f_{att} \cdot I_l\left(K_d\left(\hat{N} \cdot \hat{L}\right) + K_s\left(\hat{N} \cdot \hat{H}\right)^n\right)$$

$$\hat{H} = \frac{\hat{L} + \hat{S}}{\left|\hat{L} + \hat{S}\right|}$$

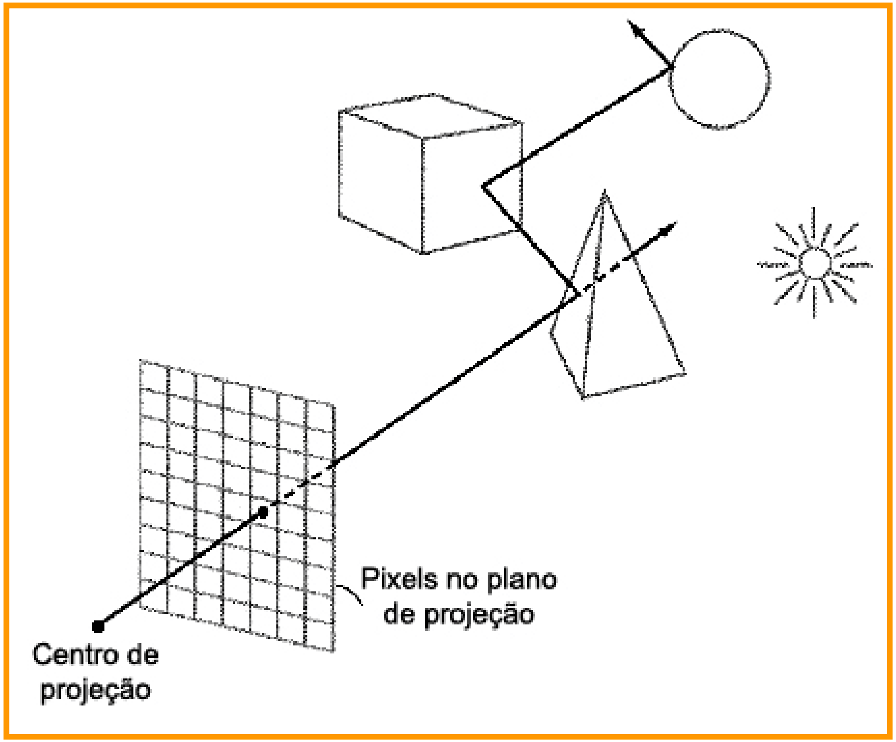


H é o vetor na bissetriz entre os vetores ***L*** e ***S***.

L, ***S*** e ***H*** são calculados uma única vez, em um ponto representativo da face (centróide). Ou seja, assume-se que o observador e a luz estão no infinito.

Ray Tracing

O algoritmo *Ray Tracing* é utilizado na ocultação e iluminação de superfícies.



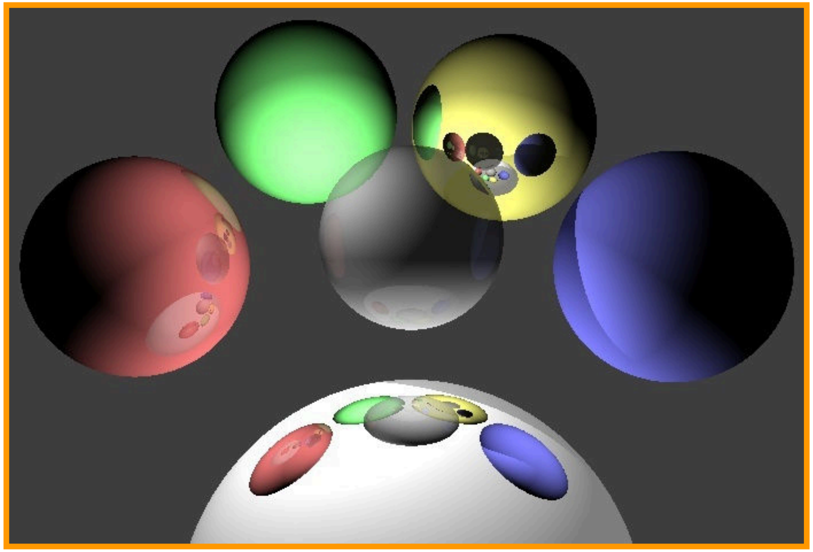
O raio propaga-se para além do objeto interceptado, somando as contribuições dos outros elementos da cena na intensidade luminosa das superfícies.

É um algoritmo de iluminação global.

