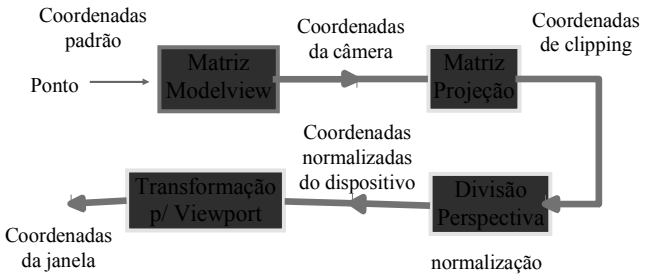
Iluminação (lighting) e Sombreamento (shading)

Cap 16: Foley
Cap 5: OpenGL
Aula 11: Notas do Dave

Revisão



Introdução

- Até aqui vimos a parte matemática (geométrica) necessária para desenhar objetos na perspectiva correta.
- Mas o que percebemos do mundo depende da luz que atinge os nossos olhos.
  - Percepção de cor
  - Mas o que é cor?
    - Partícula x onda, claro x escuro, marrom, preto etc.

Jargão

- **Illumination:** the transport of luminous flux from light sources between points via direct and indirect paths
- **Lighting:** the process of computing the luminous intensity reflected from a specified 3-D point
- **Shading**: the process of assigning colors to pixels

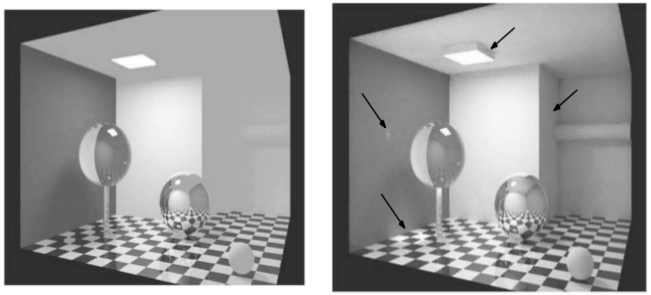
Iluminação e sombreamento

- Iluminação e sombreamento são elementos fundamentais para a geração de imagens realistas
- Porém, é muito difícil simular como a luz interage com os objetos da cena.
- O OpenGL é capaz de simular apenas modelos simples, e portanto tem realismo limitado, mas o faz de forma eficiente.
  - por exemplo, sombras não são modeladas, nem reflexões indiretas

Modelo local x global

- O modelo de iluminação do OpenGL é chamado de modelo de iluminação local, pois apenas considera a interação da luz com um objeto, independentemente dos demais.

Modelo global x local



Local Illumination Model

Global Illumination Model

Fig. 38: Local versus global illumination models.

Luz

- Fenômeno complexo
- Podemos assumir que luz é formada por fótons, partículas emitidas continuamente por uma fonte de luz.
- Cada fóton tem uma quantidade de energia, que é percebida como uma "cor"

Cor

- Outro fenômeno complexo, que vamos tratar mais tarde no curso. Por enquanto, vamos considerar cor como sendo uma tripla:

COR = [R G B]

Intensidade ou fluxo luminoso

- Quantidade de energia luminosa por unidade de área, por unidade de tempo.
- Quando fotons, viajando por um meio, encontram a superfície de um objeto pode ocorrer:
  - Reflexão
  - Transmissão
  - Absorção

Reflexão

- Reflexão pura:
  - A superfície é como um espelho
- Reflexão especular:
  - Reflexão imperfeita (pouco espalhamento), como sobre uma superfície metálica ou plastico
- Reflexão difusa:
  - A luz se espalha, e portanto não é brilhante.

Absorção

- O foton é absorvido e sua energia é dissipada na forma de calor.
- Exemplo: um objeto é percebido como verde pois reflete os fotons “verdes” e absorve os fotons de outras cores.

Transmissão

- O foton atravessa a superfície que pode ser:
  - Transparente: a transmissão é quase perfeita, como vidro
  - Translucida: sofre espalhamento, como a pele humana.

