

Unidad 7

Texturas

Introducción

Con los modelos de sombreado se obtienen superficies iluminadas pero carentes de realismo.



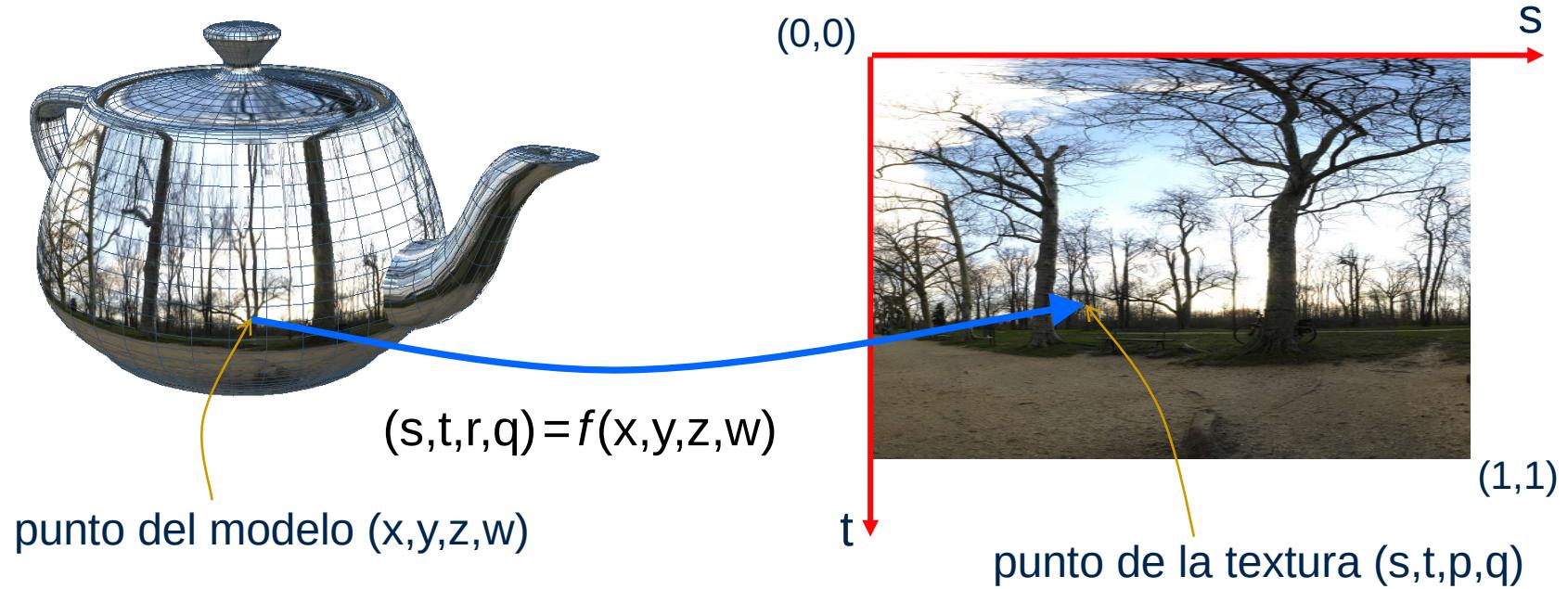
Introducción

- Un piso de césped o una madera veteada requerirían gran cantidad de polígonos.
- Las texturas permiten simular superficies complejas con pocos polígonos



Introducción

- Una **textura** es un array de píxeles (téxeles) con 1 a 4 canales (RGBA)
- Tiene entre 1 y 4 dimensiones (normalmente 2: imagen)
- Coordenadas reales en la “imagen” $(s,t,r/p,q) \in [0,1]^4$

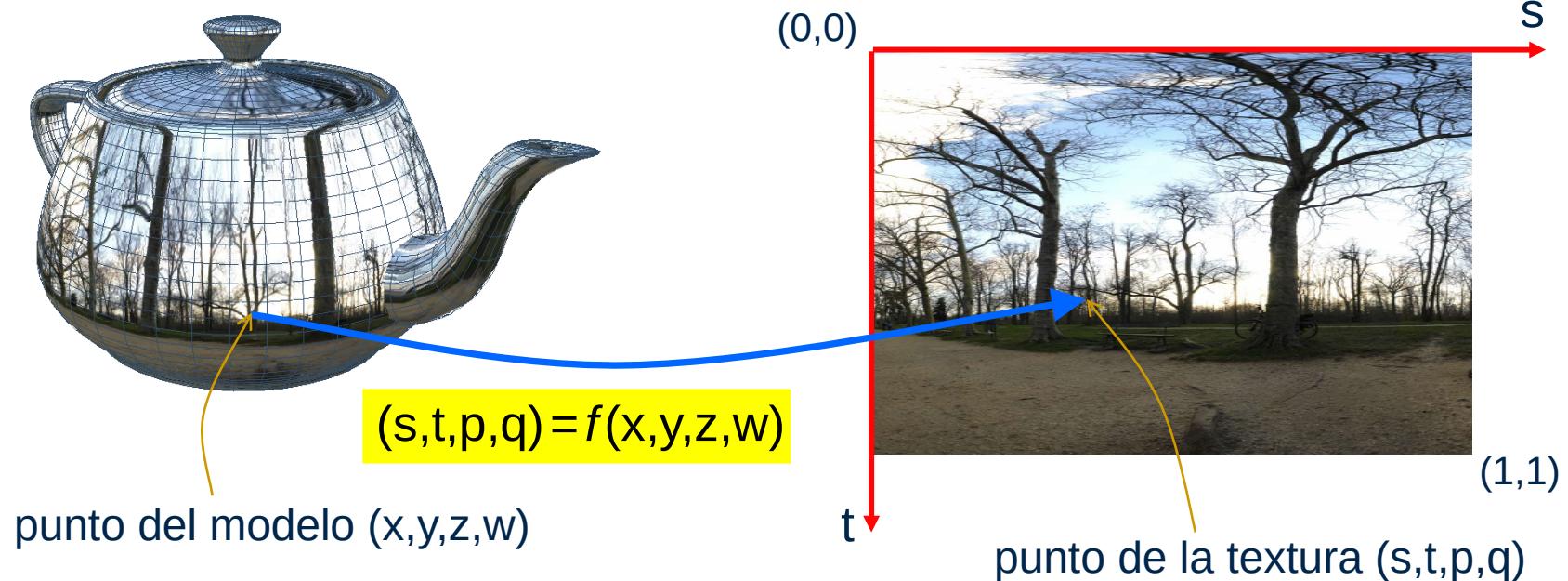


Introducción

Mapear una textura consiste en definir la función que asigna coordenadas de textura a cada punto del objeto: $(s,t,p,q) = f(x,y,z)$.

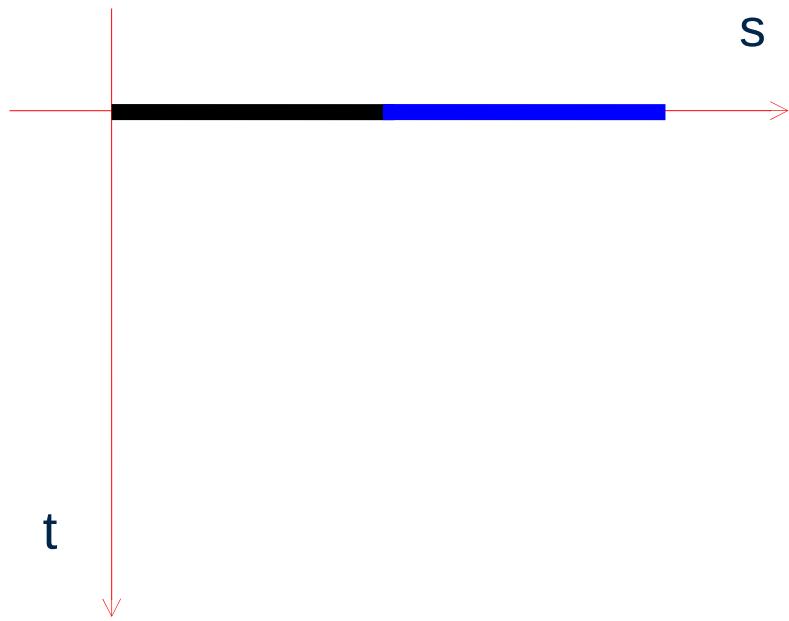
¿Cómo se define?

La define el programador ¡Como pueda! Directa y analíticamente o en pasos.

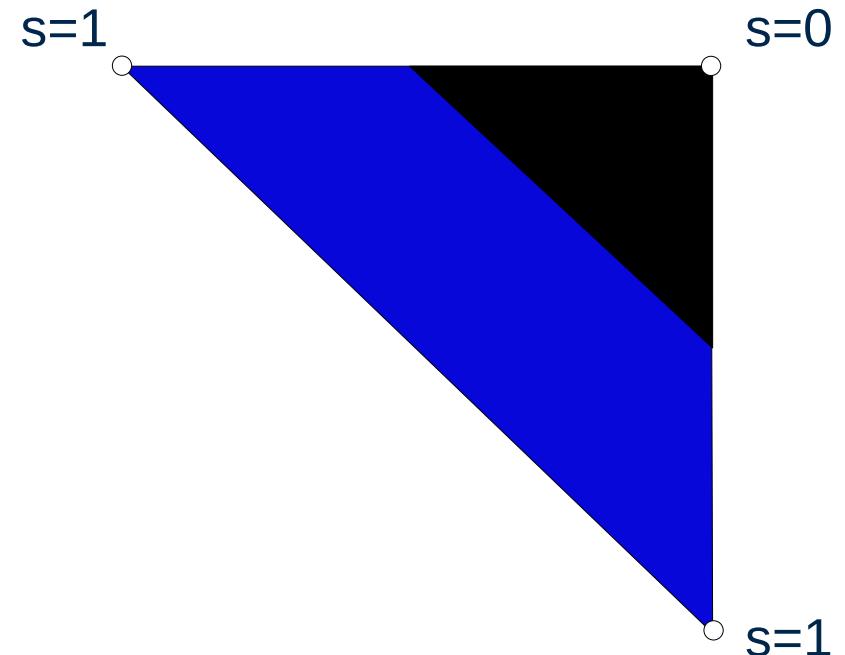


Dimensiones

Texturas 1D: $s=f(x,y,z,w)$ “imagen” de un téxel de alto.



