

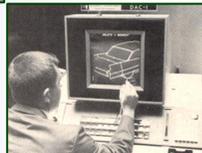


Cap.6 Iluminação



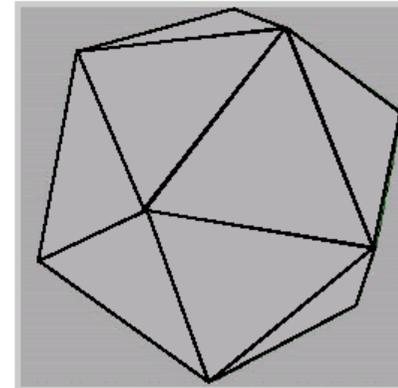
Engenharia Informática (5385)

- 2º ano, 2º semestre

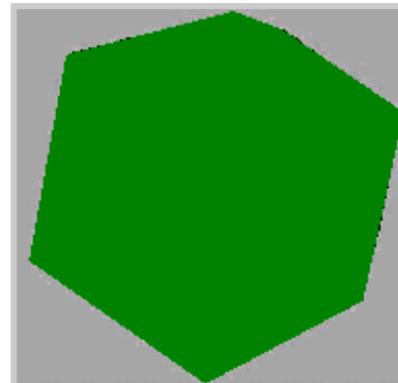


Motivação

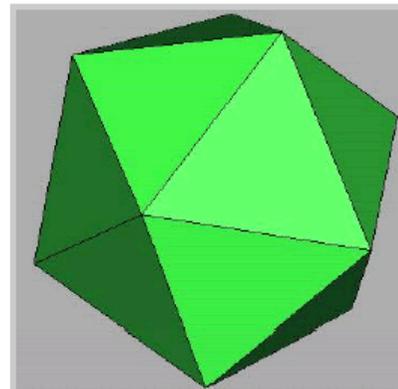
- Modelo de Iluminação = aproximação da iluminação do mundo real
- Sensação da 3-dimensionalidade, percepção da profundidade
- Na maior parte das aplicações, usa-se o modelo de iluminação de Phong porque é temporalmente mais eficiente e porque permite uma representação bastante aproximada do mundo real.



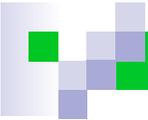
wireframe



sem
iluminação



iluminação
directa



Modelos de iluminação independentes da luz

■ Coloração em Profundidade (Depth Shading)

- Cor ou intensidade determinada somente pela "profundidade" do polígono.
- Cores ou intensidades mais escuras em pontos de maior profundidade no objecto: por exemplo, na modelação de superfícies terrestres.
- Evita os cálculos complexos dos modelos dependentes da luz.
- Faz uma simulação realística.

■ Percepção da Profundidade (Depth Cueing)

- Reduz a intensidade do pixel quando a distância ao observador aumenta.
- Simula a redução em claridade quando a distância ao observador aumenta.
- A imagem desvanece com a distância.
- Frequentemente usada em imagiologia médica.



Modelos de iluminação dependentes da luz

- O que um objecto parece depende de:
 - Propriedades da fonte de luz, tais como: cor, distância entre a fonte de luz e o objecto, direcção definida pela fonte de luz e pelo objecto, intensidade da fonte de luz
 - Características da superfície do objecto, tais como: cor e propriedades de reflexão
 - Localização do observador.
- A luz que incide na superfície dum objecto pode ser:
 - Reflectida (reflexão difusa & reflexão especular)
 - Absorvida
 - Transmitida (translucente ou transparente)
 - Combinação das três anteriores

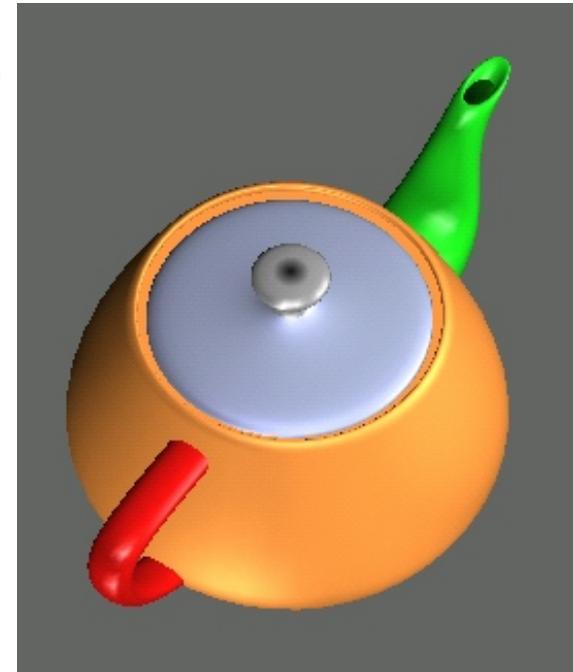
Modelos de iluminação baseados na luz

■ Iluminação Directa ou Local

- Interacção simples entre luz & objectos
- Processo em tempo-real suportado pela OpenGL
- Exemplo: *modelo de iluminação de Phong*

■ Iluminação Indirecta ou Global

- Interacção múltipla entre luz & objectos: reflexões entre *objectos*, *sombras* e *refracções*
- Não é em tempo-real (ainda)
- Exemplos: *raytracing*, *radiosidade*, *photon mapping* ...





Sobrevisão: modelos baseados na luz

■ Iluminação Directa ou Local

- Tipos de Luz
- Fontes de Luz (emissão)
- Materiais da superfície dos objectos (reflexão)

■ Iluminação Indirecta ou Global

- Sombras
- Refracções
- Reflexões entre objectos



Tipos de Luz

■ Luz Ambiente

- Vem de todas as direcções; quando atinge a superfície, espalha-se igualmente em todas as direcções.
- Consequência do espalhamento da luz: *não depende* do ponto de vista (do observador).

■ Luz Difusa

- Vem de uma direcção; quando atinge a superfície, espalha-se igualmente em todas as direcções.
- Consequência do espalhamento da luz: *não depende* do ponto de vista (do observador).

■ Luz Especular

- Vem de uma direcção; tende a reflectir na superfície numa direcção preferencial.
- Consequência do espalhamento da luz: *depende* do ponto de vista (do observador).



Tipos de Luz (cont.)

