Volume Rendering

Francisco Morillo y Ciro Durán

30 de noviembre de 2005

Visualización Médica

Figura: Múltiples tejidos cabeza humana

Visualización Médica

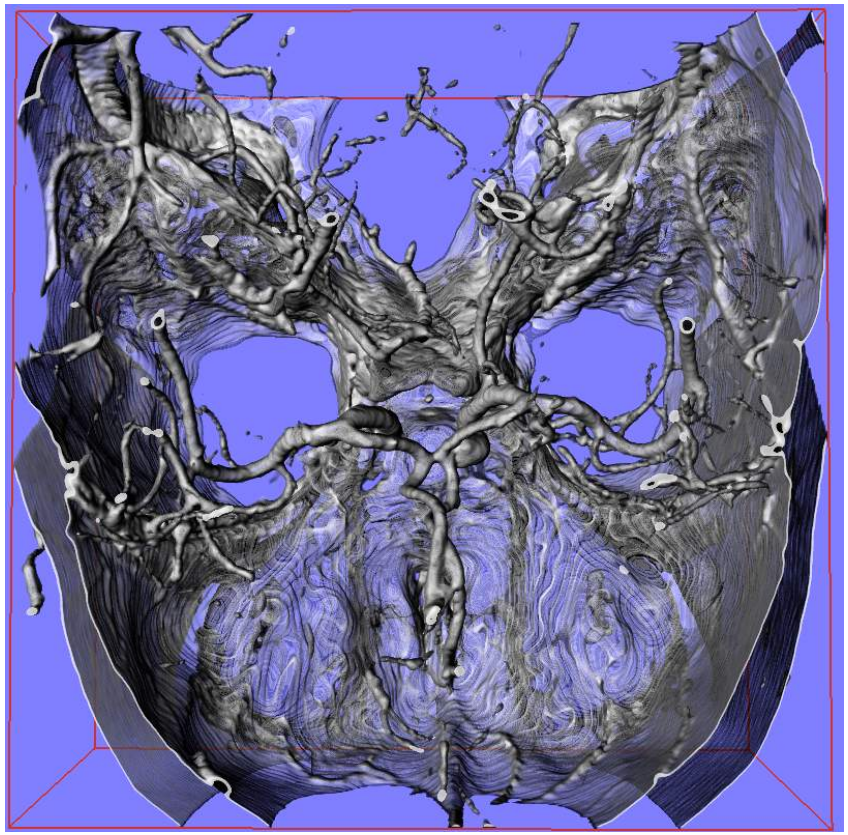


Figura: Angiograma Cerebral

Ingeniería

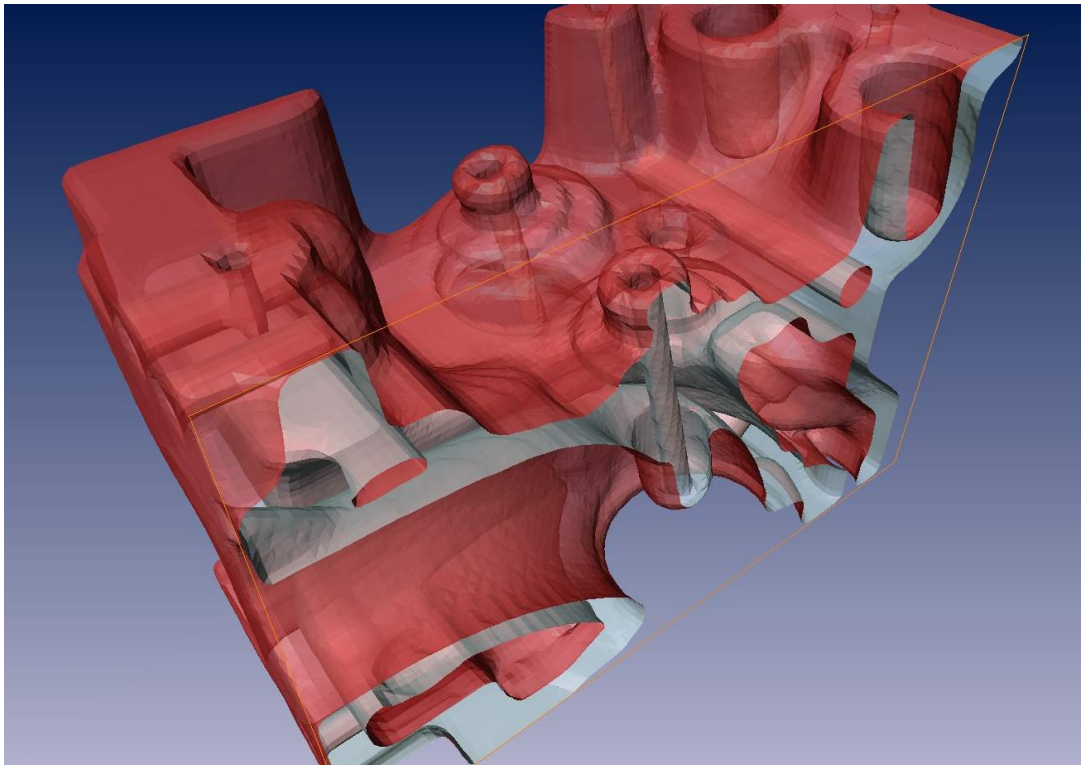


Figura: Bloque de Motor

Ingeniería

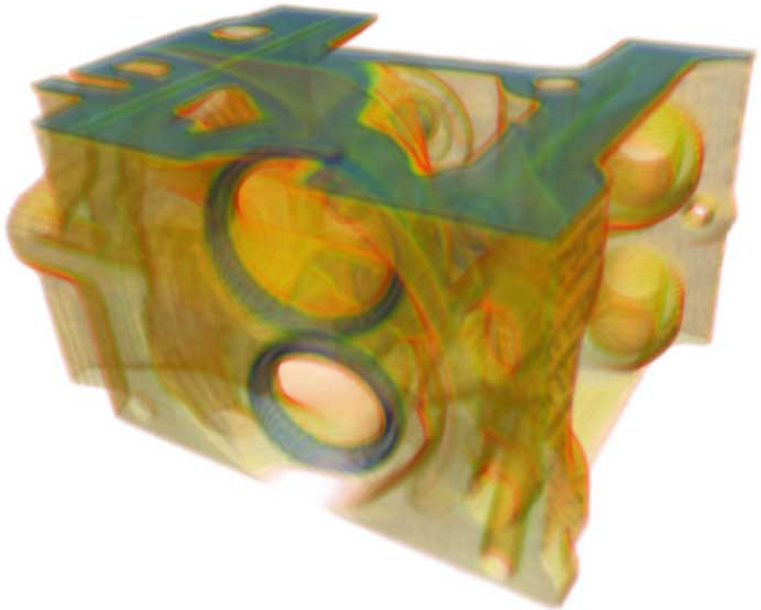


Figura: Bloque de Motor

Visualización de Fluídos

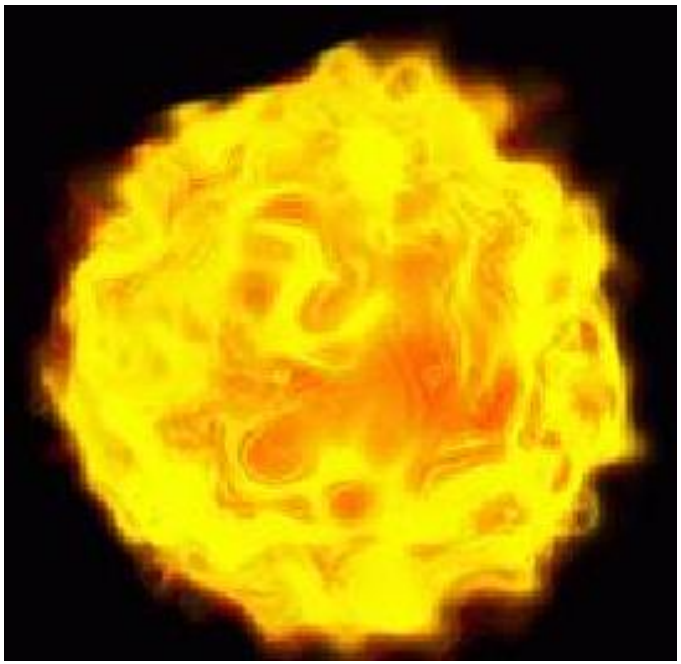


Figura: Fuego

Visualización de Fluídos

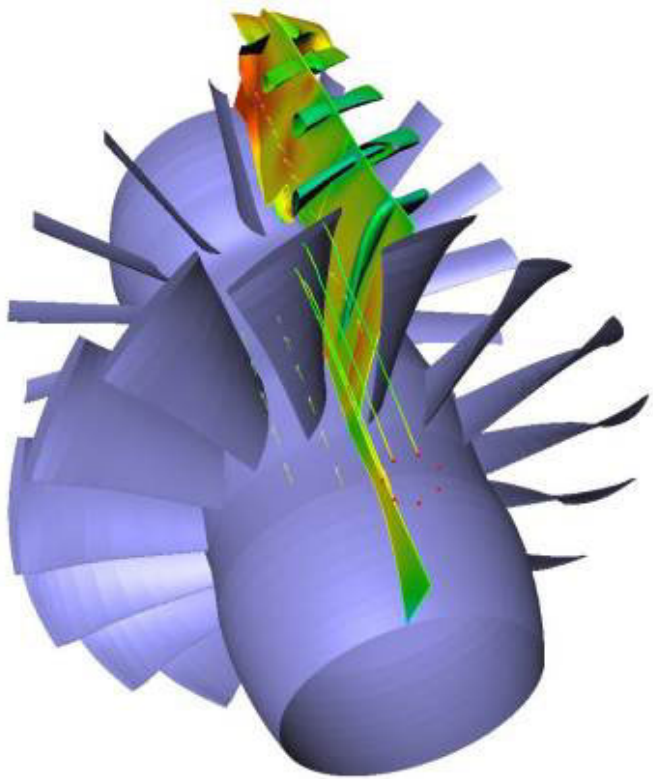


Figura: Computational Fluid Dynamics

Volume Rendering

Una técnica para elaborar representaciones 2D (las pantallas aún son 2D) de información tridimensional.

Existen muchas formas de hacer VR...

- **Ray-Casting**
- **Texture Based (2D, 3D)**
- Iso-Surface Representation
- Splatting
- Shear Warp transformation