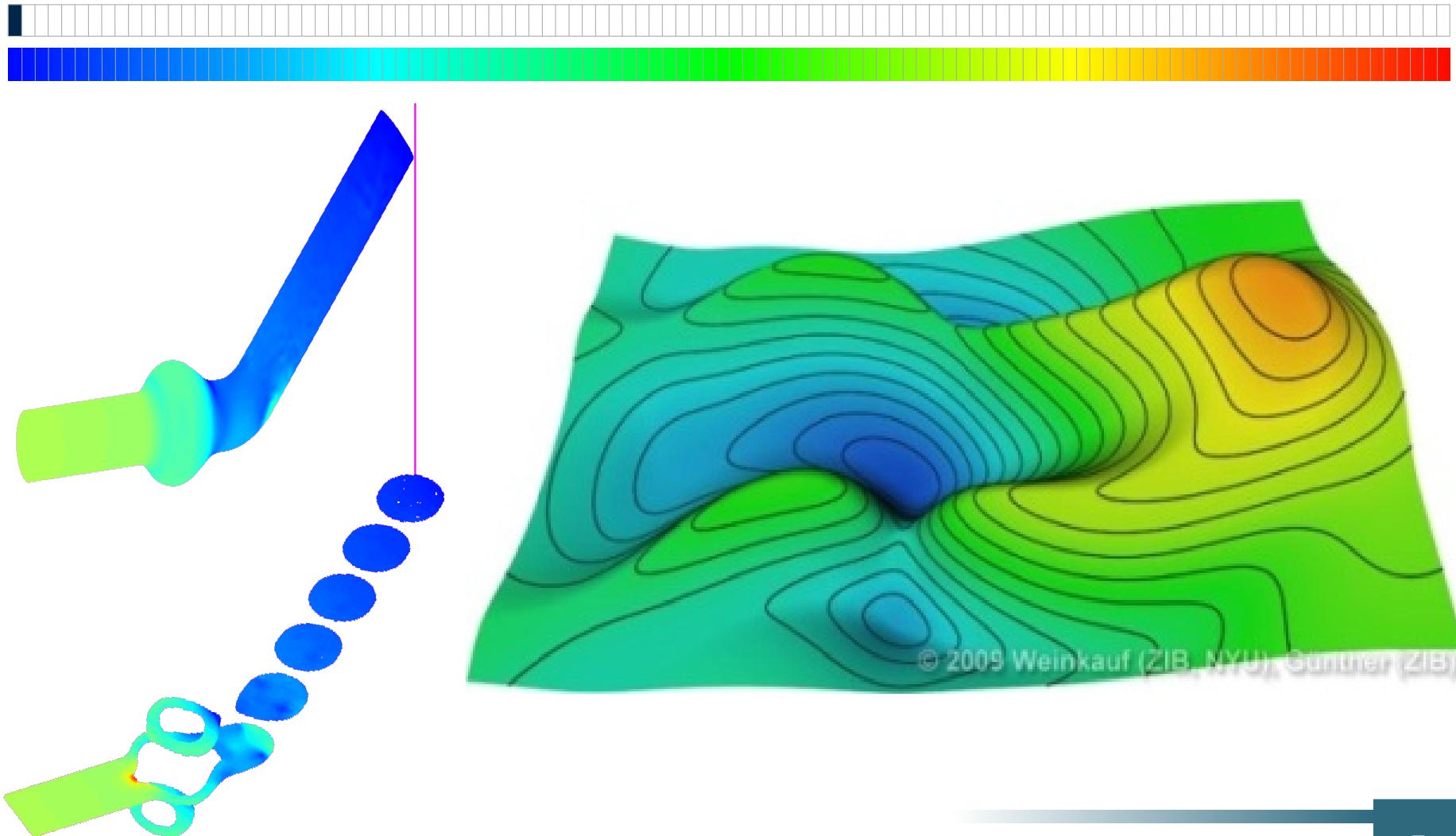
Textura 1D



Aplicada a triangulo

Dimensiones

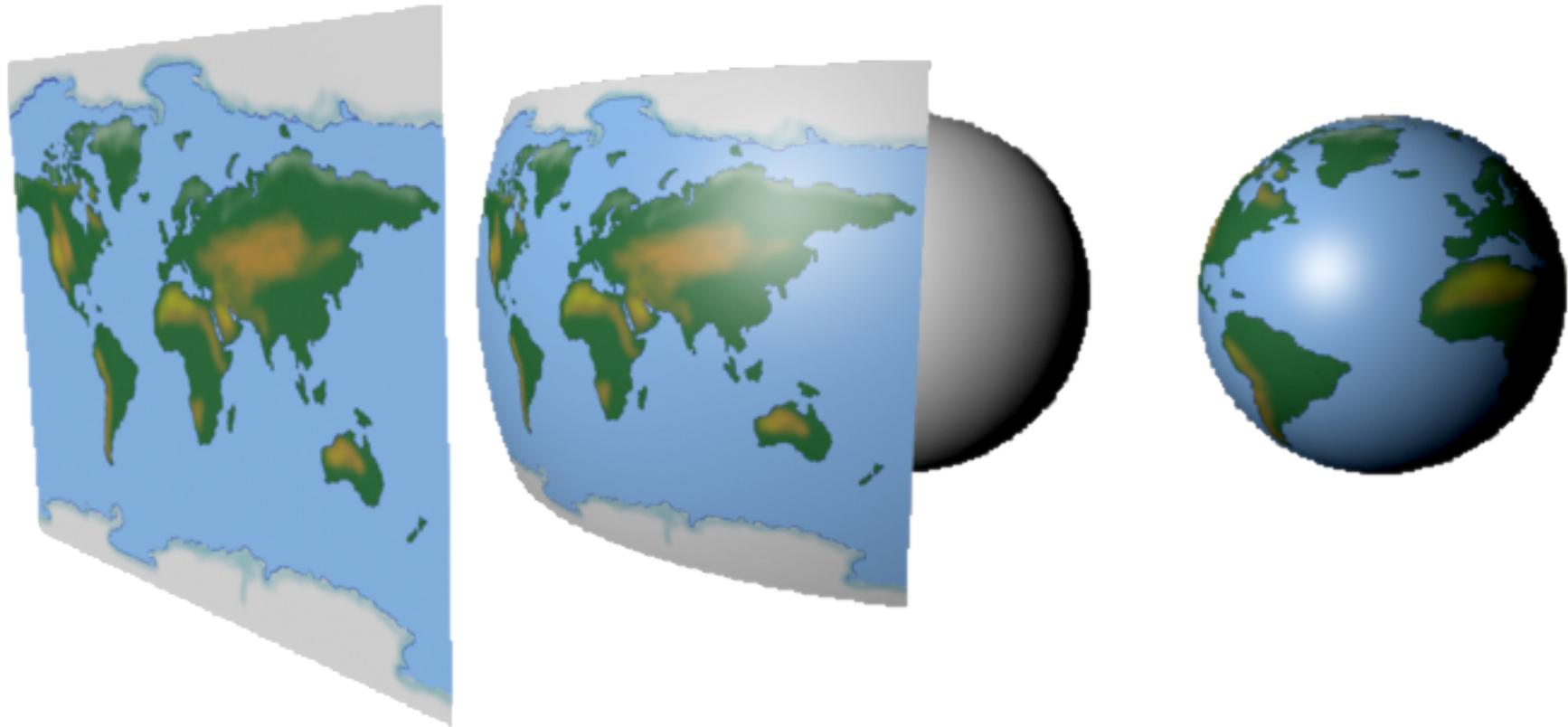
Texturas 1D: $s=f(x,y,z,w)$ “imagen” de un téxel de alto.



Dimensiones

Texturas 2D: $(s,t) = f(x,y,z,w)$: Se podría pensar que se le está pegando una imagen 2D deformable a una superficie en 3D.

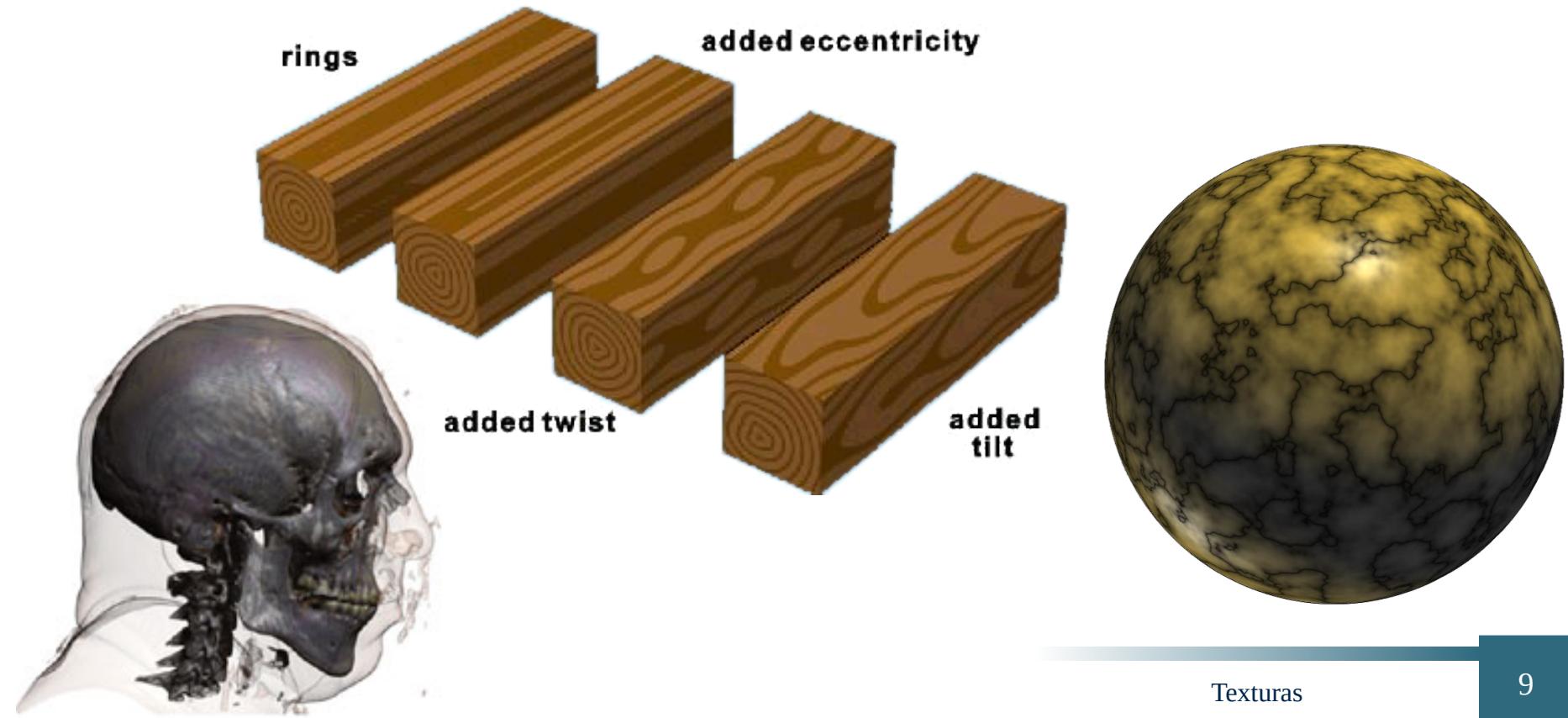
Es el tipo de textura más utilizado.



Dimensiones

Texturas 3D: $(s,t,r)=f(x,y,z)$: Volumen con "vóxeles" RGBA.
Es una pila de imágenes. Necesita mucha memoria.

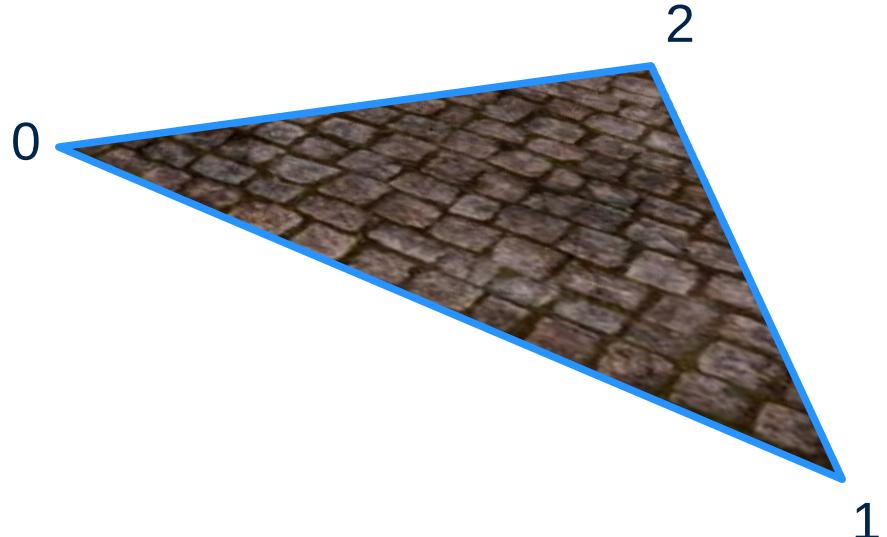
- Texturas procedurales (mármol, madera).
- Visualización médica de tomografías.