■ Luz Ambiente

- Pode ser usada para dar a sensação da *cor principal do ambiente*.
- Contribuições:
 - iluminação em contra-luz tem uma grande percentagem de luz ambiente
 - um foco de luz fora-de-portas tem uma percentagem muito pequena de luz ambiente

■ Luz Difusa

- É o tipo de luz que *mais se aproxima da cor da luz*.
- Contribuições:
 - qualquer luz que venha duma posição ou direcção particular

■ Luz Especular

- É a luz que *mais se aproxima da luz unidireccional*.
- Contribuições:
 - um feixe laser bem-colimado que incida num espelho de elevada qualidade produz quase 100 por cento de reflexão especular
 - um metal brilhante tem uma elevada componente de luz especular
 - o giz não reflecte quase nenhuma luz especular



Fontes de luz e material

■ Fontes de Luz

- Tipos:
 - ambiente, posicional, direccional, cónica
- Cor
 - A cor de luz emitida é dada pelas quantidades de luz vermelha, verde e azul.
- Número
 - Cada fonte de luz pode ser ligada ou desligada.
- Tipos de luz emitida: ambiente, difusa, especular

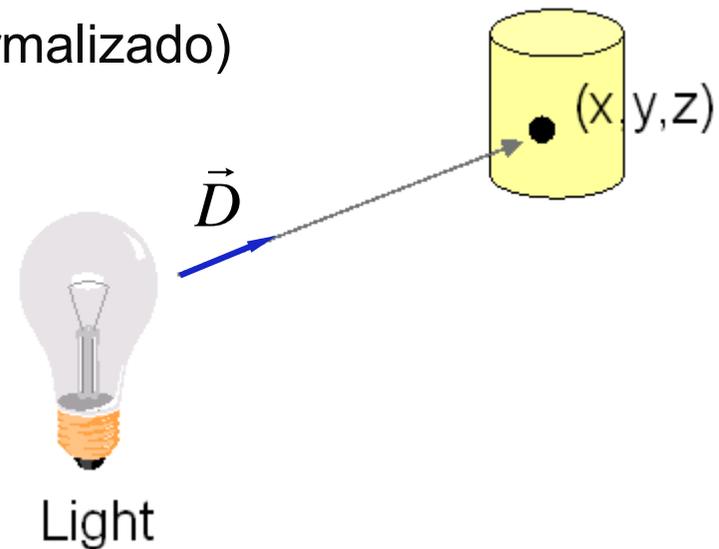
■ Material da Superfície

- Especifica como a luz é reflectida (e absorvida)
 - A cor do material é dada pela percentagem das componentes vermelha, verde e azul que são reflectidas em várias direcções.
 - Superfícies diferentes têm propriedades diferentes; algumas são brilhantes, pelo que reflectem preferencialmente a luz em certas direcções, ao passo que outras espalham igualmente a luz em todas as direcções. A maior parte das superfícies situam-se entre os dois extremos anteriores.
- Tipo de luz emitida: **emitida**
- Tipos de luz reflectida: ambiente, difusa, especular

Modelação de Fontes de Luz

■ Modelo da Fonte de Luz: $I_L(P, \vec{D}, \lambda)$

- descreve a intensidade de energia,
- que sai da fonte de luz
- e que chega à posição $P(x, y, z)$
- vinda da direcção \vec{D} (vector normalizado)
- com comprimento de onda λ



Tipos de Fontes de Luz

■ São os seguintes:

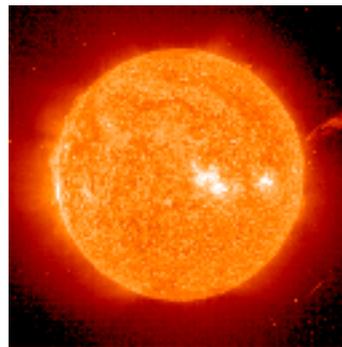
- Fonte de luz **ambiente**
- Fonte de luz **pontual**
- Fonte de luz **direccional**
- Fonte de luz **cónica** (spotlight)

LÂMPADA



fonte de luz pontual

SOL



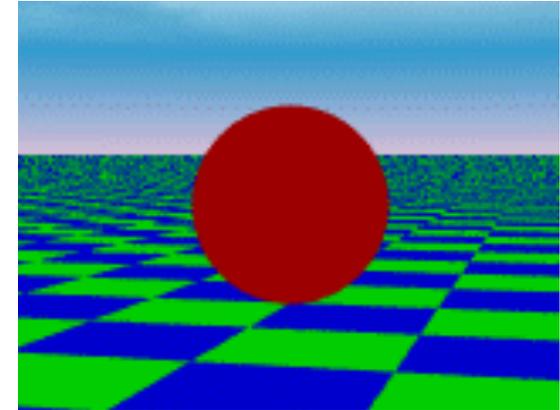
fonte de luz direccional

HOLOFOTE



fonte de luz cónica

Fonte de Luz Ambiente



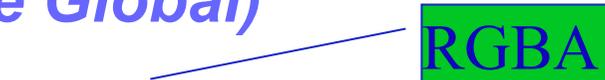
- Um objecto que não esteja directamente iluminado é ainda visível
 - por causa da luz reflectida a partir doutras superfícies
- Modelada por uma simples fonte de luz ambiente
 - Em vez de calcular as reflexões nas superfícies dos objectos, especifica-se uma **luz ambiente constante** para todas as superfícies
 - Definida somente pelas intensidades de luz ambiente RGB
- Intensidade da luz ambiente I_L que chega a um ponto $P(x,y,z)$: $I(P, I_L) = I_L$

Luz Ambiente Global em OpenGL

- Não vem de nenhuma fonte de luz em particular.
- Permite-nos ver objectos numa cena mesmo que nela não haja quaisquer fontes de luz.
- A sua intensidade RGBA é especificada pelo parâmetro `GL_LIGHT_MODEL_AMBIENT` como se segue:♪

Exemplo: (Luz Ambiente Global)

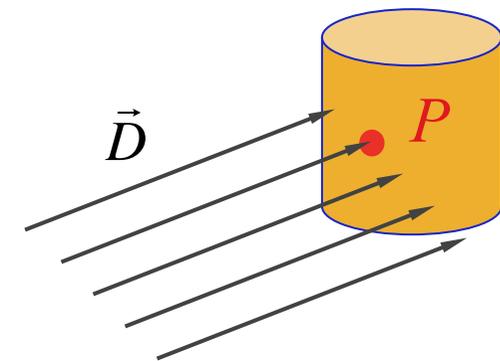
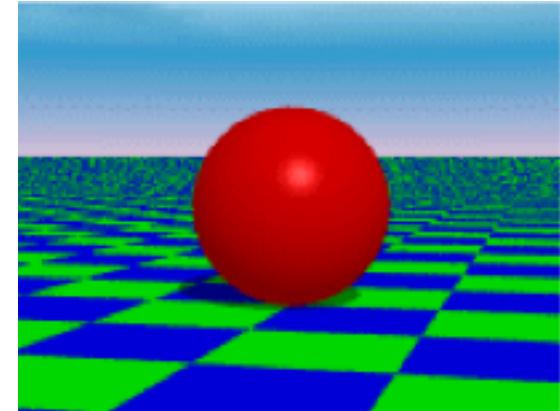
```
// sets global ambient light  
GLfloat lmodel_ambient[]={0.2,0.2,0.2,1.0};  
glLightModelfv(GL_LIGHT_MODEL_AMBIENT, lmodel_ambient);
```