Proceso de Raycasting

Esta técnica consiste en generar un rayo para cada uno de los píxeles de la imagen que van a mostrarse en pantalla. Los rayos son perpendiculares al plano de visión (ojo) y todos los rayos son paralelos entre sí. La idea general es muestrar el volumen a intervalos regulares sobre el rayo, y sobre todas las muestras hacer una composición de colores y opacidades; luego el color final del pixel es el resultado de la composición.

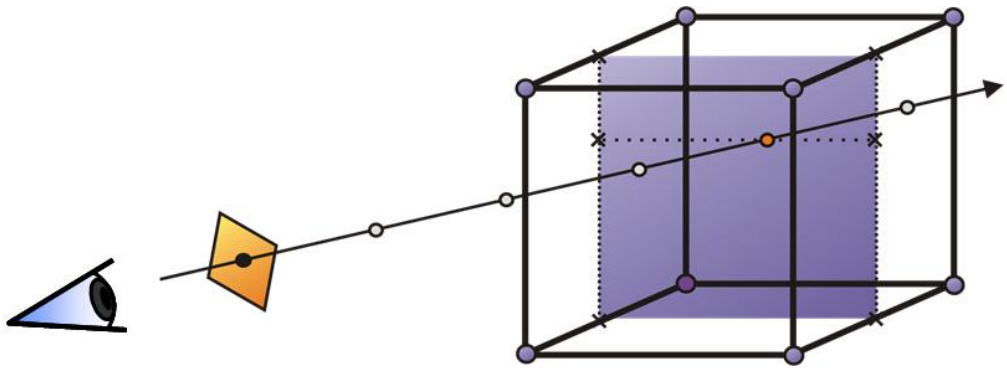


Figura: Raycasting

Composición

La composición simula el proceso de visión humano. Es una suma balanceada de los colores y opacidades de muestras en la trayectoria del rayo. La diferencia es que en la visión humana, esta suma se puede definir como una integral desde 0 a ∞ (muestras infinitas), y en VR es una suma discreta. Las fórmula que definen la composición son:

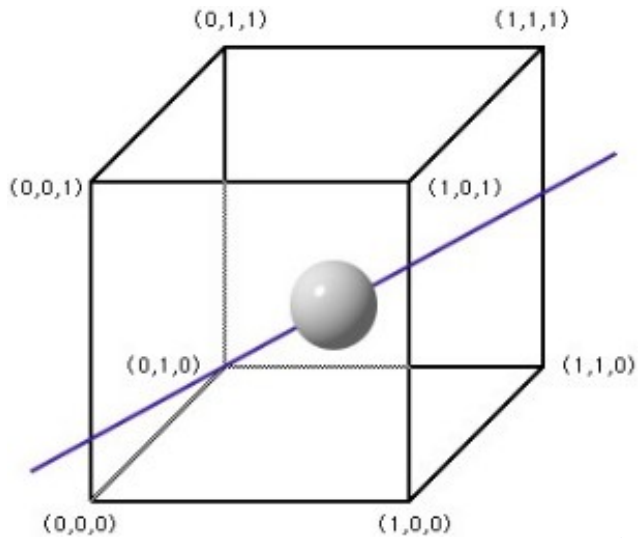
$$C = \sum_{i=1}^{n} C_i \prod_{j=1}^{i-1}(1 - A_j) \tag{1}$$

$$A = 1 - \prod_{j=1}^{n}(1 - A_j) \tag{2}$$

Interpolación Trilinear

La interpolación trilinear lo que hace es generar una muestra balanceada en su espacio tridimensional. Esto se hace tomando los valores de los 8 vecinos que rodean al punto en el espacio de la trayectoria del rayo.

$$V_{xyz} = V_{000}(1-x)(1-y)(1-z) + V_{100}x(1-y)(1-z) + V_{010}(1-x)y(1-z) + V_{001}(1-x)(1-y)z + V_{101}x(1-y)z + V_{011}(1-x)yz + V_{110}xy(1-z) + V_{111}xyz$$



Resultados de un raycaster

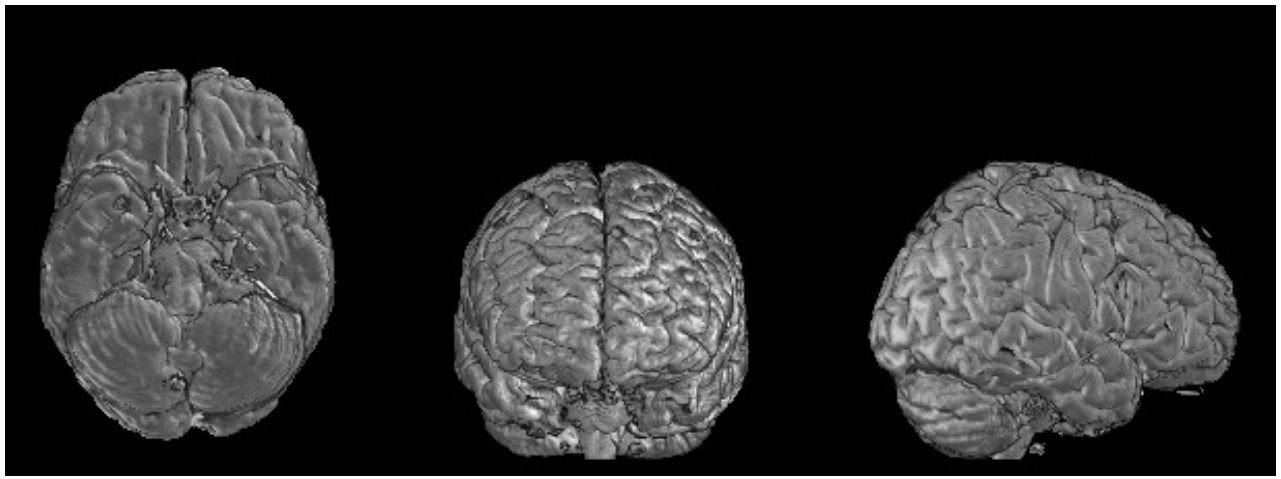


Figura: Ejemplo de un raycaster, imagen mejorada aplicando shading

Lo que ofrecen las tarjetas aceleradoras

- Operaciones de rasterización
  - Descomposición de fragmentos
  - Interpolación de colores
  - Texturización (Interpolación y combinación)
- Operaciones sobre fragmentos
  - Prueba de profundidad
  - *Alpha blending*

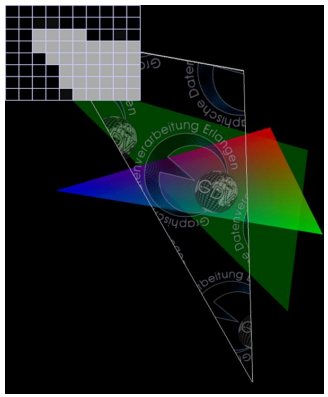


Figura: Rasterización

Volume Rendering con tarjetas aceleradoras

La técnica consiste en cargar el volumen de datos como una textura a la tarjeta gráfica (proporcionando además todas las propiedades físicas que debe tener), y mediante una serie de polígonos (llamados *proxy geometry*) dibujados de atrás hacia adelante lograr componer la imagen del volumen de datos aprovechando las capacidades de la tarjeta.

