



IA725 – Computação Gráfica I

Modelos de iluminação

15/05/2008

Shirley, capítulo 9
Red Book, capítulo 6



Iluminação em OpenGL

- O modelo de iluminação do OpenGL é baseado no modelo de Blinn-Phong, com algumas modificações.

$$I = \underbrace{k_e}_{\text{emissão}} + \underbrace{k_a I_{ga}}_{\text{ambiente global}} + \sum_{j=1}^{\text{fontes de luz}} [\underbrace{f_{att_j}}_{\text{ambiente}} (\underbrace{k_a I_{aj}}_{\text{ambiente}} + \underbrace{k_d I_{dj} \max(\vec{N} \cdot \vec{L}, 0)}_{\text{difusa}}) + \underbrace{k_s I_{sj} \max(\vec{N} \cdot \vec{H}, 0)^n}_{\text{especular}}]$$

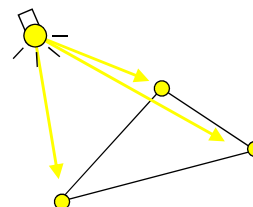
- Cada material tem uma reflectância ambiente.
- Cada material tem uma componente de emissão de luz.
- A componente ambiente do modelo de Blinn-Phong é chamada de contribuição “ambiente global.”



Iluminação em OpenGL



- Equação de iluminação é calculada para cada face ou vértice.
 - Cores são interpoladas sobre a superfície através de modos de tonalização.
- Ajustando o modelo de iluminação do OpenGL:
 - Propriedades da luz.
 - Habilitar/desabilitar computação da iluminação.
 - Propriedades dos materiais das superfícies.
 - Propriedades das fontes de luz.



Iluminação em OpenGL



- Propriedades das fontes de luz:
 - Cores / posição ou direção / tipo / atenuação.
 - Comando: `glLightfv(light, property, value).`
 - 1
 - 2
 - 3
1. Qual fonte de luz está sendo especificada.
 - Ex.: `GL_LIGHT0, GL_LIGHT1, GL_LIGHT2 ...`
 - Podem ser criadas várias fontes de luz, até 8.
 2. Qual propriedade da fonte de luz será alterada.
 - Ex.: `GL_AMBIENT, GL_DIFFUSE, GL_SPECULAR, GL_POSITION.`
 3. Valor referente à propriedade da luz.



Iluminação em OpenGL



- Exemplo de ajuste de propriedades da fonte de luz:

```
GLfloat light_ambient[] = {0.0, 0.0, 0.0, 1.0};
GLfloat light_diffuse[] = {1.0, 1.0, 1.0, 1.0};
GLfloat light_specular[] = {1.0, 1.0, 1.0, 1.0};
GLfloat light_position[] = {0.0, 0.0, 1.0, 1.0};

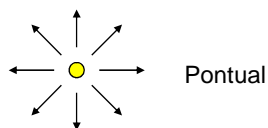
glLightfv(GL_LIGHT0, GL_AMBIENT, light_ambient);
glLightfv(GL_LIGHT0, GL_DIFFUSE, light_diffuse);
glLightfv(GL_LIGHT0, GL_SPECULAR, light_specular);
glLightfv(GL_LIGHT0, GL_POSITION, light_position);
```



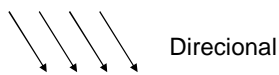
Iluminação em OpenGL



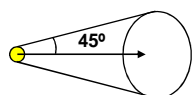
- Tipos de fontes de luz:



Pontual



Direcional



Spot

Altera apenas a componente W da posição da fonte de luz.

- W=1 para fonte de luz pontual.
- W=0 para fonte direcional.

```
GLfloat position[] = {x,y,z,w};
glLightfv(GL_LIGHT0,
          GL_POSITION, position);
```

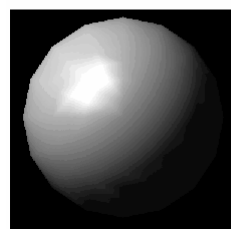
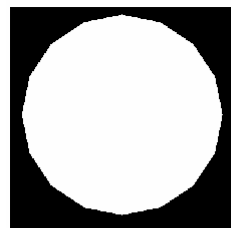
```
glLightf(GL_LIGHT0, GL_SPOT_CUTOFF,
         45.0);
GLfloat direction[] = {1.0,0.0,0.0};
glLightfv(GL_LIGHT0,
          GL_SPOT_DIRECTION, direction);
```



