Rendering

Modelos de Iluminação

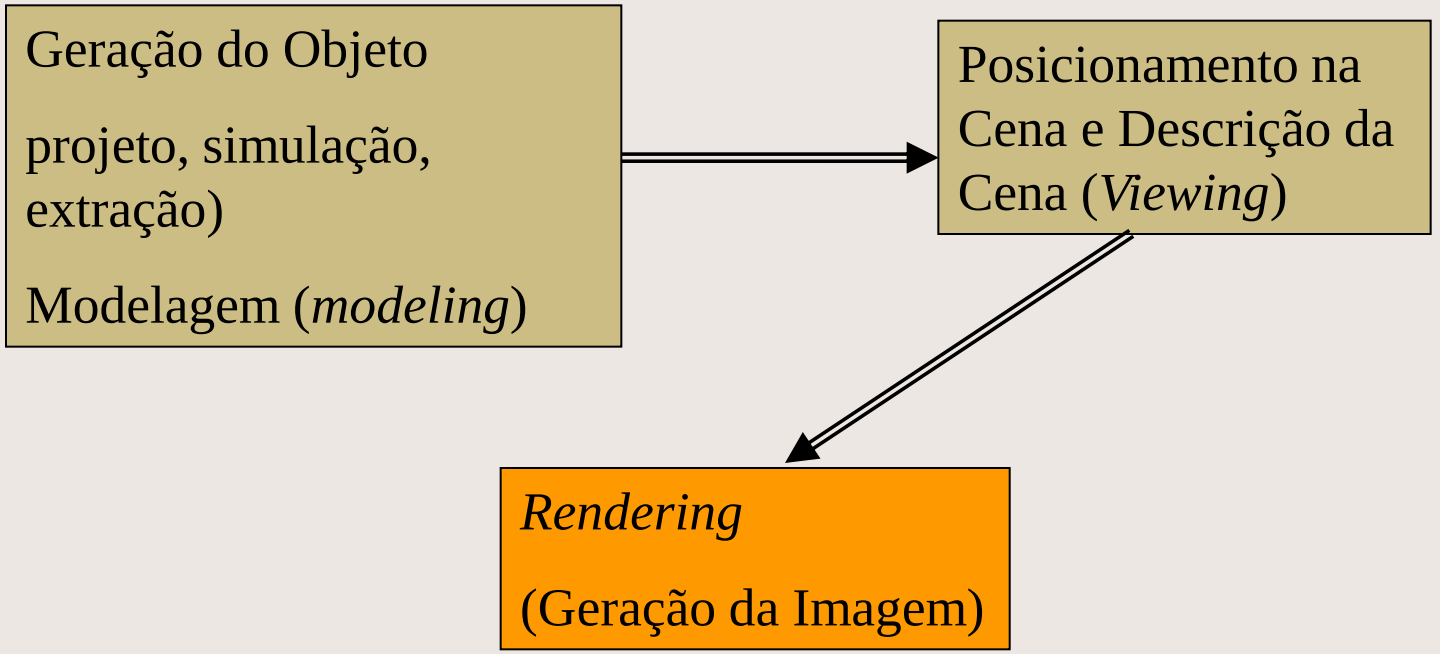
Rendering de Modelos Poligonais

Métodos de Tonalização

2005-2009

Rendering
(onde estamos no pipeline)

- Geração da imagem (matriz de *pixels*) a partir de uma descrição da cena.
- Pipeline:



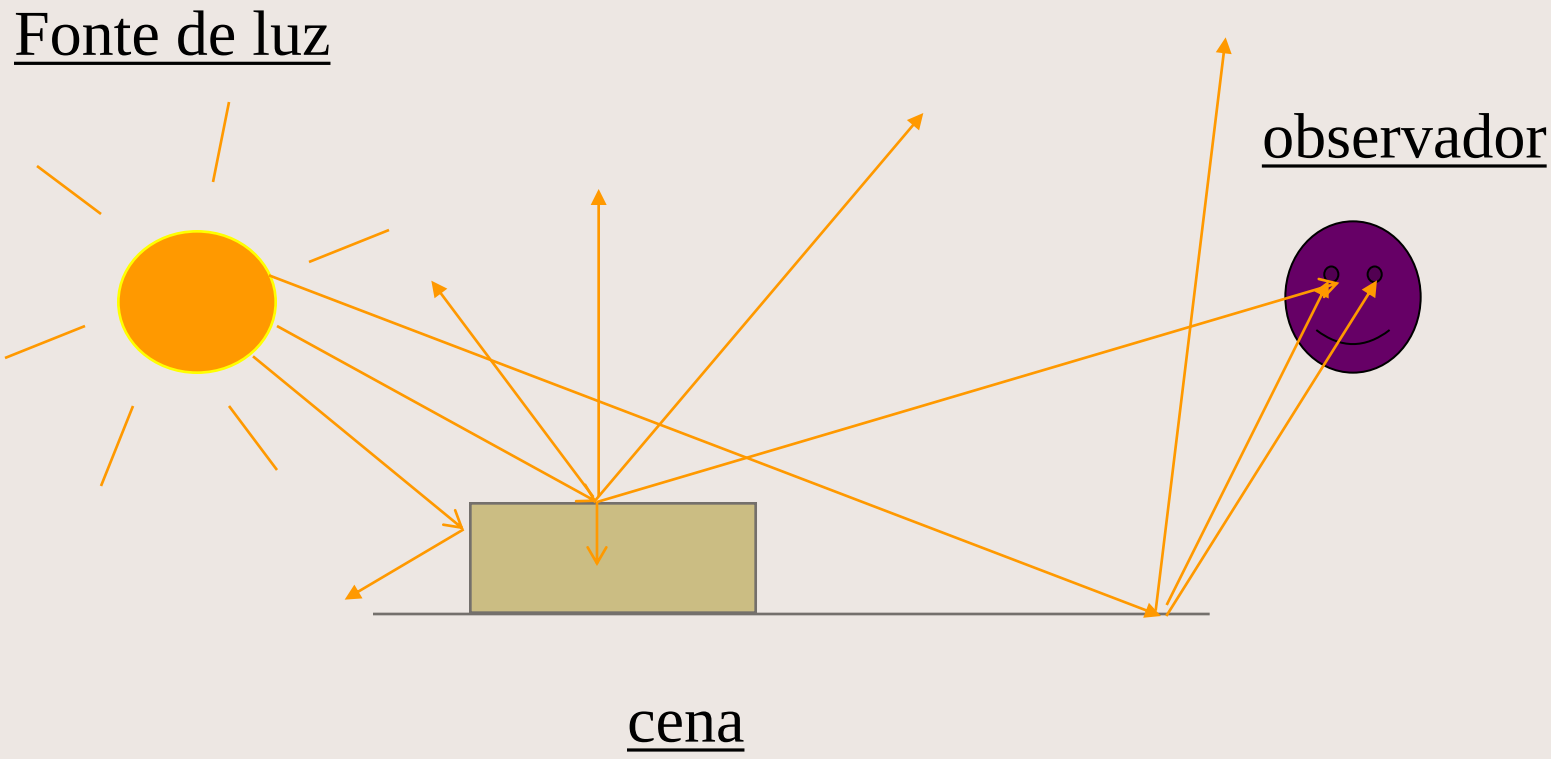
Rendering

- Geração da imagem (matriz de *pixels*) a partir de uma descrição da cena

Dados gráficos ⇒ Imagem

- Cena:
  - Modelo geométrico (geometria dos objetos)
  - Propriedades visuais das superfícies
  - Condições de iluminação ambiente
  - Ponto de observação e outros atributos da visualização

Processo Físico de Geração de uma Imagem



Síntese de Imagens 3D

- Tenta “simular”(muitas vezes, de forma bastante grosseira) o processo físico.
- Modelo de iluminação (*illumination model*, *lighting model*, *shading model*)
  - usado para “calcular” a intensidade (e a cor) da luz que o observador deve “ver” em um certo ponto da superfície do objeto.
  - Modelos básicos x *physically-based models*.

Foto-realismo em CG

- Representações geométricas precisas dos diferentes tipos de objetos
- Boa simulação dos efeitos da iluminação presentes na cena

Surface x *Volume*

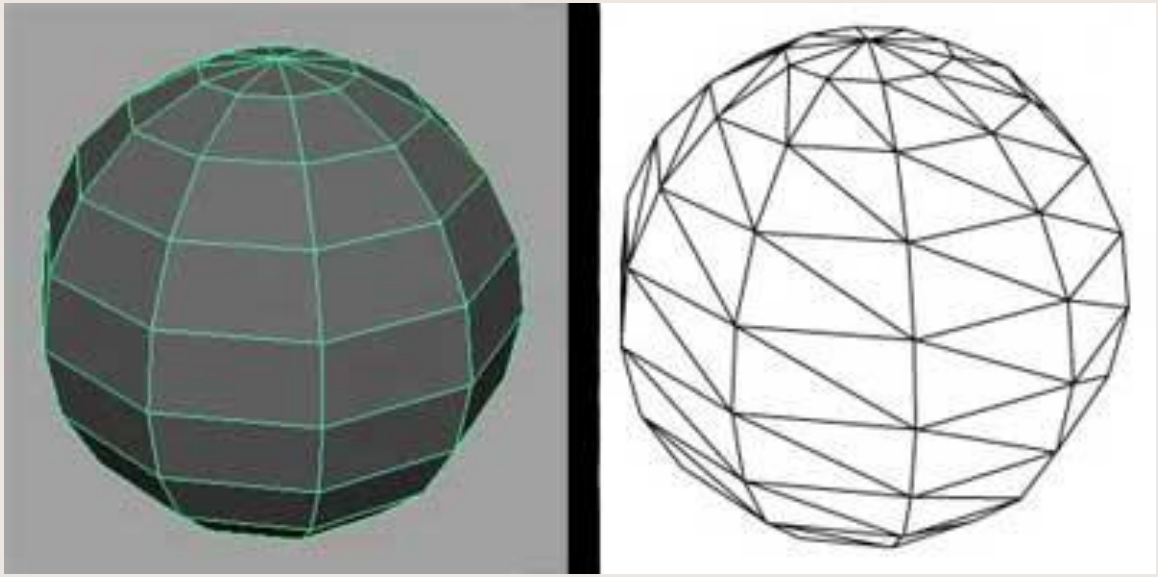
- *Surface Rendering*: cena é renderizada considerando a interação da luz com as superfícies dos objetos da cena
  - OK para a maioria dos objetos manufaturados e para muitos objetos "naturais".
- *Volume Rendering*: o *rendering* considera a interação dos raios de luz com as superfícies e com os 'interiores' dos objetos
  - água, névoa, nuvens, fogo, ...
  - Imagens médicas

http://local.wasp.uwa.edu.au/~pbourke/miscellaneous/volsample/

Wireframe x *shaded*

- Visões ‘fio-de-arame’: desenha as fronteiras das superfícies dos objetos
  - (não precisa de um modelo de iluminação! ⇒ rápidas, mas ambíguas e não “realistícas”.
  - podem exigir um processo de remoção de linhas “ocultas”.
- Visões tonalizadas (“*shaded*”): superfícies preenchidas com cor, aparência (polida, rugosa, áspera, lisa, ...) ⇒ + realismo.

Wireframe



Shading

- O processo de renderizar objetos de modo que sejam percebidos como 3D: depende de como a luz ambiente interage com a superfície
  - Ex. suponha que aproximamos uma esfera por uma malha de muitos polígonos, e colorimos usando glcolor...

Shading

- Elementos no processo:

- Modelo: malha poligonal
- Observador e parâmetros de *viewing*
- Materiais dos objetos
- Fontes de luz

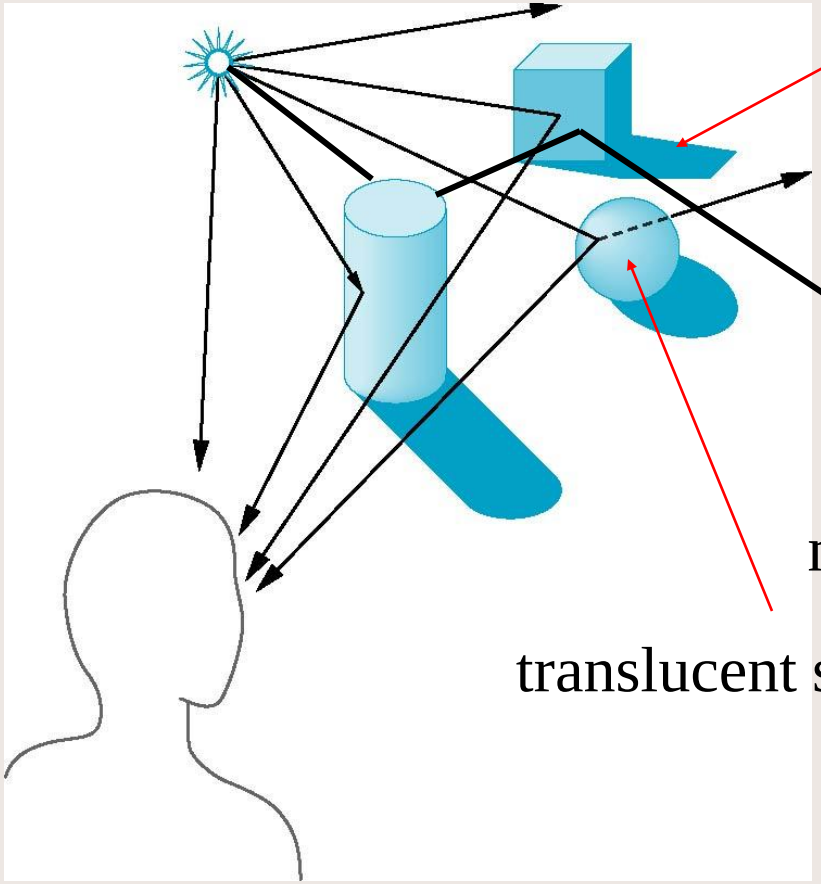
Fontes de Luz

- vemos um objeto opaco não-luminoso devido à luz refletida pela sua superfície.
- o total de luz refletida é resultado das contribuições da luz que atinge o objeto
  - vinda das fontes de luz presentes na cena
  - refletida por outros objetos na cena
- fonte de luz: termo usado para denotar um emissor de energia radiante (lâmpada, sol)

Rendering Local vs Global

- Renderização ´ideal´ requer um cálculo global de todas as interações entre as superfícies dos objetos e as fontes de luz
  - Incompatível com o modelo do pipeline gráfico em que cada polígono é renderizado independentemente (rendering local)
- Se a aparência final é razoável, solução é ok…
  - muitas técnicas para aproximar os efeitos globais (ad hoc)

Mundo real: efeitos globais



Interação luz material

- Luz que atinge um objeto é parcialmente absorvida e parcialmente espalhada (refletida/transmitida)
- As características da luz refletida determinam a cor e o brilho do objeto
  - Superfície que parece vermelha sob luz branca: componente vermelha da luz é refletida, o restante é absorvido
  - Superfície parece mais brilhante se reflete mais luz
- A luz refletida é espalhada de uma maneira que depende da orientação da superfície (em relação à fonte) e da polidez/rugosidade do material

Interação luz material

- Quatro tipos
  - Superfícies especulares (brilhantes): maior parte da luz refletida em uma direção preferencial (direção de reflexão).
    - Espellho é uma superfície especular perfeita.
  - Superfícies difusas (opacas): luz é refletida igualmente em todas as direções (para uma superfície idealmente difusa)
  - Superfícies translúcidas: parte da luz penetra na superfície e é desviada (refração)

Modelos de Iluminação

- tentam reproduzir o efeito das múltiplas interações
  - ´simular´ como a luz é refletida pelos objetos, produzindo o que percebemos como cor
  - luz que sai de um emissor e é refletida pelas múltiplas superfícies dos objetos, eventualmente atingindo o olho do observador
- modelos globais: incluem a contribuição da luz refletida/transmitida por outras superfícies da cena
- modelos locais (1a. ordem): operam como se a iluminação de uma superfície fosse independente das demais

Modelos de Iluminação

- clássico: *Phong* (padrão, simples, rápido, totalmente empírico)
- modelos físicos: para produzir resultados mais realistas usam a teoria que descreve o fenômeno físico da propagação de energia luminosa e sua interação com a superfície dos objetos.
- Ferramental teórico:
  - teoria clássica das ondas eletromagnéticas (para superfícies lisas)
  - modelos de reflexão por superfícies rugosas

Modelo de Iluminação: Exemplo

Processo de *Rendering*

- Um modelo de iluminação é integrado a um método de *rendering*: diferentes métodos podem ser usados para implementar o processo.
- Escolha envolve diversos fatores:
  - como a cena está modelada (modelo geométrico), o grau de foto-realismo desejado, o *hardware* disponível.
  - abordagens clássicas: *scanline*, *ray tracing*, radiosidade.

Métodos de *Rendering*: Classificação

- operam na **ordem da imagem** (gera a imagem *pixel* a *pixel*), ou ...
- na **ordem dos objetos** (renderiza cada objeto/primitiva na cena)
- usam **modelos de iluminação locais** (consideram apenas a contribuição direta da fonte de luz), ou ...
- **modelos globais** (que incorporam a contribuição devida à interação entre os objetos: reflexões múltiplas, transparência, sombras, ...)

