Aplicación de los árboles binarios de particionamiento espacial al modelado multirresolución en 2D y 3D.

Joaquín Huerta†, Miguel Chover†, José Ribelles†, Ricardo Quirós†
Universidad Jaume I. Departamento de Informática.
Campus de Riu Sec. 12071 Castellón. SPAIN.
huerta@uji.es

Resumen

En este trabajo se presenta un modelo que combina herramientas de modelado multirresolución con técnicas de particionamiento espacial, concretamente con árboles binarios de particionamiento espacial (árboles BSP). Mediante esta combinación de técnicas se acelera la visualización de modelos geométricos. Para ello se adapta la cantidad de detalle que se utiliza para la visualización a la distancia desde la que se esta observando cada objeto. Cuando la distancia sea grande se utiliza una representación muy simplificada puesto que en cualquier caso, los detalles no se apreciarían desde dicha distancia. Cuando el modelo esta cerca se utiliza una representación que contenga todos los detalles para que no se produzcan perdidas en la calidad de la visualización. Mediante el modelo multirresolución desarrollado se pueden obtener multitud de representaciones con distinta resolución. Para almacenar estas representaciones se utiliza un árbol BSP. El uso de este tipo de estructura proporciona una preclasificación del orden de visualización de los polígonos del modelo. Esta ordenación permite acelerar la visualización ya que no es necesaria ninguna operación adicional para calcular la visibilidad de los modelos.

Palabras clave: Árboles BSP, particionamiento espacial, modelado multirresolución.

1. Introducción

En la mayoría de las disciplinas relacionadas con la informática gráfica es frecuente el uso de modelos de gran complejidad geométrica que supera ampliamente los avances hardware. Así pues, adaptar la resolución de tales modelos a las necesidades de cada aplicación se ha convertido en un objetivo fundamental para las aplicaciones de visualización en general y sobre todo en el caso de entornos interactivos en tiempo real.

Para atenuar este problema se pueden adoptar dos soluciones: o bien usar técnicas de visualización eficientes que permitan manejar escenas muy complejas utilizando representaciones simplificadas de estas, o bien usar técnicas de modelado eficientes que permitan acelerar los cálculos de visibilidad para obtener un número aceptable de fotogramas por segundo, incluso en el caso de modelos muy complejos. En el primer grupo se situarían los *modelos multirresolución* [3] que permiten representar una escena con diferentes niveles de detalle. Utilizando estos modelos sólo se visualizan aquellas características de los objetos que contribuyen realmente al aspecto final de la imagen según las condiciones de visualización. Dentro del segundo grupo se encuentran las *Técnicas de Particionamiento Espacial*. Estas técnicas basan su eficiencia en el hecho de que aprovechan la propia estructura de datos en la que se almacena la geometría para acelerar las operaciones básicas de visualización de objetos. Una de las técnicas de particionamiento espacial más eficiente que se conoce son los *árboles BSP* (*Binary Spatial Partitioning Trees*) [1]. Estos arboles permiten disponer a priori de una clasificación de los polígonos por orden de visibilidad de forma que se evitan los cálculos en tiempo de ejecución.

El trabajo que se presenta trata de aunar las dos soluciones expuestas integrando las ventajas de las técnicas de particionamiento espacial y las de los modelos multirresolución para construir árboles BSP multirresolución.

En la siguiente sección se presentan brevemente los fundamentos teóricos de los árboles BSP. Las secciones tercera y cuarta muestran los modelos desarrollados para dos y tres dimensiones respectivamente. Por último, la sección quinta aparecen los resultados obtenidos.

