FARAUTE Ciens. y Tec., 2(2): 54-61, 2007 ISSN 1698-7418
Depósito Legal PP200402CA1617

VISUALIZACIÓN DE VOLÚMENES ASISTIDO POR HARWARE GRÁFICO

Hardware-Assisted Volume Rendering

ANGELO G. GONCALVES, JUAN C. VARGAS. y JORGE E. RODRÍGUEZ

Universidad de Carabobo. Facultad de Ciencia y Tecnología. CEMVICC. Carabobo. Venezuela.
{ agoncalv, jcvargas, jrodrigu}@uc.edu.ve

Fecha de Recepción: 15/02/2007, **Fecha de Revisión:** 25/09/2007, **Fecha de Aceptación:** 30/10/2007

Resumen

En la actualidad, la adquisición y generación de datos en las más diversas áreas del quehacer humano mantiene un crecimiento exponencial, hasta el punto de que nuestra capacidad de análisis e interpretación de dicha información está por debajo de nuestra capacidad de generación de la misma. De allí la necesidad de crear nuevas técnicas y estrategias que faciliten su interpretación y estudio. En este sentido, la construcción de modelos visuales asociados a estos datos, mapas de temperatura, densidad, actividad, velocidad, o geometrías tridimensionales ajustadas a los datos, son herramientas cada vez más usadas en este proceso de asistencia a la interpretación. Esa es la misión de la visualización de volúmenes. Ahora bien, la verdadera utilidad de estas herramientas está íntimamente ligada al tiempo de respuesta y la calidad de la imagen generada. Con el desarrollo del hardware gráfico y el advenimiento de los GPU en computadores convencionales se ha abierto la posibilidad de explotar este potencial de procesamiento para acelerar, incluso hasta alcanzar tiempo real, los procesos de visualización de volúmenes antes condenados a ser ejecutados únicamente en el CPU. En este trabajo se presenta el desarrollo de una técnica de visualización de volúmenes asistido por hardware gráfico con buffer de textura 3D, alcanzando tiempos de visualización del volumen en tiempo real y calidad aceptable de la imagen en los modelos producidos. Asimismo, los tiempos de clasificación e iluminación han sido disminuidos hasta garantizar tiempos de respuesta interactivos.
Palabras claves: Visualización de volumen, textura 3D, hardware grafico.

Abstract

Nowadays, a huge amount of data is acquired and produced everyday in several human activities. In fact, the production of new data is greater than our ability to process it. Hence, strategies to analyze this data become more and more important everyday. Volume visualization has as main goal to provide visual models to make easy the study of volume datasets. However, response times and quality of the images generated are key issues in order to provide a useful tool. Fortunately, GPU are becoming very common in conventional PC, so we can exploit this feature in order to accelerate the volume rendering (VR) process using graphic hardware. In this paper, we present a hardware-assisted VR implementation using 3D texture buffer. Our implementation allows composition in real time.
Keywords: Volume visualization, 3D texture buffer, GPU.

1. Introducción

Actualmente el uso de las computadoras para visualizar conjuntos de datos con información volumétrica es una realidad en áreas tan diversas como el análisis geofísico, dinámica de fluidos, cálculos matemáticos complejos y diagnostico médico. La mayoría de estas áreas exigen la representación de objetos y/o fenómenos en tres dimensiones para facilitar el análisis y comprensión del objeto de estudio haciendo necesario el uso de diversas técnicas de computación gráfica.