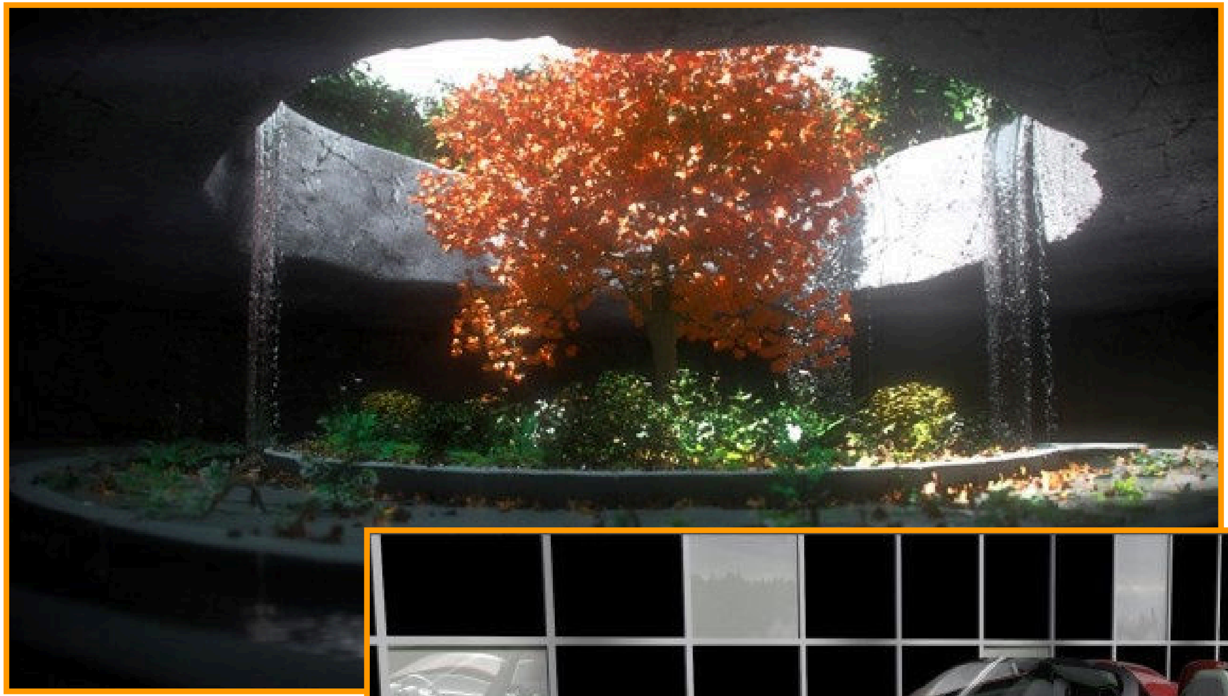
Ray Tracing

Este algoritmo permite identificar superfícies ocultas, efeitos de sombra, transparência e iluminação a partir de fontes múltiplas.

Gera cenas realistas, mas exige muita computação.