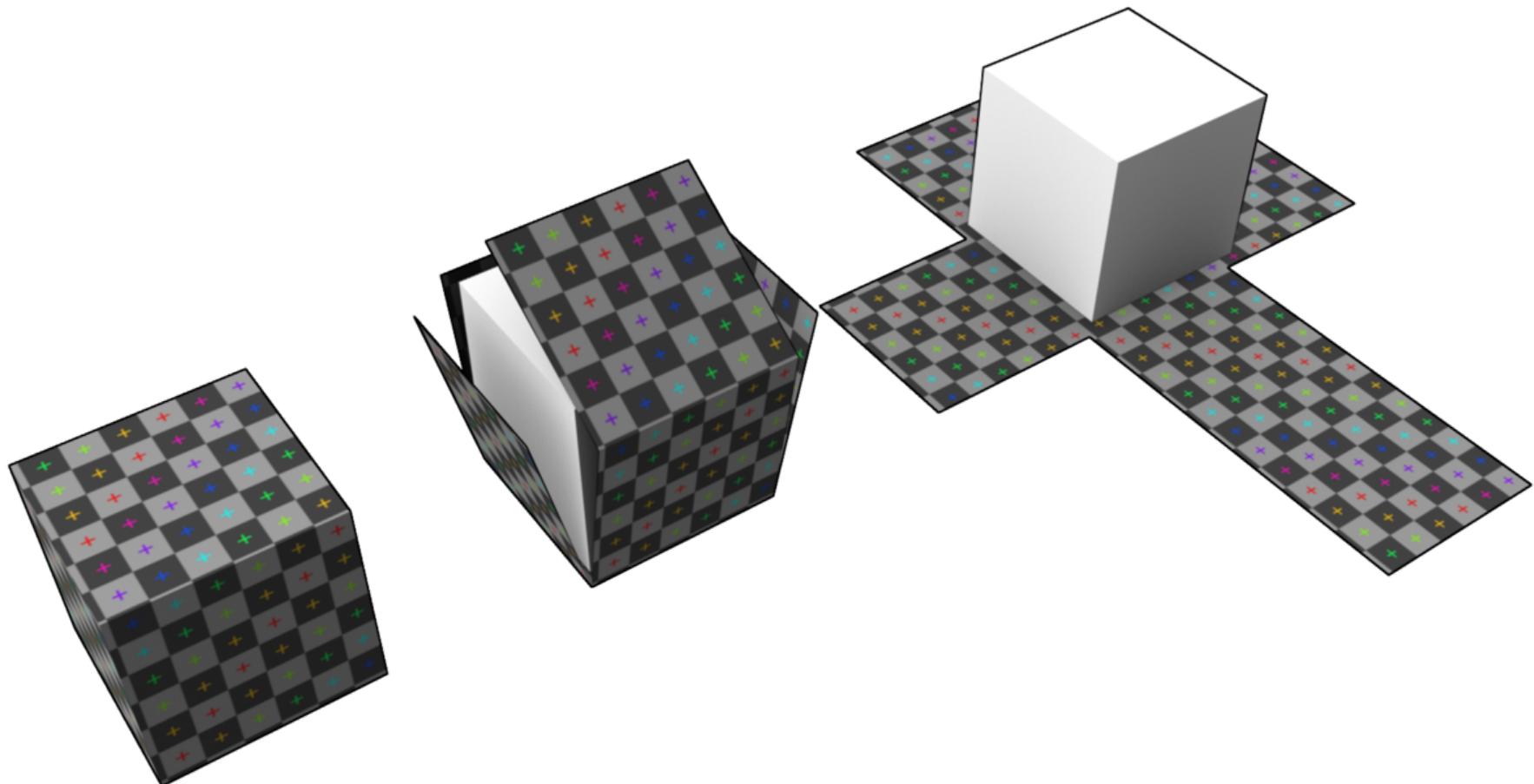
Mapeo de textura: manual

Las coordenadas de textura se asignan manualmente a **cada vértice** de la primitiva y se interpolan automáticamente en cada fragmento.

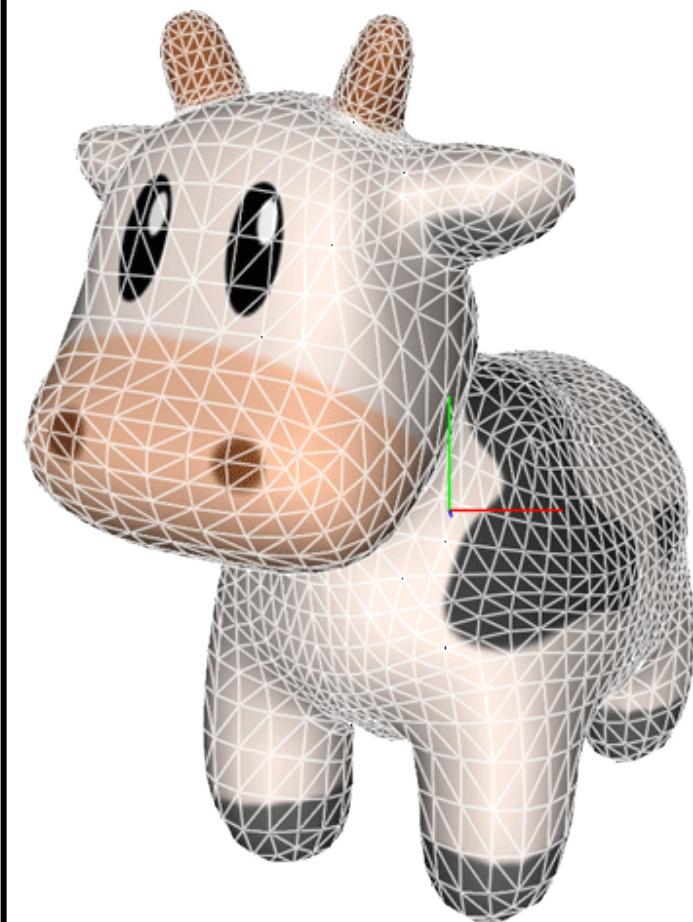
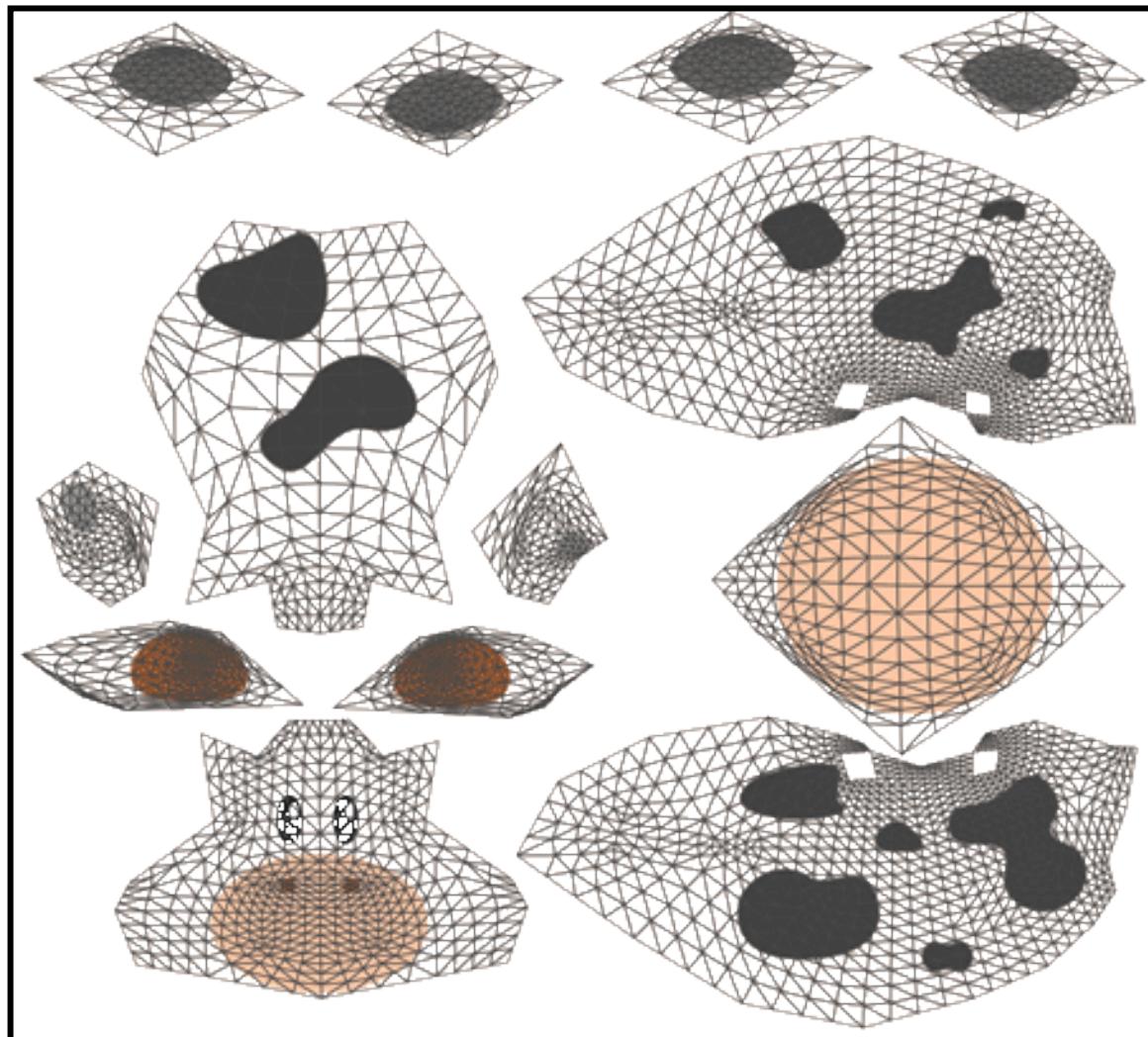
```
glBegin(GL_TRIANGLES);  
    glColor3f(r0, g0, b0);  
    glNormal3f(u0, v0, w0);  
    glTexCoord2f(s0, t0);  
    glVertex3f(x0, y0, z0);  
    glColor3f(r1, g1, b1);  
    glNormal3f(u1, v1, w1);  
    glTexCoord2f(s1, t1);  
    glVertex3f(x1, y1, z1);  
    glColor3f(r2, g2, b2);  
    glNormal3f(u2, v2, w2);  
    glTexCoord2f(s2, t2);  
    glVertex3f(x2, y2, z1);  
    ...  
    ...  
    glEnd();
```



Mapeo de textura: mapeo UV (manual)

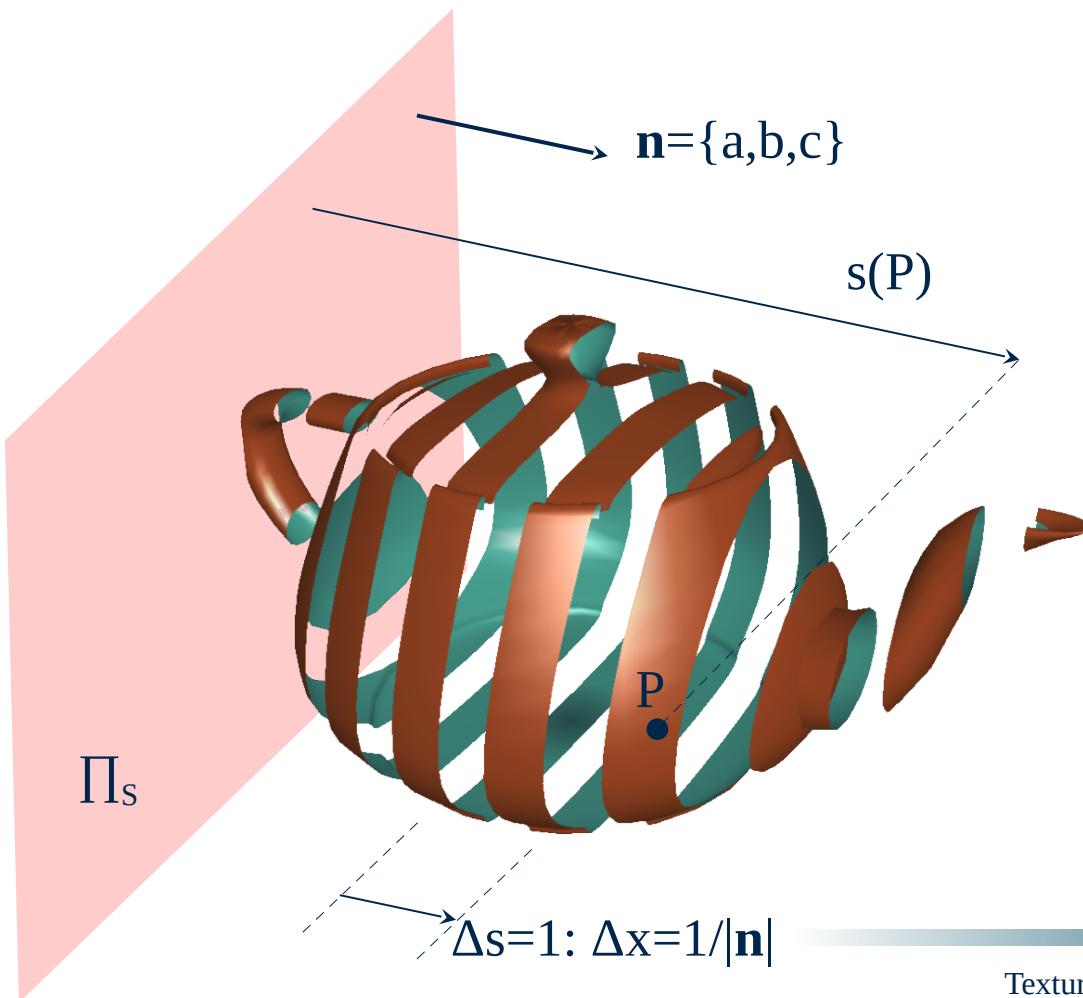


Mapeo de textura: mapeo UV (manual)



