

Aula – Computação Gráfica

Textura

Slides para uso pessoal e exclusivo durante o período de aula. Distribuição ou qualquer uso fora do escopo da disciplina é expressamente proibido.

1

1

Por quê usar textura?

Limitações computacionais

- Modelos de iluminação são limitados
 - Não descreve superfícies pintadas
 - Por exemplo, um quadro em uma parede
 - Não descreve superfícies com alto grau de detalhes
 - Por exemplo, uma parede de tijolos ou um muro chapiscado
- A modelagem dos casos acima requer modelos complexos
 - Utilização de muitos polígonos e materiais
 - Custo computacional muito alto

Problemas

- Usar mapeamento de textura
 - Funciona como um papel de parede colado em uma superfície

2

2

Mapeamento de Textura

Objetivo

- Reproduzir as propriedades de uma função sobre um objeto 3D
 - A função pode ser uma imagem ou algum outro padrão



Drummylab
https://commons.wikimedia.org/wiki/File:Texture_mapping_demonstration_animation.gif
CC0

<https://commons.wikimedia.org/wiki/File:Texturedm1a2.png>
CC BY-SA 4.0

3

3

Mapeamento de Textura

Tipos de propriedades a serem mapeadas

- Cor



https://commons.wikimedia.org/wiki/File:Freedoom_2018.png

BSD

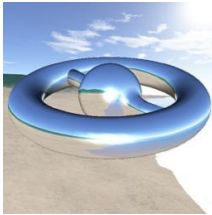
4

4

Mapeamento de Textura

Tipos de propriedades a serem mapeadas

- Coeficientes de reflexão



https://en.wikipedia.org/wiki/File:Cube_mapped_reflection_example_2.JPG

Public Domain



https://commons.wikimedia.org/wiki/File:Spoon_fi.jpg

CC BY-SA 3.0

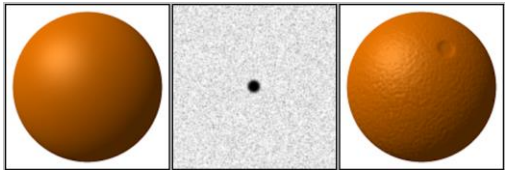
5

5

Mapeamento de Textura

Tipos de propriedades a serem mapeadas

- Perturbação do vetor normal (*bump mapping*)



<https://commons.wikimedia.org/wiki/File:Bump-map-demo-full.png>

CC BY-SA 3.0

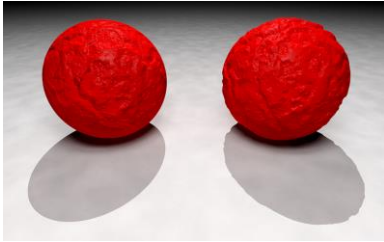
6

6

Mapeamento de Textura

Tipos de propriedades a serem mapeadas

- Perturbação do vetor normal (*bump mapping*)



https://commons.wikimedia.org/wiki/File:Bump_map_vs_isosurface2.png
Domínio público

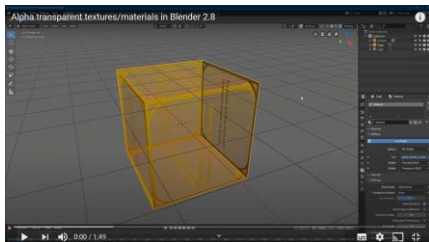
7

7

Mapeamento de Textura

Tipos de propriedades a serem mapeadas

- Transparência/Opacidade
 - <https://www.youtube.com/watch?v=z2cFVVotcs8>



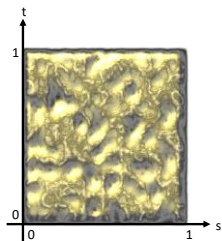
8

8

Mapeamento de Textura

Espaço de textura

- Texturas 2D são funções
 - $T(s, t)$
 - Saída pode ser
 - Cor, opacidade, ...
- Geralmente, operam
 - no intervalo entre 0 e 1
- A função pode ser dada por
 - Uma imagem $M \times N$
 - $T(s, t) = [s(M-1), t(N-1)]$