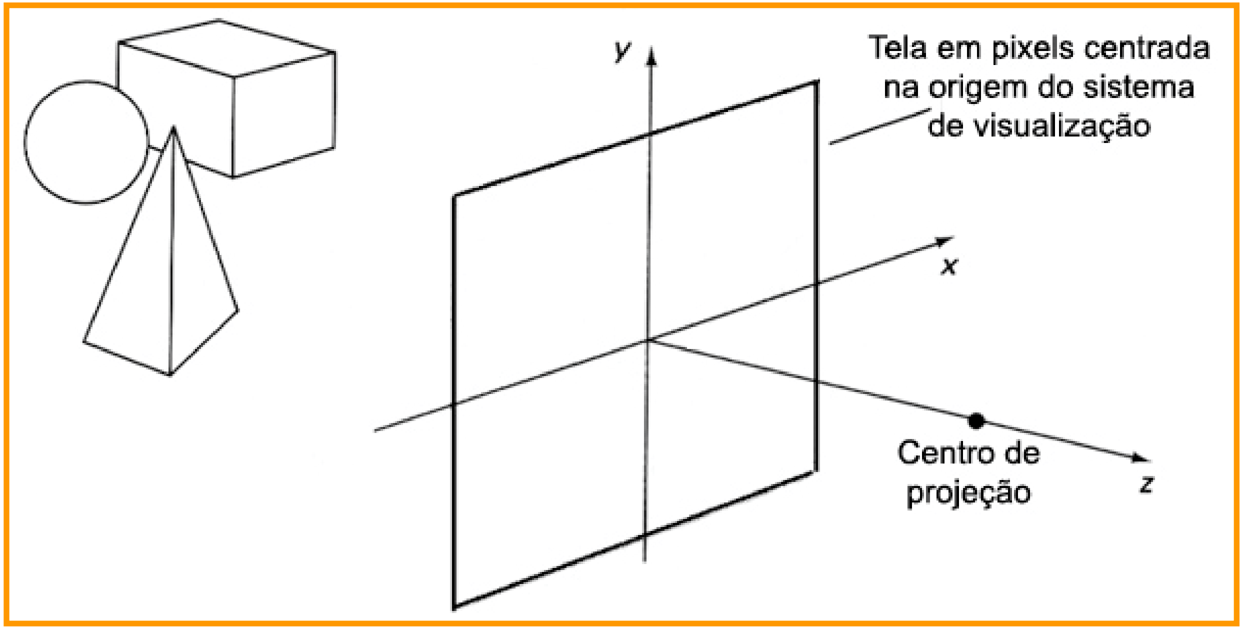


Ray Tracing



Ray Tracing – Considerações Iniciais

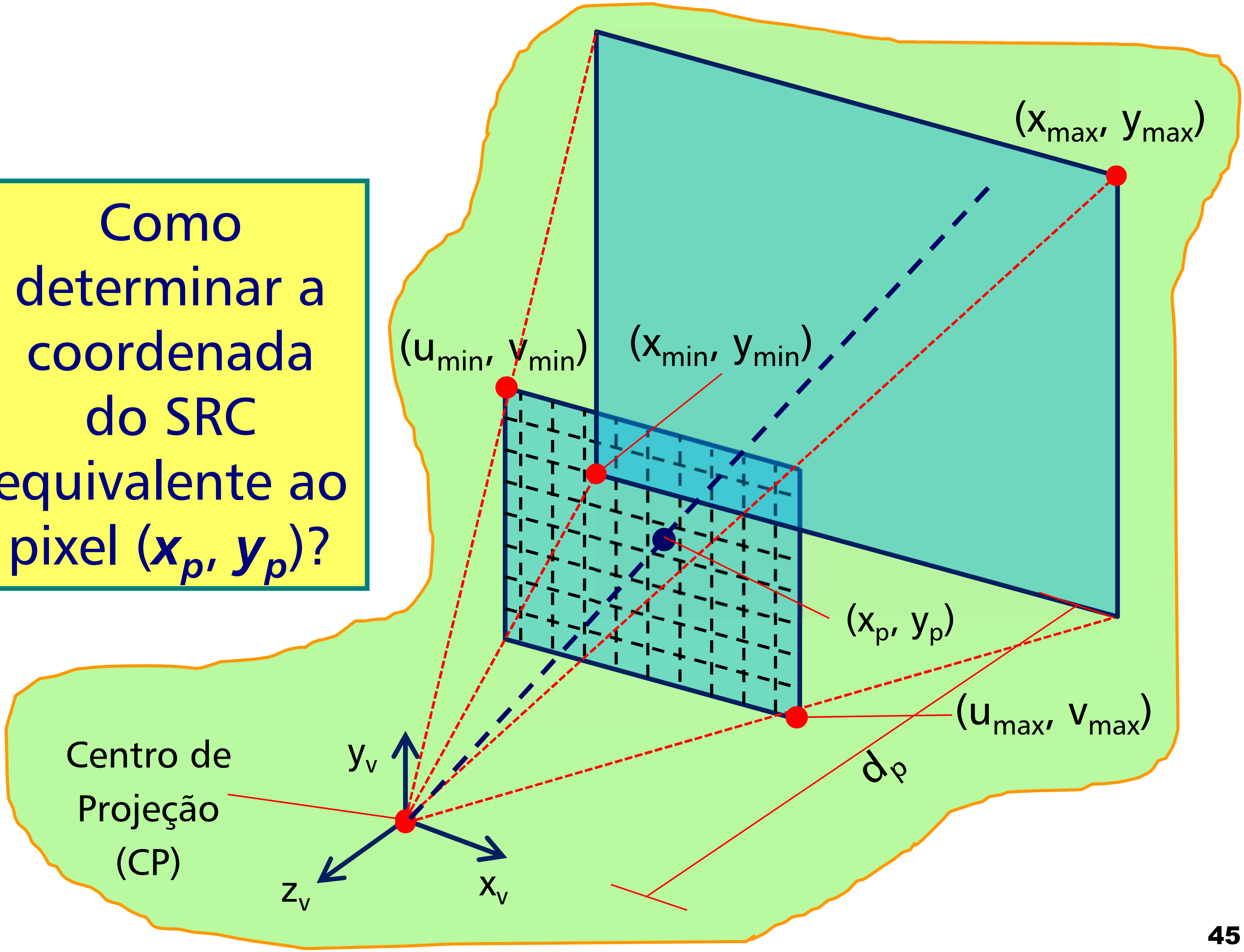
Inicialmente ajusta-se um sistema de coordenadas com os pixels da imagem sobre um plano ***xy***.



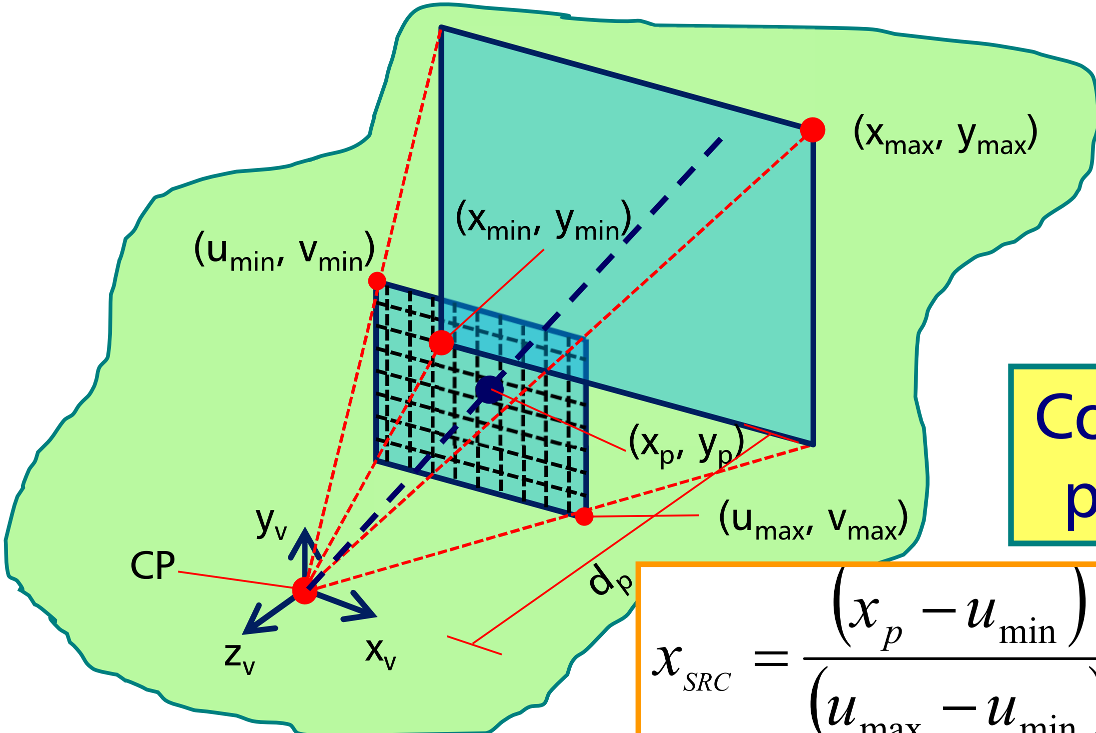
A partir do centro de projeção determina-se o raio que passa através do centro de cada pixel da tela.

