


Texturas

Módulo II: Aumento de realismo

- Alan Watt. *3D Computer Graphics*. Addison-Wesley. Third Edition. 2000
- David S. Ebert y otros. *Texturing & Modeling. A Procedural Approach*. Morgan Kaufmann. Third Edition. 2003
- ...

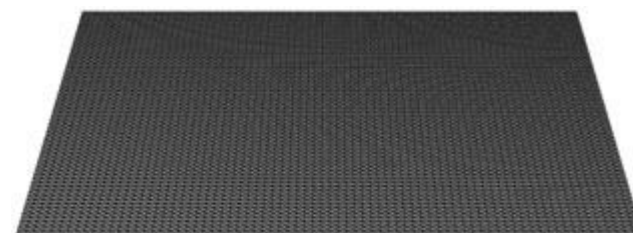
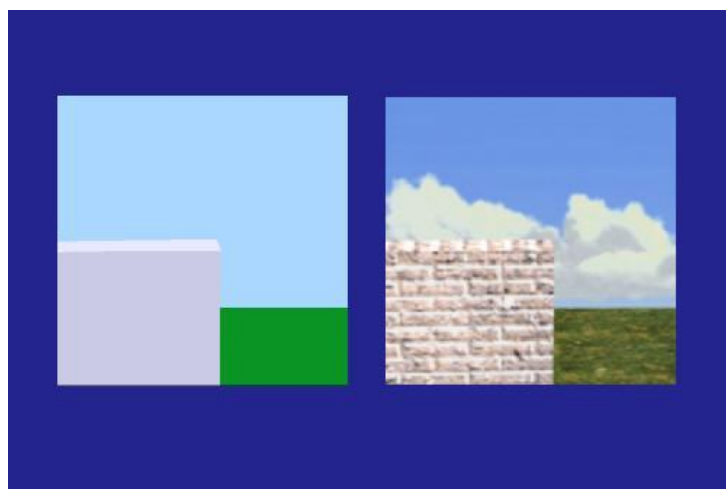
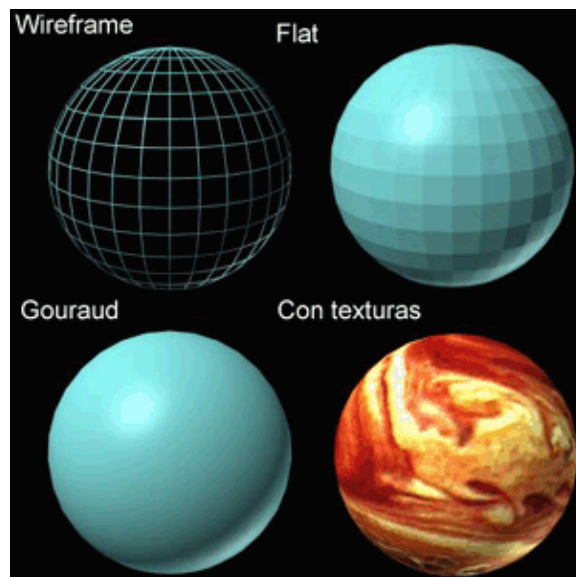
- Introducción
- Clasificación
 - Según el número de dimensiones
 - Según el tipo de información que almacenan
- Mapeado
- Dificultades
- Texturas procedurales
- Compresión
- Efectos

- Comúnmente, entendemos que una textura es un imagen 2D que  representa una propiedad de un superficie (generalmente el color – color map)
 - “The term texture is a somewhat confusing term in computer graphics and generally does not mean controlling the small-scale geometry of the surface of a computer graphics object, which is the normal meaning of the word”
- Cuando se trabajar en tiempo real:
 - se usan en la etapa de rasterizado (principalmente)
 - y codifican información de un fragemento
- Mapa de texturas: colección o array n-dimensional de valores que se **mapea** sobre otra estructura, generalmente una superficie.

- La utilización de texturas:
 - Permiten definir propiedades de la malla. Aumenta el realismo de la imagen sin aumentar número de polígonos
 - Renderizado más eficiente
 - Modelado más eficiente
 - Proporciona una clave para la percepción de la profundidad.
 - Permite simular multitud de efectos: sombras, reflejos, entornos, ...

- GPGPU

G3D



ORIGINAL MESH



DISPLACEMENT MAP



MESH WITH DISPLACEMENT

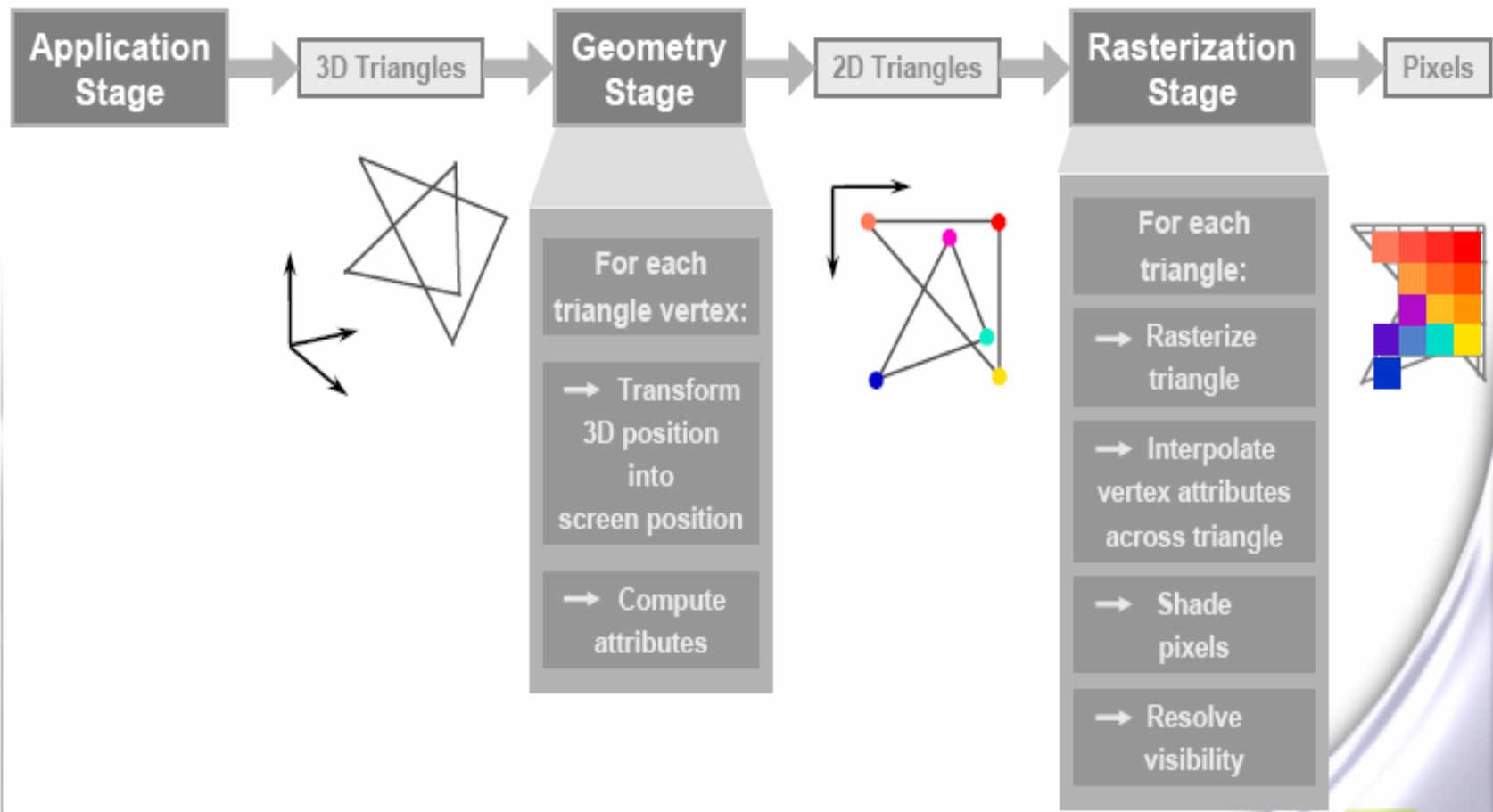
- 1974. Ed Catmull utiliza mapas de texturas por primera vez.
- 1976. Blinn y Newell introducen los mapas de reflexión o de entorno.
- 1978. Blinn introduce los mapas de bollos
- 1984. Cook los extiende creando los mapas de desplazamiento

- 1985. Peachey y Perlin describen las texturas sólidas como alternativa a las tradicionales texturas 2D.
- 1987. Reeves, Salesin y Cook: algoritmo para generación de sombras sin aliasing utilizando texturas.

...

Pipeline Gráfico (III)

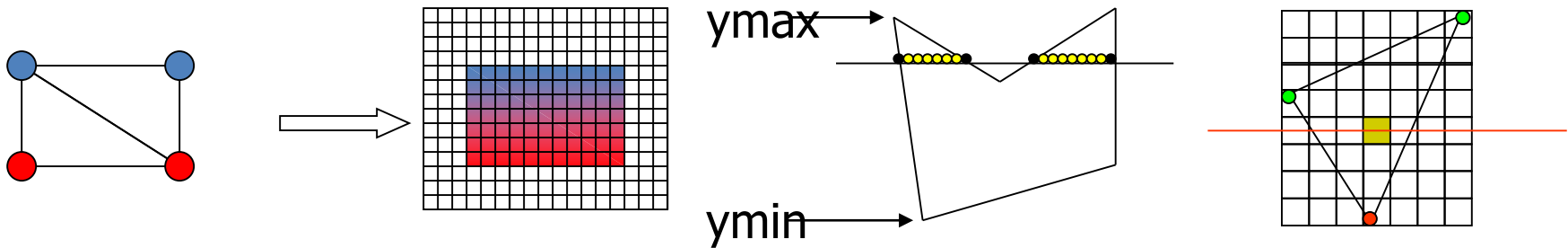
G3D



Pipeline Gráfico (IV)

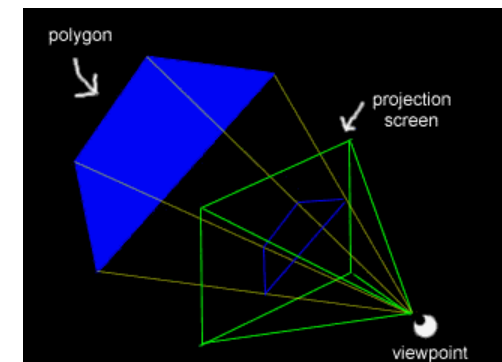
G3D

- Cálculo de los atributos de un fragmento



- Las interpolaciones deben tener en cuenta la perspectiva. Incluso las coordenadas de textura)
- En una textura se podría almacenar información de estos atributos.
 - Ahorro de tiempo de cómputo de los atributos de un fragmento
 - Se necesitan generar menos vértices

- **Cpu**
 - La aplicación genera una serie de primitas que definen la geometría del entorno virtual, las propiedades de dichas primitivas, el tipo de proyección, la iluminación ...
- **Etapas de vértices**
 - Se realizan operaciones de forma independiente sobre cada vértice:
 - Transformación de coordenadas de modelado a las coordenadas del mundo virtual (GL_MODELVIEW)
 - **Transformación o generación de las coordenadas de textura (GL_TEXTURE)**
 - Proyección (GL_PROJECTION)
 - Transformación de las normales, luces
 - Cada vértice tiene asociada mucha información: normal, color (ambiental, difuso, especular), coordenadas de textura
 - Al final de esta etapa todos los atributos de los vértices quedan determinados
 - Ejemplo modificación del color base con la iluminación
 - Programable




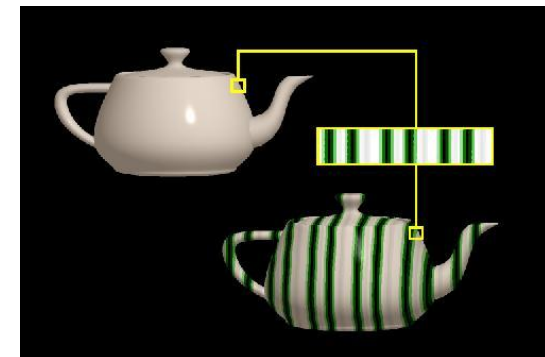
Pipeline Gráfico (II)