Fonte de Luz Direccional

- Modela uma fonte de luz pontual no infinito (e.g. sol)
 - Definida pelas intensidades de luz emitida RGB de todos os tipos, e
 - pela direcção \vec{D}
- A direcção é importante para calcular a luz reflectida
- Intensidade da luz I_L que chega ao ponto $P(x,y,z)$:

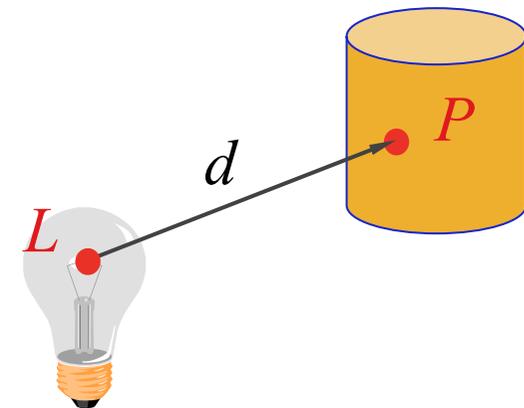
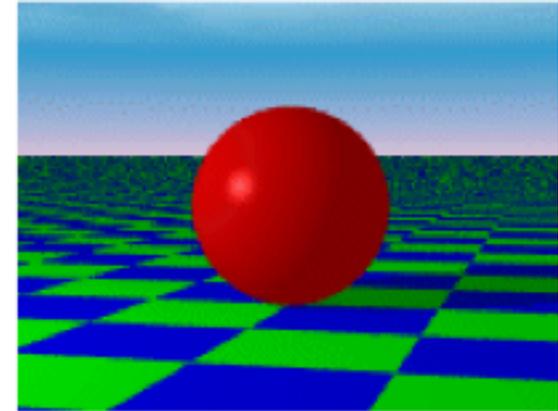
$$I(P, \vec{D}, I_L) = I_L$$



nenhuma atenuação
com a distância

Fonte de Luz Pontual

- Luz emitida a partir dum ponto numa forma *radial* em todas as direcções (fonte omni-direccional)
 - Definida pelas intensidades de luz (RGB) emitida de todos os tipos,
 - pela posição $L(x,y,z)$, e
 - pelos factores (k_c, k_l, k_q) de atenuação com a distância d a $P(x,y,z)$
- Intensidade da luz pontual I_L que chega a $P(x,y,z)$:



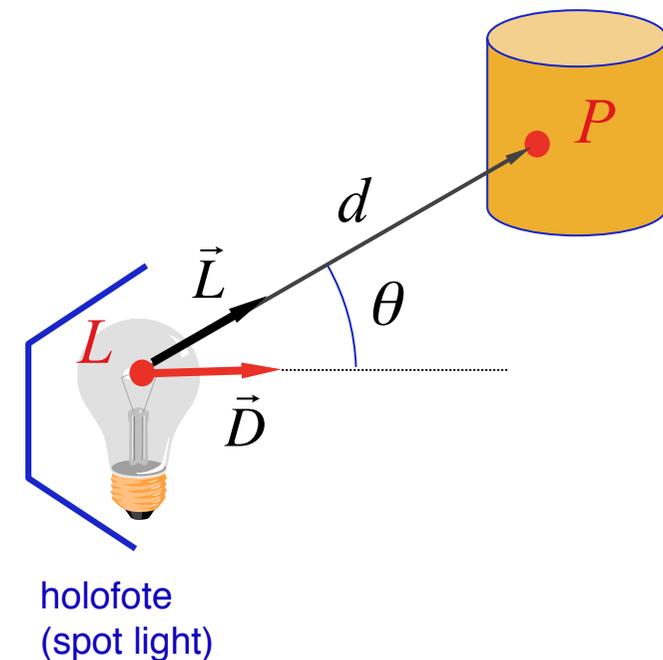
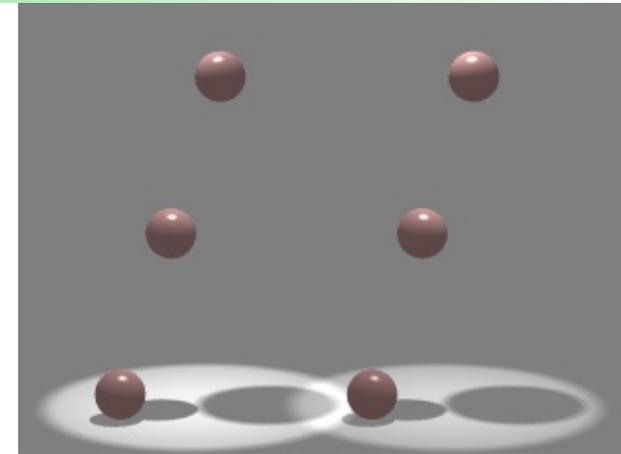
lâmpada

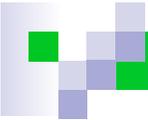
$$I(P, L, k_c, k_l, k_q, I_L) = \frac{I_L}{k_c + k_l d + k_q d^2}$$

Fonte de Luz Cônica

- Luz emitida num cone (e.g. candeeiro Luxo Jr.)
 - Definida pelas intensidades de luz (RGB) emitida de todos os tipos,
 - pela posição L , direcção \vec{D} , expoente de recorte do holofote
 - pelos factores de atenuação constante, linear e quadrática (k_c, k_l, k_q)
- Intensidade da luz de holofote I_L que chega ao ponto P (x, y, z):

$$I(P, L, k_c, k_l, k_q, I_L) = \frac{I_L (\vec{D} \cdot \vec{L})}{k_c + k_l d + k_q d^2}$$





Direcção e Posição das Fontes de Luz em OpenGL

- **Fonte de luz direccional.** Está localizada no infinito relativamente à cena.
- **Fonte de luz pontual ou posicional.** A sua distância à cena é finita.♪

Exemplo: (Fonte de Luz Direccional)

```
// sets GL_LIGHT0 with direction (x=1.0,y=1.0,z=1.0) at an
infinite position (w=0.0) in homogeneous coordinates
GLfloat light_position[]={1.0,1.0,1.0,0.0};
glLightfv(GL_LIGHT0, GL_POSITION, light_position);
```

Exemplo: (Fonte de Luz Pontual ou Posicional)

```
// sets GL_LIGHT0 at the position (x=1.0,y=1.0,z=1.0) that is
finite (w≠0.0) in homogeneous coordinates
GLfloat light_position[]={1.0,1.0,1.0,1.0};
glLightfv(GL_LIGHT0, GL_POSITION, light_position);
```

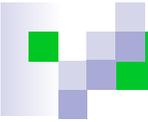
Direcção e Posição de Fontes de Luz em OpenGL (cont.)

