Visualización de modelos de sólidos basados en árboles octales ("Octrees")

RESUMEN
Un enfoque para el modelado de sólidos basado en la descomposición del espacio objeto en subespacios regulares, particulaman-
te cuadrados, que muestra en una escala reducida de árbol, es el "octree" (o árbol oculto).
Un objeto cualquiera (Universo) se descompone en 8 celdas, del mis-
tro tamaño para cada uno de los cuales se verifica si son o no homogé-
neas, es decir, si están compuesta
dentro o completamente fuera del objeto. Si esto ocurre, la des-
composición termina; en caso con-
trario se hacen de manera recí-
ticleta hacia atrás y hacia delante (en inglés: front-to-back y back-to-front, respectivamente).
La tarea de visualizar un "octree" puede hacerse de dos maneras básicas: frente-

to-hacia atrás y hacia delante (en inglés: front-to-back y back-to-front, respectivamente).

ABSTRACT
A geometrical modeling technique called Octree Encoding is characterized based on the quadtree technique.
Some display methods are presented including an algo-

Ana Esther Rodríguez
Título de postgrado y pregrado
Licenciada en Informática
Universidad de Sevilla
España

María Elena Millán, Ph.D.
Licenciada en Matemáticas-Física
Profesora Asociada
Departamento de Ciencias de la Computación
Universidad del Valle.
1. INTRODUCCION

Dentro del modelo de sólidos se pueden identificar entre otros, los siguientes tipos de sistemas [14]: los que se basan en el instanciamento de primitivas, que definen un objeto con base en un tipo de forma y en un conjunto de valores de parámetros; los de enumeración espacial, que representan un objeto mediante una lista de las celdas espaciales cubicas que este ocupa; los de descomposición en celdas, generalización del modelo anterior, en el que un objeto se representa por medio de un conjunto de celdas de distinto tamaño que no son necesariamente cubicas, ni en blanco; los basados en geometría constructiva que representan los objetos como conjuntos de sólidos primitivos concatenados y uniones de operaciones booleanas y los de representación de bordes, que presentan los objetos por medio de las superficies que los encierran (planos, superficies cuadráticas, etc.).

En estos sistemas, los objetos se construyen a partir de un número limitado de superficies y están definidas matemáticamente de primitivas y las fases de manipulación y visualización de objetos pueden representar una carga computacional significativa, debido a las primitivas, con las cuales el modelo del objeto, se deben construir las uniones con otras. Esta tarea de computación usualmente requiere búsqueda ordenamiento. El proceso fundamental de estos modelos geométricos basados en árboles octales ("octree"), es ofrecer un esquema de modelo geométrico tridimensional en el que un objeto, independientemente de su complejidad, puede codificar, manipular, analizar y visualizar inmediatamente en tiempo real o muy cercano al tiempo real [14]. El modelo se caracteriza por representar objetos tridimensionales y N-dimensionales, para cualquier resolución, en un formato de codificación común, que permita operaciones booleanas (unión, intersección y diferencia) y geométricas (translación, rotación, escalamiento) sobre cualquier objeto o conjunto de objetos y facilitar la visualización, en tiempo lineal, de cualquier número de objetos desde el punto de vista de un observador.

Esta representación describe un cuerpo sólido como un arreglo tridimensional que contiene información en donde se indica el material que se encuentra en cada punto del espacio [2]. Cuando una gran región del espacio contiene un solo material, entonces muchos elementos adyacentes del arreglo forman el mismo color y por esta razón se puede utilizar un esquema de compresión.

Para Jackins et al. [11], un "octree" es una forma de ordenamiento espacial que permite hacer operaciones más económicas, en términos de memoria. El enfoque es una extensión tridimensional de los métodos "quadtree".

De acuerdo con Yamaguchi et al. [25] y Samet [18][21], un conjunto "octree" CO se define formalmente:

1. $O \in CO$
2. $1 \in CO$
3. $O_{max}, O_{min}, O_{left}, O_{right}, O_{up}, O_{down}, O_{front}, O_{back} \in \{CO, O\}$

Note: $O_{x,y,z} \in CO$ para $x,y,z \in [0,1)$

(4) $CO$ no incluye ningún elemento que no esté contenido aplicando las reglas 1 a 3 repetidamente, 0 y 1 son hojas y los otros son nodos "octree".