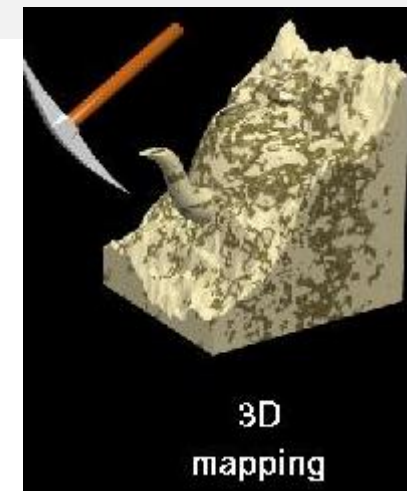
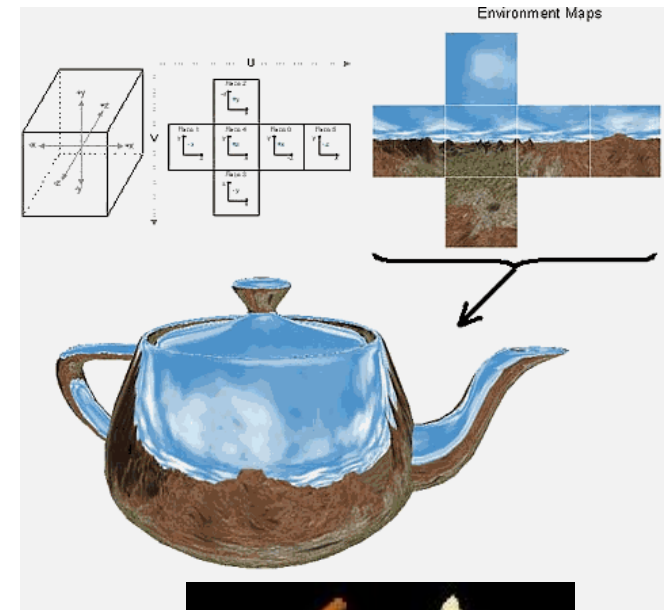
G3D

- Reensamblado
 - Los vértices se vuelven a agrupar formando primitivas (puntos, líneas, polígonos)
- Recorte
 - Todas las primitivas que están fuera del volumen de vista se ignoran.
- Rasterización y operaciones con los fragmentos
 - Transformación de números en coma flotante a números enteros
 - La unidad de operación es el fragmento (más que un píxel)
 - Pasos
 - El rasterizado genera los fragmentos
 - **Texturizado: Se accede a la textura** (distintos algoritmos).
 - Se procesa cada fragmento de forma individual
 - Posteriormente se opera con ellos.
 - Fragmento
 - **Color**
 - **Normal**
 - **Transparencia**
 - **Profundidad**
 - Coordenadas de textura
 - La mayoría de estos valores se obtiene a partir de los vértices que forman la primitiva.
 - Programable
- Operaciones sobre el framebuffer
 - Operaciones sobre los buffers generados en la etapa de rasterizado.
 - Ejemplo: Algún tipo de filtrado.

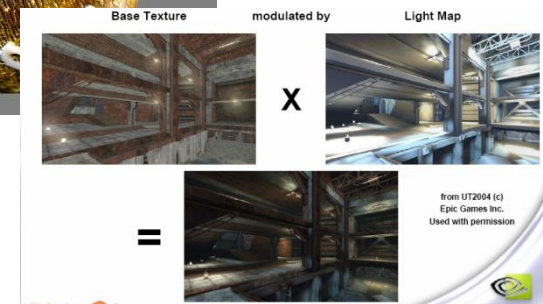
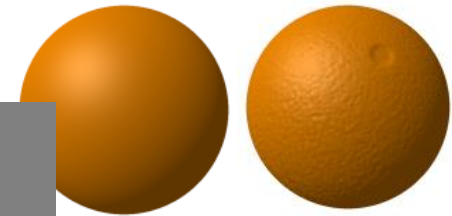
- Según el número de dimensiones
 - Unidimensionales 
 - Poco usadas
 - Bandas que varían en una sola dirección
 - Bidimensionales
 - Las más frecuentes
 - Suelen ser cuadradas y potencias de 2
 - No tienen porque ser estáticas
 - Vídeos
 - Funciones matemáticas
 - Texturas procedurales
 - Pueden generarse dinámicamente
 - [Ejemplo](#)



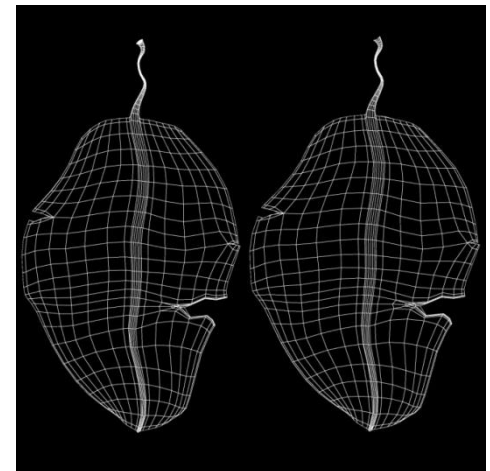
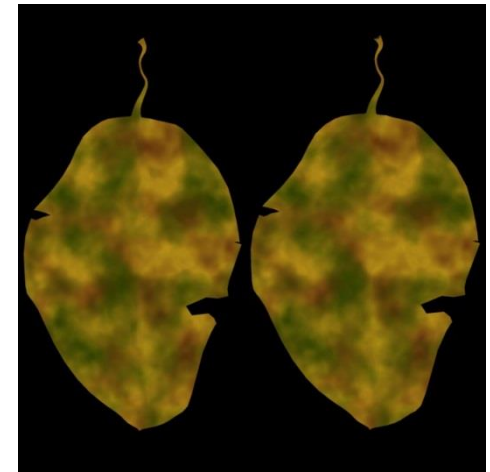
- Según el número de dimensiones
 - Texturas cúbicas
 - Extensión de las texturas bidimensionales
 - Muy usadas en mapeos de entornos.
 - [Ejemplo](#)
- Tridimensionales
 - Permiten mostrar datos volumétricos



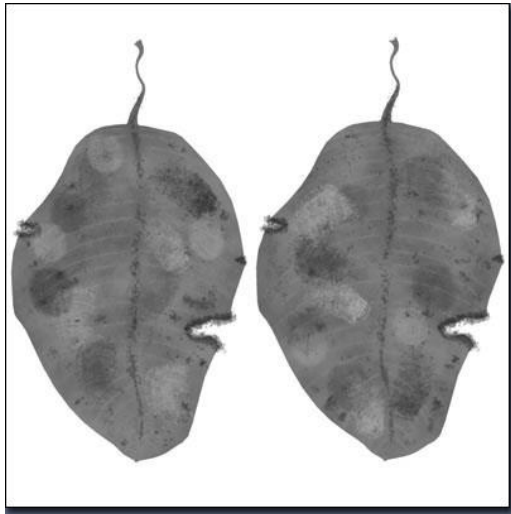
- Tipo de información que almacenan
 - Color
 - Iluminación
 - Transparencia
 - Normales
 - Geométrica
 - ...



- Definen cómo el objeto refleja la luz difusa.
- Son las más utilizadas.
- Contiene el color base del mismo.
- Albedo: porcentaje de intensidad de luz reflejado por una superficie. No debe contener ningún tipo de iluminación. No solo hace referencia al coeficiente de reflexión difuso



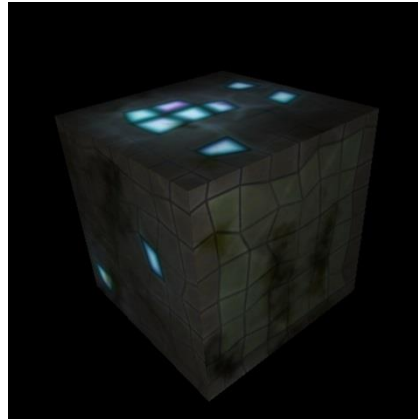
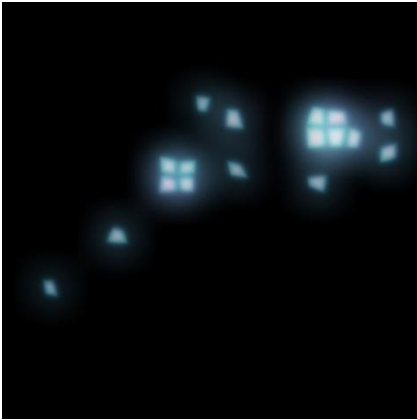
- Definen propiedades del objeto sobre cómo refleja la luz especular.
- La reflexión de la luz no es igual en toda la superficie de un objeto: suciedad, humedad, huella dactilar,...



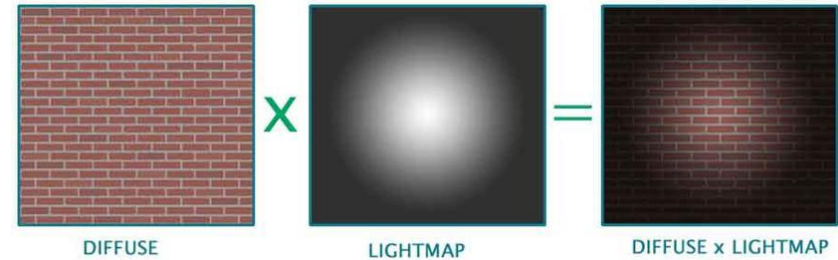
Color: mapas emisivos

G3D

- Simulan objetos que emiten luz.



- Lightmapping:
 - El nº de luces aumenta la complejidad del sombreado. Es de las etapas más costosas
 - Podemos utilizar técnicas básicas o métodos de iluminación global que nos permitan simular fenómenos más complejos como sombras



» Sun Shadow channel

Iluminación: mapas de luz y de sombras

REY JUAN CARLOS UNIVERSITY -- ATCCCIA DEPARTMENT

G3D



» Sun Shadow channel



Sombras pre-calculadas
(sombras suaves)



Sombra en tiempo real

- Ventajas:
 - Ahorro en tiempo y cómputo.
 - Podemos utilizar técnicas globales.
- Limitaciones:
 - Fuentes de luz fijas.
 - Objetos estáticos.
- Las luces y los objeto dinámicos se pueden añadir después.

■ Ambient Occlusion:



Original model



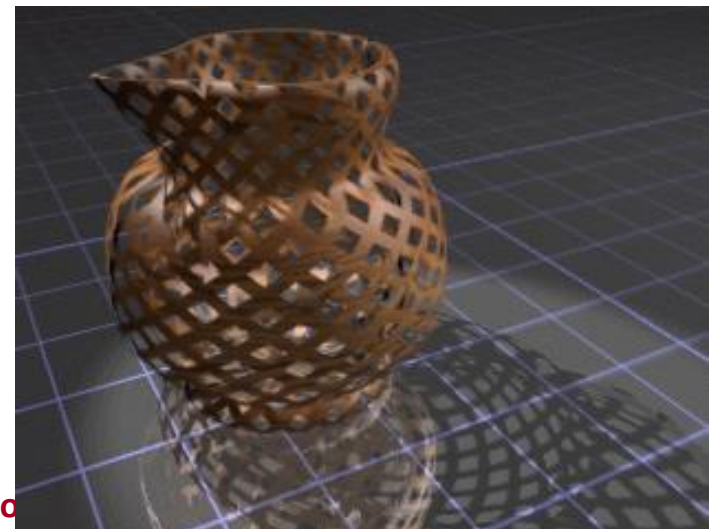
With ambient occlusion



Extracted ambient occlusion map

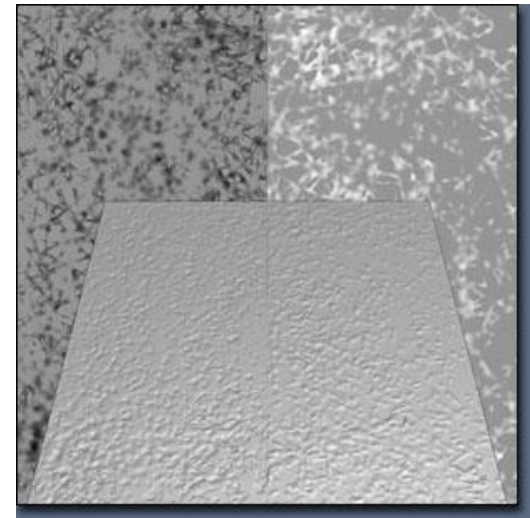
- “Complementario”: mapas de opacidad.
- $\alpha=1$ objeto opaco; $\alpha=0$ objeto transparente.
- La información de transparencia se almacena en escala de grises

Utiliza el mismo mapa como textura especular



- Modificación de normales para dar sensación de relieve.
 - Mapas de profundidad
 - Mapas de normales
- Ejemplo
- Ejemplo [[sin](#)/[con](#)/[modelos](#)]

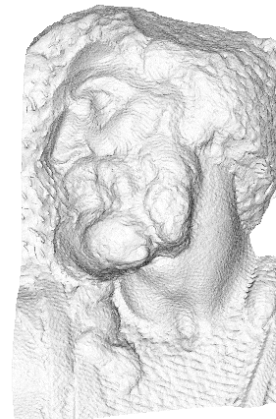
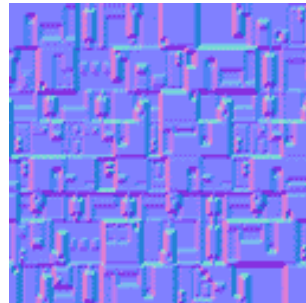
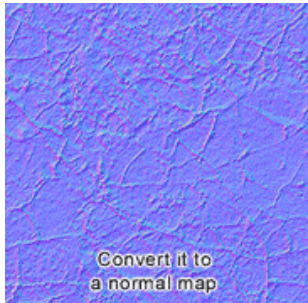
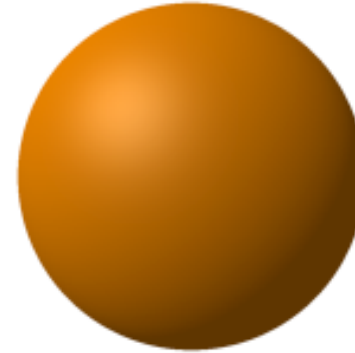
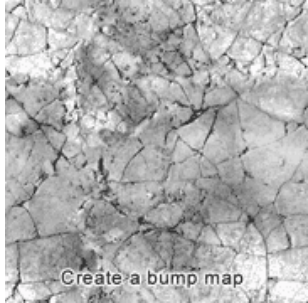
- Mapas de profundidad
 - El mapa de bollos almacena la profundidad del fragmento.
 - Texturas en escala de grises.