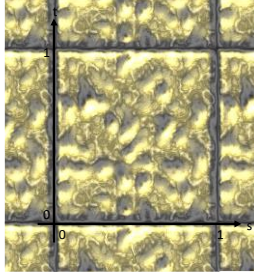
9

9

Mapeamento de Textura

Espaço de textura

- O padrão pode se repetir
 - Espaço infinito
 - $T(s, t) = [s(M-1) \bmod M, t(N-1) \bmod N]$



10

10

Função de Mapeamento

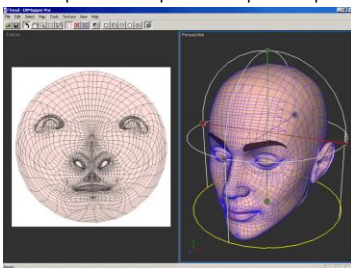
- Corresponde a forma como a textura embrulha o objeto
 - Como um papel texturizado embrulhando um presente
- Para cada ponto no espaço de textura
 - Retorna o ponto 3D o objeto de interesse
 - $(x, y, z) = F(s, t)$
- Problema
 - Geralmente, precisa-se desembulhar o objeto
 - Ou seja, mapear os pontos 3D para um ponto na textura
 - Corresponde a inversa da função de mapeamento
- Solução
 - Parametrização da superfície

11

11

Parametrização da Superfície

- Forma de mapear uma superfície 3D para um plano 2D



Antar03
https://commons.wikimedia.org/wiki/File:Texture_Mapping_example.png
 CC BY-SA 4.0

12

12

Parametrização da Superfície

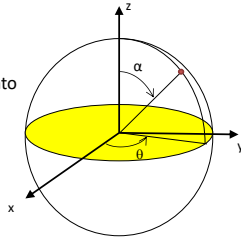
Parametrização de uma esfera

- Função de mapeamento

$$\begin{aligned}x &= \sin \alpha \cos \theta & \alpha &= \pi t \\y &= \sin \alpha \sin \theta & \theta &= 2\pi s \\z &= \cos \alpha\end{aligned}$$

- Inversa da função de mapeamento

$$\begin{aligned}\alpha &= \arccos z & t &= \frac{\arccos z}{\pi} \\ \theta &= \arctan \frac{y}{x} & s &= \frac{\arctan \frac{y}{x}}{2\pi}\end{aligned}$$



13

13

Parametrização da Superfície

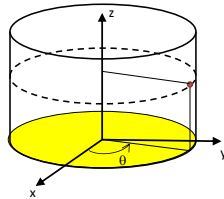
Parametrização de uma cilindro

- Função de mapeamento

$$\begin{aligned}x &= \cos \theta & \theta &= 2\pi s \\y &= \sin \theta \\z &= t\end{aligned}$$

- Inversa da função de mapeamento

$$\begin{aligned}\theta &= \arctan \frac{y}{x} & s &= \frac{\arctan \frac{y}{x}}{2\pi} \\ t &= z\end{aligned}$$



14

14

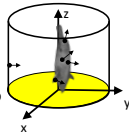
Parametrização da Superfície

Parametrização de um objeto genérico

- Pode não ter uma parametrização natural

- Pode-se usar um mapeamento de dois passos

- Usar uma superfície com mapeamento conhecido
 - Cilindro, esfera, etc.
- Colocar o objeto de interesse no interior da superfície escolhida
- Mapear a superfície simples para o objeto
 - Várias possibilidades
 - Raios do centroide a superfície simples
 - Raios das normais a superfície simples
 - Raios das normais da superfícies ao objeto
 - Entre outros



15

15

Parametrização da Superfície

Parametrização de um objeto genérico

- Exemplo

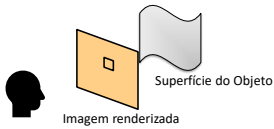


16

16

Processo de Mapeamento de Textura

Passo a passo



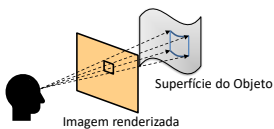
17

17

Processo de Mapeamento de Textura

Passo a passo

- Projeção do pixel de interesse na superfície
 - Saída: Vértices do pixel na superfície do objeto



