Shading (sombreamento) & *Smooth Shading*

Sistemas Gráficos/

Computação Gráfica e Interfaces

Shading & Smooth Shading

Objectivo: calcular a cor de cada ponto das superfícies visíveis.

Solução ***brute-force***: calcular a normal em cada ponto e aplicar o modelo de iluminação pretendido.

Modelos para colorir superfícies definidas por malha poligonal:

1. Sombreamento Constante
2. Sombreamento Interpolado = *Smooth Shading*
    1. Algoritmo de Gouraud
    2. Algoritmo de Phong

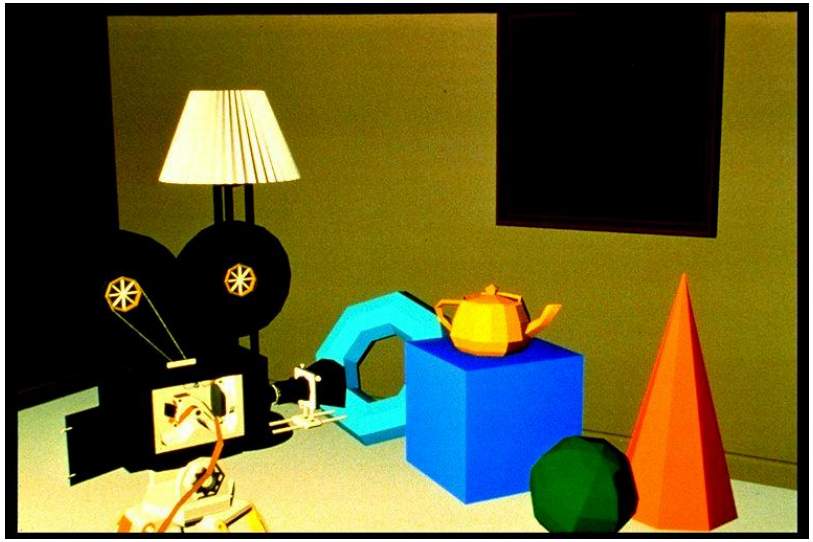
Shading

Sombreamento Constante

A cor é calculada apenas para um ponto do polígono e replicada em todos os pontos restantes do mesmo polígono.

Esta técnica considera as seguintes condições:

- A fonte de luz está no infinito, de modo que N.L é constante em qualquer ponto do polígono (raios paralelos).
- O observador está no infinito, de modo que R.V é constante em qualquer ponto do polígono
- A face é a própria superfície plana a modelar e não é uma aproximação de uma superfície curva



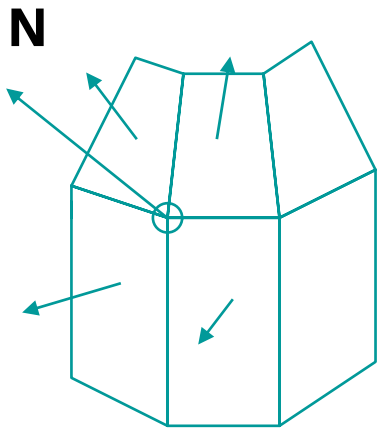
Shading

Sombreamento Interpolado ou *Smooth Shading*

Na solução anterior, se aproximarmos uma superfície curva por uma malha poligonal, verificamos descontinuidade de cor entre polígonos adjacentes (efeito de Mach Band, com descontinuidade da função de iluminação).

As soluções apresentadas a seguir ultrapassam este problema determinando a cor de um ponto por interpolação da cor definida nos vértices do polígono.

