

Volume Rendering: MIP, Ray Casting & Translucency

Cristian C. D'Alessandro, *cristian.ccd@gmail.com*

Facultad de Ingeniería y Ciencias Exactas y Naturales, Universidad Favaloro

Abstract— Partiendo de las bases del Volume Rendering se utilizó la técnica de Volume Ray Casting para reconstruir en 3D las figuras obtenidas por estudios de CT y MRI convertidas a BMP de 8 bits. El procesamiento de los datos y la modificación del algoritmo permitieron realizar un estudio más profundo utilizando las técnicas de MIP (Maximum Intensity Projection) y Translucency. Este programa demuestra las posibilidades de diagnóstico y aprovechamiento de datos que se puede realizar gracias a la utilización de estas técnicas en conjunto.

I. INTRODUCCION

La técnica de Volume Rendering ^[1] consiste en la proyección 2D de una muestra de un set de datos en 3D. Una técnica particular es la de Volume Ray Casting, un volumen de datos es almacenado en memoria y se genera un plano de visualización. Por cada pixel del plano se genera un rayo que atravesará el volumen. Al atravesarlo, el rayo ira recopilando información relevante y esta se proyecta sobre el plano, dándonos una representación del volumen desde el punto de vista del observador. Es una técnica muy utilizada en la visión computacional, imágenes médicas, video juegos y películas, como “Cars” de la empresa Pixar. ^[2]

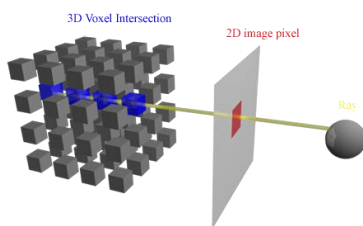


Figura 1.1, procedimiento típico para el desarrollo del algoritmo de Ray Casting ^[3].

MIP es una modificación de esta técnica, en la que por cada pixel del plano se genera el rayo, pero este atravesará todo el volumen de datos y proyectará solo el máximo. Algo muy útil a la hora de realizar MRA ^[4] (Magnetic Resonance Angiography) y PET. Translucency es otra modificación de la técnica de Volume Ray Casting que en este caso se encarga de eliminar tejidos no deseados de un estudio médico, generalmente piel. Obteniéndose una representación del esqueleto de la persona, o los vasos sanguíneos, de acuerdo al área de estudio.

El primer paso del Volume Rendering ^[5] consiste en elegir el tipo de superficie donde se visualizará la muestra, la

apreciación de las zonas de la imagen dependerá de la distribución de los rayos sobre el volumen de datos. A mayor cantidad de rayos que atraviesan este volumen, mayor será la información recopilada y se podrá generar una visualización mucho más fiel. El plano debe orientarse de acuerdo a la posición del observador que se desee, se utilizan figuras geométricas que engloben al volumen de datos para poder ubicar al plano sobre un punto de esta figura y realizar las proyecciones. Para tener una referencia donde se encuentra el volumen, la normal del plano debe indicar la dirección de los rayos. La decisión de corte del rayo reside en el tipo de técnica utilizada, como también de modificadores ajenos a la técnica que permiten una segmentación, un filtrado o una ponderación de niveles de gris, o una distancia de penetración.

Este tipo de algoritmos suelen ser de un costo computacional elevado, por lo que una buena decisión de la condición de corte es muy importante para obtener un algoritmo eficiente y reducir los tiempos de renderización.

II. DESARROLLO DEL PROGRAMA

El objetivo del programa era en principio la realización de la técnica de MIP. Debido a su semejanza con las técnicas de Volume Rendering y Translucency se optó por realizar un programa más abarcativo que permita al usuario tener una amplia gama de variantes para realizar el estudio. Se integró la técnica MIP, Volume Rendering y Translucency, junto con modificadores: Filtrado trilinear para el volumen de datos, filtrado bilinear para el plano, isodata, segmentación, profundidad del rayo, filtrados y aplicación de LUT (Look Up Tables). Este último cobra importancia a la hora de realizar la técnica de Translucency.

El primer paso del desarrollo del programa fue definir el tamaño del plano de proyección y la estructura del mismo, a cada elemento del plano le van a pertenecer 3 coordenadas correspondientes al elemento del volumen en estudio y el valor del nivel de gris. El plano tendrá una normal, que vamos a definir como la dirección de un haz normalizado y un punto dentro del plano ^[3], de este modo la referencia de la dirección de la normal nos sirve como guía para la generación de rayos. El tamaño fue definido de acuerdo a las imágenes utilizadas, con un tamaño de 512x512 con 256 niveles de gris, en formato BMP. Todos los slices son almacenados en un cubo de dimensiones correspondientes, con alocaión de memoria dinámica, para permitir en el futuro imágenes de otros tamaños. Una vez definidas las características del plano es necesario englobar al volumen de datos del estudio. El elemento geométrico que se optó fue una esfera con radio igual a la distancia entre vértices

opuestos del cubo, de esta manera nos aseguramos que el volumen se encuentre completamente contenido dentro de la esfera y la forma de esta nos permite un solo punto de contacto entre el plano y la esfera, ideal para la orientación hacia el volumen.