Algunas de las ventajas de esta representación [3], [14] son las siguientes:

- Se utiliza únicamente una forma primitiva: el cubo, cualquier objeto se puede representar con precisión del cubo de menor tamaño.
- Para todos los objetos, se utiliza solamente un único conjunto de algoritmos de manipulación y análisis, los objetos representados mediante un "octree" están espacialmente pre-ordenados, las propiedades geométricas (volumen, área, etc.) se pueden calcular fácilmente con diferente nivel de precisión.
- Los nodos de cualquier nivel del árbol, junto con los niveles superiores, describen completamente el objeto a la resolución dada por ese nivel. Así, un algoritmo que trabaja en un nivel dado no requiere manejar los volúmenes de datos que se almacenan en niveles inferiores.
- Los algoritmos de manipulación de operaciones booleanas y geométricas no requieren utilizar ni formatos de punto flotante ni divisiones o multiplicaciones enteras, su implementación es poco costosa y la mayoría de los casos solo utilizan 3 tipos de nodos.

Algunas de las desventajas más importantes [2] son:

- Un gran número de cubos o celdas se requiere para que un objeto, con detalles complejos, se pueda representar adecuadamente.
- Como el "octree" es solamente una aproximación al objeto el cálculo de las propiedades geométricas también es aproximado.
- La estructura no se incorpora fácilmente a los sistemas de "software" y requiere gran cantidad de memoria.

2. ANTECEDENTES

La codificación "octree" es una extensión de la codificación bidimensional quadtree [3]. En la estructura "quadtree" se puede describir homogénea o heterogéneamente. Una codificación "quadtree" se genera cuando la subdivisión de regiones produce cuadrantes de la región original. La estructura de datos "quadtree" es una gráfica dirigida
para el cual cada nodo contiene una lista de cuatro elementos, uno por cada uno de los cuadros en la imagen [3]. Cada elemento de datos tiene una marca que indica si el cuadrante es o no hueco. Cuando la marca indica que el cuadrante es un cuadrante particular es homogéneo, el resto de los elementos de datos toman como valor el del "pixel" que llena la región. En un cuadrante heterogéneo un elemento de datos contiene un puntero a otro nodo que subdivide la región. Un elemento de datos homogéneo de un nodo apunta a un descendiente del nodo. Cada nodo "quadtree" puede además tener 0, 1, 2, 3 o 4 descendientes.

El Elemento Universal es un elemento de datos que describen una subdivisión de la imagen y que tiene nivel 0. El nivel de un nodo en el árbol se determina con el número de subdivisiones anidadas desde este al elemento universal. Los nodos de un "quadtree" tienen asociada una única dirección de nodo, que consiste en una secuencia de punteros que describen el camino desde el elemento universal hasta el nodo. En el árbol de la siguiente (un "quadtree"), la línea gruesa muestra las subdivisiones (en pantalla 0, 1, 1, 0) desde el nodo universal hasta el nodo final. Las regiones basadas en "pixels" con frecuencia se representan como grandes bloques o cuadrantes con trados en una matriz 2x2 (n=1) o "master" [1]. Los bloques se generan subdividiendo recursivamente el formato inicial en el cuadrado de tamaño 2x2. El valor de nivel n = n, n=2, ..., 0. En un gráfico binario, cuando un bloque tiene parte de un objeto oculto se crean dos cuadrantes con nombre de "quadtree". Los nodos de un "quadtree" no tienen hijos o tienen 4. La raíz representa el "master", es el nodo de nivel inferior que representa los "quadtree" [1].

En la representación de un "quadtree" se introdujo la linearización de un bloque con respecto al "master", codificando todos los anexos en una sola clave [6], [8], [9] y un método de representación de "quadtree" que no utiliza cuadros: el "quadtree" linear. Algunas de las características de la estructura son, entre otras, las siguientes:

- solo se almacenan nodos negros,
- en la codificación de los nodos se incorporan propiedades de adyacencia en las direcciones NW, NE, SE, SW, SE, N, NO, OESTE, SUD, SUD-OESTE
- en la representación de los nodos se introducen, impulciones finales, con el fin de que no se alte hasta el nodo y se eliminan cuadros.

