

# Iluminação

André Tavares da Silva

[andre.silva@udesc.br](mailto:andre.silva@udesc.br)

Capítulo 14 de “Foley”

Capítulo 7 de Azevedo e Conci

# Roteiro

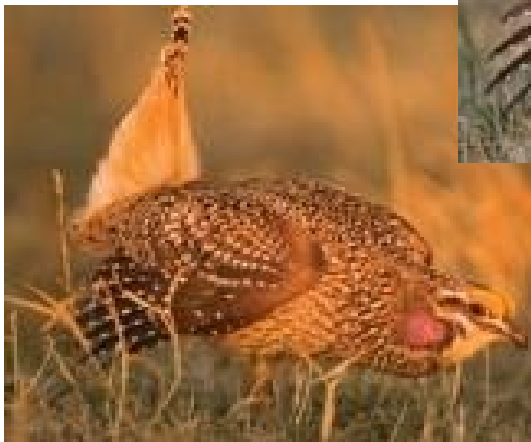
- Introdução
- Modelos de Iluminação
  - Luz Ambiente; Reflexão Difusa; Atenuação Atmosférica; Reflexão Especular; Modelo de Iluminação de Phong; Múltiplas Fontes de Luz
- Modelos de Sombreamento para Polígonos
  - Flat Shading; Interpolated Shading; Polygon-Mesh Shading; Gouraud Shading; Phong Shading
- Problemas com o Sombreamento por Interpolação
- Iluminação em OpenGL

# Introdução

- Objetivo do modelo de iluminação é determinar uma cor numa superfície, considerando suas propriedades (cor, textura, material, ...) e fatores externos, como a luz que incide sobre a superfície, num determinado ponto da mesma.
- Modelo de **iluminação** x modelo de **sombreamento** (*shading*)
  - Modelo de sombreamento determina quando o modelo de iluminação é aplicado e quais argumentos ele recebe.
- Diversos modelos são propostos para iluminação. Veremos modelos mais simples e que resolvem o problema de forma eficiente e com menor custo computacional.

# Modelo de Iluminação

- Estes modelos determinam como a luz interfere (como é recebida) numa superfície.
- Isto implica em alterações na visualização do objeto, ou seja, em como ele será renderizado.



Dependendo da luz que incide sobre um objeto, ele muda completamente a sua "renderização".

# Modelo de Iluminação

Modelo de Iluminação de Phong:

$$I_{\lambda} = I_{a\lambda} k_a O_{\lambda} + \sum_{i=1}^m f_{att_i} I_{p\lambda_i} \left[ k_d O_{d\lambda} (N \cdot L_i) + k_s O_{s\lambda} (R_i \cdot V)^n \right]$$

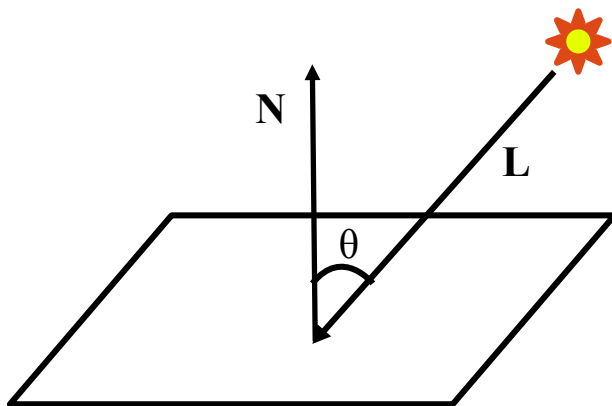
Modelo de Iluminação de David Immel et al. / James Kajiya:

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

(LTE)

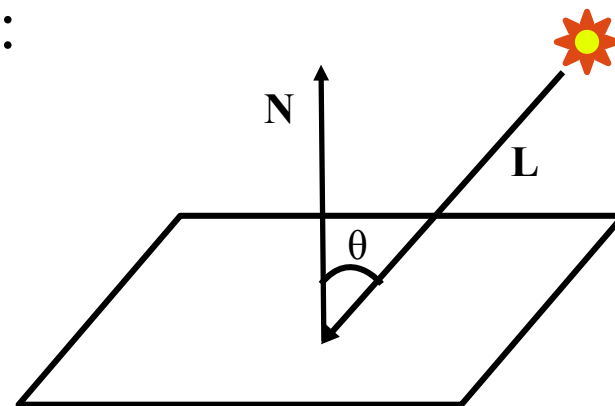
# Reflexão Difusa (Luz Difusa)

- Também conhecida como Lambert's Cosine Law (ou Reflexão Lambertiana)
- Em óptica, a **lei de Lambert** diz que a intensidade luminosa de uma superfície refletora difusa ideal é diretamente proporcional ao cosseno do ângulo  $\theta$  entre a direção da luz incidente **L** e a normal da superfície **N**.
- Com isso, a iluminação das partes do objeto varia de acordo com a distância e a direção em relação ao ponto de luz.



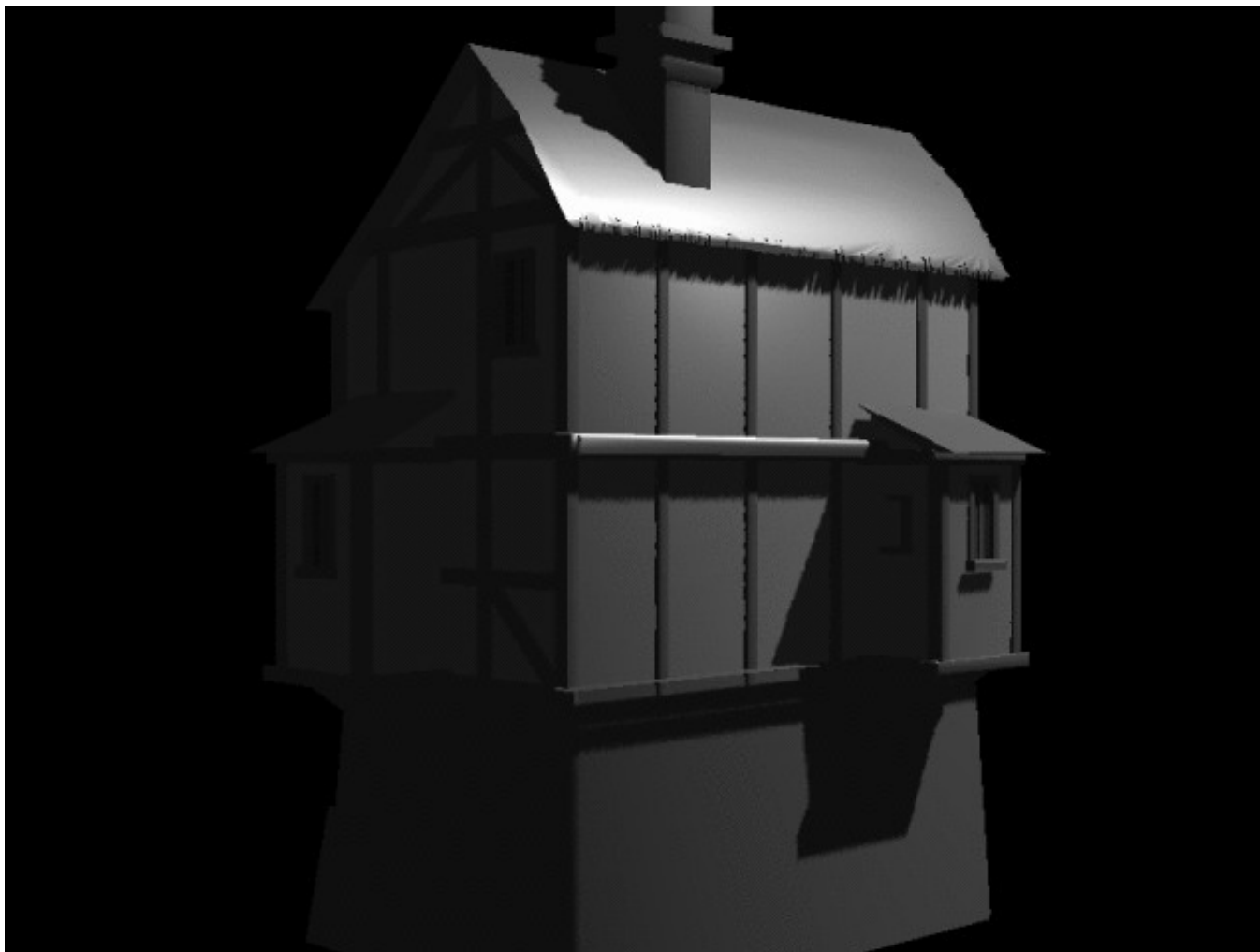
# Reflexão Difusa (Luz Difusa)