La visualización de volúmenes ha sido el campo de estudio del área de visualización científica con mayor atención en los últimos años (Levoy, 1988; Westover, 1989; Todd, 1992; Lacroute, 1994; Meiβner *et al.*, 2000). A partir de un conjunto de datos volumétricos se puede obtener un artefacto visual que facilite su interpretación. Sin embargo, dependiendo del tamaño del conjunto de datos y de su complejidad (cantidad de propiedades presentes) es más o menos fácil lograr visualizaciones de alta calidad con tiempo de respuesta en tiempo real o, al menos, cercanos a la interactividad

En este orden de ideas, la utilización de hardware de propósito específico para la aceleración de operaciones graficas ha sido una alternativa planteada (Akeley, 1993). Sin embargo con la masificación de tarjetas gráficas de cierto poder en la mayoría de los computadores personales actuales, se plantean otras estrategias de bajo costo para acelerar los tiempos de visualización de volúmenes (Cullip *et al.*, 1993; Cabral *et al.*, 1994; Dachille *et al.*, 1998; Westerman *et al.*, 1998; Meiβner *et al.*, 1999; Rezk-Salama *et al.*, 2000; Engel Klaus *et al.*, 2002; Chen *et al.*, 2005; Xu *et al.*, 2005). En tal sentido, el aprovechamiento del buffer de textura 3D presente en las tarjetas graficas actuales permite delegar en el GPU algunas tareas que convencionalmente se debían hacer en el CPU. La posibilidad de realizar algunas de las operaciones en el GPU permitiría acelerar el tiempo de visualización.

Este artículo está estructurado de la siguiente forma. La sección siguiente aborda la literatura en el área. La sección 3 presenta el modelo de visualización de volúmenes usando buffer de textura 3D escogido para su implementación. La sección 4 presenta los resultados obtenidos con la implementación hecha y la sección 5 presenta las conclusiones y el trabajo futuro propuesto.

2. Trabajos relacionados.

La visualización de volúmenes consiste en generar un artefacto visual a partir de un conjunto de datos con información volumétrica de un objeto o fenómeno que se desea analizar. Sin embargo, cada día los conjuntos de datos son de mayor magnitud y se hace necesario el desarrollo de estrategias que permitan visualizarlos en tiempo real o cercano a la interactividad. Con el desarrollo del hardware para despliegue de gráficos se ha hecho común, en un computador personal comercial, la presencia de tarjetas gráficas poderosas a bajo costo. Estas tarjetas cuentan con memoria y procesador exclusivo para la manipulación y despliegue de primitivas graficas (GPU) y pueden ser aprovechados para asignarle tareas del proceso de visualización de volúmenes que anteriormente estaban destinadas al CPU. La visualización de volúmenes aceleradas por hardware grafico es una técnica de estudio reciente, en donde las tarjetas graficas de última generación juegan un papel importante en la optimización de algoritmos de representación de volúmenes de proyección directa. Con la aparición de las tarjetas gráficas dotadas de buffer de textura 3D administradas por el propio

GPU se abrió una nueva tendencia en la visualización de volúmenes.

(Akeley, 1993), es uno de los primeros investigadores que propone el uso del hardware gráfico para acelerar la velocidad de despliegue de escenas 3D texturizadas. Sin embargo, en su caso es una arquitectura de propósito específico de alto costo como lo es la RealityEngine™ de Silicon Graphics.

(Cullip *et al.*, 1993) presentan la primera aplicación de visualización de volúmenes explotando el hardware grafico de la RealityEngine™ usando el buffer de textura 3D.

(Cabral *et al.*, 1994) demuestran matemáticamente la equivalencia entre el proceso de composición volumétrica de la integral Low-Albedo y la composición implementada usando las primitivas de mezclado del hardware gráfico.

(Dachille *et al.*, 1998; Westerman *et al.*, 1998 y Meiβner *et al.*, 1999) por su parte, también proponen una implementación realizando la composición volumétrica en el hardware grafico, pero nuevamente sobre estaciones de trabajo para gráficos de alto costo.

(Rezk-Salama *et al.*, 2000) finalmente, dan el salto hacia los computadores personales convencionales y proponen una nueva estrategia sobre esta arquitectura de bajo costo que mejora significativamente el rendimiento y la calidad de la imagen, basada en el uso de buffer de texturas 2D, más comúnmente encontrado en tarjetas gráficas convencionales, haciendo uso del multitexturizado en múltiples etapas.