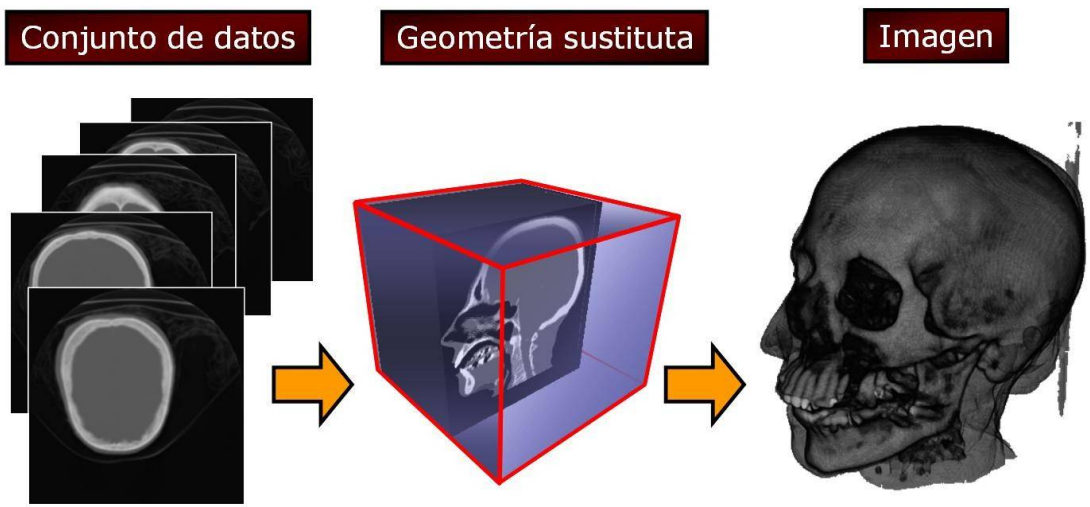


Figura: Esquema de VR basado en texturas

Tipos de textura

- Texturas bidimensionales (Polígonos orientados al objeto)
- Texturas tridimensionales (Polígonos orientados al plano de proyección)

Esquema de rendering

- Preparar el ambiente, crear la textura y bajarla a la memoria de la tarjeta.
- Preparar los *fragment shaders* y las operaciones de blending.
- Rendering de la geometría sustituta.