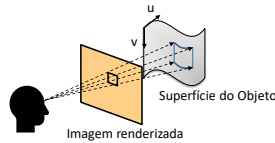
18

18

Processo de Mapeamento de Textura

Passo a passo

- Parametrização da superfície
 - Usa um mapeamento feito offline



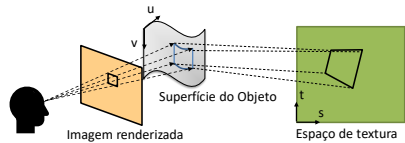
19

19

Processo de Mapeamento de Textura

Passo a passo

- Mapeamento inverso
 - Saída: Vértices da superfície do objeto no espaço de textura



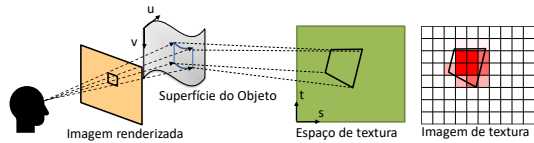
20

20

Processo de Mapeamento de Textura

Passo a passo

- Busca dos pixels (denominados "texels") na imagem de textura
 - Saída: Cor média dos texels considerando a área coberta por eles



21

21

Mapeamento de Textura em Polígonos

Coordenadas Baricentricas

- Maneira simples e rápida
- Projetar os vértices do triângulo 3D
 - Para os pontos P_1, P_2 e P_3 na imagem a ser renderizada
- Cada P_i projetado corresponde a um Q_i no espaço de textura
- Cada pixel P_j do triângulo representado por P_1, P_2 e P_3
 - É representado pelas coordenadas baricentricas
 - $P_j = \alpha_1 P_1 + \alpha_2 P_2 + \alpha_3 P_3$
- O triângulo é representado por Q_1, Q_2 e Q_3 no espaço de textura
- O pixel P_j é pintado com a cor do texel obtido com mesmos α
 - $Q_j = \alpha_1 Q_1 + \alpha_2 Q_2 + \alpha_3 Q_3$

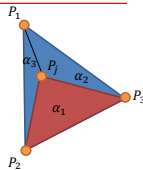
22

22

Mapeamento de Textura em Polígonos

Coordenadas Baricentricas

- Como obter os α ?
 - $P_j = \alpha_1 P_1 + \alpha_2 P_2 + \alpha_3 P_3$
- α_i representa a área do subtriângulo S
 - Normalizada pela área do triângulo T



$$\alpha_i = \frac{|detS|}{|detT|}$$

- Área de um triângulo M (vértices A, B e C) é calculada por $\frac{|detM|}{2}$

$$detM = \begin{vmatrix} A_x & A_y & 1 \\ B_x & B_y & 1 \\ C_x & C_y & 1 \end{vmatrix} = A_x B_y + A_y C_x + B_x C_y - B_y C_x - A_y B_x + A_x C_y$$

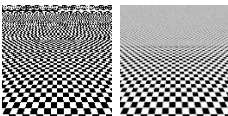
23

23

Problemas com Mapeamento de Textura

Problema: Aliasing

- Pixel da imagem renderizada é diferente do Texel
 - Implica em uma amostragem com frequência diferente
- Frequências mais altas do que a frequência de Nyquist
 - Introduzirão aliasing
- Polígonos em diferentes posições de visualização
 - Amostram diferentes frequências dos Texels

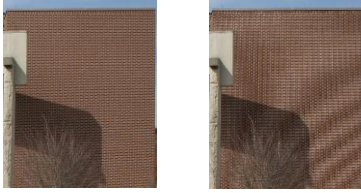


<https://commons.wikimedia.org/wiki/File:Aliased.png>
<https://en.wikipedia.org/wiki/File:Antialiased.png>
 CC BY-SA 3.0

24

24

Problemas com Mapeamento de Textura



https://commons.wikimedia.org/wiki/File:Moire_pattern_of_bricks.jpg
https://en.wikipedia.org/wiki/File:Moire_pattern_of_bricks_small.jpg
 CC BY-SA 3.0

Solução: Filtragem

- Eliminar frequências mais altas
- Pode ser feita previamente
 - Uso de mip-mapping

25

25

Problemas com Mapeamento de Textura

Problema: Deformação

- Combinações afins são distorcidas nas projeções perspectivas

Solução: Mais vértices ou uso de coordenadas homogêneas



Rainwarrior
https://commons.wikimedia.org/wiki/File:Perspective_correct_texture_mapping.jpg
 Public Domain

26

26

Mip-mapping

O que é?