- Sendo assim, reflexão difusa independe da direção do observador sendo proporcional somente ao cosseno de  $\theta$ .
- $I = I_p k_d \cos \theta$ 
  - $I_p \rightarrow$  Intensidade da fonte de luz
  - $k_d \rightarrow$  coeficiente de reflexão difusa (constante [0..1])
- Considerando vetores normalizados:
  - $I = I_p k_d (\mathbf{N} \cdot \mathbf{L})$



# Luz Ambiente

- Somente reflexão difusa:





# Luz Ambiente

- Com luz ambiente:



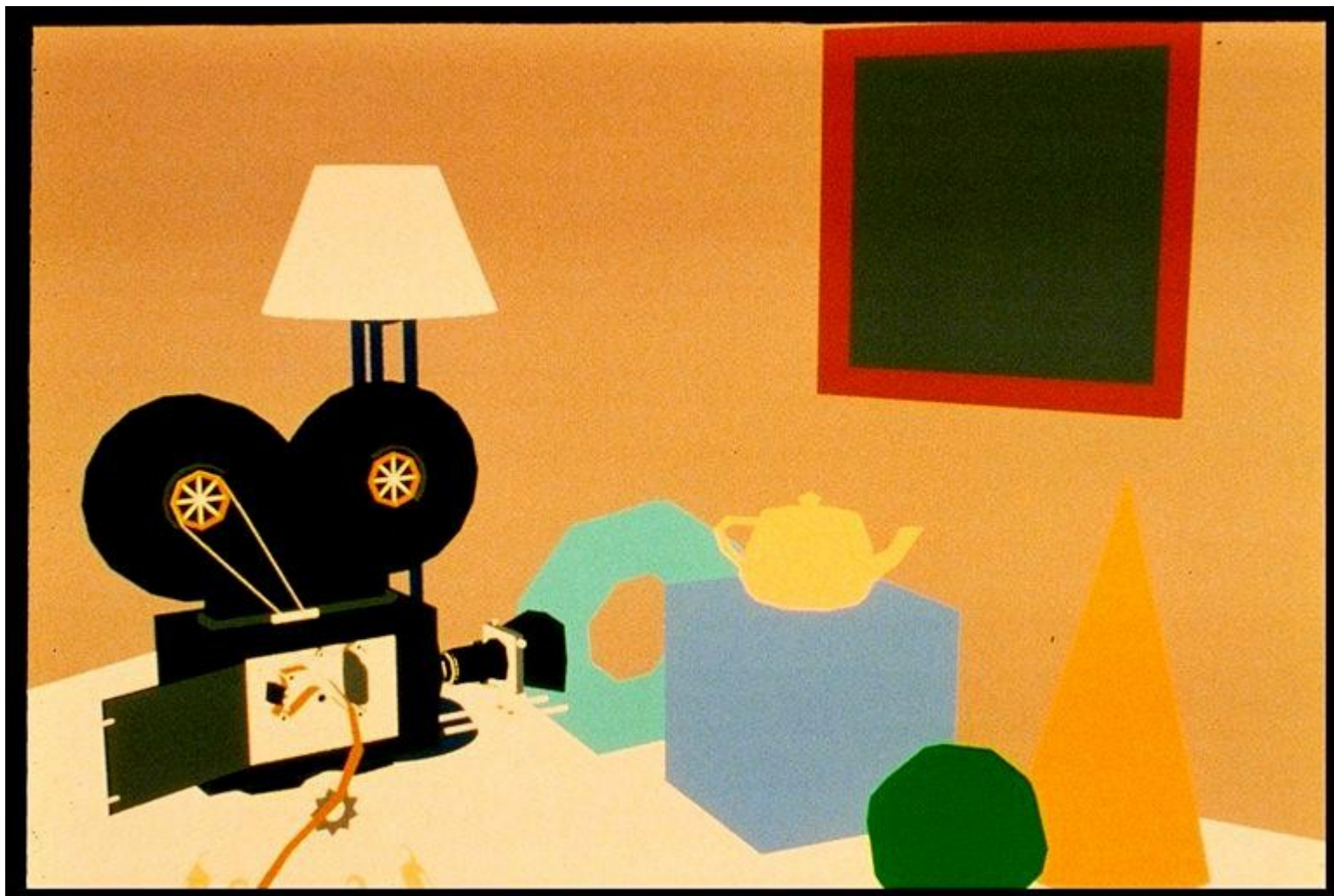
# Luz Ambiente

- Agora, ao invés de considerar apenas a reflexão Lambertiana, consideremos a existência de uma fonte de luz sem direção específica, resultante das reflexões entre as muitas superfícies presentes no ambiente: conhecida como luz ambiente.
- Num modelo de iluminação ambiente, fatores externos de iluminação são ignorados e um objeto é desenhado apenas com a cor que é intrínseca a ele.

# Luz Ambiente

- Equação:
- $I = I_a k_a$ 
  - $I \rightarrow$  é a intensidade resultante
  - $I_a \rightarrow$  é a intensidade da luz ambiente, constante para todos os objetos da cena
  - $k_a \rightarrow$  é o coeficiente de reflexão ambiente [0..1].  
É o quanto da luz ambiente é refletida pelo objeto. É a propriedade do material do objeto.