Iluminação em OpenGL



- Ligando e desligando as luzes.
- Ligando/desligando todas as luzes:
 - `glEnable(GL_LIGHTING);`
 - `glDisable(GL_LIGHTING);`
- Se a iluminação está desabilitada, as cores das primitivas é definida pelos comandos `glColor*`.
- Ligando/desligando cada fonte de luz:
 - `glEnable(GL_LIGHTn);`
onde $n = 0, 1, 2, \dots, 7$.



OpenGL - Atenuação



- Em OpenGL, o fator de atenuação pode ser constante, linear ou quadrático, e inclui a atenuação da luz *spot*.
- $f_{att} = \text{efeito_spot} + 1 / k_c + k_l d + k_q d^2$.
 - d é a distância da fonte de luz ao ponto da superfície.
 - k_c é o fator de atenuação constante (`GL_CONSTANT_ATTENUATION`). *Default* = 1.
 - k_l é o fator de atenuação linear (`GL_LINEAR_ATTENUATION`). *Default* = 0.
 - k_q é o fator de atenuação quadrático (`GL_QUADRATIC_ATTENUATION`). *Default* = 0.
- Atenuação não funciona para luzes direcionais.



OpenGL - Atenuação



- Exemplo de especificação de fatores de atenuação:
 - `glLightf(GL_LIGHT0, GL_CONSTANT_ATTENUATION, 2.0);`
 - `glLightf(GL_LIGHT0, GL_LINEAR_ATTENUATION, 1.0);`
 - `glLightf(GL_LIGHT0, GL_QUADRATIC_ATTENUATION, 0.5);`
- As contribuições de luz ambiente, difusa e especular são atenuadas.
- Contribuições de emissão e luz ambiente global não são atenuadas.
- Atenuação requer pelo menos uma divisão a mais para cada cor calculada.



OpenGL – Iluminação



- Configuração de parâmetros globais do modelo de iluminação.
 - `glLightModelf(name, param);`
 - 1
 - 2
- 1. Parâmetro do modelo de iluminação.
 - `GL_LIGHT_MODEL_AMBIENT.`
 - `GL_LIGHT_MODEL_LOCAL_VIEWER.`
 - `GL_LIGHT_MODEL_TWO_SIDE.`
- 2. Intensidade RGBA ambiente global
(`GL_LIGHT_MODEL_AMBIENT`) ou um valor 0 ou diferente de 0
(`GL_LIGHT_MODEL_LOCAL_VIEWER, GL_LIGHT_MODEL_TWO_SIDE`).



OpenGL - Materiais



- Definição da cor e propriedades da superfície de um material (Ex.: rugoso, polido, etc.)
 - Quanto a superfície reflete as luzes incidentes (coeficientes de reflexão ambiente, difusa e especular).
 - Comando: `glMaterialfv`(face, property, value).
1. Propriedade do material para qual lado da face: `GL_FRONT`, `GL_BACK`, `GL_FRONT_AND_BACK`.
 2. Qual a propriedade do material para ajustar (`GL_AMBIENT`, `GL_DIFFUSE`, `GL_SPECULAR`, `GL_SHININESS`, `GL_EMISSION`, etc).
 3. Valor a ser designado para a propriedade.



