

Rendering

(de Modelos Poligonais)
Modelos de Iluminação
Métodos de Tonalização

2005-2017

Vários autores

SCC - ICMC - USP

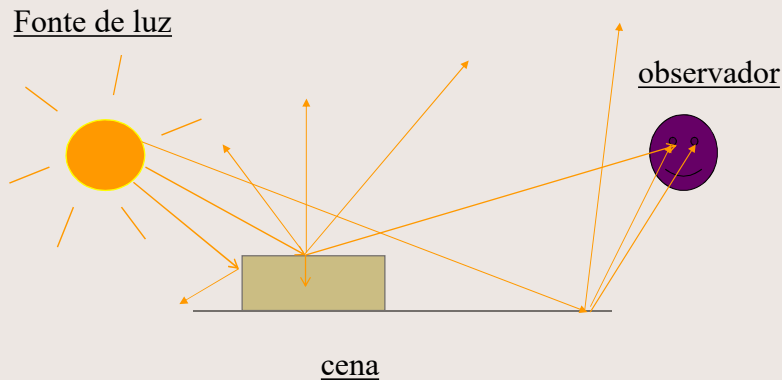
Rendering

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

Dados gráficos \Rightarrow Imagem

- Cena:
 - Modelo geométrico (geometria/topologia dos objetos)
 - Propriedades visuais das superfícies (materiais)
 - Condições de iluminação ambiente (fontes de luz)
 - Parâmetros de *viewing*: posição do observador e outros...

Processo Físico de Geração de uma Imagem



3

Síntese de Imagens 3D

- Tenta “simular”(muitas vezes, de forma bastante aproximada) 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 empíricos x *physically-based models*.

4

Foto-realismo em CG

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

5

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 o ‘interior’ dos objetos
 - água, névoa, nuvens, fogo, modelos capturados por dispositivos de imageamento, etc.

6

Wireframe x shaded

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

7

Modelos de Iluminação

- tentam “simular” como a luz é refletida pelos objetos, produzindo o que percebemos como cor
 - a luz que sai de uma **fonte emissora** e é refletida pelas múltiplas superfícies dos objetos, eventualmente atingindo o olho do observador
 - modelos locais (1a. ordem): operam como se a iluminação de uma superfície fosse independente das outras superfícies na cena
 - modelos globais: incluem a contribuição da luz refletida por outras superfícies da cena

8

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)

9

Modelos de Iluminação

- clássico: *Phong* (padrão, simples, rápido, totalmente empírico)
- modelos físicos: para produzir resultados mais realistas usam a teoria que modela 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.

10

Modelo de Iluminação: Exemplo



11

Processo de *Rendering*

- Um modelo de iluminação é integrado a um algoritmo 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 fotorealismo desejado, o *hardware* disponível.
 - abordagens clássicas: *scanline*, *ray tracing*, radiosidade.

12

Algoritmos de *Rendering*: Classificação

- operam na **ordem da imagem** (resolve o problema *pixel a pixel*), *ou*
- na **ordem dos objetos** (resolve o problema renderizando cada objeto na cena)
- Adotam **modelos de iluminação locais**: apenas a contribuição direta da fonte de luz, *ou*
- **modelos globais** (incorporam a contribuição devida à interação entre os objetos: múltiplas reflexões, transparência, sombras, ...)

13

Algoritmos Clássicos

- *scanline*: “padrão” em sistemas gráficos
 - opera sobre malhas poligonais
 - usa um modelo de iluminação local simples. Efeitos adicionais podem ser incorporados por várias técnicas *ad hoc*, como cálculo de sombras e mapeamento de textura.
 - opera na ordem dos objetos: renderiza cada polígono, em ordem arbitrária.
 - Usa o algoritmo *z-buffer* para apresentar os objetos na ordem correta