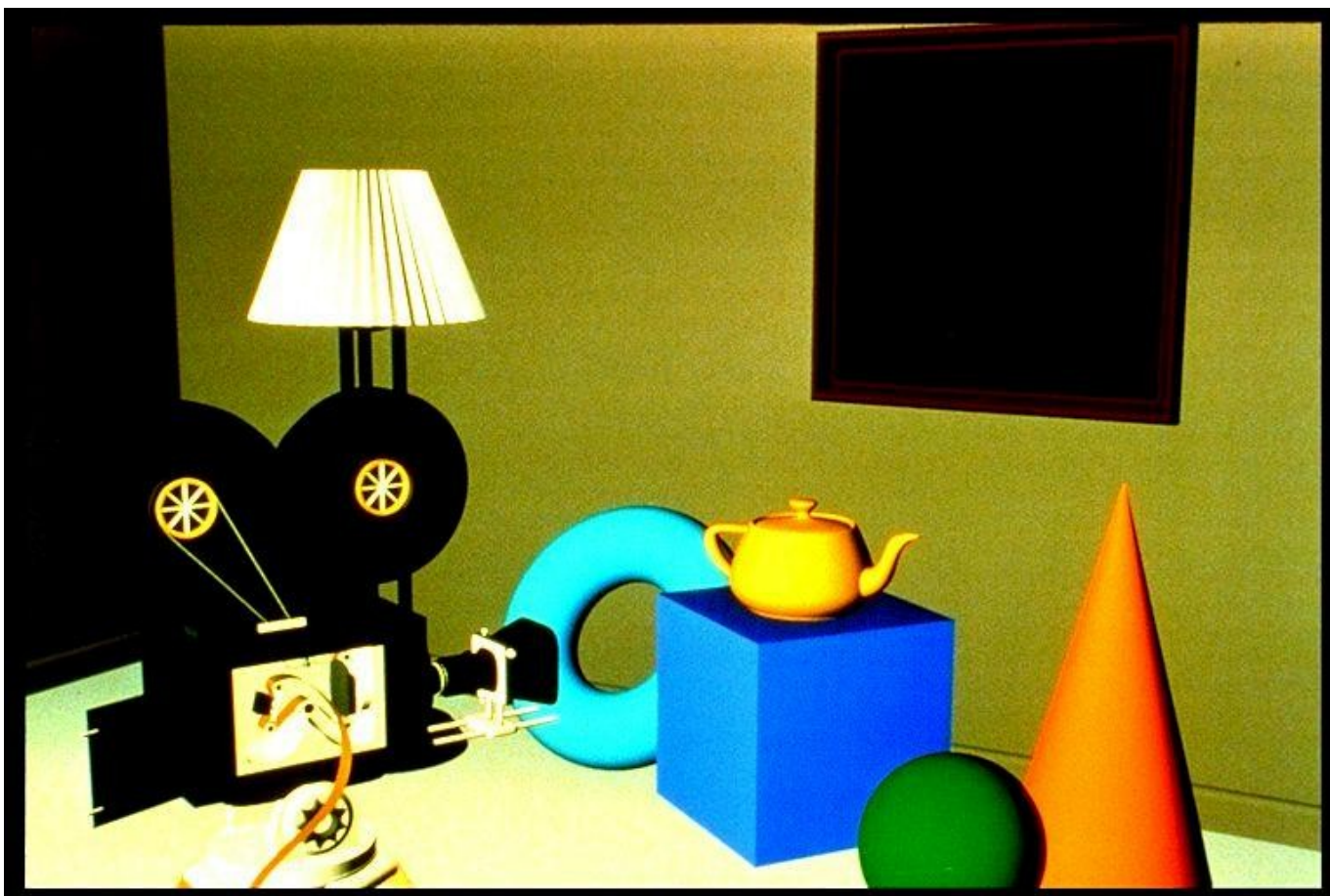
# Luz Ambiente



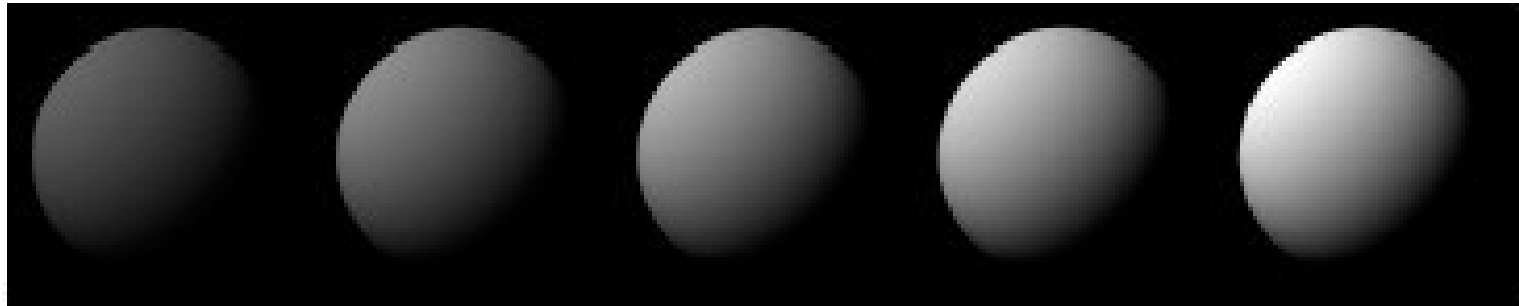
# Ambiente + Difusa

- Combinando os componentes de luz ambiente e difusa:

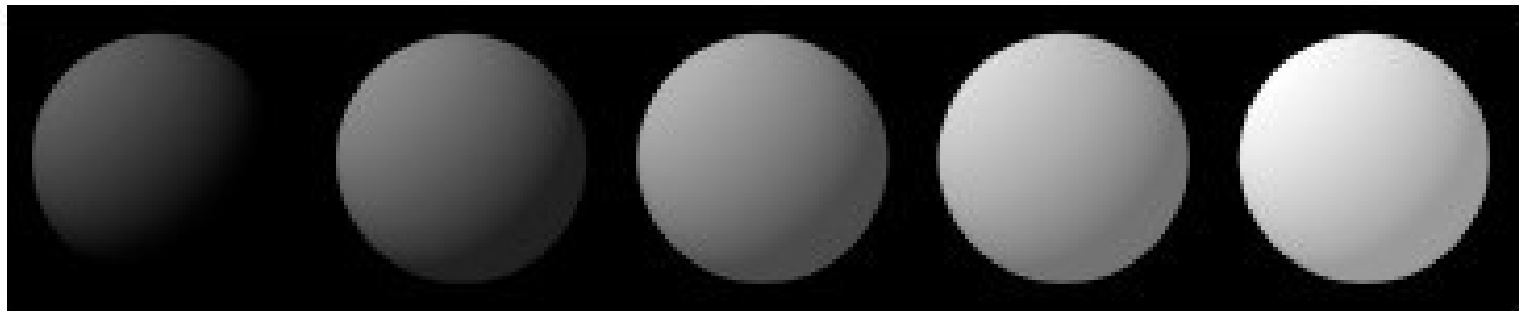
$$- I = I_a k_a + I_p kd (N \cdot L)$$



# Ambiente + Difusa



$K_d = 0.4, 0.55, 0.7, 0.85, 1.0$



$I_a = I_p = 1.0, K_d = 0.4, K_a = 0.0, .015, 0.3, 0.45, 0.6$



# Fator de Atenuação

- Caso projetarmos duas superfícies paralelas de material idêntico, iluminadas a partir do observador, equação atual não distinguirá onde termina uma superfície e onde começa outra, não importando a distância.
- Para simular esta diferença de distância, criou-se um fator de atenuação da fonte de luz ( $f_{att}$ ).

$$-I = I_a k_a + f_{att} I_p k_d (N \cdot L)$$

# Cor do objeto e fontes de luz

- Considerando as componentes de cor, a fórmula deve multiplicar cada termo pela cor do objeto, assim como é percebido pelo ambiente e pela cor da difusa. Portanto:

$$- I_{\lambda} = I_{a\lambda} k_{a\lambda} O_{\lambda} + f_{att} I_{p\lambda} k_{d\lambda} O_{d\lambda} (N \cdot L)$$

– Onde:

- $\lambda$  representa um canal RGB por vez
- $I_{\lambda}$  é a intensidade resultante para cada canal RGB
- $I_{a\lambda}$  é a intensidade da luz ambiente para cada canal RGB
- $I_{p\lambda}$  é a intensidade resultante para cada canal RGB
- $O_{\lambda}$  é cor do objeto para cada canal RGB
- $O_{d\lambda}$  é cor da luz difusa para cada canal RGB



# Atenuação Atmosférica

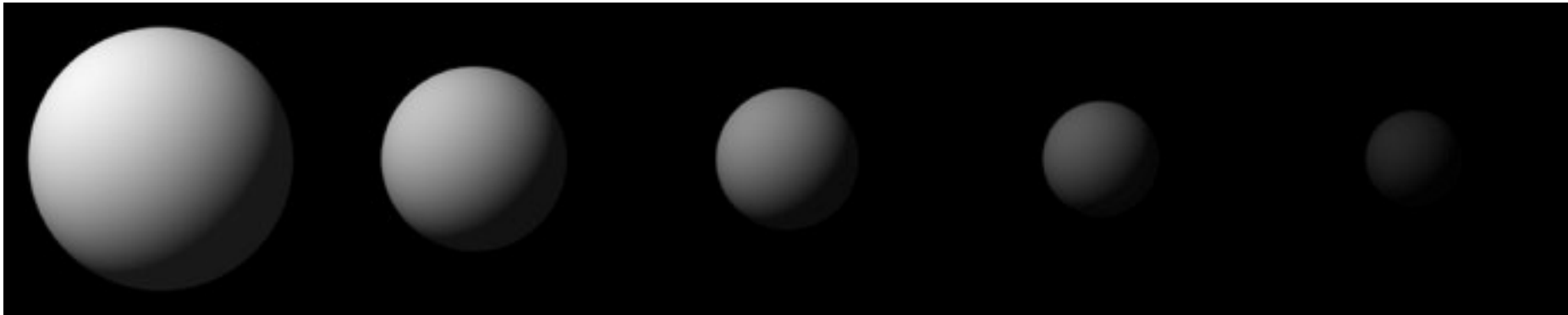
- Simulação de atenuações atmosféricas ou condições de visibilidade do ambiente (*depth cueing*):
  - Névoa, por exemplo.
  - Situações aonde os objetos possuem brilho diferente conforme a profundidade.
  - Similar a idéia de atenuação da fonte luz.
  - Objetivo é modificar uma intensidade já calculada
  - A modificação será determinada por fatores de escala ( $s_f$  e  $s_b$ ) que indicarão o combinação da intensidade com uma cor sugerida para o efeito da atenuação

# Atenuação Atmosférica



# Atenuação Atmosférica

- $I'_{\lambda} = S_{\theta} I_{\lambda} + (1 - S_{\theta}) I_{dc\lambda}$
- $S_{\theta}$  é calculado a partir do Z: 
$$S_{\theta} = S_b + \frac{(Z_0 - Z_b) - (S_f - S_b)}{(Z_f - Z_b)}$$
- $I_{dc\lambda}$  é a cor do fator de atenuação.



