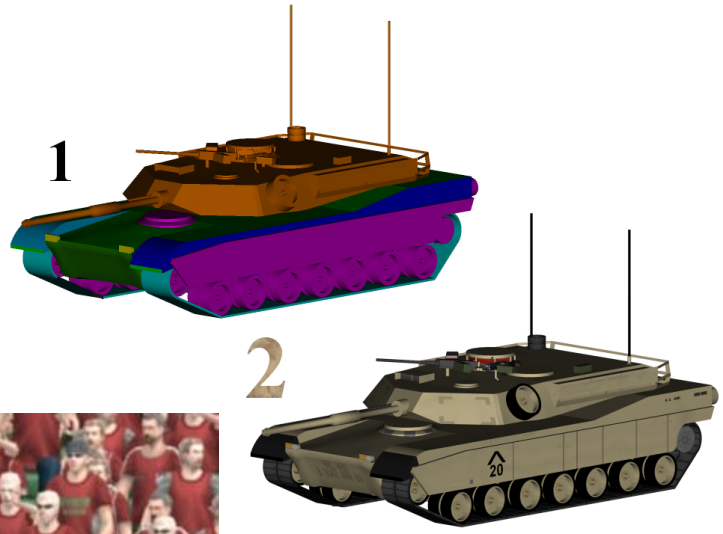
Sumário

- Arquitectura básica
- Renderização de primitivas
- Transformações
- Animação e buffer de profundidade
- Mapeamento de texturas
-
-

Mapeamento de texturas



Motivação

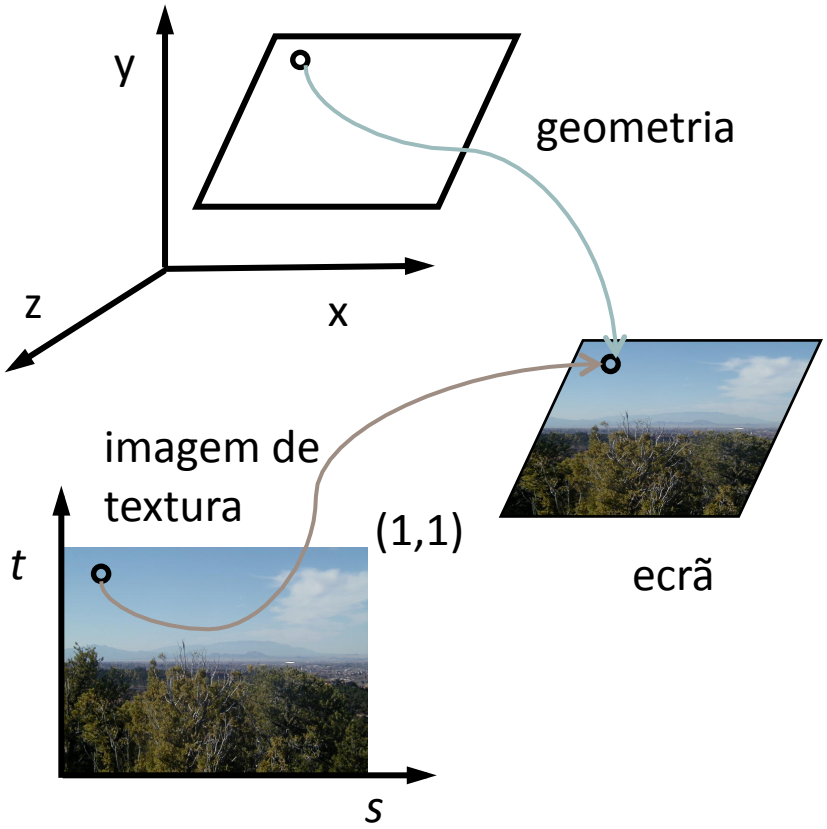
- Simulação de materiais e.g. madeira, granito
- Redução da complexidade geométrica e.g. número de polígonos
- Processamento de imagem e.g. *wrapping*
- Efeito de superfícies reflectoras e.g. espelhos