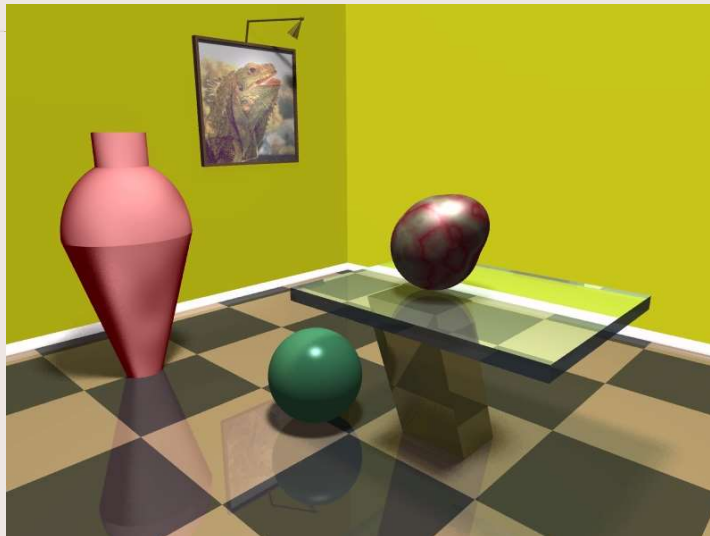
14

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
 - Incorpora 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

15

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



16

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

17

Radiosidade: Exemplo



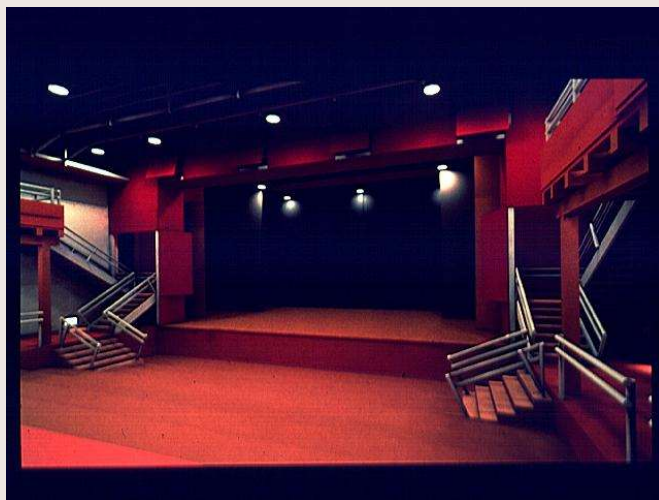
18

Radiosidade: Exemplo



19

Radiosidade: Exemplo



20

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**.

21

Fontes de Luz: Geometria

- **Puntuais**
 - 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. Aproximação para um *spot*.
- **Distribuídas**: a fonte tem área e uma geometria própria (lâmpadas fluorescentes)

22

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)

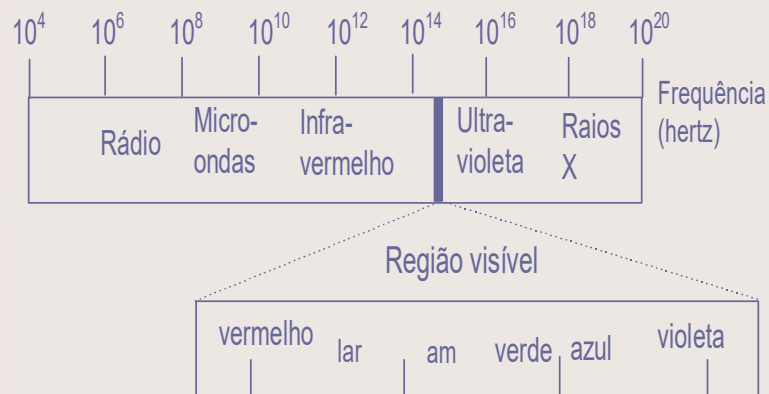
23

Cor

- Energia luminosa, ou onda eletromagnética:
 - banda visível do espectro eletromagnético: cada frequência (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

24

Cor - O Espectro Visível



25

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 bastante 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.