Mapeo automático: mapeo plano

- Cada coordenada de textura se obtiene de la distancia a un plano.
- Es el ideal para suelos, paredes y demás superficies planas.
- Implementado en OpenGL



Mapeo automático: mapeo en dos partes

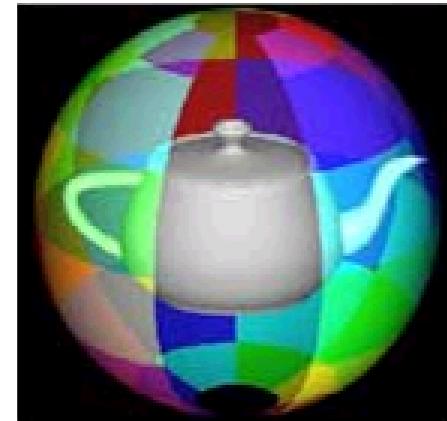
- No está implementado en OpenGL, hay que asignar coordenadas de textura a cada vértice en la CPU.
- Primero se mapea la textura sobre una superficie 3D sencilla, enrollada sobre el objeto (S-Mapping):

- Cilindro
- Esfera
- Cubo

Por ejemplo

$$u = \text{atan}2(x, z) \in [0, 2\pi] \Rightarrow s = \frac{u}{2\pi}$$

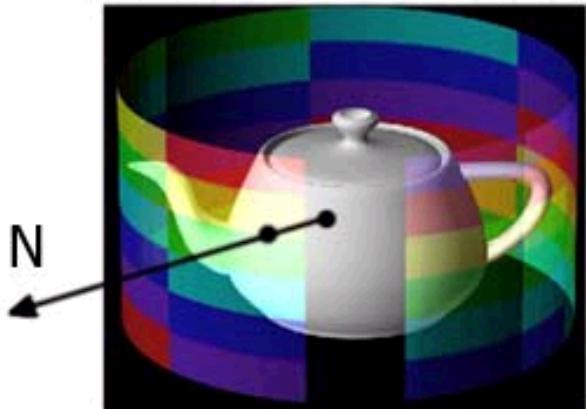
$$v = \text{atan}2(y, \sqrt{x^2 + z^2}) \in [-\pi, \pi] \Rightarrow t = \frac{v + \pi}{2\pi}$$



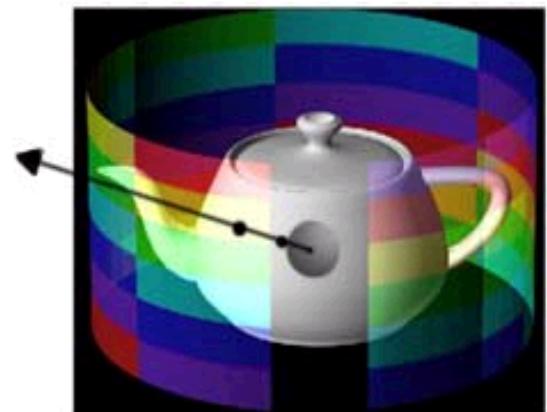
Mapeo automático: mapeo en dos partes

Luego se define como se mapea entre la superficie intermedia y el objeto (O-Mapping):

- **Object normal:** el punto en el cual la normal, partiendo del punto, corte a la superficie intermedia define las coordenadas de textura.
- **Object centroid:** Se calcula una línea entre el centroide del objeto y el punto de la superficie. El punto en el cual esta línea corte la superficie intermedia, definirá las coordenadas de textura.



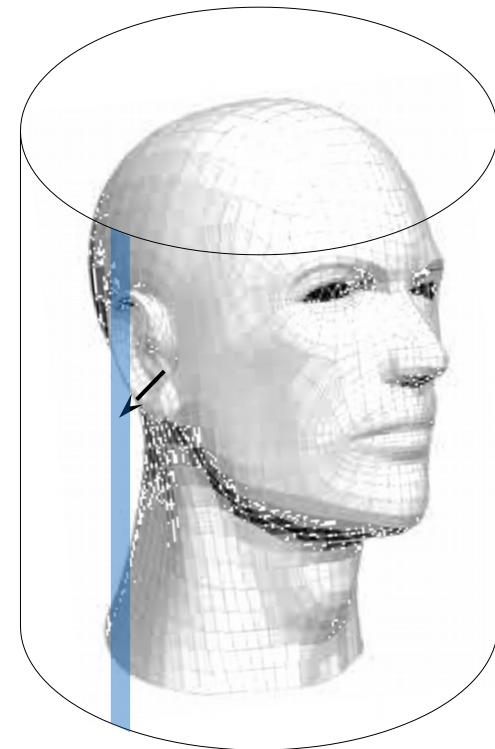
Object normal



Object centroid

Mapeo automático: mapeo en dos partes

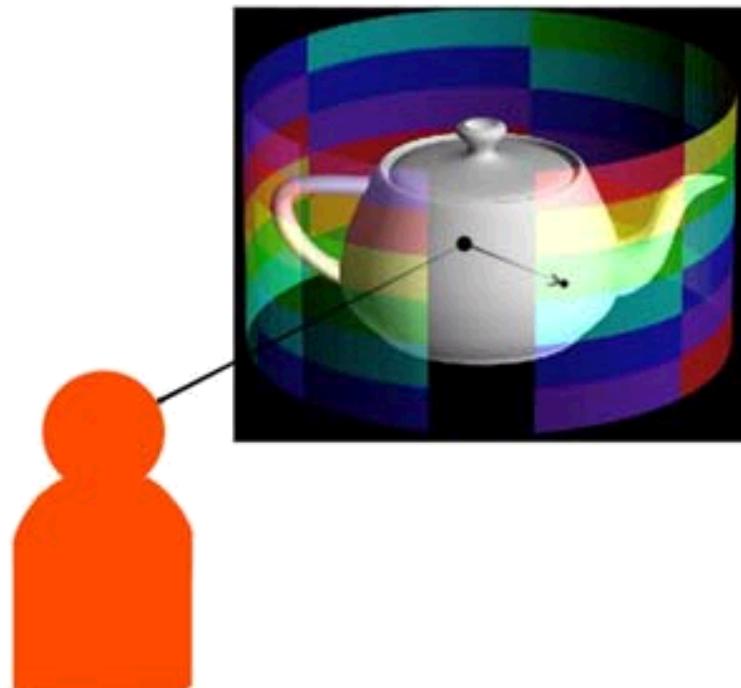
Luego se define como se mapea entre la superficie intermedia y el objeto (O-Mapping):



Mapeo automático: mapeo en dos partes

Luego se define como se mapea entre la superficie intermedia y el objeto (O-Mapping):

- **Reflected ray:** Se traza un rayo desde el punto de vista del observador, el reflejo del rayo sobre el objeto incidirá sobre la superficie intermedia, y así se decide qué puntos de la textura se corresponden con los del objeto.



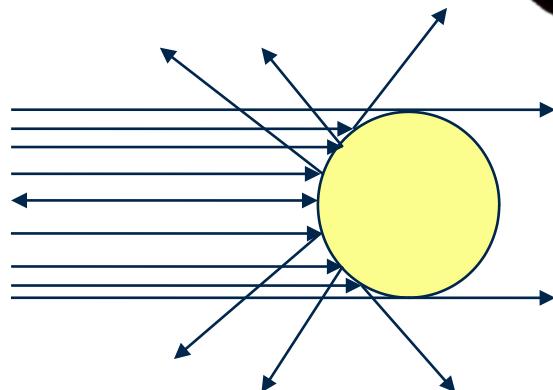
Environment Mapping

Simula los reflejos del ambiente sobre el objeto. Las coordenadas de textura se calculan por reflexión en los vértices de la primitiva.

- **Sphere map:** se define una textura que simula el reflejo en una esfera.



Textura

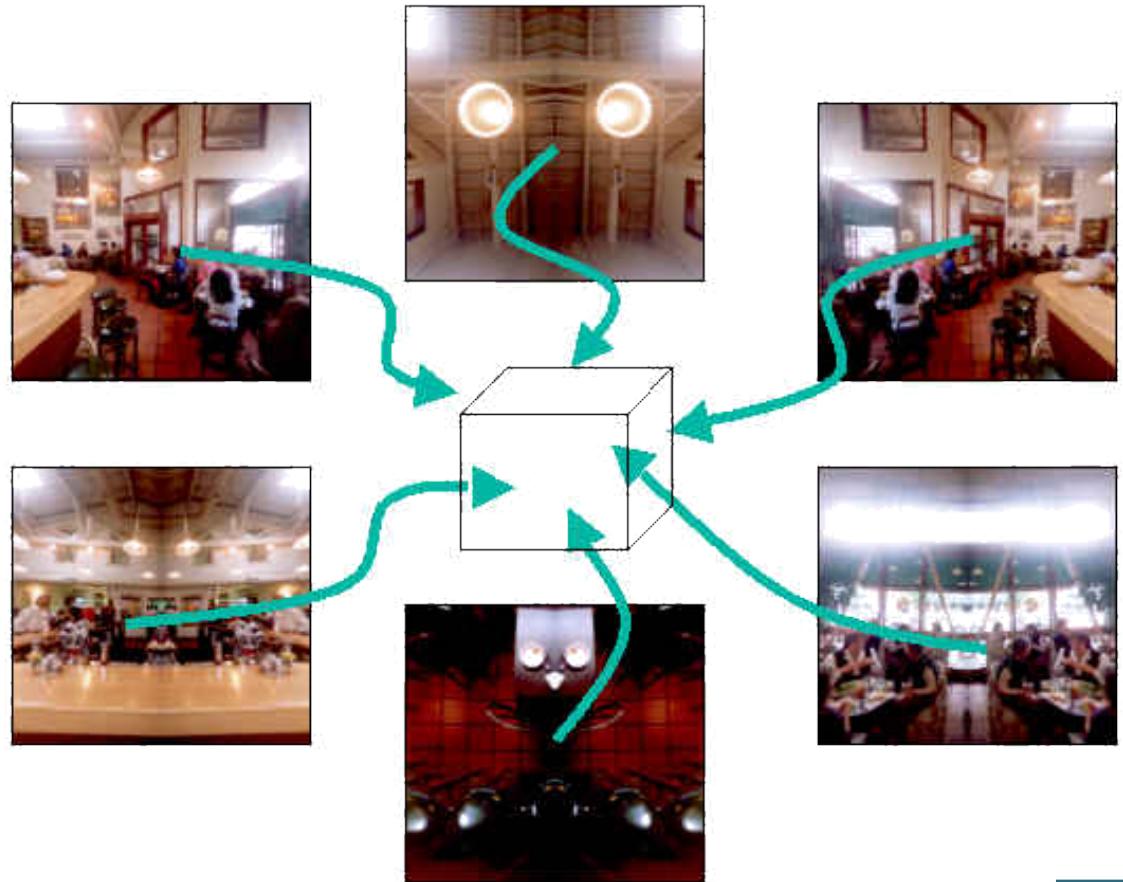
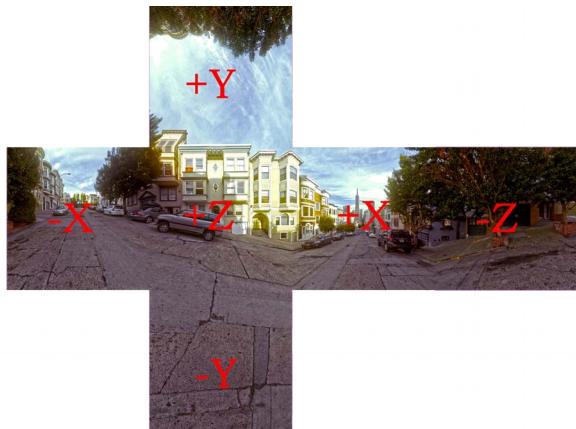


Texturas

Environment Mapping

Simula los reflejos del ambiente sobre el objeto. Las coordenadas de textura se calculan por reflexión en las caras de la primitiva.

- **Cube map:** definiendo las seis imágenes de un cubo que envuelve el objeto.