- À semelhança da fonte de luz pontual, uma fonte luz de holofote (spot light) é também uma fonte de luz posicional.
- **Fonte de luz pontual.** Por defeito, a propriedade *spotlight* está inactiva porque o parâmetro **GL_SPOT_CUTOFF** é 180.0 graus. Este valor significa que a luz é emitida em todas as direcções (o ângulo no ápice do cone é 360 graus, ou seja, não é um cone afinal de contas).
- **Fonte de luz de holofote.** O valor do ângulo de recorte **GL_SPOT_CUTOFF** do holofote está limitado ao intervalo [0.0,90.0].♪

Exemplo: (Fonte de Luz do Holofote)

```
// sets GL_LIGHT0 as a spotlight with a cutoff angle of 30 degrees
glLight(GL_LIGHT0, GL_SPOT_CUTOFF, 30.0);
```

```
// sets spotlight's direction or the light cone axis
GLfloat spot_direction[]={-1.0,-1.0,0.0};
glLightfv(GL_LIGHT0, GL_SPOT_DIRECTION, spot_direction);
```



Cor da Luz Emitida por uma Fonte de Luz em OpenGL

- A OpenGL permite associar 3 parâmetros de cor a qualquer fonte de luz particular :
 - **GL_AMBIENT.** É a intensidade RGBA da luz ambiente que uma fonte de luz particular adiciona à cena.
 - Valores RGBA por omissão: (0.0,0.0,0.0,1.0) ⇒ nenhuma luz ambiente
 - **GL_DIFFUSE.** É a intensidade RGBA da luz difusa que uma fonte de luz particular adiciona à cena.
 - Valores RGBA por omissão: (1.0,1.0,1.0,1.0) para a LIGHT0 (⇒ brilhante, luz difusa branca) e (0.0,0.0,0.0,0.0) para qualquer outra fonte de luz.
 - **GL_SPECULAR.** É a intensidade RGBA da luz especular que uma fonte de luz particular adiciona à cena.
 - Valores RGBA por omissão: (1.0,1.0,1.0,1.0) para a LIGHT0 e (0.0,0.0,0.0,0.0) para qualquer outra fonte de luz.



Cor da Luz Emitida por uma Fonte de Luz em OpenGL (cont.)

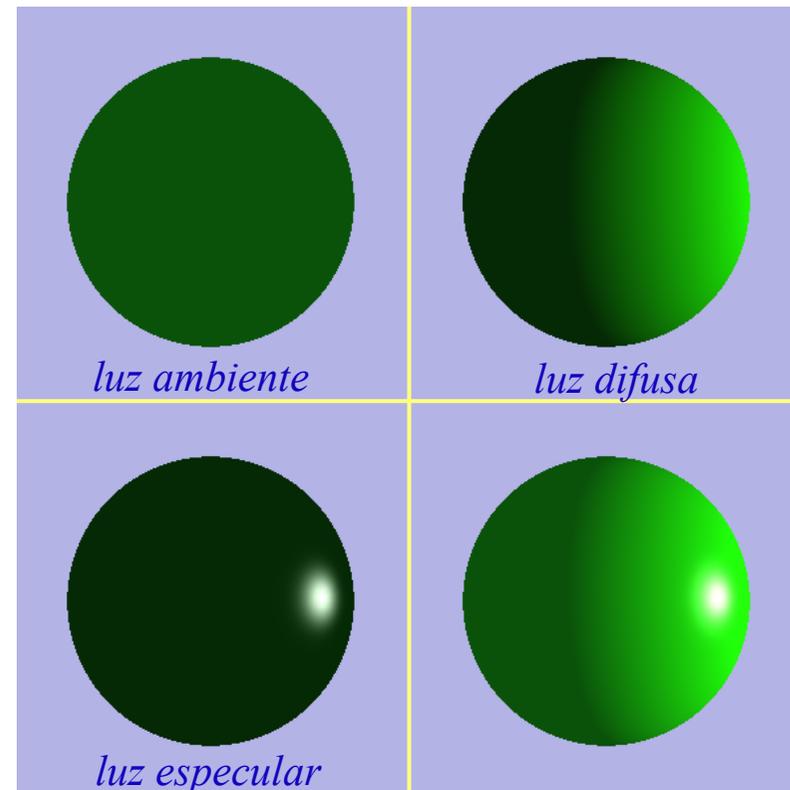
Exemplo: (Cor da Luz Ambiente, Difusa e Especular)

```
// sets the ambient component of GL_LIGHT0
GLfloat light_ambient[]={0.0,0.0,1.0,1.0};    // blue color
GLfloat light_diffuse[]={1.0,1.0,1.0,1.0};    // white color
GLfloat light_specular[]={1.0,1.0,1.0,1.0};    // white color

glLightfv(GL_LIGHT0, GL_AMBIENT, light_ambient);
glLightfv(GL_LIGHT0, GL_DIFFUSE, light_diffuse);
glLightfv(GL_LIGHT0, GL_SPECULAR, light_specular);
```

Componentes de Luz

- A imagem ao lado mostra os efeitos da luz ambiente (canto superior esquerdo), da luz difusa (canto superior direito), luz especular (canto inferior esquerdo), e todas as 3 componentes combinadas (canto inferior direito).



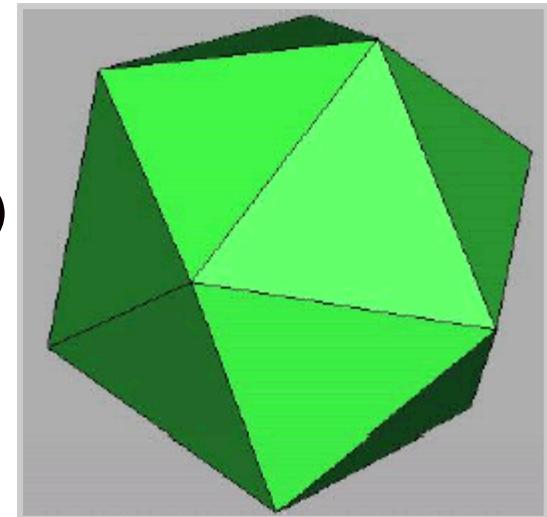
Sobrevisão: modelos baseados na luz

■ Iluminação Directa ou Local

- Tipos de Luz
- Fontes de Luz (emissão)
- Materiais da superfície dos objectos (reflexão)