26

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: 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 exhibe os dois tipos de reflexão

27

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

28

Modelo de Iluminação de Phong

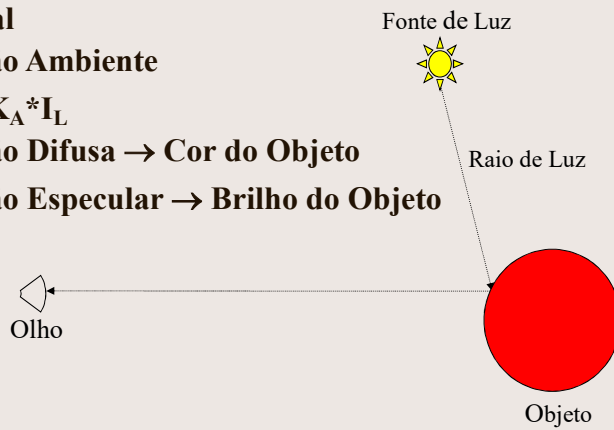
Modelo Local

Iluminação Ambiente

$$I_A = K_A * I_L$$

Iluminação Difusa → Cor do Objeto

Iluminação Especular → Brilho do Objeto

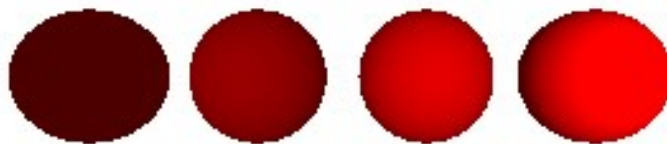


29

Modelo de Iluminação

Modelo Local

Iluminação Ambiente



30

Modelo de Iluminação e Métodos de *Rendering*

The ambient lighting in the upper-right image is approximated by a constant value. This is typical of most scanline algorithms. The middle and lower-left images were rendered with a ray tracing global illumination algorithm.



The middle image was rendered with no ambient light calculations. The lower-left image was rendered with several levels of diffuse re-reflection to give a better approximation of the ambient light in this scene.

31

Modelo de Phong: Reflexão Difusa

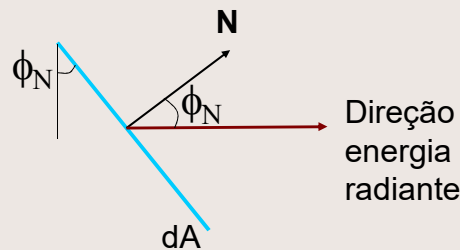
- A superfície reflete a luz incidente igualmente em todas as direções
 - Uma faixa de comprimentos de onda da luz incidente é absorvida, outra faixa é refletida: responsável pela ‘cor percebida’ do objeto.
 - reflexão independente da direção de observação
 - quantidade de luz refletida é controlada 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

32

Modelo de Phong: Reflexão Difusa

- Lei dos Cossenos de Lambert:

- a energia radiante refletida por uma pequena área de superfície dA , em qualquer direção ϕ (relativa à normal à superfície) é proporcional a $\cos\phi$.



33

Modelo de Phong: Reflexão Difusa

- a intensidade da luz refletida depende da energia radiante por área projetada perpendicular à direção ϕ_N , dada por $dA \cdot \cos\phi_N$.
- Apesar do espalhamento da luz ser igual em todas as direções (superfície refletora idealmente difusa), 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 $\cos\theta = 0$ a superfície é totalmente iluminada, e a iluminação percebida diminui a 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!)

35

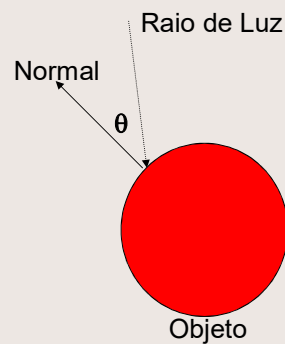
Modelo de Phong: Reflexão Difusa

Modelo Local

Iluminação Difusa (*Lei dos Cossenos de Lambert*)