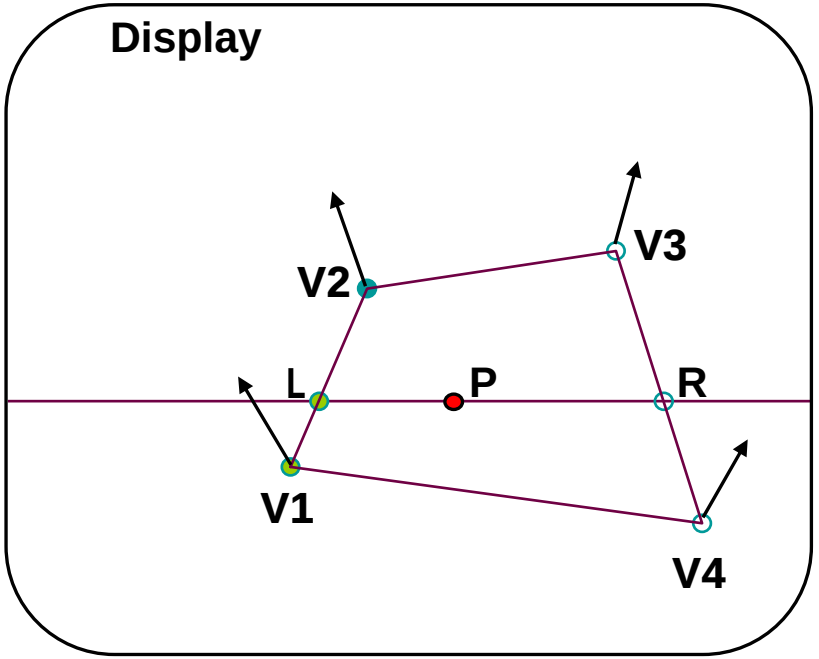
1. Necessário o conhecimento, nos vértices, das normais à superfície curva original:
   - Expressão analítica da superfície…
2. Aproximação possível:
   - Interpolação das normais dos polígonos vizinhos.



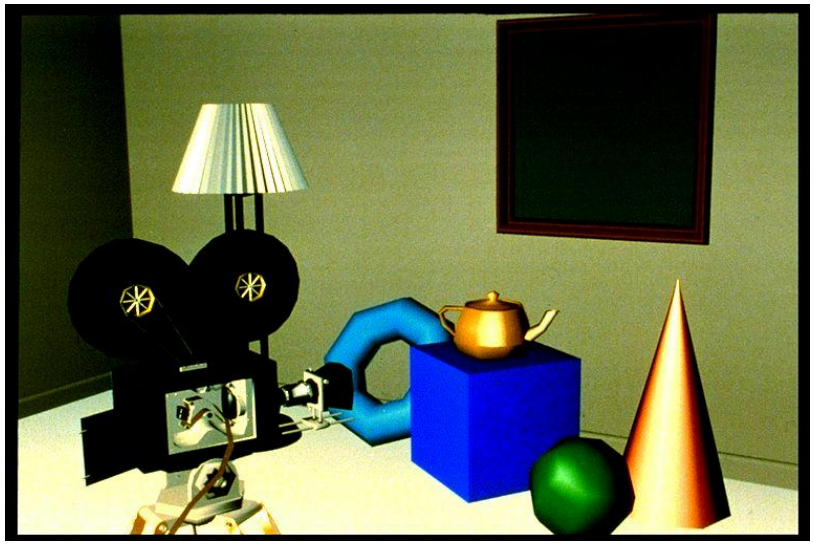
Smooth Shading

Método de Gouraud

2. Calcular a cor de cada vértice através do modelo de iluminação pretendido.
3. Calcular a cor dos restantes pontos do polígono por interpolação bi-linear.



1. Cor do ponto **L** é obtida por interpolação da cor em **V1** e **V2**
2. **R =** interpolação de **V3** e **V4**
3. **P =** interpolação de **L** e **R**

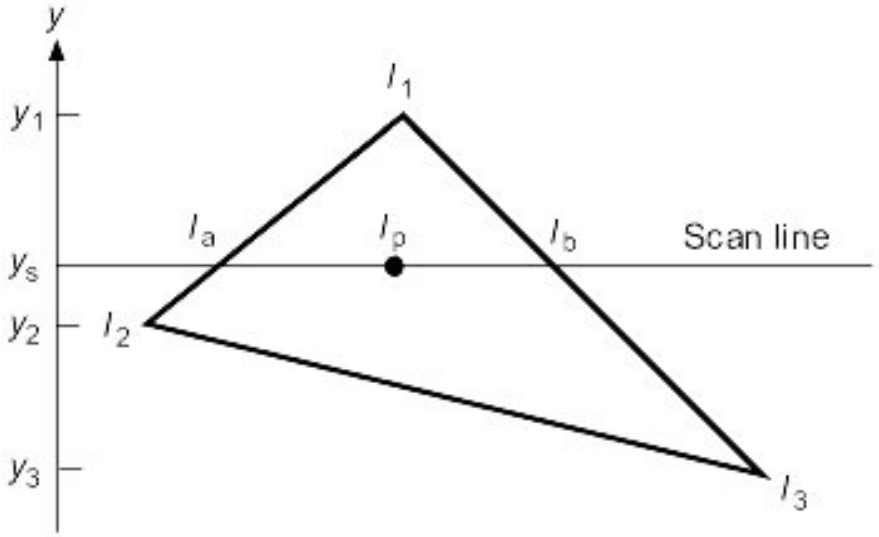


Nota-se a localização das arestas (efeito de Mach Band, com descontinuidade da derivada da função de iluminação)

