

INTEGRACIÓN DE DIFERENTES MODOS DE COHERENCIA EN LA ACELERACIÓN DEL TRAZADO DE RAYOS

Pascual González López

Pascual González López está en el Departamento de Informática. Escuela Universitaria Politécnica de Albacete. Universidad de Castilla-La Mancha.

RESUMEN

La generación de imágenes foto-realistas mediante ordenadores es hoy en día una realidad. Existen algoritmos muy sólidos que permiten reflejar de modo fiel los efectos que la luz provoca en el mundo real. El trazado de rayos es sin duda uno de sus mejores representantes, pero éste, junto a su simplicidad conceptual, lleva implícito un alto coste computacional que dificulta su aplicación en ciertos entornos. En este trabajo se ofrece una nueva alternativa que ayuda a reducir el tiempo necesario para generar una imagen con este método. Tras exponer las ideas básicas del trazado de rayos y analizar algunas de sus características, expresadas en niveles o modos de coherencia, en la última parte del trabajo se presenta un nuevo algoritmo que basa su propuesta de aceleración en explotar de modo conjunto varios de estos modos. La idea básica de este nuevo algoritmo es trazar a la vez una serie de rayos agrupados en haces (coherencia de rayos). La definición y trazado de estos haces se apoya en la descomposición de la escena en una serie de áreas homogéneas, mediante árboles octales (coherencia de objetos).

1. GENERACIÓN DE IMÁGENES «FOTO-REALISTAS»

EN los últimos años han surgido sistemas gráficos sofisticados, motivados por la aparición de métodos que permiten simular los efectos de la luz en el mundo real. Los primeros sistemas trataban los efectos de iluminación como una tarea no básica dentro del proceso de generación de imágenes en el ordenador, centrándose en tareas tales como eliminación de superficies ocultas, representación de superficies, etc.

La aparición de los trabajos de Whitted [WHI80] a comienzos de los ochenta sobre el algoritmo *trazador de rayos*, abrieron el camino a un nuevo concepto dentro de la generación mediante ordenador de imágenes sintéticas de alta calidad, denominado foto-realismo.

Estos métodos pretenden simular los efectos que la luz provoca en la escena mediante el envío y seguimiento del trazado de una serie de rayos de luz dentro de la escena tridimensional a representar. En el proceso se generan un conjunto de rayos primarios que se envían a la escena, los cuales pueden chocar con alguno de los objetos de la misma, desencadenando en este caso la generación de nuevos rayos secundarios en función de las características del objeto (rayos reflejados, rayos transmitidos o rayos de sombra). El seguimiento de cada rayo primario y de los posibles rayos secundarios a los que dé lugar nos determinará el color de cada punto de la pantalla. Como vemos, la imagen generada con estos métodos depende fuertemente de la posición del observador (*dependiente de la vista*) lo que hace a este algoritmo muy útil para tratar los efectos luminosos de naturaleza especular.

A este primer grupo de algoritmos basados en las ideas iniciales de Whitted, se unen posteriormente otros que también obtienen buenos resultados en la generación de imágenes de alto nivel de realismo. Estos nuevos métodos, denominados de *radiosidad*, separan claramente las tareas que simulan la interreflexión de la luz, de las de obtención de la imagen final, considerándose como *independientes de la vista*. Esta separación de tareas permite realizar el cálculo de la iluminación de la escena una sola vez. Estos métodos permiten representar los efectos de naturaleza difusa de manera sencilla.

Como ya hemos señalado cada uno de los métodos anteriores resulta más adecuado para captar ciertos efectos luminosos, lo cual ha motivado que recientemente hayan surgido algoritmos, denominados generalmente algoritmos de *dos pasadas* que realizan una combinación de ambos. En estos en primer lugar se realiza un cálculo de la radiosidad que permite, además del tratamiento de los efectos difusos, detectar qué objetos o partes de ellos pueden considerarse, junto a las fuentes de luz previamente definidas, como focos lumínicos dentro de la escena. Esta primera etapa sirve como preprocesado para la realización del algoritmo de trazado de rayos que se centrará en el estudio de los efectos especulares y, en algunos casos, de los efectos de sombreado.