Observação

- Na prática, cada superfície se comporta como uma combinação desses efeitos.
- Além disso, ojetos podem também emitir luz

Fontes de luz

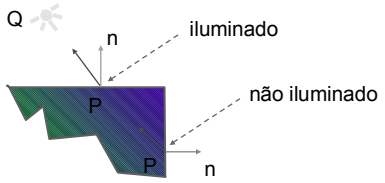
- As fontes de luz que iluminam a cena podem:
  - ter tamanhos e formas diversas.
  - ter comprimento de onda e intensidade variável com a direção de propagação.
- OpenGL: só modela fontes puntuais.
  - As fontes possuem uma função de luminância dada por 3 componentes (Lr, Lg, Lb) para as intensidades do vermelho, verde e azul respectivamente (ou seja, elas podem ter uma cor associada, e não apenas o branco).

Iluminação do ambiente

- A forma como uma cena é iluminada depende bastante de reflexões indiretas. Por isso o OpenGL modela a iluminação local do ambiente através de componentes:
  - **Luz ambiente**: claridade espalhada pelo ambiente, de forma que mesmo um ponto "escondido" da fonte de luz recebe alguma iluminação
  - **Luz pontual**: luz emitida a partir de um ponto.
- A intensidade da luz é definida por essas 2 componentes:
  - Luz ambiente: La = (Lar Lag Lab)T
  - Luz pontual: L(P0) = (Lr(P0), Lg(P0), Lb(P0))T
    - P0 = posição da fonte de luz

Que pontos são iluminados?

- A luz de uma fonte de luz pontual só deve afetar os pontos visíveis daquele ponto.
- Considere o vetor normal n em um ponto qualquer Q na superfície de um objeto.
- Q só será iluminado se o ângulo de P0Q com n for agudo.



Atenuação

- A iluminação é sujeita a atenuação proporcional ao quadrado da distância.
- A física sugere I(P,Q) = I(Q) / |P-Q|2
- Mas utilizando esse modelo, devido às simplificações do OpenGL, um ponto parecerá bem menos iluminado que na realidade.

Atenuação

- Dessa forma utilizamos uma função que inclue componentes lineares, da forma:

  I(P,Q) = I(Q) / (a + b.d + c.d2)

  - d = |P-Q|
  - (a,b,c) parâmetros de atenuação

- valores default no OpenGL:
  - (a,b,c) = (1, 0, 0) => sem atenuação

Fontes de luz direcionais

- Uma fonte de luz pode ser colocada no infinito definindo sua coordenada homogênea como 0.
- Ao meio dia, o eixo z pode ser considerado vertical, sendo o sol posicionado em (0,0,1,0)T
- Vetores dessa forma são chamados de fontes direcionais.
  - Vantagens: os raios de luz de fontes no infinito são paralelos, o que simplifica o cálculo de iluminação sobre superfícies planas.
- Spotlight: luz intensa em uma direção, com atenuação proporcional ao ângulo de abertura do raio.
  - Veja glLight().

Modelo de
Iluminação de Phong

Modelo de Phong

- Além das fontes de iluminação, é necessário modelar como a luz é refletida dos objetos para o observador.
- O modelo de iluminação de Phong é um modelo simples, baseado na combinação das seguintes componentes:
  - **Emissão**: objetos com brilho próprio
  - **Reflexão ambiente**: reflexões indiretas. Todos os objetos são iluminados da mesma forma.
  - **Reflexão de difusão**: reflexões de objetos lisos foscos.
  - **Reflexão especular**: reflexões de objetos lisos brilhantes (metálicos ou polidos).

Iluminação no OpenGL

- O opengl permite até 8 fontes de luz, dependendo da implementação utilizada.
- As propriedades de uma fonte são definidas pelo comando glLight().
- Cada fonte pode ser ligada/desligada pelo comando glEnable().
- As propriedades reflectivas de cada objeto são definidas pelo comando glMaterial(), de forma similar ao comando glColor(). Mais detalhes sobre esses comandos serão apresentados + tarde.

As componentes de luz

- O OpenGL permite que uma luz L = (Lr, Lg,Lb) seja decomposta em 3 componentes:
  - La (ambiente), Ld (difusa), e Ls (especular).
  - Cada uma dessas componentes é definida por um vetor RGB (ou RGBA).
- Como já vimos, a componente ambiente da luz é uma simplificação para modelar reflexões indiretas.
- As componentes difusas e especulares da fonte de luz são em geral definidas como iguais.