Smooth Shading

Método de Gouraud

Calculo dos valores interpolados



$$I_a = I_1 - (I_1 - I_2)\frac{y_1 - y_s}{y_1 - y_2}$$

$$I_b = I_1 - (I_1 - I_3)\frac{y_1 - y_s}{y_1 - y_3}$$

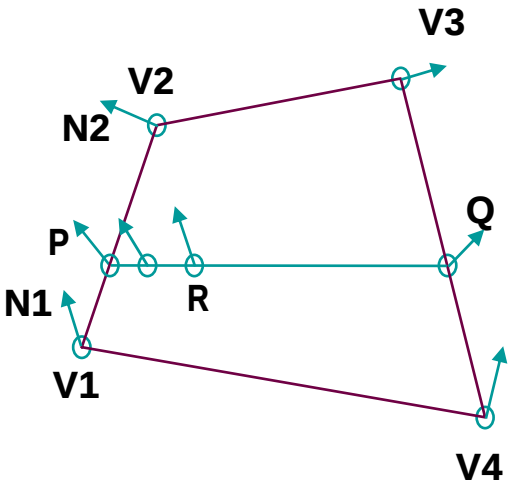
$$I_p = I_b - (I_b - I_a)\frac{x_b - x_p}{x_b - x_a}$$

Smooth Shading

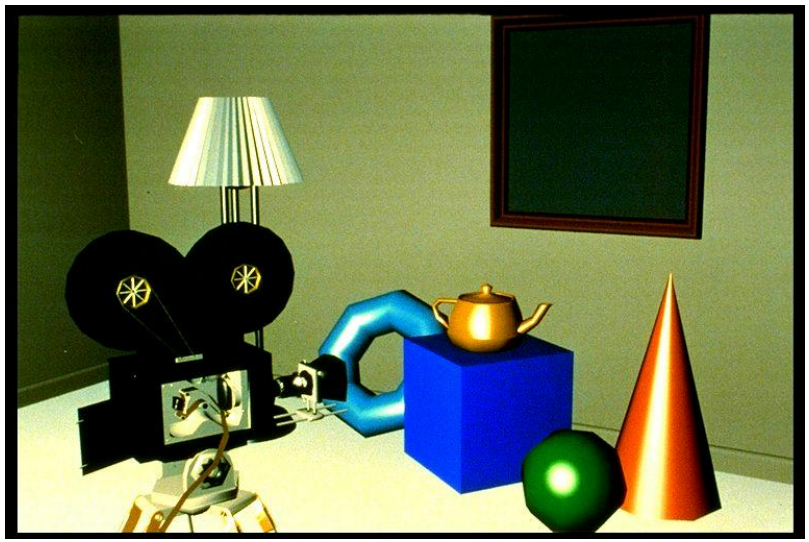
Método de Phong

Efectua a interpolação das normais em vez da cor.

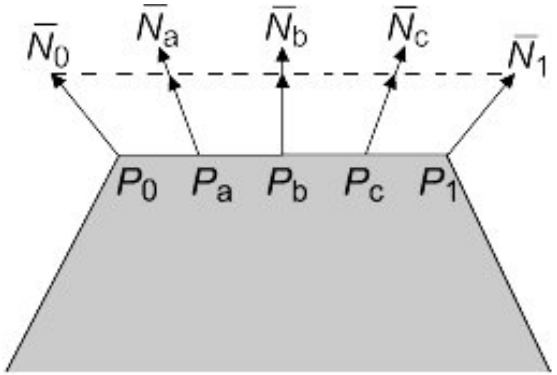
1. Para cada vértice da malha poligonal calcular o vector normal à superfície. Podem ser calculadas pela expressão analítica da superfície ou interpolando a normal dos polígonos vizinhos.
2. As normais nas arestas são calculadas através das normais nos vértices. As normais nos restantes pontos usam os pontos das arestas na mesma linha de varrimento.
3. O modelo de iluminação é aplicado em cada ponto.