Ray Tracing – Definição do Raio

Como determinar a coordenada do SRC equivalente ao pixel ($\boldsymbol{x_p}$, $\boldsymbol{y_p}$)?



Ray Tracing – Definição do Raio



Conversão do pixel para o SRC ($\boldsymbol{P_{Pix}}$):

$$x_{SRC} = \frac{(x_p - u_{min})}{(u_{max} - u_{min})} \cdot (x_{max} - x_{min}) + x_{min}$$

$$y_{SRC} = y_{max} - \frac{(y_p - v_{min})}{(v_{max} - v_{min})} \cdot (y_{max} - y_{min})$$

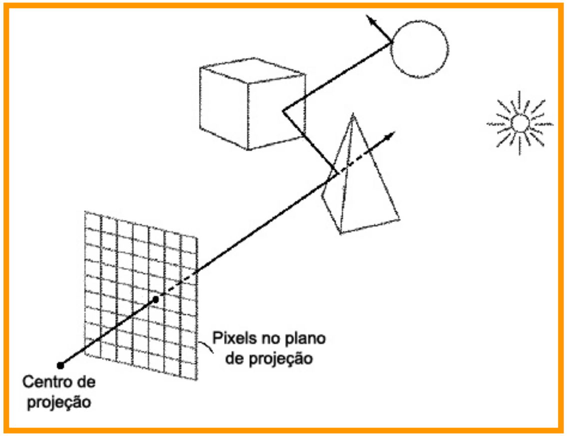
$$z_{SRC} = -d_p$$

Raio

$$\hat{u} = \frac{(P_{Pix})_{SRC} - CP_{SRC}}{\left|(P_{Pix})_{SRC} - CP_{SRC}\right|}$$

Ray Tracing – Algoritmo Básico

Testa-se cada superfície na cena para determinar se ela é interceptada pelo raio que sai do centro de projeção e passa por um pixel na tela.



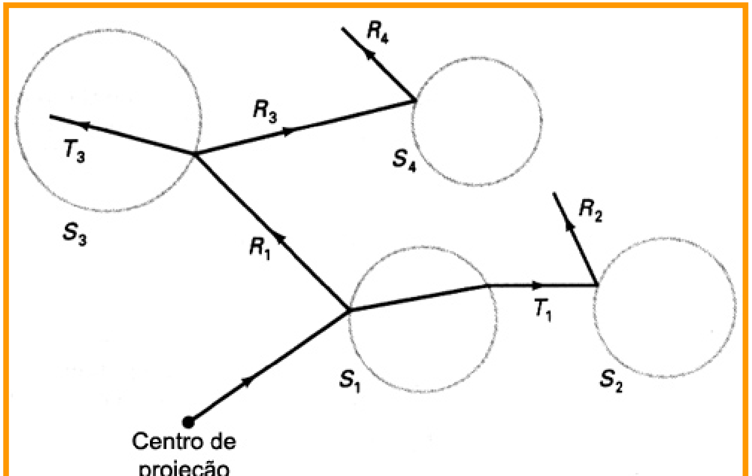
Calcula-se a distância entre o pixel (ou centro de projeção) até os objetos interceptados. A menor distância identifica o objeto visível através do pixel.

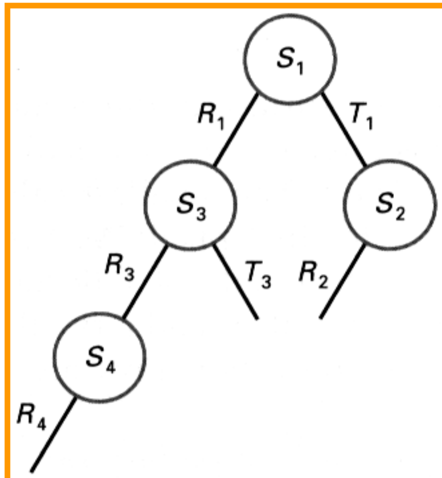
Reflete-se um raio secundário partindo do objeto visível, com ângulo de reflexão igual ao de incidência. Se o objeto for transparente emite-se outro raio secundário, na direção da refração.

Ray Tracing – Algoritmo Básico

Este processo é repetido para cada raio secundário (reflexão e refração).

Testam-se os objetos para verificar interseções e a objeto próximo no caminho do raio secundário é utilizada para definir, recursivamente, os próximos raios de reflexão e refração.





Árvore binária.
Esq. = Reflexão
Dir.= Refração

Ray Tracing – Algoritmo Básico

A profundidade máxima da árvore pode ser definida pelo usuário ou limitada pela quantidade de memória disponível.

