UNICAMP - Universidade Estadual de Campinas

MIP-MAPPING

Renata Corrêa

Silmara Pedretti Gomes

Junho-2004

Estrutura do trabalho

- Computação gráfica – síntese de imagens
- Texturas
- Mip-mapping
- Aplicações práticas
- Conclusão

Computação Gráfica

- "*Conjunto de ferramentas e técnicas para converter dados para ou de um dispositivo gráfico através do computador".*
  - *Década 50: surgimento;*
  - 60:Sutherland – conceitos estruturação de dados e o núcleo da c.g. interativa;
  - 70: algorítmos, circuitos integrados, SIGGRAPH;
  - 80: iluminação global;
  - 90: OpenGL;

Computação Gráfica

- "*Conjunto de ferramentas e técnicas para converter dados para ou de um dispositivo gráfico através do computador".*

  - Início: década de 1950;
  - 1959: surge o termo Computer Graphics – Verne Hudson;
  - 1962: Sutherland – tese: introduziu as estruturas de dados para o armazenamento hierarquias e técnicas de interação usando teclado e caneta ótica.

Computação Gráfica

- 1970:
  - Novas técnicas e algorítmos usados até hoje;
  - Evolução do hardware: circuitos integrados;
  - Primeiro computador com interface visual;
  - Reconhecimento da Computação Gráfica como área específica da ciência da computação;
  - Surgimentos de congressos como o SIGGRAPH;

Computação Gráfica

- 1980
  - Imagens de satélite e explorações interplanetárias;
  - Surgimento de técnicas de iluminação global – fotorrealismo
- A partir de 1990
  - OpenGL – plataforma comum: PC;
  - Amadurecimento: imagens impressionantes.

Realismo Visual

- Em geral queremos que a imagem criada seja convincente, isto é, mais próxima possível da realidade.
- Classificação do realismo visual
  - Parte dinâmica: movimento da cena – animações;
  - Parte estática: fotorrealismo.

Realismo Visual

- Dificuldade: “complexidade do mundo real”.
  - Cores
  - Formatos
  - Texturas

Realismo Visual

- fotografia

- fotorrealismo

Realismo Visual

Textura

- É um método de alteração da propriedade da superfície do material, com o objetivo de dar aparência de detalhes à superfície que não estão presentes em sua geometria.

Textura

- Por quê utilizar texturas?
  - Representar cada aspecto da superfície de um objeto pode se tornar muito dispendioso.
  - O mapeamento de texturas melhora o detalhe da superfície sem usar um grande número de vértices.
  - As texturas, juntamente com a luz, auxiliam até mesmo na percepção do movimento.

Textura – Definição em OpenGL

- Identificar o objeto

  glGenTextures(GLsize n, GLuint *textures);

- Associar a textura ao seu nome

  glBindTexture(GLenum target, GLuint textureName);

Textura – Definição em OpenGL

- Especificar a imagem para o objeto

  glTextImage*D(GLenum target, GLint level, GLint internalformat, GLsizei width, GLsizei height, GLint border, GLenum format, GLenum type, const GLvoid *pixels);

- Habilitar a textura

  glEnable(GL_TEXTURE_*D);

Textura

- Por ser a textura a interação da luz com a superfície do objeto, os parâmetros da função de iluminação podem ser modulados por uma função de textura.
  - mapeamento do ambiente (environment mapping);
  - mapeamento de rugosidade (bump mapping);
  - mapeamento de refração;
  - mapeamento de transparência;
  - mapeamento de textura.

Mapeamento de textura

- O mapeamento de textura corresponde à projeção de uma imagem (usualmente 2D), digitalizada ou sintetizada, sobre uma superfície 3D.