1. Normal em **P** obtida por interpolação das normais em **V1** e em **V2.**
2. **Q =** interpolação de **V3** e **V4**
3. **R =** interpolação de **P** e **Q**

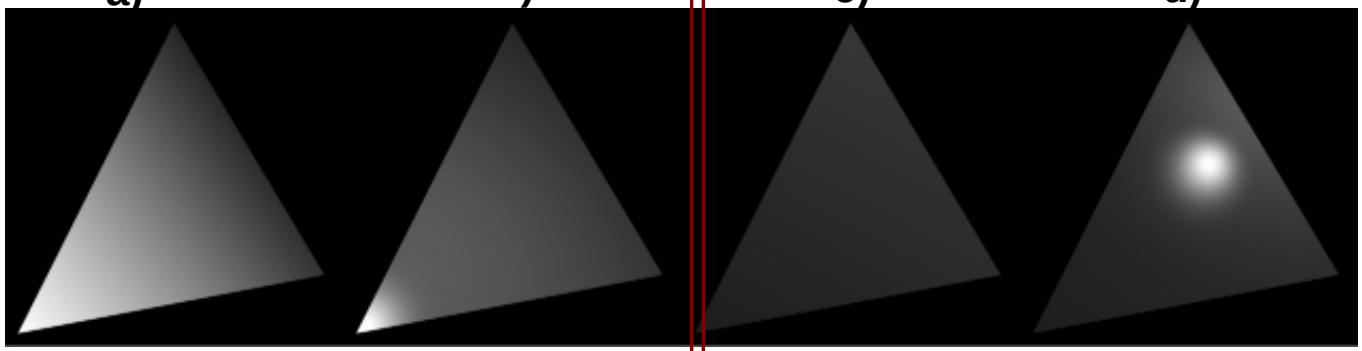


Smooth Shading



O cálculo da iluminação em cada *pixel* exige o mapeamento inverso para coordenadas do objecto depois de determinada a normal.

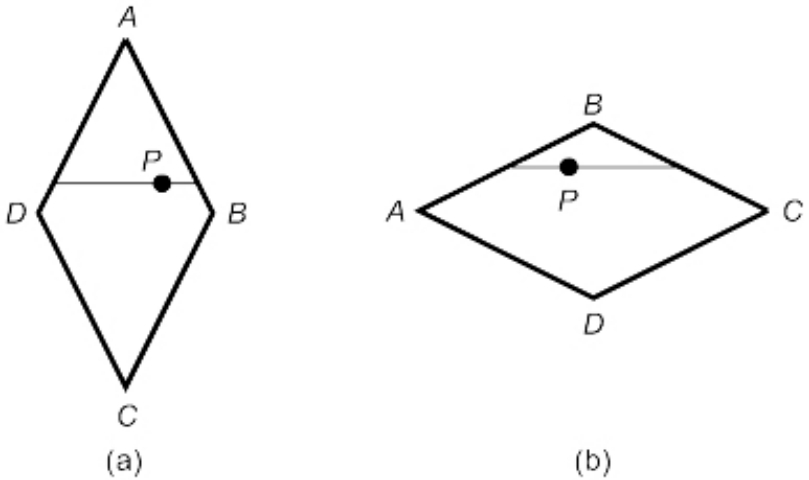
Reflexão especular com sombreamento pelo modelo de Gouraud a) e c) e Phong b) e d)



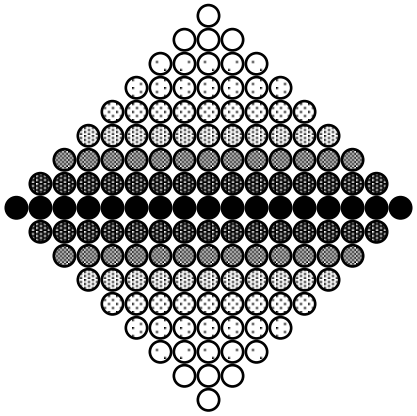
Reflexão máxima no vértice esquerdo.