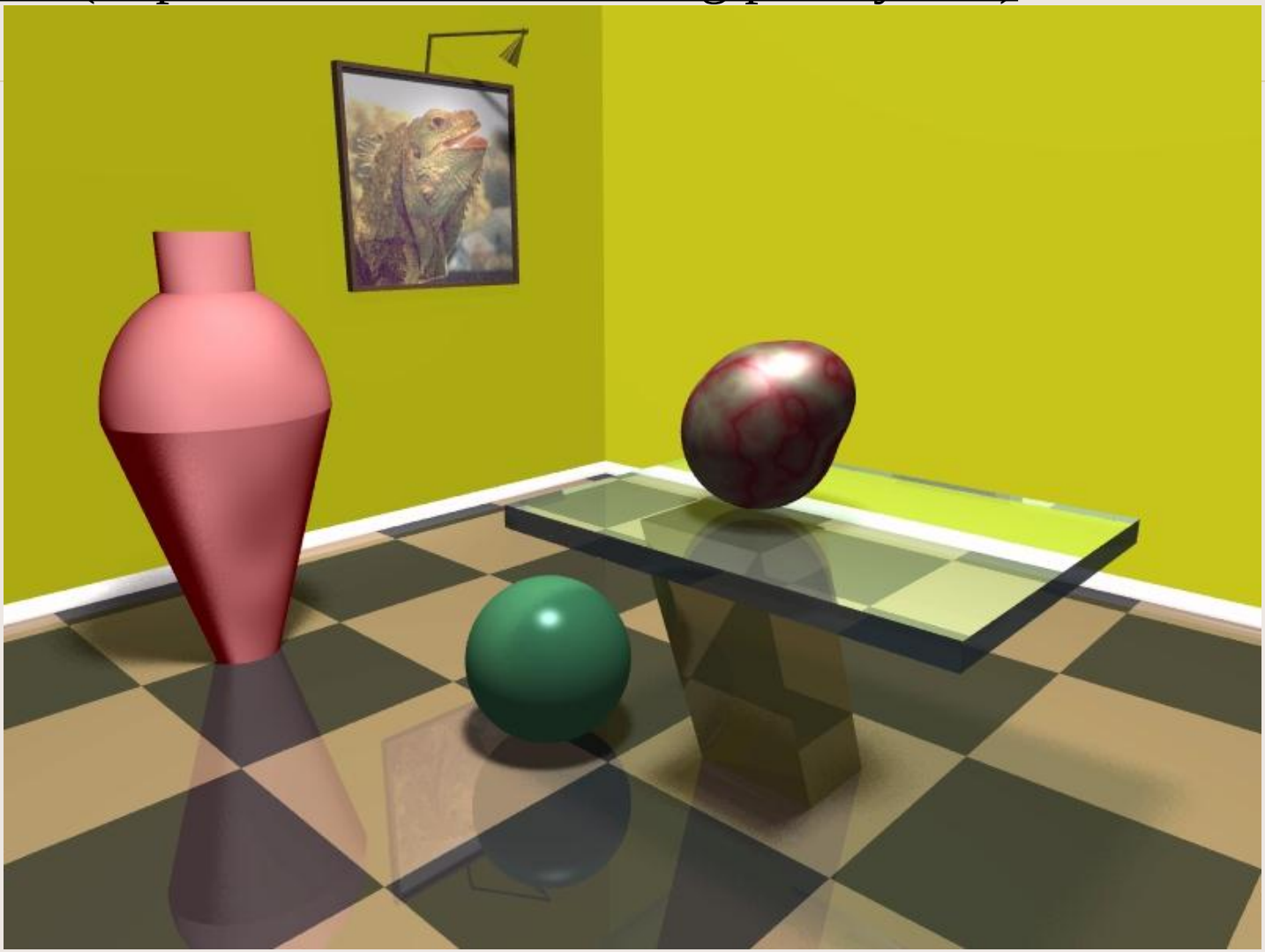
Algoritmos Clássicos

- *scanline*: “padrão” em sistemas gráficos
  - opera sobre objetos poligonais
  - usa modelos de iluminação locais simples, efeitos adicionais podem ser incorporados por várias técnicas *ad hoc*, como cálculo de sombras e mapeamento de textura
  - opera na ordem dos objetos: rasteriza a cena projetada polígono a polígono
  - associado a um processo de remoção de superfícies ocultas (tipicamente, o *z-buffer*)
  - Tipicamente, aplica o modelo de iluminação em alguns pontos do polígono (os vértices) e interpola o resultado

Algoritmos Clássicos

- *Ray tracing*: “clássico” para gerar imagens de cenas com objetos especulares
  - opera sobre diferentes geometrias
  - Opera na ordem da imagem: calcula a iluminação pixel a pixel
  - usa um modelo de iluminação global, integrando efeitos de sombra, reflexões especulares entre objetos, transparência
  - integra naturalmente o processo de remoção de superfícies ocultas
  - alto custo computacional

Figura gerada por Neal Ziring's usando o POV-RAY (http://users.erols.com/ziring/povray.htm)

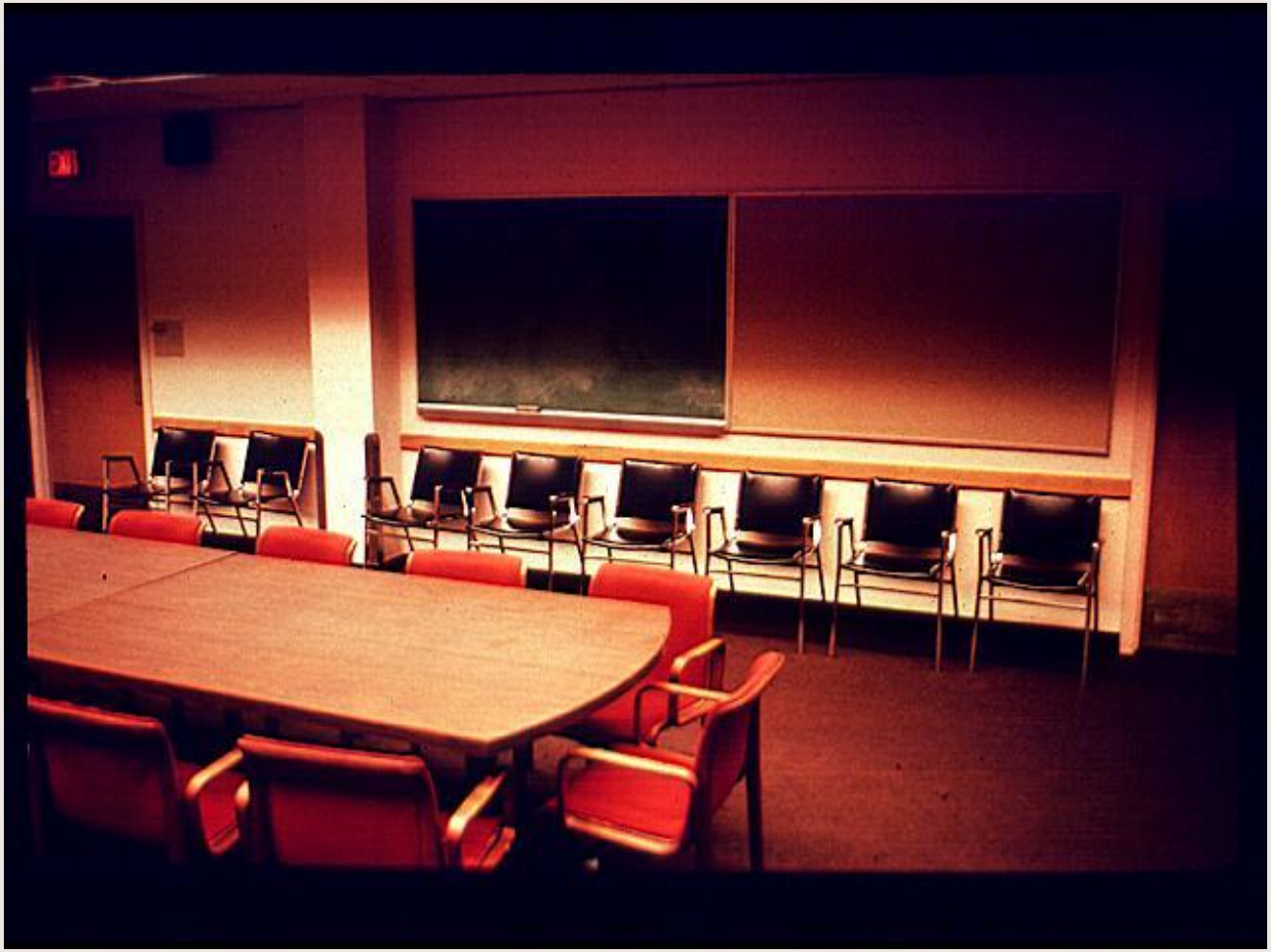


Algoritmos Clássicos

- Radiosidade:
  - modelo global
  - adequado para modelar a reflexão de luz difusa decorrente da interação da luz entre os diferentes objetos em uma cena
  - tenta simular o processo de transferência de energia radiante entre as superfícies dos objetos
  - alto custo computacional
  - foto-realismo

Radiosidade: Exemplo

Radiosidade: Exemplo



Radiosidade: Exemplo

Fontes de Luz

- Um objeto luminoso pode ser um emissor e também um refletor de luz.
- Em geral, consideramos as fontes como emissoras, apenas.
- Fontes de luz são, em geral, especificadas em termos de sua geometria (formato físico da fonte), intensidade da luz emitida, e distribuição espectral.

Fontes de Luz: Geometria

- Pontuais
  - emite luz uniformemente em todas as direções.
  - aproximação para fontes de dimensões pequenas em relação aos objetos na cena (sol, lâmpada incandescente); modelo (idealizado) simples.
- Direcionais: fonte pontual, mas que emite raios em uma única direção (ou em um intervalo). Aproximação para um *spot*.
- Distribuídas: a fonte tem área e uma geometria própria (aproximação para lâmpadas fluorescentes)

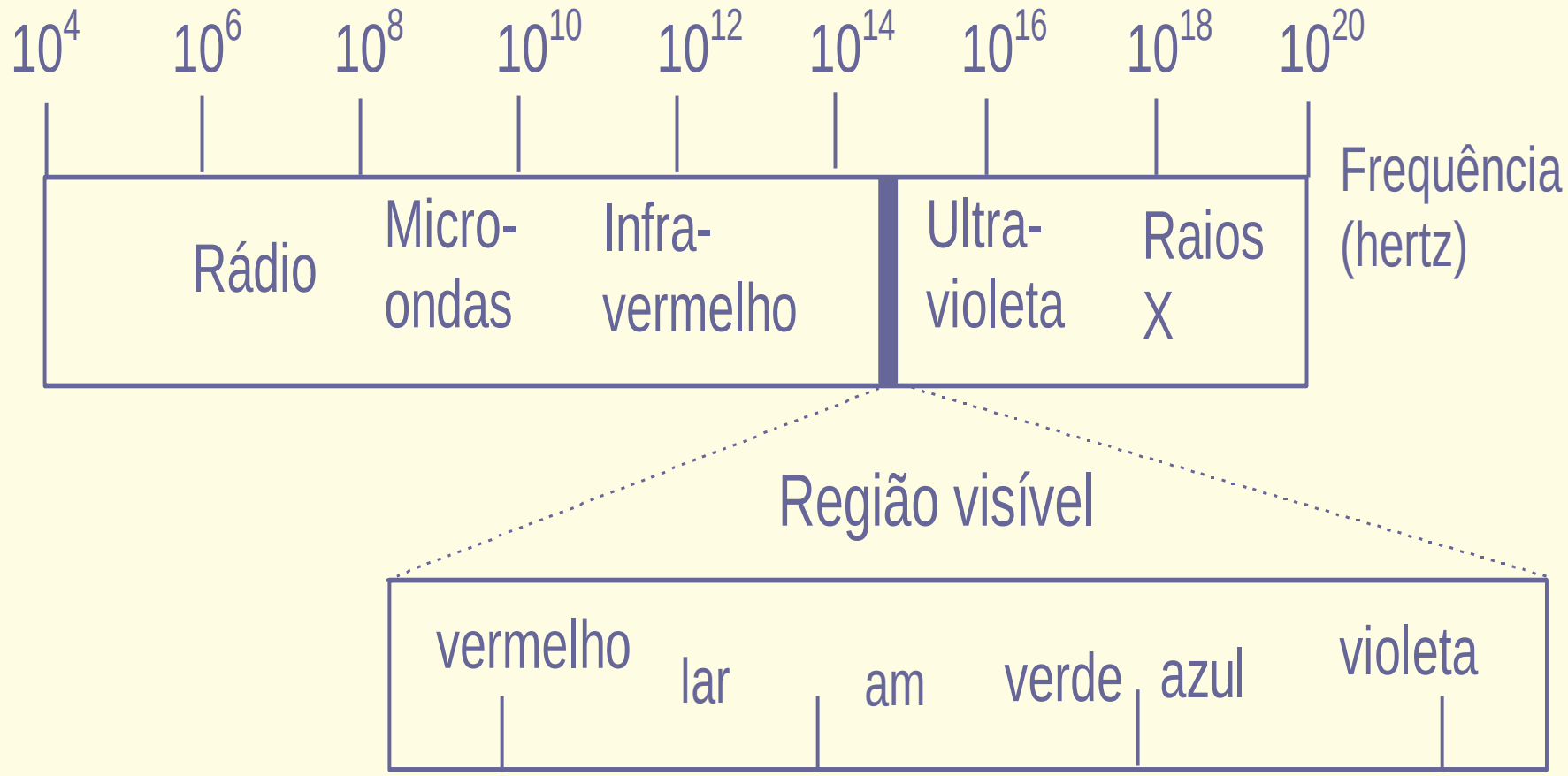
Fontes de Luz: Intensidade e Distribuição Espectral

- intensidade: função que descreve a intensidade luminosa da luz emitida, a cada ponto da superfície emissora (no caso de fontes distribuídas)
- distribuição espectral: energia luminosa emitida descrita em termos da contribuição em cada comprimento de onda do espectro visível (define a “cor” da luz)

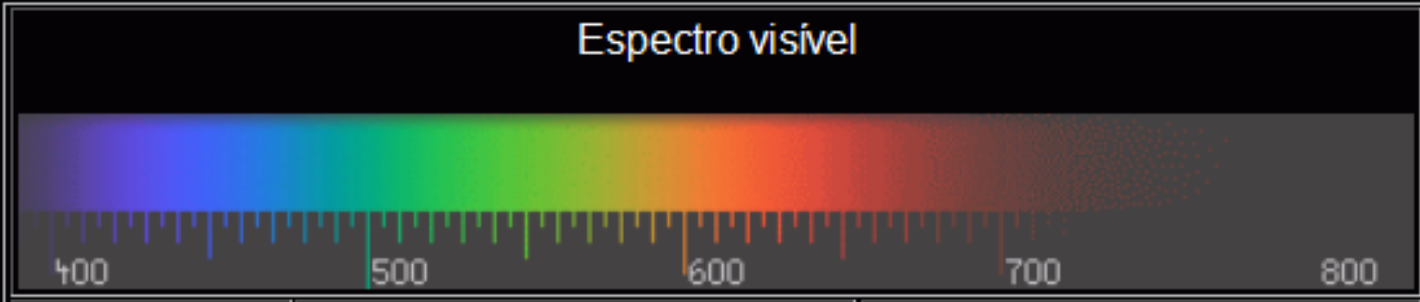
Cor

- Energia luminosa, ou onda eletromagnética:
  - banda visível do espectro eletromagnético: cada freqüência (ou, equivalentemente, cada comprimento de onda) do espectro visível corresponde a uma cor
  - vermelho: 4.3×10^{14} Hz
  - violeta: 7.5×10^{14} Hz
  - comprimentos de onda entre 700nm (vermelho) e 400nm (violeta) correspondem à luz visível

Cor - O Espectro Visível



Cor - O Espectro Visível



Cor	Comprimento de onda (nm)	Freqüência (THz)
violeta	~ 380-440	~ 790-680
azul	~ 440-485	~ 680-620
ciano	~ 485-500	~ 620-600
verde	~ 500-565	~ 600-530
amarelo	~ 565-590	~ 530-510
laranja	~ 590-625	~ 510-480
vermelho	~ 625-740	~ 625-740

Modelo de Iluminação de Phong

- Interação luz incidente/superfície
  - reflexão, absorção (calor), refração.
  - o processo real é extremamente complexo: o modelo de Phong é uma aproximação extremamente simplificada do fenômeno real (modelo empírico).
  - Considera, inicialmente, apenas a reflexão.
- Reflexão
  - quantidade de luz refletida depende do material
  - materiais lustrosos/brilhantes/lisos refletem mais luz, superfícies opacas/rugosas absorvem mais luz; materiais transparentes refratam (transmitem) parte da luz.