■ Iluminação Indirecta ou Global

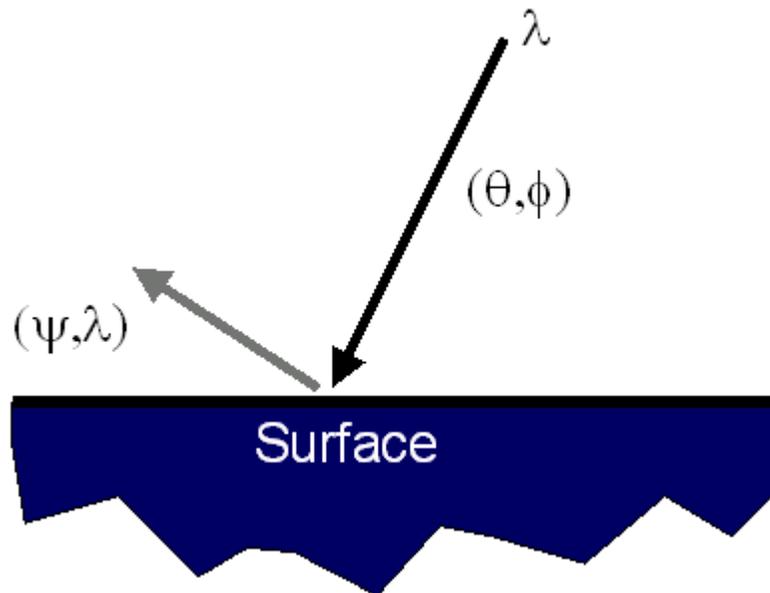
- Sombras
- Refracções
- Reflexões entre objectos



Modelo da Reflexão da Luz na Superfície dum Objecto

■ $R(\theta, \phi, \gamma, \psi, \lambda)$

- Descreve a quantidade de energia incidente na superfície
- que vem da direcção (θ, ϕ)
- e que reflecte segundo a direcção (γ, ψ)
- com o comprimento de onda λ

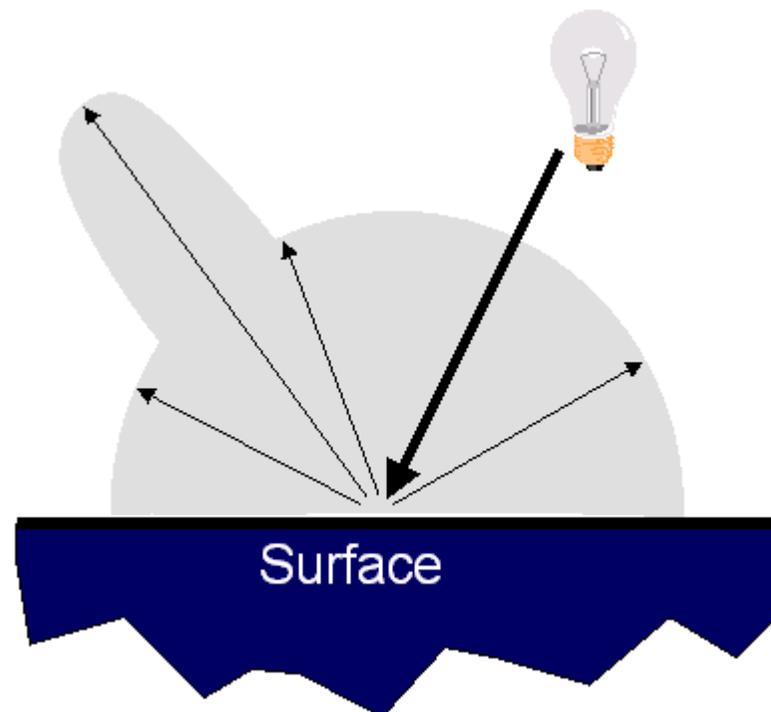


Modelo de Reflexão

■ Modelo Analítico Simplificado:

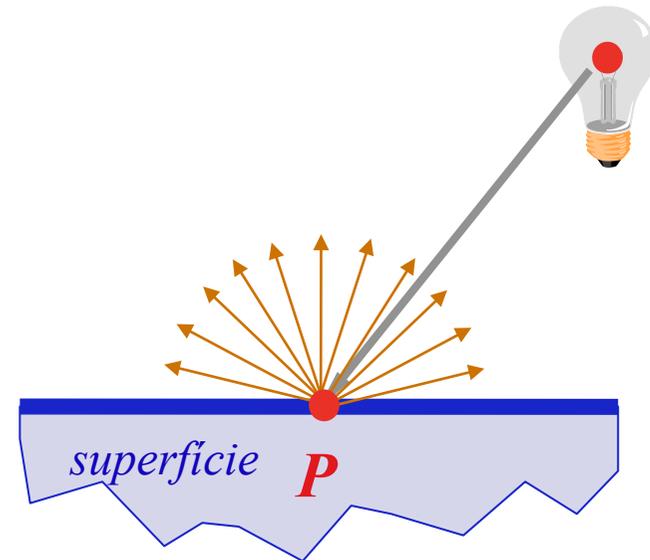
- Reflexão difusa +
- Reflexão especular +
- Emissão +
- Ambiente