(Klaus *et al.*, 2002; Chen *et al.*, 2005 y Xu *et al.*, (2005) continúan los desarrollos sobre computadores personales, explotando el potencial de cálculo del GPU, proponiendo estrategias de aceleración adicionales como terminación temprana del rayo, precálculo del sombreado y uso de *shaders* para la iluminación.

En este artículo se presenta la implementación de una aplicación completa de visualización de volúmenes, usando el buffer de textura 3D de una tarjeta gráfica de última generación sobre un computador personal.

3. Visualización de Volúmenes Usando Texturas 3D

La visualización de volúmenes convencional se logra integrando la luz a lo largo de su recorrido a través del volumen de datos. A medida que la luz atraviesa el volumen, se va calculando la cantidad de luz que es reflejada por cada partícula y la cantidad de continua su recorrido a través de nuevas partículas. Este proceso de integración a lo largo del rayo de luz se conoce como composición volumétrica y es repetido para múltiples de rayos de luz que parten del observador. La composición volumétrica de cada rayo es realizada en su totalidad por el CPU.

En un esquema basado en hardware, el volumen de datos es previamente cargado en el buffer de textura 3D. Es decir, el volumen de datos es concebido como una textura 3D. Por lo tanto, el proceso de composición volumétrica será realizado como una operación de despliegue del contenido de la textura 3D y por lo tanto, la operación es ejecutada por el GPU, descargando al CPU de esta tarea y alcanzando cierto nivel de paralelismo en el proceso de composición como se explicará más adelante.

Una vez verificada la disponibilidad de buffer de textura 3D, el volumen de datos es cargado en dicho buffer. Esto se hace a través de las directivas de las librerías de gráficos 3D. En particular, con OpenGL dicha directiva es *glTexImage3D*.

El proceso de carga del volumen en el buffer de textura supone, al mismo tiempo, un proceso de clasificación, a través del cual a cada valor de densidad del volumen de datos, se le asocia un valor de opacidad asociado. Dicho valor puede ser modificado posteriormente en forma interactiva, pero cada vez que sea modificado, el proceso de clasificación y por ende de carga del buffer de textura debe ser repetido.

Una vez cargado y clasificado el volumen en el buffer de textura, se lleva a cabo el proceso de composición volumétrica. En este proceso, los vóxeles se deben componer coherentemente de acuerdo al trayecto de la luz a través del volumen, con el objeto de obtener una imagen con apariencia volumétrica. Este proceso, en los enfoques de visualización de volúmenes asistidos por hardware grafico se realiza como una operación de mezclado (*Blending*) entre los diferentes valores de textura presentes en el buffer. Esta operación permite "transparentar" los vóxeles combinando el valor de color del fragmento actual en el *framebuffer* con el valor de color del fragmento siguiente, siempre de acuerdo al nivel opacidad pre-asignado en la fase de clasificación.

4. Detalles de la Implementación realizada

El esquema usado para la implementación de la visualización de volúmenes usando el buffer de textura 3D, consistió en introducir en el buffer de textura, múltiples planos paralelos al plano de imagen, calcular la región de cada plano que yace en el interior del espacio de textura y componerlos con la operación de mezclado. En este esquema se requiere calcular las intersecciones de los *n* planos con cada una de las aristas que conforman al cubo imaginario que delimita el espacio de textura, generando un polígonos-intersección y de esta manera obtener las coordenadas de textura asociadas a los vértices de dicho polígonos (ver Fig. 1).

La orientación de los planos depende de la ubicación del punto de visión, el cual es modificado interactivamente. Por lo tanto, cada vez que el punto de visión cambia, todos los planos son recalculados de acuerdo a la nueva normal definida por la línea de visión.