O trajeto de um raio tem fim quando:

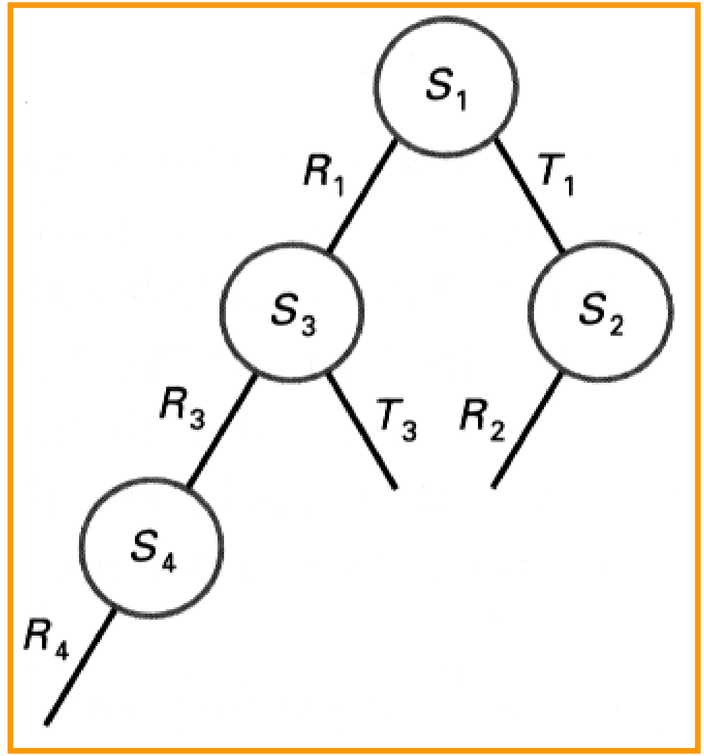
a) atinge a profundidade máxima da árvore;
b) quando o raio atinge uma fonte de luz; ou
c) quando não intercepta nenhum outro objeto.

Se o raio, que sai do pixel, não intercepta nenhum objeto é atribuído ao pixel a cor de fundo da cena. Se este raio atinge uma fonte de luz opaca, é atribuído ao pixel a cor da fonte luminosa.

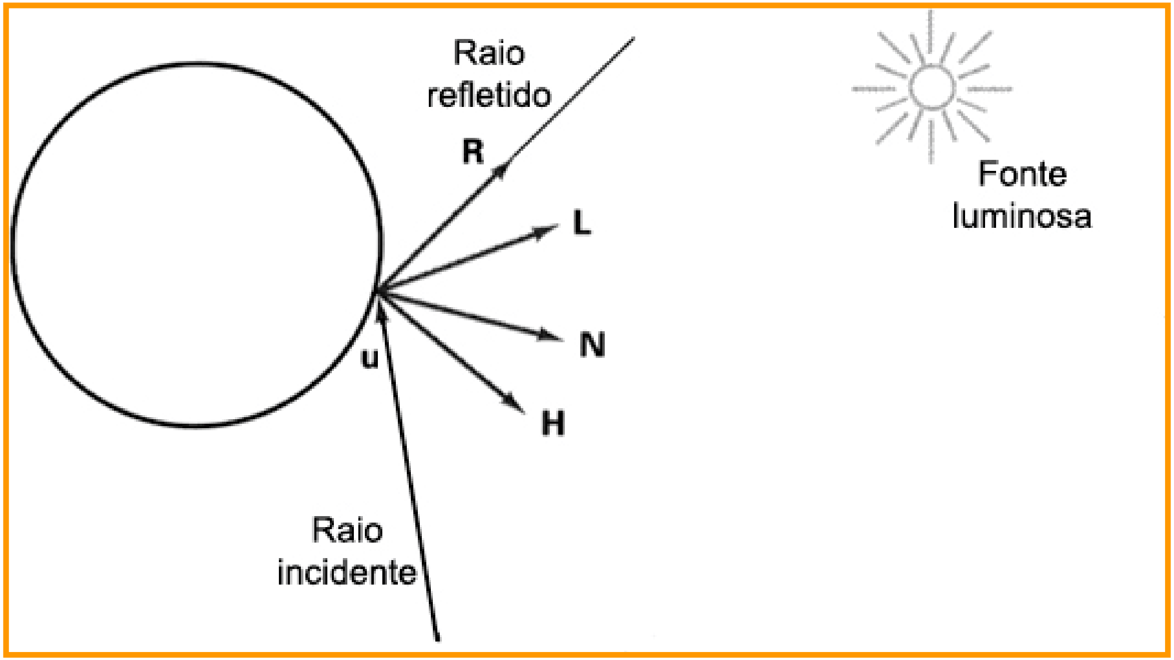
Ray Tracing – Algoritmo Básico

A intensidade atribuída ao pixel é calculada pelo acúmulo das contribuições, iniciando pelas objetos nas folhas da árvore.

A intensidade do objeto, em cada nó da árvore, é atenuada pela distância do objeto "pai" (nó em nível superior) e adicionada à intensidade deste.