Distância da luz é constante,

$$I_a = I_p = 1.0;$$

$$k_a = 0.1; k_d = 0.9;$$

$$z_f = 1.0; z_b = 0.0; s_f = 1.0; s_b = 0.1; \text{raio} = 0.09;$$

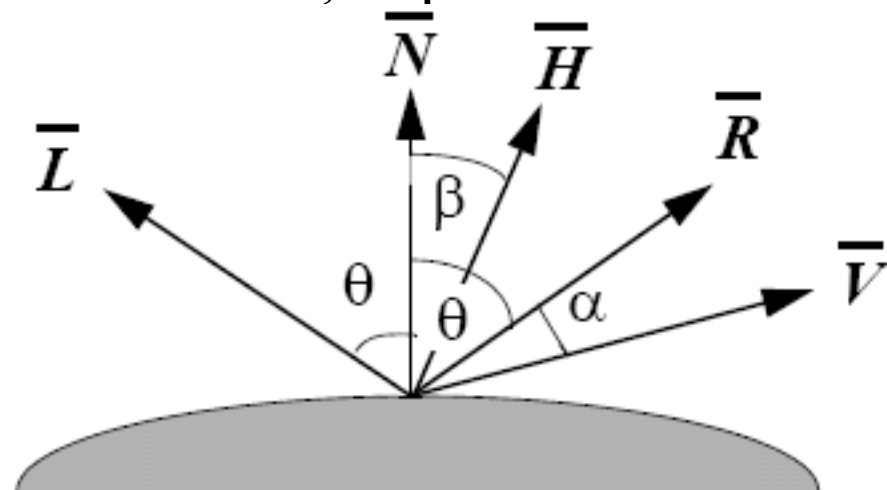
$$z = 1.0, 0.77, 0.55, 0.32, 0.09$$

# Reflexão Especular

- Círculos brilhantes (*highlight*) que aparecem em superfícies “brilhosas” (superfície especular).
- Para observar: ilumine uma bola de bilhar com uma luz branca e aparecerá um brilho causado pela reflexão especular.
- Neste ponto a cor será branca e não a cor original da bola.
- Movimentando a cabeça nota-se que o brilho se move.
- Superfícies refletem luz de forma diferente em diferentes direções.
- Portanto, o observador recebe a **cor da reflexão da luz** no objeto.

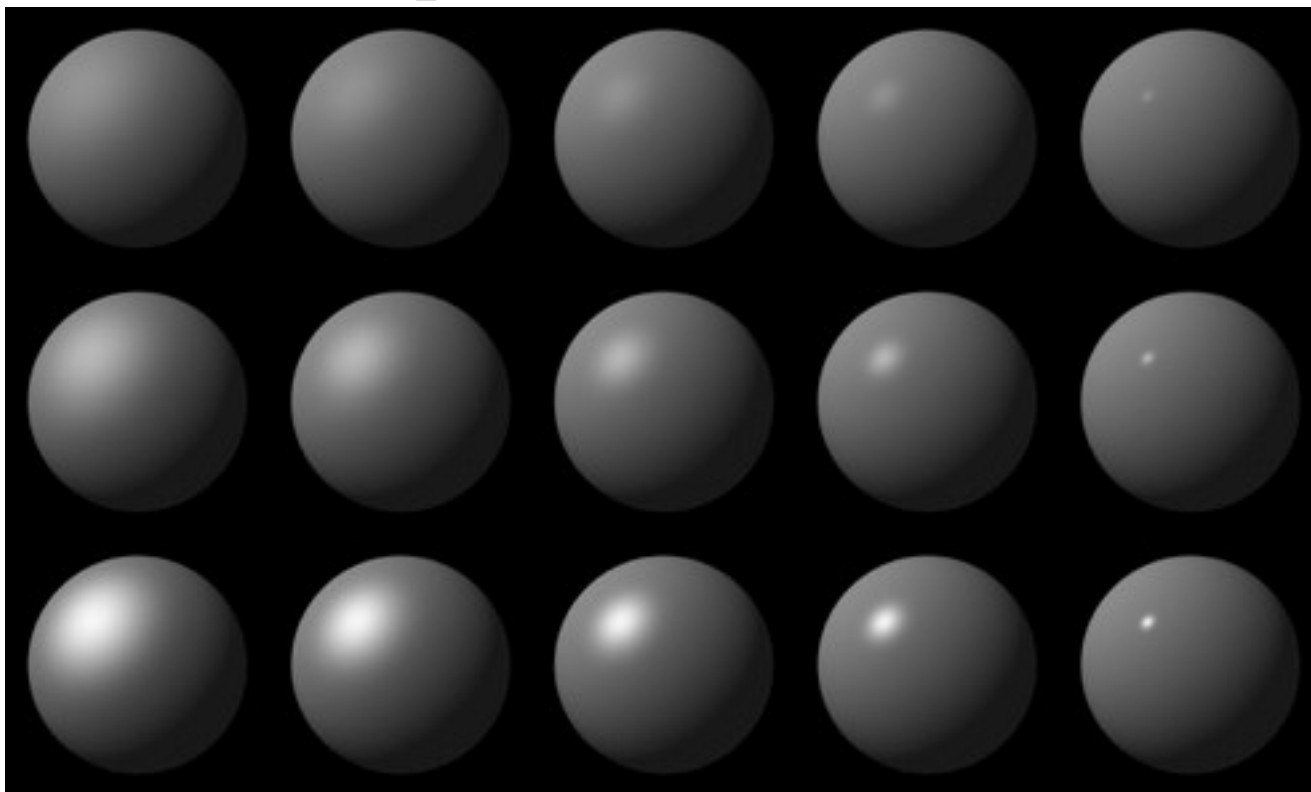
# Reflexão Especular

- Phong desenvolveu em 1975, um modelo para a parcela de reflexão especular que assumia que:
  - A máxima reflexão especular ocorre quando  $\alpha$  é zero e cai rapidamente quando  $\alpha$  cresce;
  - Essa mudança é representada por  $\cos^n \alpha$ , onde  $n$  é o expoente de reflexão especular.
    - Valores de  $n$  variam de 1 até centenas, dependendo do material simulado.



# Reflexão Especular

- Efeito da especular:



$I_a = I_p = 1.0$ ,  $K_a = 0.1$ ,  $K_d = 0.45$

$n = 3.0, 5.0, 10.0, 27.0, 200.0$

$K_s = 0.1, 0.25, 0.5$

# Reflexão Especular

- Acrescentando a especular na equação:

$$I_{\lambda} = I_{a\lambda} k_a O_{\lambda} + f_{att} I_{p\lambda} \left[ k_d O_{d\lambda} (N \cdot L) + k_s O_{s\lambda} (R \cdot V)^n \right]$$

– Onde:

- $k_s$  é o coeficiente de contribuição da especular
- $O_{s\lambda}$  é cor da luz especular para cada canal RGB
- $R$  é o vetor dado pelo raio de luz refletido na superfície
- $V$  é o vetor definido pela posição do observador e superfície

# Múltiplas Fontes de Luz

- Acrescentando mais luzes na equação:

$$I_{\lambda} = I_{a\lambda} k_a O_{\lambda} + \sum_{i=1}^m f_{att_i} I_{p\lambda_i} \left[ k_d O_{d\lambda} (N \cdot L_i) + k_s O_{s\lambda} (R_i \cdot V)^n \right]$$