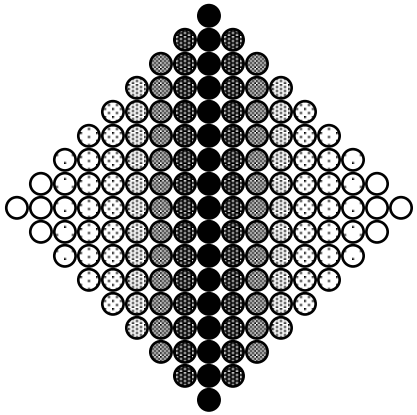
Reflexão máxima no centro do polígono.

Problema do Sombreamento Interpolado

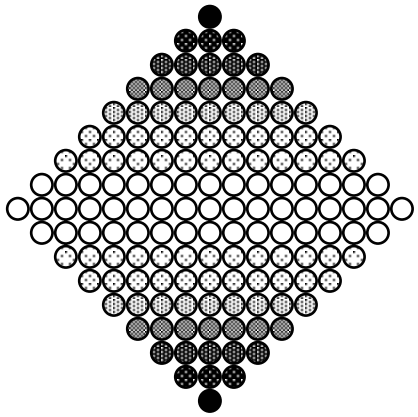


O resultado depende da orientação do polígono.
Em (a) o cálculo de P usa as cores dos vértices A,D,B.
Em (b) o cálculo de P usa as cores dos vértices A,B,C.