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

$$0 \leq \theta \leq 90^\circ$$



36

Modelo de Phong: Reflexão Difusa

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

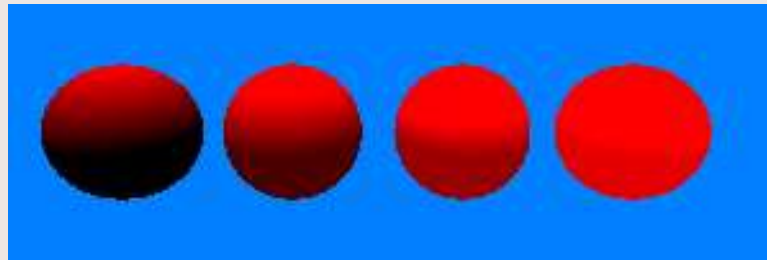
$$I_l = I_a K_a + K_d I_l (\mathbf{N} \cdot \mathbf{L})$$

38

Modelo de Phong: Reflexão Difusa

Modelo Local

Iluminação Difusa + Ambiente



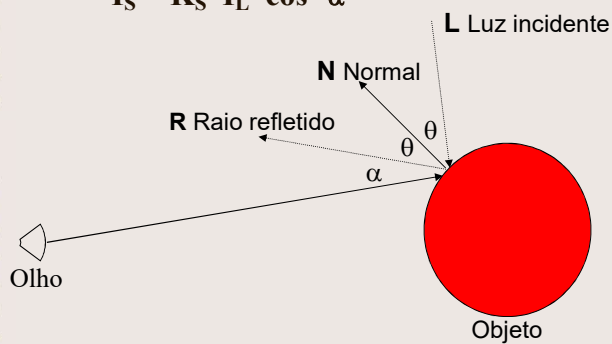
39

Modelo de Phong: Reflexão Especular

Modelo Local

Iluminação Especular (por Phong Bui Tuong)

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



40

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, \mathbf{R} , e a direção de observação, \mathbf{S} .
 - Intensidade da reflexão especular depende de fatores:
 - propriedades do material. 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.

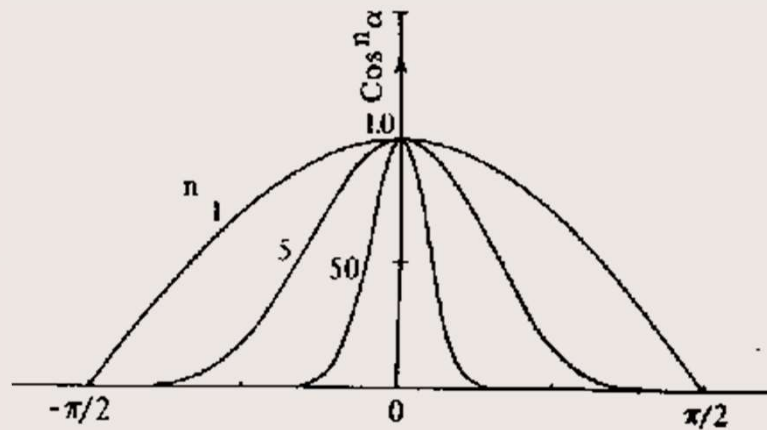
41

Modelo de Phong: Reflexão Especular

- O termo especular de Phong é descrito por
$$I_{\text{specular}} = W(\theta, \lambda) I_i \cos^n \alpha$$
- Para materiais opacos, a reflexão especular é aproximadamente constante para todos os ângulos de incidência \Rightarrow Phong aproximou a função por uma constante: $I_s = K_s I_i (\mathbf{S} \cdot \mathbf{R})^n$
- o vetor \mathbf{R} pode ser calculado a partir de \mathbf{L} e \mathbf{N}
- múltiplas fontes de luz: soma as contribuições de cada uma

42

Modelo de Phong: Reflexão Especular



45

Modelo de Phong: Reflexão Especular

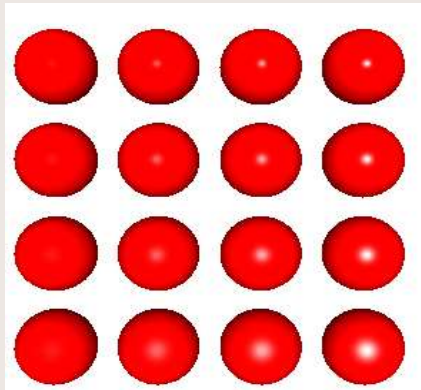
- Valor de n determinado pelo tipo de superfície:
n grande (> 100) para superfícies mais polidas,
n pequeno (até 1) para superfícies mais opacas.