2.1 Codificación de nodos negros
El cuadrante NW se codifica con 0, el NE con 1, SW con 2 y SE con 3. Cada "tuple" se codifica con un entero en base 4, donde cada dígito representa la subdivisión del cuadrante desde el cuadrante superior. El orden de la secuencia de los dígitos refleja el orden de subdivisión de mayor a menor.

Los dígitos tienen un peso 4n, n=1,2,3,..., identificando el cuadrante al que pertenece el "pixel" en la h-ésima subdivisión.

Procedimiento de codificación
Para un entero (r, d) en base n, (r, d, r, d) en base 4 que indica los cuadrantes que representan el "pixel" (i, j) [7,8].

- C = C + C + C + ... + C
- d = d + d + d + d + d
- C, d pueden tomar los valores 0, 1, 2, 3, 4 según indican el cuadrante al que pertenece el "pixel" (i, j) en cada nivel de subdivisión.

El procedimiento para calcular k es el siguiente:

CODIFICAR (m, a, d, k)

m: nivel de descomposición
d: r: variable anexada
Para i=0 hasta 2, incremento i=1
Si c di=0 y di+1 entonces k=1
Si c di=0 y di+1 entonces k=2
Si c di=0 y di+1 entonces k=3
Después de codificar cada "pixel" negro, éstos se ordenan y se almacenan en una lista que se puede condensar en aquellos casos en los que cuatro "pixeles" tienen la misma representación, excepto el último dígito. En este caso, ese último dígito se reemplaza por una marca especial.

Para el ejemplo de la figura el "quadtree" lineal corresponde a las (001, 003, 012, 013, 02, 03, 10, 11, 20, 101, 102, 112, 113, 20, 30, 31, 32, 33) y a los octantes que se encuentran dentro de él tienen etiquetas L (VACIO), V, o MEZCLA (M), en cualquier otro caso.

3. OBJETOS Y OCTREES

Codificación "octree"

Este esquema es similar a los enfoques de enumeración espacial y descomposición en celdas [14]. Sin embargo, desde el punto de vista de almacenamiento, los datos se almacenan en una estructura jerárquica de árbol, en la que los nodos representan cubos diminutos decreciendo exponencialmente de tamaño. Cada nodo en el árbol, corresponde a una región del universo y tiene uno o más valores que definen la región. Un nodo es terminal ú ha si su valor describe completamente la región. Si no, el nodo apunta a 8 hijos que representan 8 subregiones u octantes del nodo padre. Una de las ventajas adicionales, a las enumeradas en la introducción, es que las operaciones de visualización de áreas ocultas y la detección de intersecciones son de complejidad lineal porque todos los objetos se mantienen predeterminedos espacialmente [14]. Si se recorre el árbol, en una secuencia adecuada, y las regiones del espacio se visitan en una dirección uniforme, los algoritmos de visualización de áreas ocultas no requieren búsqueda, ni ordenamiento. Los árboles que representan los objetos que se van a visualizar, se recorren simplemente en un orden específico, dependiendo del punto de vista del observador.

En este caso, el concepto de codificación "quadtree" se extiende directamente a estructuras de tres dimensiones (codificación "octree") [9], en el que se utilizan regiones del espacio, en lugar de áreas. Por consiguiente, una región del espacio es homogénea cuando todos estos "velas" (elementos volumen) son del mismo material o heterogénea en el caso contrario, pero en el cual ésta se divide, sobre los ejes primarios para producir octantes o de un mismo color. Los nodos octantes contienen 8 elementos de datos que corresponden a los octantes de una región en el espacio.

Las regiones vacías se representan con "voids" que contienen el valor especial, vacío. El borde de un objeto es un límite entre dos tipos de material que impide que "voids" adyacentes formen cubos de tamaño mayor. Los objetos en consideración son amarillos, luminosos, y azules, el arrojado por un objeto es un límite entre dos tipos de material que impide que "voids" adyacentes formen cubos de tamaño mayor. Un objeto, ordenado, de diámetro 2", se divide en 8 cubos de diámetro 2". Este proceso continúa de manera que en cada paso un arreglo de diámetro 2" se divide en 8 arreglos en forma de cubos de diámetro 2".