Rotação de 90º



Resultado

Texturas

Sistemas Gráficos/

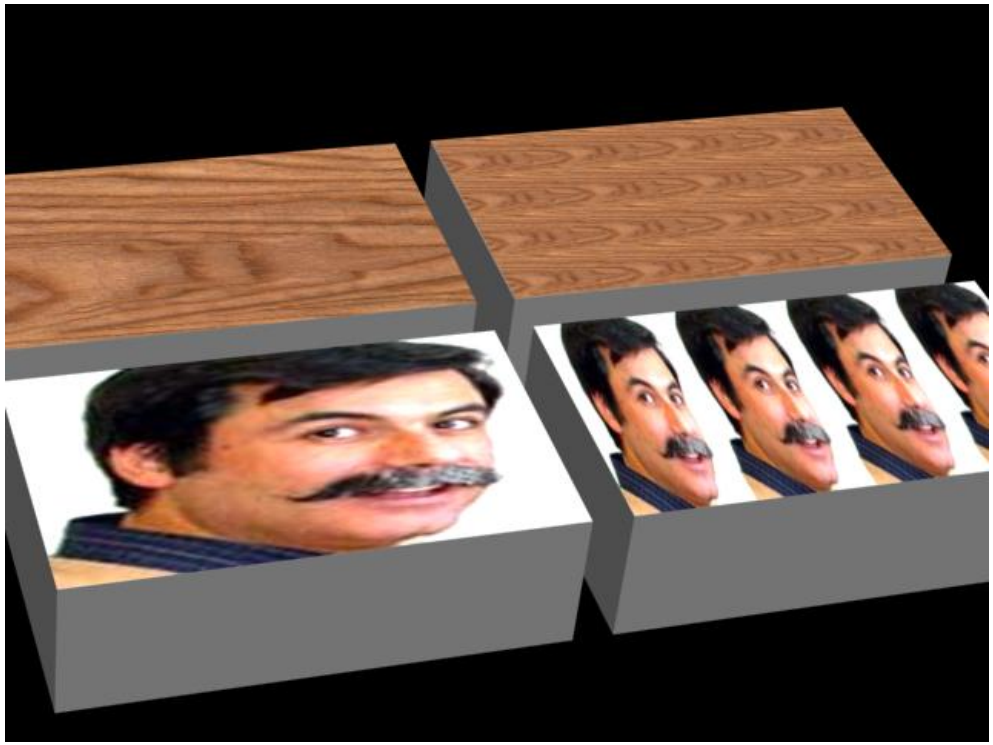
Computação Gráfica e Interfaces

Texturas

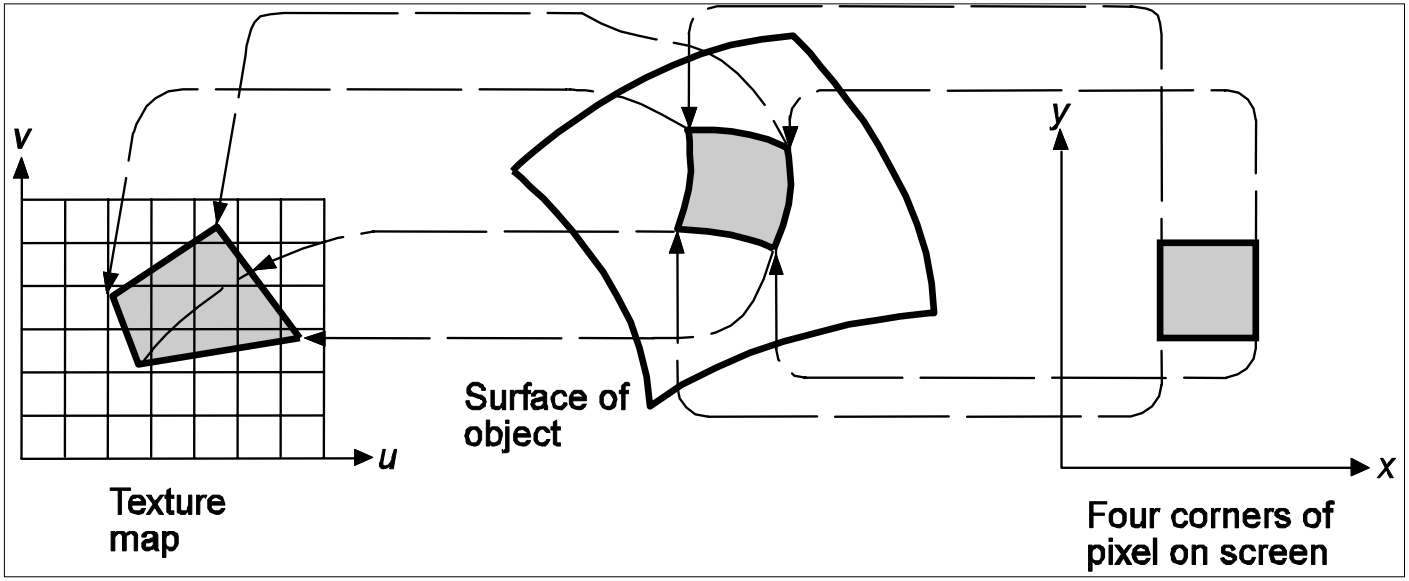
- Permitem obter detalhe visual sem aumentar o detalhe geométrico
- Tipos mais vulgares
  - Mapeamento de texturas (imagens 2D)
    - Uma imagem “colada” sobre um polígono (papel de parede)
      - Representação de uma pintura num quadro
      - Simulação de uma paisagem fora de uma janela
      - Superfície de madeira
  - *Bump Mapping Textures*
    - Além da imagem 2D, cria-se sensação de relevo (rugosidade)
      - Casca de laranja
      - Casca de morango
      - Tijolos
  - Texturas 3D
    - A textura evolui continuamente no “interior” dos objectos
      - Volume de Madeira
      - Volume de Mármore

Mapeamento de Texturas (2D)

- *Pixels* da Imagem denominam-se "*texels*"
- Imagem de textura tem coordenadas (u,v) Є [0,1]



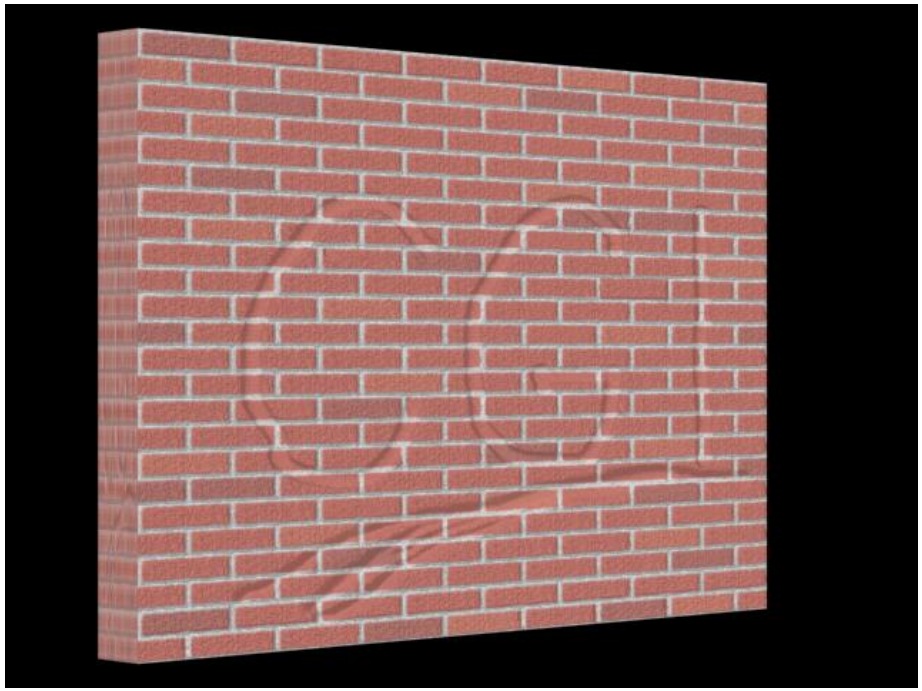
Mapeamento de Texturas (2D)