Modelo de Iluminação de Phong

- Reflexão difusa: luz incidente refletida igualmente em todas as direções.
  - determina a cor do objeto
  - predominante nas superfícies opacas
- Reflexão especular: a reflexão é mais intensa em uma direção (dada pelo ângulo de reflexão especular)
  - *highlights*: regiões de brilho intenso
  - predominante superfícies muito lisas/lustrosas (“espelhos”)
- A maioria das superfícies/materiais exibe os dois tipos de reflexão

Modelo de Iluminação de Phong

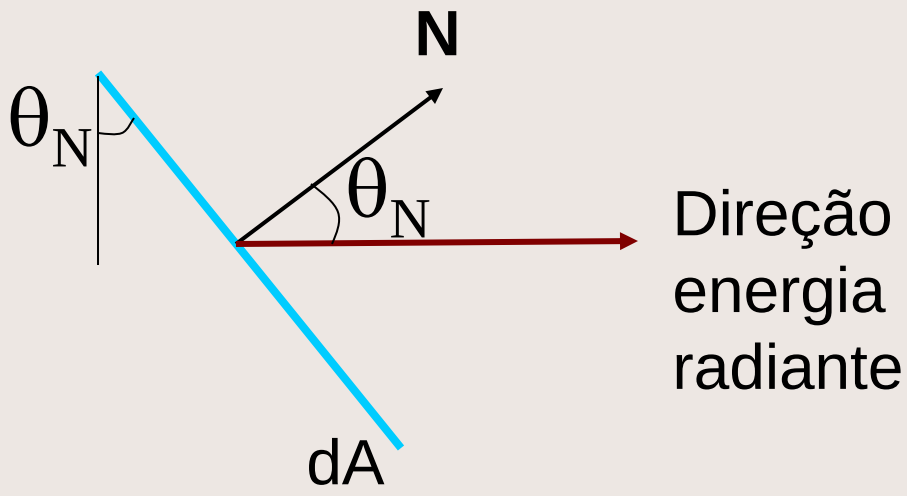
- modelo considera inicialmente o comportamento de uma superfície idealmente difusa
- depois inclui o comportamento de uma superfície idealmente especular
- e inclui ainda um componente de iluminação ambiente
  - para “aproximar” a contribuição dos objetos não emissores para a iluminação da cena, usa um termo de iluminação constante, que atinge da mesma forma (ou quase) todos os objetos

Modelo de Phong: Reflexão Difusa

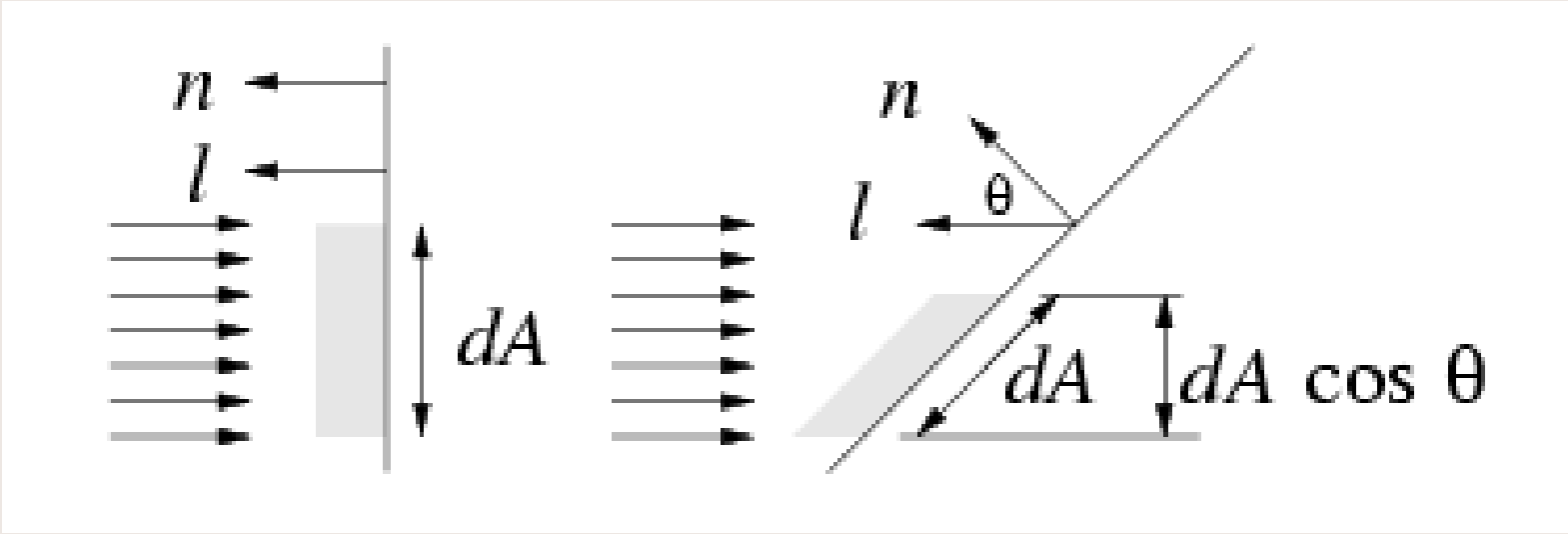
- A superfície reflete a luz incidente igualmente em todas as direções
  - Uma faixa de comprimentos de onda da luz incidente é absorvida, outra faixa é refletida: responsável pela ‘cor percebida’ do objeto.
  - reflexão independente da direção de observação
  - quantidade de luz refletida é controlada por um parâmetro $K_d \in [0,1]$ (coeficiente de reflexão difusa)
  - assume superfície refletora idealmente difusa: reflexão em qualquer ponto da superfície é governada pela Lei dos Cossenos de Lambert

Modelo de Phong: Reflexão Difusa

- Lei dos Cossenos de Lambert:
  - a energia radiante refletida por uma pequena área de superfície dA, em qualquer direção θ (relativa à normal à superfície) é proporcional a cosθ.



Modelo de Phong: Reflexão Difusa



Modelo de Phong: Reflexão Difusa

- intensidade da luz refletida depende da energia radiante por área projetada perpendicular à direção θ_N, dada por $dA.\cos\theta_N$.
- Em uma superfície refletora idealmente difusa o espalhamento da luz é igual em todas as direções, mas a intensidade do brilho percebido depende da orientação da superfície em relação à fonte de luz.
  - Uma superfície orientada perpendicularmente em relação à luz incidente parece mais iluminada do que outra orientada obliquamente (porque a primeira recebe mais luz).

Modelo de Phong: Reflexão Difusa

- Se θ é o ângulo entre a direção da luz incidente e a normal à superfície, então a área projetada do pedaço de superfície dA na direção perpendicular à da luz incidente é proporcional a $\cos\theta$
  - Se $\theta = 0$ a superfície é totalmente iluminada, e a iluminação percebida diminui à medida em que θ aumenta.
  - Modelo assume fonte de luz pontual
  - Cálculo é feito em coordenadas do mundo ou coordenadas de visualização, antes das transformações de *shearing* e perspectiva (que alteram as normais!)

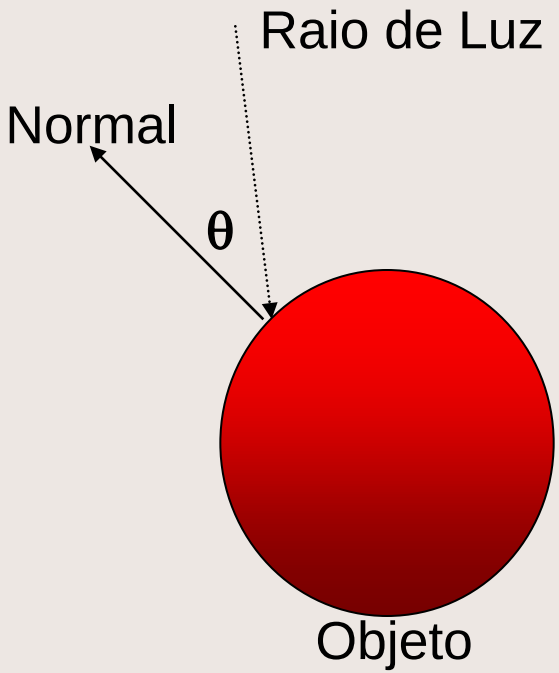
Modelo de Phong: Reflexão Difusa

Modelo Local

Iluminação Difusa (*Lei dos Cossenos de Lambert*)

$$I_D = K_D * I_L * \cos\theta$$

$$0 \leq \theta \leq 90^o$$



Modelo de Phong: Reflexão Difusa

- θ: ângulo entre vetor direção da luz incidente e vetor normal à superfície.
- A área projetada de uma região da superfície, perpendicular à direção da luz, é proporcional a $\cos\theta \Rightarrow$ quantidade (intensidade) de iluminação recebida depende de $\cos\theta$.
- Equação da reflexão difusa devida à luz vinda de uma fonte pontual: $I_{ld} = K_d I_l \cos\theta$.
- Superfície é iluminada pela fonte se $\theta \in [0, 90^o]$. Para **N**, **L** vetores unitários:

$$I_{ld} = K_d I_l (\mathbf{N} \cdot \mathbf{L})$$

ver http://alpha.mini.pw.edu.pl/~kotowski/Grafika/IlluminationModel/Index.html

Modelo de Phong: Reflexão Difusa

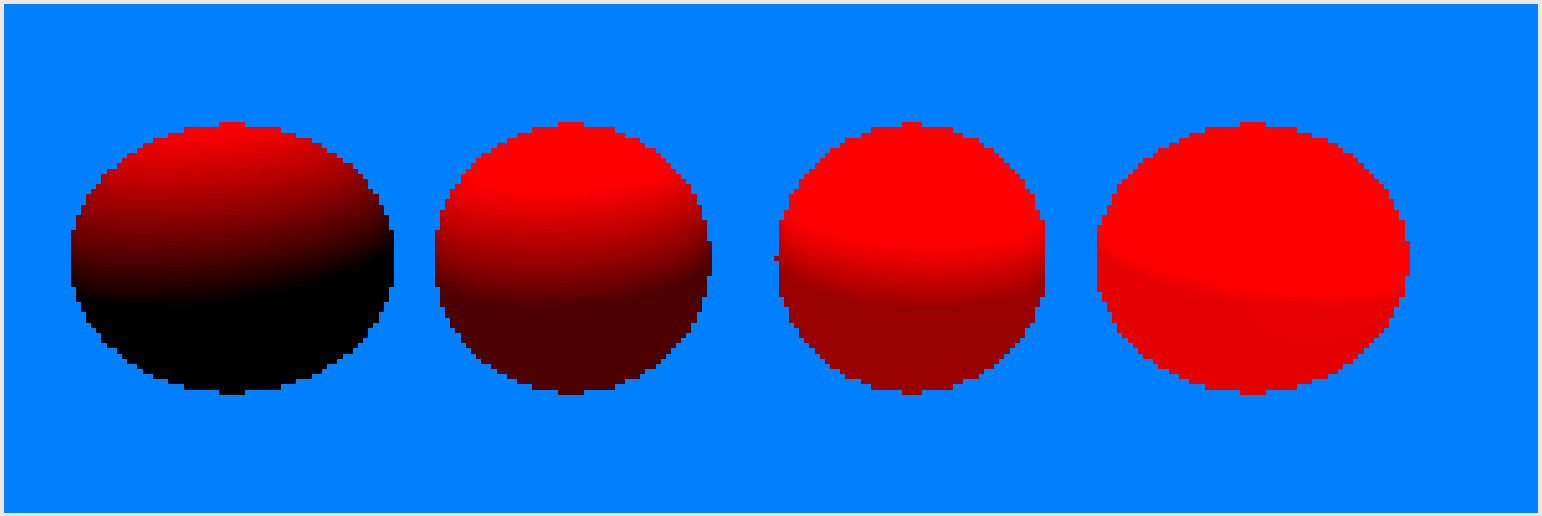
- Pode-se combinar as contribuições (difusas) devidas à luz ambiente e à fonte de luz pontual
  - caso contrário o objeto só será visível caso receba iluminação direta da fonte, o que está longe da realidade!
- Constante K_a introduzida para controlar a intensidade da iluminação ambiente para cada superfície:

$$I_{difusa} = I_a K_a + K_d I_l (\mathbf{N} \cdot \mathbf{L})$$

Modelo de Phong: Reflexão Difusa

Modelo Local

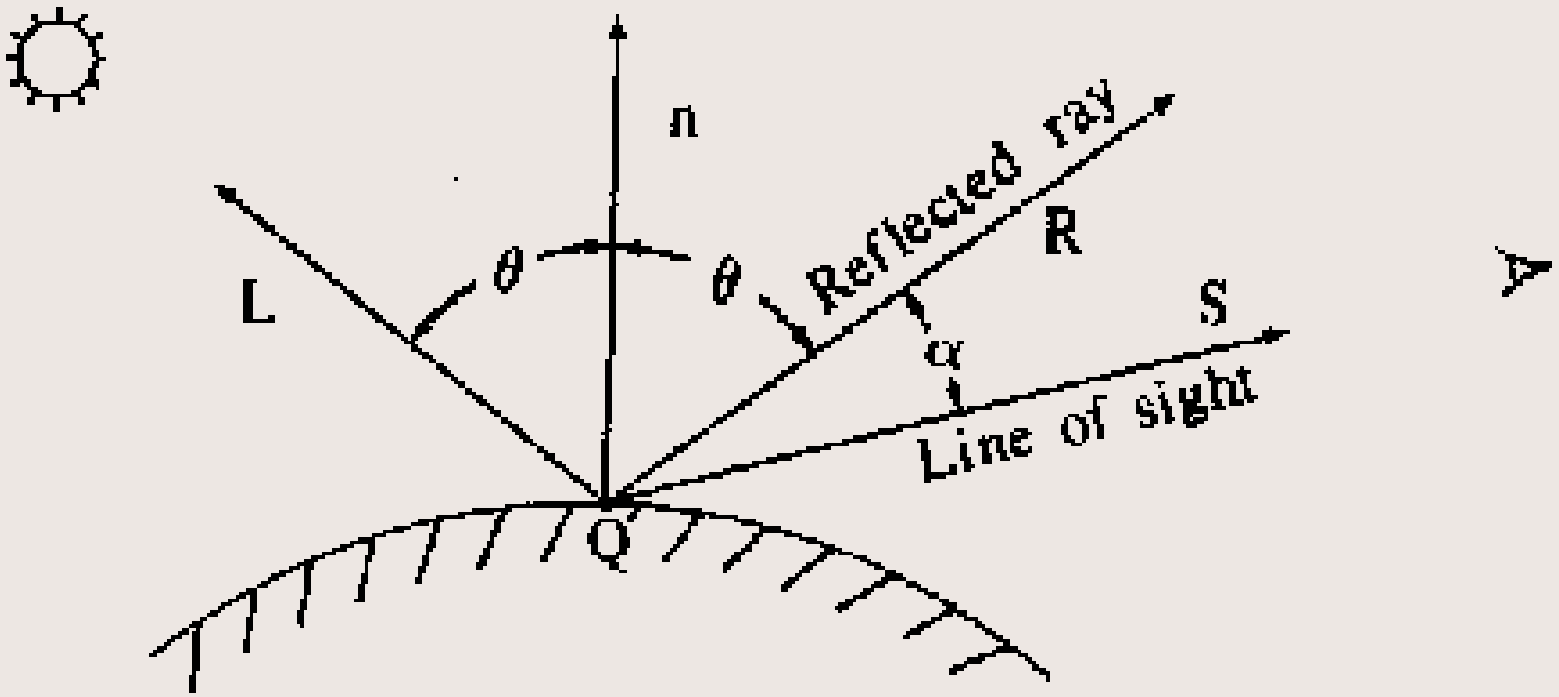
Iluminação Difusa + Ambiente



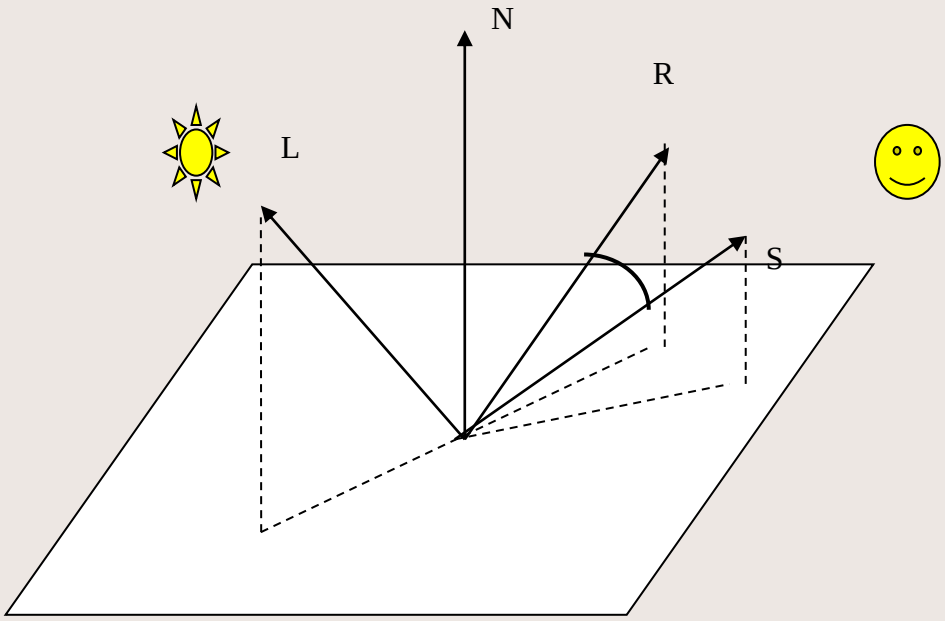
Modelo de Phong: Reflexão Especular

- Resultado da reflexão quase total da luz incidente em uma região concentrada em torno de um ângulo de reflexão especular
- Ângulo formado entre a direção de reflexão especular ideal, **R**, e a direção de observação, **S**
- Para um refletor ideal (espelho), **S** e **R** coincidem, e $\alpha = 0$