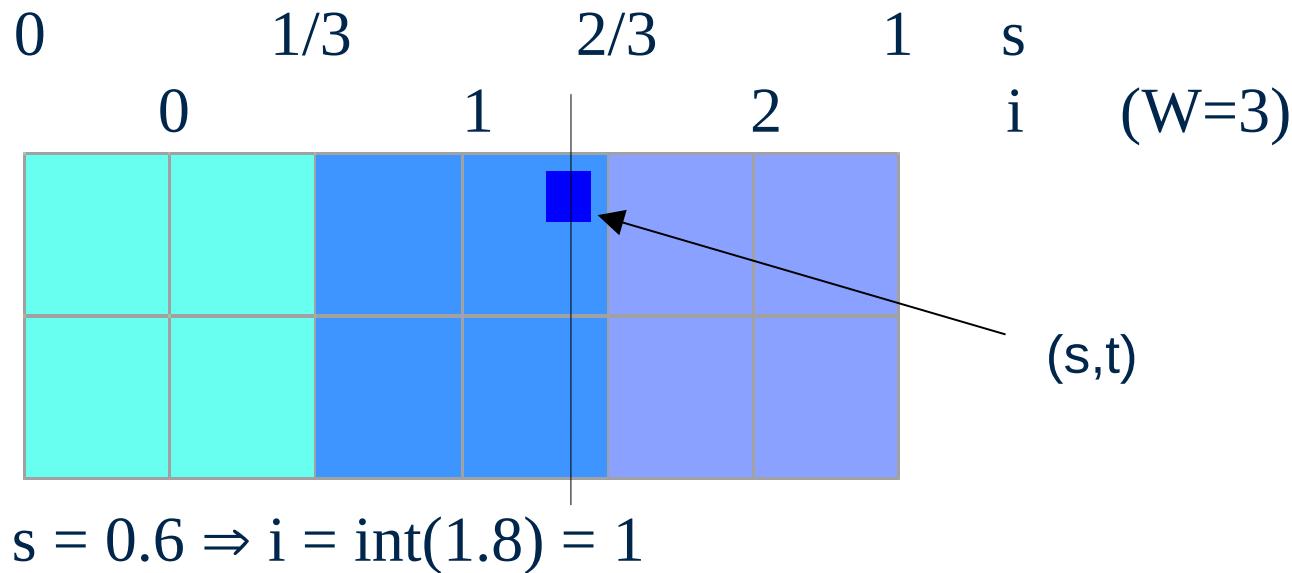
Filtrado

Cada fragmento tiene coordenadas de textura ($strq$) entre 0 y 1.

El punto con esas coordenadas cae en el texel:

$$i = \text{int}(sW), j = \text{int}(tH)$$

¿Qué color de la imagen se asigna al fragmento?



Filtrado

Los píxeles (fragmentos) y los téxeles tienen distinto "tamaño".

- **Magnification:** Cuando el objeto con textura se magnifica o acerca demasiado, la textura aparece pixelada. Un téxel ocupa varios píxeles.
- **Minification:** Cuando el objeto se achica o aleja demasiado, un píxel incluye varios téxeles y la textura puede parecer cortada o interrumpida.

Filtrado

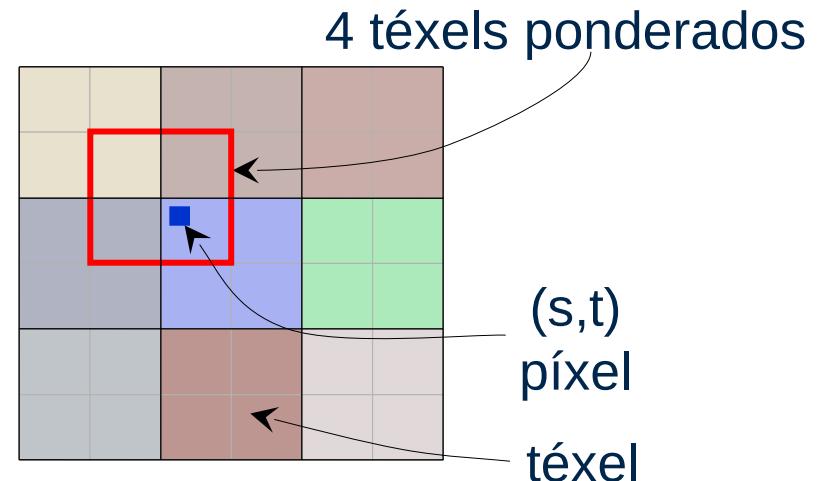
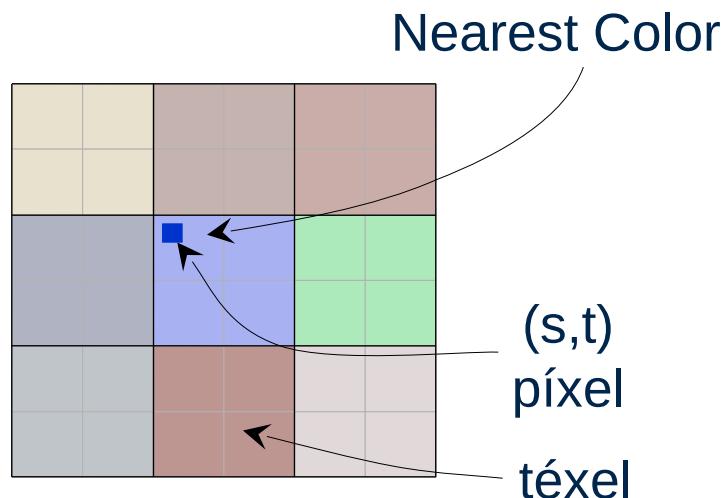
Los píxeles (fragmentos) y los téxeles tienen distinto "tamaño".

- **Magnification:** Cuando el objeto con textura se magnifica o acerca demasiado, la textura aparece pixelada. Un téxel ocupa varios píxeles.
- **Minification:** Cuando el objeto se achica o aleja demasiado, un píxel incluye varios téxeles y la textura puede parecer cortada o interrumpida.

Filtrado: ampliación o magnification

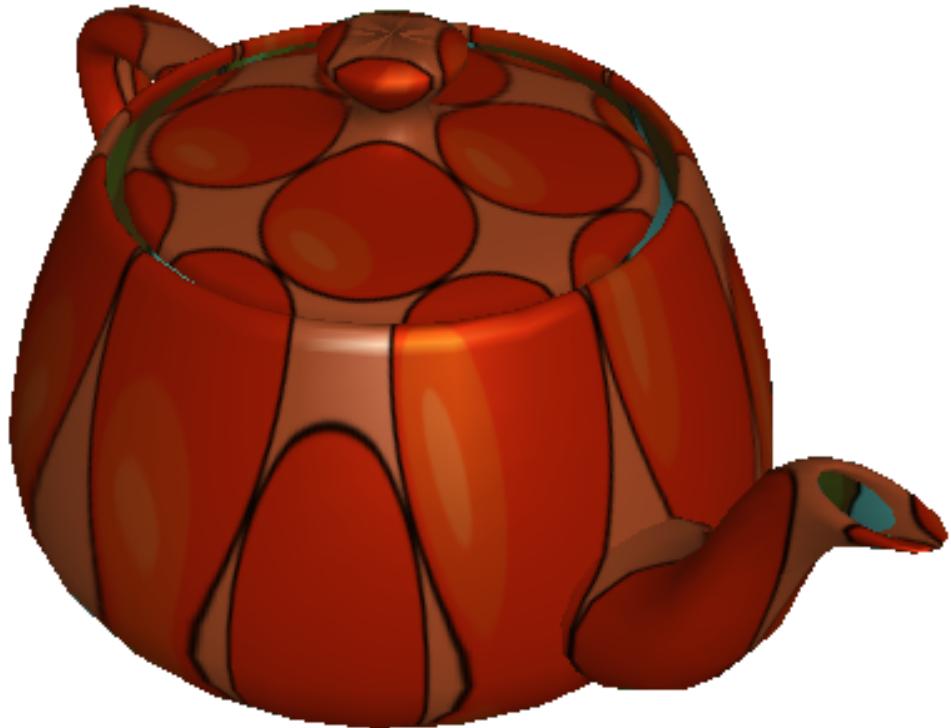
GL_NEAREST: se elige el téxel que contiene las coordenadas de textura del fragmento. Es el filtro por defecto y el menos costoso, pero produce el efecto de pixelado.

GL_LINEAR: Se realiza una interpolación bilineal (promedio ponderado) entre los colores de los cuatro téxeles más cercanos al punto (s,t) . La imagen aparece borrosa, pero elimina un poco el pixelado.



Filtrado: ampliación o magnification

GL_NEAREST vs. GL_LINEAR

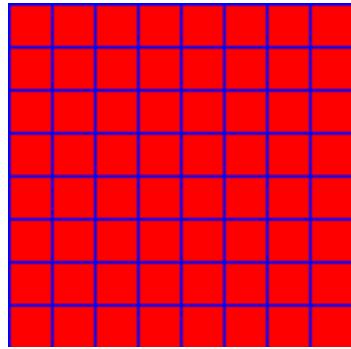


Filtrado: reducción o minification

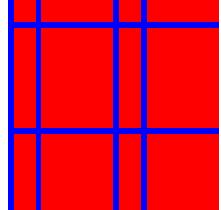
GL_NEAREST: se elige el téxel que contiene las coordenadas (s,t) del fragmento (centro). Si la posición cambia ligeramente se producen los efectos de dentado y fluctuación de la textura.

GL_LINEAR: Interpolación bilineal entre los 2×2 téxeles cercanos. Elimina un poco el dentado (2×2).

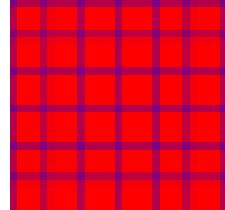
¿Qué otra cosa se podría hacer?



