

Visualización de Volúmenes: Aplicaciones en Ciencias Geológicas

Silvia Castro¹, Miguel Danzi², Claudio Delrieux², Marcelo Larrea¹ y Andrea Silvetti¹
Instituto de Ciencias e Ingeniería de Computación
Universidad Nacional del Sur
Alem 1253 – Bahía Blanca – ARGENTINA
usdelrie@criba.edu.ar

Abstract

Se desarrollan técnicas y presentan detalles de implementación de un sistema de visualización de volúmenes. El mismo está aplicado a un problema particular de ciencias geológicas, pero puede ser utilizado para la representación de datos volumétricos de cualquier origen. La representación de sólidos traslúcidos se efectúa a partir de una descripción de su superficie, con modelos usuales de aproximación de superficies (B-splines). Cuando la topología del sólido a representar es conocida a priori, esta aproximación se puede realizar con splines sobre dominios rectangulares. En cambio, si la topología es desconocida o puede cambiar, se utilizan B-splines sobre dominios triangulares. La densidad volumétrica y el modelo de iluminación se calculan aproximando el modelo de Blinn-Kajiya con el espesor de los sólidos. El cómputo de dicho espesor se realiza en forma sencilla a partir de la conversión-scan de la superficie del sólido, por lo que los resultados son obtenidos en tiempos relativamente pequeños.

¹Departamento de Ciencias de la Computación

²Departamento de Ingeniería Eléctrica

1 Introducción

La *visualización científica* es una de las tecnologías derivadas de las Ciencias de la Computación que actualmente están revolucionando con mayor fuerza las metodologías de investigación científica en todos sus campos [7, 8, 36]. Por visualización se entiende el empleo de técnicas derivadas de la computación gráfica utilizadas para la representación de datos científicos de diverso tipo [1, 31]. Dentro de la investigación en visualización científica, la representación de datos volumétricos se destaca por las dificultades computacionales que plantea, pero al mismo tiempo concentra la mayor atención en la investigación actual [33, 37, 45]. Esto es así porque constituye una de las innovaciones más importantes y de mayor aplicabilidad al producir una adecuada representación gráfica computacional de datos que por una u otra razón no pueden representarse en términos de gráficos convencionales [39, 46, 47].

La representación volumétrica de datos es actualmente de gran utilidad en la investigación científica en temas tan diversos como en matemática, medicina, ciencias naturales e ingeniería [16, 23], y es utilizada para representar datos que pueden provenir de sensores, como en el caso de tomógrafos o de satélites, o bien pueden provenir de tareas computacionales anteriores, como por ejemplo de simulaciones o de análisis por elemento finito. Al mismo tiempo, los resultados de la visualización volumétrica de estos datos no son meramente una representación cuantitativa de los mismos, es decir, no se busca necesariamente la presentación fiel de valores. Por el contrario, se busca un entendimiento global de determinadas propiedades del modelo o de la simulación que produjo los datos [21, 26]. Estos objetivos son sumamente exigentes en términos de tecnología, tanto de hardware como de software [6, 28].

En este trabajo presentamos un desarrollo de visualización de volúmenes aplicado a un problema de representación de datos en ciencias geológicas. En el mismo se busca la representación de compuestos mineralógicos, de modo de brindar una herramienta intuitiva que brinde ayuda en el proceso de clasificar muestras con respecto a patrones dados. Los patrones de minerales son representados como un sólido traslúcido en un espacio prismático de compuestos [22]. Para ello fue necesario aproximar dichos datos por medio de superficies algebraicas utilizando los métodos tradicionales de la computación gráfica [2, 42]. La representación de sólidos traslúcidos a partir de dicha superficie se llevó a cabo utilizando técnicas de transparencias y α -buffer [4, 5, 14].

Los datos de una muestra particular se podrían representar en el espacio prismático por medio de un ícono adecuado, y la clasificación consistiría simplemente en determinar si dicho ícono yace dentro o fuera del sólido que representa el patrón o tendencia de un mineral determinado. Las muestras constituyen normalmente un conjunto numeroso y eventualmente creciente, por lo cual es necesario buscar una alternativa adecuada para poder clasificar un grupo de muestras en conjunto. Por ejemplo, es posible plantear que dichas muestras constituyen otro sólido virtual dentro del espacio prismático. Para representar el conjunto de muestras, entonces, se utiliza la misma técnica que para la representación del "sólido patrón", es decir, superficies algebraicas. En este caso, como la topología de los datos no puede suponerse rectangular, dado que éstos pueden estar dispuestos de una manera no predecible, se optó por aproximar la superficie por medio de dominios triangulares [12, 20].

El trabajo está organizado de la siguiente manera: en la siguiente sección se plantean los detalles específicos de la aplicación. En la sección 3 se presenta un resumen de las principales técnicas de representación de datos volumétricos que se han desarrollado actualmente, se discuten sus ventajas y desventajas en función del problema a resolver. Luego se presentan algunos detalles de la representación de volúmenes por medio de superficies algebraicas, especialmente de B-splines triangulares. En la sección 5 se combinan los resultados de las secciones anteriores, y se presentan los resultados más importantes del trabajo. En particular, se desarrollan técnicas sencillas para representar sólidos a partir de una descripción de su superficie y se discuten algunos aspectos formales del modelo de iluminación planteado, así como su representación dentro de los esquemas cromáticos usuales. En la sección 6, por último, se discuten las conclusiones y se plantean problemas y mejoras para trabajos futuros.

2 Descripción del problema de aplicación

Un problema existente en mineralogía es la representación de compuestos mineralógicos, de modo de brindar una herramienta intuitiva que auxilie el proceso de clasificar muestras con respecto a patrones dados. Los patrones de minerales son representados en un espacio prismático de compuestos, en este caso óxidos minerales [22] (ver figura 1). Un conjunto "histórico" de muestras de un determinado mineral conforma una "tendencia", la cual puede representarse como un sólido, considerado como un "patrón" frente al cual se contrastan muestras particulares (ver figura 2).