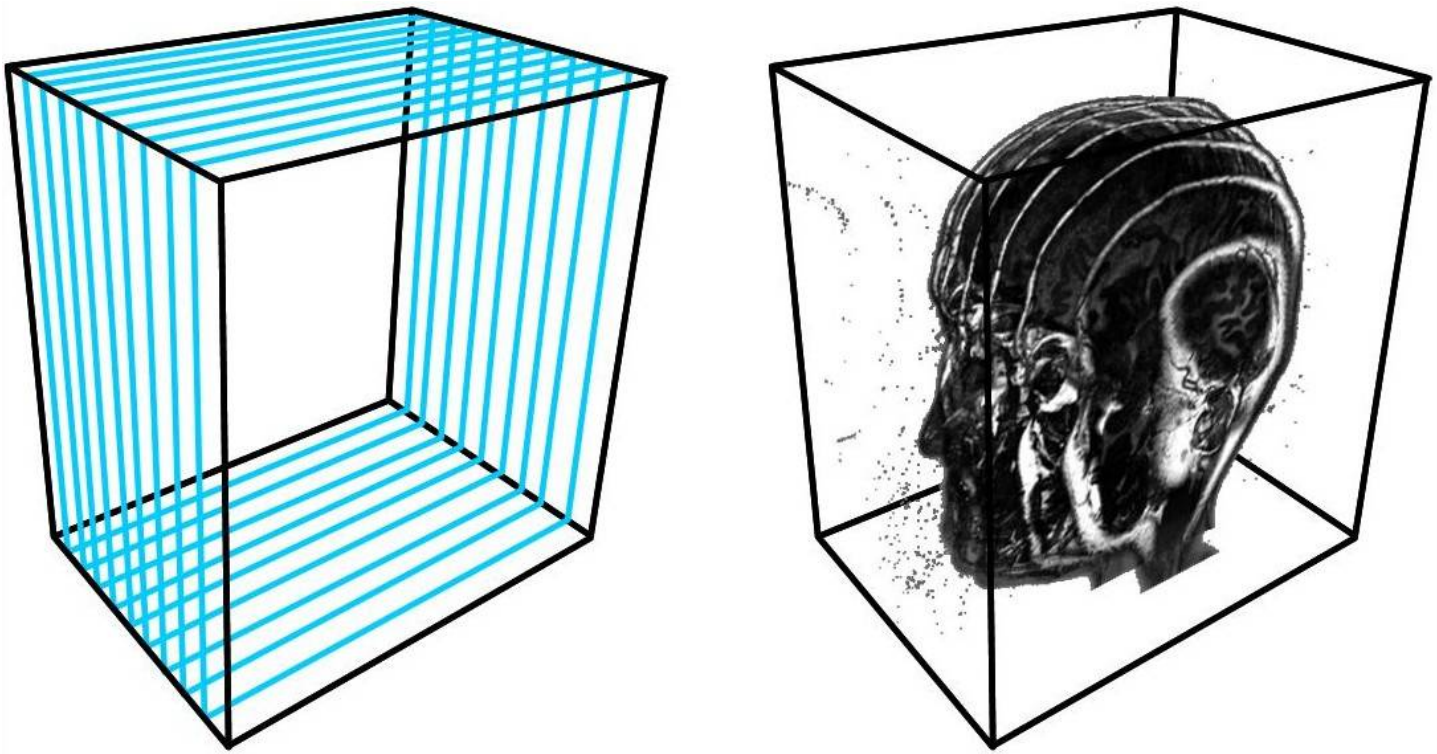


Figura: TBVR con texturas bidimensionales

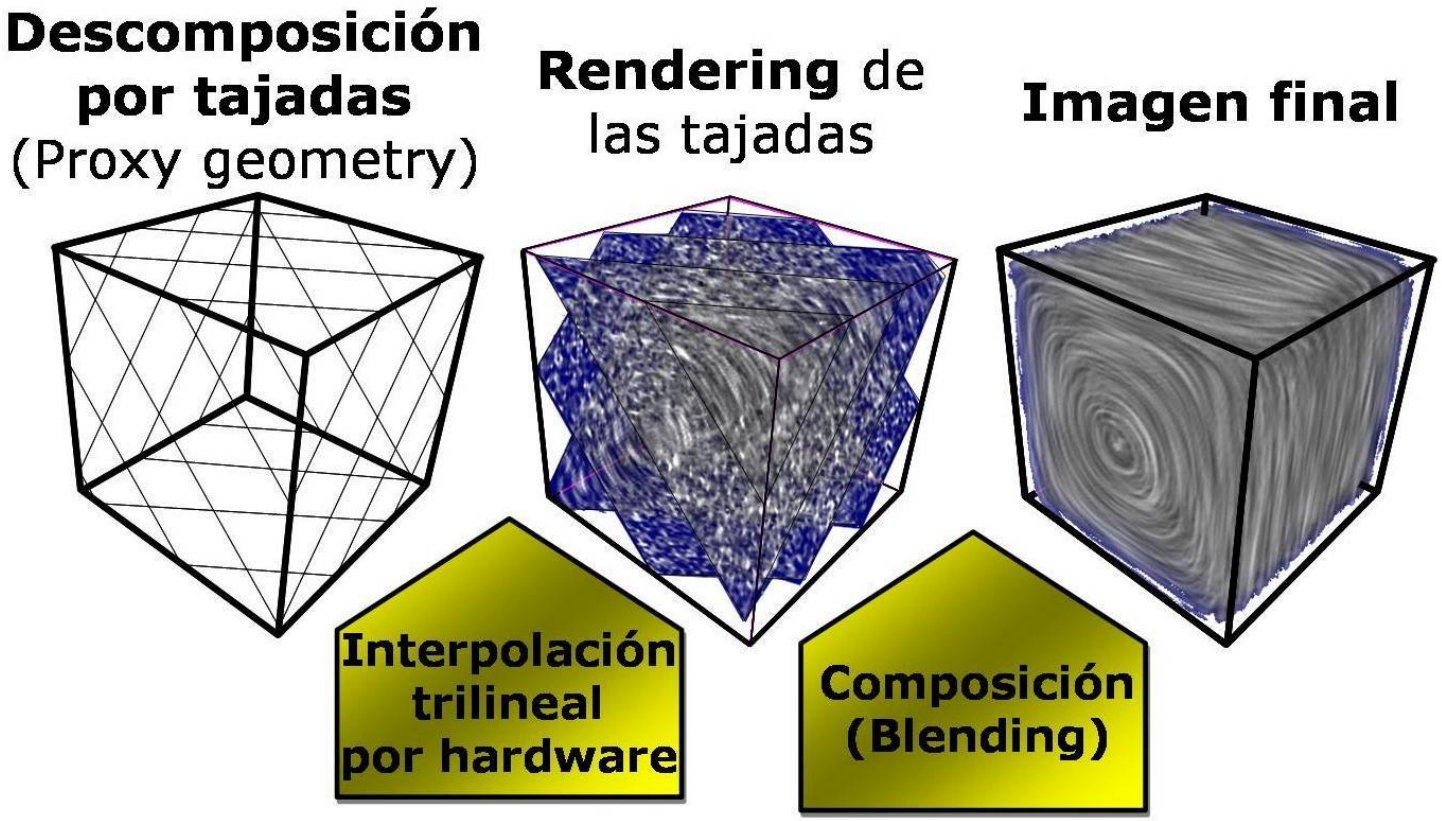


Figura: TBVR con texturas tridimensionales

Texturas bidimensionales

Interpolaciones bilineales en vez de interpolaciones trilineales. Artefactos en los bordes de cada tajada.

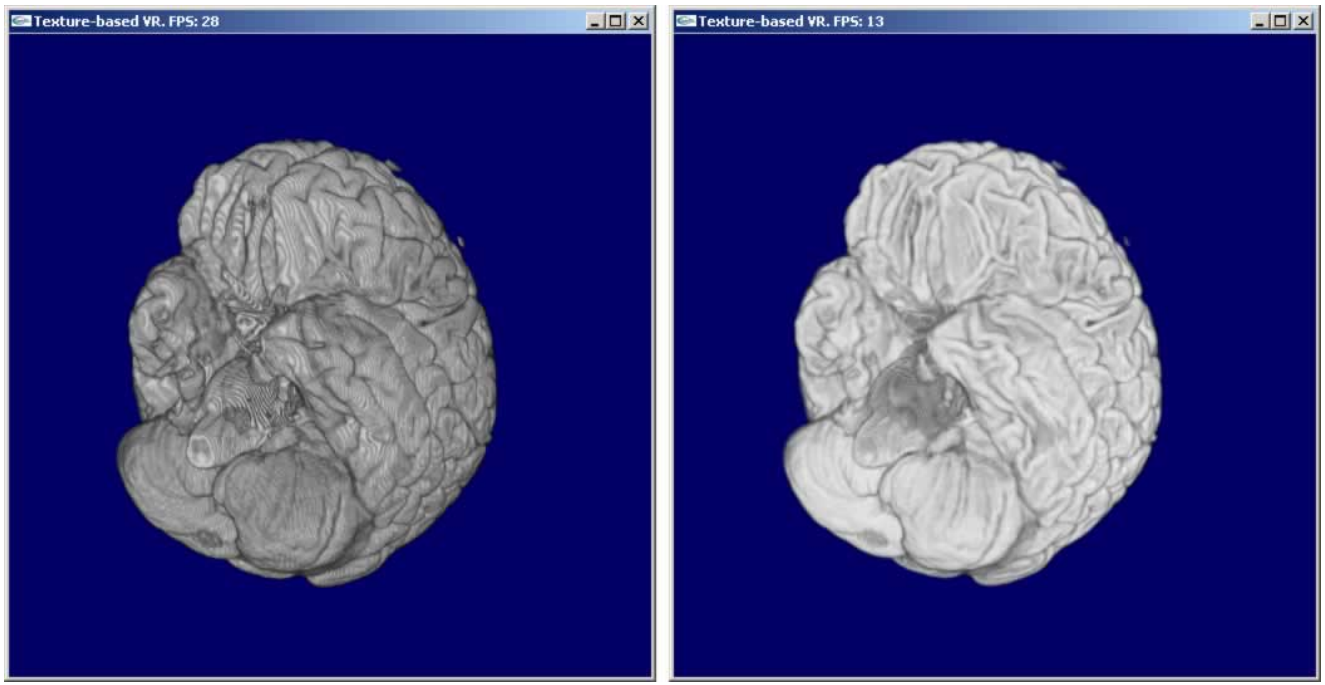


Figura: Interpolaciones bilineales causan artefactos

Texturas bidimensionales

Pop-ups. Causados por el cambio de los puntos de muestreo entre conjuntos de texturas.