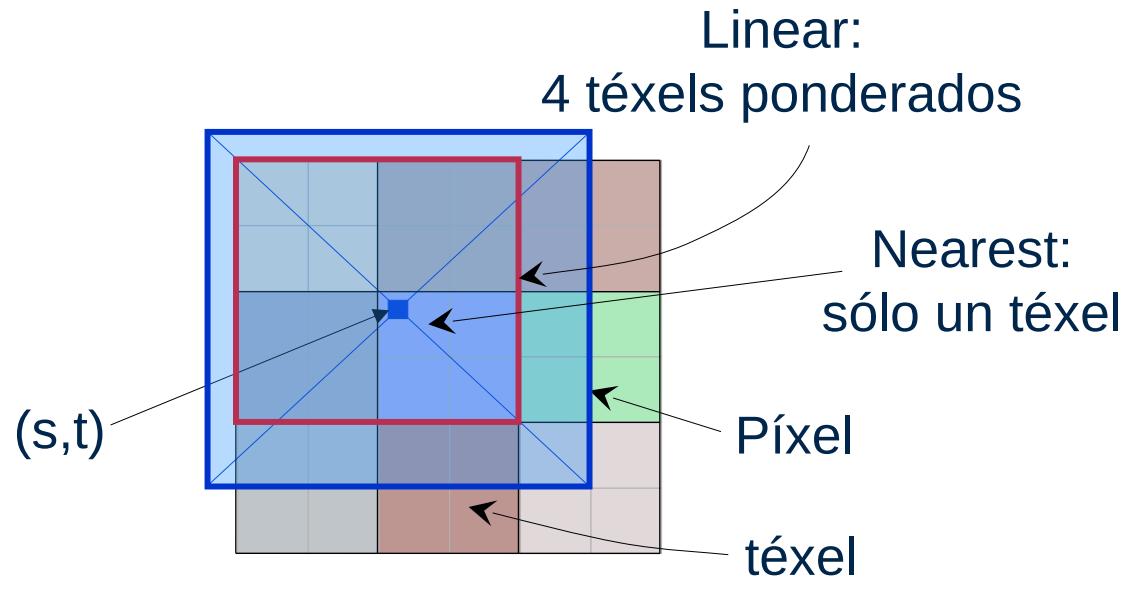
Textura



Nearest

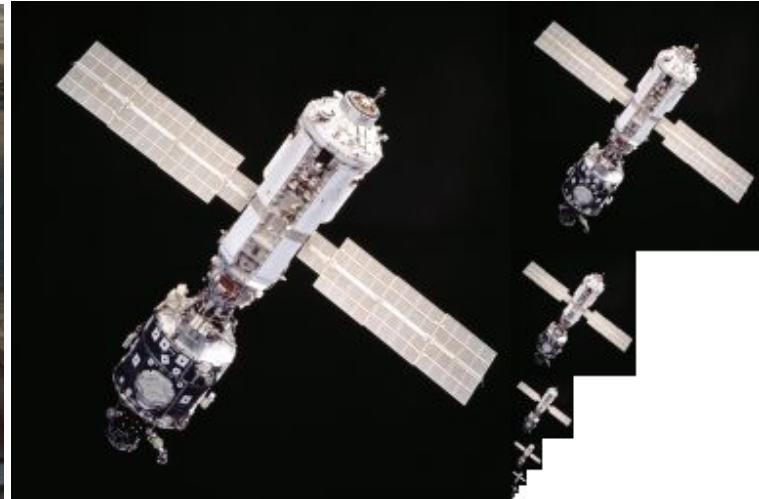


Linear

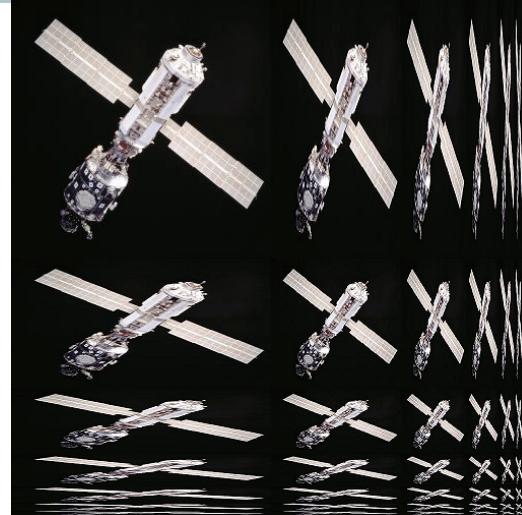
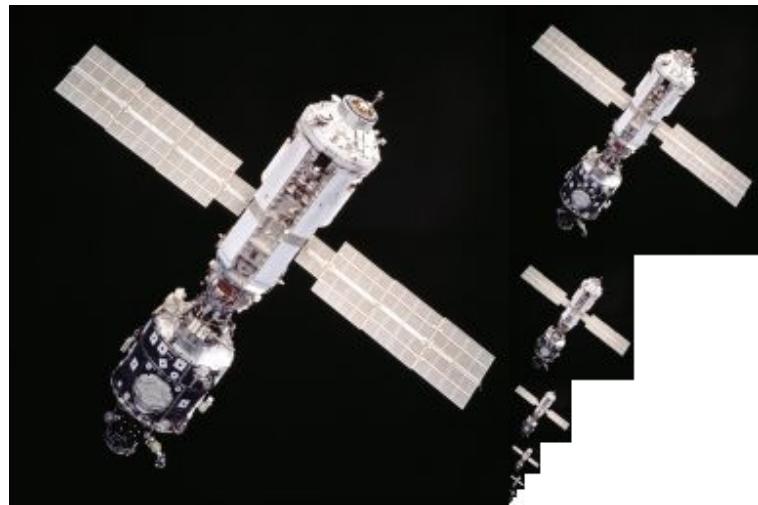


Filtrado: reducción o minification

- Para interpolar más píxeles se utilizan los **Mipmaps**.
- Versiones reducidas de la misma textura, con resoluciones decrecientes en potencias de 2 hasta 1x1.
- Las sucesivas reducciones a la mitad se van promediando y cada nivel reducido promedia promedios.
- Tiene un consumo mayor de memoria (<150%).



Filtrado: reducción o minification



Isotrópico



Anisotrópico

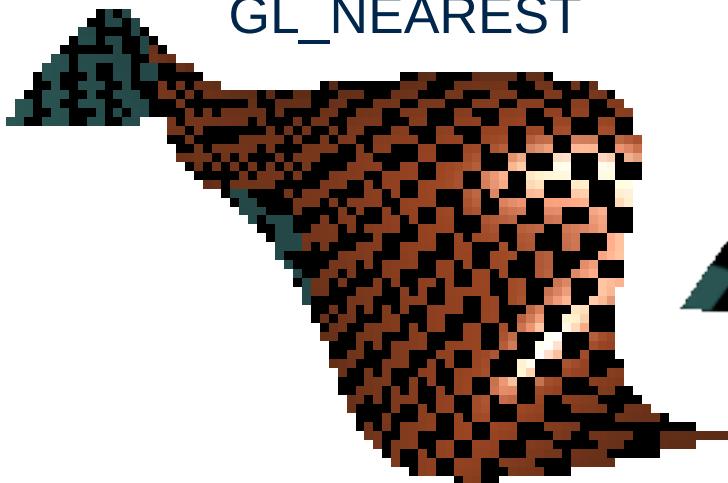
Filtrado: reducción o minification

Se usa la función `glTexParameter()` para establecer más opciones en los **filtros de minimización**.

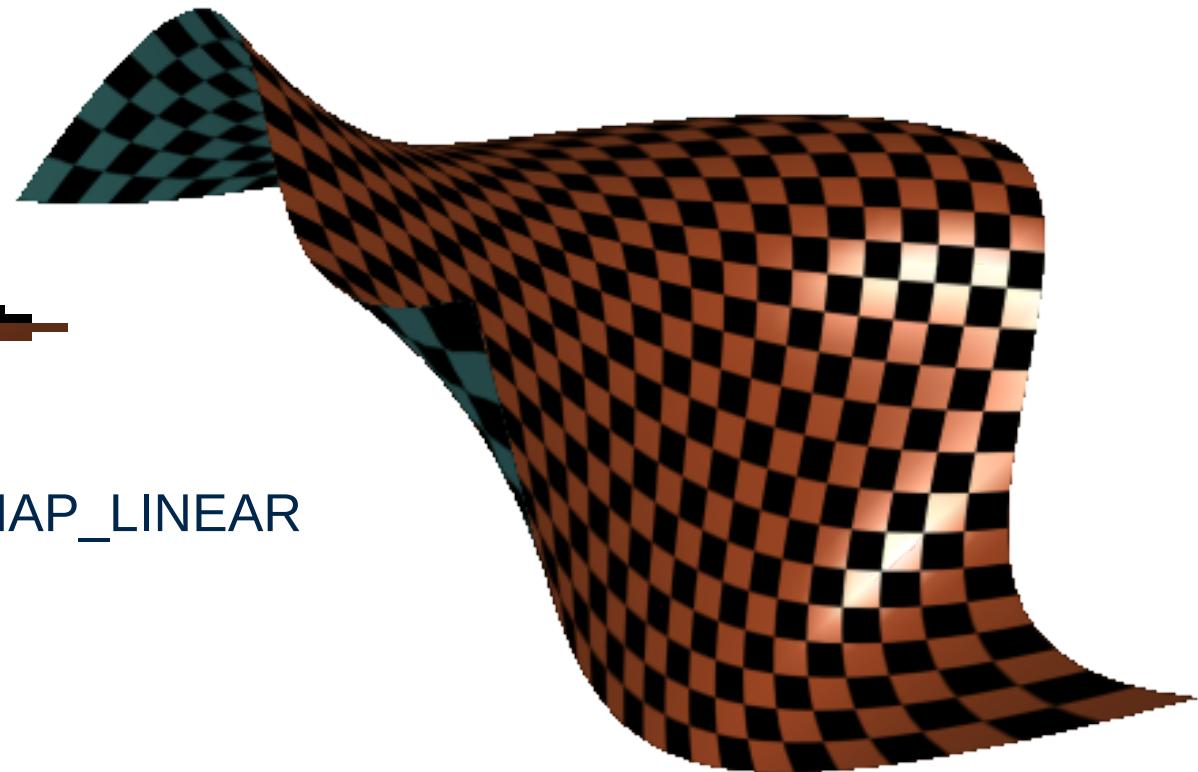
| Filtro | Descripción |
|--|--|
| <code>GL_NEAREST_MIPMAP_NEAREST</code> | Utiliza la imagen más cercana a la resolución de la pantalla. Usaremos el filtro <code>GL_NEAREST</code> cuando usemos este mapa. |
| <code>GL_LINEAR_MIPMAP_NEAREST</code> | Utiliza la imagen más cercana a la resolución de la pantalla. Usaremos el filtro <code>GL_LINEAR</code> cuando usemos este mapa. |
| <code>GL_NEAREST_MIPMAP_LINEAR</code> | Interpolan linealmente entre las dos imágenes más cercanas a la resolución de la pantalla. Usaremos el filtro <code>GL_NEAREST</code> cuando utilicemos este mapa. |
| <code>GL_LINEAR_MIPMAP_LINEAR</code> | Interpolan linealmente entre las dos imágenes más cercanas a la resolución de la pantalla. Usaremos el filtro <code>GL_LINEAR</code> cuando utilicemos este mapa. |

Filtrado: reducción o minification

GL_NEAREST

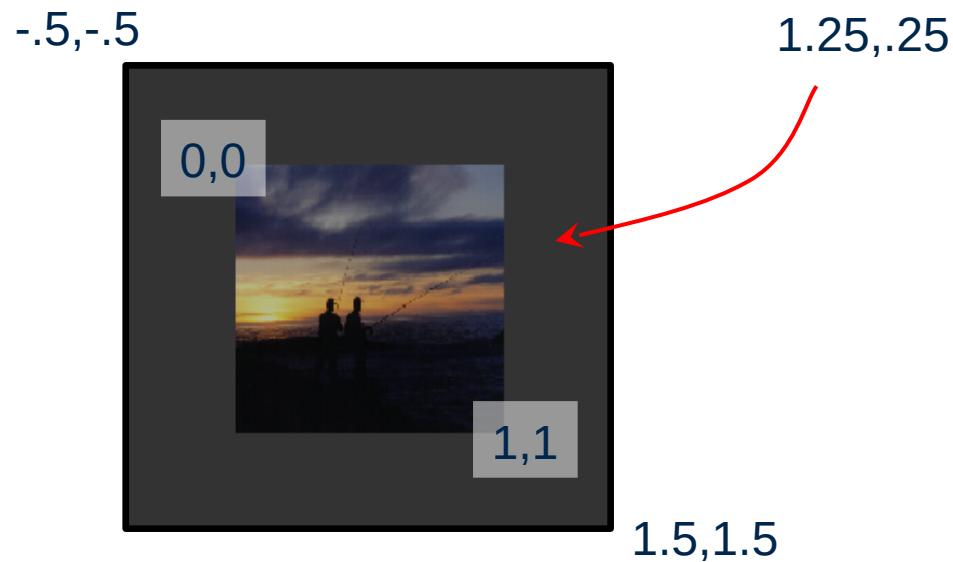


GL_LINEAR_MIPMAP_LINEAR



Wrapping

Cada fragmento ya tiene coordenadas de textura, pero son números reales que no están entre 0 y 1



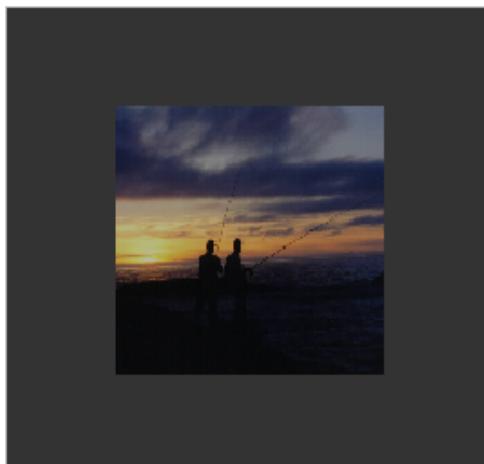
Wrapping: clamp

Las coordenadas de textura se reasignan a [0,1] según el modo de **wraping**.

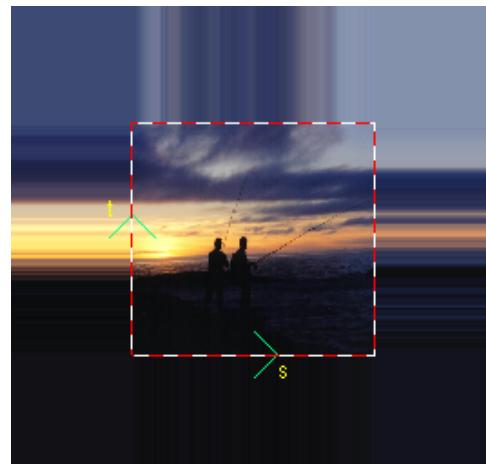
Clamp: Repite los téxels de los bordes. (ej.: Calcomanía)
if ($s > 1$) $s = 1$; if ($s < 0$) $s = 0$;

En OpenGL:

```
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_CLAMP);  
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_CLAMP);
```



Textura



Aplicación con Clamp

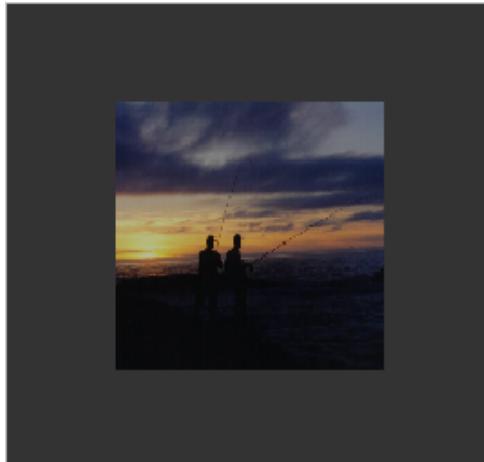
Wrapping: repeat

Las coordenadas de textura se limitan a [0,1] según el modo de **wraping**.