Modelo de Phong: Reflexão Especular



Vetores no modelo de Phong

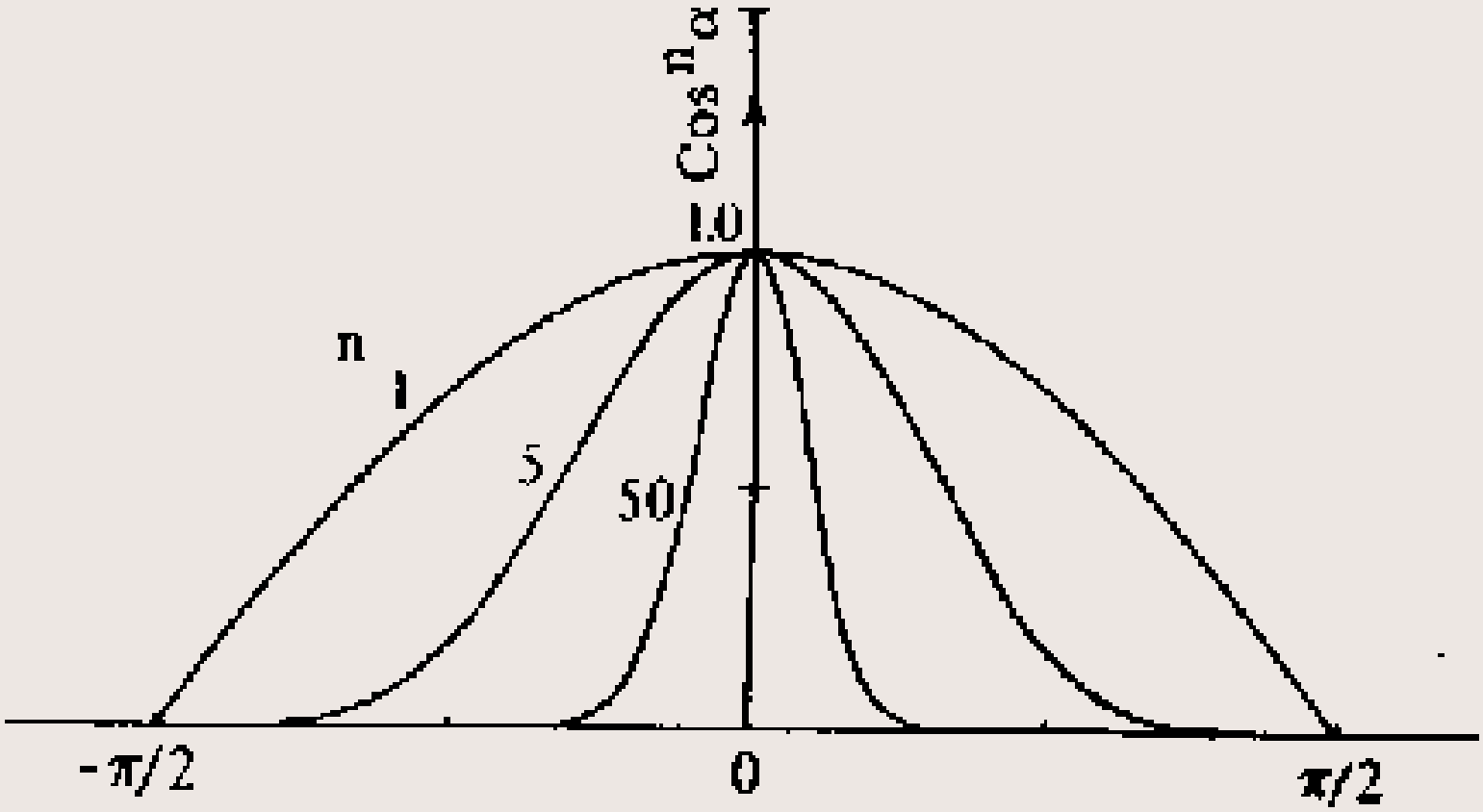


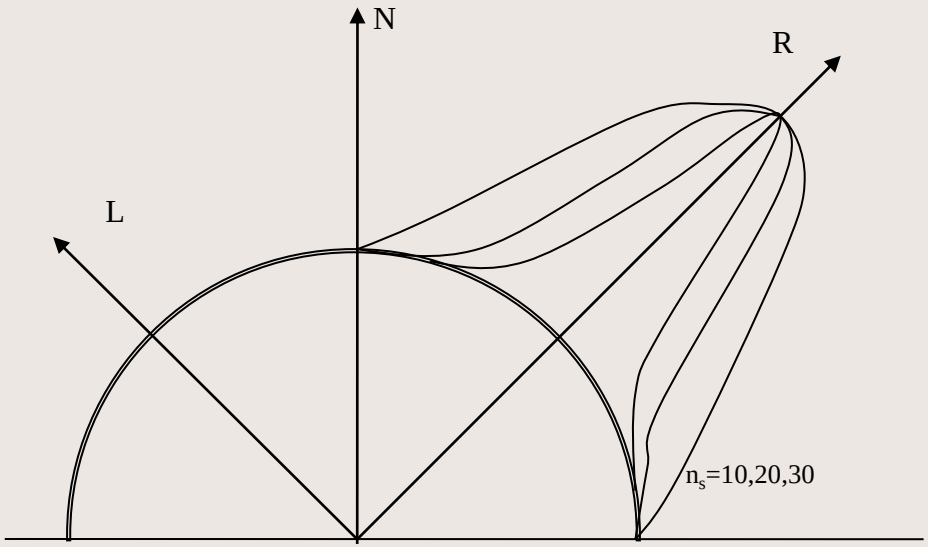
Direções no modelo de Phong

Modelo de Phong: Reflexão Especular

- Superfície idealmente especular: toda luz incidente refletida na direção **R**
  - a luz refletida só será vista se a direção de observação e a direção de reflexão coincidirem.
- Objetos refletores não ideais: reflexão especular em uma região finita ao redor do vetor **R**
  - quanto mais refletora (polida) a superfície, menor a amplitude dessa região
  - a variação na intensidade especular em função do ângulo de incidência é descrita pela Lei de Fresnel
  - Phong propôs um modelo empírico para modelar esse comportamento, que define a intensidade da reflexão proporcional a $\cos^n \alpha$, $\alpha \in [0, 90^o]$.

Modelo de Phong: Reflexão Especular





Variação da radiância reflectida com V, para uma direção de incidência L e vários n_s

Modelo de Phong: Reflexão Especular

- Valor de *n* determinado pelo tipo de superfície: n grande (> 100) para superfícies mais polidas, n pequeno (até 1) para superfícies mais opacas.
- Intensidade da reflexão especular depende de fatores:
  - propriedades do material, ângulo de incidência, distribuição espectral da luz incidente
  - Variações da intensidade especular (para luz monocromática) podem ser aproximadas por uma função coeficiente de reflexão especular, definida para diferentes superfícies (materiais) $W(\theta,\lambda)$.
  - em geral, $W(\theta,\lambda)$ aumenta a medida que aumenta θ. A variação da intensidade da reflexão especular em função do ângulo de incidência é governada pela Lei de Fresnel.

Modelo de Phong: Reflexão Especular

- O termo especular de Phong é descrito por

$$I_{specular} = W(\theta, \lambda) I_l \cos^n \alpha$$

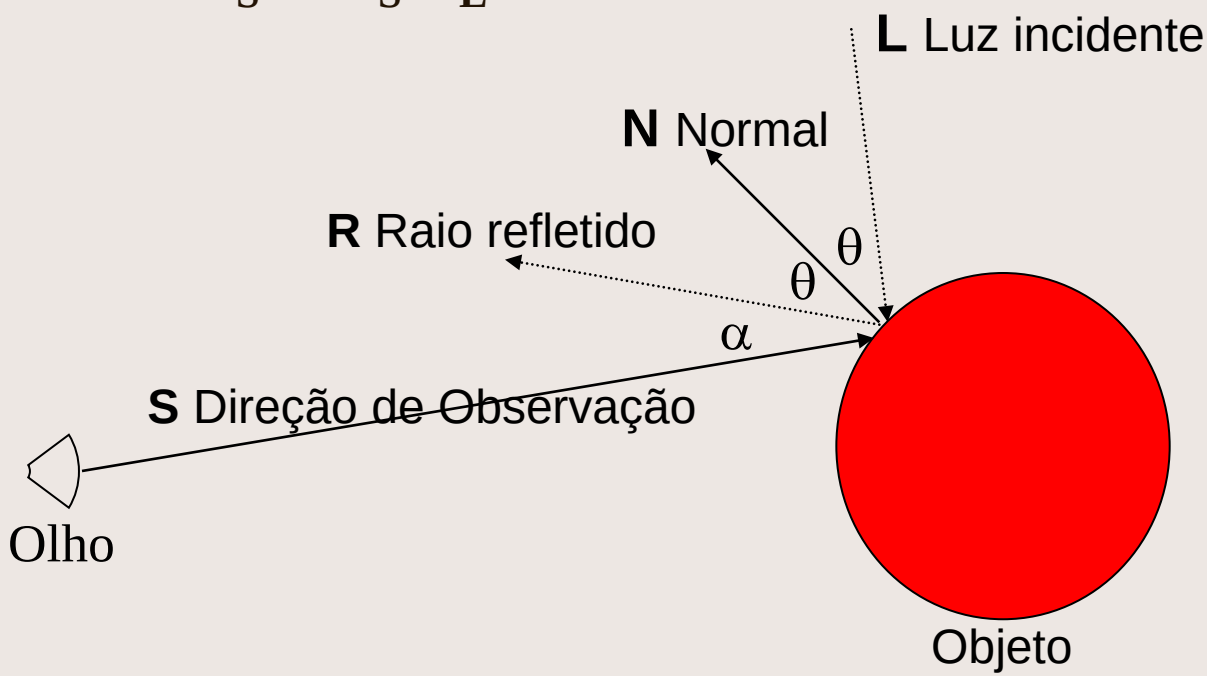
- Para materiais opacos, a reflexão especular é aproximadamente constante para todos os ângulos de incidência $\Rightarrow$ Phong aproximou a função por uma constante: $I_s = K_s I_l (\mathbf{S} \cdot \mathbf{R})^n$
- o vetor **R** pode ser calculado a partir de **L** e **N**
- múltiplas fontes de luz: soma as contribuições de cada uma

Modelo de Phong: Reflexão Especular

Modelo Local

Iluminação Especular (por *Phong Bui Tuong*)

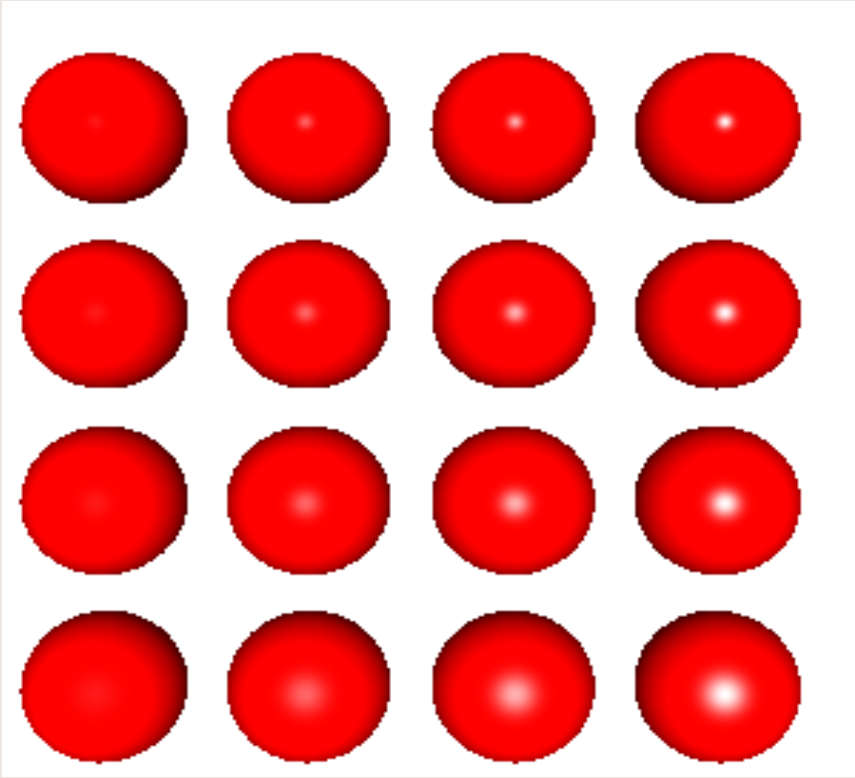
$$I_S = K_S * I_L * \cos^n \alpha$$



Modelo de Phong: Reflexão Especular

Modelo Local

Iluminação Especular



Modelo de Phong: luzambiente

Modelo Local

Iluminação Ambiente

$$I_A = K_A * I_L$$

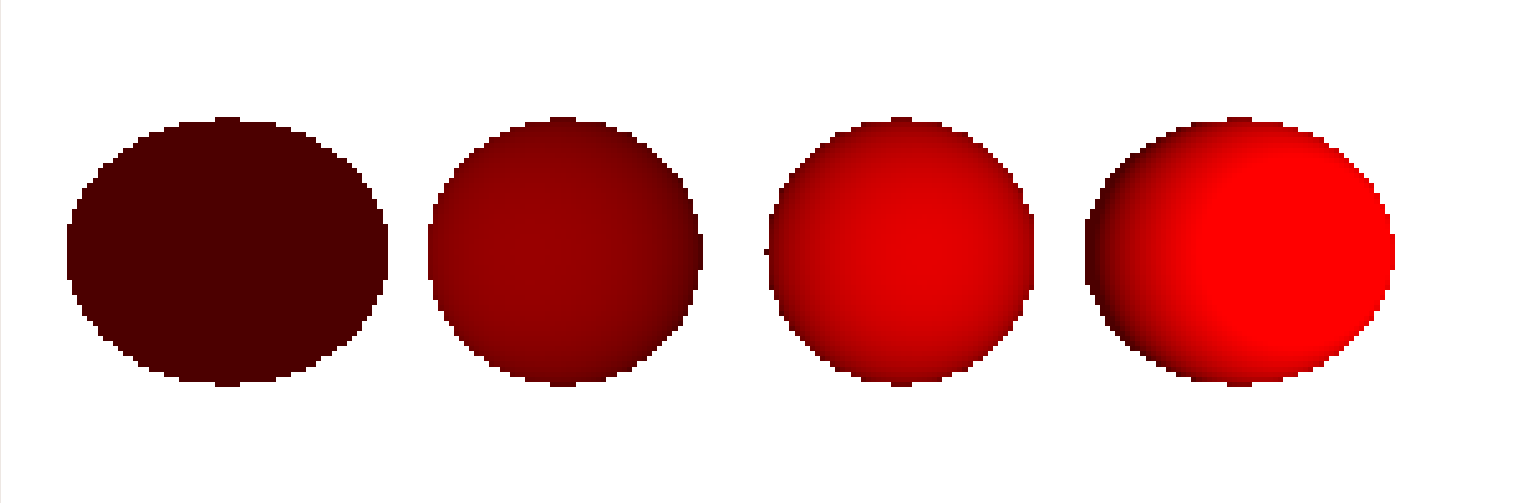
Iluminação Difusa → Cor do Objeto

Iluminação Especular → Brilho do Objeto

Modelo de Iluminação

Modelo Local

Iluminação Ambiente



Modelo de Iluminação e Métodos de *Rendering*

Modelo de Phong

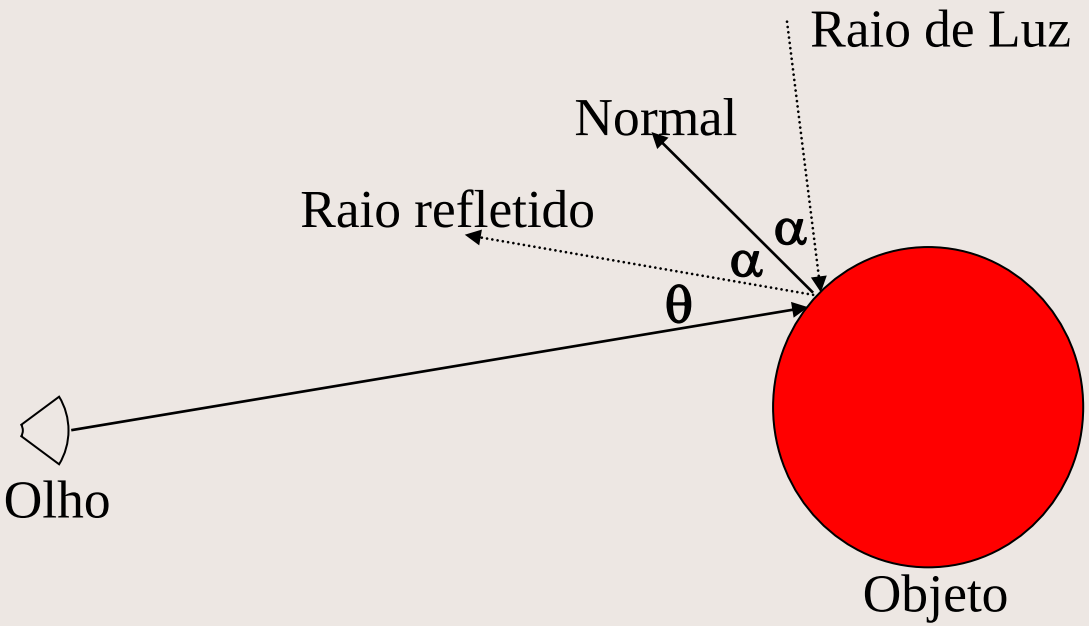
- Ver tb. http://alpha.mini.pw.edu.pl/~kotowski/Grafika/IlluminationModel/Index.html
- Imagens nos exemplos a seguir: curso CG Ken Brodlie, University of Leeds: http://www.comp.leeds.ac.uk/kwb/gi21/

  (imagens por Alan Watt)

Modelo de Phong completo

Modelo Local Completo

$$I = I_A + I_D + I_S$$



Modelo de Phong completo

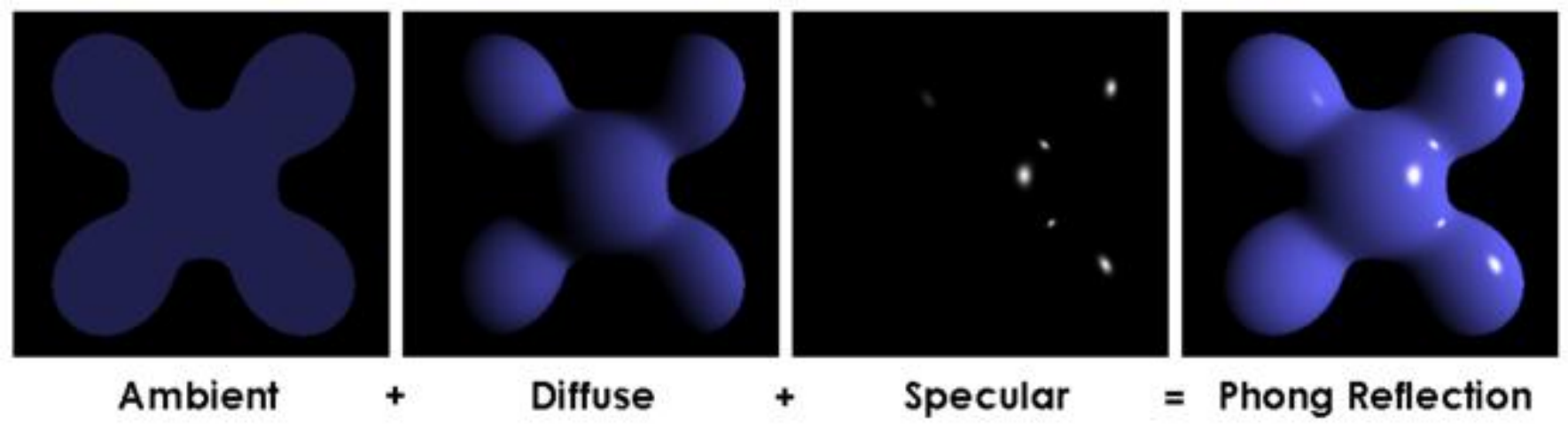
- Componente ambiente
  - Captura o efeito de uma certa quantidade de luz atingindo a superfície vinda igualmente de todas as direções
  - Não associada a uma fonte emissora
  - Constante sobre toda a superfície
  - Não depende da normal à superfície, nem do ponto de observação
- Componente difusa
  - Captura o efeito da luz sendo refletida igualmente em todas as direções (como uma superfície opaca/rugosa espalha a luz)
  - Associada a uma fonte de luz pontual ou direcional
  - Depende da direção da luz e da normal à superfície
  - Intensidade é maior na região em que as normais à superfície se aproximam da direção da fonte de luz

Modelo de Phong completo

- Componente especular
  - Captura o efeito da luz sendo refletida por uma superfície lisa/polida
  - Luz refletida (*highlights*) em uma direção preferencial (como reflexão em um espelho perfeito, que ocorre em uma direção apenas)
  - Depende da normal à superfície, do ponto de observação, e da posição da fonte de luz
- Exemplo...