- Sequências pré-calculadas e otimizadas de imagens
- Várias versões da mesma imagem com tamanhos diferentes



Mulad, based on a NASA image
https://commons.wikimedia.org/wiki/File:MipMap_Example_STS101.jpg
 CC BY-SA 3.0

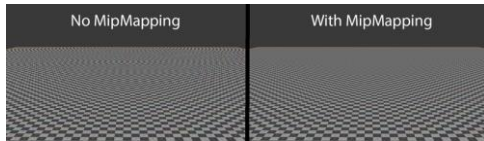
27

27

Mip-mapping

Vantagens

- Ganha no tempo de renderização
 - Imagens já estão pré-computadas e são carregadas mais fácil
- Melhora o nível de detalhes
 - Imagens são pré-computadas com algoritmos mais eficientes



[BilyBob CornCob
https://commons.wikimedia.org/wiki/File:Mipmap_Aliasing_Comparison.png](https://commons.wikimedia.org/wiki/File:Mipmap_Aliasing_Comparison.png)
CC0

28

28

Mapeamento de Textura em OpenGL

- Objetos são representados por polígonos em OpenGL
- Polígonos são compostos por vértices e arestas
- Cada vértice pode ser associado a um valor no espaço de textura
 - O mapeamento entre o vértice e a textura é calculado offline
 - Vem da parametrização da superfície
- Para associar uma textura (valor mapeado) a um vértice usa-se
 - `glTexCoord2fv (u, v);`
 - Chamar junto ao `glVertex3f` do vértice de interesse

29

29

Mapeamento de Textura em OpenGL

Passos antes de mapear texturas

- Habilitar o mapeamento
 - `glEnable(GL_TEXTURE_2D);`
- Criar container de textura
 - `void glGenTextures(GLsizei n, GLuint * textures);`
 - Cria *n* containers e armazena o ponteiro em *textures*
- Aponta para o container de textura a ser usado
 - `void glBindTexture(GLenum target, GLuint texture);`
 - Aponta para o container (do tipo especificado em *target*) descrito em *texture*

30

30

Mapeamento de Textura em OpenGL

Passos antes de mapear texturas

- Especificar e carregar a textura
 - `void glTexImage2D (GLenum target, GLint level, GLint internalFormat, GLsizei width, GLsizei height, GLint border, GLenum format, GLenum type, const GLvoid *pixels);`
 - Exemplo
 - `glTexImage2D (GL_TEXTURE_2D, 0, GL_RGBA, 256, 256, 0, GL_RGBA, GL_UNSIGNED_BYTE, image);`
 - Carrega uma imagem (apontada por *image*) de 256x256 armazenada como unsigned byte no formato RGBA
- OBS: Imagens em potência de 2 são mais eficientes

31

31

Mapeamento de Textura em OpenGL

Configura parâmetros da textura

- Controla a interação entre a cor da textura e do polígono
 - `void glTexEnvf(GLenum target, GLenum pname, GLfloat param);`
 - Exemplo
 - `glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_MODULATE);`
 - GL_MODULATE multiplica o valor da textura e do polígono
 - GL_BLEND mistura linearmente o valor da textura, do polígono e do ambiente
 - GL_REPLACE substitui o valor do polígono pelo da textura

32

32

Mapeamento de Textura em OpenGL

Configura parâmetros da textura (outros parâmetros)

- Continuidade da textura
 - Exemplo
 - `glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_CLAMP);`
 - `glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_REPEAT);`
- Filtros de magnificação e minimificação
 - Exemplo
 - `glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR); //ou GL_NEAREST`
 - `glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR); //ou GL_NEAREST`

33

33

Mapeamento de Textura em OpenGL

Resumo

- Preparação
 - Ler a imagem de textura para uma matriz T
 - Criar um container C para textura *glGenTextures*
 - Quando for usar o container, apontar p ele com *glBindTexture*
 - Carregar a imagem T para o contêiner C com *glTexImage2D*
 - Liberar a imagem T que já está carregada
- Uso
 - Configurar parâmetros gerais *glTexEnvf*, *glTexParameterf*, etc.
 - Apontar a textura de interesse com *glBindTexture*
 - Conectar cada vértice com a sua respectiva coordenada de textura
 - Usar *glTexCoord2f* quando emitir um vértice com *glVertex3f*

34

34

Perguntas ?????

35

35