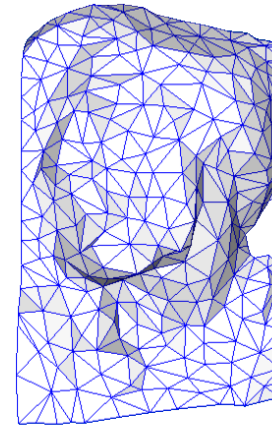
- Mapas de normales
 - Texturas RGB ($R \rightarrow X$, $G \rightarrow Y$, $B \rightarrow Z$)
 - Se almacenan los vectores perpendiculares a la superficie:
 - Vector perpendicular $(127, 127, 255) \rightarrow (0, 0, 1)$
 - Tonos azules: vectores normales.
 - Tonos rojos: variación horizontal.
 - Tonos verdes: variación vertical.
 - Los mapas de bollos pueden transformarse en mapas de normales

Normales: Mapas de bollos

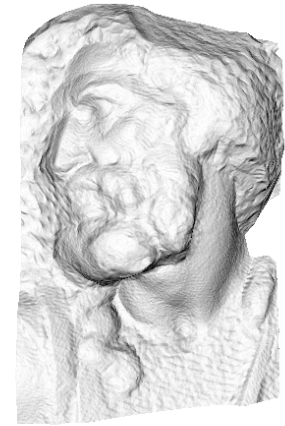
G3D



original mesh
4M triangles



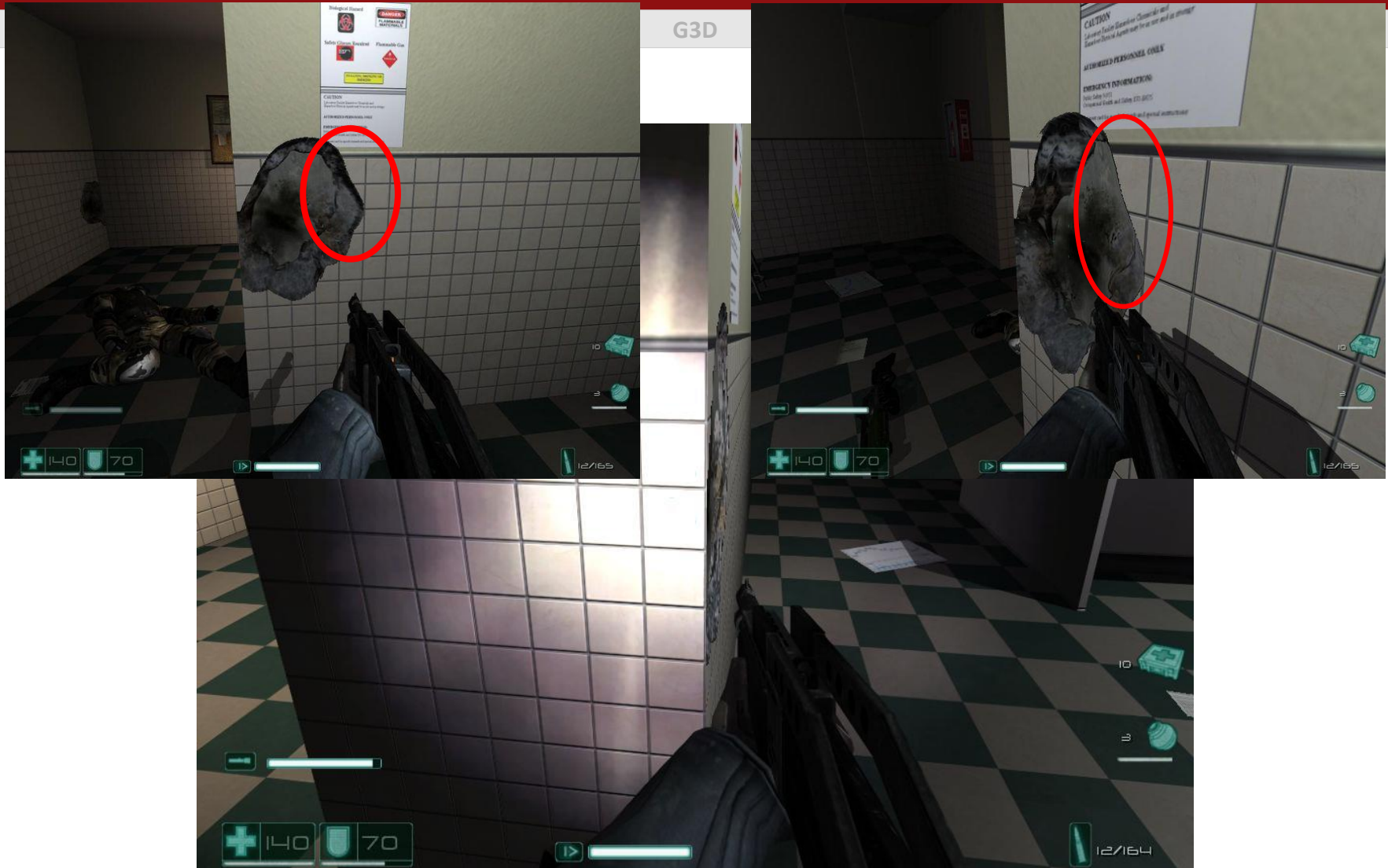
simplified mesh
500 triangles



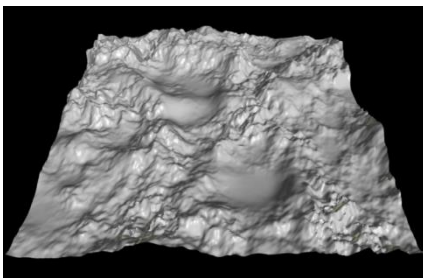
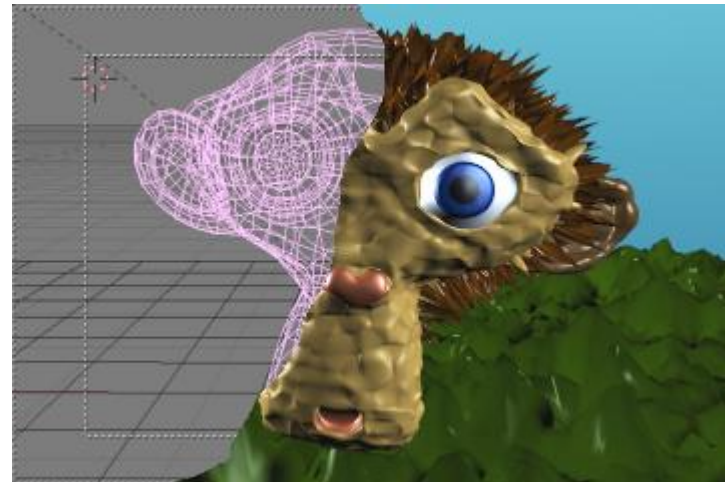
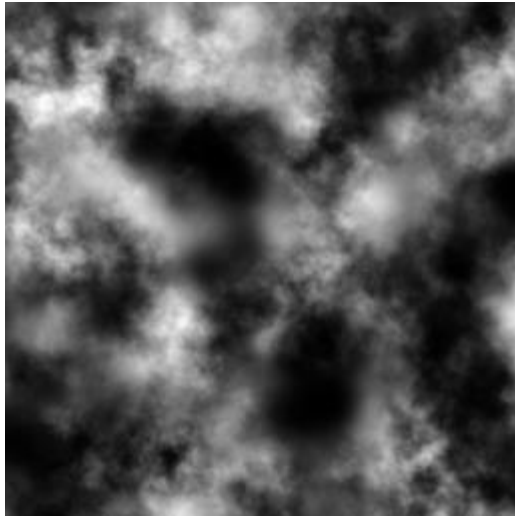
simplified mesh
and normal mapping
500 triangles

- También offset mapping o virtual displacement mapping
- Mejora sobre los mapas de normales.
- **Texturas con auto-oclusión en función del punto de vista del usuario.**
- Las texturas tienen mayor sensación de profundidad y realismo.

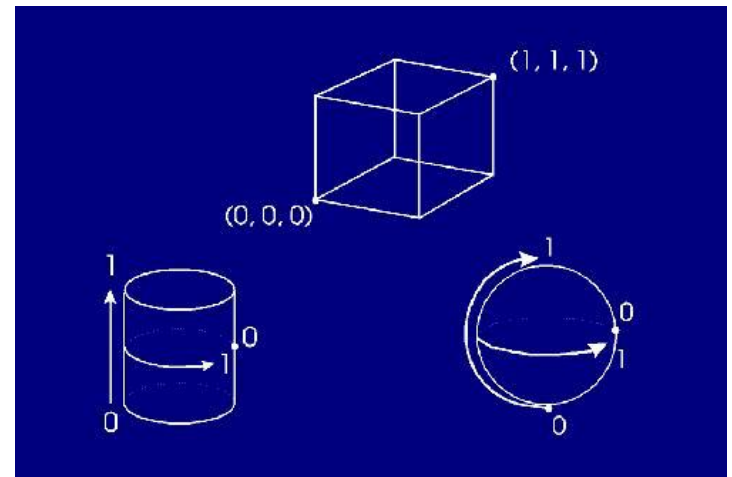
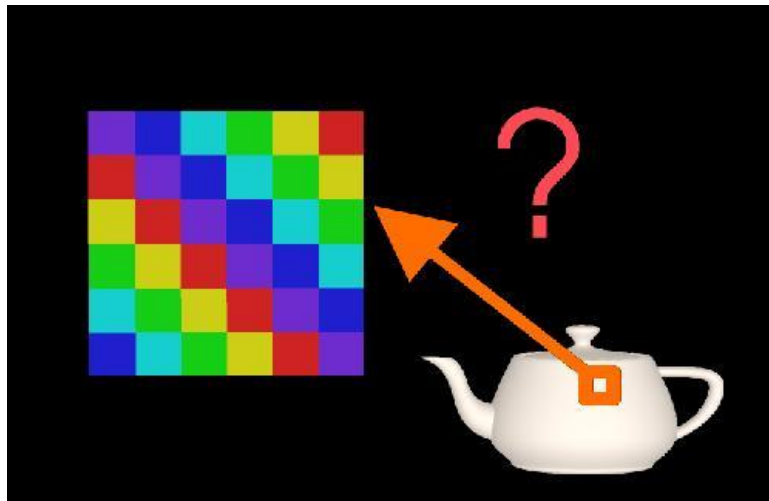
Normales: parallax mapping



- En la textura se almacena el desplazamiento de fragmentos **o vértices**.
- **Las texturas pueden almacenar: valores no normalizados y en coma flotante**
- Mapas de altura o de desplazamiento
 - Almacenan información de la altura.
 - Escalas de grises.
 - Permiten auto-oclusiones, auto-sombras y siluetas.