Modelo de Phong completo

http://en.wikipedia.org/wiki/Phong_shading



v. tb.

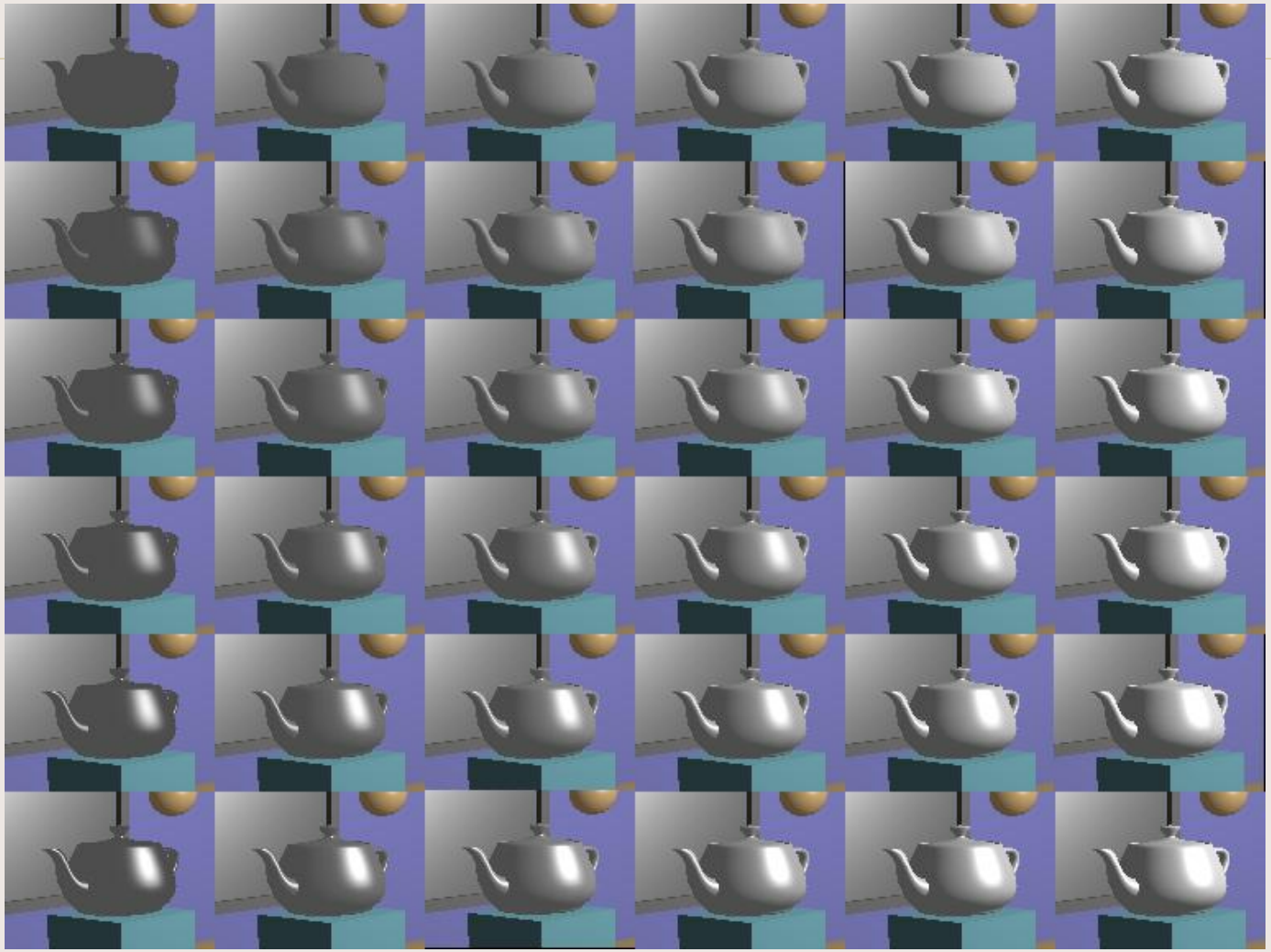
http://www.inf.ufsc.br/~awangenh/CG/raytracing/iluminacao.html

Example - Ambient Reflection

Example - Ambient and Diffuse

Ambient, Diffuse and Specular

Phong illumination model: K_s 0.0 to 1.0, K_d 0.0 to 1.0 (K_a = 0.7, n = 10.0)



K_s

K_d

Phong Illumination Model: K_s 0.0 to 1.0; n = 10.0 to 810.0 (K_a = 0.7, K_d = 1.0)

n

K_s

Modelo de Phong completo

- Múltiplas fontes de luz (digamos, *m*):

$$I = I_a K_a + \sum_{j=1,m} I_{lj}\{K_d(\mathbf{N} \cdot \mathbf{L}) + K_s(\mathbf{R} \cdot \mathbf{S})\}$$

- Incorporação de cor: a cor da luz refletida é uma função do comprimento de onda da luz incidente
  - a equação de iluminação deve ser expressa como uma função das propriedades de cor das fontes de luz e das superfícies dos objetos.
  - cm geral, superfícies são iluminadas por fontes de luz branca
  - No modelo RGB: especifica-se os componentes RGB que descrevem a luz das fontes (I_{lj}) e as cores das superfícies (K_d e K_s)

$$I_R = I_{aR} K_{aR} + \sum_{j=1,m} I_{ljR}\{K_{dR}(\mathbf{N} \cdot \mathbf{L}) + K_{sR}(\mathbf{R} \cdot \mathbf{S})\}$$

Incorporação de Cor

- Uma forma de definir as cores das superfícies é especificar seus coeficientes de reflexão em termos de seus componentes RGB (K_{dR}, K_{dG}, K_{dB}, idem para K_s e K_a)
  - expressos como triplas RGB (no intervalo [0,1])
- calcula-se uma aproximação para a cor amostrando a função de iluminação nos 3 comprimentos de onda correspondentes às três primárias R, G, B.

Melhorias no modelo

- Incorporação de Cor
  - Amostragem limitada do espectro da luz emitida, nas faixas de comprimento de onda correspondentes a R, G e B.
  - A intensidade calculada (3 valores no intervalo [0,1] será quantizada para valores inteiros no intervalo [0,255]).
  - originalmente, Phong setou Ks como uma constante independente da cor $\Rightarrow$ reflexões especulares da mesma cor da luz incidente (em geral, branca) (aparência plástica).

Melhorias no modelo

- Atenuação devida à distância
  - energia radiante vinda de uma fonte pontual é atenuada por um fator quadrático ($1/d^2$) $\Rightarrow$ superfície mais distante da fonte recebe menos luz.
  - na prática, é usado um fator de atenuação linear em relação à distância (1/d, ou uma função mais complexa) para garantir uma variação mais suave.

Melhorias no modelo

- Transparência
  - superfícies transparentes, em geral, refletem e transmitem luz.
  - as equações de iluminação devem ser modificadas para incluir a contribuição da luz que passa pela superfície (vinda de objetos refletores posicionados atrás dela).
  - Transmissão difusa e especular: efeitos realistas requerem um modelo de refração da luz

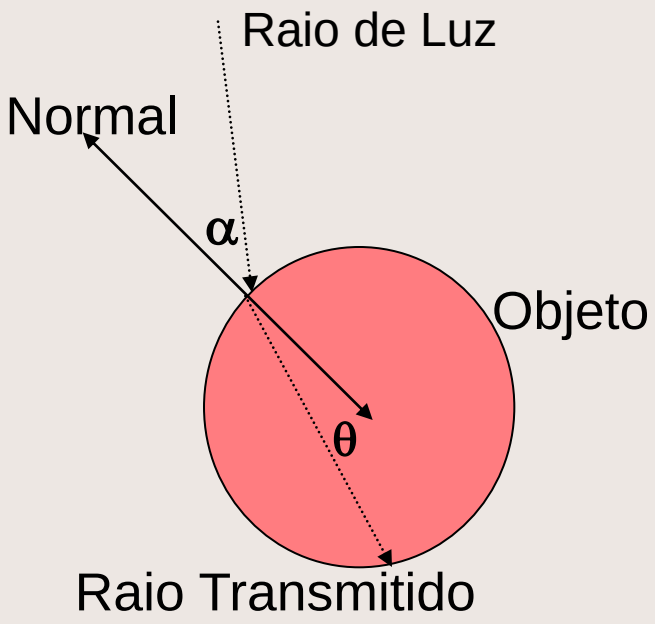
Melhorias no modelo

- Transparência
  - *Lei de Snell*: determina a direção da luz refletida, a partir da direção da luz incidente e dos coeficientes de refração de cada material
  - esse índice é, na verdade, uma função do comprimento de onda, mas é aproximado por uma constante
  - a partir da Lei de Snell pode-se determinar o vetor unitário que dá a direção do raio refratado

Transparência

Lei de Snell

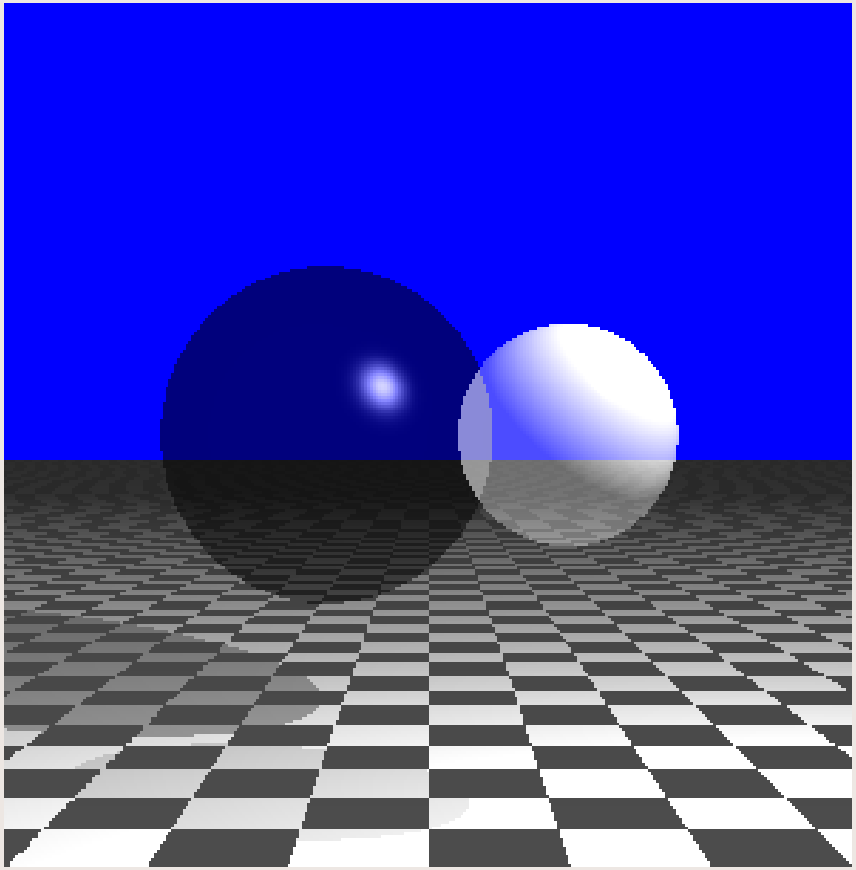
$$\eta_{\alpha} * \sin \alpha = \eta_{\theta} * \sin \theta$$



Melhorias no modelo

- Transparência
  - uma abordagem simplista ignora o desvio, e simplesmente combina a intensidade calculada para a superfície transparente (superfície 1) com a intensidade calculada para outra superfície 2, visível através dela, segundo um fator de transparência t:
    - $I = (1 - t)I_1 + tI_2 \; 0 \leq t \leq 1$
  - aproximação linear não adequada para superfícies curvas, ou objetos que espalham luz, como nuvens...

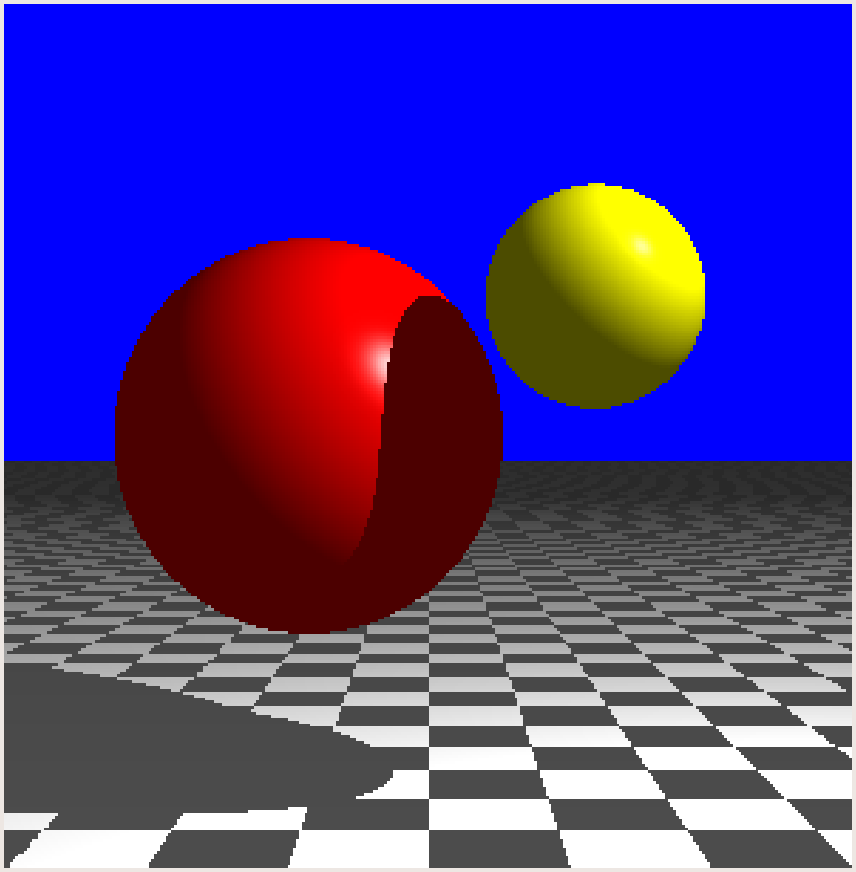
Transparência por interpolação: exemplo



Melhorias no modelo

- Sombras
  - importante para realismo e *depth cueing*
  - umbra e penumbra
  - precisa localizar as áreas em que as fontes de luz produzem sombra