Junto a esta línea de investigación conjunta siguen teniendo vigencia aquellas que pretenden optimizar tanto el trazado de rayos como la radiosidad de forma independiente. En este contexto, nuestro trabajo se centra en el análisis del algoritmo trazador de rayos, sus problemas y sus posibles formas de aceleración. Tras el estudio teórico de las características en las que se desenvuelve el trazado de rayos, expresadas en niveles de coherencia, en la última parte de este trabajo se aporta una nueva alternativa de aceleración.

2. EL ALGORITMO DE TRAZADO DE RAYOS: ESTRATEGIAS DE ACELERACIÓN

La idea básica del algoritmo trazador de rayos es lanzar rayos desde el observador hacia la escena y determinar para cada uno de ellos la posible intersección con los diferentes objetos de la misma. El algoritmo inicial propone para ello la realización de dos tareas básicas: (1) calcular la posible intersección de un rayo con cada uno de los objetos de la escena; (2) en caso de que el rayo intersekte con más de un objeto, entonces se selecciona aquel objeto que esté más próximo al observador pues será el primero que el rayo encontrará en su camino.

Como vemos este método propone realizar de forma exhaustiva los cálculos de todas las posibles intersecciones de un rayo con cada objeto de la escena, por lo que la mayoría de los cálculos realizados no son útiles. Su coste, como apunta Whitted, depende casi exclusivamente de una sola operación: *calcular para cada rayo el punto de intersección entre éste y una entidad geométrica atómica en el espacio tridimensional*, llegando a suponer un 95% del tiempo total en la representación de escenas complejas. Por ello, este algoritmo, aunque muy simple de implementar, es muy costoso computacionalmente, tanto que puede ser que la generación de una escena pueda durar minutos e incluso horas dependiendo fundamentalmente de su complejidad (número de objetos en dicha escena) y de la resolución deseada (número de puntos de la pantalla). Junto a lo anterior este método, como muchos otros, sufre los efectos del enmascaramiento de altas frecuencias (aliasing), cuya solución suele abordarse mediante las técnicas de sobremuestreo [ARV89]. Estas técnicas en cualquiera de sus alternativas (uniforme, adaptativo, estadístico o estocástico) proponen trazar por cada punto de la pantalla varios rayos, multiplicando de este modo el número total de rayos a trazar y, por tanto, el tiempo de generación.

Este excesivo coste computacional y tiempo de generación, ha motivado la creación de una importante línea de investigación, todavía no cerrada, en la que, a través de diferentes soluciones, se pretenden solventar o reducir los problemas anteriormente expuestos.

Asociadas a la idea de acelerar el proceso de trazado de rayos se han propuesto varias estrategias que, como indican Arvo y Kirk [ARV89], pueden resumirse en: (1) reducir el coste medio derivado del cálculo de la intersección del rayo con el entorno; (2) reducir el número total de rayos que intersektan con el entorno; y (3) sustituir los rayos individuales por entidades más generales y complejas. A estas tres alternativas se les suele denominar «cálculo más rápido de intersecciones», «menor número de rayos» y «rayos generalizados», respectivamente.

La mayoría de estas alternativas basan su formulación en el estudio de los diferentes niveles de coherencia que presenta el trazado de rayos y que permiten la variación del algoritmo inicial para conseguir una mayor eficiencia. Encontramos básicamente cuatro formas de coherencia:

Coherencia de objetos:

Posiciones próximas en el espacio tienden a estar ocupadas por un solo objeto de la escena o un conjunto reducido de ellos. Por tanto, la escena puede dividirse:

1. En un conjunto de celdas mediante un proceso de descomposición espacial que permiten realizar una simulación más precisa de la trayectoria de un rayo en la escena. Este proceso de descomposición se puede realizar de dos modos: descomposición uniforme en celdas de igual tamaño; descomposición no uniforme en celdas de diferente tamaño utilizando para ello tanto árboles binarios como octales.
2. Mediante una jerarquía de volúmenes de contorno. De este modo se reduce, para cada rayo, el número de objetos con los que debemos realizar el cálculo de su posible intersección.

Coherencia de rayos:

Rayos muy próximos entre sí, es decir, aquellos cuyo origen y dirección son muy semejantes, probablemente intersectan con los mismos objetos dentro de la escena. Esta propiedad se ha utilizado para reducir el coste computacional al trazar en vez de rayos individuales, haces de rayos. Estas técnicas encuentran como problema básico la determinación del tamaño y la forma del haz, es decir, determinar los rayos que pueden trazarse a la vez.