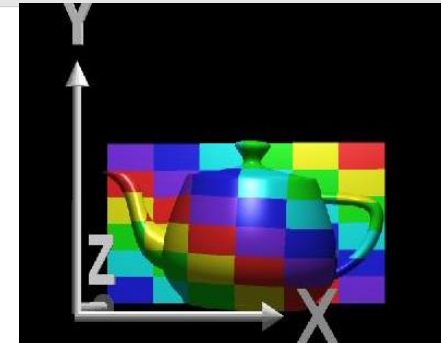


- Texture mapping: Se debe definir una correspondencia entre el modelo y la textura (fragmento o vértice y **texels**)
 - Mapeado automatico: Estas técnicas envuelven el objeto en algún tipo de figura o volumen contenedor (**map shape**)

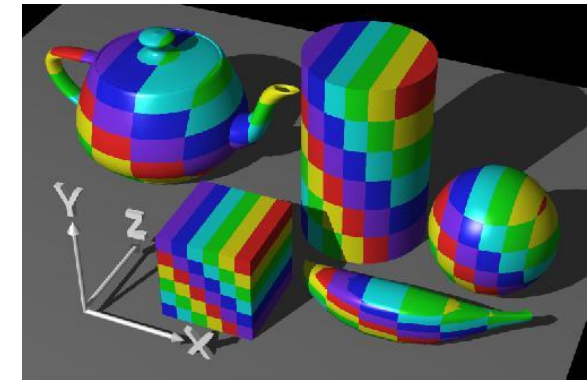


■ Mapeado plano

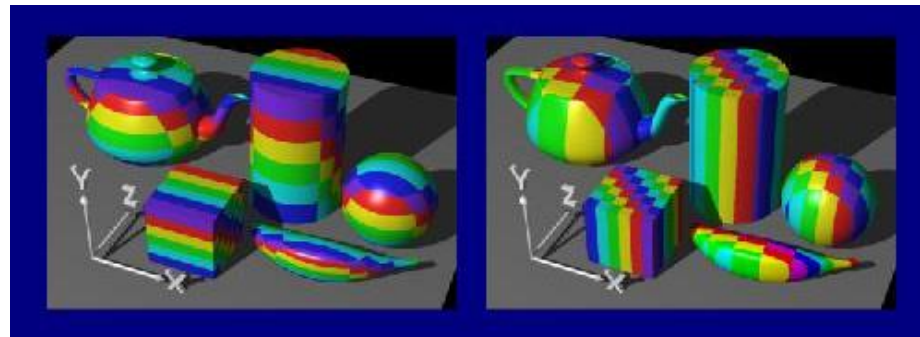
- Se proyecta la figura sobre un plano
- Se debe definir la orientación del plano
- Problemas en las zonas superiores
- El más sencillo
- Utilidad reducida
- Ejemplos:
 1. Se ha anulado la coordenada z. El color no varía con dicha coordenada.
 2. En el resto de ejemplos se eliminan distintas coordenadas.



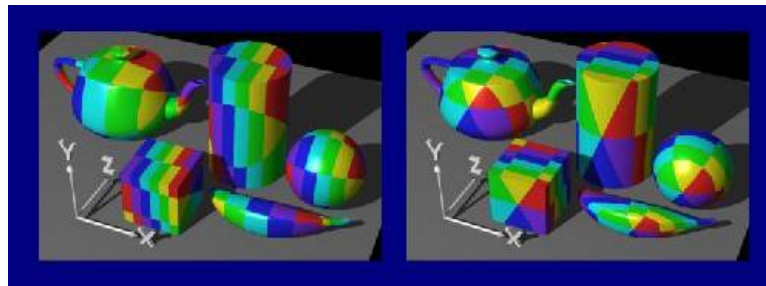
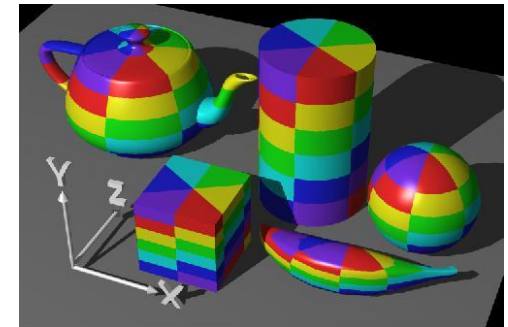
1



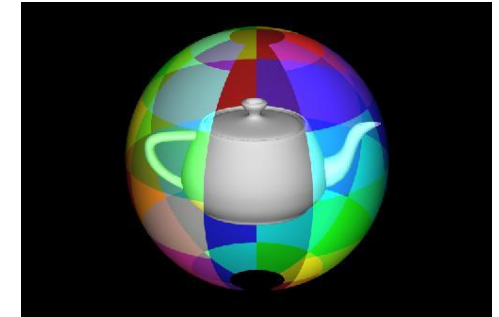
2



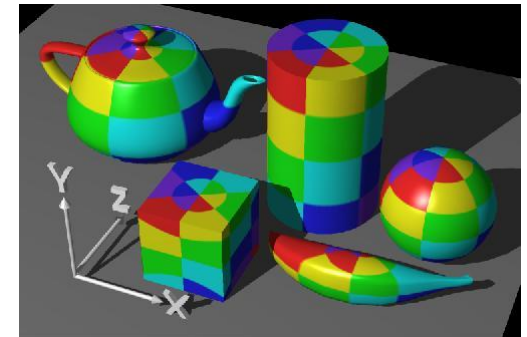
- Mapeado Cilíndrico
 - El objeto se envuelve en un cilindro
 - Las coordenadas del objeto se transforman a coordenadas cilíndricas
 - $(x,y,z) \rightarrow (\text{radio}, \text{ángulo}, \text{altura})$
 - A la hora de proyectar la textura sólo se tiene en cuenta el ángulo y la altura.
 - Debe tenerse en cuenta la orientación del cilindro



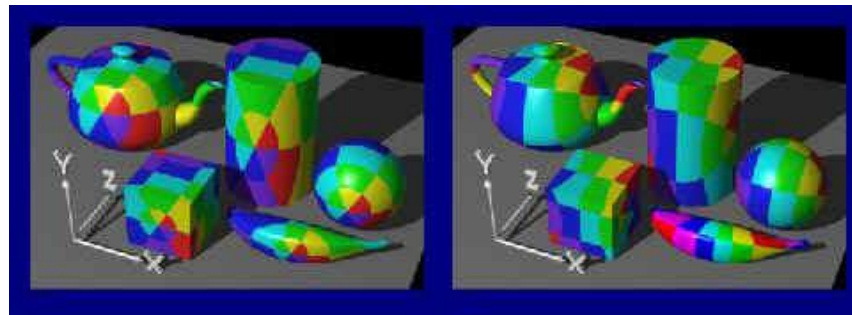
- Mapeado esférico
 - El objeto se envuelve en una esfera
 - $(x,y,z) \rightarrow (r, \alpha, \theta)$
 - Para el mapeo sólo se usa alpha y theta
 - Depende de la forma en la que la textura se envuelva alrededor de la esfera.



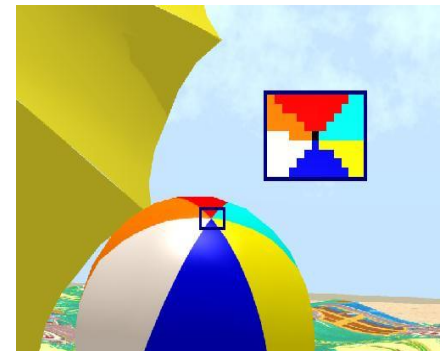
1



2



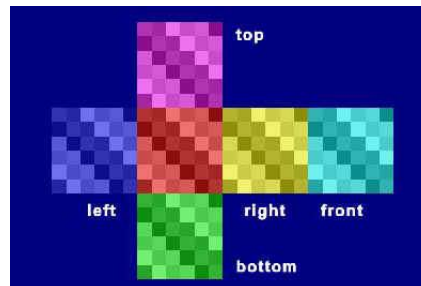
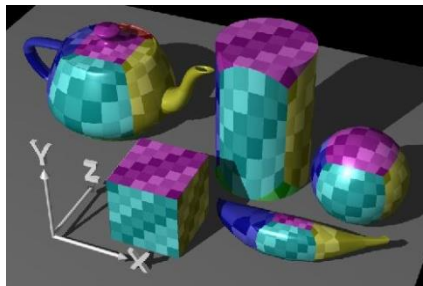
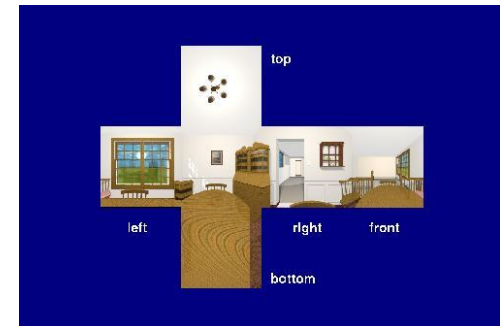
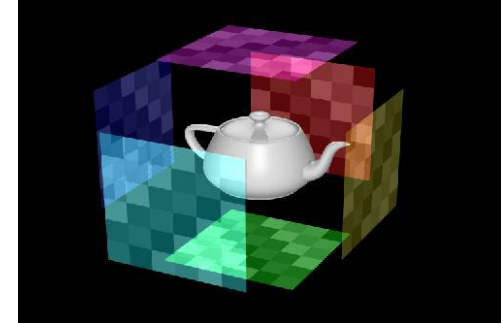
- Comparación entre el mapeado esférico y el cilíndrico.
- El mapeado cilíndrico tiene más resolución de la necesaria en los polos
 - ¿Qué texel escoger?
- El mapeado cilíndrico está muy limitado a la hora de representar la parte superior de los objetos



Mapeado (VI)

G3D

- Mapeado cúbico
 - Se envuelve el objeto en un cubo
 - Similar al mapeado plano
 - Muy utilizado para simular entornos

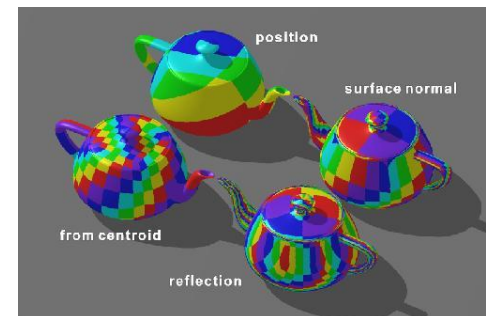
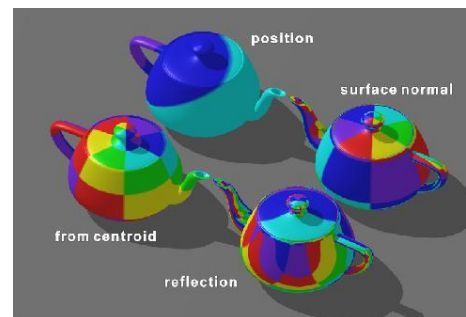
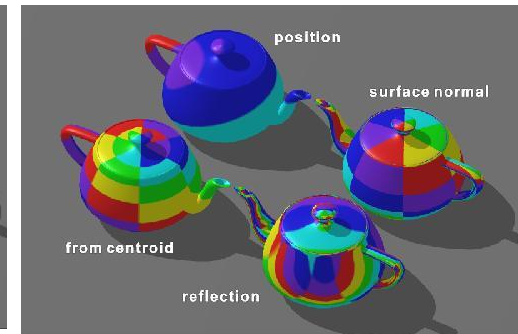
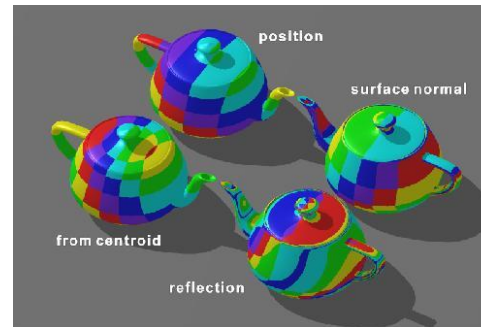
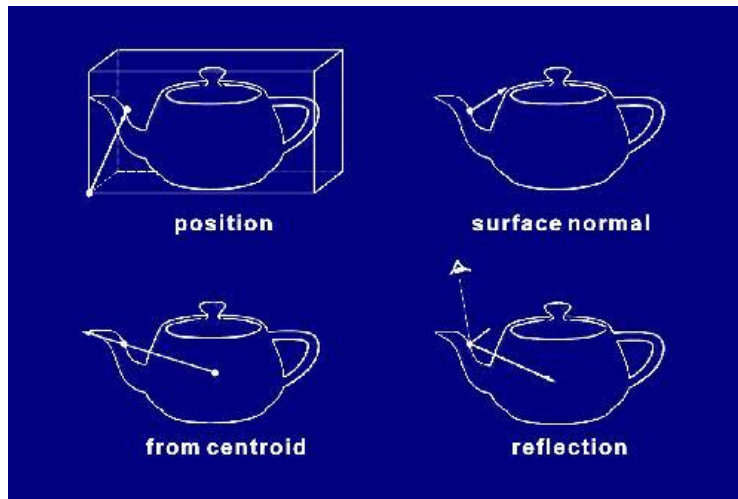