Sombras: exemplo

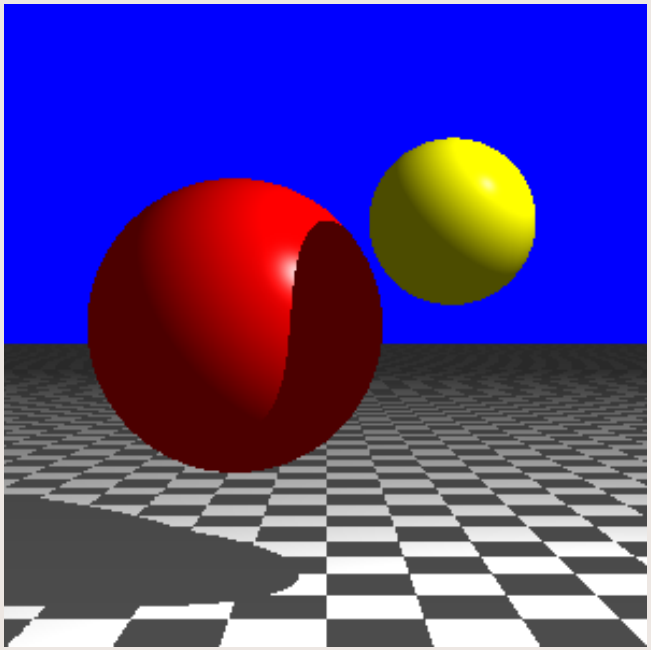


Sombra

Modelo Global

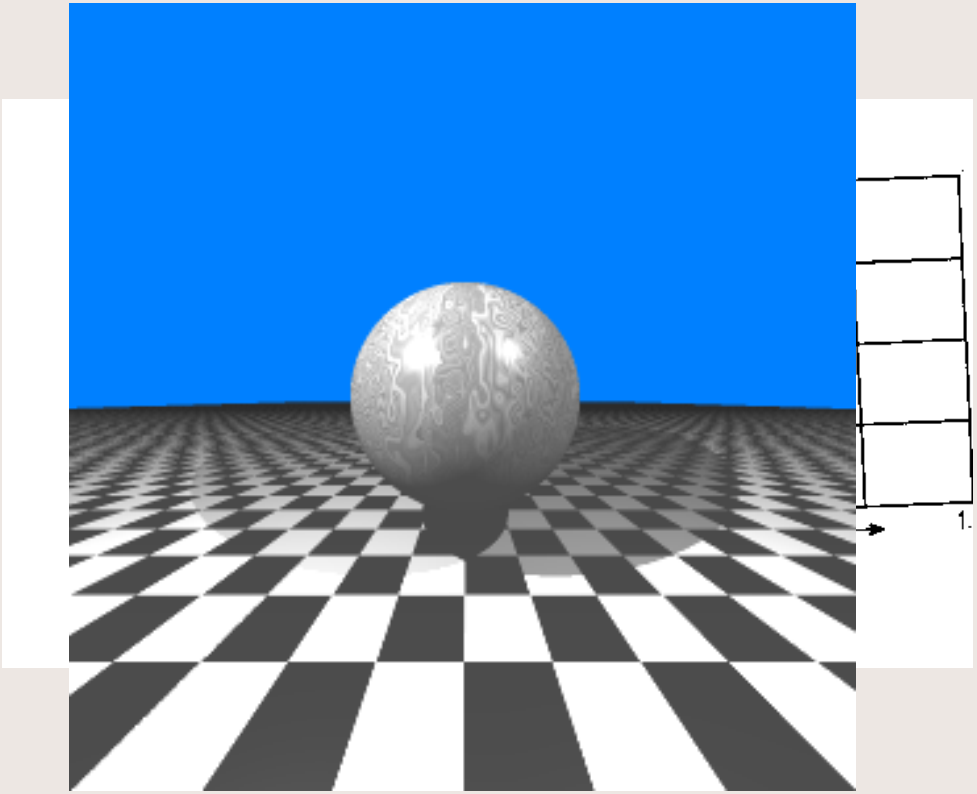
Sombras

Detecção de Pontos Não Iluminados Diretamente

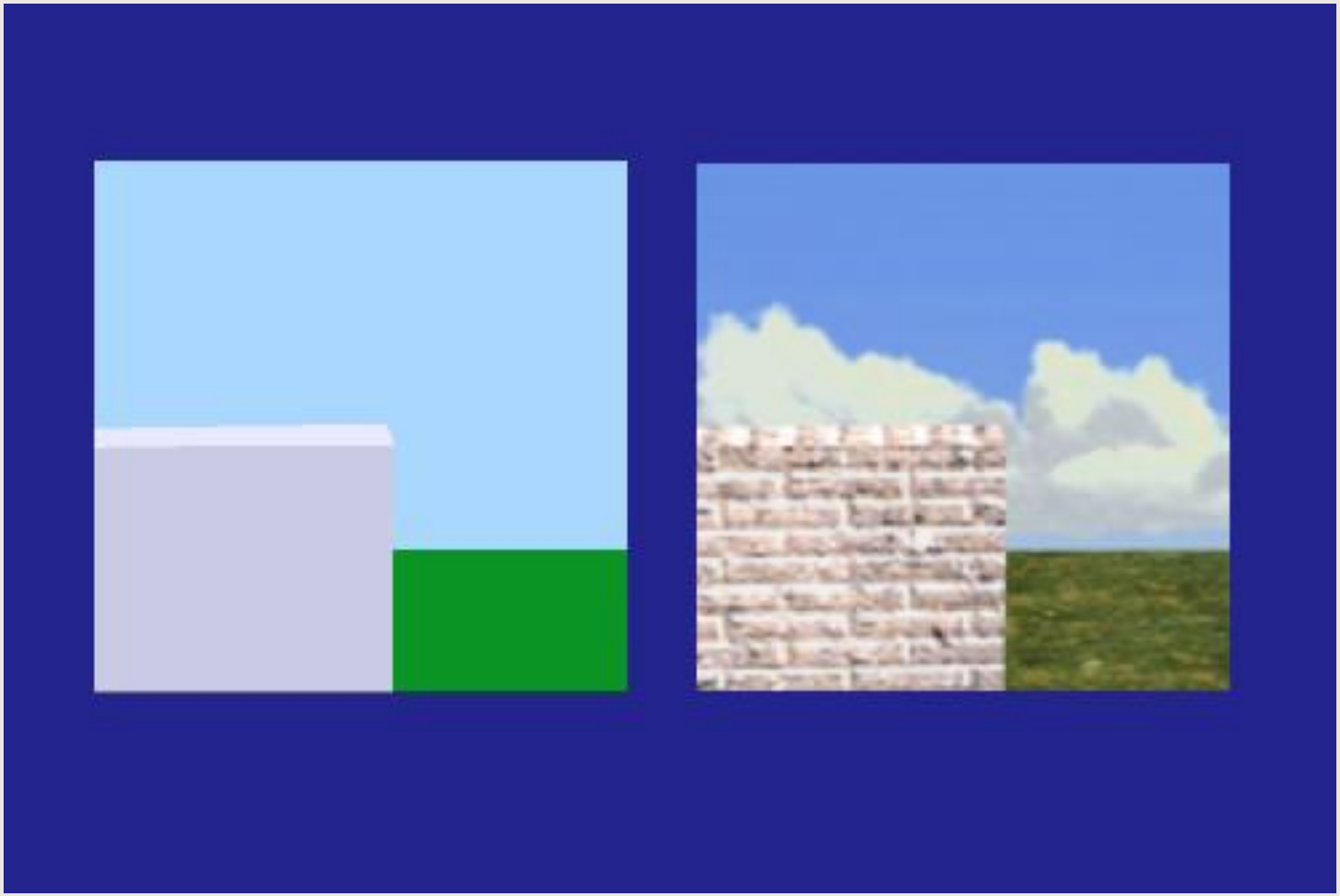


Se Ponto é Iluminado ($I_L = 1$) senão ($I_L = 0$)

Textura



Textura: exemplo



Textura: exemplo

Modelo de Iluminação Global

- modelo local completo +
  - sombras
  - reflexões múltiplas
  - transparência
  - texturas

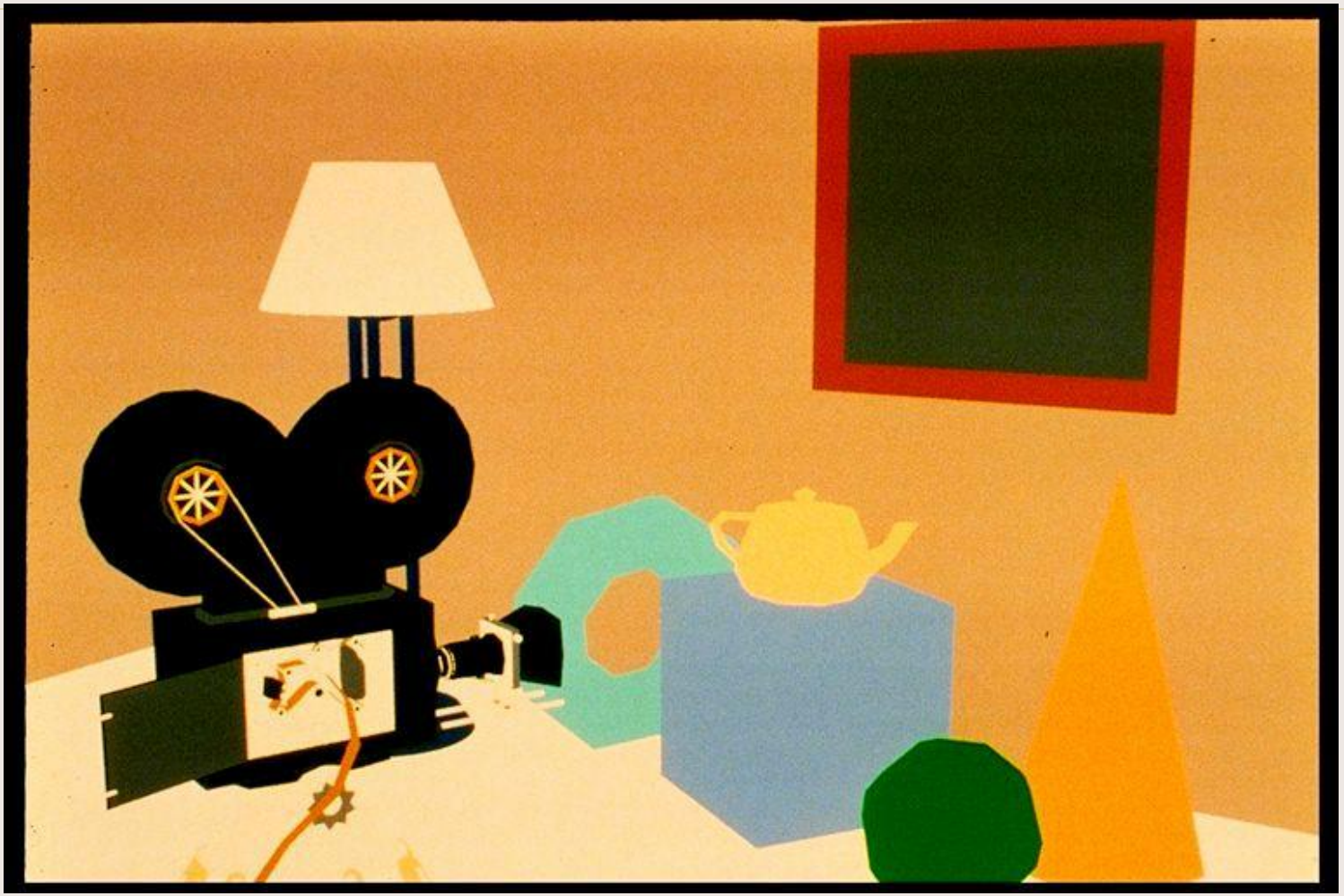
Modelos de *Shading* (tonalização)

- um método para aplicar um modelo de iluminação local a um objeto (em geral, modelado como uma malha poligonal)
- Normalmente, o método de *shading* é integrado a um algoritmo *scanline (scanline graphics)*
  - o processo de tonalização é feito para cada face visível dos modelos que compõem a cena, para determinar a cor (tom, intensidade) associada a cada ponto visível da face
  - seria muito custoso calcular o modelo de iluminação em cada ponto de cada face visível para determinar a cor

Modelos de *Shading*

- 4 modelos:*Constant*, *Faceted*, *Gouraud*, e *Phong*
  - ordem crescente de qualidade de imagem e de custo computacional
- *Constant Shading*
  - calcula uma única cor (tom, or *shade*) para todo o objeto (todas as faces)
  - não há variações de tonalidade ao longo do objeto, i.e., na verdade, não há *shading*.

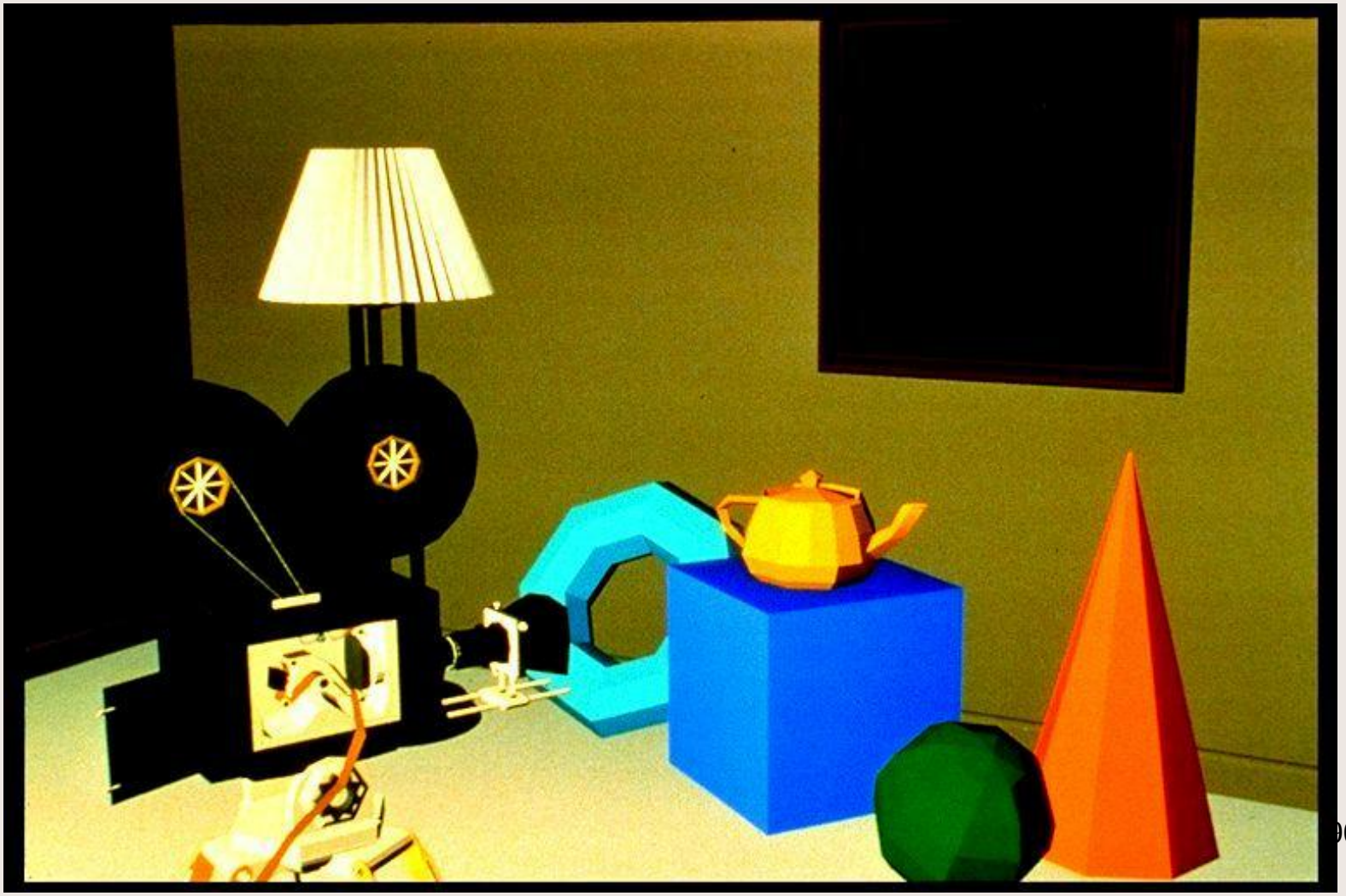
Constant Shading



Flat shading

- modelo mais simples: calcula uma cor (tonalidade) para cada polígono (face)
- Toda a face associada a uma cor única, calculada aplicando o modelo de iluminação
- vetor **L** no modelo: vai de qualquer ponto no polígono à posição da fonte de luz
- em geral, usa apenas os termos ambiente e de reflexão difusa do modelo de iluminação
- Simples e rápido, mas arestas entre faces são acentuadas
- Em OpenGL: glShadeMode(GL_FLAT)

Flat shading

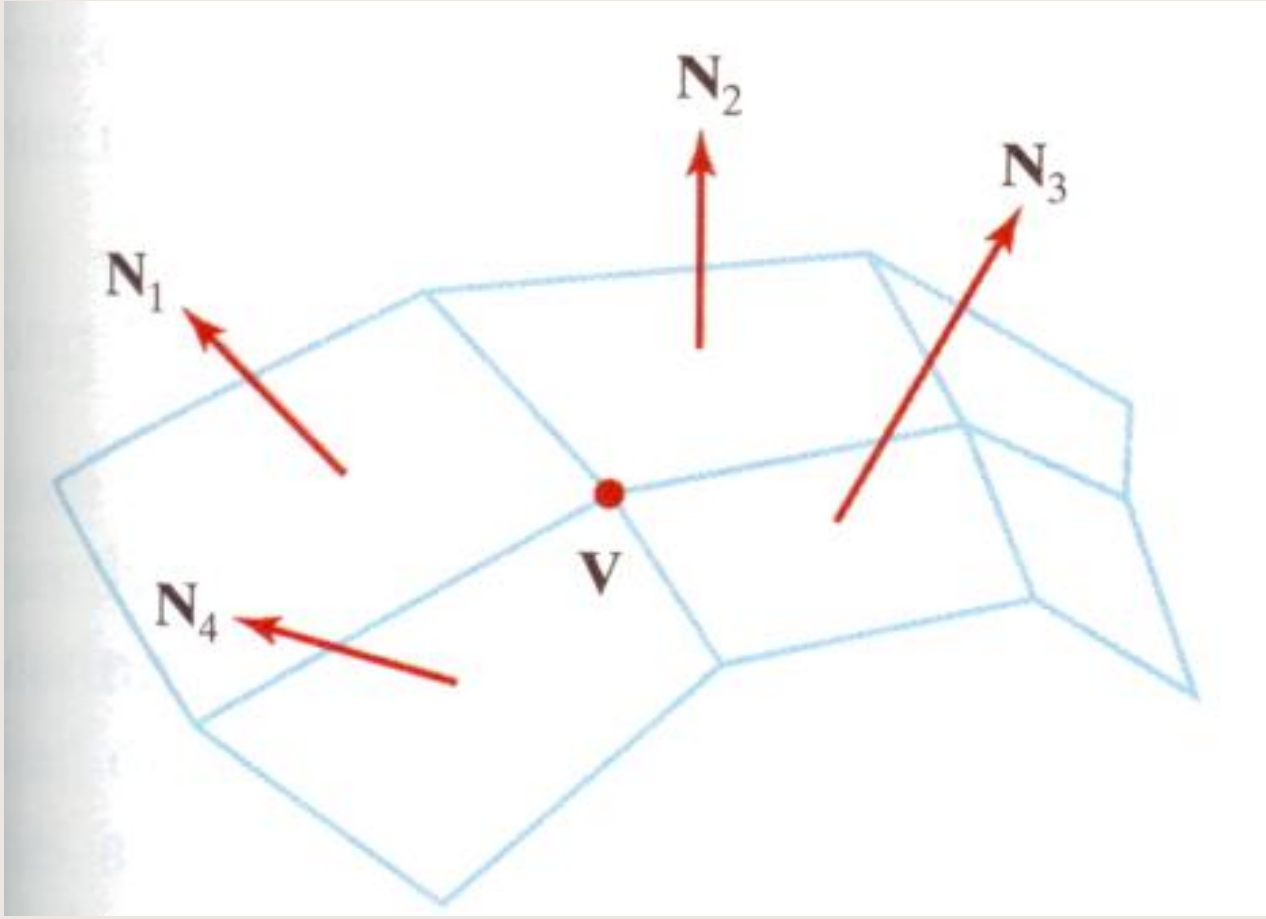


Gouraud Shading

- Interpola cores: aplica o modelo de iluminação nos vértices de cada face poligonal para obter a cor (intensidade) em cada vértice da face
- interpola os valores obtidos nos vértices (I_R,I_G,I_B) para determinar a cor nos pontos interiores aos polígonos
- interpolação bi-linear das intensidades ao longo das linhas de varredura