El punto de visión del observador seleccionado produce una rotación en el plano, que se realiza sobre el eje y luego se traslada al punto original, teniendo en cuenta que la normal siempre apunte al centro de la esfera. Los ángulos de rotación del plano se calculan en base a la posición del observador, dados una dirección cartesiana de la normal y sabiendo que esta se encuentra sobre la esfera, podemos calcular un ángulo de azimut y una elevación:

$$Azimuth = \text{atan}\left(\frac{Y}{X}\right) \quad (1)$$

$$Elevación = \text{acos}(Z) \quad (2)$$

Siendo los ejes cartesianos representados en la figura 1.2.

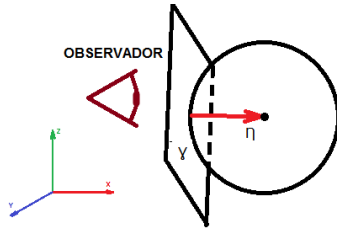


Figura 2.1. El plano siempre apuntará al interior de la esfera que contiene todos los datos de las imágenes, lo que nos asegura una representación sobre el plano.

Para las rotaciones del plano se utilizaron matrices de rotación ^[6] en todos los ejes,

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix} \quad (3)$$

$$R_y(\theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \quad (4)$$

$$R_z(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (5)$$

El cálculo del ángulo de rotación es relativo a la posición anterior del plano, información que se obtiene fácilmente de la normal al plano. Una vez rotado el plano y orientado, se recorren los elementos del plano 1 por 1 y se generan los rayos con direcciones características dadas por,

$$Rx = Nx * \alpha + Px \quad (6)$$

$$Ry = Ny * \alpha + Py \quad (7)$$

$$Rz = Nz * \alpha + Pz \quad (8)$$

Donde Rx, Ry, Rz son las direcciones en los ejes correspondientes del rayo, Nx, Ny, Nz son las direcciones de la normal al plano, Px, Py, Pz son los puntos del pixel donde se origina el rayo y α es el valor que irá iterando del rayo al atravesar el volumen de datos.

III. RAY CASTING E INTERPOLACIÓN TRILINEAR

Este es el caso principal y base de las técnicas que luego se comentarán en el trabajo, principalmente se diferencia de las otras técnicas por la forma en que corta el rayo. Al atravesar el volumen, el rayo se detendrá ante el primer elemento que se encuentre por encima del umbral, el cual será proyectado al plano. De esta manera lo que observaremos es el contorno del volumen de datos, pero no tendremos acceso a los voxels interiores. El cálculo de la trayectoria del rayo no es exacta, lo que introduce un error ya que el rayo pasa por el voxel cuyas coordenadas están dadas por el entero del cálculo de trayectoria (6), (7) y (8), es por lo que se introduce una mejora, la interpolación trilinear ^[8]. Esta difiere de la bilineal debido a que se encuentra en un volumen, el cálculo del valor del pixel entonces estará dado por el promedio de los 8 vecinos multiplicados por las distancias a los ejes. El resultado ^[9] es una imagen más suave y agradable a la vista.

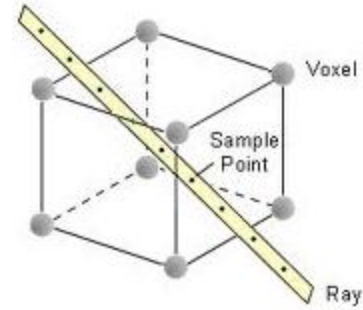


Figura 3.1. Justificación de la interpolación trilinear

$$\begin{aligned} \text{valor} = & [(1 - dz) * (1 - dy) * (1 - dx) * v1 + (1 - dz) \\ & * (1 - dy) * (1 + dx) * v2 + (1 - dz) \\ & * (1 + dy) * (1 - dx) * v3 + (1 - dz) \\ & * (1 + dy) * (1 + dx) * v4 + (1 + dz) \\ & * (1 - dy) * (1 - dx) * v5 + (1 + dz) \\ & * (1 - dy) * (1 + dx) * v6 + (1 + dz) \\ & * (1 + dy) * (1 - dx) * v7 + (1 + dz) \\ & * (1 + dy) * (1 + dx) * v8] / 8 \end{aligned} \quad (9)$$

V1 a V8 corresponden a los voxels vecinos, dx, dy y dz corresponden a la parte decimal del valor de la coordenada del voxel.



Figura 3.2. Volume rendering con filtrado trilinear (izquierda) y sin filtrado (derecha)

IV. MIP

En MIP^[7] la iteración de α en (6), (7) y (8) acaba cuando el rayo llega al extremo opuesto de la esfera, ya que este algoritmo selecciona el valor máximo del volumen de datos y lo proyecta al plano. Esta técnica falla al producir una pérdida de información de la imagen, al quedarnos solo con el máximo, toda sensación de profundidad se pierde y solo recabamos el contorno de los vasos sanguíneos o los huesos que corresponden a los valores más altos de intensidad.