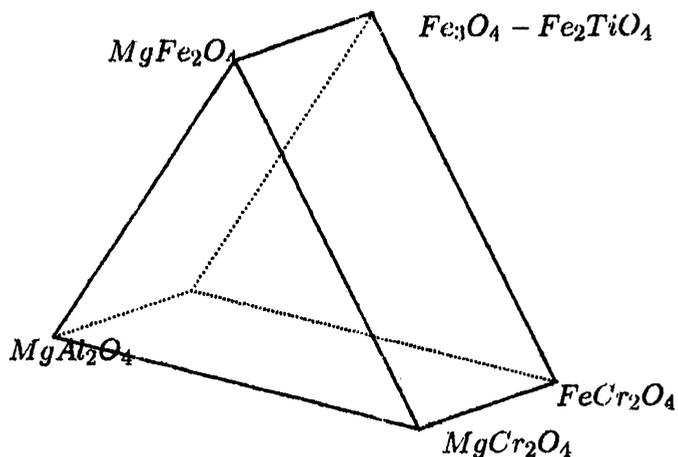


Figura 1: Espacio prismático de posibles compuestos para una variedad de rocas [22].

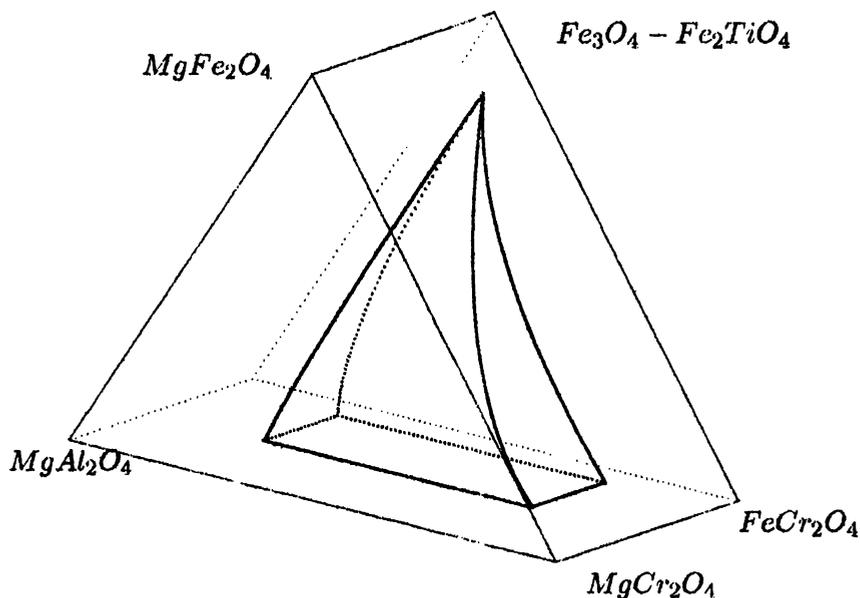


Figura 2: Tendencias establecidas para determinadas rocas de una variedad de óxidos minerales (en este caso basaltos [22]).

Los datos de una muestra particular se podrían representar en el espacio prismático por medio de un ícono adecuado, y la clasificación consistiría simplemente en determinar si dicho ícono yace dentro o fuera del "sólido patrón". Sin embargo, normalmente las muestras constituyen un conjunto numeroso y eventualmente creciente, por lo cual es necesario buscar una alternativa más adecuada para poder clasificar un grupo de muestras en conjunto. En particular, es posible plantear que dichas muestras constituyen virtualmente otro sólido dentro del espacio prismático.

Una solución para aproximar dicho sólido consistiría en utilizar íconos con una determinada extensión espacial (por ejemplo esferas), efectuando una coalescencia o *clustering* de las mismas.

Este camino, sin embargo, es antiintuitivo para el usuario. La solución propuesta consiste en utilizar la misma técnica que para la representación del sólido patrón, es decir, superficies algebraicas. Como la topología de los datos no puede asumirse rectangular, dado que éstos pueden estar dispuestos de una manera no predecible, se optó por aproximar la superficie por medio de dominios triangulares [12, 20]. Como veremos en las siguientes secciones, una ventaja de representar los sólidos por medio de superficies es que permite computar diversas propiedades (el modelo de iluminación, la intersección entre sólidos, etc.) de un modo adecuado y sin recurrir a una voxelización de los datos.

3 Visualización de volúmenes

Los datos geológicos a representar no tienen una manifestación visible por sí misma. Para encontrar una representación gráfica de los mismos, utilizaremos técnicas de visualización, las cuales tienen como objetivo elaborar una representación intermedia que pueda ser procesada para producir una imagen. Esta representación intermedia es arbitraria, y por lo tanto normalmente se elige para que facilite el proceso intuitivo de la percepción. En nuestro caso, como vimos en la sección anterior, se eligió representar los datos como volúmenes traslúcidos. En esta sección resumiremos las soluciones usuales al problema de efectuar un rendering adecuado de dichos volúmenes.

Las técnicas usuales de rendering de volúmenes están normalmente asociadas a una representación de los mismos en estructuras de celdas volumétricas o *voxels* [37, 46]. De esa manera, un voxel en particular representa el factor de ocupación que el sólido posee en una determinada fracción del espacio tridimensional. En gran parte de las aplicaciones usuales (por ejemplo en medicina, meteorología, etc.) [15, 29, 23] la representación con voxels es el resultado natural de la adquisición o computación misma de los datos. Estos datos son eventualmente preprocesados para extraer y enfatizar adecuadamente las características intuitivamente adecuadas en una determinada aplicación y para un determinado propósito en su visualización (por ejemplo, destacar determinadas áreas en la representación visual de una tomografía) [38, 44]. Una vez que los datos volumétricos están adecuadamente preparados para el rendering, el mismo procede según algoritmos de mayor o menor sofisticación.

Una de las primeras técnicas de rendering de volúmenes [13] consiste en graficar por capas el volumen de datos. Normalmente el volumen de datos se hace coincidir con los ejes del sistema de coordenadas del mundo [30], de modo que el eje z (hacia donde mira el observador) coincida con uno de los ejes del volumen de datos. Planos perpendiculares a dicho eje son entonces procesados de adelante hacia atrás. El procesamiento es sencillo, consistiendo en una proyección paralela de los datos al buffer de pantalla, utilizando alguna técnica de *pseudocoloring* [34, 35, 41] para asociar los valores a representar con colores de una paleta predeterminada (por ejemplo, asociar un determinado color a un determinado tejido). Cada voxel, en función de su valor, tiene a su vez una determinada transparencia, es decir que no es necesariamente opaco, permitiendo que se visualice parcialmente las partes del volumen que se encuentran detrás. La transparencia en cada dirección visual se computa acumulándola en un α -buffer de pantalla [4]. Para emular una proyección tridimensional, los datos de las capas se van desplazando una determinada distancia en x e y a medida que éstas son más distantes en el eje z . Esta técnica es bastante primaria, pero por esa misma razón es implementable directamente con hardware específico. Su mayor limitación consiste en que, al no existir un sólido propiamente dicho en ningún momento del procesamiento, no es posible una representación con realismo [11, 42], por ejemplo, la interacción con iluminantes o con otros objetos.