46

Modelo de Phong: Reflexão Especular

Modelo Local

Iluminação Especular



47

Modelo de Phong

- Ver também:
<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)

48

Example - Ambient Reflection



49

Example - Ambient and Diffuse



50

Ambient, Diffuse and Specular



51

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

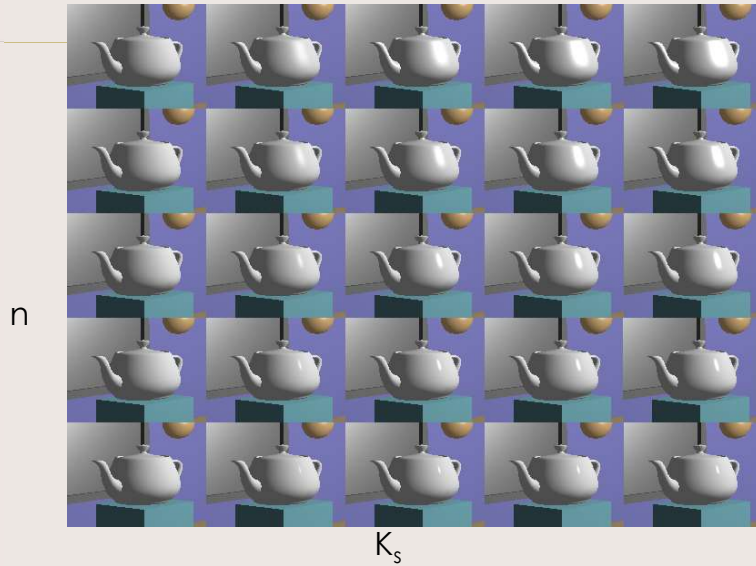


K_s

K_d

52

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

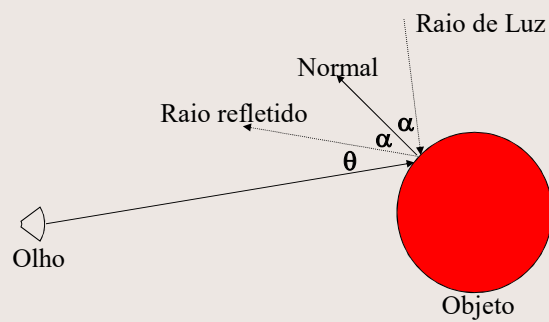


53

Modelo de Phong completo

Modelo Local Completo

$$\mathbf{I} = \mathbf{I}_A + \mathbf{I}_D + \mathbf{I}_S$$



54

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

55

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

56

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.
 - em 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)

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

57

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.

58

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 K_s como uma constante independente da cor \Rightarrow reflexões especulares da mesma cor da luz incidente (em geral, branca) (aparência plástica).

59

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.

60

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

61

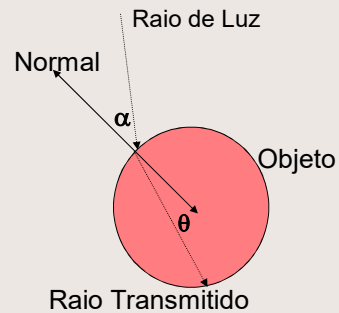
Melhorias no modelo

- 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

62

Transparência

Lei de Snell
 $\eta_{\alpha} * \sin \alpha = \eta_{\theta} * \sin \theta$



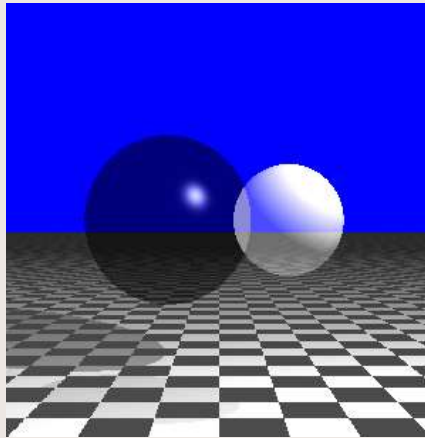
63

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...