Coherencia de imagen:

Un objeto 3D tiende a ocupar después de su transformación a dos dimensiones posiciones contiguas en la pantalla, mientras que diferentes objetos suelen ocupar posiciones no contiguas. De este modo puntos de la pantalla (pixel) contiguos probablemente tengan un color semejante. Este modo de coherencia no es fácil de aplicar al algoritmo trazador de rayos, aunque podemos encontrar algunas implementaciones que realizan un pre-procesado que permite explotar después esta característica en el trazador de rayos.

Coherencia de fotograma:

Esta forma de coherencia es una ampliación del modo anterior al que se ha añadido la dimensión temporal y puede ser utilizada

en secuencias de imágenes en procesos de animación. Las imágenes (fotogramas) en una secuencia de animación tienden a ser semejantes y, por tanto, se puede optimizar el tiempo requerido para generar una imagen utilizando información de una imagen previa.

Junto a los niveles de coherencia la aceleración de este algoritmo se ve favorecida por el hecho de que los diferentes rayos pueden trazarse independientemente aportando, por tanto, grandes posibilidades de paralelismo. La introducción de versiones paralelas facilita la reducción del tiempo total de generación de imágenes permitiendo que podamos acercarnos al objetivo final de obtener imágenes de alto realismo en un tiempo de procesamiento reducido.

Esta aceleración, debida a la realización concurrente de varias operaciones, se puede conseguir también desde varias perspectivas: (1) vectorización; (2) utilización de máquinas paralelas de propósito específico para tratar este problema; (3) ejecución sobre máquinas paralelas de propósito general. Todas estas alternativas han sido abordadas por numerosos investigadores pudiendo encontrarse abundante bibliografía, entre otros, en el estudio de Arvo y Kirk [ARV89].

En nuestro trabajo pretendemos aprovechar conjuntamente tanto la coherencia de objeto como la coherencia de rayos, trazando haces de rayos sobre escenas descompuestas espacialmente a través de árboles octales. En la siguiente sección aportaremos un algoritmo que nos permita definir y trazar haces de rayos en este tipo de escenas.

3. COMBINACIÓN DE LA COHERENCIA DE OBJETOS Y LA COHERENCIA DE RAYOS

En este trabajo, como ya hemos indicado, proponemos utilizar de modo conjunto las características de coherencia de objetos y de coherencia de rayos. Esta nueva propuesta pretende reducir el tiempo necesario para realizar el trazado de rayos a través del árbol octal asociado a la escena a generar. Este tiempo, como indican McNeill et al. [MCN92], depende de la estructura de datos y de la escena en concreto, no siendo en ningún caso despreciable y llegando a suponer en la generación de algunas escenas un 40% del tiempo total de procesamiento.

En los siguientes apartados estudiaremos técnicas que permitan trazar los rayos primarios agrupados en haces en vez de trazarlos individualmente. Para ello nos apoyaremos en técnicas de descomposición basadas en árboles de orden n (octales para 3D y de orden cuatro para 2D) que permiten estructurar la escena a representar. Finalmente se muestran los resultados obtenidos por este algoritmo en escenas 2D descompuestas mediante árboles de orden cuatro (quadtrees).

3.1. División de la escena y definición de las estructuras necesarias

Uno de los problemas básicos con que se encuentran los distintos investigadores que trabajan con la coherencia de rayos es la definición del tamaño y la forma de los haces de rayos. Del mismo modo resulta costoso la determinación de la pertenencia de un rayo a un determinado haz. Esta dificultad, ha hecho que algunas soluciones, como la propuesta por Speer et al. [SPE86] en la que se pretendía aportar regiones cilíndricas para definir los haces, obtengan resultados peores que la solución estándar. En cualquier caso, aunque no todas las soluciones que utilizan la coherencia de rayos son satisfactorias, la idea general que subyace tras esta característica es muy interesante. Su relevancia es mayor a medida que aumenta el número de rayos a trazar y la proximidad entre dichos rayos. Esta situación se produce cuando deseamos obtener una imagen con una alta resolución o cuando introducimos dentro de nuestro algoritmo técnicas de sobremuestreo para resolver los problemas de enmascaramiento de altas frecuencias (aliasing).

En nuestro caso, el problema de la definición del tamaño y la forma de los haces se resuelve aprovechando la estructura que se obtiene tras dividir la escena 3D en áreas homogéneas mediante un árbol octal (figura 1), dando lugar a una estructura de descomposición no uniforme.