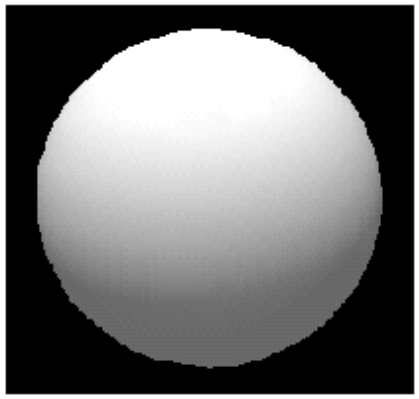


Imagem original

textura

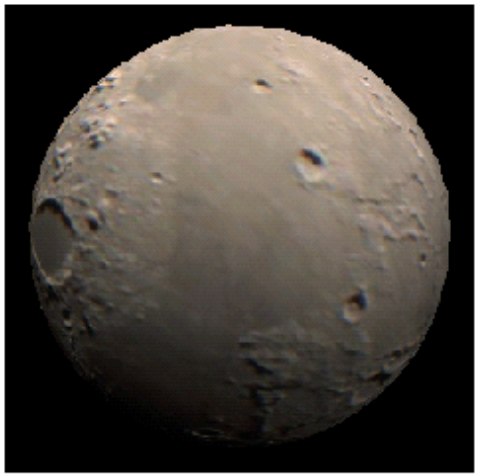


Imagem texturizada

Mapeamento de textura

- Para a aplicação da textura é preciso criar uma relação entre os vértices da textura e os vértices dos polígonos sobre os quais se deseja mapear a textura escolhida.

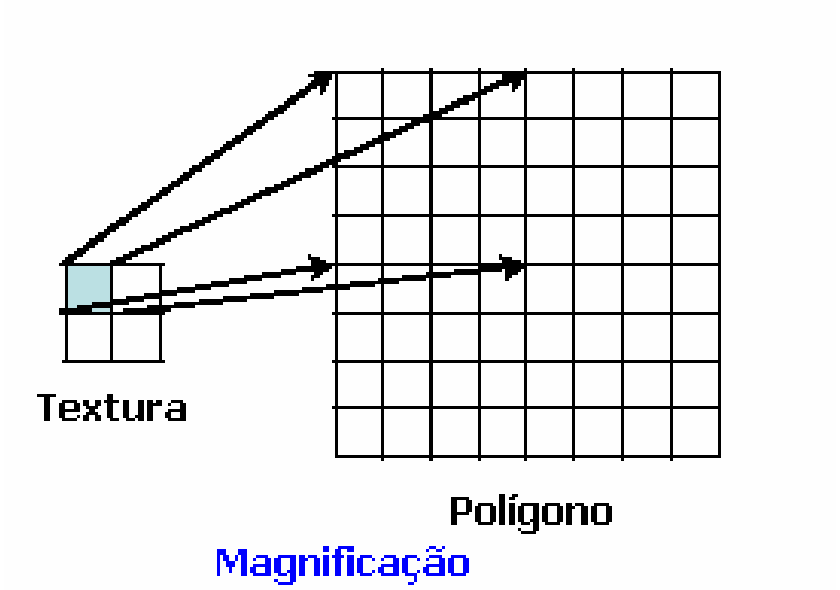
```
glTexCoord2f(x, y);
```

Mapeamento de textura

- Para objetos 3D complexos, mapear texturas ainda não é trivial.

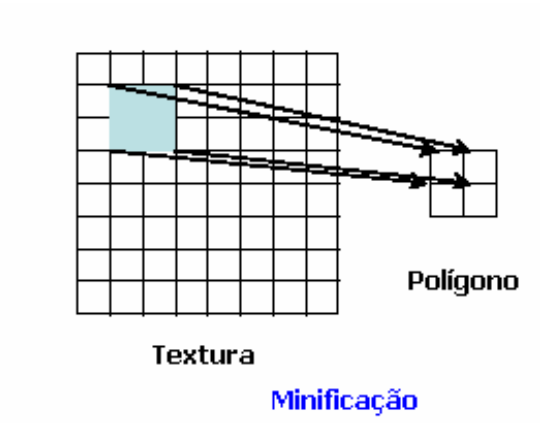
Mapeamento de textura

- Problemas:
  - Um texel pode cobrir mais de um pixel (magnificação)
  - Mais de um texel pode cobrir um pixel (minificação)



Mapeamento de textura

- Soluções para magnificação e minificação:
  - Vizinho mais próximo;
  - Interpolação linear;
  - Mip-mapping.