- En las transparencias anteriores sólo se tenía en cuenta la figura contenedora.
- A la ahora de proyectar o mapear la textura hay que establecer un método que nos permita extraer las coordenadas del objeto para luego proyectarlas sobre la figura contenedora (definir el **map entity**):
 - Posición respecto a la BB
 - Normal de la superficie
 - Centroide del objeto
 - Reflexión del punto de vista del observador sobre el objeto

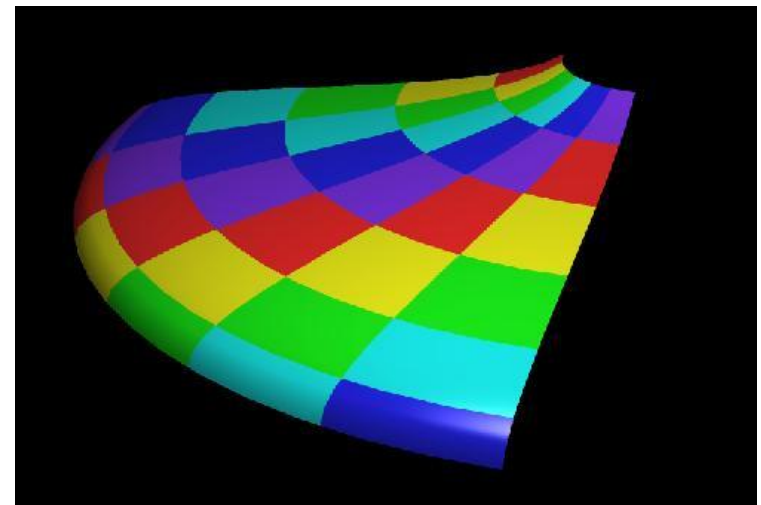
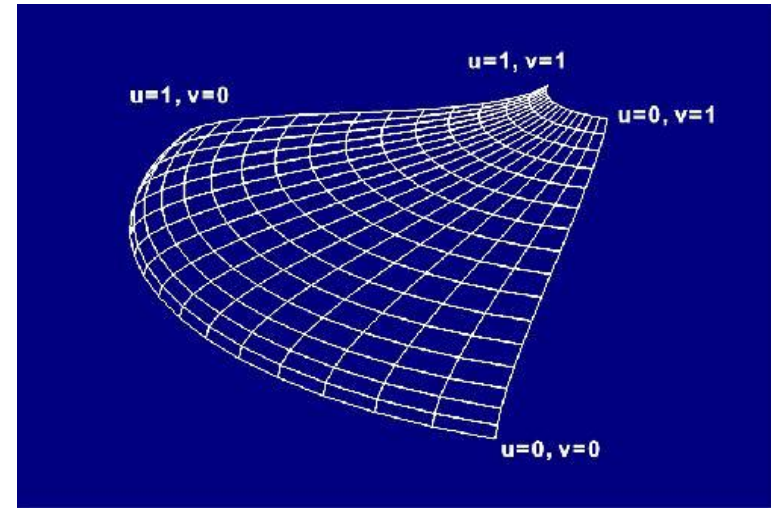
Mapeado (VIII)

G3D

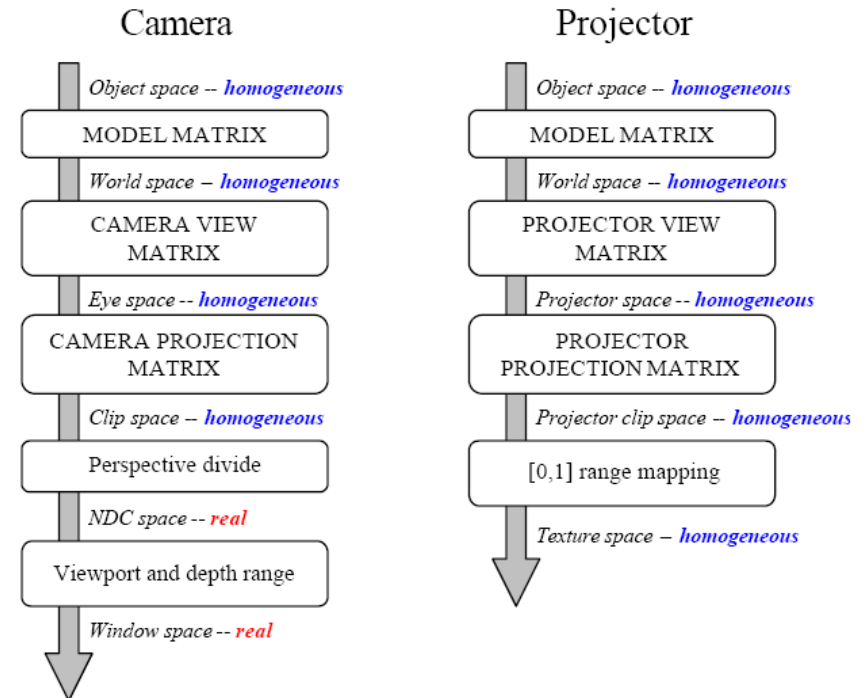
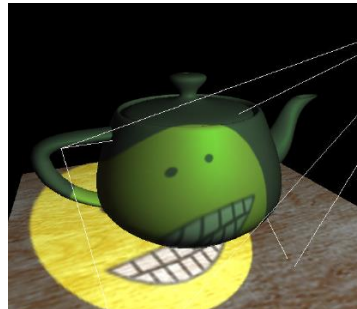
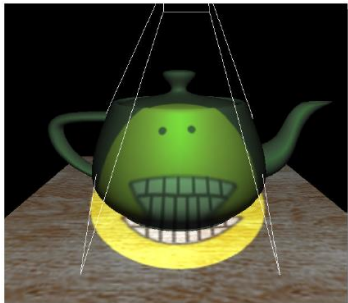
- Dependiendo de la figura envolvente unos métodos funcionarán mejor que otros



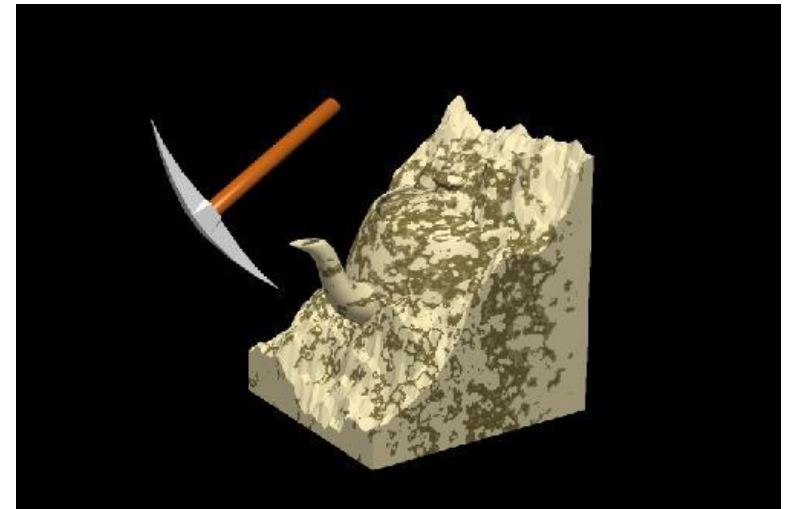
- Mapeado u,v.
 - A cada vértice se le asigna una coordenada de textura
 - El mapeado se genera de forma explícita (más difícil de modelar)
 - Mucho más potente
 - **Se suelen utilizar coordenadas normalizadas en texturas cuadradas.**
 - **En texturas no cuadradas se utiliza la posición del texel.**
 - Mapeado no lineal
 - **No requiere ni shape map, ni shape entity.**



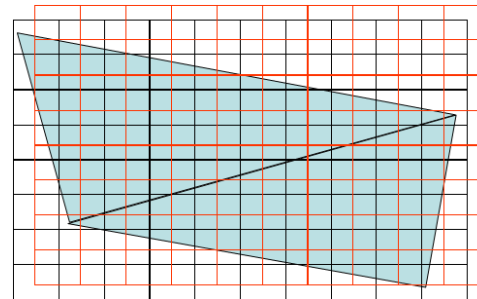
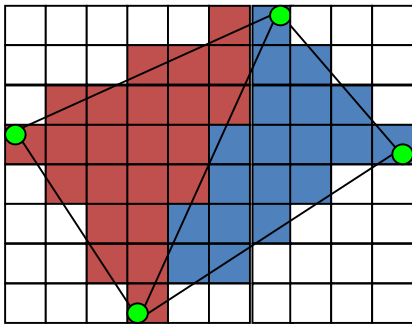
- Proyectar la textura
 - Se proyecta la textura como si se tratase de un foco
 - Shadow mapping
 - Utilizado por Pixar (se utilizó en *Toy Story*)



- Mapeado en texturas 3D
 - No se necesita figura envolvente
 - La obtención del valor es directa
 - Elevado uso de memoria

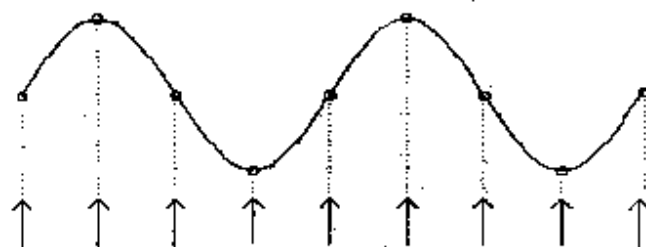


- Aliasing
 - Las fronteras de las primitivas aparecen escalonadas
 - Repetir el rasterizado desplazando la rejilla (x2, x4...).
 - Fusionar los distintos resultados

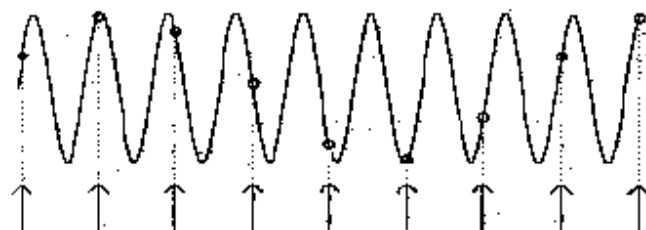


- Frecuencia de Nyquist

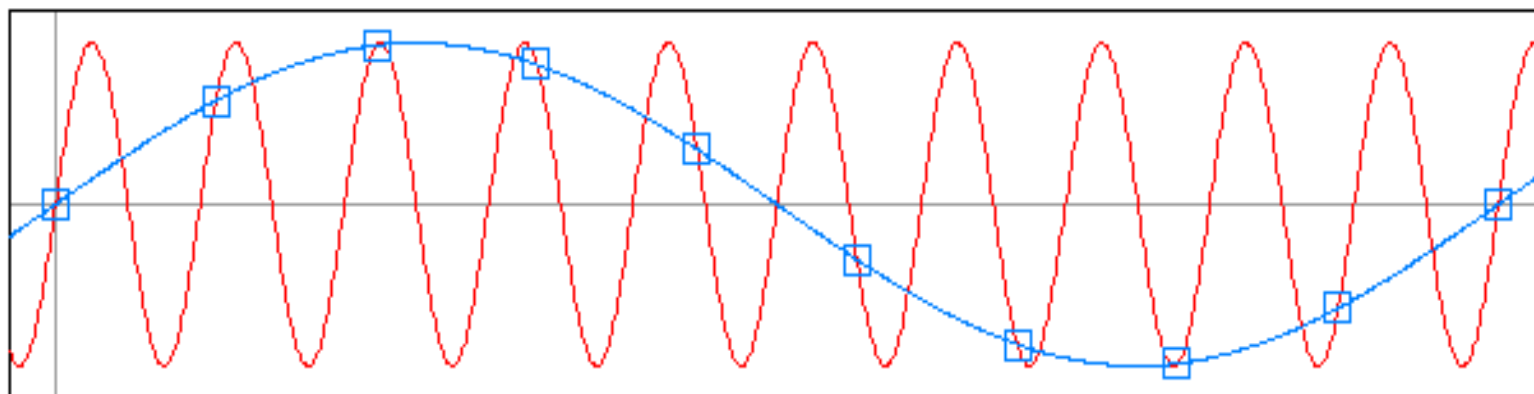
G3D



(a) Point sampling within the Nyquist limit

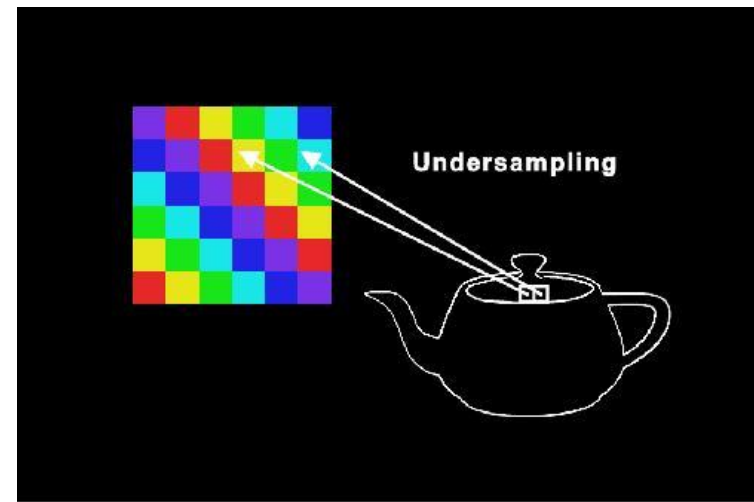
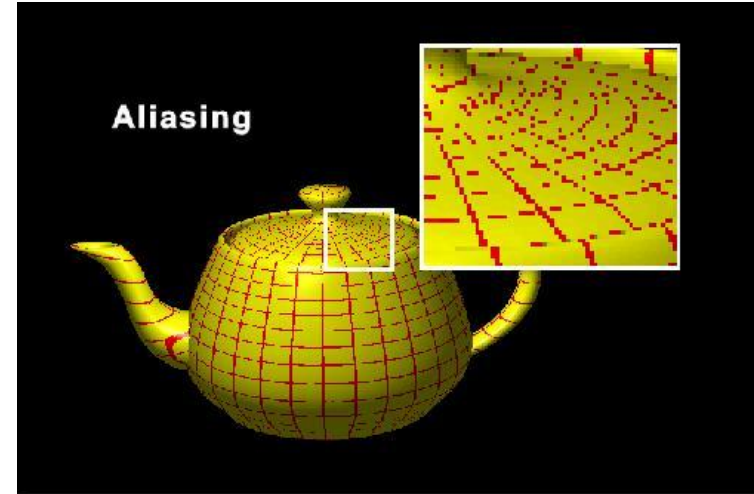