64

Transparência por interpolação: exemplo



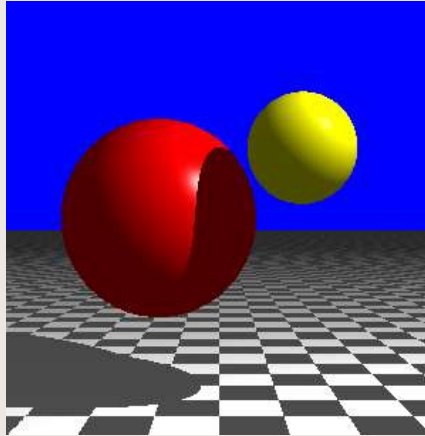
65

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

66

Sombras: exemplo



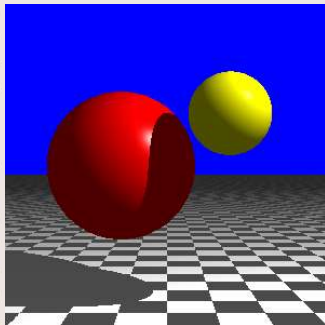
67

Sombra

Modelo Global

Sombras

Deteccão de Pontos Não Iluminados Diretamente



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

68

Textura

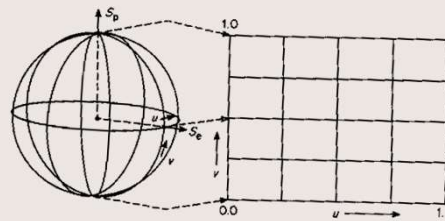
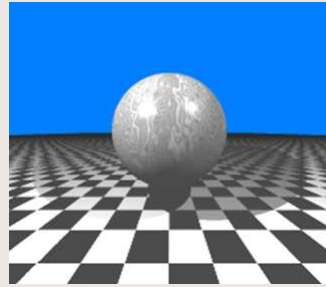
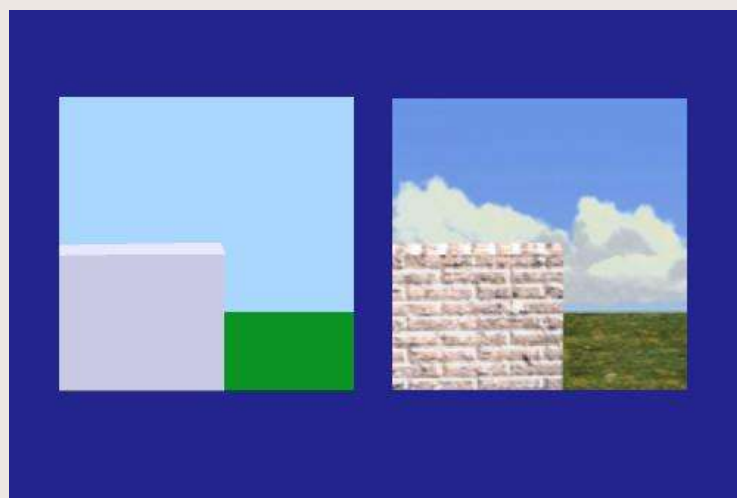


Fig. 5. Inverse mapping for a sphere.