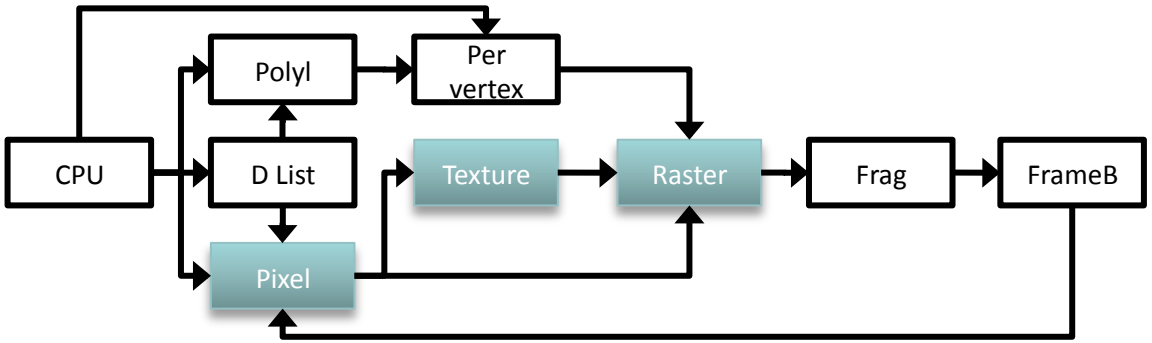
Princípio

- Mapeamento de uma imagem - conjunto de cores a 1, 2 ou 3D, denominados texels – num objecto geométrico

Processamento de texturas

- Imagens e geometria possuem canais de processamento separados, juntando-se apenas na fase de rasterização, o que significa que a complexidade da geometria não é afectada pela complexidade da textura
- O tamanho das texturas deve ser múltiplo de 2





Operações para aplicação de texturas

Tarefas em geral

- Especificação da textura
  - Leitura ou geração da imagem
  - Afectação da imagem à textura
  - Disponibilização do modo de textura
- Afectação das coordenadas da textura aos vértices
- Definição de parâmetros de textura
  - *Wrapping*, filtragem, etc.

Tarefas para criação de objectos de textura

- Definição de filtro de textura
- Definição de função de textura
- Definição do modo de *wrapping* da textura
- Definição de atributos para correcção da perspectiva
- *Binding* do objecto de textura
- Definição das coordenadas de textura para cada vértice (incluindo a sua geração)

GL_TEXTURE_1D

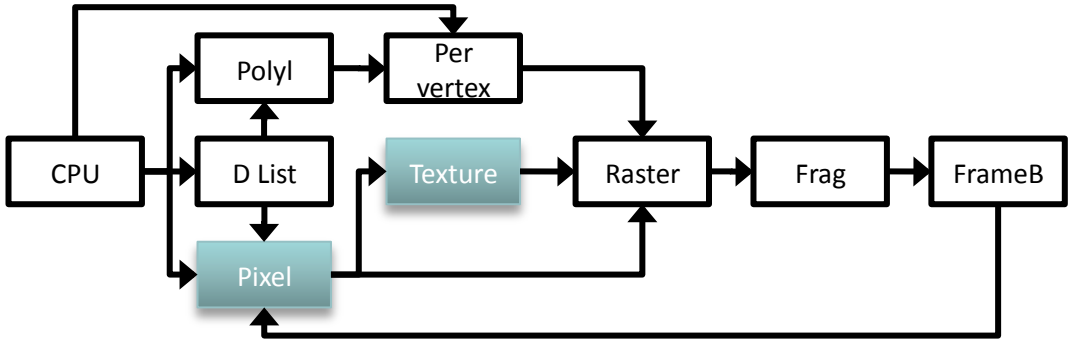
GL_TEXTURE_2D

GL_TEXTURE_3D

Obs.:

- Algumas tarefas são opcionais
- Permite simplificações internas

Objectos de textura



Enquadramento

- Uma imagem por cada objecto de textura, a qual pode ser partilhada em vários contextos gráficos

Geração dos nomes da textura

- glGenTextures(n, *textIds)

Criação do objecto de textura com informação sobre a textura e o estado

- glBindTexture(target, id)