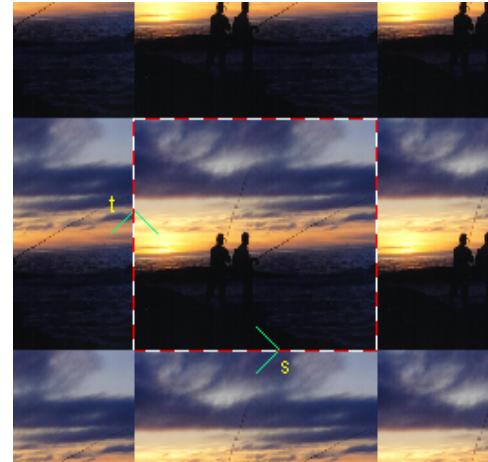
Repeat: Repite la textura completa. (ej.: ladrillos, tela)
 $s = s - \text{int}(s);$

En OpenGL:

```
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_REPEAT);  
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_REPEAT);
```



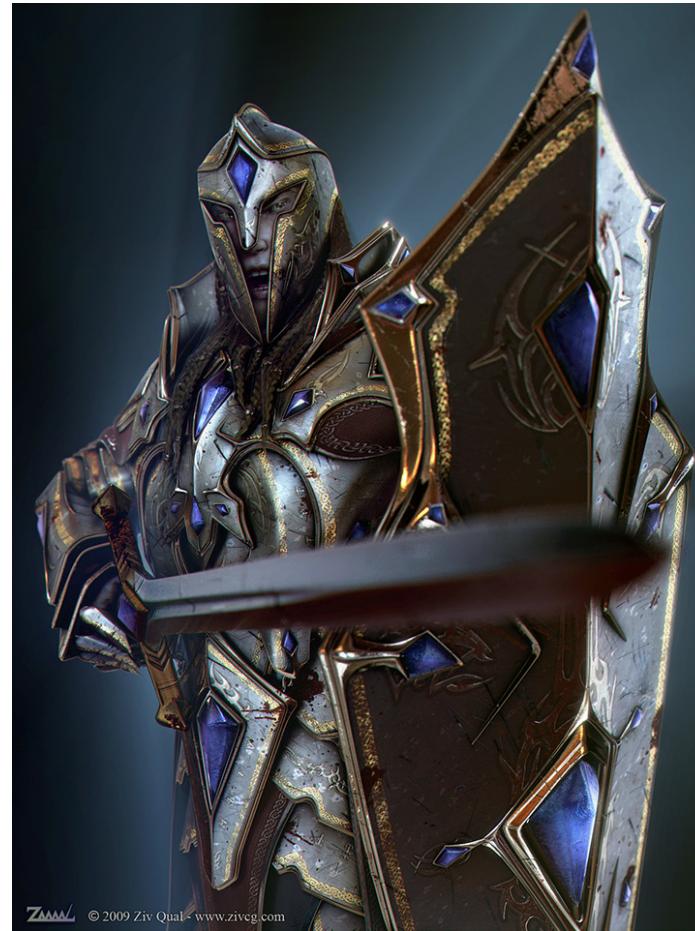
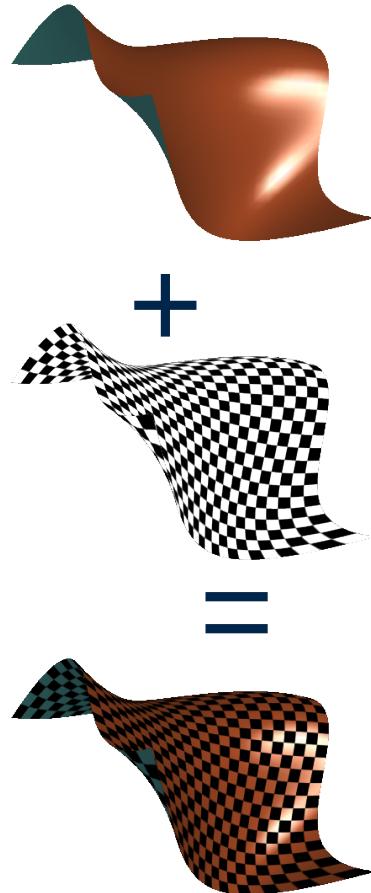
Textura



Aplicación con Repeat

Modos de mezcla

Ya hay un color de textura para el fragmento.
¿Qué hacemos con el color que ya tenía?



Modos de mezcla

- **Replace:** El fragmento recibe el color y alpha de la textura.
Ejs.: césped, pared, entorno sobre cromado, alambrado, rejas.
- **Decal:** Textura con alpha (sin alpha = Replace). Se mezclan el color del fragmento de objeto (C_f) y el de la textura (C_t), en proporciones definidas mediante el alpha de la textura (A_t).
El alpha del fragmento no cambia.
Ejs.: calcomanía con partes opacas/transparentes/invisibles

| Base Internal Format | Decal Texture Function |
|----------------------|---------------------------------------|
| GL_RGB | $C = C_t,$ $A = A_f$ |
| GL_RGBA | $C = C_f(1-At) + C_tAt,$ $A = A_f$ |

Modos de mezcla

- **Modulate:** Oscurece la superficie. Multiplica cada color del objeto por el de la textura ($C_t \in [0,1] \Rightarrow C \leq C_f$).
También puede modular la transparencia (alpha) hasta hacer invisibles algunas partes del objeto (fuego, humo, nube).
Multiplicar colores sirve para ciertos efectos especiales

| Base Internal Format | Modulate Texture Function |
|----------------------|---------------------------------|
| GL_RGB | $C = C_f C_t,$ $A = A_f$ |
| GL_RGBA | $C = C_f C_t,$ $A = A_f A_t$ |

Modos de mezcla

- **Blend:** El color resultante es una mezcla del color de la superficie (C_f) y otro que define el usuario (C_c), predominando más uno u otro dependiendo del color de la textura (C_t), que opera como parámetro de interpolación. C_c se define mediante

```
glTexEnvfv(GL_TEXTURE_ENV, GL_TEXTURE_ENV_COLOR, Cc);
```

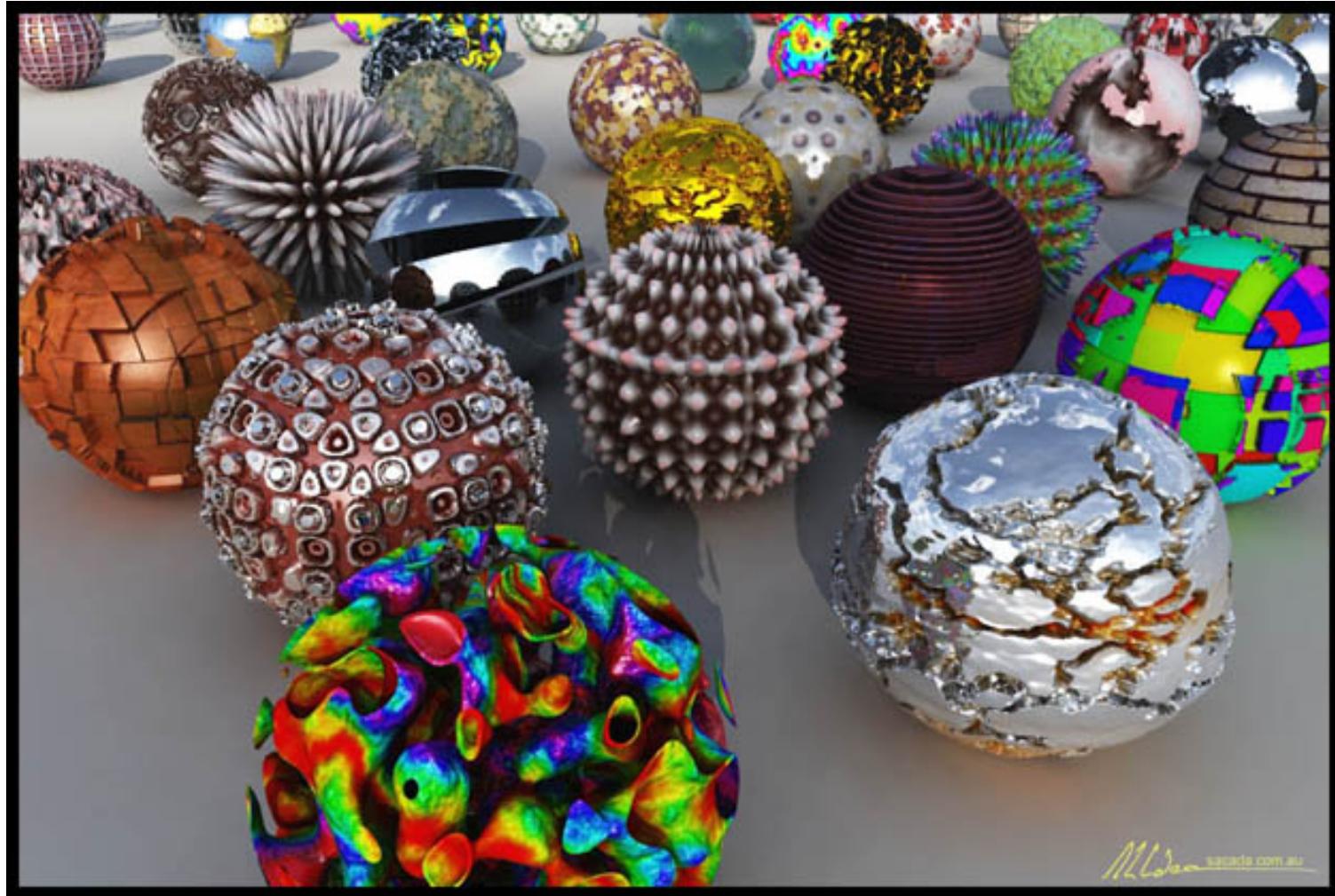
R, G y B (no A), modulan entre el color actual y C_c .

Sirve para oxidar, vetear o marmolar una superficie.

| Base Internal Format | Blend Texture Function |
|----------------------|--|
| GL_RGB | $C = C_f(1-C_t) + C_cC_t,$ $A = Af$ |
| GL_RGBA | $C = C_f(1-C_t) + C_cC_t,$ $A = AfAt$ |