21 [1] [12]. La región cubica que se obtiene de esta forma es un OCTANTE. Usando "octree", un objeto se divide en octantes, cada un octante puede identificarse con un valor: LLENO (L) si todos los cubos unitarios que se encuentran dentro de él tienen etiqueta L (VACIO), V, si todos los cubos unitarios que están dentro de él tienen como etiqueta V y MEZCLA (M), en cualquier otro caso.

4. VISUALIZACIÓN DE "OCTREES"

Los algoritmos de visualización transforman una representación "octree" de un objeto en una imagen gráfica. De acuerdo con Carboon [2], los algoritmos de visualización se pueden recorrer un "octree" de atrás-hacia-frente o de frente-hacia-atrás. En esta parte se presentan algunos métodos de visualización. Un primer método, propuesto por Frazier, Gordon y Reynolds [4], permite mostrar objetos sólidos con diferentes colores de "voids", utilizando un algoritmo atrás-hacia-frente (AARF), el segundo describe el método de visualización del tipo atrás-hacia-antes (FAA) propuesto para Doodly y Torborg [3] y el tercer se corresponde al enfoque propuesto por Gargan [6], que utiliza "quadtrees" en la visualización.

4.1. Un algoritmo frente-hacia-atrás (Frazier et al.)

El método propuesto se recorre por capas ("cubes"), filas y columnas, el arreglo en orden decreciente respecto de la distancia al observador. Se asume que el objeto que se va a pintar se divide en capas perpendicularmente al eje Z. Los ejes X e Y están junto al objeto como en la siguiente figura.

El proceso se especifica la orientación del objeto por...
El esquema A permite dar al paciente sobre el eje longitudinal z, B es la inclinación sobre el eje x, y C es la retención sobre el eje y. Debido a que los datos, en este tipo de aplicaciones, se almacenan por capas y dentro de cada capa por filas, los "vóxels" se recorren de manera que el bucle más externo es el de las capas (dirección z), dentro de cada capa se define otro bucle externo para las filas (dirección y) y un bucle más interno se define en la dirección x. Las coordenadas de un "vóxel" en el espacio (x,y,z) corresponden en el Escalamiento IMAG s:

\[ x = T \cdot x', \quad y = T \cdot y', \quad z = T \cdot z' \]

donde T es la matriz de rotación definida así:

\[
\begin{pmatrix}
T_{xx} & T_{xy} & T_{xz} \\
T_{yx} & T_{yy} & T_{yz} \\
T_{zx} & T_{zy} & T_{zz}
\end{pmatrix}
\]

Las dimensiones de la caja que contiene el objeto son [X, Y, Z] y las de la ventana donde se piensa [Xw, Yw, Zw]. El tamaño de la ventana es [xw,yw] y el número de niveles de gris es Zg, siendo s el factor de escala.

4.2. Algoritmo de visualización (Doober y Torborg)

En este enfoque se trabaja en un modelo de visualización "octree" en una imagen codificada como "quadtree". La imagen que genera representa una proyección perspectiva del enrejado o D que emula un observador distante que está viendo directamente la corte frontal del universo. Desde este punto de vista, cada "pixel" en la imagen está alineado con una cadena de "vóxels" en el arranque espacio. Adicionalmente, cada cuadrante del "octree" está eliminado con un par de octantes, uno directamente en frente del otro. Una visualización retrospectiva se obtiene llevando a cabo una transformación geométrica sobre el objeto "octree", antes de iniciar el proceso de visualización.

El problema de visualización de objetos se divide en problemas simples: el de eliminación de caras ocultas por un rayo en el que los "vóxels" visibles no han en un punto de vista el problema de figuras, en el que la orientación de las sombras determina la ubicación de iluminación reflejada al observador y el de semi-transparente que no se detalla aquí.

4.2.1. Desarrollo del algoritmo

El algoritmo utiliza un conjunto de funciones para relacionar la base de datos que almacenó, la estructura de arranque y la eliminación. La función OCTREE(DATA, ELEMENT, INDEX) se utiliza para acceder a los descendientes del nodo OCTREE, DATA ELEMENT, en el "octree" del objeto, especificado por INDEX. Si la región es homogénea, se utiliza la función OCTREE, para acceder a los descendientes de un nodo de dato; en otro caso, se utiliza un puntero de OCTREE, DATA ELEMENT, para ir a un nodo particular.