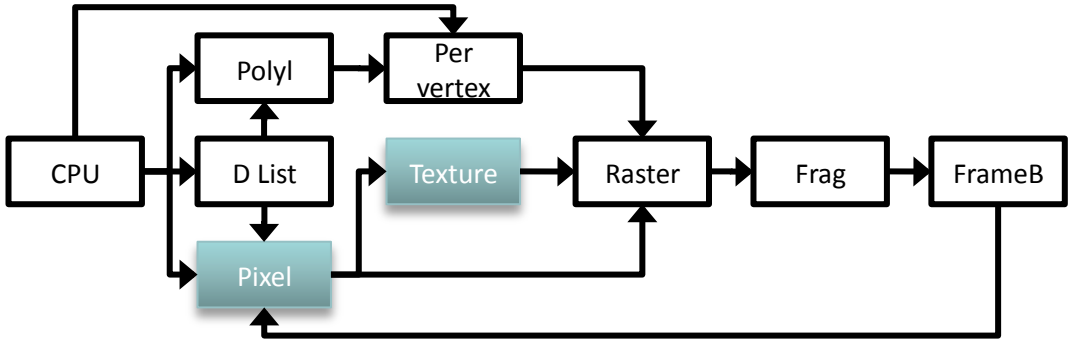
Definição da imagem de textura a partir de um vector de texels residentes em memória

- glTexImage2D(target, level, components, w, h, border, format, type, *texels)
- A dimensão da imagem deve ser múltipla de 2
- As cores dos texels são processados pelo canal de processamento dos pixels
- Os pixels podem sofrer alterações de escala /filtragem: gluScaleImage (format, w_in, h_in, type_in, *data_in, w_out, h_out, type_out, *data_out)

Outros métodos para especificação da imagem de textura

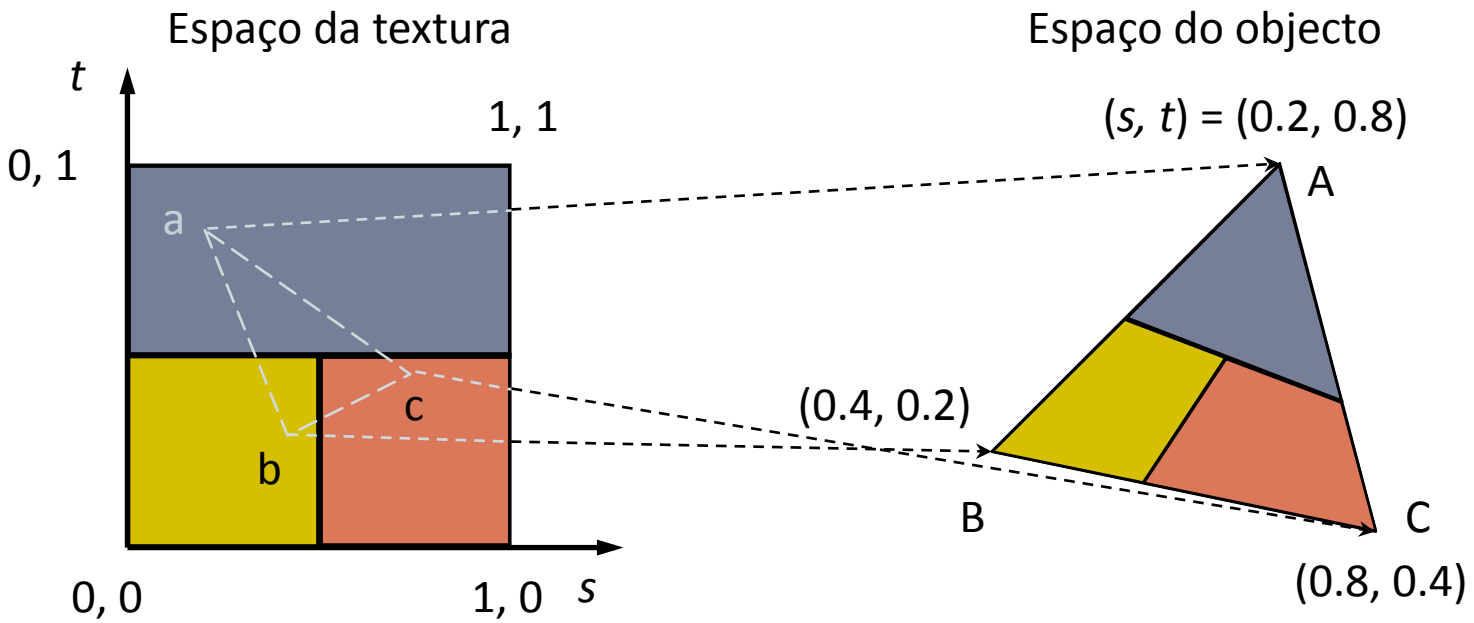
- Utilização do framebuffer (buffer corrente) como fonte da imagem de textura: glCopyTexImage1D(..) (ou 2D)
- Modificação de parte de uma textura já definida: glTexSubImage*D(…)
- Utilização de ambas as técnicas: glCopyTexImage2D(), …