# Modelos de Sombreamento (*Shading Models*)

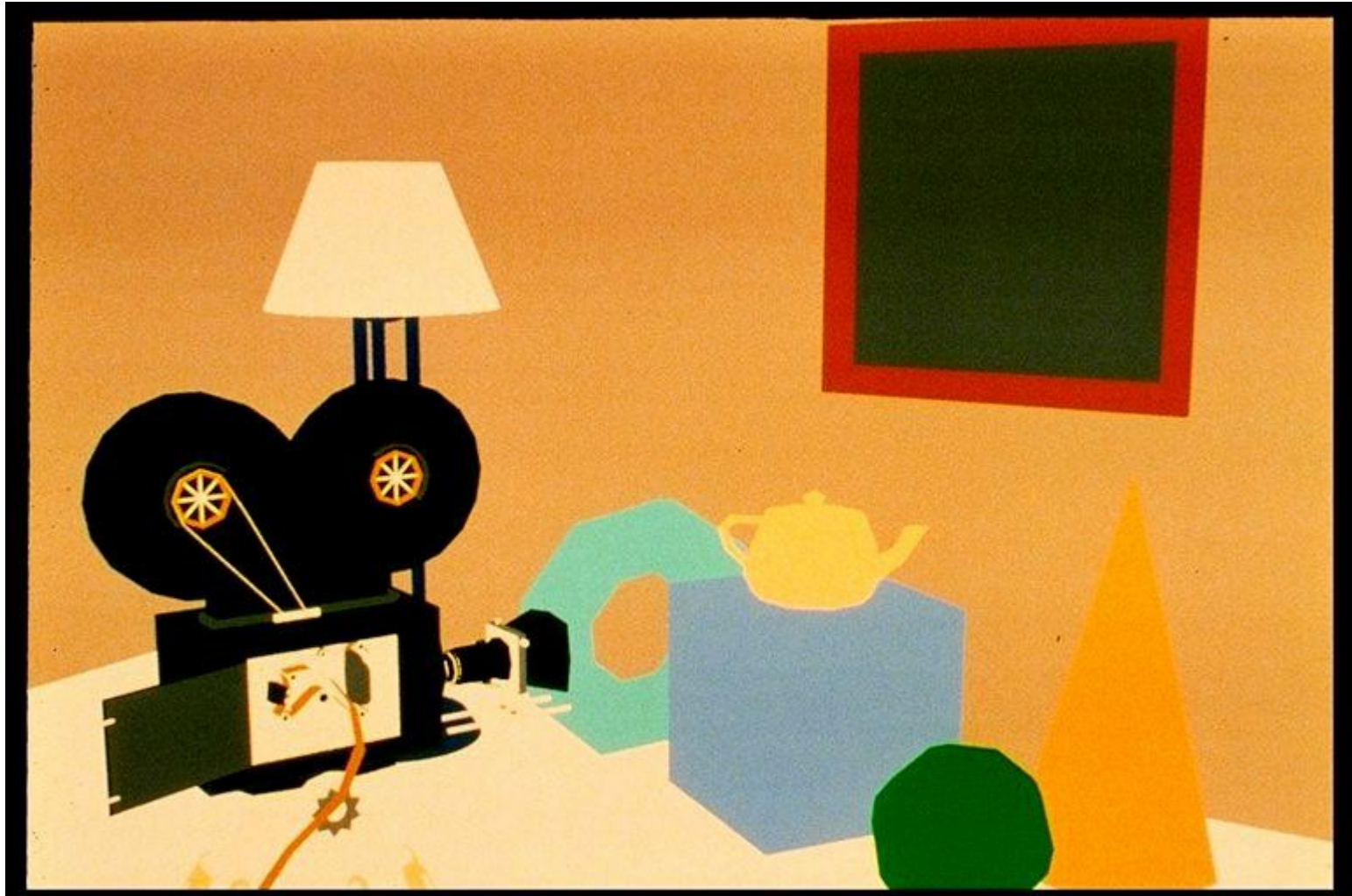
# Modelos de sombreamento

- Como preencher um polígono considerando aspectos de iluminação?
  - Solução 1: uma idéia é calcular a normal em cada ponto visível da superfície e aplicar um modelo de iluminação para cada ponto (força bruta).
    - Problema: Custo computacional é muito elevado.
  - Solução 2: modelos de sombreamento.
    - Executam o preenchimento do polígono de forma eficaz e eficiente.

# Modelos de sombreamento

- Constant shading
- Flat shading
- Interpolated shading
- Polygon mesh shading
- Gouraud Shading
- Phong Shading

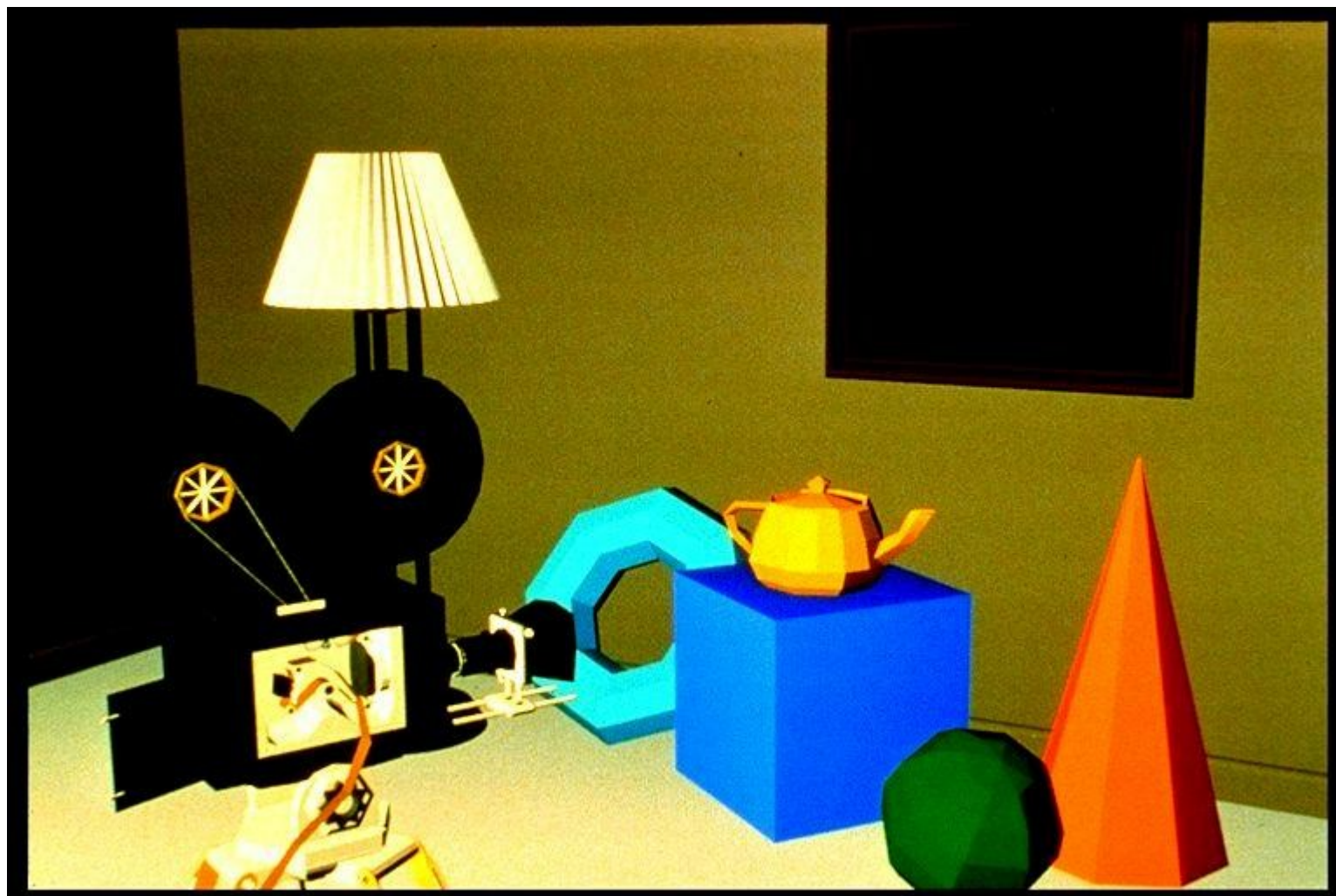
# Constant shading



# Flat shading

- Neste modelo de preenchimento (ou sombreamento), a cor de todos os pontos de polígono é constante.
  - O modelo de iluminação é aplicado uma vez para cada polígono.
- Abordagem torna-se válida diante das seguintes condições:
  - Fonte de luz situa-se no infinito, então  $N.L$  é constante ao longo da face.
  - Observador situa-se no infinito, então  $N.V$  é constante ao longo da face.
  - O polígono representa uma superfície sendo modelada e não uma aproximação para uma superfície curva.
- Se duas primeiras condições são falsas, precisamos de algum método para determinarmos  $L$  e  $V$ , como centro do polígono ou primeiro vértice.

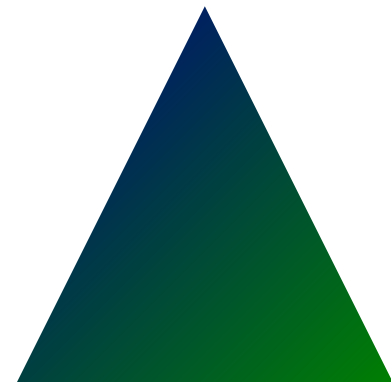
# Flat shading





# Interpolated shading

- Proposto por Wylie, Romney, Evans e Erdahl em 1967
- Informação de sombreamento é interpolada linearmente ao longo de um triângulo a partir dos valores determinados para seus vértices
- Interpolação do sombreamento não é fisicamente correta



# Interpolated shading

- Problemas de superfícies curvas: aparência facetada.
  - Flat Shading e Interpolated Shading geram resultados com essas características
- Efeito de Mach Band
  - Diferença da intensidade em uma aresta é acentuada se existe uma descontinuidade de uma intensidade
  - Causado por um fenômeno ótico
- Métodos apresentados até aqui apresentam esse efeito





# Polygon mesh shading

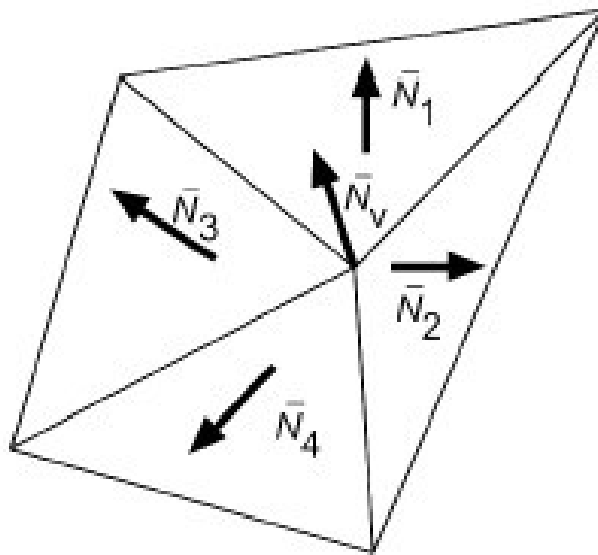
- Existem dois modelos para malhas poligonais que aproveitam as informações fornecidas pelos polígono adjacentes para simular uma superfície suave:
  - Gouraud Shading
  - Phong Shading

# Polygon mesh shading

- Também conhecido como sombreamento por interpolação de intensidade ou sombreamento por interpolação de cor;
- Elimina descontinuidades de intensidade;
- Não elimina completamente mudanças de intensidade;
- Estende conceito de sombreamento interpolado, considerando a superfície e não cada polígono individualmente.