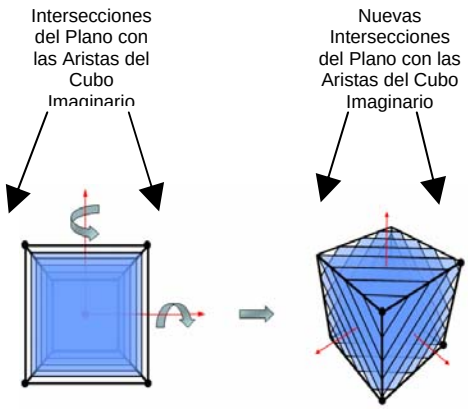

Fig. 1. Inserción de planos en el espacio de textura

4.1 Clasificación de los Datos

El proceso de clasificación usando buffer de textura 3D, consiste en almacenar junto con la densidad asociada a un determinado material del volumen de datos, un valor de opacidad en el buffer de textura. Este valor será utilizado en el proceso de composición volumétrica posteriormente. Para efectos de poder modificar los valores de opacidad en forma interactiva, es necesario mantener una copia del volumen original en memoria. Cada vez que los parámetros de clasificación sean modificados, el buffer de textura será nuevamente cargado con los nuevos valores de opacidad asociados

4.2 Iluminación de los Datos

Se utilizó el modelo de iluminación de Phong para dar color al volumen. Los componentes de iluminación tomados en

cuenta fueron el componente de iluminación difusa (*Id*) y el componente de iluminación ambiental (*Ia*). El componente de iluminación especular fue descartado para evitar el cálculo de la iluminación cada vez que el punto de visión es modificado. Las formulas para el cálculo de la iluminación son las siguientes:

$$I = IaKa + IdKd \quad (1)$$

Donde:

$$Id = Id' (N*L) \quad (2)$$

Ia es la intensidad de la iluminación ambiental. *Id'* es el color del objeto. *N* es el vector normal a la superficie y *L* es la dirección de la luz. *Ka* es el coeficiente empírico que depende de las propiedades ópticas del material del objeto ($0 \leq Ka \leq 1$) y *Kd* es el coeficiente empírico de reflexión que depende de la longitud de onda de la luz.

El vector *N*, normal de la superficie, es aproximada por el gradiente en escala de grises del volumen de datos. Dicho gradiente es expresado como:

$$\nabla f(x,y,z) = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z} \right) \quad (3)$$

Donde cada componente es calculada por diferencias centrales divididas:

$$\frac{\partial f}{\partial x} = f(x+1,y,z) - f(x-1,y,z) \quad (4)$$

$$\frac{\partial f}{\partial y} = f(x,y+1,z) - f(x,y-1,z) \quad (5)$$

$$\frac{\partial f}{\partial z} = f(x,y,z+1) - f(x,y,z-1) \quad (6)$$

4.3 Composición Volumétrica

Al tener los polígonos-intersección asociados con sus respectivas coordenadas de textura, se procede a realizar la composición volumétrica, a través de la operación de mezclado (*blending*) de *texels* o valores de textura, tomando en cuenta el valor de opacidad asociado en tiempo de clasificación. Esta operación se lleva a cabo usando la función *Blending* de OpenGL. La función de mezclado o *Blending* recibe dos parámetros, el primero es el fuente (*source*) y el segundo es el destino (*destination*) y los combina para obtener el color final. Para reproducir con exactitud la integral low-albedo de composición volumétrica de atrás hacia delante (*BTF: Back to Front*), se usaron como factor fuente el parámetro **GL_SRC_ALPHA** y como factor destino

GL_ONE_MINUS_SRC_ALPHA.

En definitiva, la función *Blending* quedaría configurada como sigue:

glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);

Para comprender cómo la función *Blending* reproduce el proceso de composición volumétrica de la integral Low-Albedo, veamos el siguiente desarrollo. Sea C_s el color (R, G, B) fuente o entrante y C_d el color existente en el *framebuffer*. Sea, además, α_{src} el valor Alfa de opacidad asociado al Color entrante. El color C resultante de hacer el *blending* entre C_s y C_d es:

$$C = (C_s S + C_d D) \quad (7)$$

donde *S* y *D* están determinados por los parámetros establecidos en la función de mezclado. En nuestro caso:

$$S = \alpha_{src}$$

$$D = 1 - \alpha_{src}$$

Desarrollando la función de mezclado, partiendo de la formula de composición del

Blending, a través de los diferentes planos de composición obtenemos:

$I_1 = C_1\alpha_1$

$I_2 = C_2\alpha_2 + (C_1\alpha_1)(1 - \alpha_2)$

$I_3 = C_3\alpha_3 + [C_2\alpha_2 + (C_1\alpha_1)\ (1 - \alpha_2)]\ (1 - \alpha_3)$

$I_4 = C_4\alpha_4 + \{C_3\alpha_3 + [C_2\alpha_2 + (C_1\alpha_1)\ (1 - \alpha_2)]\ (1 - \alpha_3)\}\ (1 - \alpha_4)$

Donde:

$I_4 = C_4\alpha_4 + C_3\alpha_3\ (1 - \alpha_4) + [C_2\alpha_2\ (1 - \alpha_3)\ (1 - \alpha_4)] + C_1\alpha_1\ (1- \alpha_2)\ (1 - \alpha_3)\ (1 - \alpha_4)$

En general, desarrollando el n-ésimo término:

$I_n = C_n\alpha_n + C_{n-1}\alpha_{n-1}\ (1 - \alpha_n) + \ldots + C_1\alpha_1\ (1- \alpha_2)\ (1 - \alpha_3)\ (1 - \alpha_4)$

Expresando la ecuación en términos de sumatorias y productorias obtenemos:

$$In = \sum_{i=n}^{0} C_i\alpha_i \prod_{j=n}^{i}(1-\alpha j) \qquad (8)$$

Donde (8) es la expresión discretizada de la integral Low-Albedo de composición volumétrica. En conclusión, la función *Blending* con la parametrización indicada reproduce el proceso de composición volumétrica descrito por la integral Low-Albedo.

Vale destacar que, mientras las implementaciones en software de este proceso de composición volumétrica, deben resolver esta operación en el CPU para cada rayo emitido desde el plano de imagen; la implementación que se propone delega esta responsabilidad en el GPU, realizando el proceso de composición en paralelo para todos los pixeles del *framebuffer*. Esta es la razón por la cual el proceso de proyección directa de volumen asistido por hardware gráfico, incrementa la velocidad de visualización hasta obtener tiempo real, en contraposición con los enfoques basados en software, donde difícilmente se alcanzan tiempos cercanos a la interactividad.

5. Resultados

Estos resultados fueron obtenidos ejecutando la aplicación desarrollada, sobre un PC Pentium4, 1.66Ghz, con una tarjeta gráfica Nvidia GeForce PX 6800 con 256MB de memoria de video.

La siguiente tabla, muestra los tiempos (medidos en segundos) de la técnica desarrollada en este trabajo (Hw) y la técnica de Ray Casting(RC) basada en software, para tres volúmenes típicos de la literatura:

	Dimensiones	**Tamaño en MB**	**Hw (Seg)**	**RC (Seg)**
Aneurisma	256x256x256	16,77MB	0.016	0.208
Pié	256x256x256	16,77MB	0.016	0.208
Motor	256x256x128	8,38MB	0.015	0.158

Tabla 1. Comparación en tiempo de visualización

Los resultados evidencian la obtención de tiempo real en la composición y visualización del volumen usando nuestra implementación basada en hardware para volúmenes de datos de hasta 256^3. Ya que permite mostrar más de 30 cuadros por segundo (<0.0333 seg. por cuadro).

Las figuras 2 y 3, muestran un aspecto de las visualizaciones producidas por nuestra implementación para los conjuntos de datos tomográficos asociados a un pie humano y un bloque de motor.

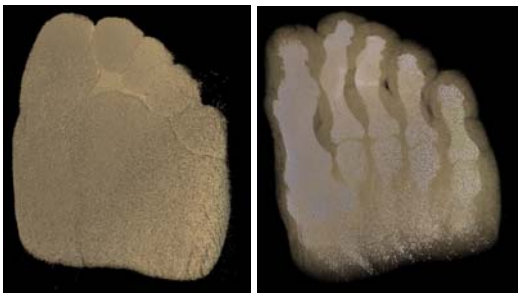

Fig. 2. Pie Humano opaco y traslucido.