(b) Point sampling beyond the Nyquist limit



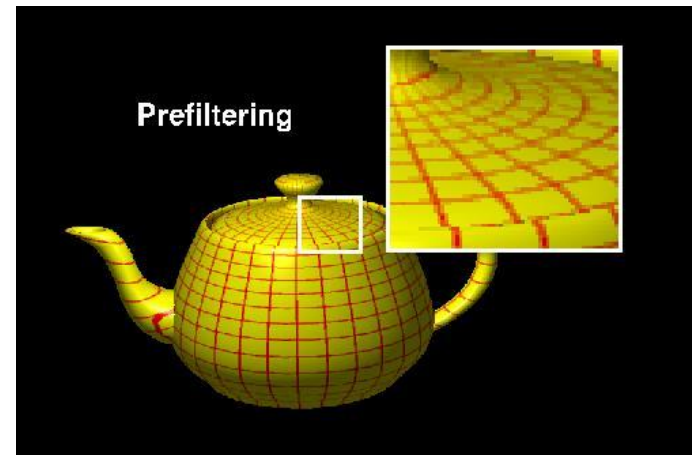
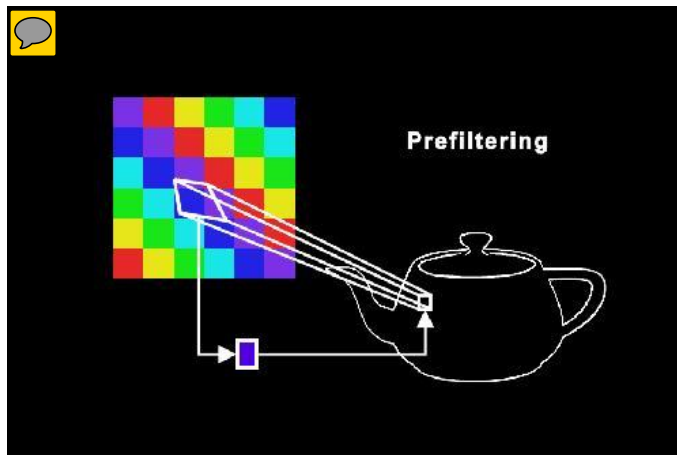
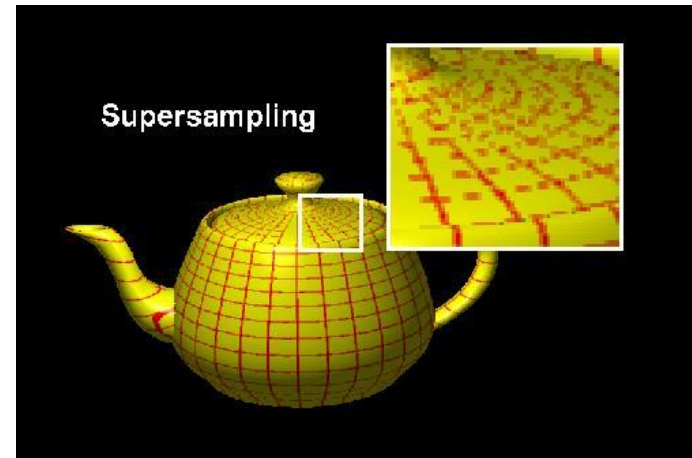
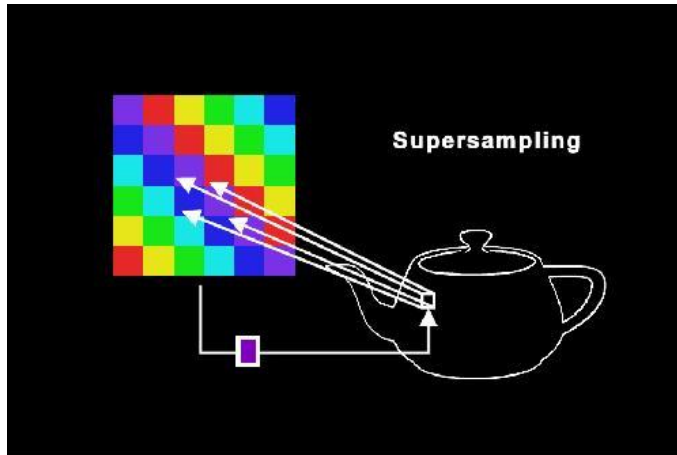
■ Aliasing en texturas

- Al proyectar una textura, ¿qué texel escoger?
- Muestreo demasiado bajo
- Puede que el texel no sea representativo
 - Fronteras escalonadas
 - Información incorrecta
- Supersampling
 - Se toman las 4 esquinas del píxel y se proyectan.
 - Se ponderan los 4 valores obtenidos
- Pre-filtering
 - Se trata el píxel como una región en la textura
 - Se pondera esa región
 - Costoso



Dificultades: aliasing

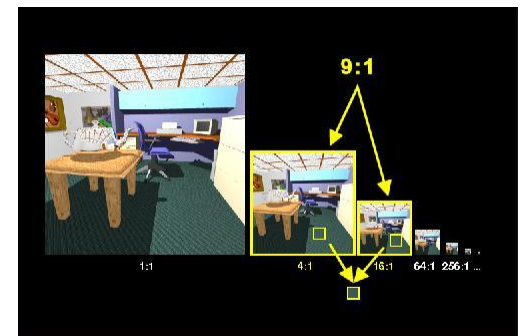
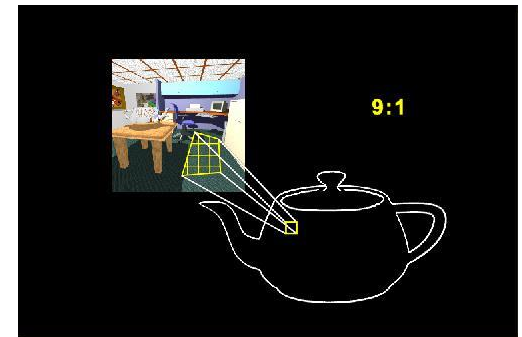
G3D



■ Aliasing en texturas

■ Texturas multimapa (mip-map)

- Ajusta la textura según la resolución
- Calculadas en preproceso (más rápidas)
- Para su generado automático tiene que ser potencia de 2
- Filtro paso bajo de un nivel a otro
- En la fase de mapeo de texturas se escoge el nivel de detalle que mejor ajuste la región
- [Ejemplo](#)



- Aliasing en texturas

- Filtrados

- Bilineal

- Interpolación con 4 téxeles
 - Se combina con mid-maps
 - Problema en la transición de niveles
 - Blur

- Trilineal

- Interpolado del filtrado bilineal entre los niveles midmaps más cercanos
 - Blur

■ Aliasing en texturas

■ Filtrado anisotrópico (AF)

- No utiliza mipmaps
- Calcula la textura inline
- Útil cuando se necesita más resolución en alguna dirección
- A la hora de generar la textura tiene en cuenta la orientación de la primitiva
- Costoso computacionalmente, requiere mucha memoria
- No introduce emborronamiento, conserva el detalle
- Sistema basado en máscaras elípticas

Dificultades

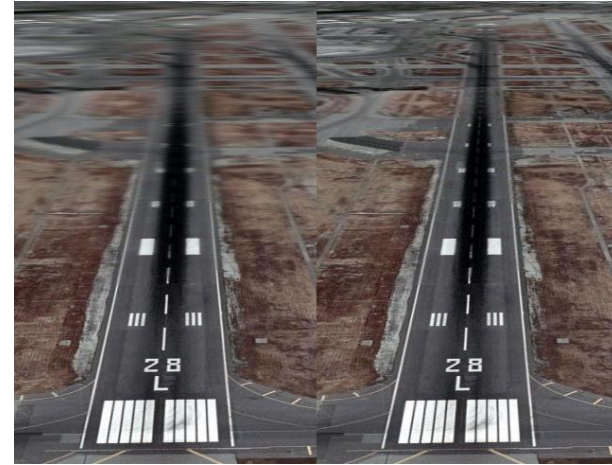
G3D



Sin filtrado



Filtrado bilineal



Filtrado bilineal

Filtrado anisotrópico

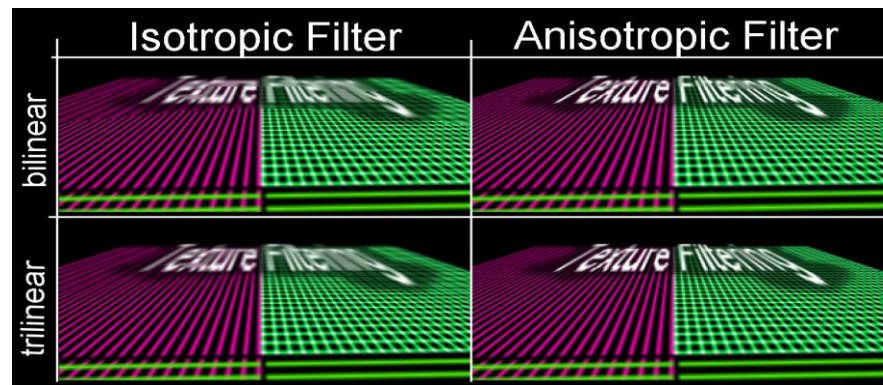
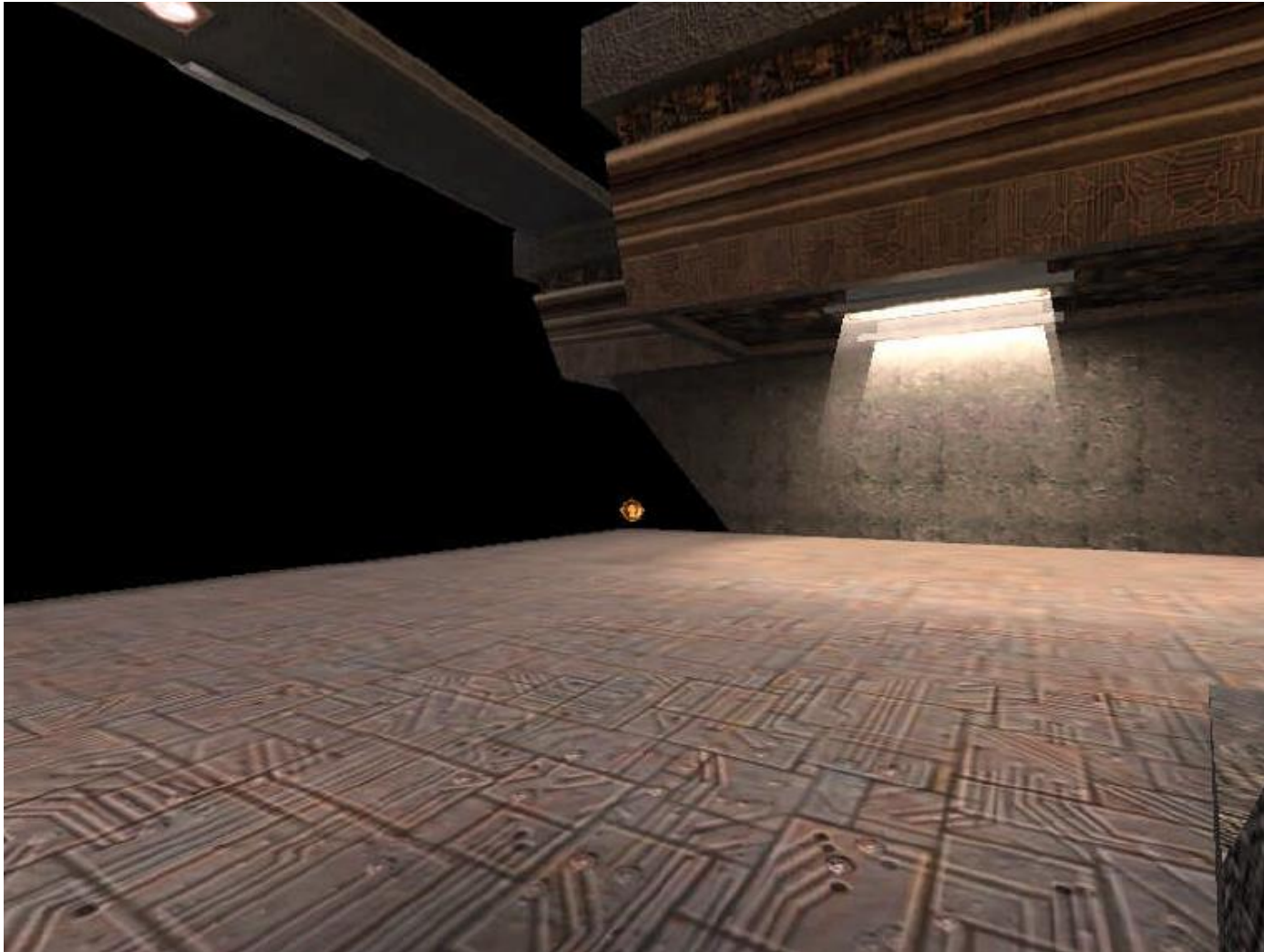


