

Métodos para Renderização de Superfície

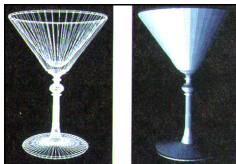
Profa. Ana Luísa D. Martins Lemos

June 13, 2018

- Baseado no modelo de iluminação, um método de renderização de superfície é usado para determinar a cor dos pixels
- O modelo de iluminação pode ser usado de formas diferentes para definir a cor de uma superfície
 - Ray-tracing: executado em cada pixel projetado (realismo)
 - Scan-line: executado em alguns pixels e interpolado no restante (tempo-real)

- A maioria das APIs gráficas reduz o processamento usando algoritmos de scan-line
 - As intensidades são calculadas nos vértices e interpoladas nas posições restantes dos polígonos

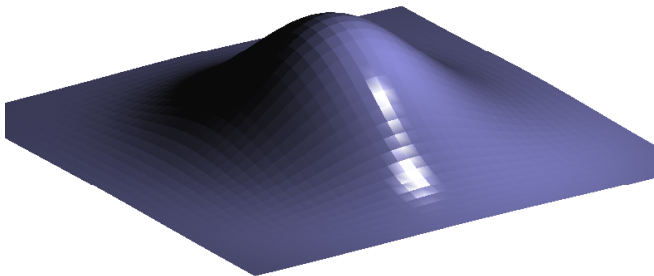
- O método mais simples para renderizar uma superfície é usar a mesma cor para todos os seus pontos (**flat surface rendering**)
- Emprega-se o modelo de iluminação para determinar a intensidade das 3 componentes RGB em uma única posição da superfície
 - Vértice ou centróide do polígono



- O **flat surface rendering** normalmente define resultados precisos se
 - O polígono é uma face de um poliedro e não uma seção de uma superfície curva
 - Todas as fontes de luz estão distantes o suficiente da superfície de forma que $N.L$ e a função de atenuação são constantes
 - A posição de visão é distante o suficiente do polígono de forma que $V.L$ é constante
- Mesmo se alguma dessas condições for falsa, uma boa aproximação pode ser conseguida se os polígonos empregados forem pequenos

Métodos de Renderização de Superfície

Intensidade Constante



Métodos de Renderização de Superfície

Gouraud surface rendering



- O esquema de **Gouraud surface rendering** interpola linearmente as intensidades nos vértices por toda face do polígono de um objeto iluminado
- Desenvolvido para aproximar superfícies curvas, amenizando as transições de intensidade entre polígonos adjacentes
 - Elimina as descontinuidades de intensidades do **flat surface rendering**

Métodos de Renderização de Superfície

Gouraud surface rendering

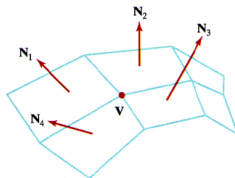


- Cada polígono de uma superfície é processado usando o seguinte procedimento
 - 1 Determina o vetor unitário normal médio em cada vértice do polígono
 - 2 Aplica o modelo de iluminação em cada vértice para obter as intensidades
 - 3 Interpola linearmente as intensidades dos vértices sobre a área projetada do polígono

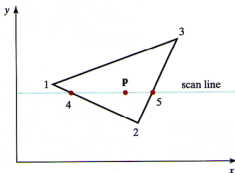
- O vetor normal médio em um vértice é obtido fazendo a média das normais de todos os polígonos que compartilham esse vértice

$$N_V = \frac{\sum_{k=1}^n N_k}{|\sum_{k=1}^n N_k|}$$

- Usando essas normais o modelo de iluminação é então executado para calcular as intensidades em cada vértice



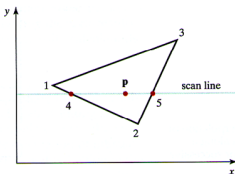
- Esses valores de intensidade são então interpolados para se obter as intensidades ao longo de *scan-lines* que intersectam a área projetada do polígono



- As intensidades das interseções das scan-lines com as arestas dos polígonos são calculadas interpolando linearmente as intensidades dos pontos finais das arestas

- Por exemplo, a intensidade em 4 pode ser calculada considerando somente o deslocamento vertical da scan-line

$$I_4 = \frac{y_4 - y_2}{y_1 - y_2} I_1 + \frac{y_1 - y_4}{y_1 - y_2} I_2$$



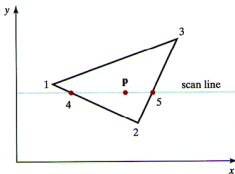
- A intensidade em 5 pode ser obtida da mesma forma interpolando verticalmente as intensidades em 2 e 3

Métodos de Renderização de Superfície

Gouraud surface rendering

- Considerando as intensidades obtidas em 4 e 5, as intensidades em qualquer ponto p da scan-line são obtidas interpolando na horizontal

$$I_p = \frac{x_5 - x_p}{x_5 - x_4} I_4 + \frac{x_p - x_4}{x_5 - x_4} I_5$$



- Esse método é conhecido como interpolação bilinear e é executado para os 3 componentes RGB separadamente