Baseado no modelo proposto por Phong



Reflexão Difusa

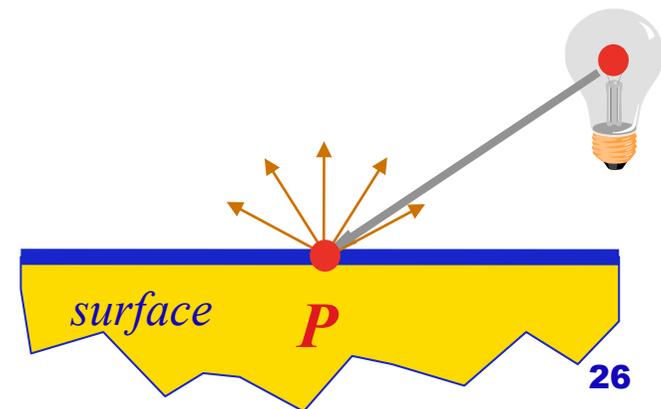
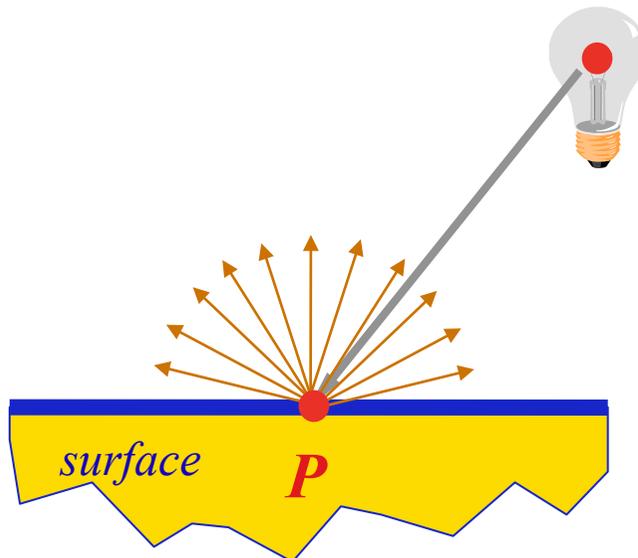
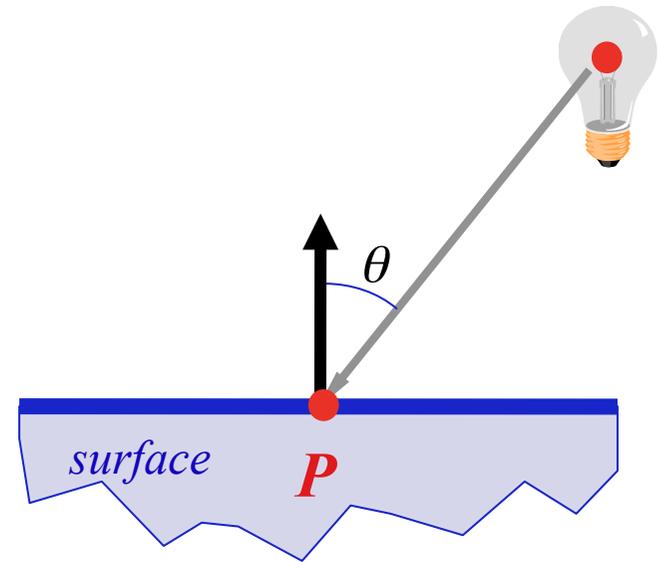
- Um **reflector difuso ideal**, a nível microscópico, é uma superfície muito rugosa (exemplo do mundo real: giz)
- Devido a estas variações microscópicas, um raio de luz incidente é igualmente reflectido em todas as direcções acima do hemisfério definido pela superfície.
- Ou seja, assume-se que a **superfície reflecte igualmente em todas as direcções**.



Reflexão Difusa

■ Como é que a luz é reflectida?

- Depende do ângulo θ da luz incidente.
- Quanto maior é o valor de θ , menor é a quantidade de luz reflectida.
- A quantidade de luz reflectida depende da posição da fonte de luz e da posição do objecto, mas é independente da posição do observador.



Reflexão Difusa

■ Modelo Lambertiano

- Lei dos Cosenos de Lambert (producto interno)

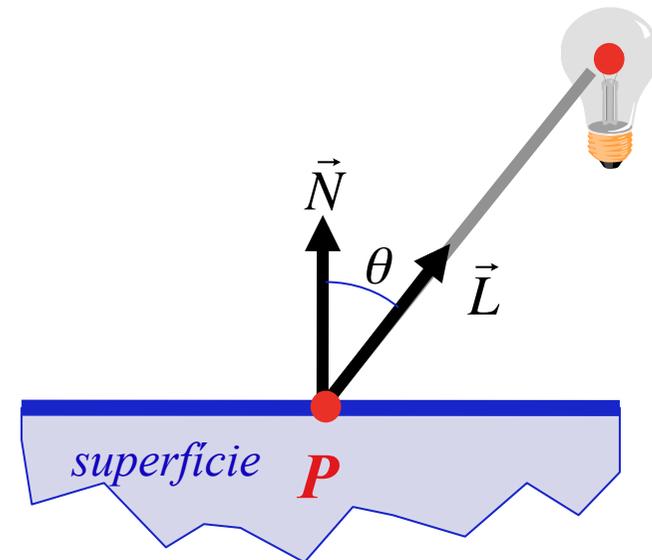
A intensidade de luz difusa I_D reflectida pela superfície no ponto $P(x,y,z)$ é proporcional ao cosseno do ângulo entre o vector \vec{L} direcção da fonte de luz e o vector normal \vec{N} à superfície em $P(x,y,z)$.

- I_D = intensidade de luz difusa reflectida
- I = intensidade da fonte de luz em $P(x,y,z)$
- K_D = coeficiente de reflexão da superfície ($0 \leq K_D \leq 1$)
- θ = entre 0 e 90 graus

$$I_D = K_D I \cos \theta$$

$$\text{com } \cos \theta = \frac{\vec{N} \cdot \vec{L}}{\|\vec{N}\| \|\vec{L}\|} = \vec{N} \cdot \vec{L},$$

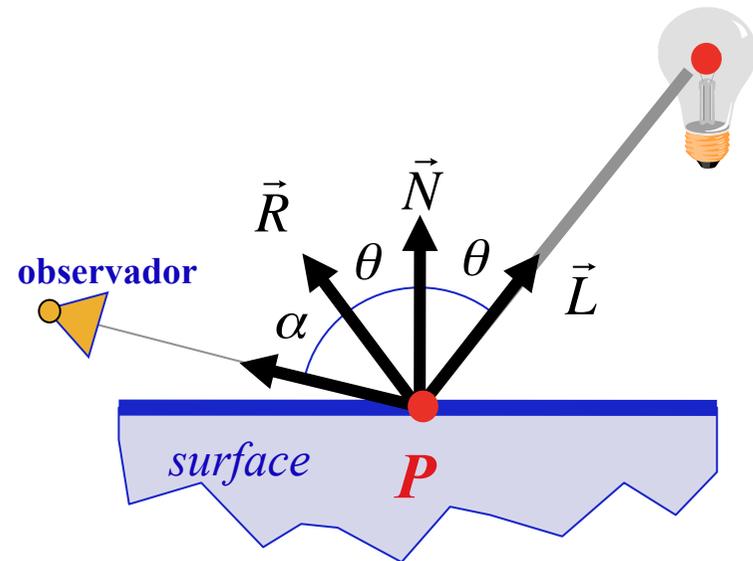
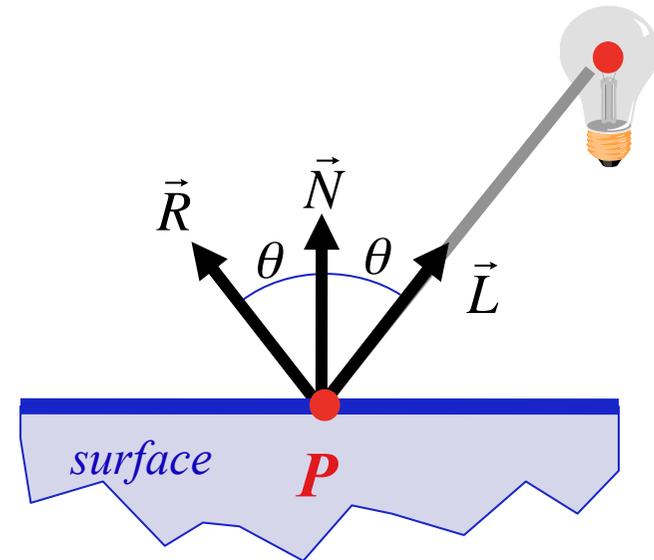
em que \vec{N} e \vec{L} são vectores unitários



$$I_D = K_D (\vec{N} \cdot \vec{L}) I$$

Reflexão Especular

- Reflexão é mais intensa junto do ângulo de reflexão
 - Exemplos: espelhos, metais
 - Visível quando o ângulo θ da luz incidente é igual ao ângulo de luz reflectida em direcção ao observador.
- Quanta luz é vista?
 - Depende do ângulo θ da luz incidente e do ângulo α ao observador
 - Se o reflector não é perfeito, a intensidade da luz reflectida diminui rapidamente quando o ângulo α ao observador aumenta relativamente ao ângulo de incidência.♪