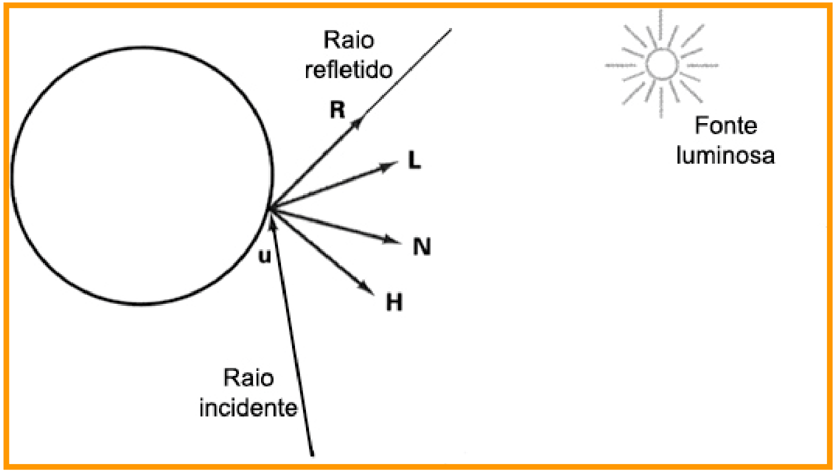


Ray Tracing – Cálculo da Iluminação



u = vetor que indica a direção do raio;
N = vetor normal à superfície;
R = vetor na direção da reflexão;
L = vetor apontado na direção da luz (Há objeto no caminho até a luz? = raio da sombra → apenas I_a);
H = vetor na bissetriz entre –***u*** e ***L***.

Ray Tracing – Cálculo da Iluminação



$$\hat{u} = \frac{(P_{Pix})_{SRC} - CP_{SRC}}{\left|(P_{Pix})_{SRC} - CP_{SRC}\right|}$$

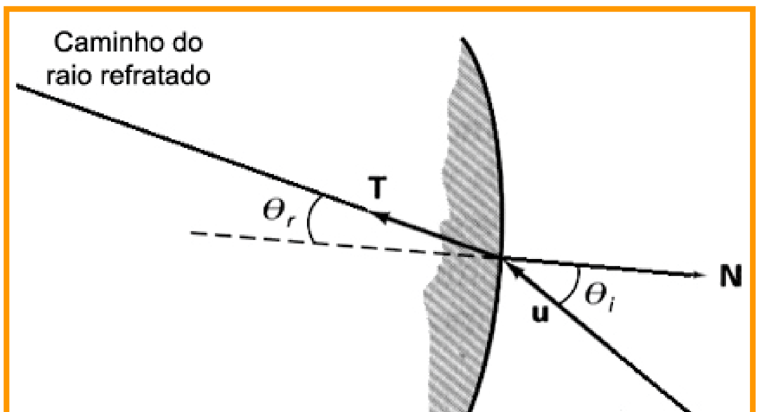
$$\hat{R} = \hat{u} - \left(2\hat{u} \cdot \hat{N}\right)\hat{N}$$

$$\hat{H} = \frac{\hat{L} - \hat{u}}{\left|\hat{L} - \hat{u}\right|}$$

$$I_a = I_{la} \cdot K_a$$

$$I_d \approx K_d\left(\hat{N} \cdot \hat{L}\right)$$

$$I_s \approx K_s\left(\hat{H} \cdot \hat{N}\right)^n$$

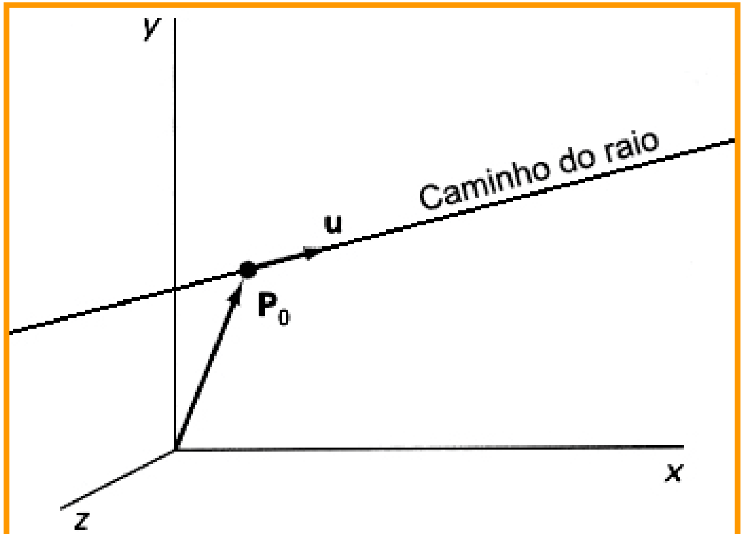


$$\hat{T} = \frac{\eta_i}{\eta_r}\hat{u} - \left(\cos\theta_r - \frac{\eta_i}{\eta_r}\cos\theta_i\right)\hat{N}$$

$$\cos\theta_r = \sqrt{1 - \left(\frac{\eta_i}{\eta_r}\right)^2\left(1 - \cos^2\theta_i\right)}$$

Ray Tracing – Calculando Interseções

Um raio pode ser descrito através de uma posição inicial P_0 e um vetor direção unitário u.



As coordenadas de um ponto P, ao longo do raio, a uma distância s a partir de P_0, é calculada pela equação do raio.

$$P = P_0 + s\hat{u}$$

P_0 pode assumir as coordenadas do pixel (P_{Pix}) no plano de projeção ou do centro de projeção (P_{CP}).