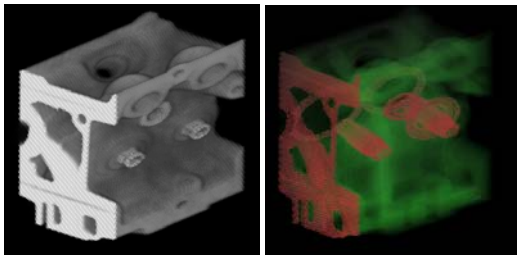

Fig. 3. Bloque de motor opaco y traslucido.

6. Conclusiones

Se desarrolló e implementó una técnica de visualización de conjuntos de datos volumétricos explotando el potencial del buffer de textura 3D de las tarjetas gráficas convencionales. Adicionalmente se logró, con el algoritmo desarrollado, alcanzar tiempos de respuesta interactivos en la visualización del conjunto de datos volumétricos aún con movimientos arbitrarios de la cámara.

7. Agradecimientos

Este trabajo ha sido posible gracias al financiamiento hecho por el CDCH-UC a través del proyecto CDCH 2005-12.

8. Bibliografía

Brian, C., N. Cam & J. Foran. (1994). Accelerated Volume Rendering and Tomographic Reconstructions Using Texture Mapping Hardware. *Symposium on Volume Visualization*. 91-98, Washington D.C. ACM.

Chen, Y., L. Peng, W. Bo-liang & X. Xiu-ying. (2005). High Quality Volume Rendering using Programmable Graphics Hardware. Department of Computer Science. Xiamen University. China.

Dachille, F., K. Kreeger, B. Chen, I. Bitter & A. Kaufman. (1998). High-Quality Volume Rendering Using Texture Mapping Hardware. Center for Visual Computing and Department of Computer Science. State University of New York at Stony Brook.

Engel, K., M. Kraus & T. Ertl. (2002). High-Quality Pre-Integrated Volume Rendering Using Hardware-Accelerated Pixel Shading. Visualization and Interactive Systems Group. University of Stuttgart. Germany.

Kurt, A. (1993). Reality Engine Graphics. *Computer Graphics*, SIGGRAPH 93, Anahiem. ACM.

Levoy, M. (1988). Display of Surfaces from Volume Data. *IEEE Computer Graphics and Applications.* 8(3):29-37.

Meiβner, M., J. Huang, D. Bartz, K. Mueller & R. Crawfis. (2000). A Practical Evaluation of Popular Volume Rendering Algorithms. *Volume Visualization*. 2000: 81-90.

Meiβner, M., U. Hoffmann & W. Straβer. (1999). Enabling classification and shading for 3D texture mapping based volume rendering using OpenGL and extensions. Visualization'99.

Philippe, L. (1994). Fast Volume Rendering Using a Shear-Warp Factorization of the Viewing Transformation. Computer Systems Laboratory Stanford University, Computer Science Department.

Rezk-Salama, M., K. Engel, M. Bauer, G. Greiner & Ertl Thomas. (2000). Interactive Volume Rendering on Standard PC Graphics Hardware Using Multi-Texturing and Multi-Stage Rasterization. Eurographics / SIGGRAPH Workshop on Graphics Hardware. 109–118. Interlaken, Switzerland.

Timothy, C. & N. Ulrich. (1993) Accelerating Volume Reconstruction with 3D Texture Hardware. Technical Report TR93-027, University of North Carolina, Chapel Hill.

Todd, E. (1992). A Survey of Algorithms for Volume Visualization. *Computer Graphics*. 26(3): 194-201.

Westover, L. (1989). Evaluation for Volume Rendering. Symposium on Volume Visualization, Siggraph.

Westerman, R & T. Ertl. (1998). Efficiently using graphics hardware in volume rendering applications. SIGGRAPH, Computer graphics Conference.

Xu, F. & M. Klaus. (2005). Accelerating Popular Tomographic Reconstruction Algorithms on Commodity PC Graphics Hardware. *IEEE Transactions on Nuclear Science.* 52(3): 654-663.