Una función similar OCTREE, DATA ELEMENT, INDEX accede a los descendientes de nodos "octree". La función booleana OCTREE, DATA ELEMENT, ... determina si la región o regiones representadas por el (los) elemento(s) de datos son homogéneos. VAL1(DATA, ELEMENT, INDEX) invierte el valor del dato que leña en la región descrita por DATA ELEMENT. Si esta región es homogénea, el valor es 1; en el caso contrario, es 0.

La función COLOR, VALUE, es una tabla que contiene el color de cada uno de los materiales a visualizar. VALU para una región "octree" de fondo, en la tabla para encontrar el color de la región descrita.

Finalmente, CREATE, NEW, OCTREE, NODO, DATA, ELEMENT, ... toma como entrada...
de 4 elementos de datos que representan los cuadrantes de un área en la imagen. Si el área descontada por estos es heterogénea se crea un nuevo nodo "quadtree" que se lleva con estos elementos y un puntero al nuevo nodo. En otro caso, se devuelven el elemento de datos homogéneo correspondiente.

4.2.2. Eliminación de caras ocultas
Un "octree" es una estructura que facilita recorrer un árbol de manera que los nodos se visiten en una dirección consistente, a través del espacio. Esto se hace recorriendo el "octree" en profundidad-primero mientras el procesamiento de los elementos de datos se hace en un orden preestablecido. Si los elementos de datos en cada nodo se procesan en orden ascendente (0, 1, ..., 7) los cuatro octantes del frente del universo (0, 1, 2, 3) se visitan antes que los cuatro de atrás (4, 5, 6, 7). De igual manera, los cuatro octantes del frente del octante-cero se procesan antes que los cuatro de atrás. Esto continúa recursivamente hasta que los elementos de más al frente del universo se visitan primero y los que están más atrás de último. La facilidad de visitar los "volumes" en un orden especialmente predeterminado hace que el problema de caras ocultas sea sencillo de resolver.

El conjunto de "volumes" visibles se extrae de un "octree" de entrada a través del procedimiento recursivo que transforma el "octree" en un "quadtree", que representa la imagen directamente en el frente de la región "octree". El resto del "octree" permanece sin cambios. El procedimiento trabaja recursivamente así: Si el "octree" representa una región homogénea, el "quadtree" está siempre definido. Si la región "octree" no es vacía, entonces, la imagen última se oscurece y se reprimiere. El nuevo "quadtree" tiene el color correspondiente al material que se reprimiere. Si la región "octree" es vacía, el "quadtree" se devuelve sin cambios. Si es heterogénea, el "quadtree" es indefinido y se crea un nuevo nodo:"quadtree". El valor de cada cuadrante del nuevo nodo "quadtree" está determinado por el par de octantes eliminados con ese cuadrante. Los octantes de y 4 del "octree", determinan el resultado del cuadrante 0 del "quadtree". Similarmente, los octantes 1 y 6 del "octree" determinan el resultado del cuadrante 1 en la imagen y así para los otros dos cuadrantes. Como el valor de cada elemento de datos "quadtree" depende de dos octantes del "octree", se hacen dos llamadas al procedimiento por cada elemento del nuevo nodo. Los octantes de atrás (4, 5, 6, 7) se procesan primero que los del frente (0, 1, 2, 3) para todos los cuatro cuadrantes.

Después de que ambos, los del frente y los de atrás, han sido procesados el "quadtree" mantienen el resultado visible de la región "octree". Este procedimiento se invoca pasando el elemento del universo del "octree" que se va a visualizar y un elemento de datos homogéneo para el "quadtree" de entrada que representa el color del fondo del universo.

4.3. Enfoque con transformaciones de vistas (Gargantini, Walsh y Wu)
Bajo este enfoque se llevan a cabo transformaciones de vistas para visualizar un objeto representado por un "octree". Los algoritmos trabajan sobre un "octree" lineal, utilizando un mecanismo (BIT) asociado para mantener registro de la ausencia o presencia de vectores similares. El "octree" lineal incorpora una prueba de visibilidad ordenada en profundidad para los "volumes", mientras que el mecanismo BIT incorpora información sobre visibilidad de las caras de cada "volumes".