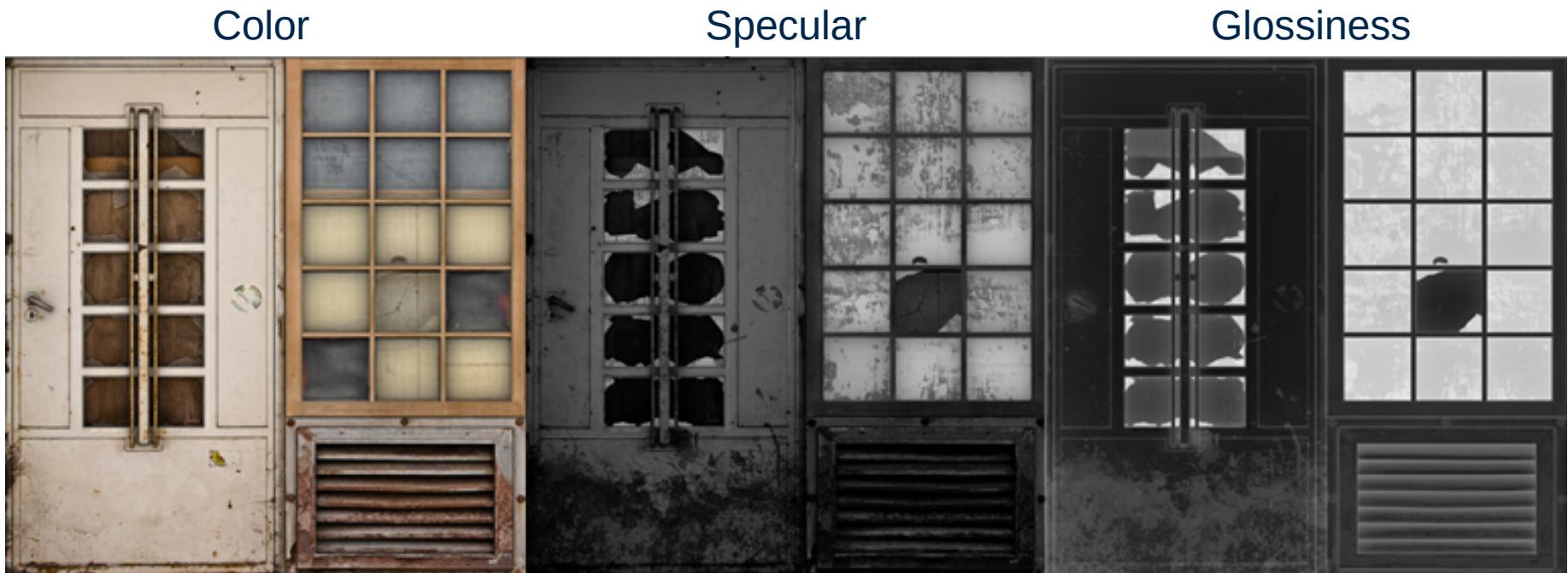
Efectos especiales

Las texturas se pueden utilizar para otra información ademas de RGBA



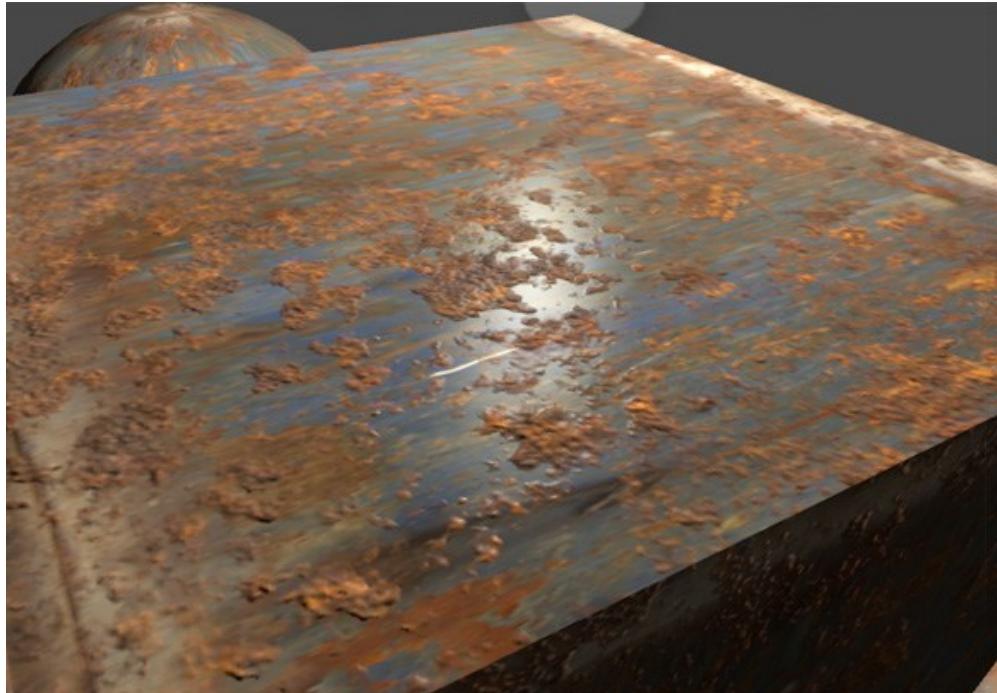
Efectos especiales: gloss y specular maps

- **Gloss map:** textura que indica que tan pulida es la superficie para en fragmento. Se utiliza como \mathbf{q} en el modelo de Phong.
- **Specular map:** textura que indica que tan brillosa o metálica es la superficie en cada fragmento. Multiplica el término especular.



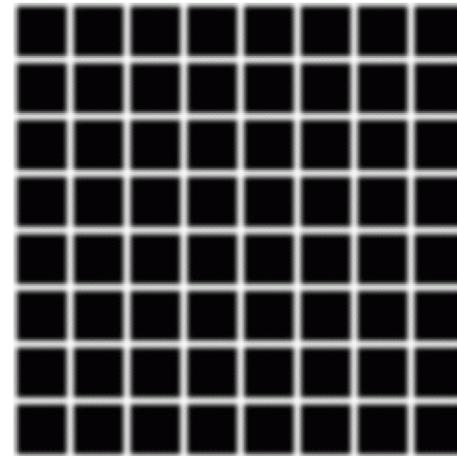
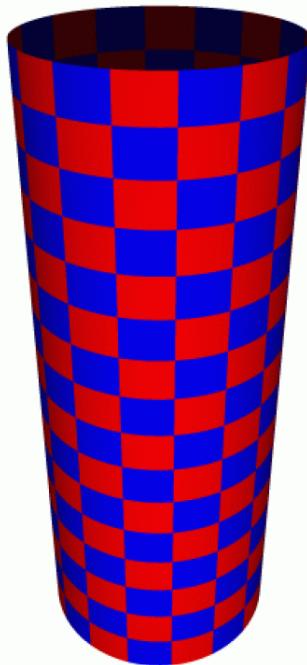
Efectos especiales: gloss y specular maps

- **Gloss map:** textura que indica que tan pulida es la superficie para en fragmento. Se utiliza como **q** en el modelo de Phong.
- **Specular map:** textura que indica que tan brillosa o metálica es la superficie en cada fragmento. Multiplica el término especular.



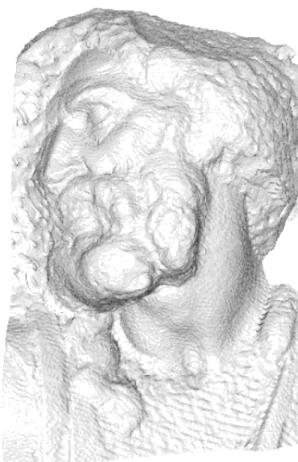
Efectos especiales: Bump Mapping

- La geometría del objeto no es afectada, solo se modifican sus vectores normales alterando su dirección para simular depresiones y picos en su superficie.
- La textura actúa como campo de alturas: las derivadas parciales de la intensidad modifican las normales de cada fragmento.

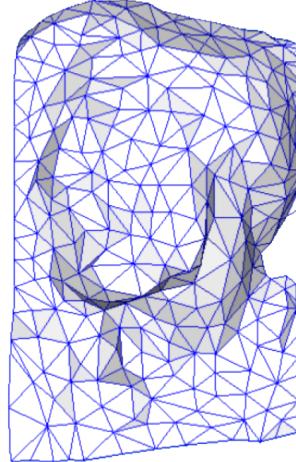


Efectos especiales: Normal Mapping

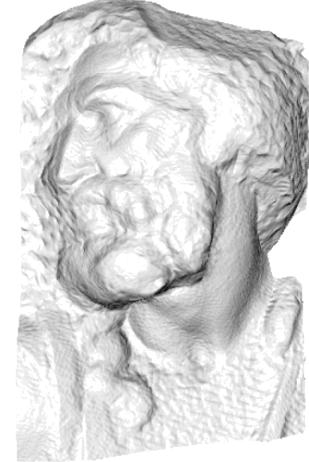
La textura contiene la normal a asignar en cada fragmento, medida en el “espacio tangente” de la primitiva.



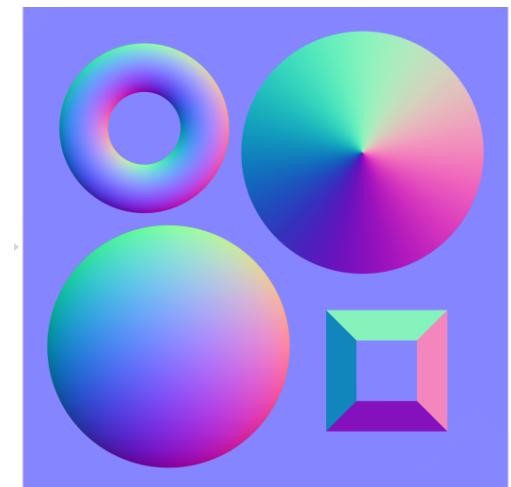
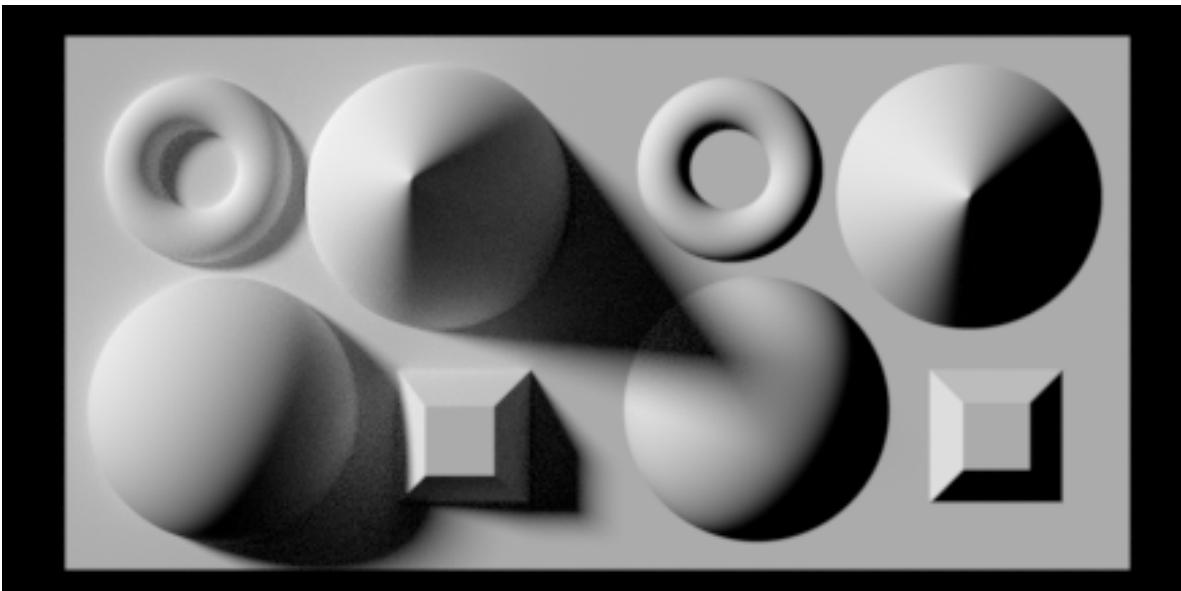
original mesh
4M triangles



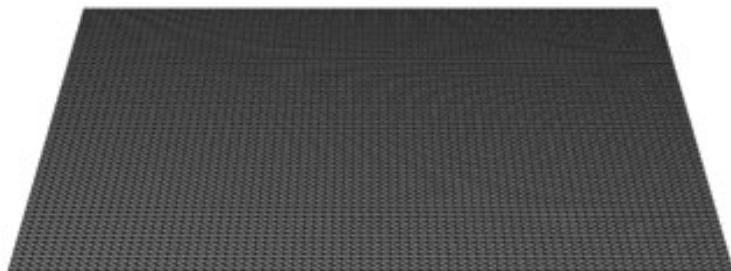
simplified mesh
500 triangles



simplified mesh
and normal mapping
500 triangles



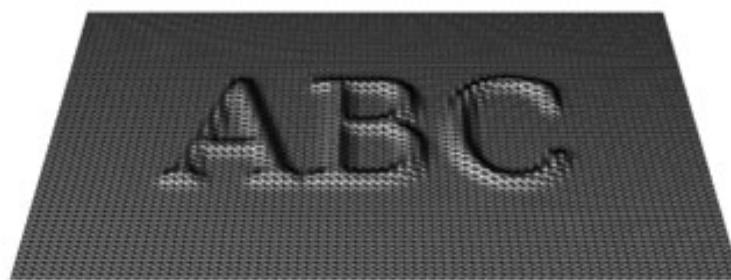
Efectos especiales: Displacement Mapping



ORIGINAL MESH



DISPLACEMENT MAP



MESH WITH DISPLACEMENT

Utiliza la textura para desplazar los vértices originales.



Bibliografía

- OpenGL reference para comprender que hace cada comando.
- Red-Book: capítulo 9 (Texture Mapping)
- Rosalee Wolfe - DePaul Univ. -
Teaching Texture Mapping Visually
- David Green - Drexel Univ - lecture 7