Figure 1. Comparison of isotropic and anisotropic texture filtering.

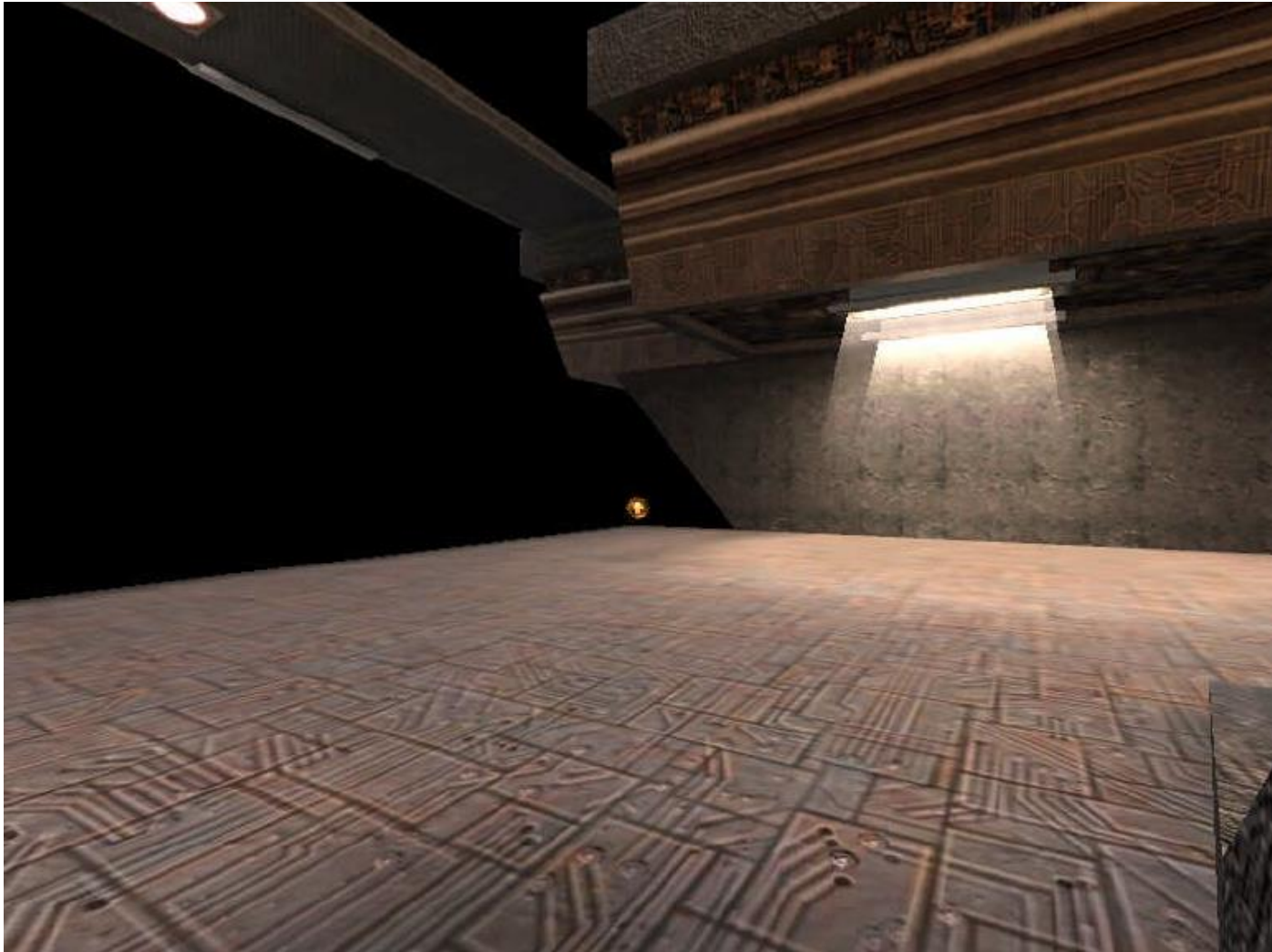
Bilinear

G3D



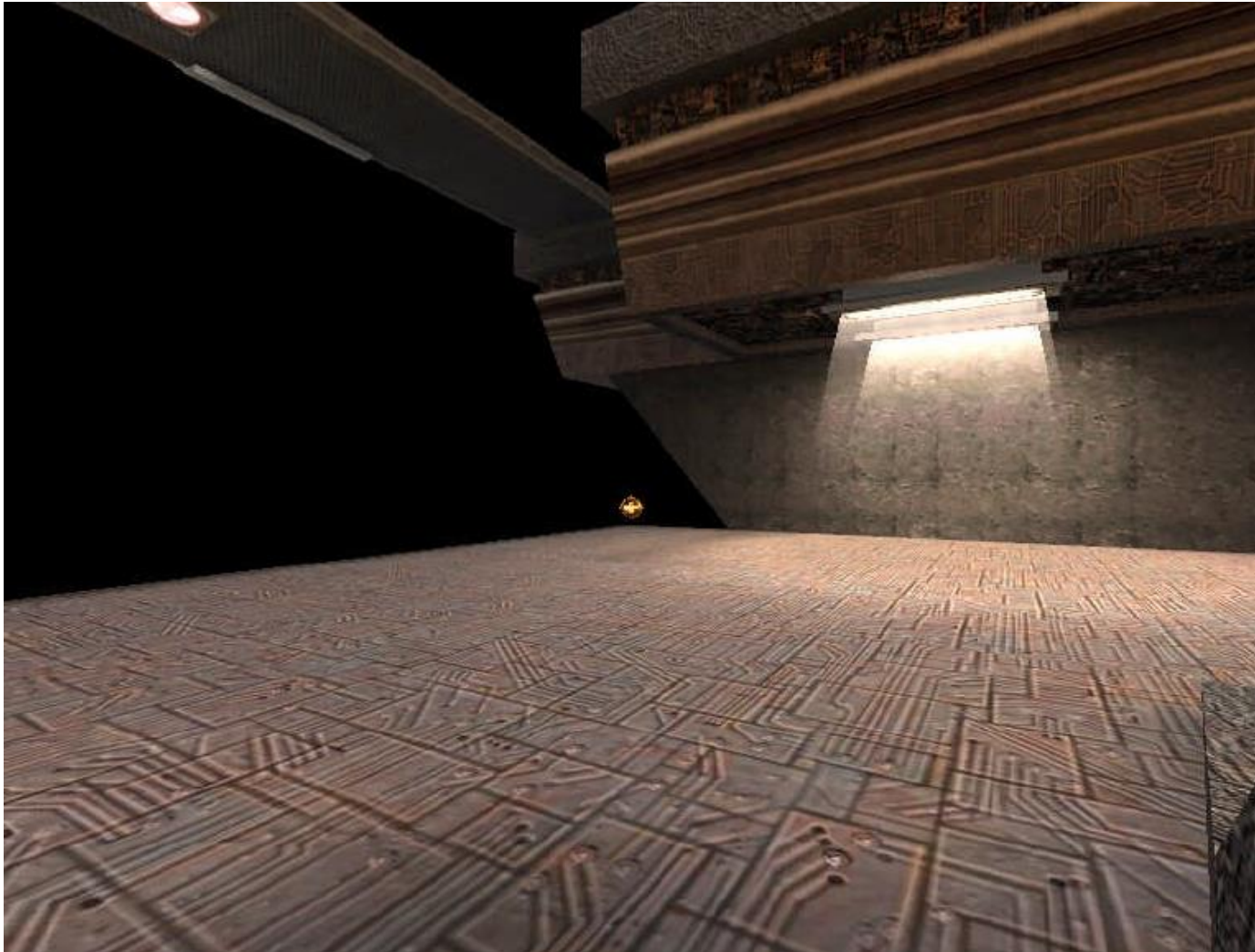
Trilinal

G3D



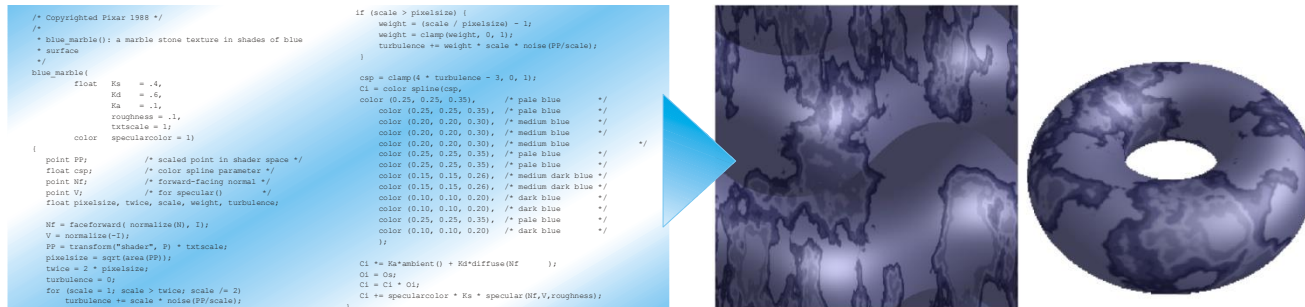
Anisotrópico

G3D



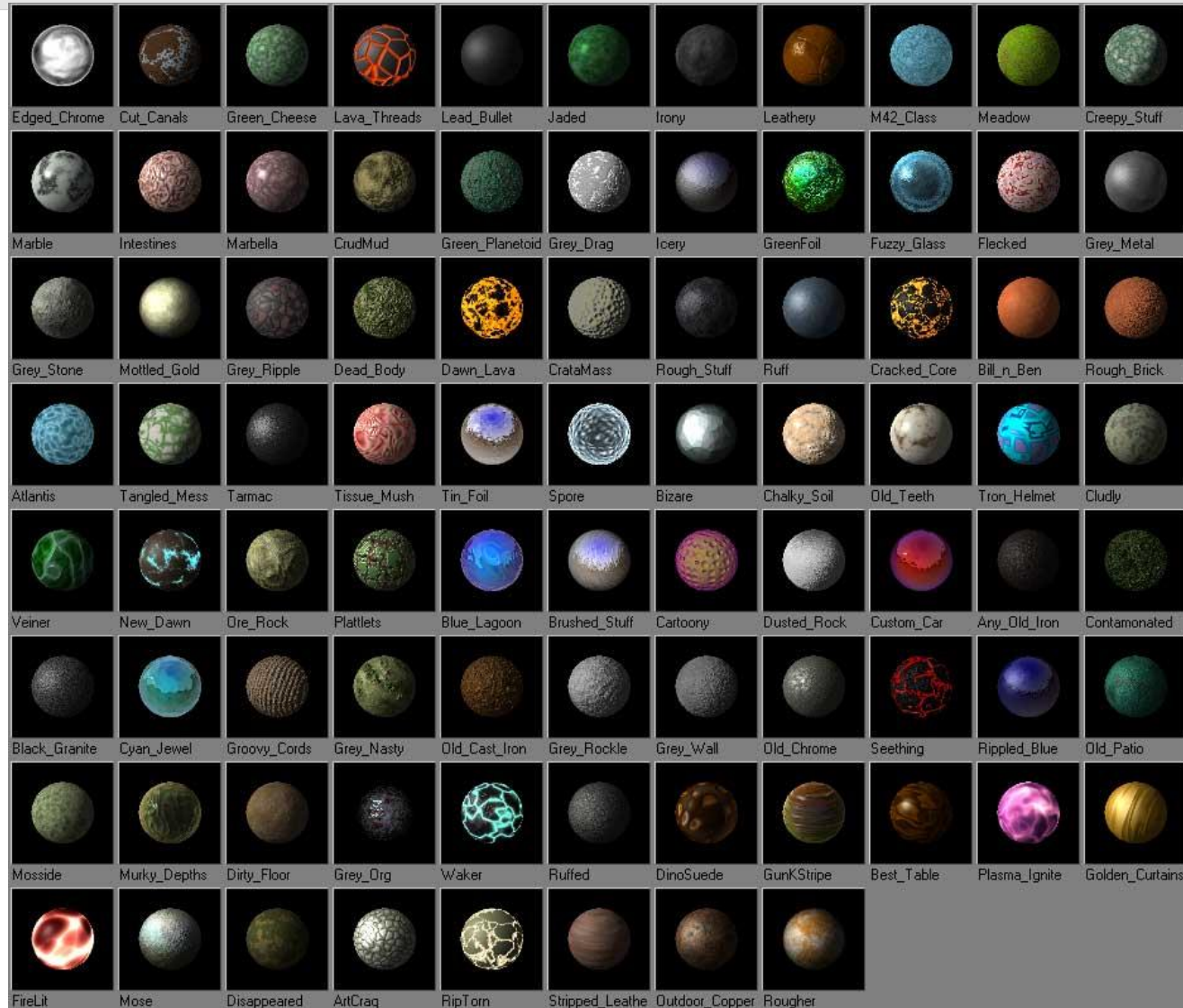
- Tienen una resolución limitada.
 - Cantidad de memoria que ocupan.
 - Mayor detalle significa más submuestreos y filtrados.
- La repetición de texturas (tilling)
 - Complica su dibujo si se quiere un diseño atractivo.

- Son algoritmos matemáticos (programas) a partir de los que se construye la imagen de la textura.
 - Se pueden programar en la GPU.
- Surgieron como forma sencilla de sintetizar texturas.

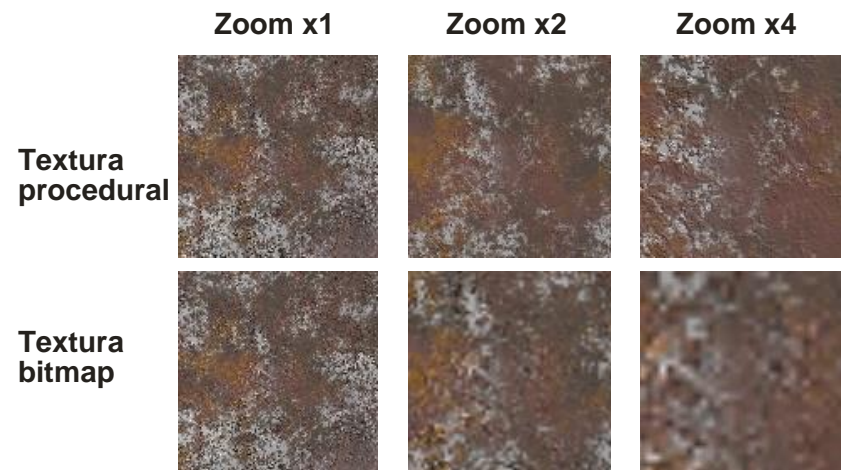


Procedurales: ejemplos

G3D



- Densidad
 - Mapa de bits: discreto.
 - Textura procedural: para todo R^n
- No tienen por qué ser repetitivas.
- No hay “tilling”.
- Multirresolución
- Uso de memoria



- No se dibujan.
 - Las herramientas habituales no sirven.
- Pueden ser sorprendentes: a veces muy difícil de predecir el resultado.
- Computacionalmente voraces
 - Ejecución en GPU.
- ¿Geometrías complejas?



- Con las texturas podemos simular efectos complejos
 - exclusivos de los modelos de sombreado globales
 - simplificar otros efectos

■ Multitexturing

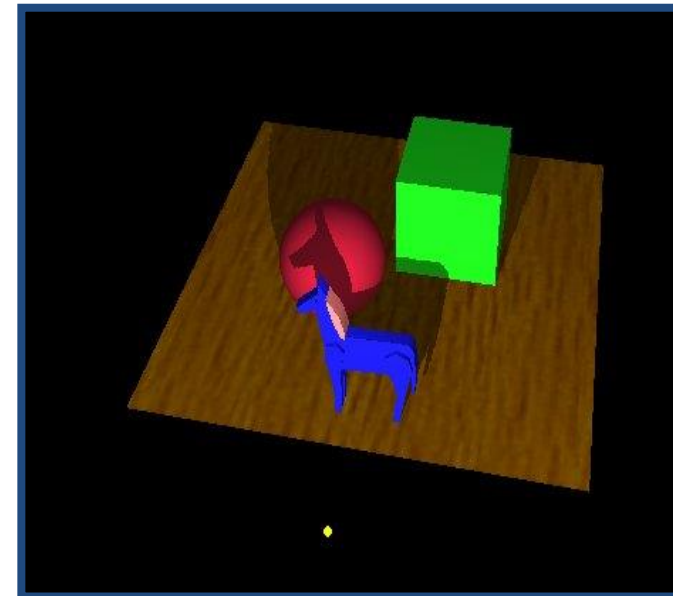
- Se pueden aplicar varias texturas a un modelo
 - Una con información de color y otra con un texture bumping
 - Fusionar dos texturas con información de color...
 - [Ejemplo](#)

■ Renderizar sobre una textura o varias

- Es la base de multitud de técnicas
 - Environmental Mapping
- [Ejemplo](#)



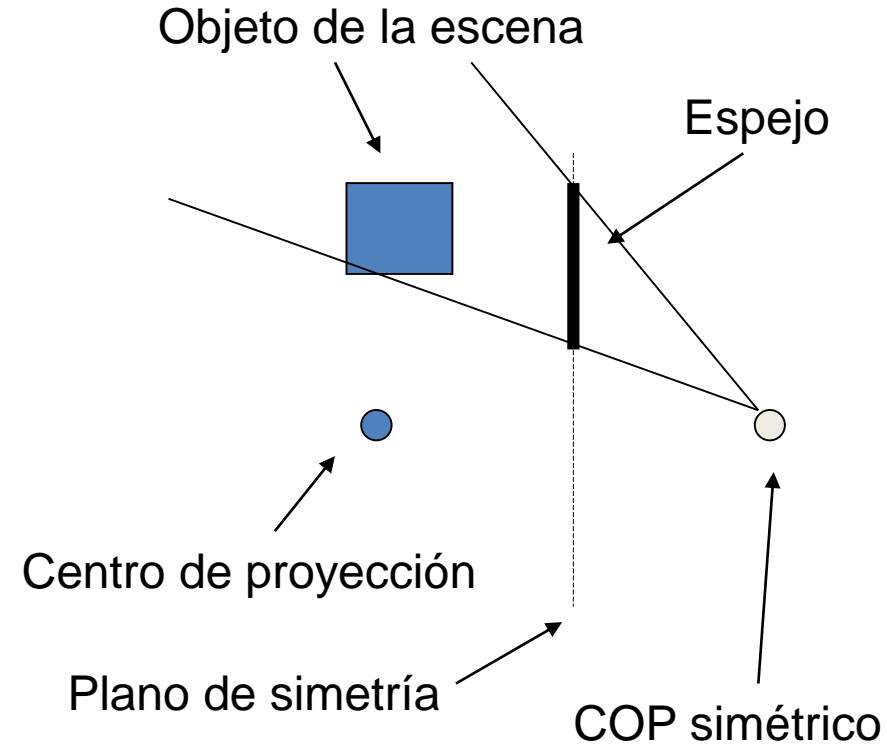
- Shadow mapping
 - Existen multitud de técnicas y variaciones que permiten calcular sombras en tiempo real
 - Proyectar una textura es una de las más sencillas