Cor

- A cor de um objeto no OpenGL determina a intensidade de luz refletida por ele.
- Seja C = (Cr, Cg, Cb), e assuma que cada componente seja normalizada entre [0,1]
  - Cr = fração da componente de luz vermelha refletida pelo objeto, idem para Cg e Cb

Luz e Cor

- Quando a luz de uma fonte L=(Lr, Lg, Lb) incide sobre um objeto de cor C = (Cr, Cg, Cb), a intensidade de luz refletida é dada pelo produto: $L * C = (LrCr, LgCg, LbCb)$
  - Note que essa é uma multiplicação componente a componente, e não um produto escalar ou vetorial.

Exemplo

L = (1, 1, 1) -> luz branca

C = (0.0, 0.50, 0.0) -> objeto verde

⇒L * C = (0.0, 0.50, 0.0)

Mas para uma luz azul La = (0, 0, 1)

⇒La * C = (0, 0, 0) -> objeto preto

Alguns cuidados

- Como no OpenGL a cor de um objeto é definida por 3 componentes, o objeto pode ter cores diferentes para (Ca, Cd, Cs),
- Note que a cor resultante é definida também pela fonte de luz, e não apenas a cor do objeto.
- No OpenGL, você especifica quanto de cada fonte é refletida por um objeto. Parece estranho que um objeto possa refletir mais luz ambiente vermelha que especular ou difusa, mas é apenas um modelo.
  - Em geral, definimos as componentes difusa e especular como iguais.

Vetores relevantes para sombreamento

- As sombras são funções da relação entre o observador, as fontes de luz, e os objetos.
- Os vetores definidos a seguir podem ser considerados como centrados no ponto sobre o qual é feito o cálculo do sombreamento.
  - **normal vector (n):** perpendicular e para fora da superfície.
  - **view vector (v):** aponta para o observador
  - **light vector (l):** aponta para a fonte de luz
  - **reflection vector (r):** indica a direção de reflexão
  - **halfway vector (h):** vetor entre l e v.

Vetores

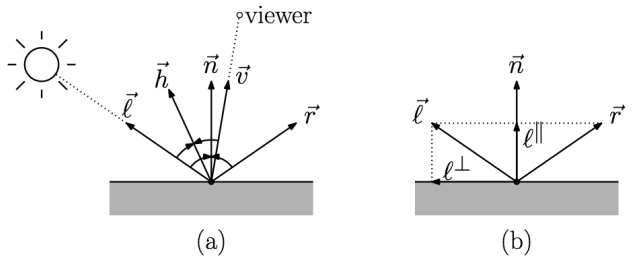


Fig. 42: Vectors used in Phong Shading.

Equações de iluminação

- Quase nenhum objeto é puramente difuso ou puramente especular.
- O modelo de iluminação de Phong assume que é possível modelar qualquer superfície (sem textura) através da combinação dessas duas componentes, e a componente ambiente.
  - Nesse modelo, quando a energia luminosa bate em um ponto da superfície, uma porcentagem ka, kd, e ks dessa energia é refletida respectivamente nas componetes da luz ambiente, de difusão e especular.
  - Se La, Ld, e Ls forem as intensidades RGB da fonte de luz ambiente, de difusão, e especular. Em geral, Ls = Ld, e La é ajustada de forma a modelar a quantidade de luz devido a reflexões indiretas.

Cálculo de Vetores Normais

- **Vetor normal**: a partir de 3 pontos não colineares
  - sujeito a erros quando os pontos são quase colineares ou não exatamente coplanares
- **Normal por área**:
  - usa todos os pontos de um polígono
  - menos sujeito a erros
  - todos os pontos estão num plano ax + by + cz + d = 0.
  - (a, b, c) = vetor normal ao plano
- **Normal** a superfícies implícitas
- **Normal** a superfícies paramétricas

Luz ambiente

- Luz ambiente é a mais simples de ser modelada
- Seja:
  - C: cor do objeto
  - Ia: intensidade de luz ambiente refletida
  - ka: coeficiente de reflexão ambiente da superfície.
    - fração da luz ambiente que é refletida pela superfície
    - 0 <= ka <= 1
  - então: Ia = ka . La . C