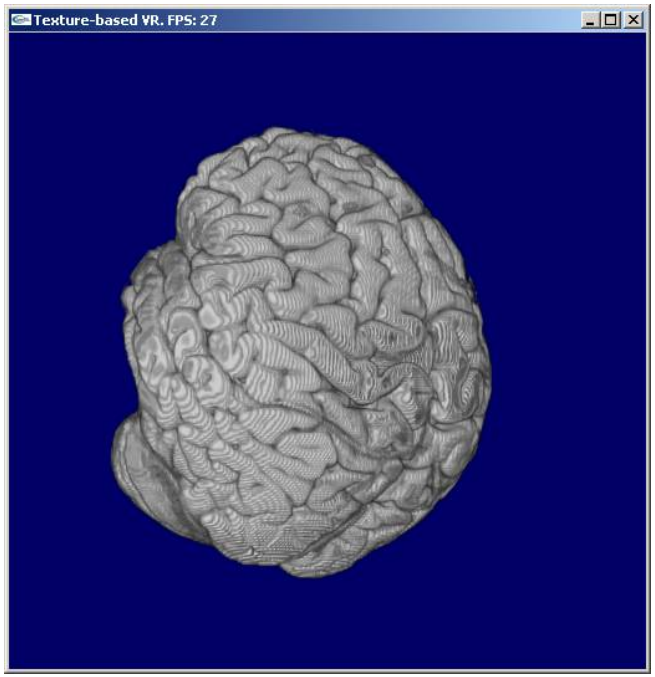


Figura: Pop-up ilustrado

Texturas bidimensionales

Pop-ups. Causados por el cambio de los puntos de muestreo entre conjuntos de texturas.

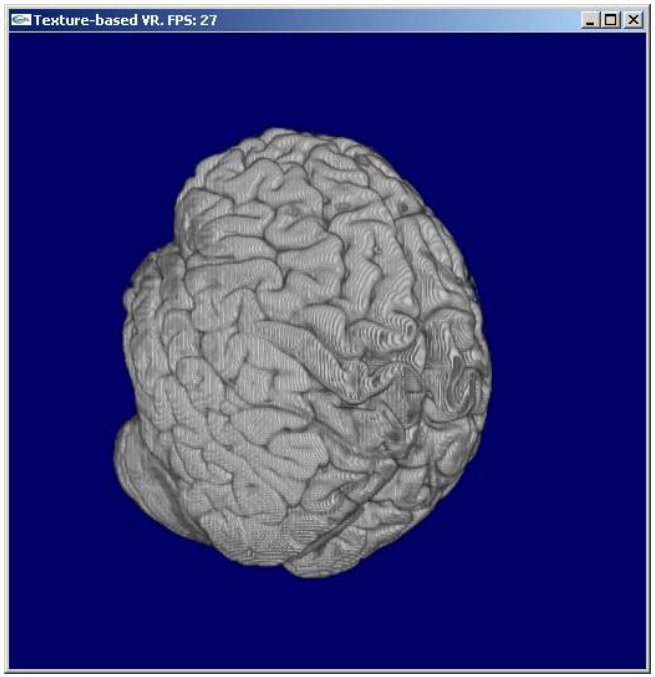


Figura: Pop-up ilustrado

Texturas bidimensionales

Frecuencia de muestreo inconsistente. La emisión y la absorción no son simuladas correctamente.

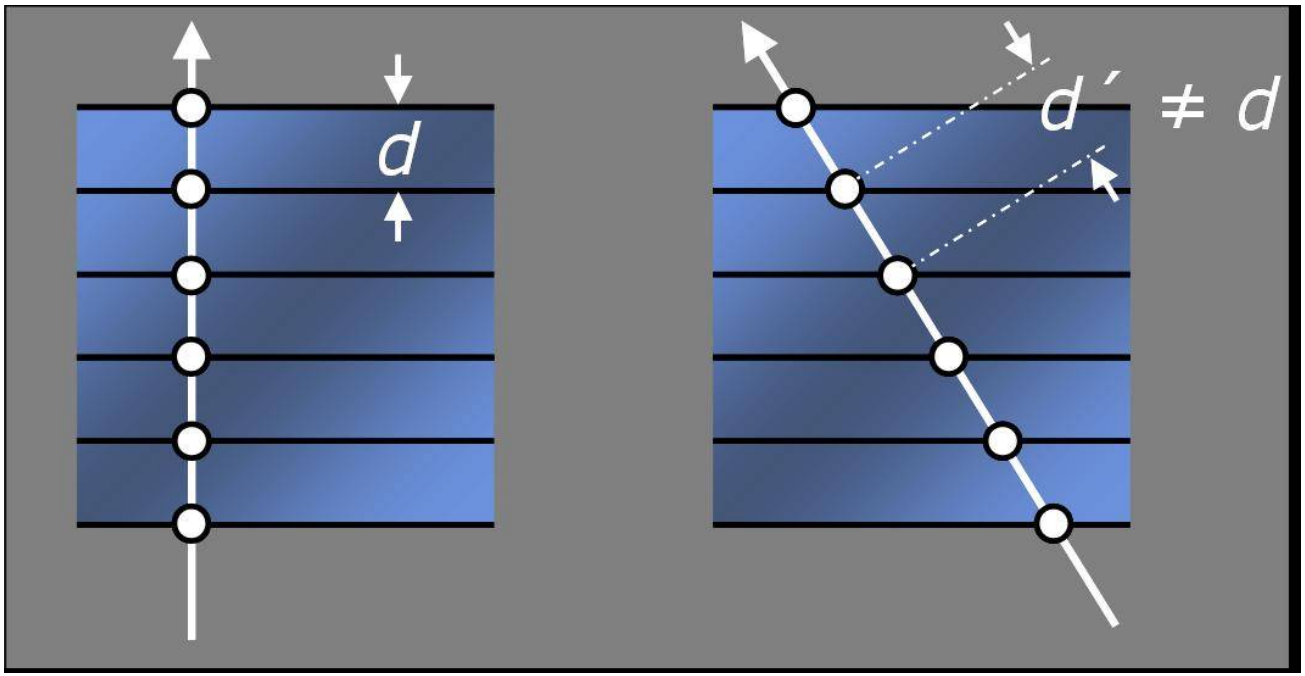


Figura: Distancias de muestreo que varían con el ángulo de visión

Texturas tridimensionales

El acceso a texturas tridimensionales es comparativamente más lento que el acceso a texturas bidimensionales.

Es hora de actualizar la información

El venerable diagrama del pipeline presentado en el *Red Book* no representa los avances de los últimos años en materia de hardware para computación gráfica. Ahora es mucho más flexible.