2. Árboles BSP

Los árboles BSP [1] son una generalización de los árboles K-D[7] en la que los planos de partición pueden tener cualquier orientación. Producen una subdivisión jerárquica y recursiva de un espacio convexo de dimensión d en regiones convexas de la misma dimensión. Proporcionan una representación computacional del espacio que puede utilizarse al mismo tiempo como estructura de búsqueda y como modelo de representación geométrica. Mediante estos árboles se consiguen soluciones eficientes para un gran número de problemas haciendo uso de dos propiedades muy simples que ocurren cuando un plano separa dos o más objetos:

1. Cualquier objeto en un lado del plano no puede intersectar ningún objeto situado al otro lado.
2. Dado un punto de vista, los objetos que estén en el mismo lado de un plano que dicho punto de vista nunca serán ocultados por objetos que estén en el lado opuesto

Estas propiedades se pueden hacer independientes de la dimensión usando el termino hiperplano para referirnos a líneas en 2D, planos en 3D y en general para dimensión d a un subespacio de dimensión $d-1$ definido por una ecuación lineal. La construcción de un árbol BSP se realiza dividiendo recursivamente un subespacio por medio de un hiperplano en dos semiespacios. La figura 1 muestra la representación de un polígono mediante un árbol BSP. Se trata de un ejemplo simple en dos dimensiones que fácilmente puede extenderse para tres dimensiones.

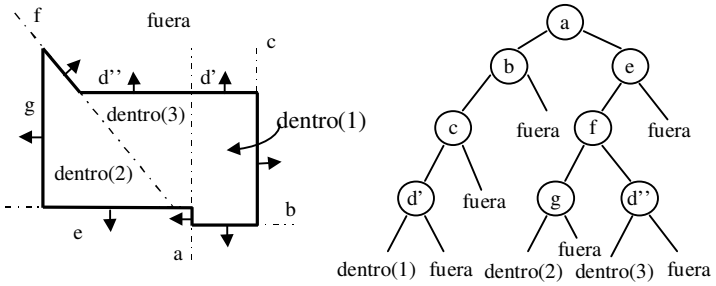


Figura 1: Representación de un polígono mediante un árbol BSP

3. Árboles BSP multirresolución en 2D

El objetivo de este trabajo es que los árboles generados sean multirresolución y para conseguirlo su estructura debe ser de tal manera que los nodos superiores representen una aproximación inicial del objeto, y dicha aproximación se vaya refinando a medida que descendemos por las ramas del árbol. El criterio utilizado en la construcción del árbol es el de añadir primero aquellos planos que más cantidad de detalle aporten a la representación y luego los que menos, de forma progresiva. La diferencia entre un árbol BSP convencional y un árbol BSP multirresolución, para la representación de polígonos, se muestra en las figuras 1 y 2.

Como se puede observar, el árbol de la figura 1 produce una partición del polígono en tres regiones de un tamaño similar. Por el contrario, la partición producida por el árbol de la figura 2 es tal que existe una región de mayor tamaño que representa una buena aproximación inicial del polígono más dos pequeñas regiones que refinan el detalle. Almacenando en los nodos superiores del árbol los planos que delimitan la región de mayor tamaño y en sus descendientes los que delimitan las regiones más pequeñas se consigue que el árbol tenga una estructura multirresolución. Mediante un simple

proceso de poda se pueden desechar los detalles menos significativos de un modelo, reduciendo su complejidad y conservando su apariencia. Solo se permitirá realizar la poda en algunas ramas del árbol, ya que de lo contrario se obtienen resultados sin sentido.

