



Tema 6: Aplicación de Texturas 2D

J. Ribelles

SIE020: Síntesis de Imagen y Animación
Institute of New Imaging Technologies, Universitat Jaume I

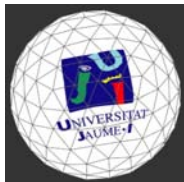
Contenido

- 1 Introducción
- 2 Coordenadas de Textura
- 3 Creación de una Textura 2D
- 4 Muestreo de la Textura

Introducción

Introducción

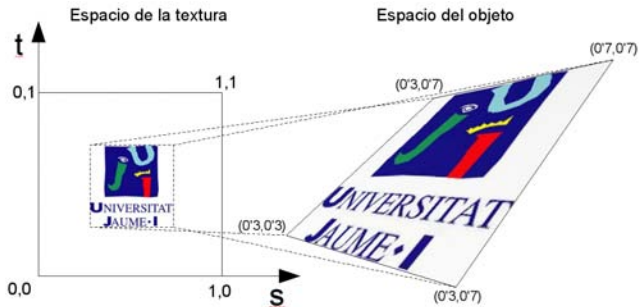
- El uso de texturas para aumentar el realismo visual de las aplicaciones es muy frecuente.
- Se va a mostrar cómo se puede utilizar una imagen 2D a modo de mapa de color de manera que el valor definitivo de un determinado píxel dependa de la iluminación de la escena y de la textura.
- En concreto, se presenta cómo utilizar OpenGL para aplicar una textura 2D sobre una superficie.



Coordenadas de Textura

Descripción

- Las coordenadas de textura son un atributo más de los vértices.
- Es responsabilidad del programador suministrar estas coordenadas.
- El rango de coordenadas válido en el espacio de la textura es entre 0 y 1, independientemente del tamaño en píxeles de la textura.



Creación de una Textura 2D

Son tres pasos:

- 1 Crear un objeto textura.
- 2 Asignar la unidad de textura.
- 3 Especificar para cada vértice de la superficie sus coordenadas de textura.

Listado 1: Lectura de un archivo de imagen en formato RGB

```
GLubyte *leeTextura (char *nombreFichero, int ancho, int alto) {  
  
    GLubyte *textura = (GLubyte*) malloc (  
        sizeof(GLubyte)*3*ancho*alto);  
  
    FILE    *fichero;  
  
    fichero = fopen (nombreFichero, "rb");  
  
    fread (textura, sizeof(GLubyte), ancho*alto*3, fichero);  
    fclose (fichero);  
    return textura;  
}
```

Listado 2: Ejemplo de creación de una textura

```
GLubyte *textura= leeTextura("metal.rgb", 256, 256);

// Solicita un nombre
GLuint nombre;
glGenTextures (1, &nombre);

// Crea un objeto textura
glBindTexture (GL_TEXTURE_2D, nombre);

// Especifica la textura RGB de talla 256x256
glTexImage2D (GL_TEXTURE_2D, 0, GL_RGB, 256, 256, 0, GL_RGB,
              GL_UNSIGNED_BYTE, textura);

// Repite la textura tanto en s como en t
glTexParameteri (GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_REPEAT);
glTexParameteri (GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_REPEAT);

// Filtrado lineal de la textura
glTexParameteri (GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR);
glTexParameteri (GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR);

// Activa la unidad de textura 0
glActiveTexture (GL_TEXTURE0);

// Asigna el objeto textura a dicha unidad de textura
glBindTexture (GL_TEXTURE_2D, nombre);

// Obtiene el indice de la variable del Shader de tipo sampler2D
GLuint loc= glGetUniformLocation (program, "Img");

// Indica que Img del Shader use la unidad de textura 0
glUniform1i (loc,0);
```

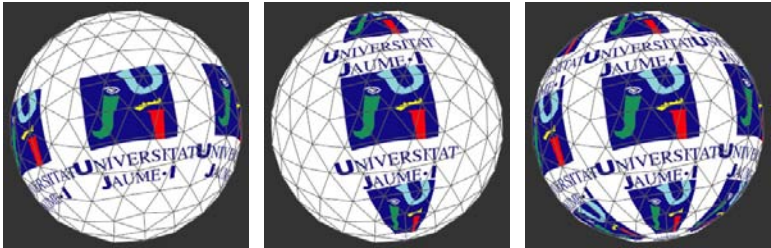


Figura: Ejemplo de repetición de textura. A la izquierda se ha repetido la textura en sentido s , en el centro en sentido t , y a la derecha en ambos sentidos.

Muestreo de la Textura

Descripción

- Las coordenadas de textura se proporcionan para cada vértice y son interpoladas en el *pipeline* del procesador gráfico.
- En el procesador de fragmentos se utiliza la orden *texture* para acceder al texel correspondiente a dicho fragmento.
- Dicha función devuelve el color que ya ha sido procesado según los parámetros de repetición y filtrado especificados por el programador.

Listado 3: Acceso a la textura desde el Fragment Shader

```
uniform sampler2D lmg;  
  
in  vec2 coordTextura;  
out vec4 colorFragmento;  
  
void main()  
{  
    colorFragmento = texture (lmg, coordTextura);  
}
```