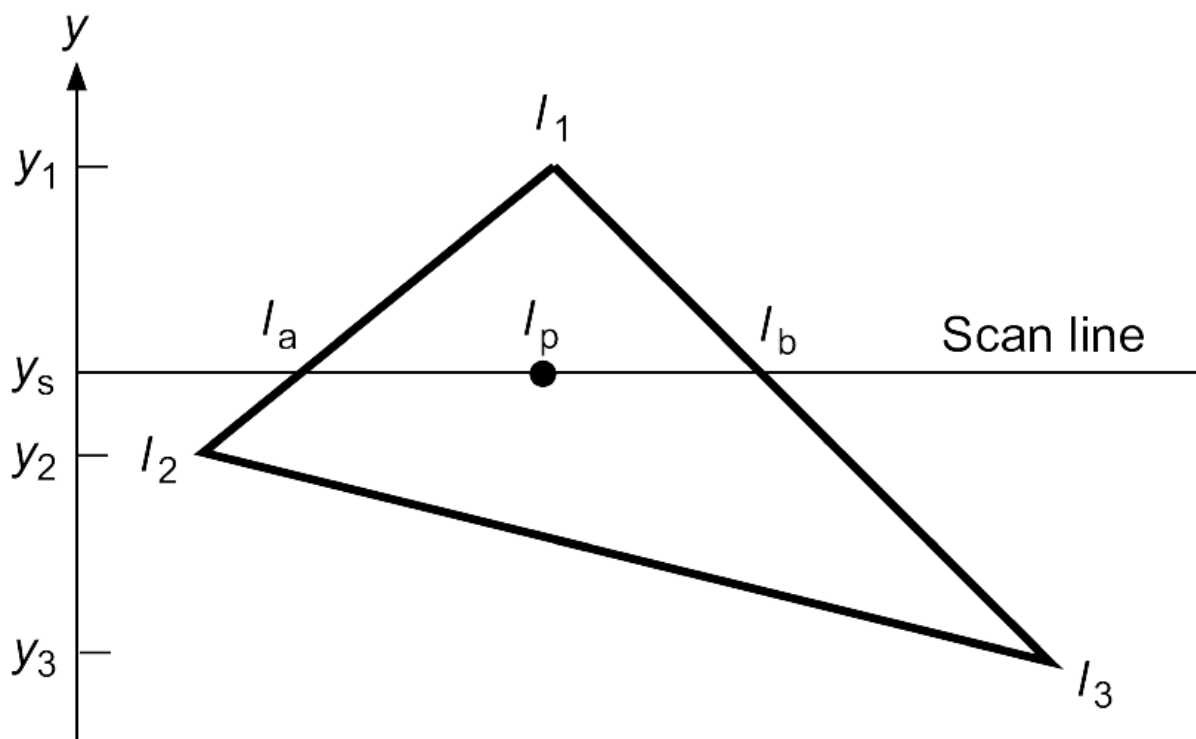
# Polygon mesh shading

- Necessita da normal de cada vértice.
- Pode-se obter esta normal fazendo a média entre as normais das superfícies às quais o vértice pertence.



# Gouraud Shading

- Próximo Passo:
  - Calcular a intensidade no vértice, aplicando um modelo de iluminação.
  - Após isso, calcular a interpolação destas intensidades