Figura 4.1. MIP

Sin embargo sigue teniendo importancia para ciertos tipos de diagnósticos. Este inconveniente se soluciona realizando cortes del rayo a distintas profundidades y con distintos ángulos, de tal manera de recomponer la imagen y devolverle la sensación de profundidad.

V. TRANSLUCENCY

El rayo atraviesa toda la esfera, pero se realiza el promedio de todos los voxeles que superan el umbral. Antes de realizar el promediado se aplica una LUT que pondera los pixeles de mayor intensidad y atenúa los de menor intensidad, siguiendo la formula dada por (10)

$$LUT(x) = 255 * e^{\frac{(x-256)^2}{(2*20^2)}} \quad (10)$$

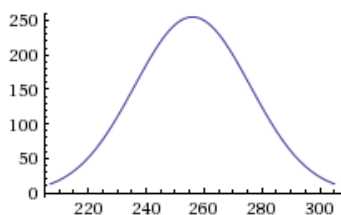


Figura 5.1. LUT utilizada en Transparency

Se utilizó una función gaussiana por la característica de atenuación de los valores de intensidad vecinos y la posibilidad de cambiar el ancho y alto de la misma de manera sencilla. Experimentalmente fue la mejor opción a la hora de aplicar una ponderación.



Figura 5.2. Transparency

VI. HERRAMIENTAS DE OPTIMIZADO

Para poder potenciar la utilización del software se incluyeron herramientas que mejoran la presentación de los resultados. Un filtrado bilineal, isodata, filtros pasa altos y pasa bajos, selección de profundidad de penetración, selección de umbrales para segmentación, cambio de brillo y contraste, ecualización, histograma, zoom y distintas LUTs para el ponderado de intensidades de acuerdo al rango deseado (bajo, medio, alto). Es importante que para la elección del ángulo del observador se cuente con una vista previa que permita dicha elección, por lo que se realizó la técnica de Volume Rendering pero con un subsampling de 8 voxeles, es así que se muestrea cada 8 pixeles, pero se mejora con una interpolación cada 2 dada por el promedio de los vecinos más cercanos distintos a 0. Dando una sensación de contorno más definida y más útil para la selección de la vista.

VII. MEJORAS Y LIMITACIONES

En un análisis de tiempos, el programa responde a un $O(n^3)$, debido a la cantidad de ciclos anidados, para un estudio de 150 imágenes de 512x512 se realizan un total de 193 millones de iteraciones si los rayos atraviesan toda la esfera, por lo que es esencial limitar la cantidad de instrucciones, y condicionamientos dentro de los ciclos. Dado que el procesamiento es por CPU, el rendimiento estará dado por el microprocesador de la PC y se verá severamente limitado. Una solución inmediata sería la renderización por GPU dedicado, disminuyendo drásticamente los tiempos de cálculo. Existen técnicas que mejoran drásticamente los tiempos de ejecución: árboles y métodos recursivos^{[11][12]}

VIII. CONCLUSIONES Y AGRADECIMIENTOS

El trabajo demostró que el procesamiento de imágenes en aplicaciones médicas posee un gran potencial para el médico y puede considerarse como una herramienta fundamental para todo profesional de la salud. En particular las técnicas de recabado de información de CT, PET y MRI como las mencionadas anteriormente poseen un relación íntima con estos dispositivos ya que los programas como el presentado

en este informe son los que permiten explotar el potencial de estos dispositivos.

Este trabajo fue posible gracias a la colaboración de los docentes de la materia y a los consejos de colegas de la cursada, que permitieron solucionar los inconvenientes durante el desarrollo del software.

IX. REFERENCIAS

- [1] http://en.wikipedia.org/wiki/Volume_rendering
- [2] *Ray Tracing for the Movie 'Cars', Pixar Studios*
PH Christensen et al.
- [3] *Interactive Ray Tracing for Volume Visualization*
S Parker et al.
- [4] http://en.wikipedia.org/wiki/Magnetic_Resonance_Angiography
- [5] http://en.wikipedia.org/wiki/Volume_ray_casting
- [6] http://en.wikipedia.org/wiki/Rotation_matrix
- [7] http://en.wikipedia.org/wiki/Maximum_intensity_projection
- [8] http://en.wikipedia.org/wiki/Trilinear_interpolation
- [9] http://www.grc.nasa.gov/WWW/winddocs/utilities/b4wind_guide/trilinear.html
- [10] *Volume Rendering in Three-Dimensional Display of SPECT Images*
- [11] <http://www.cs.princeton.edu/courses/archive/fall00/cs426/lectures/raycast2/raycast2.pdf>
- [12] <http://www.google.com/patents?id=BYyFAAAAEBAJ&printsec=abstract&zoom=4#v=onepage&q&f=false>