Mapeamento da textura



Mapeamento com base nas coordenadas paramétricas da textura, sendo especificado para cada um dos vértices

glTextCoord*()



Geração de coordenadas de textura

Geração automática

- glEnable(GL_TEXTURE_GEN_{STRQ}) e glTexGen{ifd}[] ()

Especificação de um plano

- Ax+By+Cz+D=0
- O plano pode sofrer transformações geométricas

Modos de geração

- Textura fixa ao objecto (tipo parede): GL_OBJECT_LINEAR
- Textura fixa no espaço e objecto a deslocar através da textura (tipo luz no fundo de aquário de peixes): GL_EYE_LINEAR
- Objecto denota o meio ambiente (tipo utilização de espelhos): GL_SPHERE_MAP

Modos de aplicação de texturas

Modos de filtragem para ampliar ou reduzir texels, já que estes nem sempre são do mesmo tamanho dos pixels

- Ampliação
- Redução
- Mipmapping, como instância especial de filtros de redução

Modos de wrapping

- Corte (GL_CLAMP)
- Repetição (GL_REPEAT)

Funções de textura

- Descrição da forma como os fragmentos de cor das primitivas são combinados com as cores dos texels
- Pixels e texels podem ser multiplicados, linearmente combinados ou o texel pode substituir a cor do fragmento

Modos de filtragem

Controlo da forma como os pixels são ampliados ou reduzidos

- Ampliação, redução ou texturas mipmapped
- glTexParameteri(target, type, mode);
- Tipo de filtro: GL_TEXTURE_MIN_FILTER, GL_TEXTURE_MAG_FILTER
- Modo do filtro: GL_NEAREST, GL_LINEAR ou modos especiais de mipmapping

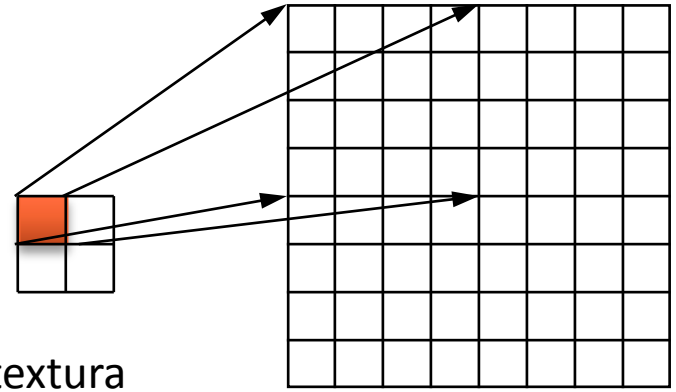
Texturas mipmapped

- Permitem o mapeamento de texturas pré-filtradas de resolução inferior através da criação de várias aproximações da imagem original em resoluções mais baixas. É pois um método LOD (level of detail)
- Cada nível corresponde a uma imagem que é metade da altura e largura da do nível anterior (ex: nível 0 com 32x8 texels, nível 1 com 16x4, e assim sucessivamente até 1x1)
- Menos erros de interpolação para pequenos objectos texturados
- Modos especiais de mipmapping: GL_NEAREST_MIPMAP_NEAREST, GL_NEAREST_MIPMAP_LINEAR, GL_LINEAR_MIPMAP_NEAREST, GL_LINEAR_MIPMAP_LINEAR
- Criação automática de mipmaps
  - gluBuild2DMipmaps(..) (1D e 3D) e/ou gluBuild2DMipmapLevels(..) (1De 3D)
- O nível de mipmap é declarado aquando da definição de textura - glTexImage*D(, level, ..)