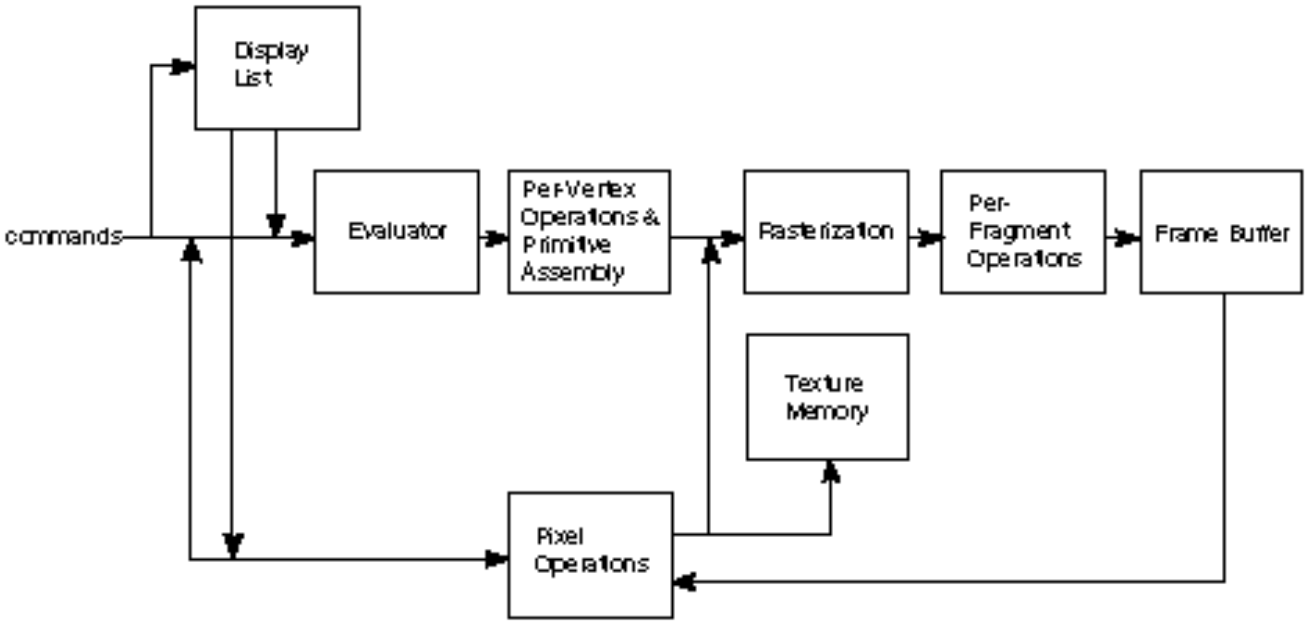


Figura: El esquema tradicional del pipeline

El concepto de shader

Un *shader* en el contexto de las tarjetas gráficas es un pedazo de código que reemplaza una parte del pipeline tradicional de OpenGL. A este pipeline sin modificar se le suele llamar *función fija* (fixed function). En particular, las dos partes del pipeline que puede reemplazar son el procesador de vértices (esto es un *vertex shader*), o el procesador de fragmentos (esto es un *fragment shader*).

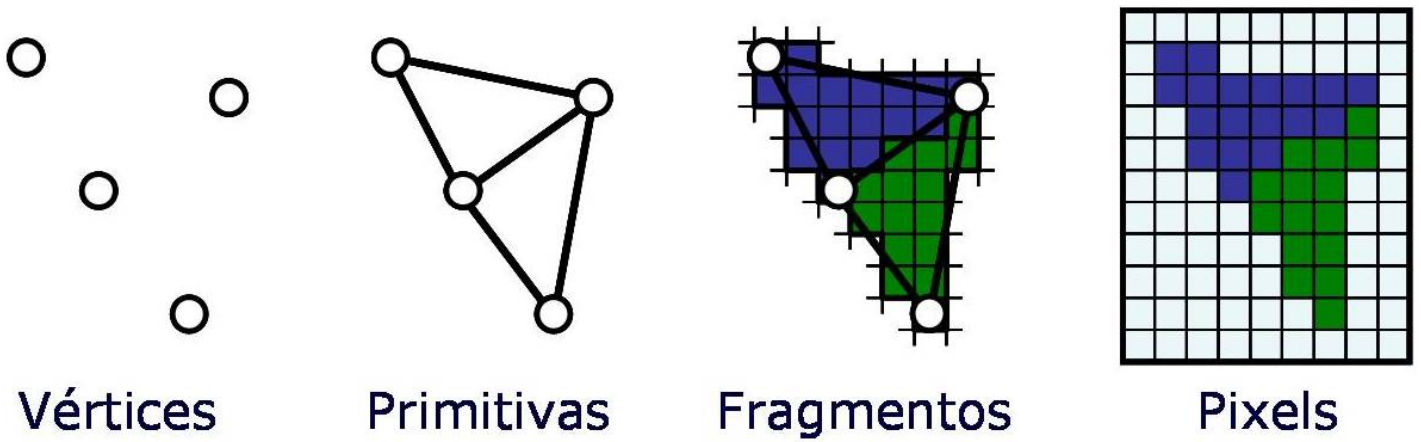


Figura: La transformación de vértices a píxeles

¿Dónde están los shaders en un programa?

Los shaders **NO** reemplazan a OpenGL ni a un lenguaje de propósito general (como C). Los shaders son programas que se cargan desde OpenGL. Actualmente son trozos de código escritos para un *assembler* específico para la tarjeta gráfica. Este assembler está definido por extensiones de OpenGL. Diversas instituciones se han dado a la tarea de crear lenguajes de alto nivel para programar shaders, de manera de ofrecer a los artistas una oportunidad de programar sus propios shaders sin necesidad de adentrarse en los detalles técnicos de la tarjeta. Ejemplos: nVidia está promoviendo su lenguaje Cg, y la ARB de OpenGL está promocionando el GL Shading Language (GLSL).