Componente difusa

- Reflexão difusa: a luz proveniente de qualquer direção é refletida uniformemente em todas as direções.
  - motivo: micro irregularidades na superfície
  - conhecido como refletor lamberciano.
- A quantidade de luz refletida em um ponto vai depender da posição desse ponto com relação a fonte de luz (vetor de luz l), e a normal (n) a superfície naquele ponto (lei dos cosenos de Lambert)
- Seja kd o coeficiente de reflexão de difusão da superfície, então Id, a componente difusa, é dado por:
  - Id = kd max(0, n.l) Ld C

Componente especular

- Superfícies metálicas lisas ou muito polidas se tornam altamente refletoras, exibindo um brilho ou pontos brilhantes caracteristicos.
  - para modelar esse fenomeno, o modelo de Phong utiliza o vetor de reflexão r.
  - O OpenGL utiliza o halfway vector, por ser um pouco mais eficiente, e gerar os mesmos resultados
    - observe que n.h é máximo quando o observador é colocado na direção de r (direção de reflexão), e assim, n.h pode ser usado para modelar a intensidade da componente especular.

Componente especular

- Seja:
  - ks o coeficiente de reflexão especular, e
  - α o brilho (shininess).
  - Quando α (> 1) aumenta, o ângulo de reflexão diminue, assim como o tamanho do ponto brilhante correspondente. O brilho pode ter valores bem grandes para superfícies altamente especulares.
  - Is : componente especular
  - Is = ks [max (0, n.h) ^ α] Ls
- Obs: Id e Is estão sujeitos a atenuação, pois dependem da distância do objeto a fonte de luz.

Conclusão

Combinando todas as componentes definidas até agora com Ie (luz emitida pela fonte), a luz total refletida de um ponto sobre um objeto de cor C, iluminado por uma fonte L, e distante d da fonte de luz, temos que:

I = Ie + Ia + (Id + Is) / (a + bd + cd^2)

Obs: para múltiplas fontes de luz, some as intensidades de cada fonte.

Iluminação no OpenGL

```
Glfloat ambient[4] = {0.5,0.5,0.5,1.0}
Glfloat diffuse[4] = {1.0, 0.0, 0.0, 1.0}
Glfloat position[4] = {2.0, 4.0, 5.0, 1.0}

glEnable(GL_LIGHTING);
glEnable(GL_LIGHT0);

glLightfv(GL_LIGHT0, GL_AMBIENT, ambient);
glLightfv(GL_LIGHT0, GL_DIFFUSE, diffuse);
glLightfv(GL_LIGHT0, GL_POSITION, position);
```

Eliminação de superfícies escodidas

```
glutInitDisplayMode(GLUT_DEPTH | ... );
glEnable(GL_DEPTH_TEST);
```

Para desenhar:

```
glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
```

desenhar em qualquer ordem, definindo a normal para cada vértice

Shade model: exemplo de init

```
GLfloat mat_specular[] = { 1.0, 1.0, 1.0, 1.0 };
GLfloat mat_shininess[] = { 50.0 };
GLfloat light_position[] = { 0.0, 0.0, 1.0, 0.0 };

glClearColor (0.0, 0.0, 0.0, 0.0);
glShadeModel (GL_SMOOTH);

glMaterialfv(GL_FRONT, GL_SPECULAR, mat_specular);
glMaterialfv(GL_FRONT, GL_SHININESS, mat_shininess);

glLightfv(GL_LIGHT0, GL_POSITION, light_position);
glEnable(GL_LIGHTING);   glEnable(GL_LIGHT0);
glEnable(GL_DEPTH_TEST);
```

E agora?

- Faça o EP2, você vai entender melhor o modelo de iluminação e sombreamento do OpenGL
- Com isso você vai aprender:
  - Perspectiva no OpenGL: glLookAt e gluPerspective
  - Luz e Cor
    - luz ambiente, especular, difusão e emissão
    - atenuação
  - Modelo de Iluminação de Phong
  - Vetores de iluminação e normais