Métodos de Renderização de Superfície

Gouraud surface rendering

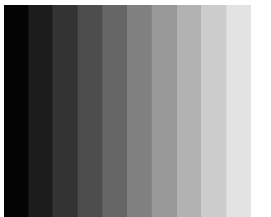
- Essa interpolação de intensidades elimina descontinuidades mas tem alguns problemas
 - Brilhos na superfície podem apresentar formatos estranhos
 - Intensidades claras ou escuras podem parecer “riscadas” (**mach bands**)



Métodos de Renderização de Superfície

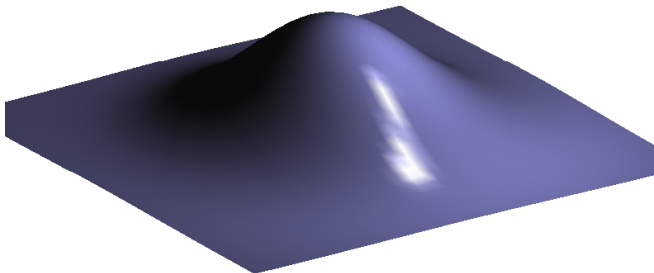
Gouraud surface rendering

- Efeito de **mach bands** consiste em faixas claras ou escuras que são percebidas próximo das fronteiras entre duas regiões de diferentes gradientes de luz



Métodos de Renderização de Superfície

Gouraud surface rendering



Métodos de Renderização de Superfície

Phong surface rendering



- Um método mais preciso de interpolação é conhecido como **Phong surface rendering**
- Ao invés de interpolar valores de intensidade, normais são interpoladas
 - Cálculos mais precisos de intensidades
 - Brilhos mais realísticos nas superfícies
 - Redução do efeito *mach-band*
- Computacionalmente mais caro que o método de Gouraud

Métodos de Renderização de Superfície

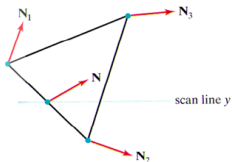
Phong surface rendering



- Cada polígono é processado usando o seguinte procedimento
 - 1 Determina o vetor unitário normal médio em cada vértice do polígono
 - 2 Interpola linearmente as normais dos vértices sobre a área projetada do polígono
 - 3 Aplica o modelo de iluminação nas posições ao longo da scan-line para calcular as intensidades dos pixels usando as normais interpoladas

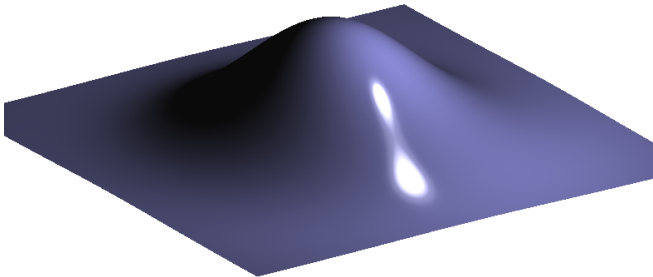
- O procedimento de interpolação das normais é o mesmo da interpolação das intensidades do método de Gouraud
- Por exemplo, o vetor normal N é verticalmente interpolado a partir das normais nos vértices 1 e 2 fazendo

$$N = \frac{y - y_2}{y_1 - y_2} N_1 + \frac{y_1 - y}{y_1 - y_2} N_2$$



Métodos de Renderização de Superfície

Phong surface rendering



- Podemos usar dois métodos de renderização de superfície: (1) de intensidade constante (*flat*); (2) modelo de Gouraud (*smooth*)
 - Não existe suporte para o modelo de Phong
- Para definir o método de renderização usamos

```
glShadeModel(rendering_method)
```

- Onde `rendering_method` pode ser `GL_FLAT` e `GL_SMOOTH`

- Para se definir a normal usamos

```
glNormal3*(Nx, Ny, Nz);
```

- Com o sufixo dependendo do tipo de parâmetro b, s, i, f e d, ou com a adição de v caso o parâmetro seja um vetor
- A normal é um valor de estado do OpenGL e tem valor padrão igual a (0.0, 0.0, 1.0)

- Para renderização de intensidade constante definimos apenas uma normal para cada polígono

```
glNormal3fv(normal_vector);  
glBegin(GL_TRIANGLES);  
    glVertex3fv(vertex1);  
    glVertex3fv(vertex2);  
    glVertex3fv(vertex3);  
glEnd();
```

- Para renderização de Gouraud uma normal deve ser definida para cada vértice

```
glBegin(GL_TRIANGLES);  
    glNormal3fv(normal_vector1);  
    glVertex3fv(vertex1);  
  
    glNormal3fv(normal_vector2);  
    glVertex3fv(vertex2);  
  
    glNormal3fv(normal_vector3);  
    glVertex3fv(vertex3);  
glEnd();
```

- Apesar dos vetores normais não precisarem ser especificados com tamanho unitário, fazendo isso reduzimos o custo computacional
- É possível solicitar ao OpenGL que qualquer vetor normal que não seja unitário seja normalizado chamando

```
glEnable(GL_NORMALIZE);
```

- Esse comando renormaliza todas as normais às superfícies incluindo as que foram modificadas por transformações geométricas de escala e cisalhamento