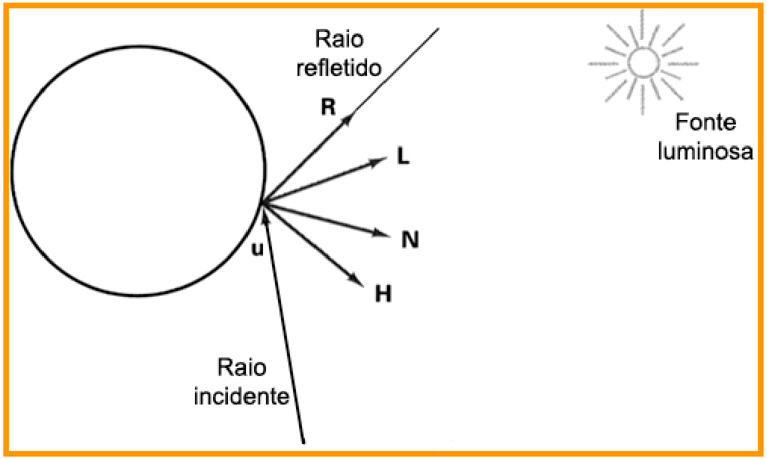
$$\hat{u} = \frac{P_{pix} - P_{cp}}{\left|P_{pix} - P_{cp}\right|}$$

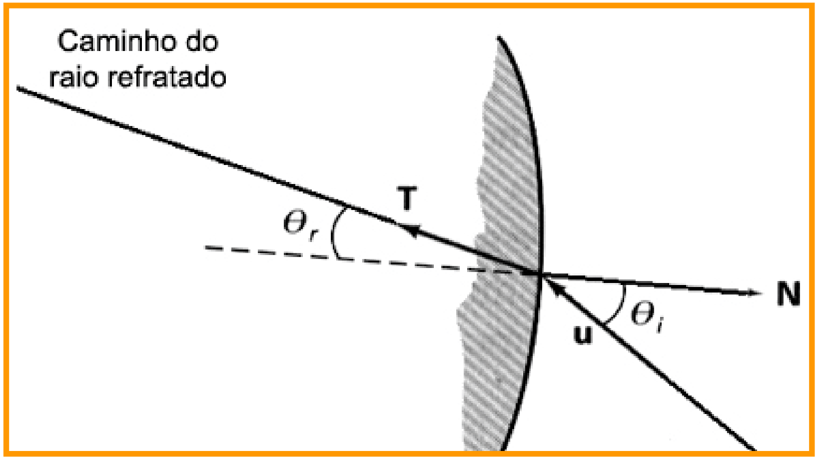
Ray Tracing – Calculando Interseções

Em cada objeto interceptado $\boldsymbol{P_0}$ e o vetor $\boldsymbol{u}$ são atualizados para o ponto da interseção e os raios secundários, respectivamente.

$$P = P_0 + s\hat{u}$$

Na direção de reflexão $\boldsymbol{u} = \boldsymbol{R}$, enquanto na direção da transmissão $\boldsymbol{u} = \boldsymbol{T}$.

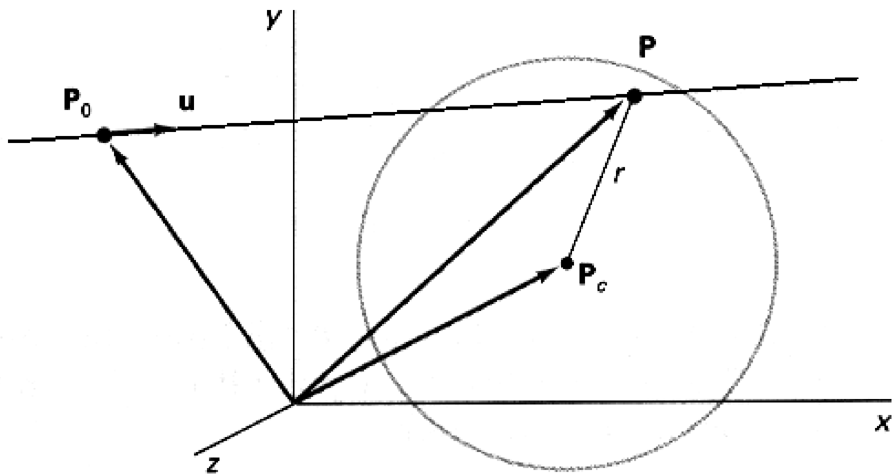




Ray Tracing – Calculando Interseções

A interseção é calculada resolvendo-se a igualdade:
equação do raio = equação do objeto

O objeto mais simples para o *ray tracing* é a esfera.



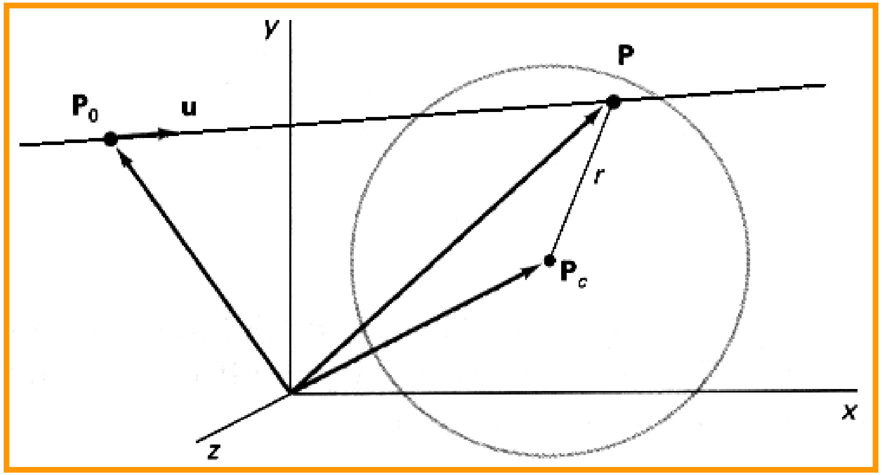
Qualquer ponto ***P*** sobre a superfície da esfera satisfaz a equação:

$$\left|P - P_c\right|^2 - r^2 = 0$$

Substituindo em P a equação do raio:

$$\left|P_0 + s\hat{u} - P_c\right|^2 - r^2 = 0$$

Ray Tracing – Calculando Interseções



Fazendo:

$$\Delta P = P_c - P_0$$

e expandindo a equação anterior, temos:

$$s^2 - 2(\hat{u} \cdot \Delta P)s + \left(|\Delta P|^2 - r^2\right) = 0$$

Cuja solução é:

$$s = \hat{u} \cdot \Delta P \pm \sqrt{(\hat{u} \cdot \Delta P)^2 - |\Delta P|^2 + r^2}$$

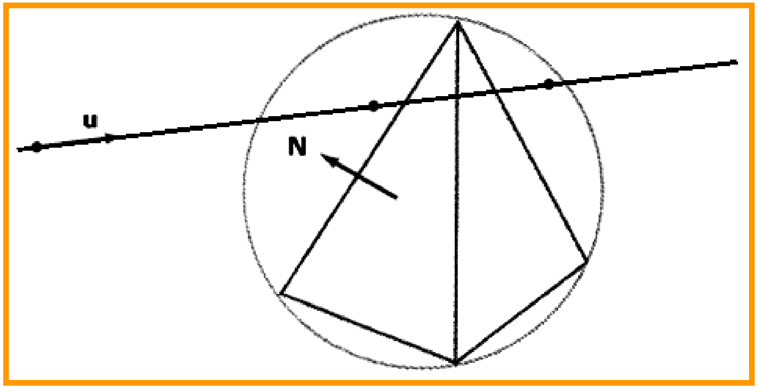
Se o radicando for negativo o raio não intercepta a esfera. A menor solução, aplicada na equação do raio, fornece o ponto de interseção ***P*** entre raio e esfera.

Ray Tracing – Calculando Interseções

Para esferas pequenas e distantes do início do raio ($r^2 << |\Delta P|^2$), reescrevemos o cálculo de ***s*** para evitar erros de arredondamento.

$$s = \hat{u} \cdot \Delta P \pm \sqrt{r^2 - \left|\Delta P - (\hat{u} \cdot \Delta P)\hat{u}\right|^2}$$

O cálculo da interseção entre o raio e um poliedro requer muito mais cálculos. Por isso usamos a esfera como um pré-filtro.



Se o raio não intercepta a esfera então não interceptará nenhuma das faces do poliedro.

Ray Tracing – Calculando Interseções

Se o raio intercepta a esfera envolvente então será necessário testar todas as faces frontais do poliedro.

Uma face frontal satisfaz o teste: $\hat{u} \cdot \hat{N} < 0$

Para cada face frontal identificada, resolvemos a equação do plano: $\vec{N} \cdot P = -D$

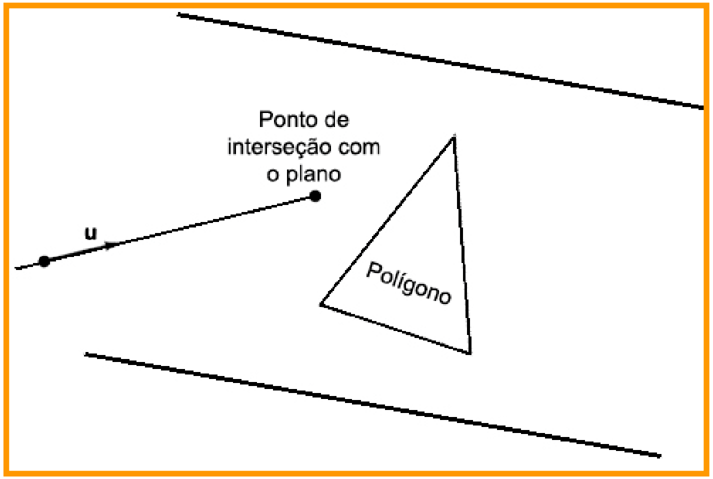
Onde o vetor ***N*** = (***A***, ***B***, ***C***) e ***D*** é o quarto parâmetro da equação do plano. ***P*** é o mesmo no plano e no raio, se: $\vec{N} \cdot \left(P_0 + s\hat{u}\right) = -D$

$$s = -\frac{D + \vec{N} \cdot P_0}{\vec{N} \cdot \hat{u}}$$

Aplicando ***s*** na equação do raio, teremos o ponto de interseção ***P***, com o plano.

Ray Tracing – Calculando Interseções

O ponto ***P*** corresponde a uma posição no plano infinito que contém a face testada, mas ***P*** pode não estar dentro dos limites desta face.



Aplica-se um teste "dentro-fora" para determinar se o raio intercepta a região interna à face testada.

Quando um raio intercepta várias faces ou poliedros, então a interseção mais próxima é aquela que apresenta o menor valor calculado para ***s***.