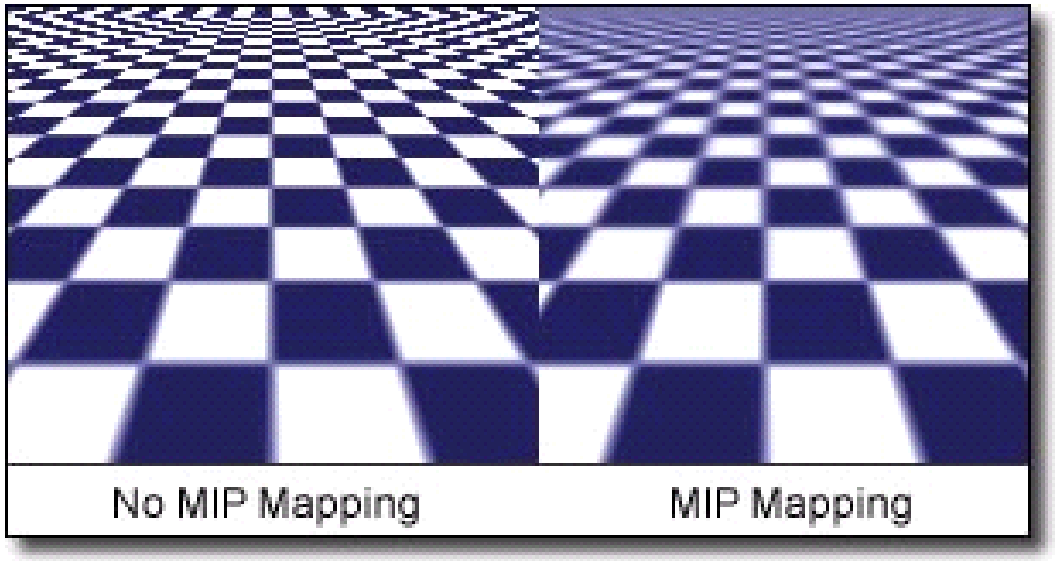


Mip-mapping

- MIP vêm das iniciais de “Multum In Parvo” e significa: muitas coisas num lugar pequeno.
  - É uma técnica de pré-filtragem utilizada para melhorar a qualidade visual de imagens sintéticas.

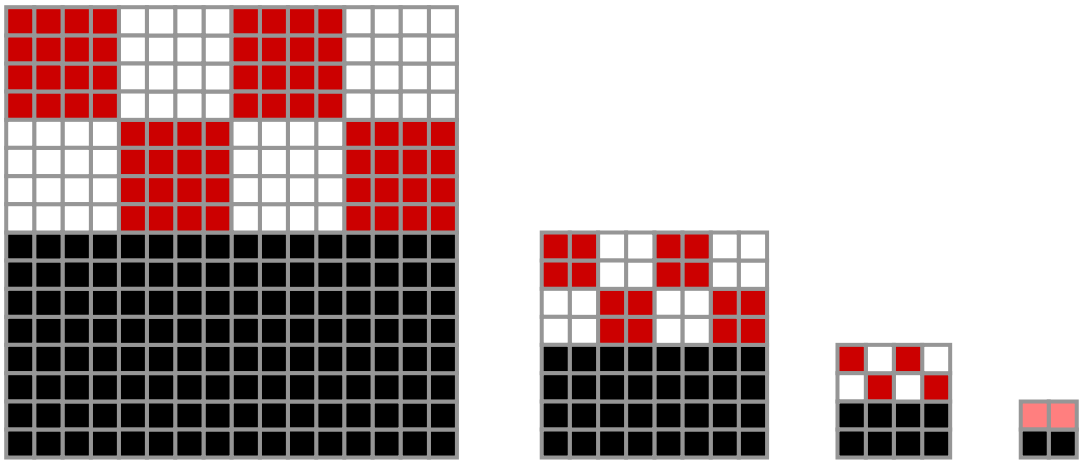
Mip-mapping

- Problemas que a técnica visa resolver:
  - Moire;
  - Baixa resolução quando a imagem é aproximada.