Reflexão Especular

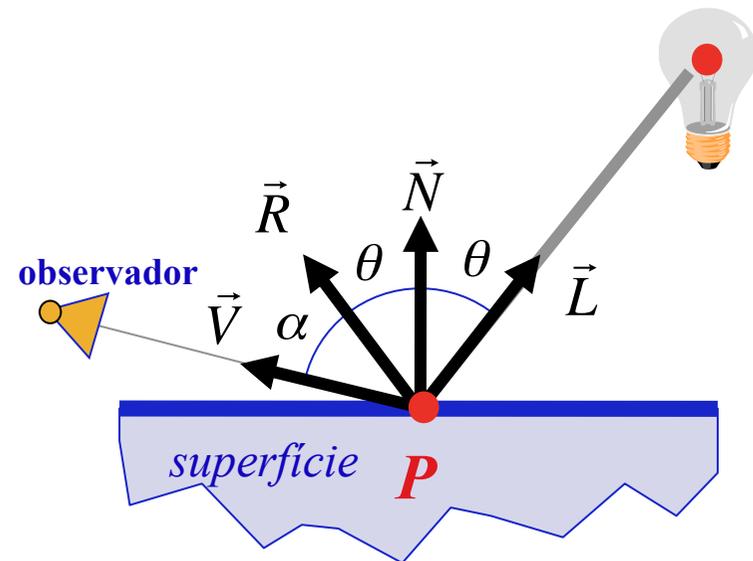
■ Modelo de Phong

- $(\cos \alpha)^n$
- $n = \text{expoente de reflexão especular (reflector perfeito } n=\infty)$
- $I_S = \text{intensidade de luz especular reflectida}$
- $I = \text{intensidade de luz em } P(x,y,z)$
- $K_S = \text{coeficiente de luz especular reflectida } (0 \leq K_S \leq 1)$
- $\theta = \text{entre } 0 \text{ e } 90 \text{ graus}$

$$I_S = K_S I (\cos \alpha)^n$$

$$\text{com } \cos \alpha = \frac{\vec{V} \cdot \vec{R}}{\|\vec{V}\| \|\vec{R}\|} = \vec{V} \cdot \vec{R},$$

e \vec{V} e \vec{R} vectores unitários



$$I_S = K_S (\vec{V} \cdot \vec{R})^n I$$

Luz emissiva gerada por uma fonte de luz superficial

- É produzida por uma **fonte de luz superficial**, não por uma fonte de luz pontual.
- Representa a luz emitida directamente por um polígono ou disco dum objecto.
- Isto é preciso porque alguns objectos do mundo-real, nomeadamente lâmpadas, emitem luz.
- Assim, se uma lâmpada faz parte duma cena, temos de especificar não só a sua fonte de luz posicional, mas também que o material da sua superfície emite luz.♪

material emissivo $\neq (0,0,0,0)$



$$I_{EL} = I_E$$



Luz emissiva dum fonte de luz superficial em OpenGL

- Pela especificação dum cor RGBA para `GL_EMISSION`, podemos fazer com que um objecto pareça emitir luz daquela cor.
- Dado que os objectos do mundo-real (excepto luzes) não emitem luz, podemos usar o parâmetro `GL_EMISSION` para simular o funcionamento de lâmpadas e outras fontes de luz de uma cena..
- Contudo, uma fonte de luz superficial não funciona realmente como uma fonte de luz. Para conseguir tal efeito, é necessário também criar uma fonte de luz e posicioná-la na mesma posição que o objecto emissivo de luz.

Exemplo: (Luz emissiva a partir do material da superfície dum objecto)

```
GLfloat mat_emission[]={0.3,0.2,0.2,0.0};  
glMaterialfv(GL_FRONT, GL_EMISSION, mat_emission);
```

Reflexão Ambiente

- Representa a reflexão de toda a iluminação indirecta
 - Luz ambiente que resulta da iluminação dum cena através da luz reflectida pelos objectos existentes na cena.
 - Calcular a luz ambiente dum forma exacta seria muito complicado.
 - Por isso, adopta-se o modelo simplificado que assume que a luz ambiente é uniforme em todo o ambiente.
 - I_{AL} = intensidade de luz ambiente reflectida
 - K_A = coeficiente de luz ambiente reflectida
 - I_A = intensidade de luz ambiente ♪