Otros métodos más sofisticados buscan extraer la *representación* de un objeto tridimensional a partir del volumen de datos. Una de las primeras técnicas [17] consiste en procesar capa por capa al volumen de datos, en función de un determinado valor umbral. De esa manera, es posible identificar en una capa dada aquellos voxels en los cuales ocurre una transición cercana al valor umbral. Dichos voxels conforman un contorno. Entre dos capas adyacentes,

entonces, es posible vincular los contornos para determinar un esqueleto de polígonos. El conjunto de polígonos encontrado entre todas las capas procesadas de esta manera constituye una representación del sólido con una estructura "intermedia", en este caso, una superficie. Esta estructura de polígonos permite la visualización de los datos originarios, y tiene la ventaja de ser una estructura "tradicional" en el sentido de la computación gráfica, es decir, es un conjunto de polígonos, el cual puede graficarse con los algoritmos usuales, utilizando cara oculta, sombreado, iluminación, etc. [10, 18]. Sin embargo, esta técnica encuentra problemas cuando no es directo encontrar el esqueleto de polígonos entre dos capas sucesivas (por ejemplo si ocurren discontinuidades topológicas).

Otra solución, más estable con respecto a este tipo de problemas, es la denominada "marching cubes" [27], en la cual se clasifican los voxels que pertenecen a una superficie umbral. Un voxel pertenece a la superficie umbral si por lo menos uno de sus vértices está por debajo del valor umbral y por lo menos otro está por encima. En este caso, cada uno de los ocho vértices de un voxel puede asumir un valor por debajo o por encima del umbral. El total de todos los casos posibles es $2^8 = 256$, pero por consideraciones de simetría se reducen a solo 14. Para cada uno de dichos casos es posible aproximar la superficie umbral con polígonos sencillos (normalmente triángulos) que cortan al voxel, y al mismo tiempo ubicar los voxels vecinos en los cuales dicha superficie debe continuar.

Una solución más completa (y también más compleja) para el rendering de volúmenes consiste en arrojar rayos desde el observador hacia el sólido, de una manera similar al ray tracing [19, 43] pero computando el comportamiento de la luz a través del volumen. Esta técnica, denominada ray casting, comienza por considerar que cada voxel es el modelo de un objeto físico, en el cual ocurre un fenómeno de interacción con la luz y con los rayos visuales provenientes de los demás voxels. Por lo tanto, es necesario establecer un modelo de iluminación que, a diferencia de los modelos tradicionales en computación gráfica como el de Phong [32], considere la interacción de la luz con una densidad volumétrica. Estos modelos fueron estudiados por Blinn [3] y por Kajiya [24, 25], llegando ambos a una formulación matemática similar.

Dada una densidad volumétrica $D(x, y, z)$ y un rayo visual \vec{v} que la atraviesa desde t_1 a t_2 (ver figura 3), consideraremos por un lado la iluminación acumulada en su interacción con una distribución de energía luminosa I que representa una determinada condición de iluminación, y por otro lado la densidad acumulada por el rayo desde que ingresa al sólido en t_1 . Sea entonces un punto t entre t_1 y t_2 . La iluminación que recibe dicho punto es la sumatoria de las intensidades de las fuentes luminosas puntuales.

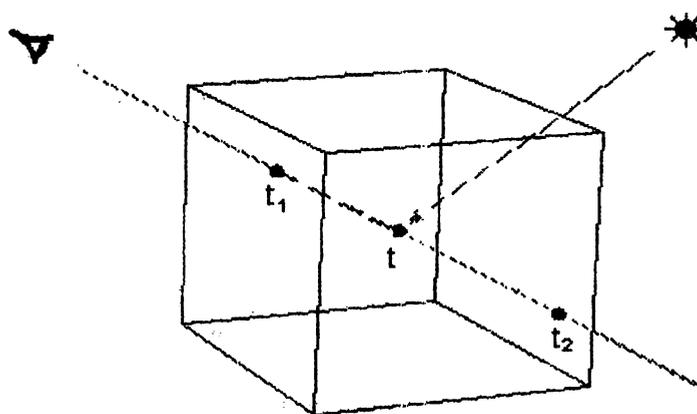


Figura 3: Modelo de iluminación de Blinn-Kajiya.

El modelo considera que la densidad volumétrica puede pensarse como una distribución gaussiana de partículas idealmente especulares. Una partícula en t , entonces, reflejará luz hacia el observador según

$$I = \sum_{i=1}^n I_i \cdot P \cdot D(t) \cdot \cos\theta_i + a,$$

donde cada I_i representa la intensidad luminosa del i -ésimo iluminante puntual, a es un término ambiente, θ_i es el ángulo entre la dirección que va de I_i a t y la dirección de t al observador, P es una función de reflexión, y $D(t)$ el valor de la densidad en t . Este valor es atenuado por la densidad misma del volumen desde t hasta t_1 . La función de atenuación que se deriva del modelo Blinn-Kajiya es una exponencial de la densidad acumulada desde t hasta t_1 , es decir

$$A = e^{-r \int_{t_1}^t D(s) ds}$$

donde r es una constante dimensional (ver figura 3). De esa manera, la energía luminosa reflejada por una partícula en t será igual a $L.A$, y la energía luminosa total que llega al observador al mirar en dirección a t , atravesando el voxel desde t_1 a t_2 es

$$E = \int_{t_1}^{t_2} L.A.dt = \int_{t_1}^{t_2} \sum_{i=1}^n (e^{-r \int_{t_1}^t D(s) ds}) I_i.P.D(t). \cos \theta_i + a dt.$$

Los algoritmos basados en esta técnica, normalmente simplifican esta ecuación según ciertas consideraciones. Por ejemplo, si los rayos visuales son paraxiales, es decir, con pequeña desviación angular respecto del eje z , entonces la distancia de t_1 a t_2 es constante en todos los voxels, y por lo tanto la integral puede aproximarse con una productoria.