Mip-mapping

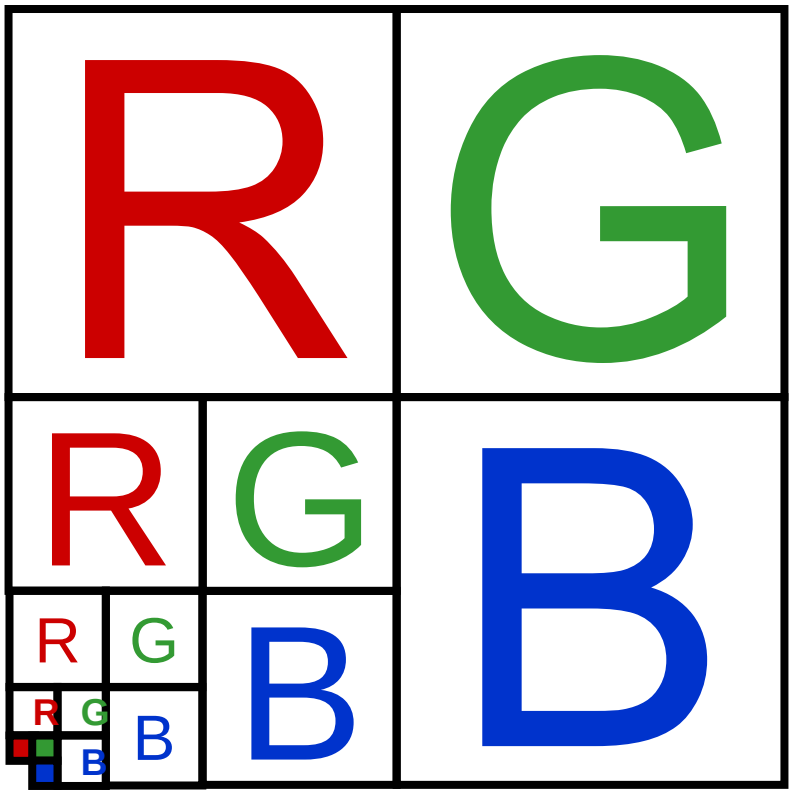
- A técnica consiste em pré-filtrar a imagem, criando múltiplas cópias de um mapa de textura, todas derivadas da primeira, com resolução cada vez menor.



Textura original *Versões de menor resolução*

Mip-mapping

- Armazenamento da imagem na memória



Mip-mapping - OpenGL

```
AUX_RGBImageRec *pTexArray[6];

pTexArray[0] = auxDIBImageLoad( ".\\tex256.bmp" );
pTexArray[1] = auxDIBImageLoad( ".\\tex128.bmp" );
pTexArray[2] = auxDIBImageLoad( ".\\tex64.bmp" );
pTexArray[3] = auxDIBImageLoad( ".\\tex32.bmp" );
pTexArray[4] = auxDIBImageLoad( ".\\tex16.bmp" );
pTexArray[5] = auxDIBImageLoad( ".\\tex8.bmp" );

glTexImage2D(GL_TEXTURE_2D, 0, 3, pTexArray[0]->sizeX, pTexArray[0]-
>sizeY, 0, GL_RGB, GL_UNSIGNED_BYTE, pTexArray[0]->data);

glTexImage2D(GL_TEXTURE_2D, 1, 3, pTexArray[1]->sizeX, pTexArray[1]-
>sizeY, 0, GL_RGB, GL_UNSIGNED_BYTE, pTexArray[1]->data);

...

glTexImage2D(GL_TEXTURE_2D, 5, 3, pTexArray[5]->sizeX, pTexArray[5]-
>sizeY, 0, GL_RGB, GL_UNSIGNED_BYTE, pTexArray[5]->data);
```