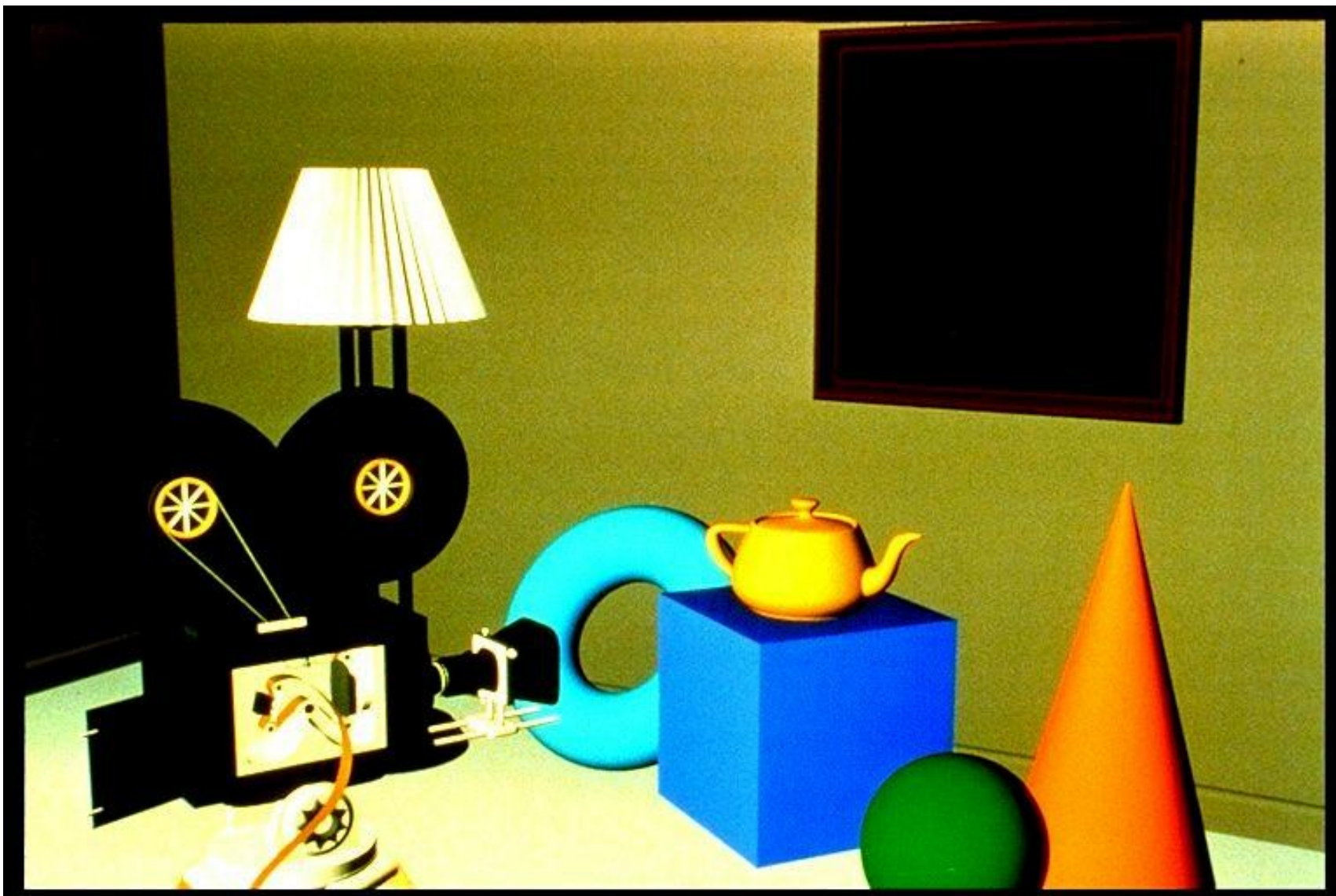


$$I_a = I_1 - (I_1 - I_2) \frac{y_1 - y_s}{y_1 - y_2}$$

$$I_b = I_1 - (I_1 - I_3) \frac{y_1 - y_s}{y_1 - y_3}$$

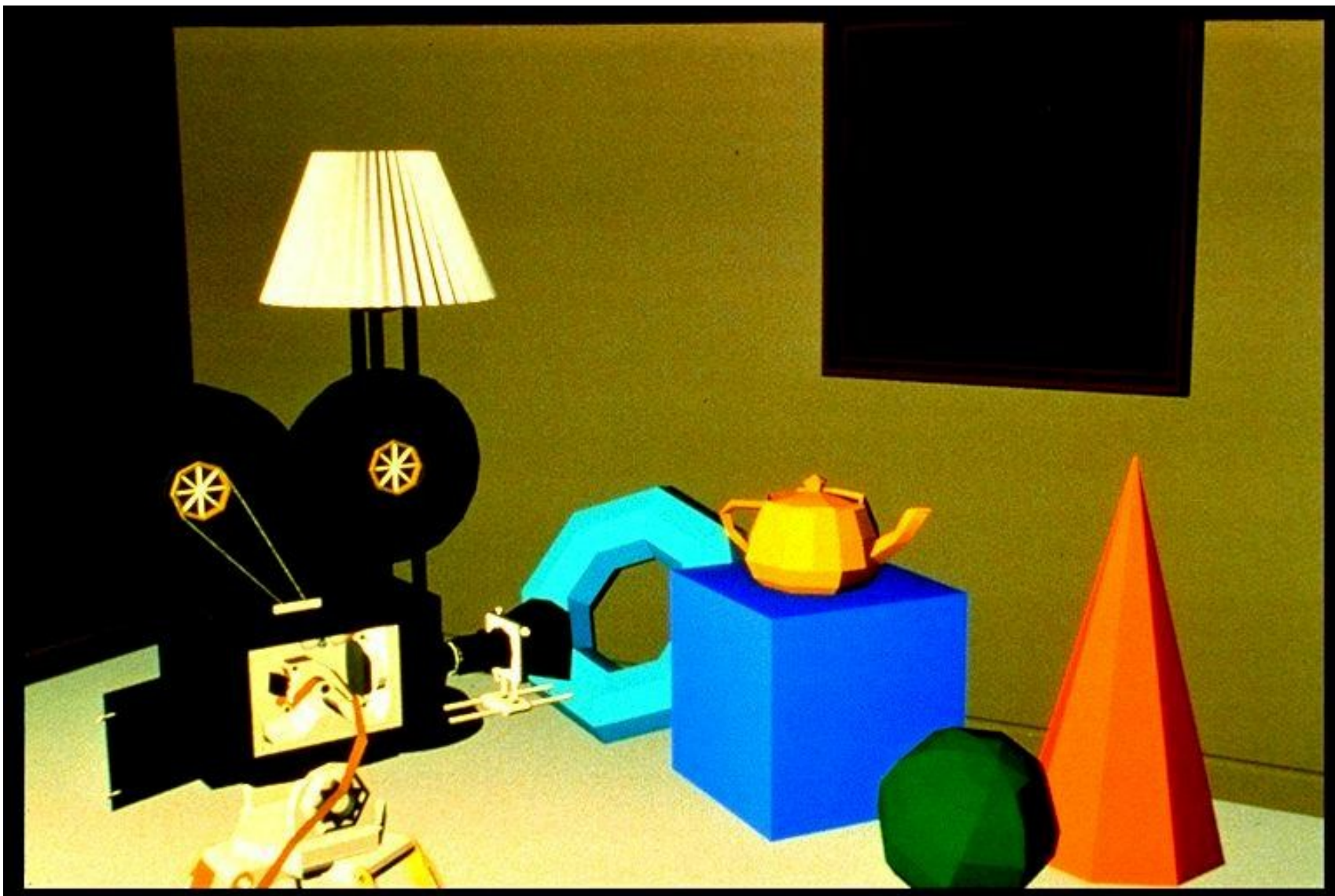
$$I_p = I_b - (I_b - I_a) \frac{x_b - x_p}{x_b - x_a}$$

# Gouraud Shading





# Comparando: Flat shading

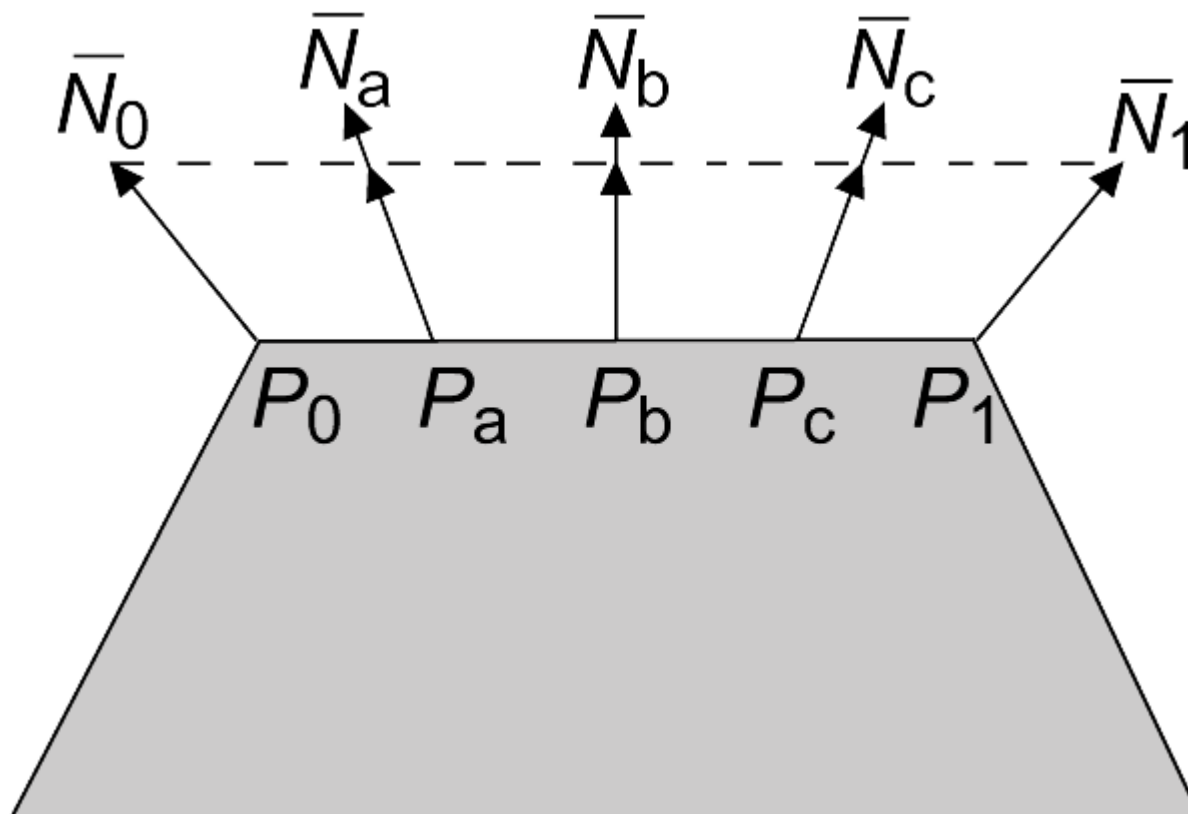


# Phong Shading

- Também conhecido como interpolação do vetor normal
- Interpola a normal da superfície, ao invés da intensidade
- Normais obtidas a partir das normais dos vértices
- Processo:
  - Encontra a normal para o ponto dado
  - Calcula-se a intensidade com aquela normal