- Dois passos:
  - 4 cantos do *pixel* são mapeados na superfície (s,t)
  - 4 pontos (s,t) são mapeados no espaço da textura (u,v)
  - a cor resultante é extraída das cores dos *texels* incluídos na área resultante (filtragem)
    - Cor de um só texel... (maus resultados)
    - Média pesada das cores dos texels
    - Outras filtragens mais poderosas...

Bump Mapping Textures

- Simulação de rugosidade...
  ...sem aumento de geometria



Exemplo em 3DStudio MAX:

Imagem a mapear

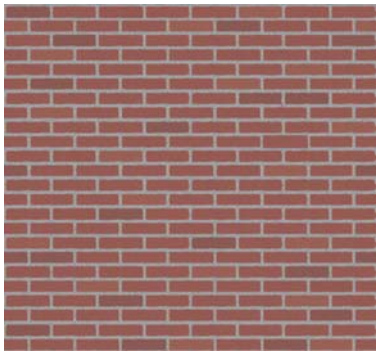
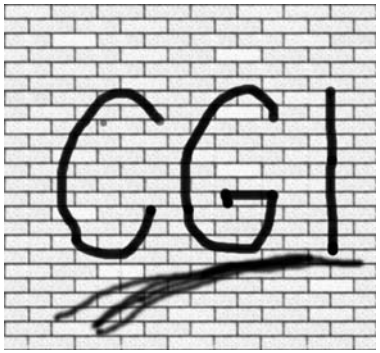
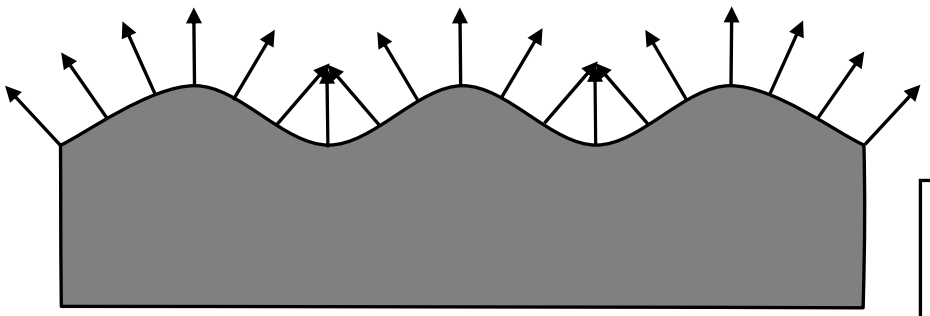