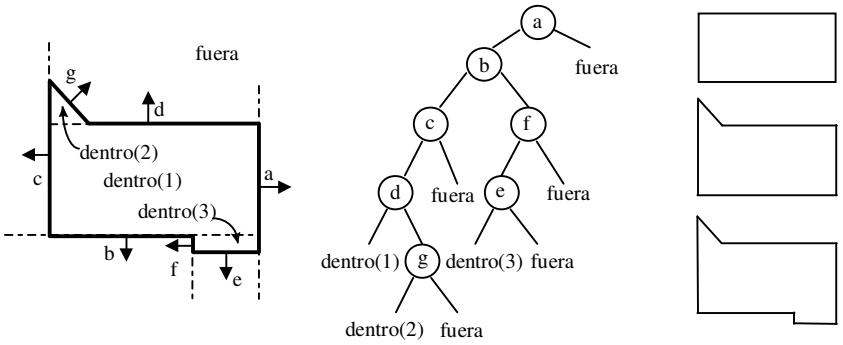


Figura 2: Polígono, árbol BSP multirresolución y secuencia de aproximaciones

3.1- Construcción del Árbol

En primer lugar, usando alguno de los métodos de aproximación presentados en [4] se calcula una aproximación inicial del polígono. El número de lados de la misma es tres. Después se crea el árbol *BSP* 2D que contiene dicha aproximación. Este árbol se compone de nodos en los que se almacenan planos de partición 2D. En cada plano se almacenan varias aristas que representan la apariencia del polígono con varias resoluciones. A cada arista se le asigna un rango de visibilidad que determina cuando se visualizará cada arista. Se trata de un mecanismo muy simple: cuando se vaya a realizar la visualización bastara con determinar con que precisión se desea visualizar el objeto para extraer del árbol las aristas cuyo rango contenga ese valor.

Si alguno de los planos insertados corta alguna de las aristas del polígono se añaden nuevos vértices a la lista de vértices en dichos puntos de corte. A continuación se clasifican los vértices de la lista en secuencias de vértices que se encuentren en la misma región del árbol. El proceso se aplica de forma recursiva hasta que la lista de secuencias llegue a estar vacía.

3.2- Visualización del modelo

En la etapa de visualización, el primer problema a resolver es el de decidir que representación se va a utilizar de entre todas las posibles, es decir, que aristas se van a visualizar de entre todas las aristas que contiene el modelo. En [2] se presenta un estudio de diversos métodos de estimación y selección del nivel de detalle. En este trabajo, para decidir que aristas deben visualizarse, se usa un método sencillo que estima la precisión aceptable para las condiciones de visualización en cada momento. A dicho parámetro se le denomina *umbral* y se calcula como una función lineal del porcentaje de la ventana que ocupa la proyección de la caja de inclusión del objeto. Dicho porcentaje se puede ajustar según la precisión que se desee en la visualización.

Una vez determinado el *umbral*, el procedimiento consiste en clasificar recursivamente el punto de vista dentro del árbol visualizando primero las aristas que están en nodos delante de este, luego los que están en el mismo nodo y finalmente los que están en nodos detrás del mismo. De entre las aristas que se encuentren en cada nodo, solo aquellas cuyo rango de visibilidad contenga el umbral serán visualizadas.

Para la realizar la visualización se utiliza un procedimiento de delante hacia detrás, al contrario que el popular algoritmo del pintor. Se basa en la estructura denominada "*dinamic screen*" [6] que permite ir marcando las zonas de la ventana que ya han sido dibujadas y también permite conocer cuando se han completado todos los pixeles de la ventana.

4. Árboles BSP multirresolución en 3D

El modelo descrito en el capitulo anterior para el caso de dos dimensiones se puede extender al caso de tres dimensiones aplicándolo a mallas de triángulos cerradas. El modelo resultante es un Árbol BSP multirresolución *3D*, en el que la aproximación inicial es un tetraedro en lugar de un triángulo y los planos del árbol contienen triángulos en lugar de aristas.

En el caso tridimensional, el modelo que se propone se compone de un tetraedro inicial al cual se van añadiendo nuevos tetraedros mediante operaciones booleanas de *unión* o *diferencia*. Dicho tetraedro inicial debe representar una aproximación inicial de la malla y los tetraedros siguientes aportan de forma progresiva los detalles restantes, hasta llegar a la representación exacta de la malla original. La figura 3 muestra un ejemplo sencillo de este tipo de representación