En este trabajo se utilizará una simplificación ulterior a este modelo, en el cual se computa la luz que va hacia el observador en función del espesor del sólido. Esto determina que la expresión de la ecuación de iluminación que aproxima la integral de Blinn-Kajiya con una productoria, asume la forma de una exponencial. Los detalles de este modelo, así como su implementación, serán discutidos en la sección 5.

4 Representación de volúmenes con B-splines

Regresando ahora a nuestro problema de aplicación, podemos hacer notar que ninguna de las técnicas mencionadas para rendering de volúmenes es estrictamente aplicable. Esto se debe a la naturaleza misma de los datos a representar. Dado que el conjunto de muestras puede ser pequeño y disperso, su representación directa como un volumen de datos es muy mala. Por dicha razón se pensó en construir un objeto sólido a partir de los datos, tanto del sólido patrón como de las muestras. Esto tampoco se puede realizar utilizando las técnicas asociadas a un valor umbral discutidas más arriba (esqueleto de polígonos o marching cubes). Por dicha razón se consideró obtener una representación de la superficie a partir de puntos de control, específicamente, la construcción de superficies por medio de B-splines [2, 12]. Los B-splines son utilizados para la representación de curvas y superficies, pero en nuestro caso fueron adaptados para la representación implícita de volúmenes. Los puntos de control que determinan la forma de las superficies B-splines se obtienen a partir de los datos a representar.

Para el sólido patrón fue posible la aproximación de su superficie con B-splines rectangulares convencionales (ver figura 4). En los B-splines rectangulares, la superficie se aproxima como producto tensorial de curvas, las cuales están determinadas al recorrer los puntos de control en direcciones paramétricas ortogonales. Dado un conjunto $P_{i,j}$ de puntos de control, la representación biparamétrica de la superficie S de grado k queda determinada por

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^m P_{i,j} N_{i,k}(u) N_{j,k}(v),$$

donde los polinomios B-spline de orden k se definen recursivamente como

$$N_{i,k}(u) = \frac{(u - t_i) N_{i,k-1}(u)}{t_{i+k-1} - t_i} + \frac{(t_{i+k} - u) N_{i+1,k-1}(u)}{t_{i+k} - t_{i+1}},$$

para una secuencia de nudos t_0, \dots, t_n en el espacio paramétrico. La recursión termina con $N_{i,1}(u)$, el cual es igual a 1 si $t_i \leq u < t_{i+1}$, y es igual a 0 en todo otro caso. Ésto determina que los puntos de control deben conformar una estructura topológicamente rectangular, la cual se deforma geoméricamente en función de la ubicación de los puntos de control [9].

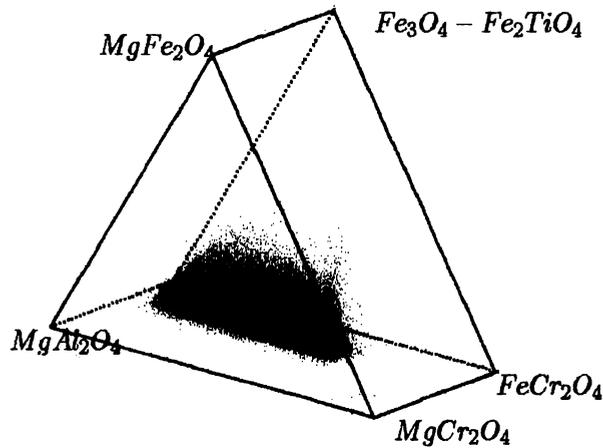


Figura 4: Representación de un patrón de muestras como un sólido en el espacio de muestras.

Como el sólido patrón es una figura establecida, entonces es posible aproximarlos con un B-spline rectangular único y definitivo. Ésto no es posible, sin embargo, para el conjunto de muestras, dado que no puede asumirse una topología rectangular subyacente a la organización del mismo. En una aplicación como la estudiada en este trabajo los datos a representar están dispersos en forma arbitraria, por lo cual la única organización subyacente consiste en una triangulación arbitraria. Por lo tanto es prácticamente imposible encontrar una correspondencia entre los datos y algún conjunto de puntos de control rectangulares. Por dicha razón se utilizaron B-splines en dominios triangulares [12, 20, 40]. En el desarrollo de superficies sobre dominios triangulares se utiliza la descripción geométrica en términos de coordenadas baricéntricas.

Un multiíndice $\vec{\beta} = (\beta_1, \beta_2, \dots, \beta_{k+1})$ es una tupla de $k + 1$ enteros no negativos. La norma $|\vec{\beta}|$ de un multiíndice es $|\vec{\beta}| = \sum_{j=1}^{k+1} \beta_j$. Consideremos un vector de $k + 1$ variables $\mathbf{u} = (u_1, u_2, \dots, u_{k+1})$, tales que $u_1 + u_2 + \dots + u_{k+1} = 1$. Consideremos un triángulo en un espacio paramétrico \mathbf{R}^2 con vértices $\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3$, y un cuarto punto \mathbf{t} . La representación baricéntrica de \mathbf{t} es

$$\mathbf{t} = u_1 \mathbf{t}_1 + u_2 \mathbf{t}_2 + u_3 \mathbf{t}_3,$$

donde $\mathbf{u} = (u_1, u_2, u_3)$ son las coordenadas baricéntricas de \mathbf{t} . Si las coordenadas baricéntricas son todas positivas, entonces \mathbf{t} , cae dentro del plano determinado por $\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3$. Dado entonces un espacio paramétrico \mathbf{R}^2 en el que se ha determinado un conjunto de nudos \mathbf{t}_i , una triangulación de \mathbf{R}^2 es un conjunto $T = \{\Delta(I) = [\mathbf{t}_{i_0}, \mathbf{t}_{i_1}, \mathbf{t}_{i_2}] | I = (i_0, i_1, i_2)\}$. Supongamos que a cada \mathbf{t}_i se le asigna una secuencia de nudos $\mathbf{t}_{i_0}, \mathbf{t}_{i_2}, \dots, \mathbf{t}_{i_n}$ de modo que $\mathbf{t}_{i_0} = \mathbf{t}_i$ y que tres nudos cualesquiera conforman un triángulo apropiado. Dado un grado n para nuestra superficie, consideremos los multiíndices $|\vec{\beta}| = n$. Dado un triángulo I y un multiíndice $\vec{\beta}$, consideremos los conjuntos

$$V_{\vec{\beta}}^I = \{\mathbf{t}_{i_0,0}, \dots, \mathbf{t}_{i_0,\beta_0}, \dots, \mathbf{t}_{i_2,0}, \mathbf{t}_{i_2,\beta_2}\}.$$