69

Textura: exemplo



70

Textura: exemplo



71

Modelo de Iluminação Global

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

72

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

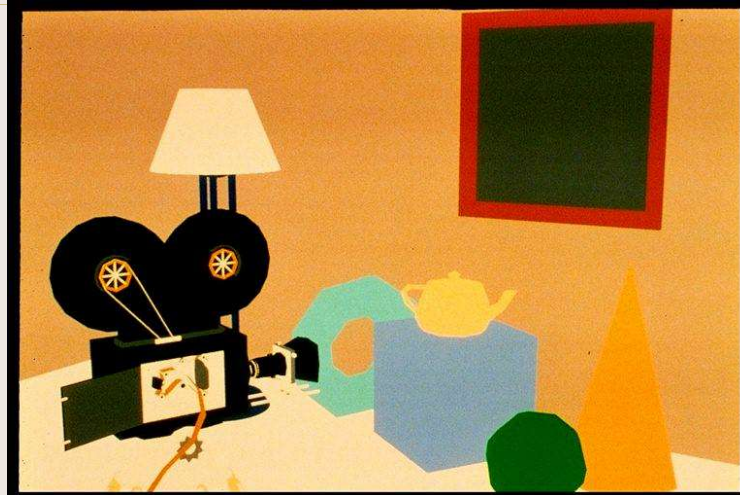
73

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*.