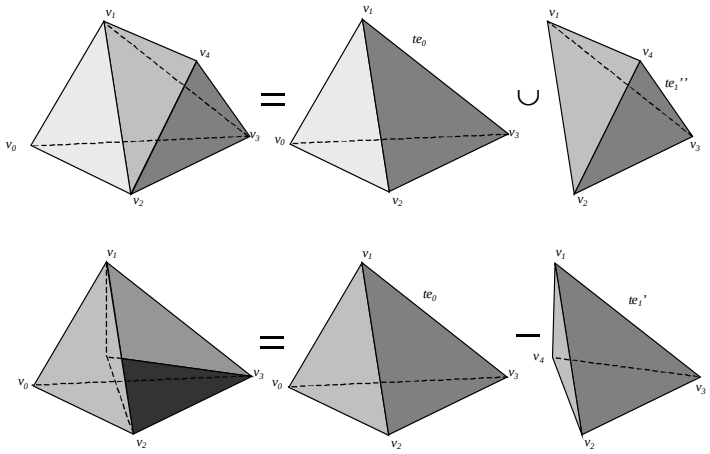


Figura 3: Modelo basado en tetraedros

El proceso de construcción del árbol comienza aplicando alguno de los métodos de cálculo de aproximaciones descritos en [4]. Como resultado se obtiene una aproximación inicial formada por cuatro vértices y cuatro triángulos no coplanares. Esta constituye el tetraedro inicial partir del cual se construye el modelo *MRBSP*. Una vez calculada dicha aproximación inicial, se crea el árbol que inicialmente tiene cuatro nodos, la raíz y sus tres descendientes traseros.

A medida que se van insertando los nuevos planos en el árbol se determina si estos producen algún corte en los triángulos de la malla. En caso afirmativo, se calculan dichos puntos de corte y se añaden a la lista de vértices. Además se dividen los triángulos afectados por los cortes y se actualiza la lista de triángulos añadiendo los nuevos triángulos que resultan de la división.

A continuación se realiza el proceso de particionamiento y clasificación de la malla. Como resultado se obtiene un conjunto de submallas formadas por. un conjunto de triángulos adyacentes que están en una misma región del árbol. La figura 3 muestra el conjunto de submallas resultantes de este proceso.

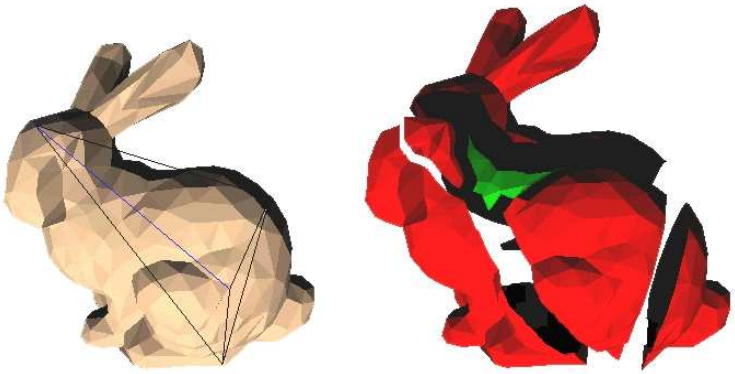


Figura 4: Submallas resultantes del proceso de particionamiento.

Dichas submallas se almacenan en una lista ordenada por volumen. El proceso se aplica recursivamente a los elementos de dicha lista y termina cuando la lista de submallas se vacía. En cada momento se procura añadir al modelo la submalla que mayor contribución aporte a la disminución del error. Dado que la función de medida de error utilizada esta basada en el cálculo de volúmenes, el orden de proceso de submallas según su volumen es el que mejores resultados produce.

Para determinar los límites del *rango de visibilidad* de los triángulos, se usa una función heurística que se ha denominado *precisión*. Dicha función refleja la exactitud con la que se esta aproximando la malla original en cada momento del proceso de creación del árbol. Este parámetro es equivalente al definido para el caso *2D*.

Como resultado del proceso de construcción del árbol se obtiene un modelo en el que aparecen *ranuras* no deseadas, inexistentes en la malla original, como las que se muestran en la figura 5. Dichas ranuras se pueden detectar fácilmente, ya que dan lugar a la aparición en el árbol de dos nodos situados en planos coincidentes, con dos triángulos idénticos, cuya orientación es opuesta. Una vez detectadas, para evitar que se visualicen se modifica la estructura del modelo eliminando dichas ranuras.

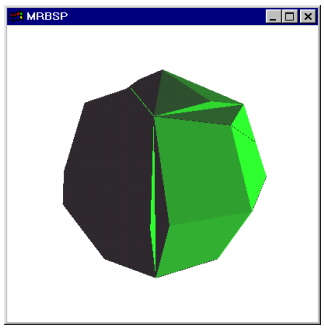


Figura 5: Ranuras.