4.3.1. Definiciones
Un "raster" 3D, es un "array" de "volumes" 2x2x2 donde r es la resolución. Un gráfico tridimensional es un "raster" 3D en el que cada "volumes" tiene asignado un atributo (nivel de gris, color, textura, atole-nes). El conjunto de "volumes" con ese atributo es el fondo y los demás son el objeto. Cuando hay grandes bloques de "volumes" con el mismo atributo se representan por medio de un "octree". Un "octree" lineal es una forma comprimida de un "octree" regular en el que se almacenan solo los trazos que pertenecen al objeto 3D. Con una etiqueta clave que conste en un diagramo octal. Esta clave permite encontrar, fácilmente, el camino desde la raíz hasta la hoja del "octree" regular correspondiente y, en consecuencia, la subdivision del "raster" que produjo el octante correspondiente a igual que el tamaño de este octante y su localización en el "raster". Los octantes se etiquetan como aparece en la tabla:

<table>
<thead>
<tr>
<th>Octante</th>
<th>Dígito octal</th>
<th>Dígito 3-bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>arriba-este</td>
<td>0</td>
<td>000</td>
</tr>
<tr>
<td>arriba-norte</td>
<td>1</td>
<td>001</td>
</tr>
<tr>
<td>arriba-norte-este</td>
<td>2</td>
<td>010</td>
</tr>
<tr>
<td>arriba-norte-oeste</td>
<td>3</td>
<td>011</td>
</tr>
<tr>
<td>debajo-norte-este</td>
<td>4</td>
<td>100</td>
</tr>
<tr>
<td>debajo-norte-oeste</td>
<td>6</td>
<td>101</td>
</tr>
<tr>
<td>debajo-este</td>
<td>7</td>
<td>111</td>
</tr>
</tbody>
</table>

Etiquetamiento de octantes
El "octree" lineal de un objeto es un vector ordenado de códigos octales. N es al número de nodos en el "octree" lineal de un objeto. Los vectores de un "volumes" Q son aquellos "volumes" con los que Q comparte una cara, en cualquier de las seis principales direcciones (sur, norte, este, oeste, frente, atrás). El borde de un objeto tridimensional es el conjunto de "volumes" del objeto que tienen al menos un vecino en el fondo, o están en el borde del "raster".

4.3.2. Transformación de coordenadas 3D
El objeto se localiza originalmente en un sistema de
coordenadas de mano-derecha (x,y,z), llamado coordenadas del mundo (WCS) y la revolución alrededor del origen de WCS al observador está dada por el vector (x,y,z), llamado plano normal (VPN). El proyectil se convierte en WCS en otro sistema coordenado de mano-derecha (UVN) llamado coordenadas de referencia visual (VRN). Después de rotar los ejes x, y, z, sobre el eje y, se obtiene el orden en que el eje positivo de la z coincide con el VPN, que está ahora en el plano x-y. Hasta ahora, han rotado tres ejes delantero del origen, pero el origen está, convencionalmente, localizado en una esquina del "raster" y se quiere visar el objeto alrededor del centro del "raster" para que éste permanezca en el "raster". Para hacer esto, primero se trasladan los ejes x, y para que el origen está en el centro, se ejecutan las rotaciones y finalmente se trasladan de nuevo los ejes a la posición original de manera que el WCS y el VRN sean idénticos para el centro del "raster". Para un "raster" VP/VN en el cuarto plano, llamado WVRN (W, V, R, N) y la matriz "Viewing" llamada VIEW-MATRIX, por la cual cualquier vector deba multiplicarse para una representación de WCS a VRN es,

\[
\begin{align*}
\sin ^{2} \theta - \cos ^{2} \theta & = 0 \\
0 & = \cos \theta \\
0 & = \sin \theta \\
\end{align*}
\]

\[
\text{VEWMATRIZ} = \text{cos} \theta \cdot \text{sin} \phi \cdot \text{cos} \psi + \text{sin} \theta \cdot \text{sin} \phi \cdot \text{sin} \psi + \text{cos} \theta \cdot \text{cos} \phi
\]

Donde \( (\theta, \phi, \psi) \) son los ángulos de rotación. Los siguientes ejes x, y, z se llaman UV, V y N respectivamente.