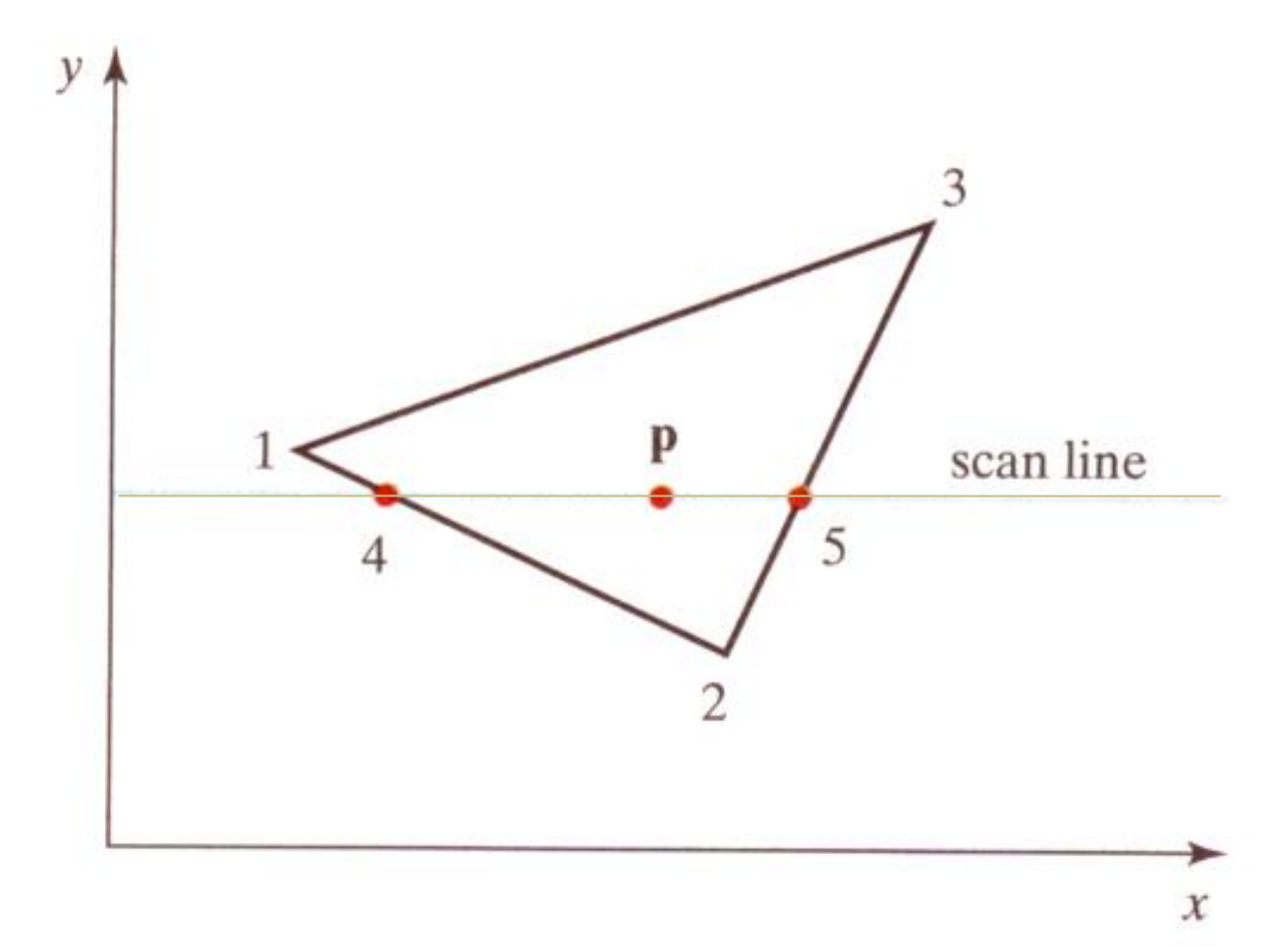
Gouraud Shading

Fonte: Hearn & Baker



Gouraud Shading

Fonte: Hearn & Baker



Gouraud Shading: Algoritmo

1. determina a normal **N** em cada vértice do polígono
2. usa **N** e **L** para calcular a intensidade *I* em cada vértice do polígono (usando o modelo de iluminação)
3. usa interpolação bi-linear para calcular a intensidade $I_{R,G,B}$ em cada *pixel* no qual o polígono visível é projetado
4. “pinta” o *pixel* de acordo com a cor determinada

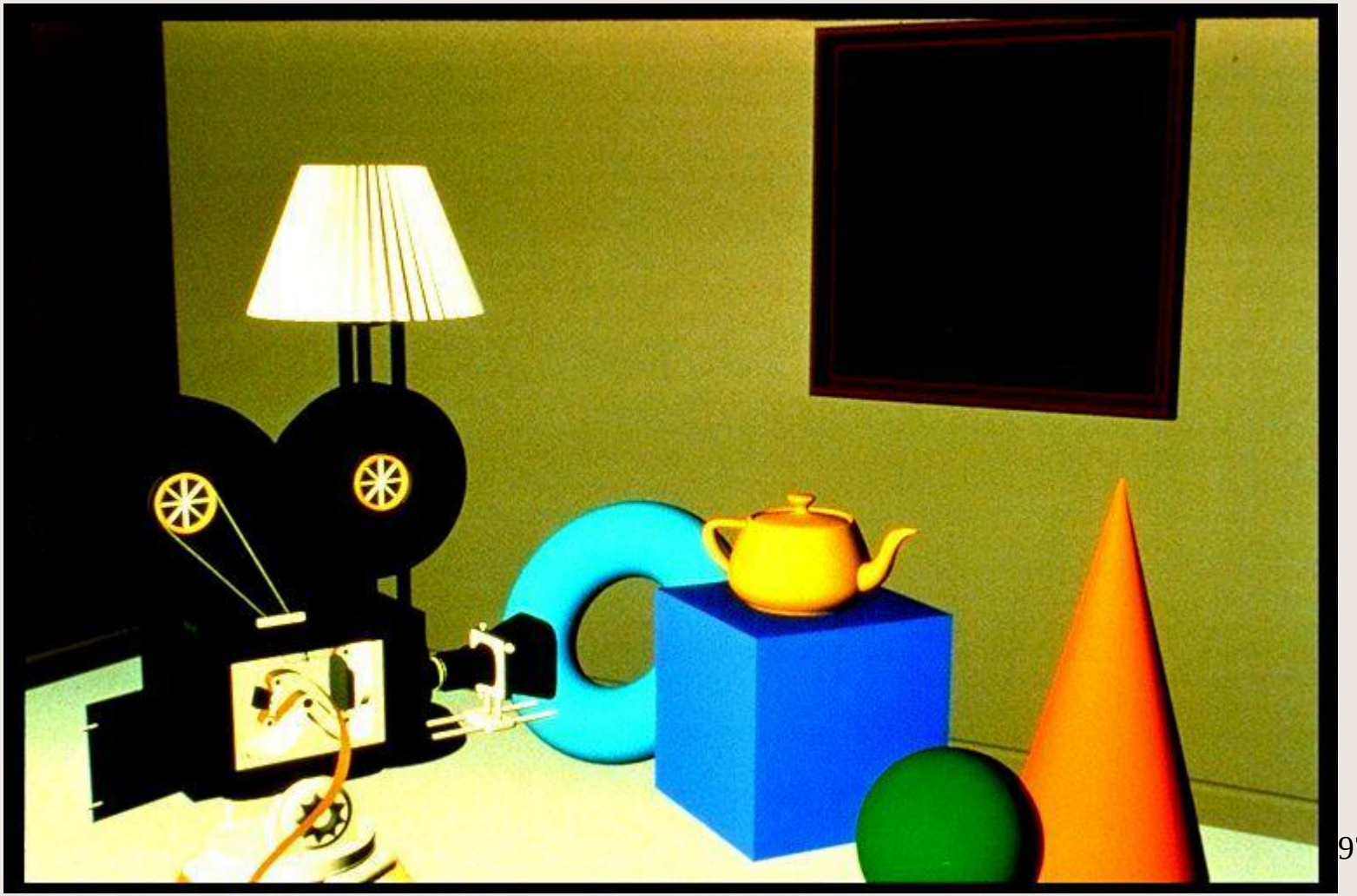
Gouraud Shading

- Como calcular **N** para um vértice?
  - podemos tomar a média das normais às faces que compartilham o vértice... (precisa buscar essa informação na estrutura de dados...)
- e a interpolação bi-linear?
  - interpola os valores em 2 vértices para obter os valores nas arestas formadas por eles
  - para cada linha de varredura interpola os valores nas arestas para obter o valor em cada pixel no interior

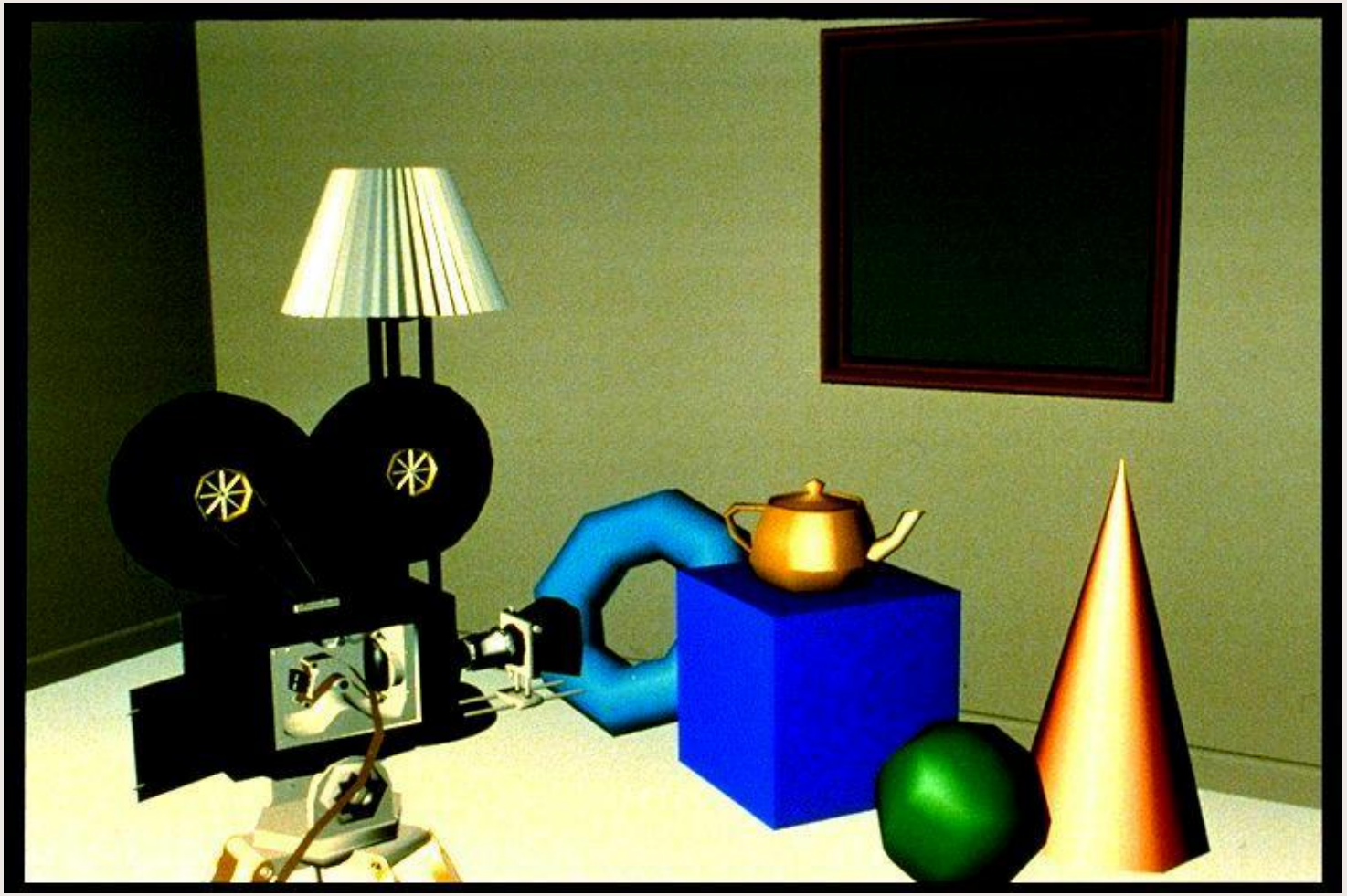
Gouraud Shading

- suaviza as transições entre faces: aparência muito melhor que o '*faceted*'
- não é muito caro computacionalmente
- por outro lado, suaviza faces que deveriam ser mantidas (p. ex., cubo)
- não captura bem os *highlights* especulares, porque as intensidades são computadas apenas nos vértices

Gouraud Shading (sem *highlight* especular)



Gouraud Shading (com *highlight* especular)

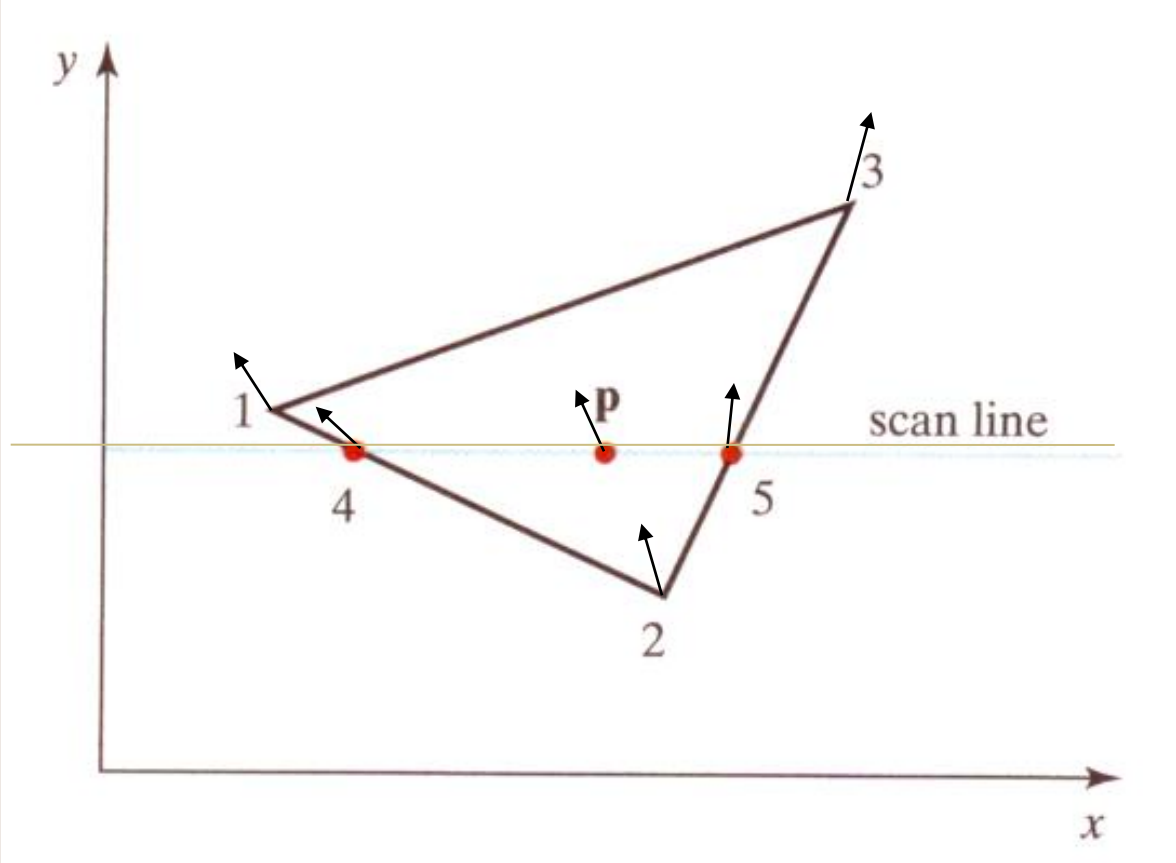


Phong Shading

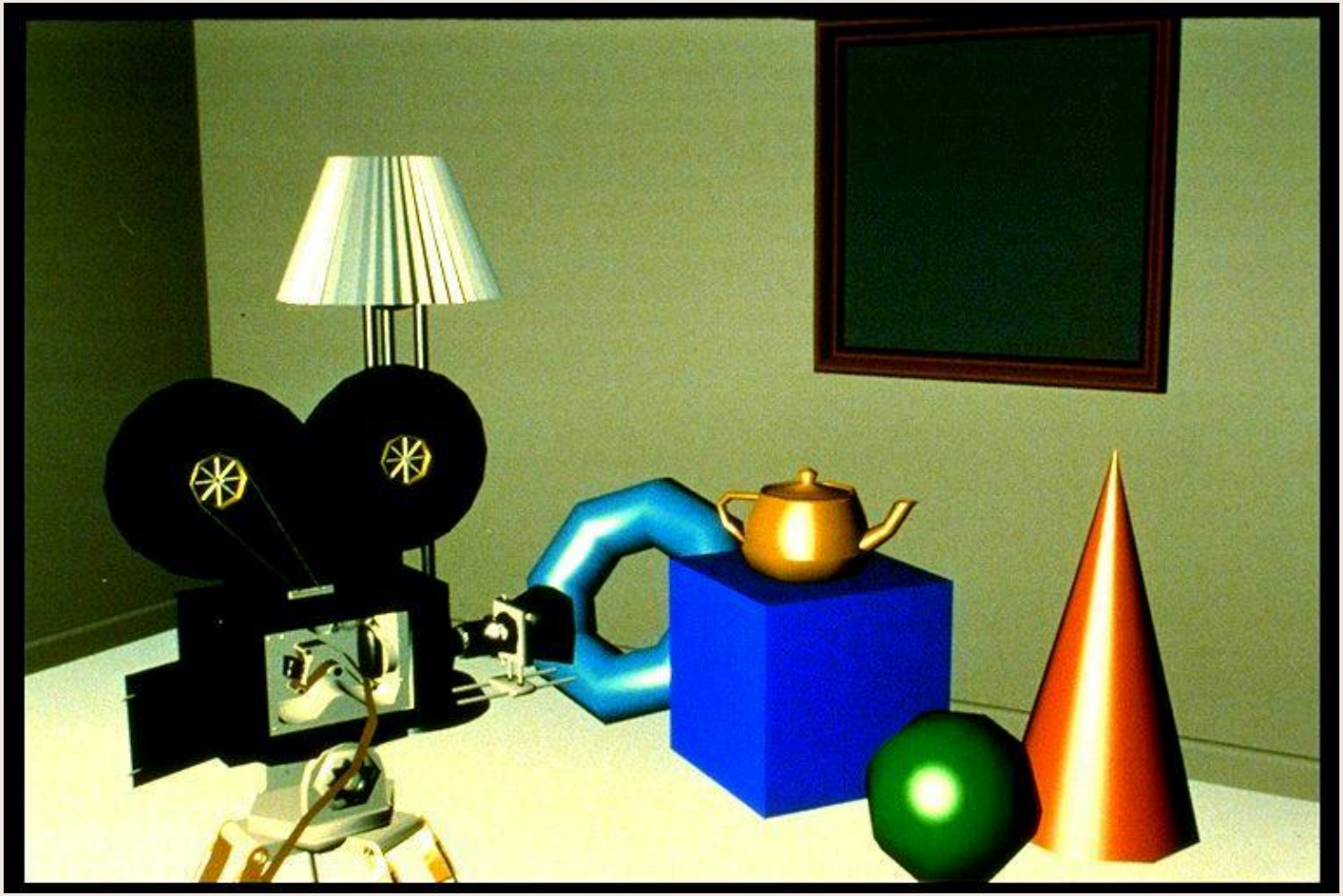
- Calcula as normais nos vértices, interpola para determinar a normal em cada ponto da face
  - Normais em pontos ao longo de uma aresta calculadas por interpolação linear dos valores nos vértices (e precisam ser re-normalizadas)
  - Normais em pontos no interior da face calculadas por interpolação linear das normais nas arestas (e re-normalizadas)
- Aplica o modelo de iluminação de Phong em cada ponto visível do polígono para determinar I
- Melhor que *Gouraud* para capturar *highlights* especulares
- Custo computacional muito maior