El procedimiento que se utiliza para realizar la visualización es idéntico al utilizado en el caso 2D. En este caso, en lugar de visualizar aristas se visualizan triángulos ya que se trabaja con mallas de triángulos. Así pues, estos se pasan directamente a la rutina de visualización, sin necesidad de realizar ningún cálculo adicional.

5. Resultados y conclusiones

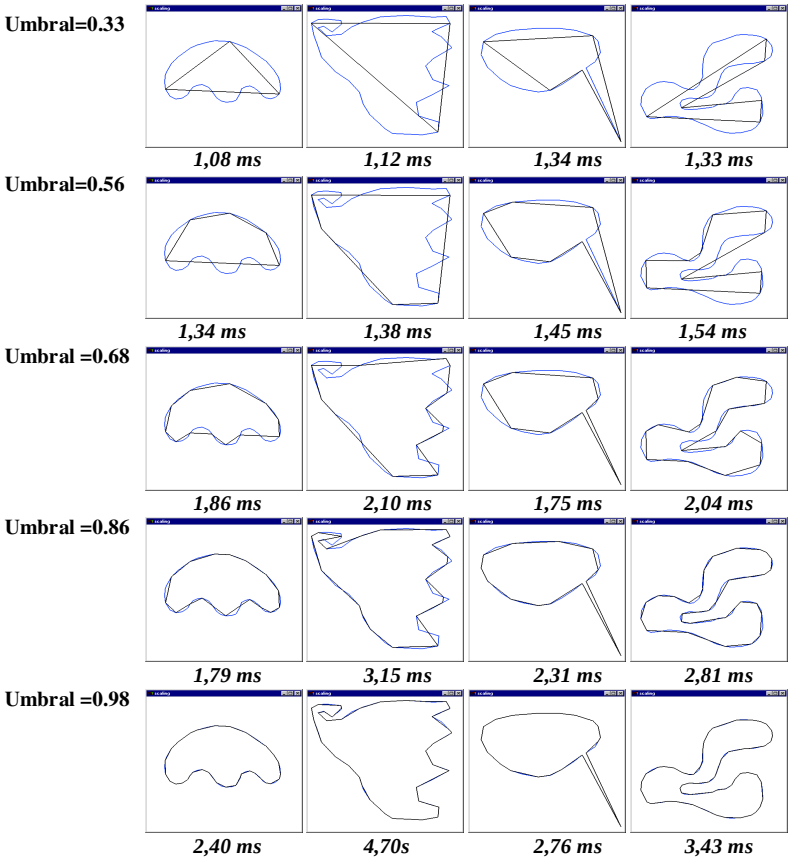


Figura 6. Visualización en función del umbral.

La figura 6 presenta el resultado de aplicar el método de aproximación por escalado para construir un modelo *MRBSP 2D* de algunos polígonos de ejemplo. En dicha figura pueden apreciarse las distintas representaciones que se obtienen en función del *umbral* de precisión que se establezca para la visualización. También puede observarse los tiempos de visualización para cada una de las representaciones.

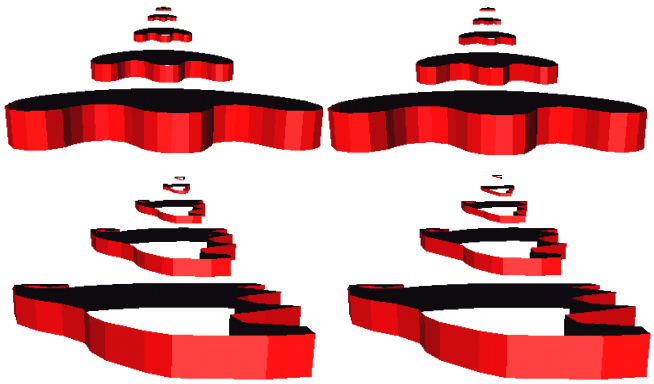
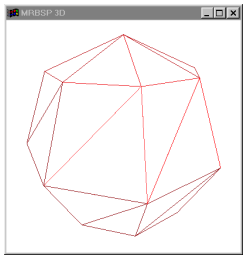


Figura 7. Izquierda: Objeto original visualizado desde varias distancias. Derecha: Aproximaciones del objeto desde las mismas distancias.