En dicho caso se define el B-spline triangular para el multiparámetro \mathbf{u} como $M_{\vec{\beta}}^I(\mathbf{u}) = M(\mathbf{u} | V_{\vec{\beta}}^I)$ en forma recursiva:

$$M(\mathbf{u} | \mathbf{v}_0, \dots, \mathbf{v}_m) = \sum_{j=0}^2 \lambda_j(\mathbf{u}) M(\mathbf{u} | \mathbf{v}_0, \dots, \hat{\mathbf{v}}_j, \dots, \mathbf{v}_m).$$

En esta última ecuación, la notación $\hat{\mathbf{v}}_j$ indica que el nudo \mathbf{v}_j se elimina de la secuencia de nudos $\mathbf{v}_0, \dots, \mathbf{v}_m$, y $\lambda_j(\mathbf{u})$ representa la j -ésima coordenada baricéntrica de \mathbf{u} con respecto al triángulo $\mathbf{v}_{i_0}, \mathbf{v}_{i_1}, \mathbf{v}_{i_2}$. Es decir, la recursión consiste en elegir tres nudos en $\mathbf{v}_0, \dots, \mathbf{v}_m$ tales que conformen un triángulo apropiado, y efectuar una sumatoria de los tres splines sobre las secuencias reducidas, cada uno ponderado por su factor $\lambda_j(\mathbf{u})$. La recursión termina cuando la secuencia $\mathbf{v}_0, \dots, \mathbf{v}_m$ tiene solo tres nudos. En este caso, sea $d(\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2)$ igual a dos veces el área determinada por el triángulo $\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2$. Entonces $M(\mathbf{u}|\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2) = \frac{1}{d(\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2)}$ si \mathbf{u} es un punto interior al triángulo que conforman, y 0 en todo otro caso. Por último, se define el B-spline normalizado $N_{\vec{\beta}}^I(\mathbf{u})$ para asegurar invariancia afin:

$$N_{\vec{\beta}}^I(\mathbf{u}) = d(\mathbf{t}_{i_0, \beta_0}, \mathbf{t}_{i_1, \beta_1}, \mathbf{t}_{i_2, \beta_2}) M_{\vec{\beta}}^I(\mathbf{u}).$$

Los B-splines triangulares normalizados $N_{\vec{\beta}}^I(\mathbf{u})$ son funciones de mezcla (como en los B-splines convencionales) y por lo tanto se demuestra que

$$\sum_I \sum_{|\vec{\beta}|=n} N_{\vec{\beta}}^I(\mathbf{u}) = 1.$$

Por último, dado un conjunto de puntos de control (ver figura 5) $\mathbf{P}_{\vec{\beta}, I}$, la superficie B-spline triangular de grado n sobre la triangulación T es

$$S(\mathbf{u}) = \sum_I \sum_{|\vec{\beta}|=n} N_{\vec{\beta}}^I(\mathbf{u}) \mathbf{P}_{\vec{\beta}, I}.$$

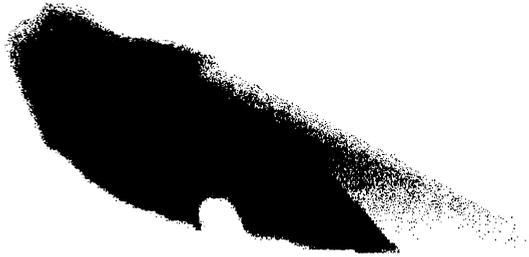


Figura 5: Un conjunto de muestras representado por medio de B-splines triangulares.

5 Detalles de la implementación

Una de las principales motivaciones para esta aplicación proviene de la posibilidad de producir visualizaciones interactivas, es decir, que el usuario pueda interactuar con el sistema de visualización y que éste responda en tiempos relativamente pequeños. Por dicha razón fueron necesarias las técnicas de rendering acelerado que se presentan a continuación. Como quedó expresado en la sección anterior, la representación de los volúmenes se realizó por medio de B-splines, los cuales fueron evaluados para determinados valores del espacio paramétrico, produciendo una aproximación a la superficie por medio de un conjunto de triángulos. Esta aproximación es arbitraria, pudiendo elevarse la cantidad de muestras (y por consiguiente la cantidad de triángulos) si mayor precisión fuese necesaria.

Luego, debe integrarse la iluminación a lo largo de los caminos ópticos que concurren al punto de vista (ver figura 6a). Una forma eficiente de realizar este cómputo consiste en efectuar primero la transformación perspectiva. De esa manera es posible computar en el espacio imagen dichos caminos ópticos, ya que se transforman en paralelos al plano de proyección. Considerando que los sólidos traslúcidos a representar son homogéneos y que la fuente luminosa se encuentra

aceptablemente lejos y radia una iluminación unitaria, es posible factorizar la densidad D y la atenuación A dentro de la integral de Blinn-Kajiya. Suponiendo ahora (sin pérdida de generalidad) que los voxels tienen tamaño unitario, y al ser recorridos en forma perpendicular (por haberse aplicado previamente la transformación perspectiva), obtenemos para un voxel unitario:

$$\begin{aligned}
 E &= \int_{t_1}^{t_2} (e^{-r \int_{t_1}^t D(s).ds}).I.P.D(t).cos\theta dt \\
 &= \int_0^1 (e^{-r \int_{t_1}^t D(s).ds}).I.P.D(t).cos\theta dt \\
 &= \int_0^1 (e^{-r \int_0^1 D(s).ds}).I.P.D(t).cos\theta dt \\
 &= \int_0^1 (e^{-rD}).D(t).dt \\
 &= De^{-rD}.
 \end{aligned}$$

Un camino óptico que parte del observador y atraviesa el sólido, entonces, acumula una energía luminosa y una transparencia computada en función del espesor t del sólido a lo largo de dicho camino:

$$E = D(1 - (1 - e^{-rD})^t).$$

Teniendo en cuenta que este cómputo se realiza luego de haber transformado los triángulos según la transformación perspectiva, el espesor t es simplemente la distancia en el eje z (perpendicular al plano de proyección) entre el triángulo más cercano y el más lejano a dicho plano (ver figura 6b).