Mip-mapping

- Para decidir qual nível de mapa utilizar, temos que calcular o fator de compressão.

$$\text{fator de compressão} = \frac{\text{área retangular textura}}{\text{área retangular pixel}}$$

- O fator de compressão nos diz quantos texels cabem num pixel, na média.

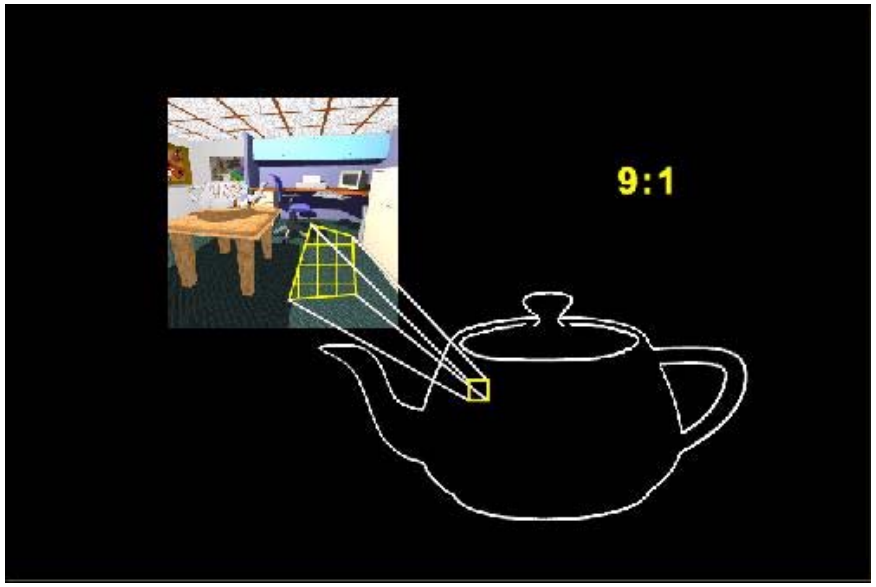
Mip-mapping

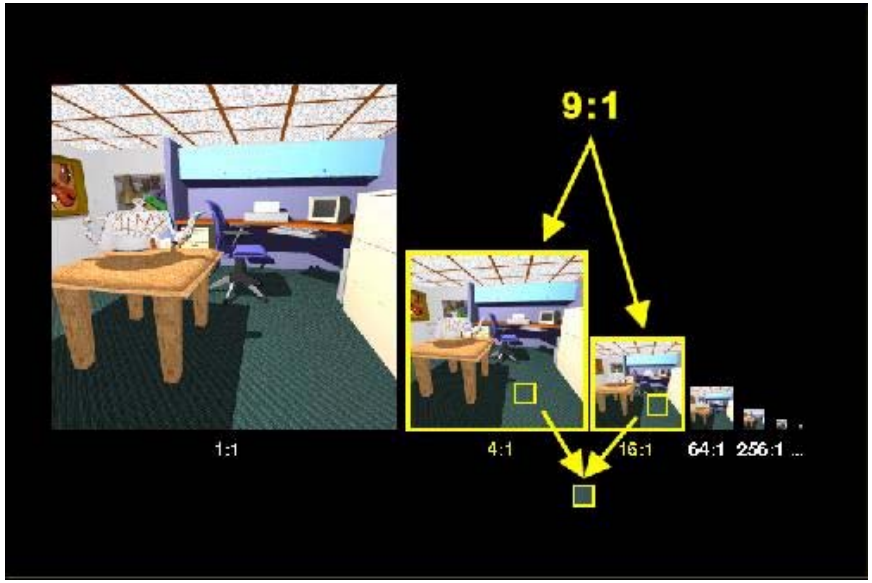
d = fator de compressão

- Se d <=1: temos menos de um texel por pixel.
  - Retorna o mapa de maior resolução
- Se d >= tamanho do bitmap: o bitmap inteiro cabe dentro de um único pixel.
  - Retorna a cor do texel do canto inferior direito