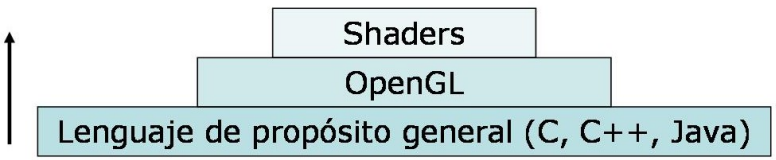


Figura: El contexto de los shaders en el programa

Ventajas

- El hardware de la tarjeta gráfica está especializado en operaciones con vectores, y con variables de punto flotante.
- Cada vértice se procesa por separado, al igual que cada fragmento. Esto ofrece posibilidades de paralelización que aceleran los cálculos.

Limitaciones

- El hardware altamente especializado hace difícil la programación general en los shaders, o la programación de algoritmos que no se presten para paralelizarse.
- Límites en el número de instrucciones y en el poder de expresividad de los shaders.
- Los cálculos se hacen a niveles muy bajos. El debugging no es una tarea sencilla siempre.

Beneficios de los shaders en nuestra tesis

- Reducción de la memoria utilizada en la tarjeta gráfica.
- Modificación del color y transparencia del tejido en tiempo de corrida.
- Reducción en los tiempos de cálculo de iluminación del volumen.

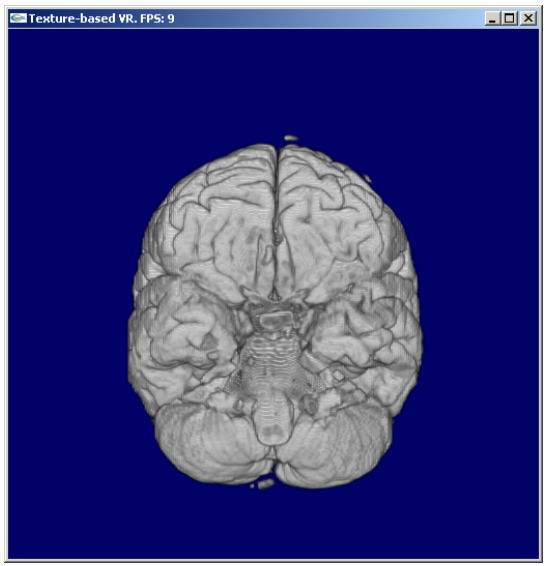


Figura: Modificación del color y transparencia de un tejido

Beneficios de los shaders en nuestra tesis

- Reducción de la memoria utilizada en la tarjeta gráfica.
- Modificación del color y transparencia del tejido en tiempo de corrida.
- Reducción en los tiempos de cálculo de iluminación del volumen.

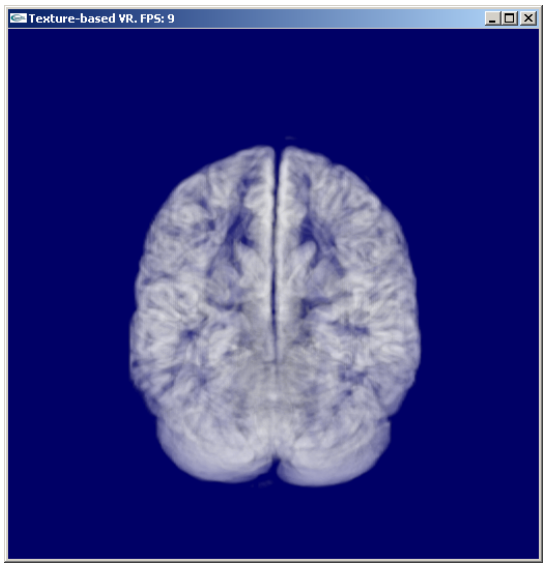


Figura: Modificación del color y transparencia de un tejido

Beneficios de los shaders en nuestra tesis

- Reducción de la memoria utilizada en la tarjeta gráfica.
- Modificación del color y transparencia del tejido en tiempo de corrida.
- Reducción en los tiempos de cálculo de iluminación del volumen.

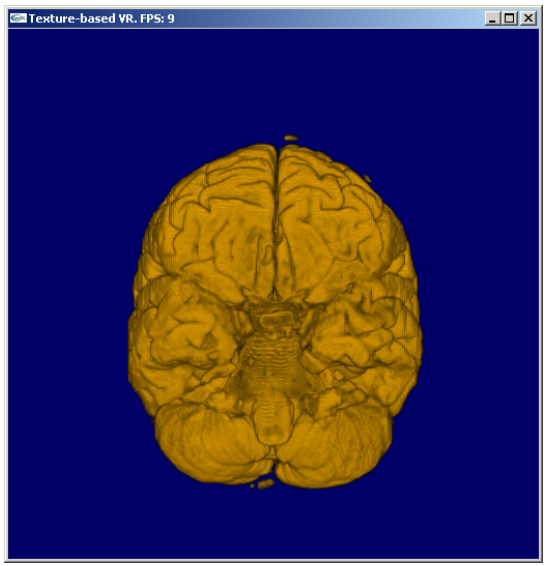


Figura: Modificación del color y transparencia de un tejido

Resumen I

- Volume Rendering es una representación 2D de información en 3D.
- Entre otras maneras de hacer rendering volumétrico está la representación mediante isosuperficies, el rendering volumétrico directo (mediante raycasting o mediante texturas), splatting, etc.
- En el rendering volumétrico directo el volumen de datos se intenta simular los fenómenos de absorción y emisión de la luz que atraviesa unas partículas. El fenómenos de dispersión es actualmente demasiado costoso en términos de computabilidad.

Resumen (y II)

- La técnica de Raycasting consiste en lanzar rayos, cada uno correspondiente con un pixel de la pantalla, hacia el volumen de datos. En cada rayo se toma una muestra del volumen de datos cada cierta distancia. Estas muestras luego se componen de atrás hacia adelante para obtener el color final.
- En rendering volumétrico basado en texturas se utilizan las capacidades de mapeo de texturas y de blending de las tarjetas gráficas de hoy en día para hacer la composición que se haría mediante raycasting.
- Los shaders son programas que reemplazan parte del pipeline de OpenGL, y extienden la funcionalidad de las tarjetas gráficas permitiendo hacer operaciones personalizadas sobre los vértices y sobre los fragmentos.

¿Preguntas?

¿Dónde bajar estas láminas?

Estas láminas están disponibles en:
`http://www.ldc.usb.ve/~ciro/clase-volumerendering.pdf`