4.3.3. Caras ocultas
Una de las ventajas de la representación de objetos 3D, por medio de estaciones, es que la mayoría de los problemas de visibilidad se resuelven simplemente vislumbrando las coordenadas del VPN y el código octal. Usar nuevo VPN=x, y, z, y suponemos que x, y, z. De esta manera, exactamente "ocultas" de un cubo son visibles como se muestran en la tabla siguiente.

<table>
<thead>
<tr>
<th>Visibilidad de un nodo respecto de las coordenadas VPN</th>
<th>VNP</th>
<th>Caras Visibles</th>
</tr>
</thead>
<tbody>
<tr>
<td>x&lt; 0 (x&lt; 0)</td>
<td>sur (lado)</td>
<td></td>
</tr>
<tr>
<td>y&lt; 0 (y&lt; 0)</td>
<td>lados (lados)</td>
<td></td>
</tr>
<tr>
<td>z&lt; 0 (z&lt; 0)</td>
<td>frontal (frontal)</td>
<td></td>
</tr>
</tbody>
</table>

Si exactamente una coordenada es 0, entonces exactamente 3 caras son visibles: si dos coordenadas son 0, exactamente una cara es visible. Se puede definir un orden de prioridad (basado en la codificación de "ocultas" líneas) permitiendo establecer, recu- rreando, el orden en el cual los subocultes de un octante están se deben ocultar en la forma anterior. El subocultante más lejano se examina primero, seguido por los tres vecinos de cara (en cualquier orden). Luego, se examinan los tres vecinos de cara (en cualquier orden) del subocultante más cercano, seguido, finalmente, por el máximo subocultante más cercano. El subocultante más cercano corresponde a los tres caras visibles; que se determinan con base en la tabla de visibilidad de nodos y cuyo código octal se encuentra en la tabla de etiquetado de octantes. Los subocultantes se examinan en orden decreciente al número de "bits" por el cual sus dígitos octales divergen del subocultante más cercano.

Si el código octal se ve desde la esquina frontal-orientada del octante \((x< 0, y< 0, z< 0)\), el subocultante más cercano tiene enteros octales \(6,7,8\) y uno de los 36 posibles ordenamientos de \(1,2,3,4,5,6,7,8\). El orden que se usa para todos los otros nodos se decríbe en la tabla siguiente:

<table>
<thead>
<tr>
<th>Orden de prioridad para etiquetado de octantes de acuerdo al VPN ((-x,-y,-z))</th>
<th>Código octal</th>
<th>Orden</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0, 6, 7, 8</td>
<td>1, 2, 3, 4, 5, 6, 7</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0, 6, 7, 8</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>6, 7, 8</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>6, 7, 8</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>6, 7, 8</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>6, 7, 8</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
</tbody>
</table>

4.3.4. El Procedimiento
Un primer algoritmo, dividido recientemente el "ocultos" lineales en octantes, los examina en orden de prioridad y cuando encuentra un subocultante, que es un nodo, éste se incluye en un "pipeline" de visualización. Este algoritmo determina todas las caras visibles de un nodo. Un enfoque más eficiente que no desarrollado para crear un "pipeline" de visualización, se basa en el hecho de que cuando una cara de un nodo objeto se comparte con el objeto, o con un nodo de color diferente, se puede detectar automáticamente, mientras se execute el algoritmo de detección del borde 3D para encontrar los "ocultos" superficie.

El algoritmo para determinar si estas bordes o límites de un objeto, entregan todos los "ocultos" suponiendo que un conjunto de códigos octales. Para indicar cuáles caras son un "oculto" margen faltante, el algoritmo asigna al nodo "oculto" un "bit" de bloque-de-fondo, uno por cada una de las seis direcciones principales, donadas por E,S,D,W,N,F. Utiliza, ademá, un "bit" adicional, bloque-de-fondo, para un
5. BIBLIOGRAFÍA


AUTORES

Ana Esther Rodríguez es candidata a doctorado en informática en la Universidad Politécnica de Madrid (España). Actualmente trabaja en el área de Bases de Datos del proyecto de Política Agraria Comunitaria de la Unión Europea.

Marta Elena Millán, Doctor en Informática de la Universidad Politécnica de Madrid es profesora del Departamento de Ciencias de la Comunicación de la Universidad del Valle.