

# Modelos de tonalização: Flat, Gouraud e Phong

Guilherme Henrique de Souza Nakahata

Universidade Estadual do Paraná - Unespar

04 de Julho de 2024

- Iteração das fontes de luz;
- Definindo cor de cada pixel (rasterização);
- Tonalização Flat;
- Tonalização Gouraud;
- Tonalização Phong;

**Iluminação Phong  $\neq$  Tonalização Phong**

Não confundir com o modelo de iluminação de Phong.

- Por polígono;
- Interpolação por vértices;
- Interpolação dos vetores normais.

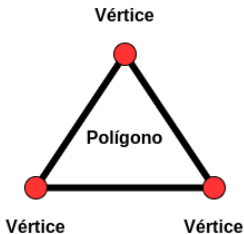


Figure: Exemplo entre Vértice e Polígono.

- Calculada uma única vez por polígono;
- Menor custo computacional;
- Visualizações fotorrealísticos;
- Aparenta as faces dos polígonos da malha.

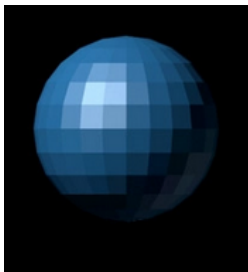


Figure: Sombreamento flat.

- Henri Gouraud (1971);
- Subconjunto de pontos;
- Interpolação;
- Vértices dos polígonos.

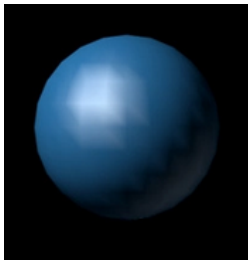


Figure: Sombreamento Gouraud.

- Intensity interpolation shading;
- Color interpolation shading;
- Interpola a intensidade da luz;
- Descontinuidade de cores (Bandas de Mach).



Figure: Bandas de Mach.

Fonte: RATLIFF, Floyd (1965)

- Projeção em perspectiva;
- Reflexões especulares;
- Equação incremental (z-Buffer);

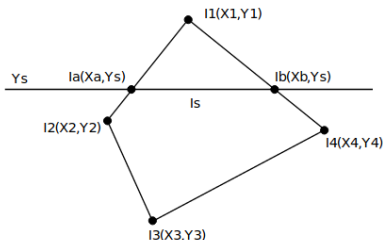


Figure: Cálculo da intensidade da luz refletida.

Fonte: XICHUN, GUO (1996)

- Bui-Tuong Phong (1975);
- Interpolação dos vetores normais;
- Normal para cada face;
- Aplicado a todos os pontos da superfície.

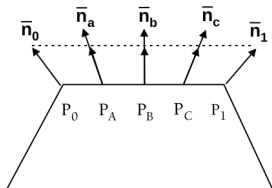


Figure: Interpolação dos ângulos no modelo Phong.

Fonte: FOLEY (1993)





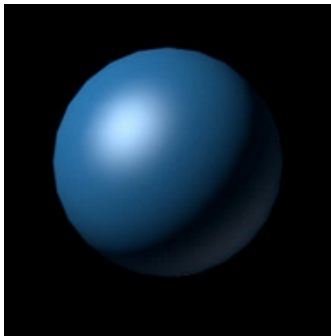


Figure: Sombreamento Phong.

## Tipos de iluminação

- Luz ambiente;
- Luz direcional;
- Luz pontual;
- Luz Holofote.



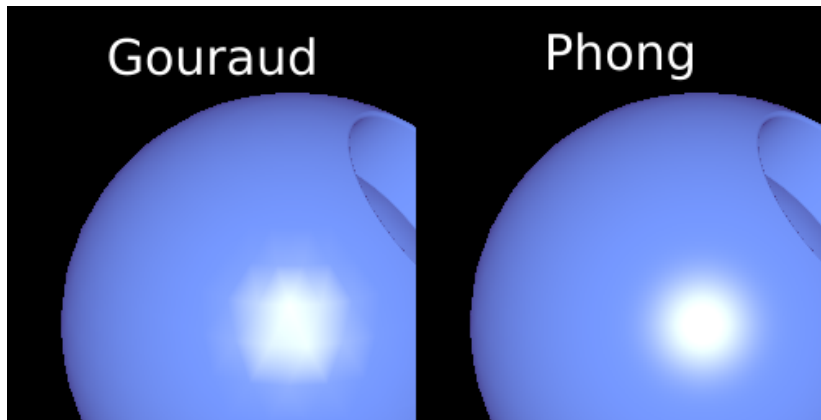


Figure: Comparação entre Gouraud e Phong.