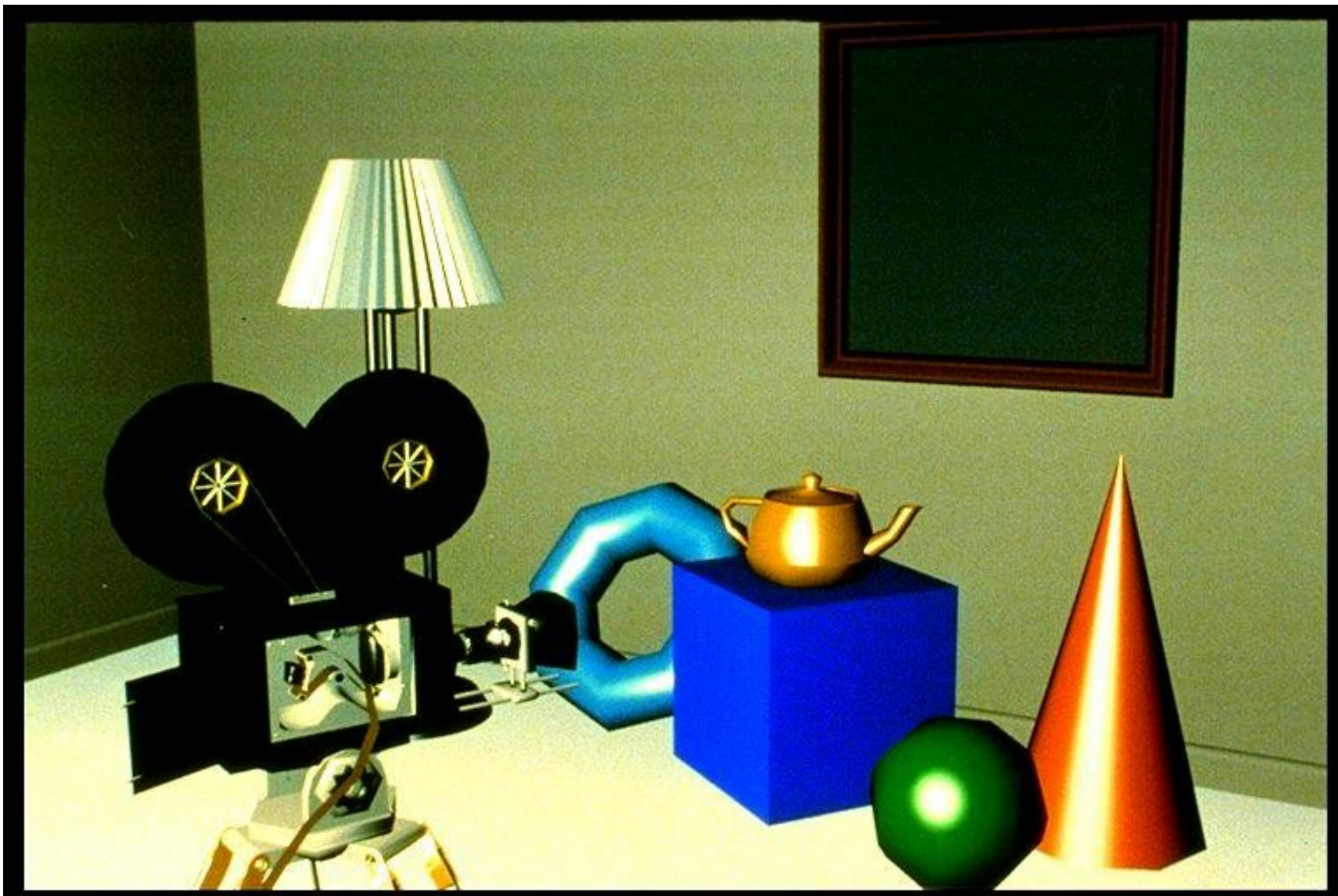
# Phong Shading

- Processo:
  - Encontra a normal para o ponto dado
  - Calcula-se a intensidade com aquela normal





# Phong Shading



# Gouraud x Phong

- Resultados obtidos com Phong shading são mais realistas
- Gouraud pode apresentar problemas, principalmente com componente especular.
- Porque?

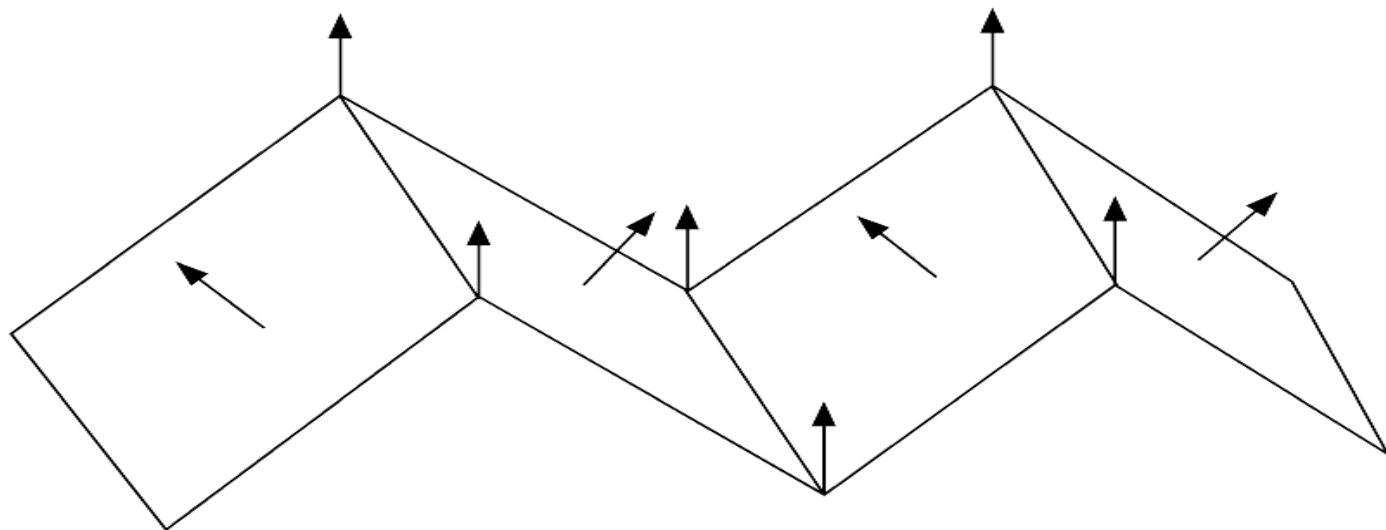
# Gouraud x Phong

- Gouraud pode apresentar problemas, principalmente com componente especular.



# Problemas com Abordagens Baseadas em Interpolação

- Silhueta Poligonal
- Dependência de Orientação
- Problemas com Vértices Compartilhados
- Normais nos Vértices não representativos



# Iluminação em OpenGL

# Materiais em OpenGL

- As propriedades do material dum objecto definem como ele interage com as fontes de luz por forma a produzir a sua cor final.
- As propriedades do material são definidas através de:
  - `glMaterial{fi}(GLenum face, GLenum pname, T param) ;`
  - `glMaterial{fi}v(GLenum face, GLenum pname, T *params) ;`
- Os objetos podem ter materiais diferentes para os polígonos visíveis (anteriores) e para os polígonos ocultos (posteriores).

# Propriedades dos Materiais em OpenGL

- `GL_AMBIENT`, `GL_DIFFUSE`, `GL_SPECULAR` e `GL_AMBIENT_AND_DIFFUSE` são usadas para definir como o material interage com as componentes respectivas da fonte de luz.
- `GL_SHININESS` controla o extensão do realce ou brilho especular.
- `GL_EMISSION` controla a quantidade de luz que um objecto emite.

# Cor do Material em OpenGL

- Normalmente, quando a iluminação é ativada, a cor primária (especificada por `glColor()`) é ignorada.
- Contudo, pode ser conveniente mudar as cores do material através de `glColor()` em vez de usar `glMaterial()`. Isto é possível se for ativada a cor do material com:
  - `glEnable(GL_COLOR_MATERIAL);`
- As componentes de material (ambiente, difusa, ambiente e difusa, ou especular) e as faces (anteriores, posteriores, ou ambas) afetadas pela cor do material podem ser controladas com:
  - `glColorMaterial(GLenum face, GLenum mode);`



# Normais em OpenGL

- A normal corrente é activada com:
- `void glNormal3{bsifd}(TYPE nx, TYPE ny, TYPE nz);`
- `void glNormal3{bsifd}v(const TYPE *v);`
- As normais devem ter comprimento unitário para que os resultados sejam corretos. Se a matriz *modelview* mudar o comprimento das suas normais, há que renormalizá-los usando:
- `glEnable(GL_NORMALIZE);`
- Se se variar a escala dum modo uniforme, uma forma alternativa a `GL_NORMALIZE` é:
- `glEnable(GL_RESCALE_NORMAL);`

# Modelo de Luz em OpenGL

- O modelo de iluminação pode ser modificado usando:
- `void glLightModel{if}(GLenum pname, TYPE param);`
- `void glLightModel{if}v(GLenum pname, const TYPE *param);`
- As propriedades que podemos modificar incluem as seguintes:
- `GL_LIGHT_MODEL_AMBIENT` – controla a luz ambiente global aplicada a todos os aobjectos
- `GL_LIGHT_MODEL_LOCAL_VIEWER` – controla se o observador está no infinito (modelo menos aproximado) ou está na posição da câmara (modelo mais aproximado)

# Modelo de Luz em OpenGL

- `GL_LIGHT_MODEL_TWO_SIZE` — controla se a iluminação é calculada para as faces anteriores e posteriores separadamente ou não.
- `GL_LIGHT_MODEL_COLOR_CONTROL` — permite que a OpenGL interpole a cor especular separadamente, aplicando-a então após a texturização, por forma a preservar os brilhos reflexos.