Phong Shading

Fonte: Hearn & Baker

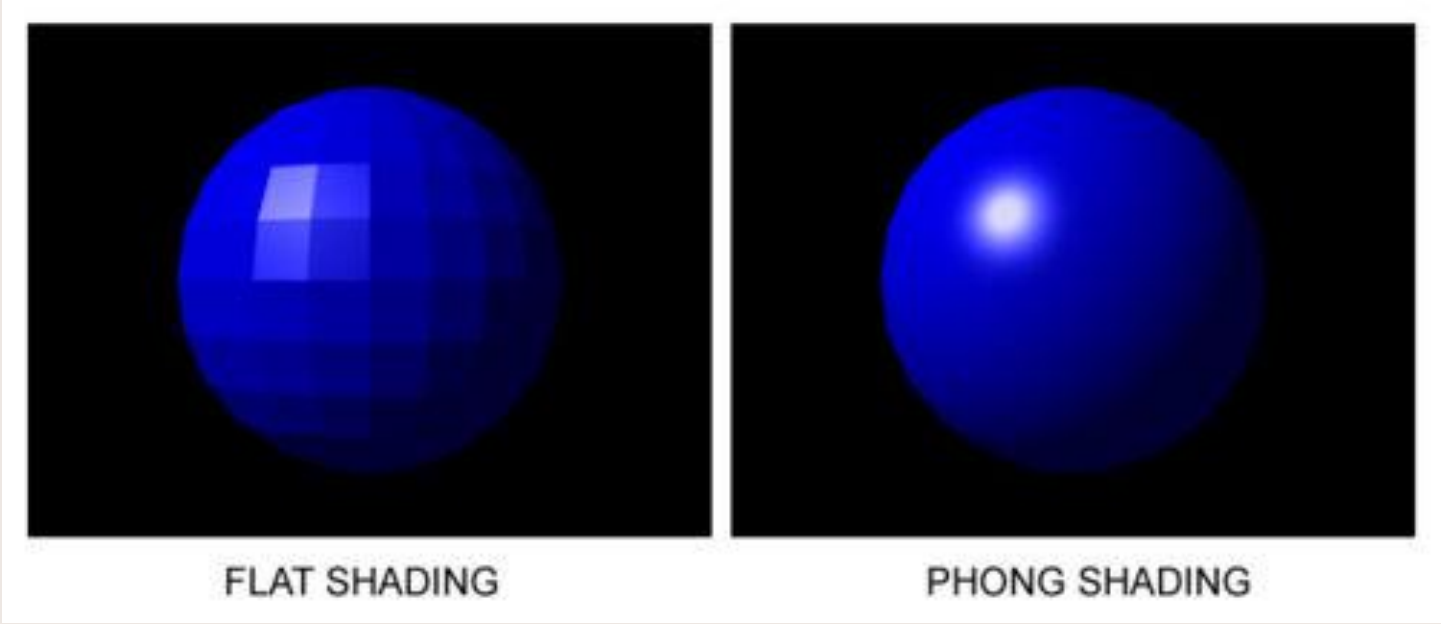


Phong Shading



Phong vs Flat

http://en.wikipedia.org/wiki/Phong_shading



Observação

- OPenGL suporta 2 tipos de shading:

  `glShadeModel(renderingType)`

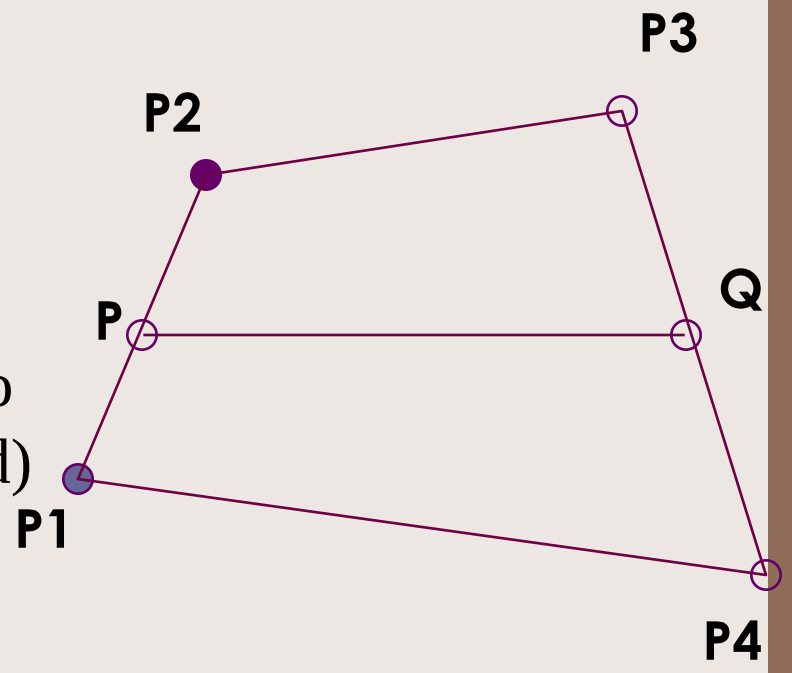
  `GL_FLAT:` flat shading

  `GL_SMOOTH:` Gouraud shading

- Porque não Phong?
  - Phong requer que as normais sejam passadas ao longo do *rendering pipeline* para o ‘screen space’
  - OpenGL tonaliza os vértices em *viewing coordinates* e em seguida descarta as normais: impossível fazer Phong shading

Observação - anomalias

- Interpolação é executada no ‘screen space’, depois da transformação perspectiva
  - Suponha P2 bem mais distante que P1. P está no meio em *screen space*, então recebe intensidade (Gouraud) ou normal (Phong) 50 : 50
  - no SRU, P está na verdade mais próximo de P1 do que de P2



Fontes de Luz

- Para ativar fonte de luz:
  - **`glEnable (source);`**
  - ***Source:*** constante cujo nome é ***GL_LIGHT_i***, começando com ***`GL_LIGHT0`***
  - Quantas?
    - **`glGetIntegerv( GL_MAX_LIGHTS, &n );`**
- ativar cálculo de cores pelo modelo de iluminação
  - **`glEnable (GL_LIGHTING);`**

Fontes de Luz

- Para configurar propriedades de cada fonte:
  `glLightfv(`*`source, property, value`*`);`
  - **`Property`** é uma constante designando:
    - Coeficientes de cor no modelo de iluminação
      - ***`GL_AMBIENT, GL_DIFFUSE, GL_SPECULAR`***
    - Geometria da fonte
      - ***`GL_POSITION, GL_SPOT_DIRECTION`***,
        `GL_SPOT_CUTOFF, GL_SPOT_EXPONENT`
    - Coeficientes de atenuação
      - ***`GL_CONSTANT_ATTENUATION,`***
        `GL_LINEAR_ATTENUATION,`
        `GL_QUADRATIC_ATTENUATION`

Propriedades de Material

- Especificadas por

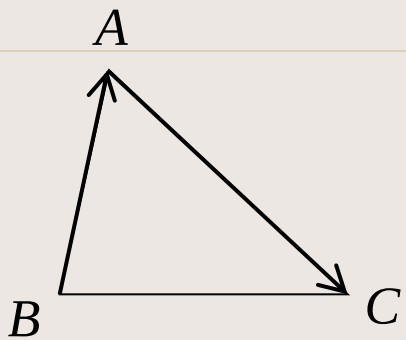
  glMaterialfv (*face, property, value*)

  - ***Face*** indica quais lados da superfície se quer configurar:
    - ***GL_FRONT, GL_BACK, GL_FRONT_AND_BACK***
  - ***Property*** designa a propriedade do modelo de iluminação
    - ***GL_AMBIENT, GL_DIFFUSE, GL_SPECULAR, GL_EMISSION, GL_SHININESS***

Geometria

- Além das propriedades da luz e do material, a geometria do objeto também é importante
  - A posição dos vértices com relação ao olho e à fonte luminosa contribui no cálculo dos efeitos atmosféricos
  - A *normal* é fundamental
    - Não é calculada automaticamente
    - Precisa ser especificada com **`glNormal ()`**

Computando o Vetor Normal



- Triângulo
  - Dados três vértices,

$$\vec{n} = \text{normalizar}((A - B) \times (C - A))$$

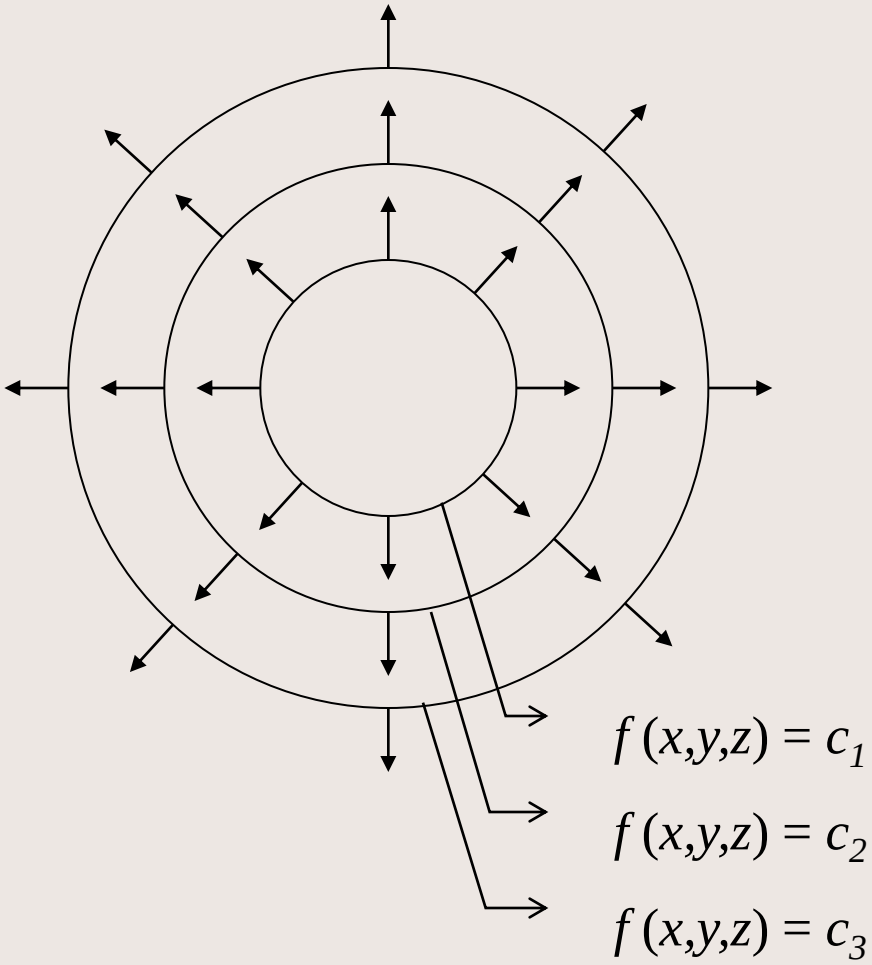
- Polígono planar
  - Uma opção é usar a fórmula do triângulo para quaisquer 3 vértices
    - Sujeito a erros (vetores pequenos ou quase colineares)
  - Outra opção é determinar a equação do plano
    - $ax + by + cz + d = 0$
    - Normal tem coordenadas (a, b, c)

Calculando o Vetor Normal de Superfícies Implícitas

- Normal dada pelo vetor gradiente

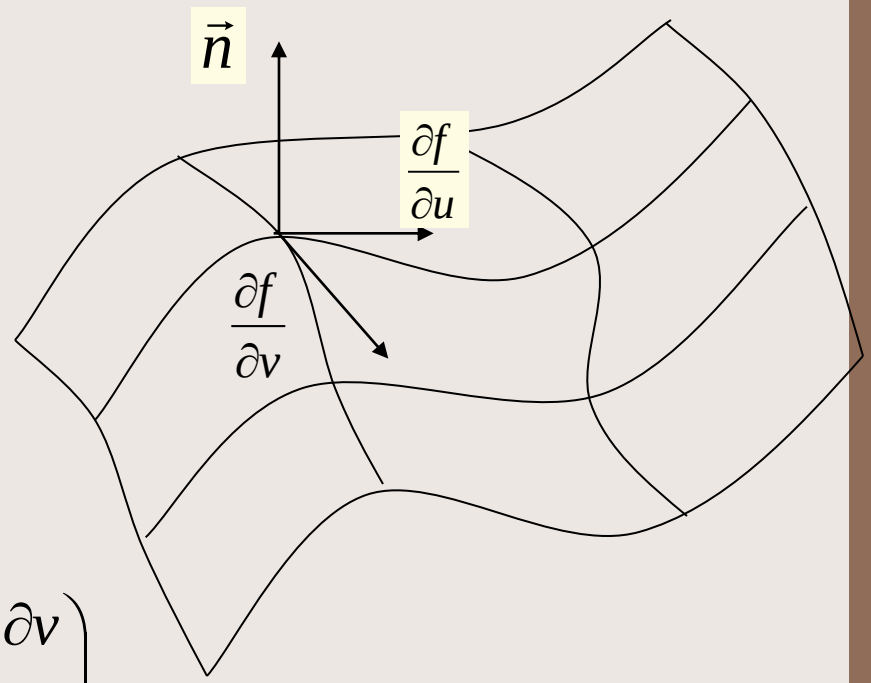
$$f(x,y,z) = 0$$

$$\vec{n} = \begin{pmatrix} \partial f / \partial x \\ \partial f / \partial y \\ \partial f / \partial z \end{pmatrix}$$



Calculando o Vetor Normal de Superfícies Paramétricas

- Normal dada pelo produto vetorial dos gradientes em relação aos parâmetros *u* e *v*



$$P = \begin{pmatrix} f_x(u,v) \\ f_y(u,v) \\ f_z(u,v) \end{pmatrix}$$

$$\vec{n} = \frac{\partial f}{\partial u} \times \frac{\partial f}{\partial v} = \begin{pmatrix} \partial f_x / \partial u \\ \partial f_y / \partial u \\ \partial f_z / \partial u \end{pmatrix} \times \begin{pmatrix} \partial f_x / \partial v \\ \partial f_y / \partial v \\ \partial f_z / \partial v \end{pmatrix}$$

Iluminação Ambiente

- Componente que modela como uma constante o efeito da reflexão de outros objetos do ambiente
- Depende dos coeficientes GL_AMBIENT tanto das fontes luminosas quanto dos materiais
- Tb. pode usar luminosidade ambiente não relacionada com fontes luminosas
  - **`glLightMaterialfv (GL_LIGHT_MODEL_AMBIENT, params)`**
- Contribuição dada por

$$A = I_a k_a$$

Atenuação

- Para fontes de luz posicionais ($w = 1$), pode definir um fator de atenuação que leva em conta a distância d entre a fonte de luz e o objeto sendo iluminado
- Coeficientes definidos pela função **`glLight ()`**
- Default: sem atenuação (c_0=1, c_1=c_2=0)

$$aten = \frac{1}{c_0 + c_1 d + c_2 d^2}$$

Cor final

- Atenuação é aplicada sobre as componentes difusa e especular
- A fórmula que calcula a cor de um vértice devida a uma fonte luminosa *i* é dada por:

$$C_i = A_i + aten\left(D_i + S_i\right)$$

- Portanto, no total, a cor é dada pela contribuição da iluminação ambiente (parcela não associada com fontes de luz) somada à luz emitida e às contribuições C_i

$$C = Amb + E + \sum A_i + aten\left(D_i + S_i\right)$$

Bibliografia

- **curso de CG da ACM SIGGRAPH) (de onde foram tiradas muitas das imagens):** www.education.siggraph.org/materials/HyperGraph/hypergraph.htm
- **ANGEL, E. Interactive Computer Graphics, Addison-Wesley, 3rd. Ed.**
- **GLASSNER, Andrew S. (Edited) -** ***An Introduction to Ray Tracing*****, Academic Press, 1989.**
- **BAKER, M. Pauline e HEARN, Donald -** ***Computer Graphics with OpenGL*****, Prentice Hall.**
- **FOLEY, James D., VAN DAM, Andries, FEINER, Steven e HUGHES, John -** ***Computer Graphics: Principles and Practice*** **- Addison-Wesley Ed., 1990.**