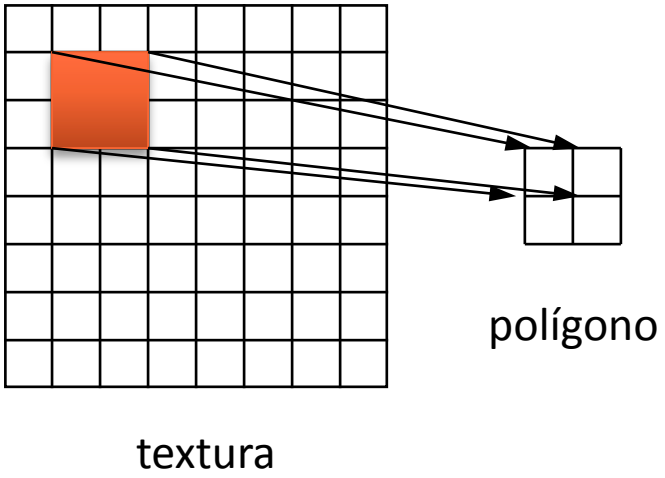
Tipos de filtros: GL_TEXTURE_MAG_FILTER, GL_TEXTURE_MIN_FILTER

Modos: GL_NEAREST, GL_LINEAR

Ampliação



Redução



Obs.: Mipmaps só em redução, com modos próprios

Modo de wrapping

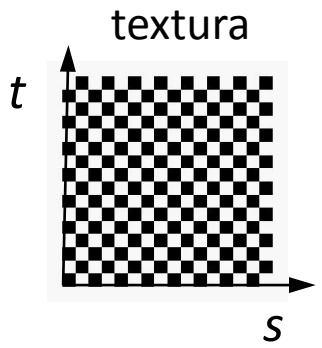
Forma de proceder se a coordenada de textura estiver fora do intervalo [0,1]

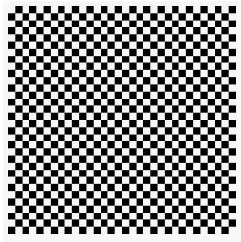
- glTexParameteri()
- Modo GL_REPEAT: para as coordenadas fora de [0,1] só conta a parte fraccionária
- Modo GL_CLAMP: utiliza o valor dos limites (0 ou 1)

Exemplos:

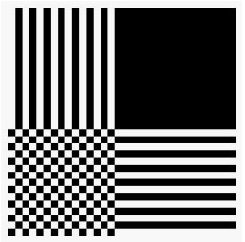
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_REPEAT)

glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_CLAMP)





GL_REPEAT



GL_CLAMP

Funções de textura

Controlo da forma como a textura é aplicada, ou seja, como os texels e os fragmentos de cores são combinados

- glTextEnv{fi}[v] (GL_TEXTURE_ENV, *prop*, *param*)

Modos usuais

- GL_MODULATE, multiplicando texel e cor
- GL_BLEND, mistura linear de texel, fragmento e cor de ambiente
- GL_REPLACE, substituindo a cor do fragmento com texel

Obs.: se *prop* é GL_TEXTURE_ENV_COLOR, então *param* especifica a cor a usar pela função de textura GL_BLEND

Correcção de perspectiva

Métodos para interpolação de cores com coordenadas de textura

- Linearmente no espaço de ecrã
- Utilização de valores de profundidade de perspectiva (mais lento)

Problemático em polígonos “nos limites”

- glHint (GL_PERSPECTIVE_CORRECTION_HINT, hint), com *hint* a ser definido com um dos valores GL_DONT_CARE, GL_NICEST, ou GL_FASTEST