74

Constant Shading



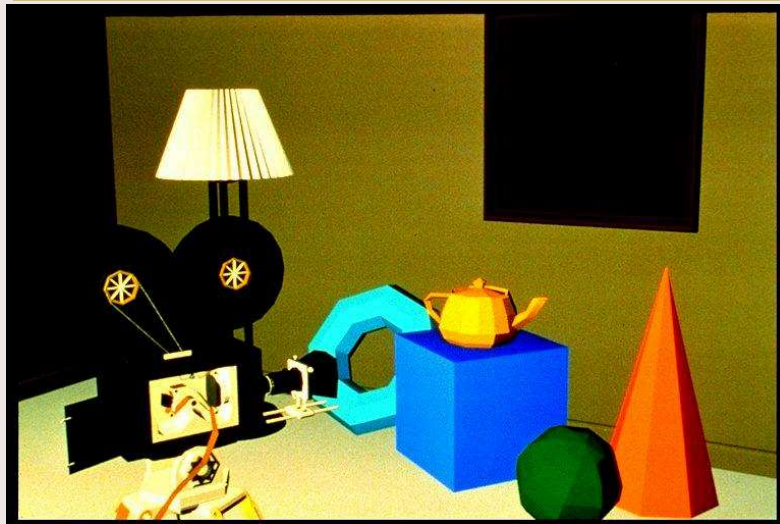
75

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)`

76

Flat shading



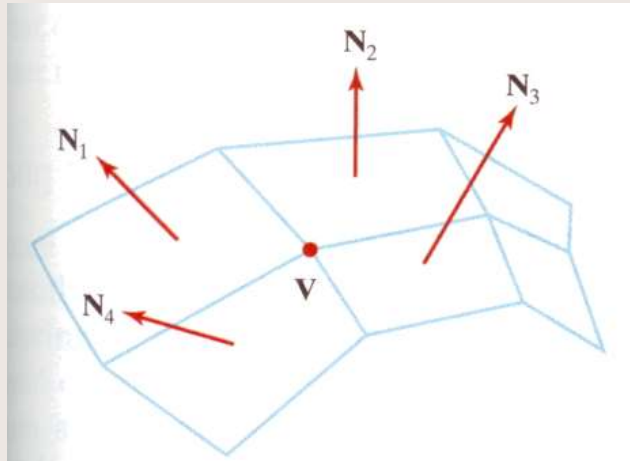
Gouraud Shading

- 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

78

Gouraud Shading

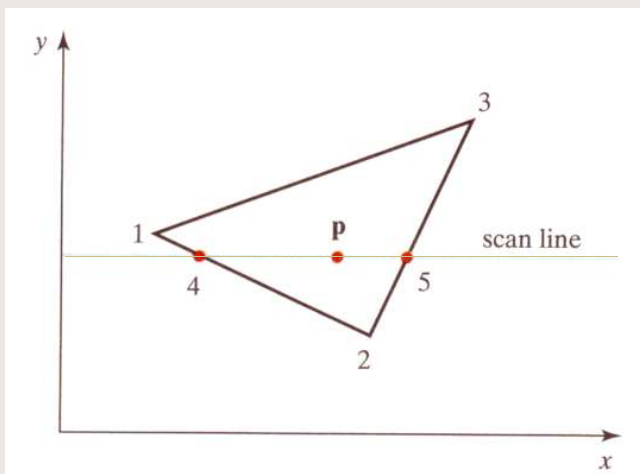
Fonte: Hearn & Baker



79

Gouraud Shading

Fonte: Hearn & Baker



80

Gouraud Shading: Algoritmo

1. determina a normal \mathbf{N} em cada vértice do polígono
2. usa \mathbf{N} e \mathbf{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

81

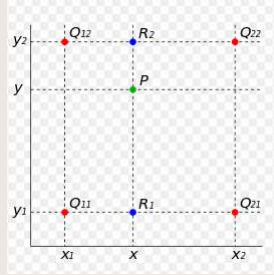
Gouraud Shading

- Como calcular \mathbf{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

82

Gouraud Shading

- e a interpolação bi-linear? Fonte: wikipedia



$$f(x, y_1) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{11}) + \frac{x - x_1}{x_2 - x_1} f(Q_{21})$$
$$f(x, y_2) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{12}) + \frac{x - x_1}{x_2 - x_1} f(Q_{22})$$

$$f(x, y) \approx \frac{y_2 - y}{y_2 - y_1} f(x, y_1) + \frac{y - y_1}{y_2 - y_1} f(x, y_2)$$

83

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

84

*Gouraud Shading (sem highlight
especular)*



*Gouraud Shading (com highlight
especular)*



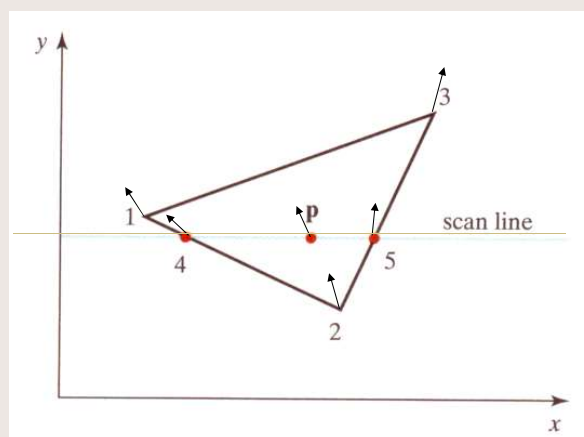
Phong Shading

- 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

87

Phong Shading

Fonte: Hearn & Baker



88

Phong Shading



89

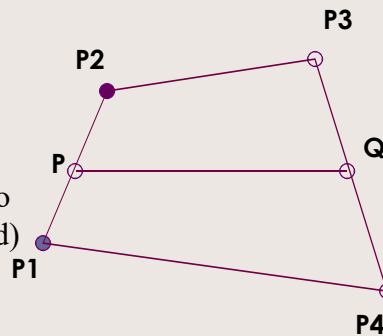
Observação

- 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

90

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



91

Recursos

- Olhar arquivo: SlidesExemploOpenGL.zip para instruções e exemplos OpenGL para:
 - Transformações
 - Iluminação
 - Animação e Texturas
 - Modelos Poligonais

92

Bibliografia

- curso de CG da ACM SIGGRAPH) (de onde foram tiradas muitas das imagens):
<http://www.siggraph.org/education/materials/HyperGraph/hypergraph.htm>
- GLASSNER, Andrew S. (Edited) - *An Introduction to Ray Tracing*, Academic Press, 1989.
- BAKER, M. Pauline e HEARN, Donald - *Computer Graphics*, Prentice Hall Ed, 1997.
- FOLEY, James D., VAN DAM, Andries, FEINER, Steven e HUGHES, John - *Computer Graphics: Principles and Practice* - Addison-Wesley Ed., 1990.