Quais os modelos de tonalização vistos hoje?

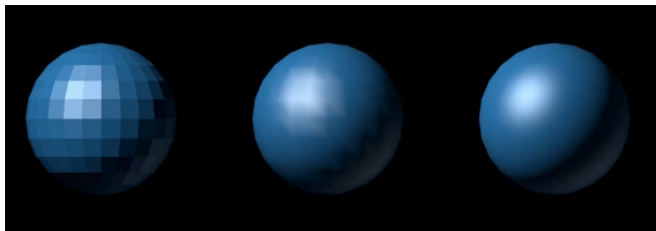


Figure: Modelos de tonalização.

- Mais polígonos melhor resolução?
- Qual o melhor modelo de tonalização?

- Mais polígonos melhor resolução?
- Qual o melhor modelo de tonalização?

## Modelo de tonalização

- Custo computacional;
- Aplicação;
- Tempo de atualização;
- Objetivo final.



- Uma imagem ou padrão aplicado a uma superfície tridimensional;
- Adicionar detalhes e realismo aos objetos renderizados em um ambiente gráfico;
- Mapeada em uma superfície 3D usando coordenadas de textura;
- Cada ponto na superfície do objeto tem um conjunto correspondente de coordenadas de textura;
- Determina qual parte da imagem da textura será aplicada a ele;
- As coordenadas de textura podem variar de 0 a 1;
- Usadas para determinar a posição na textura onde o ponto da superfície será mapeado;

- As texturas podem ser usadas para simular uma variedade de características visuais como:
- Cor;
- Rugosidade;
- Reflexividade;
- Transparências;
- Características complexas (aparência de pele ou superfícies de terreno).

- Elas podem ser aplicadas a objetos em uma ampla variedade de aplicações:
  - Jogos de vídeo;
  - Animações;
  - Simulações;
  - Renderização de filmes.
- As texturas podem ser criadas digitalmente ou podem ser fotografias reais que foram capturadas e mapeadas em objetos 3D;
- Há uma grande variedade de técnicas e algoritmos para criar e manipular texturas;
- Permitindo aos artistas e desenvolvedores alcançar uma ampla gama de efeitos visuais realistas.

- Mapeamento de textura:
  - Aplica uma imagem a uma superfície 3D usando coordenadas de textura.
- Texturização procedural: Texturas sintéticas.
  - Criadas usando funções matemáticas e procedimentos que descrevem características visuais;
  - Padrões de repetição, rugosidade, granulação e distorção.

- Mapeamento de relevo:
  - Usa texturas para simular a aparência de detalhes em relevo;
- Texturização baseada em procedimentos físicos:
  - Essa abordagem utiliza equações físicas e propriedades de materiais reais;
  - Cria texturas que simulam a aparência de materiais específicos, como metal, madeira, pele, entre outros.

```
1 void display() {  
2     glClear(GL_COLOR_BUFFER_BIT);  
3  
4     glBindTexture(GL_TEXTURE_2D, textureID);  
5  
6     glBegin(GL_QUADS);  
7         glTexCoord2f(0.0, 0.0); glVertex2f(-0.5, -0.5);  
8         glTexCoord2f(1.0, 0.0); glVertex2f(0.5, -0.5);  
9         glTexCoord2f(1.0, 1.0); glVertex2f(0.5, 0.5);  
10        glTexCoord2f(0.0, 1.0); glVertex2f(-0.5, 0.5);  
11    glEnd();  
12  
13    glFlush();  
14 }
```

# Obrigado! Dúvidas?

Guilherme Henrique de Souza Nakahata

[guilhermenakahata@gmail.com](mailto:guilhermenakahata@gmail.com)

<https://github.com/GuilhermeNakahata/UNESPAR-2024>