 - Crear una textura en la que se almacenará la profundidad de los objetos
 - Deshabilitar texturas, luces ...
 - Renderizar los objetos sombreadores desde el foco lumínico.
 - Activar luces, texturas...
 - Cuando se renderice se realizará el siguiente test:
 - Se proyectar los fragmentos utilizando la matriz de proyección de la fuente lumínica.
 - Se comprobará la profundidad del fragmento
 - Si es mayor que la almacenada en el Shadow Map, el fragmento se pinta de negro.



■ Reflejos

■ Plano de simetría

- Se renderiza la escena en la posición simétrica al observador
- Se utiliza como plano de simetría el mismo que contiene al usuario
- Se almacena el buffer de color en una textura
- Se renderiza la escena utilizando la textura generada en el paso anterior en el espejo
- Sólo es valido para objetos planos



■ Reflejos

■ Mapeo de entornos

- Más flexible que la técnica anterior y más usada
- Se utiliza la posición de la cámara y la normal del objeto para generar la coordenada de textura
- Mapeado circular

- 1 textura
- Difícil general la textura en tiempo real
- Textura preprocesada

■ Mapeado parabólico dual

- Es el mismo principio que subyace en las lentes parabólicas
- Función de mapeado
- 2 parábolas, 2 imágenes
- Las imágenes pueden generarse en tiempo real

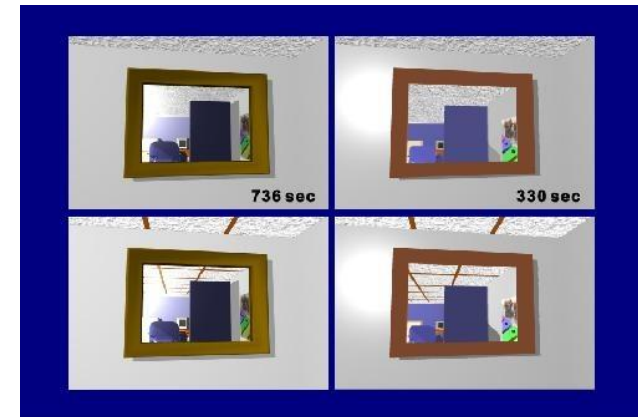
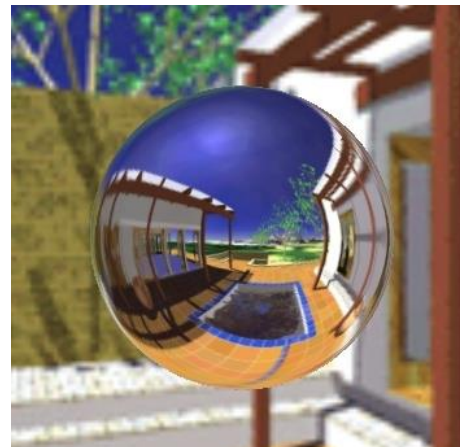
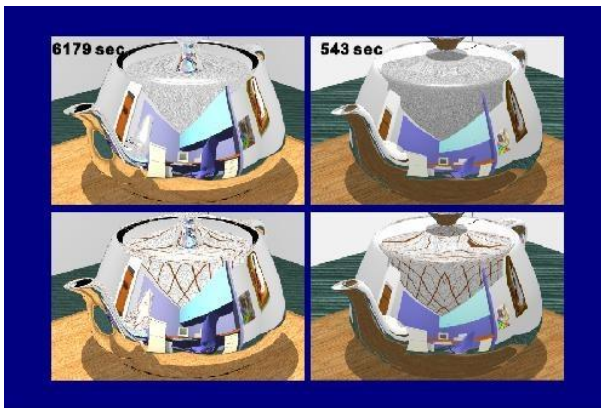
■ Mapeado cúbico

- 6 texturas
- El más utilizado
- Fácil generar la textura en tiempo real (ajustar frustum)



$$f(x, y) = \frac{1}{2} - \frac{1}{2}(x^2 + y^2), \quad x^2 + y^2 \leq 1$$

- Reflejos
 - Mapeo de entornos
 - Ejemplos:
 - Comparación con Ray-Tracing
 - [Ejemplo](#)



Efectos (IV)

G3D

- Image Base Render
 - Utilizamos un conjunto de mapas de bits para representar un objeto
 - Dependiendo del punto de vista mostramos un bitmap u otro.