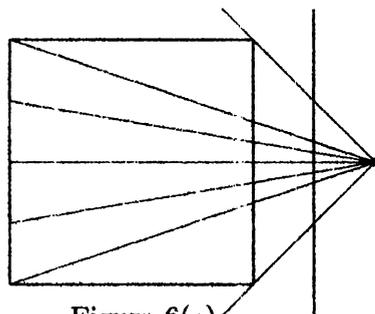


Figura 6(a)

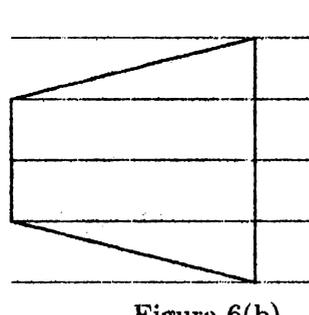


Figura 6(b)

El cómputo del espesor t se realiza durante la conversión-scan de los triángulos que conforman el sólido. Para ello se recorre el espacio imagen a x creciente, manteniendo una lista de todos los triángulos "activos" (que tienen por lo menos un pixel para dicho x) y realizando la conversión para todos ellos simultáneamente (ver figura 7). De esa manera, durante la conversión se va computando el z correspondiente a cada pixel para cada triángulo, y por consiguiente la diferencia entre los z de dos triángulos para un pixel dado es el espesor t buscado.

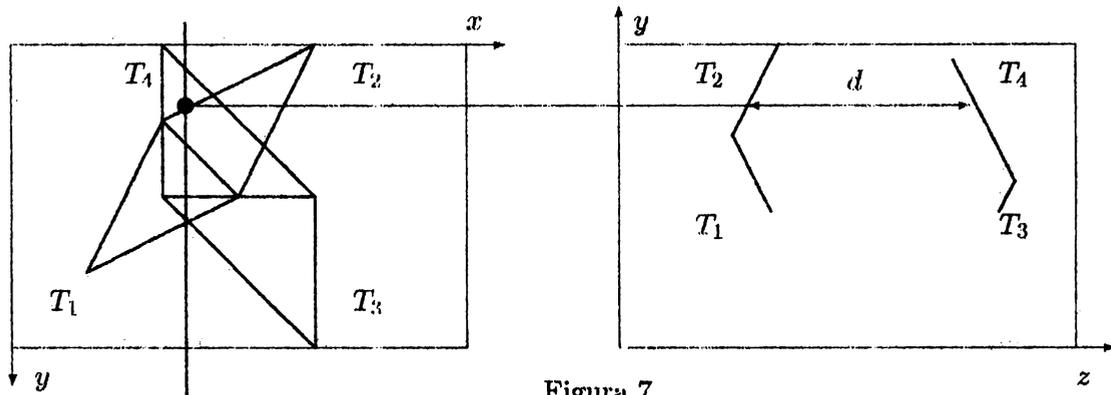


Figura 7

Cuando en el rayo óptico es posible encontrar dos pares de triángulos, ésto se debe a que a lo largo de dicho rayo es posible encontrar o bien dos veces al mismo sólido (por tener éste convexidades), o bien a otro sólido. En ese caso, la ecuación de iluminación sin modificar produce un resultado visualmente realista (ver figura 8a), pero que no es necesariamente útil a la hora de resaltar la relación entre ambos volúmenes de datos. Para destacar la relación entre distintos sólidos, en particular la intersección de los mismos, es posible modificar la ecuación de iluminación. En nuestro caso, dado que los sólidos representan espacios de muestras, se desarrollaron modificaciones al modelo de iluminación de modo tal de poder destacar la intersección de ambos sólidos (figura 8b).



Figura 8

Las posibilidades de modificación de la ecuación de iluminación pueden verse en el siguiente grupo de figuras, donde un sólido de muestras sencillo atraviesa al sólido patrón. En la figura 9a se observa el resultado de efectuar el rendering con la ecuación de iluminación sin modificar. En la figura 9b se da precedencia al sólido patrón y en la 9c al sólido de muestras. Por último, en la 9d se destaca el sólido intersección modificando artificialmente su color.

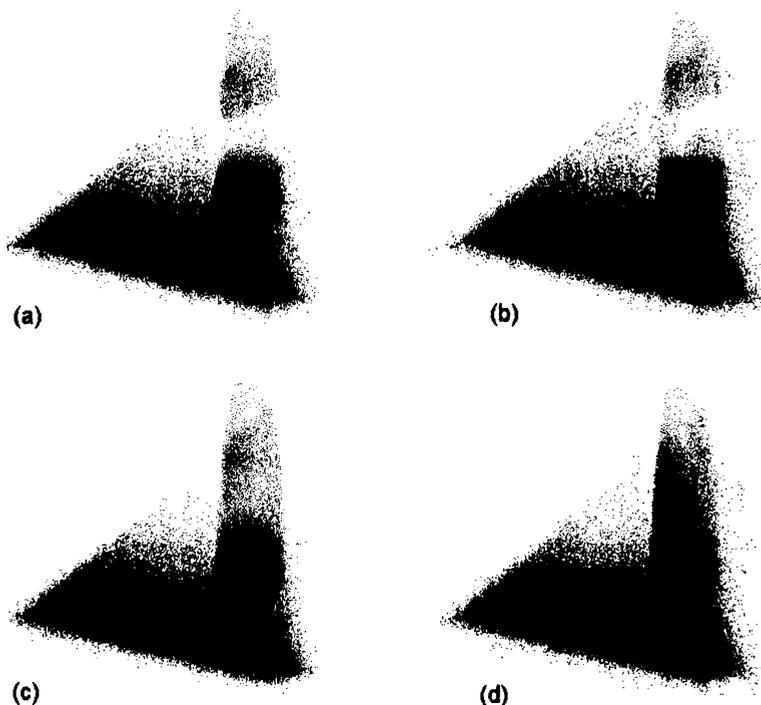


Figura 9

Es importante destacar que este mecanismo permite transformar la representación de sólidos con superficies en una representación con volúmenes de datos, es decir, es posible voxelizar los