$$I_{AL} = K_A I_A$$

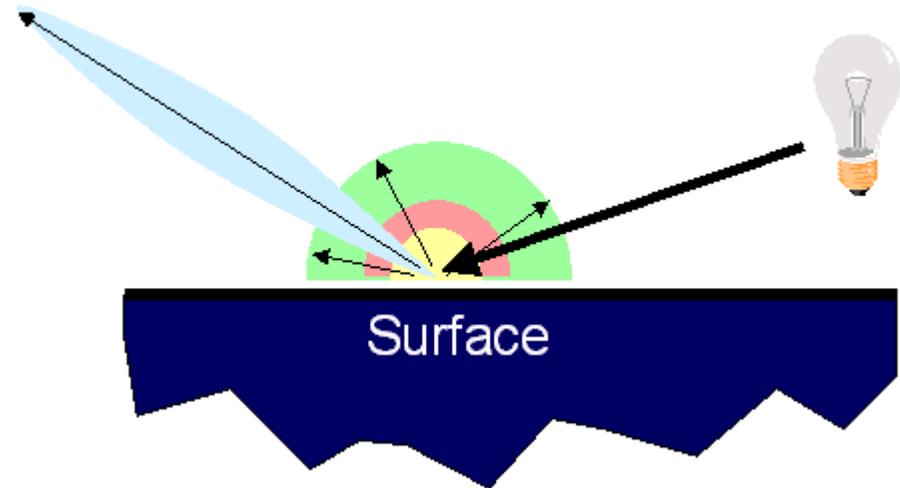
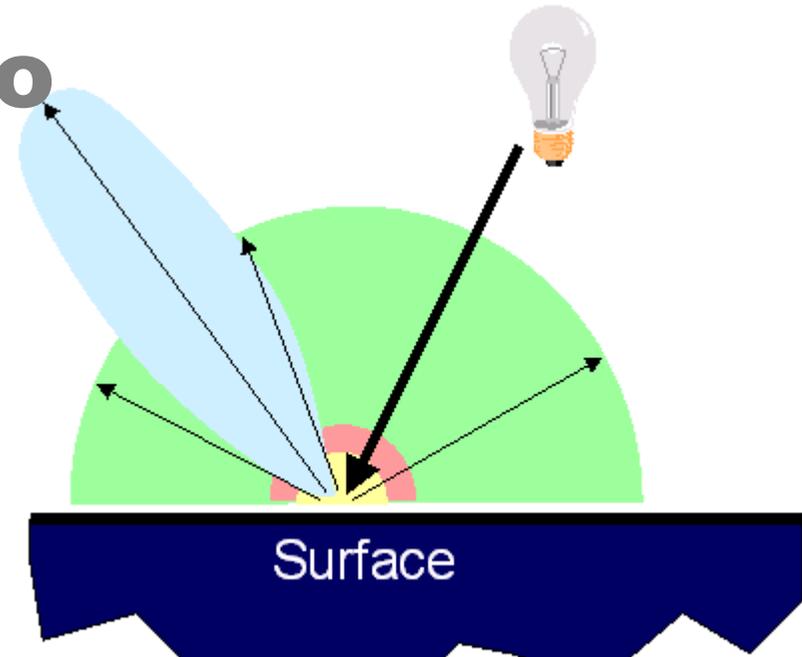
Isto evita a complexidade da iluminação global!

Modelo de Reflexão

■ Modelo Analítico

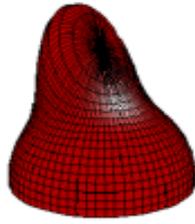
Simplificado:

- Reflexão difusa +
- Reflexão especular +
- Emissão +
- “Ambiente”



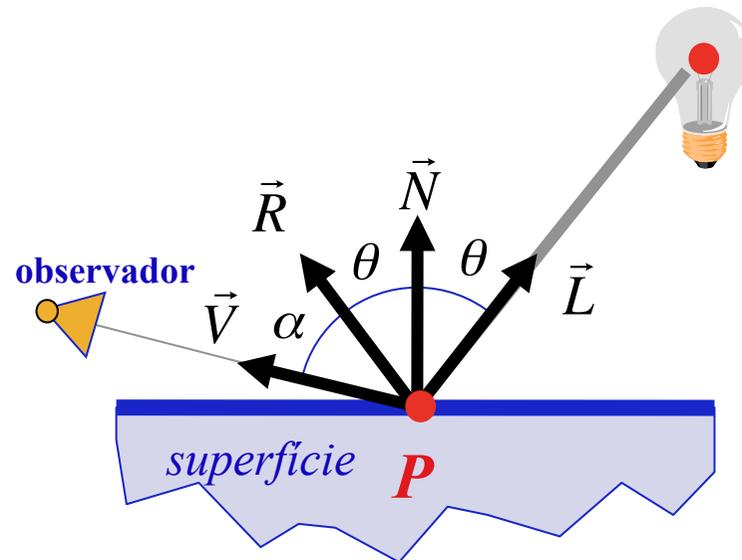
Modelo de Reflexão

- Soma das Componentes Difusa, Especular, Emissiva e Ambiente

Phong	ρ_{ambient}	ρ_{diffuse}	ρ_{specular}	ρ_{total}
$\phi_i = 60^\circ$				
$\phi_i = 25^\circ$				
$\phi_i = 0^\circ$				

Cálculo da Iluminação da Superfície

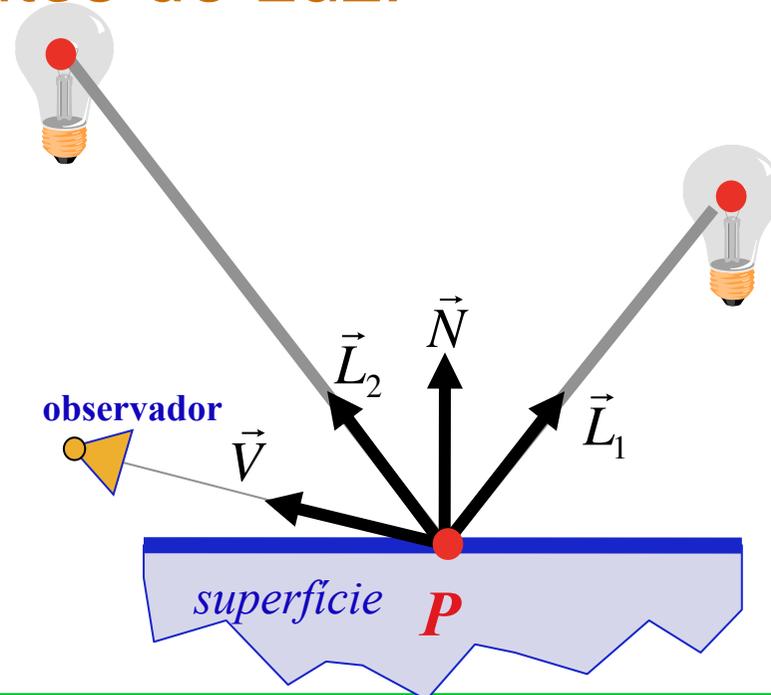
■ 1 Fonte de Luz:



$$I = I_E + K_A I_A + K_D (\vec{N} \cdot \vec{L}) I + K_S (\vec{V} \cdot \vec{R})^n I$$

Cálculo da Iluminação da Superfície

■ Várias Fontes de Luz:



$$I = I_E + K_A I_A + \sum_{i=1}^{\# \text{ lights}} [K_D (\vec{N} \cdot \vec{L}_i) + K_S (\vec{V} \cdot \vec{R}_i)^n] I_i$$



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 objectos 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_AMBIENT_AND_DIFFUSE`, e `GL_SPECULAR` 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 activada, 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 se activar 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) afectadas 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 correctos. Se a matriz *model/view* 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);
```



The Lighting Model in 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 objectos
 - **GL_LIGHT_MODEL_LOCAL_VIEWER** – controla se o observador está no infinito (modelo menos aproximado) ou está na posição da câmara (modelo mais aproximado)
 - **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.

Sobrevisão: modelos baseados na luz

■ Iluminação Directa ou Local

- Tipos de Luz
- Fontes de Luz (emissão)
- Materiais da superfície dos objectos (reflexão)



■ Iluminação Indirecta ou Global

- Sombras
- Refracções
- Reflexões entre objectos