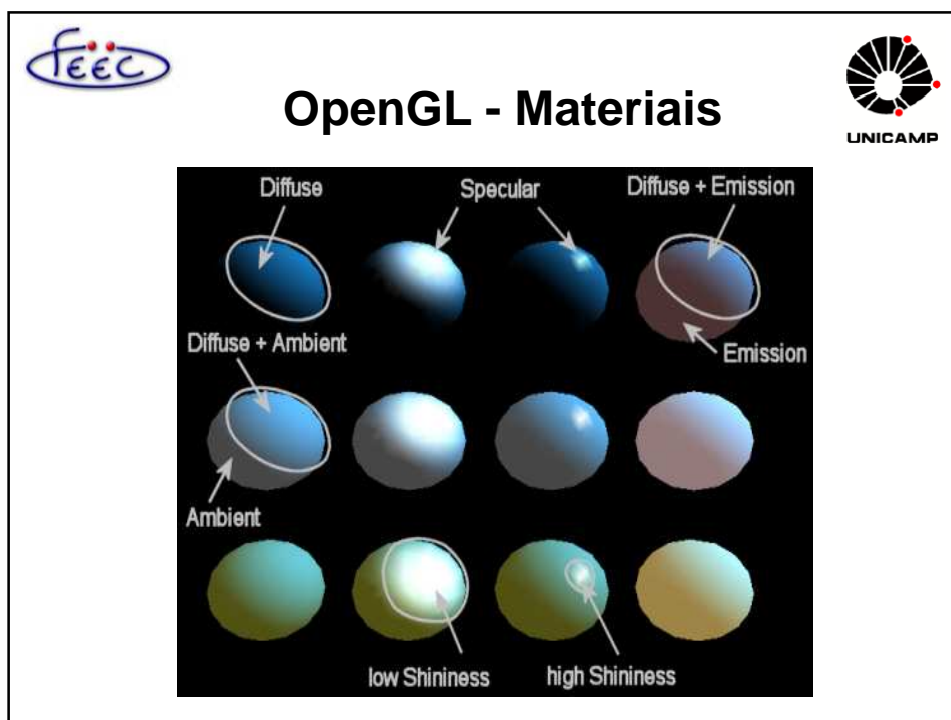


OpenGL - Materiais



- Exemplo de definição das propriedades de um material:

```
GLfloat mat_amb_diff[] = {1.0, 0.5, 0.8, 1.0};
GLfloat mat_specular[] = {1.0, 1.0, 1.0, 1.0};
GLfloat shininess[] = {5.0};

glMaterialfv(GL_FRONT_AND_BACK,
             GL_AMBIENT_AND_DIFFUSE, mat_amb_diff);
glMaterialfv(GL_FRONT, GL_SPECULAR, mat_specular);
glMaterialfv(GL_FRONT, GL_SHININESS, shininess);
```

Tonalização

- Até agora vimos alguns métodos para calcular iluminação local em pontos da superfície.
 - Utilizando as propriedades das fontes de luz (posição, direção, ângulo de abertura).
 - Utilizando as propriedades das superfícies e materiais (vetor normal, reflectância).
 - Utilizando a direção de visão para reflexão especular.
- Por eficiência, esse cálculo é feito somente para alguns pontos da superfície (Ex.: Vértices).
- Vamos explorar métodos para aplicar a iluminação (ou outra cor) a todos os *pixels* da superfície rasterizada.



Tipos de tonalização



- Em renderização de modelos poligonais, há 3 principais tipos de tonalização:
 - Tonalização *flat*.
 - Tonalização Gouraud.
 - Tonalização Phong.
- Esses tipos correspondem basicamente a:
 - Uma cor por polígono.
 - Interpolação de cores a partir dos vértices.
 - Cor calculada para cada *pixel*.



Tonalização *flat*



- A equação de iluminação é avaliada para apenas um ponto do polígono.
- A cor obtida é utilizada para preencher todo o polígono.



$$centroid = \frac{1}{vertices} \sum_{i=1}^{vertices} \bar{p}_i$$

- Para qual ponto do polígono a equação deve ser avaliada?



Tonalização *flat*



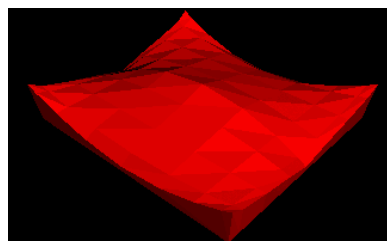
- Produz um aspecto de geometria facetada.
 - Desejável para a visualização de poliedros.
 - Indesejável para a visualização de malhas que aproximem superfícies suaves.
- A causa está na descontinuidade da variação do vetor normal nas bordas dos polígonos.
- Como esse efeito pode ser suavizado?
 - Em outras palavras, como eliminar/reduzir as descontinuidades?