sólidos. La forma más directa de hacerlo consiste en asignar una dimensión arbitraria a los voxels, relacionando la dimensión en x e y de los voxels con la dimensión en x e y de los pixels durante la conversión-scan. De esa manera, la matriz tridimensional de datos donde se almacena la voxelización se carga del siguiente modo. Durante la conversión-scan de los triángulos activos se determina el x e y de la pantalla. Ésto determina las direcciones en x e y en la matriz de datos, pudiendo corresponder a una única fila (si la dimensión de la matriz en x e y es menor o igual que la resolución del espacio imagen) o a más de una (si sucede a la inversa). Para dicha fila y para un sólido dado, las celdas cuyo z está comprendido entre el triángulo del sólido más cercano y el más lejano al plano de proyección acumulan el dato de que dicho sólido está presente. De esa forma es posible voxelizar los sólidos como un subproducto del algoritmo de rendering presentado. Esta voxelización permite la visualización de los volúmenes de datos resultantes utilizando las técnicas convencionales reseñadas más arriba.

6 Conclusiones y trabajo futuro

Se presentó un desarrollo de visualización de volúmenes aplicado a un problema de representación de datos en ciencias geológicas. Como la naturaleza de los datos a representar no permite la aplicación directa de las técnicas usuales de rendering de volúmenes, los mismos fueron aproximados por medio de superficies algebraicas utilizando B-splines convencionales y B-splines triangulares. La representación de sólidos traslúcidos a partir de dicha superficie se llevó a cabo utilizando técnicas de transparencias derivadas de una simplificación adecuada del modelo de Blinn-Kajiya. Esta simplificación posibilita la modificación del resultado visual de los volúmenes para mejorar la comprensión intuitiva por parte del usuario, pudiendo destacarse un sólido con respecto al otro, o el resultado de la intersección entre los sólidos. El algoritmo de rendering utilizado es muy eficaz, por lo que permite la visualización en tiempos interactivos. Al mismo tiempo, este algoritmo permite generar una voxelización de los datos como subproducto.

Entre las posibles mejoras que se pueden realizar, la más importante consiste en comparar los resultados visuales obtenidos con el algoritmo presentado con los obtenidos al visualizar los datos voxelizados con alguna técnica de rendering de volúmenes (por ejemplo ray casting). Ésto tiene la ventaja de la generalidad, dado que el ray casting necesita como entrada un volumen de datos, mientras que el algoritmo presentado requiere una representación por medio de triángulos, almacenada en estructuras de datos determinadas (lo cual no es flexible). Otra mejora consiste en agregar reflexión especular en la superficie de los sólidos, lo cual brinda una mejor comprensión de la forma y textura de los mismos. Para ello es necesario agregar al modelo de iluminación un término similar al de la ecuación de Phong, lo cual se puede complicar a la hora de interpolar los normales de los triángulos.

Referencias

- [1] R. S. Avila, I. M. Sobierajski, and A. E. Kaufman. Towards a Comprehensive Volume Visualization System. In *Visualization '92 Proceedings*, pages 13-20, Los Alamitos, CA, 1992. IEEE Technical Committee on Computer Graphics, IEEE Computer Society Press.
- [2] R. Bartels, J. Beatty, and B. Barsky. *An Introduction to Splines for Use in Computer Graphics and Geometric Modelling*. Springer-Verlag, New York, 1987.
- [3] James F. Blinn. Light Reflection Function for Simulation of Clouds and Dusty Surfaces. *ACM Computer Graphics*, 16(3):21-29, 1982.
- [4] James F. Blinn. Compositing I-Theory. *IEEE Comp Graphics and Applications*, 14(5):83-87, 1994.
- [5] James F. Blinn. Compositing II-Practice. *IEEE Comp Graphics and Applications*, 14(6):78-82, 1994.

- [6] J. Chalinger. Scalable Parallel Volume Ray Casting for Nonrectilinear Computational Grids. In *1993 Parallel Rendering Symposium Proceedings*, pages 81–88, Los Alamitos, CA, 1993, ACM Press.
- [7] S. Cunningham, J. R. Brown, and M. McGrath. Visualization in Science and Engineering Education. In G. M. Nielson and B. D. Shriver, editors, *Visualization in Scientific Computing*, pages 48–58. IEEE Computer Society Press, Los Alamitos, CA, 1990.
- [8] T. A. Defanti, M. D. Brown, and B. H. McCormick. Visualization: Expanding Scientific and Engineering Research Opportunities. In G. M. Nielson and B. D. Shriver, editors, *Visualization in Scientific Computing*, pages 32–47. IEEE Computer Society Press, Los Alamitos, CA, 1990.
- [9] Claudio Delrieux and Daniel Formica. Caracterización y Evaluación de Algoritmos de Aproximación de Curvas y Superficies. *Revista Telegráfica Electrónica*, 904, 1989.
- [10] Claudio Delrieux, Daniel Formica, Fernando Caba, and Esteban Pedroncini. Una Solución Eficiente al Problema de la Cara Oculta. *Revista Telegráfica Electrónica*, 934, 1991.
- [11] R. Drebin, L. Carpenter, and P. Hanrahan. Volume Rendering. *ACM Comp Graphics*, 21(4):65–74, 1988.
- [12] Gerald Farin. *Curves and Surfaces for Computer Aided Geometric Design*. Ac Press, NY, 1988.
- [13] J. Farrell. Colour Display and Interactive Interpretation of Three-Dimensional Data. *IBM Journal of Research and Development*, 27(4):356–366, 1983.
- [14] J. Foley, A. Van Dam, S. Feiner, and J. Hughes. *Computer Graphics. Principles and Practice*. Addison-Wesley, Reading, Massachusetts, second edition, 1990.
- [15] T. Foley, D. Lane, G. Nielson, and R. Ramaraj. Visualizing Functions over a Sphere. *IEEE Computer Graphics and Applications*, 10(1):32–40, 1990.
- [16] H. Fuchs, M. Levoy, and J. K. Lam. Interactive Visualization of 3D Medical Data. In G. M. Nielson and B. D. Shriver, editors, *Visualization in Scientific Computing*, pages 140–146. IEEE Computer Society Press, Los Alamitos, CA, 1990.
- [17] S. Ganapathy and T. Dennehy. A New General Triangulation Method for Planar Contours. *ACM Computer Graphics*, 16(3):69–75, 1983.
- [18] W. K. Giloi. *Interactive Computer Graphics - Data Structures, Algorithms, Languages*. Prentice-Hall, Englewood Cliffs, New Jersey, 1978.
- [19] A. Glassner. *An Introduction to Ray Tracing*. Academic Press, Cambridge, Massachusetts, 1991.
- [20] G. Greiner and H. Seidel. Modelling with Triangular B-Splines. *IEEE Computer Graphics and Applications*, 14(2):56–60, 1994.
- [21] R. B. Haber and D. A. McNabb. Visualization Idioms: A Conceptual Model for Scientific Visualization Systems. In G. M. Nielson and B. D. Shriver, editors, *Visualization in Scientific Computing*, pages 74–93. IEEE Computer Society Press, Los Alamitos, CA, 1990.
- [22] Stephen E. Haggerty. Opaque Mineral Oxides in Terrestrial Igneous Rocks. In Douglas Rumble III, editor, *Oxide Minerals*, pages 101–295. Mineralogical Society of America, Washington, DC, 1976.
- [23] W. Hibbard and D. Santek. Visualizing Large Meteorological Data. In G. M. Nielson and B. D. Shriver, editors, *Visualization in Scientific Computing*, pages 147–152. IEEE Computer Society Press, Los Alamitos, CA, 1990.
- [24] James T. Kajiya. The Rendering Equation. *ACM Computer Graphics*, 20(4):143–150, 1986.
- [25] James T. Kajiya and B. Von Herzen. Ray Tracing Volume Densities. *ACM Computer Graphics*, 18(4):91–102, 1984.
- [26] Peter Keller and Mary Keller. *Visual Cues: Practical Data Visualization*. IEEE Computer Society Press, Los Vaqueritos, CA, 1990.