Imagem de rugosidade



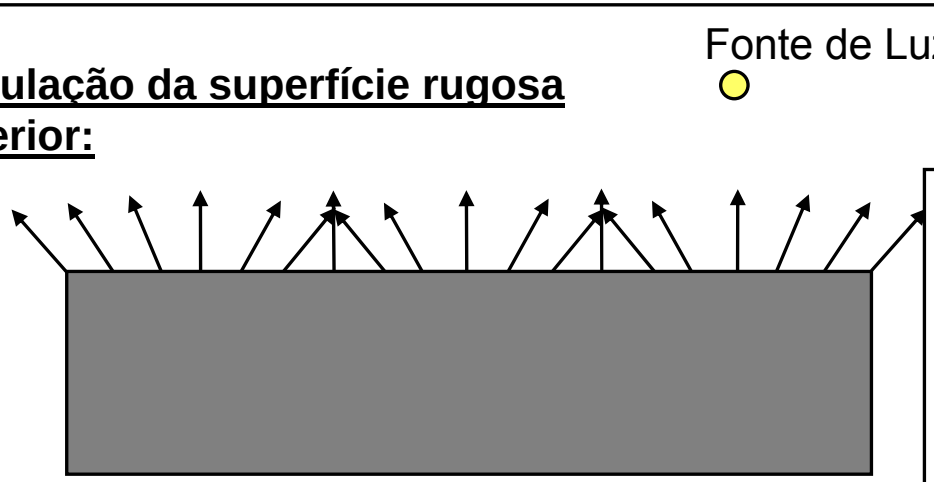
Bump Mapping Textures

Superfície rugosa "original"



A iluminação evolui, ponto a ponto, de acordo com a inclinação das normais respectivas

Simulação da superfície rugosa anterior:



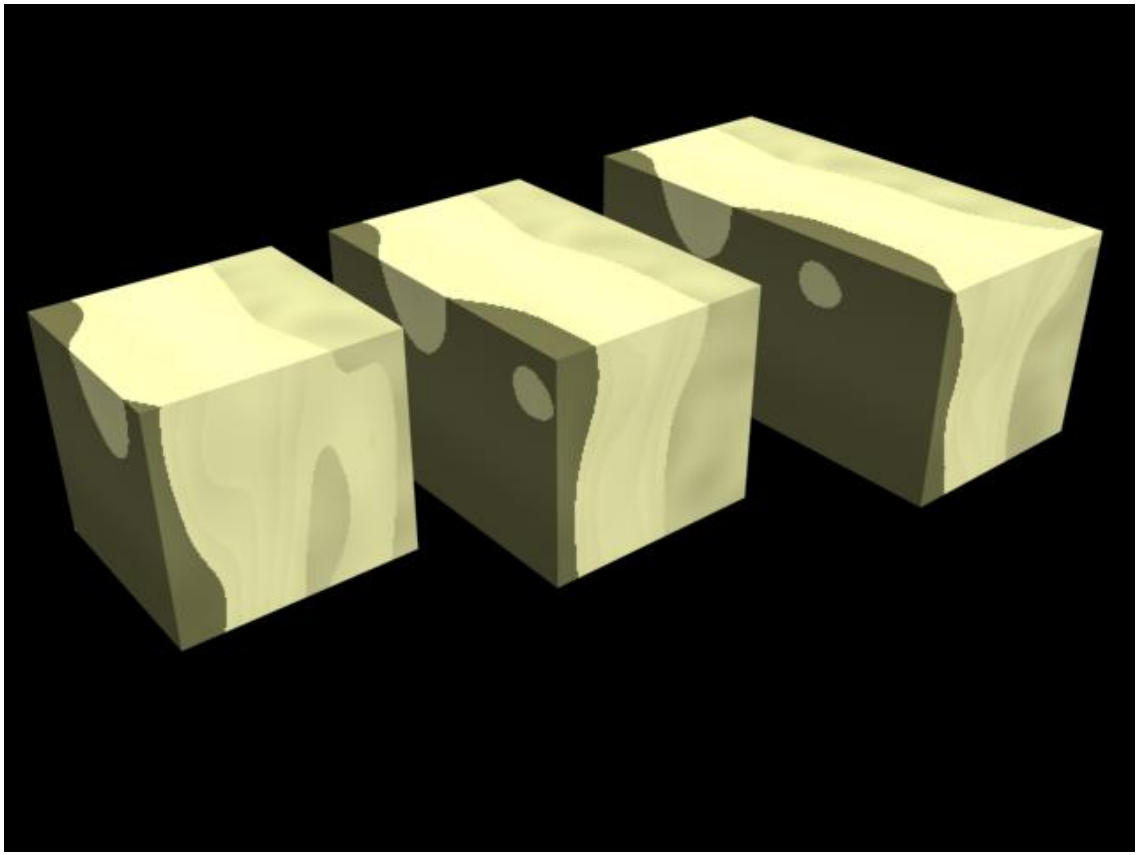
Afecta-se a direcção da normal (cálculo da iluminação)

Resultado semelhante ao anterior...

MAS COM GEOMETRIA SIMPLES!

Texturas 3D

– Evolução contínua no “interior” dos objectos



- Função devolve cor em função das coordenadas espaciais (***x,y,z***)