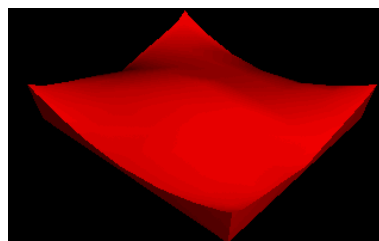
Tonalização Gouraud



- A equação de iluminação é avaliada para cada vértice do polígono.
- A primitiva é preenchida com a interpolação bilinear das cores calculadas nos vértices.



Flat



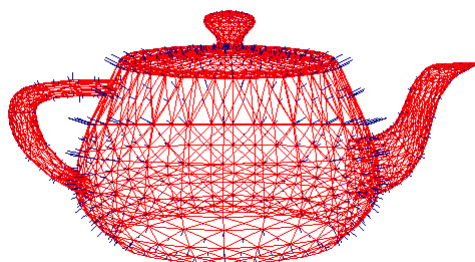
Gouraud



Tonalização Gouraud



- Em malhas de polígonos, requer o cálculo de normais aos vértices.



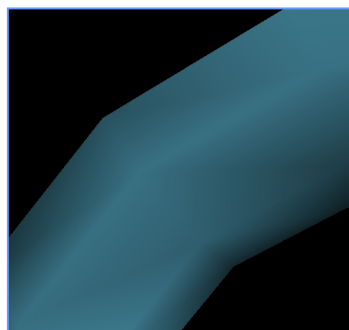
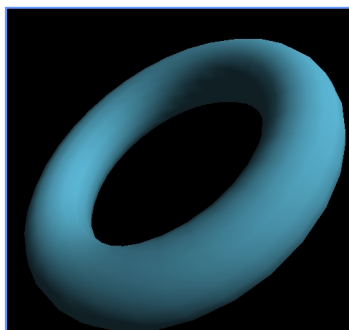
- Como calcular essas normais? Qual o melhor método?



Tonalização Gouraud



- Resultados não são perfeitos.
 - *Mach bands*.

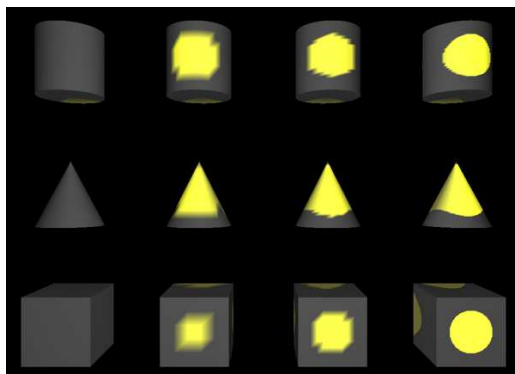




Tonalização Gouraud



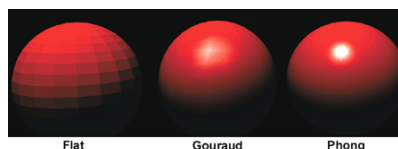
- Resultados não são perfeitos.
 - Perda ou espalhamento do brilho especular (depende da resolução da malha).



Tonalização Phong



- Os vetores normais estimados nos vértices são interpolados linearmente para cada *pixel*.
 - Normais interpoladas precisam ser normalizadas.
- Modelo de iluminação é então avaliado para cada *pixel*.
- Não confundir com o modelo de iluminação de Phong.



Flat

Gouraud

Phong



OpenGL - Tonalização



- OpenGL suporta dois modelos de tonalização ativados pelo comando `glShadeModel`:
 - `GL_FLAT`: Iluminação é avaliada uma vez por polígono.
 - `GL_SMOOTH`: Iluminação é avaliada para cada vértice. Equivalente à tonalização de Gouraud.
 - OpenGL utiliza um modelo de iluminação de Blinn-Phong, mas não possui suporte nativo à tonalização de Phong.
 - Atualmente, placas gráficas programáveis podem implementar tonalização de Phong (e muitas outras coisas).



OpenGL - Tonalização



- Cálculo de vetores normais:
 - O vetor normal atual é especificado com o comando `glNormal*` e é aplicado para todos os vértices subsequentes.
 - Devem ser de tamanho unitário, mas o tamanho das normais é afetado por transformações de escala.
 - Solução: Re-normalizar as normais manualmente, ou habilitar o estado `GL_NORMALIZE`.