- [27] W. Lorensen and H. Cline. A High Resolution 3D Surface Construction Algorithm. *ACM Computer Graphics*, 21(4):163-169, 1987.
- [28] B. Lucas. A Scientific Visualization Renderer. In *Visualization '92 Proceedings*, pages 227-234, Los Alamitos, CA, 1992, IEEE Computer Society Press.
- [29] T. Mihalasin, J. Timlin, and J. Schwegler. Visualizing Multivariate Functions, Data, and Distributions. *IEEE Computer Graphics and Applications*, 11(3):28-35, 1991.
- [30] W. Newman and R. Sproull. *Principles of Interactive Computer Graphics*. McGraw-Hill, NY, 1973.
- [31] G. M. Nielson and B. D. Shriver. *Visualization in Scientific Computing*. IEEE Computer Society Press, Los Alamitos, CA, 1990.
- [32] Bui-Tong Phong. Illumination for Computer-Generated Pictures. *Communications of the ACM*, 18(6):311-317, 1975.
- [33] V. Raujan and A. Fournier. Volume Models for Volumetric Data. *IEEE Comp*, 27(7):28-36, 1994.
- [34] P. Rheingans. Color, Change and Control for Quantitative Data Display. In *Visualization '92 Proceedings*, pages 252-259, Los Alamitos, CA, 1992. IEEE Technical Committee on Computer Graphics, IEEE Computer Society Press.
- [35] P. K. Robertson. Visualizing Color Gamuts: a User Interface for the Effective Use of Perceptual Color Spaces in Data Displays. *IEEE Computer Graphics and Applications*, 8(5):50-64, 1988.
- [36] L. Rosenblum. Scientific Visualization at Research Laboratories. *IEEE Comp*, 22(8):68-100, 1989.
- [37] G. Sakas and J. Hartig. Interactive Visualization of Large Scalar Voxel Fields. In *Visualization '92 Proceedings*, pages 29-38, Los Alamitos, CA, 1992. IEEE Technical Committee on Computer Graphics, IEEE Computer Society Press.
- [38] R. Santancy, D. Silver, N. Sabusky, and J. Cao. Visualizing Features and Tracking Their Evolution. *IEEE Computer*, 27(7):20-27, 1994.
- [39] C. Upson and M. Keeler. V-Buffer: Visible Volume Rendering. *ACM Comp Graphics*, 22(4):59-64, 1988.
- [40] Sebastian Urbicain and Claudio Delrieux. Modelos de Superficies con Topologías Arbitrarias. In *INFOCOM '96*, pages 462-471, Buenos Aires, Argentina, 1996. Congreso Internacional de Informática y Telecomunicaciones.
- [41] Colin Ware. Color Sequences for Univariate Maps: Theory, Experiments and Principles. *IEEE Computer Graphics and Applications*, 8(5):41-49, 1988.
- [42] Alan Watt and Mark Watt. *Advanced Animation and Rendering Techniques*. Addison-Wesley, London, 1992.
- [43] T. Whitted. An Improved Illumination Model for Shaded Displays. *Communications of the ACM*, 23(6):343-349, 1980.
- [44] C. Williams, J. Rasure, and C. Hansen. State of the Art of Visual Languages for Visualization. In *Visualization '92 Proceedings*, pages 202-209, Los Alamitos, CA, 1992. IEEE Technical Committee on Computer Graphics, IEEE Computer Society Press.
- [45] R. D. Williams, F. L. Wefer, and T. E. Clifton. Direct Volumetric Visualization. In *Visualization '92 Proceedings*, pages 99-106, Los Alamitos, CA, 1992. IEEE Technical Committee on Computer Graphics, IEEE Computer Society Press.
- [46] J. R. Wright and J. C. Hsieh. A Voxel-based, Forward-Projection Algorithm for Rendering Surface and Volumetric Data. In *Visualization '92 Proceedings*, pages 340-348, Los Alamitos, CA, 1992. IEEE Technical Committee on Computer Graphics, IEEE Computer Society Press.
- [47] T. S. Yoo, U. Neumann, H. Fuchs, and S. Pizer. Direct Visualization of Volume Data. *IEEE Computer Graphics and Applications*, 12(4):64-71, 1992.