Mip-mapping

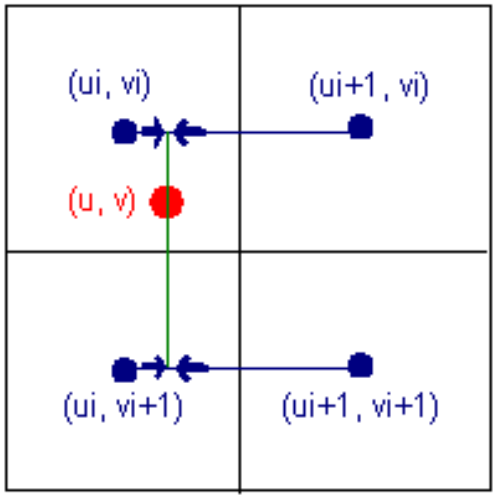
- Se d>1 e < tamanho do bitmap: encontramos as duas potências de 2 entre as quais d se encontra.
  - Retorna os dois mapas mais apropriados;





Mip-mapping

- A seguir:
  - Calcular a cor em cada mapa;



  - Interpolar os valores entre os dois mapas.

Mip-mapping – OpenGL

- Os filtros são aplicados na textura com a utilização do comando :

glTexParameter*(GLenum target, GLenum pname,GLfloat param)

Mip-mapping – OpenGL

pname	parameter
GL_TEXTURE_MAG_FILTER	■GL_NEAREST ■GL_LINEAR
GL_TEXTURE_MIN_FILTER	■GL_NEAREST ■GL_LINEAR ■GL_NEAREST_MIPMAP_NEAREST ■GL_NEAREST_MIPMAP_LINEAR ■GL_LINEAR_MIPMAP_NEAREST ■GL_LINEAR_MIPMAP_LINEAR

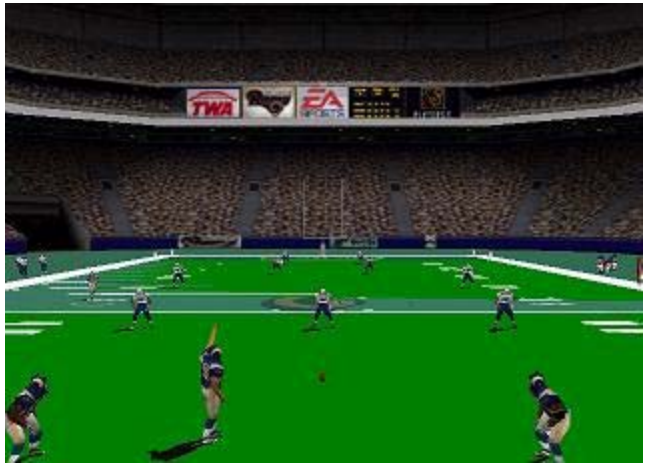
Mip-mapping

- Exemplo em OpenGL:

Aplicações práticas

- Nintendo 64: efeitos especiais como o mip-mapping e anti-aliasing foram usados pela primeira vez em um videogame.

Aplicações práticas



Madden NFL 2001

Bug’s Life

Conclusão

- O mip-mapping é um filtro que traz bons resultados no mapeamento de texturas;
- Evita o cálculo do valor filtrado para cada pixel;
- Consome apenas 1/3 a mais de memória que o mapeamento comum;
- Embora ele não resolva todos os problemas de aliasing, pode ser utilizado com outras técnicas para trazer resultados ainda melhores.

FIM

Perguntas?

Silmara Pedretti Gomes

Renata Corrêa