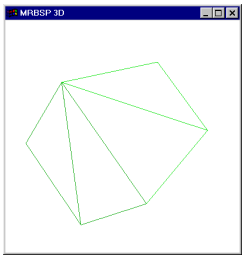
La figura 7 muestra al mismo tiempo la visualización que se obtiene utilizando un modelo monorresolución y el modelo *MRBSP 2D*. En este caso se aprecia como cambia el umbral de representación empleado para la visualización en función de la distancia de observación. Dicho modelo presenta como inconveniente fundamental su elevado coste espacial. Ello se debe a la gran cantidad de cortes que se producen. Como se ha expuesto anteriormente, cada corte supone la inclusión de un nuevo vértice y dos nuevos triángulos. El numero de cortes tiene un coste de $O(n^2)$ [5]. La tabla 1 muestra los resultados obtenidos para algunas mallas de ejemplo sencillas.

	Modelo 1	Modelo 2	Modelo 3	Modelo 4
Vértices iniciales (N_V)	26	114	482	1106
Triángulos iniciales (N_T)	48	224	960	2208
Cortes (N_C)	65	881	4730	14989
Vértices totales (N_V+ N_C)	91	995	5212	16095
Número nodos N_N	124	1306	6979	21398
Número tetraedros (N_D)	46	426	2260	6823
Ranuras eliminadas	30	199	1032	2974

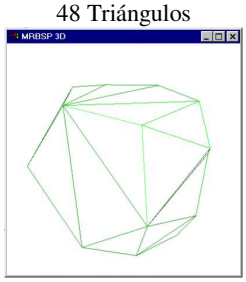
Tabla 1: Resultados de la construcción del modelo MRBSP.



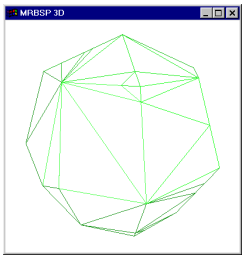
Modelo monorresolución. 48 Triángulos



MRBSP. 12 Triángulos. Precisión=0.54



MRBSP. 30 Triángulos. Precisión=0.78



MRBSP. 78 Triángulos. Precisión=1.0

Figura 8: Resultados para el árbol MRBSPT 3D.

La figura 8 muestra el resultado obtenido al construir el modelo MRBSP 3D del modelo 1 referenciado en la tabla 1. Arriba a la izquierda aparece la malla original monorresolución. Las otras tres imágenes muestran el modelo multirresolución para tres umbrales de precisión diferentes.

6. Bibliografía

[1] H Fuchs, Z. Kedem, B. Naylor. *On Visible Surface Determination by a Priori Tree Structures*, Computer Graphics, vol. 14, nº 3, paginas. 124-133, Junio 1980.

[2] T.Funkhouser y C. Sequin. *Adaptive display algorithm for interactive frame rates during visualization of complex environment*. Proc. SIGGRAPH 93, páginas 247-254. 1993.

[3] P. Heckbert, M. Garland, *Multiresolution Modeling for Fast Rendering*, Proceedings of Graphics Interface '94. páginas 43-50, 1994.

[4] J. Huerta. *Aproximación multirresolución de los árboles binarios de particionamiento espacial.* Tesis Departamento de Informática. Universidad Jaume I Castellon. 1999.

[5] Paterson, M.S., Yao, F.F. 1990 *Efficient Binary Space Partitions for Hidden Surface Removal and Solid Modeling, Discrete Computational Geometry*, no 5, pp 485-503.

[6] R. A. Reynolds, D. Gordon y L. S. Chen*, A Dynamic Screen Technique for Shaded Graphics Display of Slice-Represented Objects*, Computer Vision, Graphics and Image Processing, Volumen 38, número 3, páginas 275-298, Junio 1987

[7] F. Subramanian, D. Fussel. *K-D Trees*. Technical repport. Computer Science Department. University of Texas. 1989.