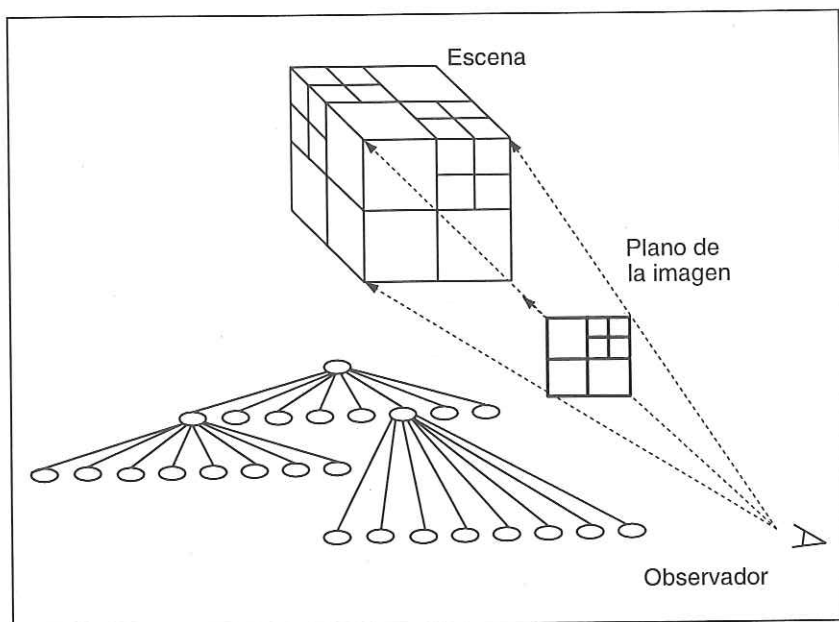


FIGURA 1.
Descomposición de la escena mediante un árbol octal (octree).

Un árbol octal es una representación jerárquica a través de la cual el espacio es subdividido en cada proceso de descomposición en ocho celdas (nodos) de igual tamaño, llamadas octantes. Los nodos del árbol podrán ser de tres tipos: (1) intermedios, es decir, aquellos que están divididos en otros de nivel inferior; (2) vacíos, es decir, aquellos nodos terminales en la jerarquía que no contiene en su interior ningún objeto de la escena; (3) ocupados, es decir, aquellos nodos terminales que contiene un número, generalmente reducido, de objetos de la escena.

El proceso de descomposición de la escena tiene en cuenta el número de objetos contenidos en un determinado nodo, subdividiéndose si dicho número supera un determinado umbral. A la hora de determinar el umbral debemos tener en cuenta que debe ser suficientemente pequeño para que contenga pocos objetos y suficientemente grande para optimizar el trazado de haces a través de la estructura. Un mayor número de objetos implica para cada rayo que llegue al nodo un mayor número de cálculos de posibles intersecciones. Por otra parte, un reducido tamaño del octante implica un proceso de descomposición más costoso y una mayor cantidad de memoria para almacenar el árbol octal. Al mismo tiempo, el tamaño de los nodos también afecta a los haces de rayos, reduciéndose la eficiencia de la traza de haces cuanto menor sea dicho tamaño.

El modo en que pretendemos recorrer la estructura de árbol impone ciertas modificaciones a la estructura estándar [SAM89]. Para realizar el trazado de los haces primarios o del resto de rayos secundarios o de sombra, en la definición de cada nodo junto a los campos que permiten referenciar a sus hijos, debemos incluir otro que permita acceder al nodo padre.

Tras la descomposición de la escena y la obtención del árbol octal asociado, debemos definir la estructura de los rayos y los haces a trazar en este tipo de escenas.

Un rayo, para realizar su trazado a través de la escena, puede definirse mediante su origen y un vector que indica su dirección.

Un haz 3D vendrá dado por un origen (al tratarse de rayos primarios, coincide con la posición del observador) y tres vectores que definen los límites del haz (`vector_origen`, `vector_lateral`, `vector_vertical`). Uno de ellos nos determina un punto de referencia (el extremo inferior derecho del haz) y los otros dos nos indicarán la amplitud del haz tanto en el sentido horizontal como vertical, como se indica en la figura 2. En función de los tres vectores de referencia podemos determinar aquellos rayos que están contenidos en un haz analizando y comparando los vectores directores de los rayos con los que definen el haz.

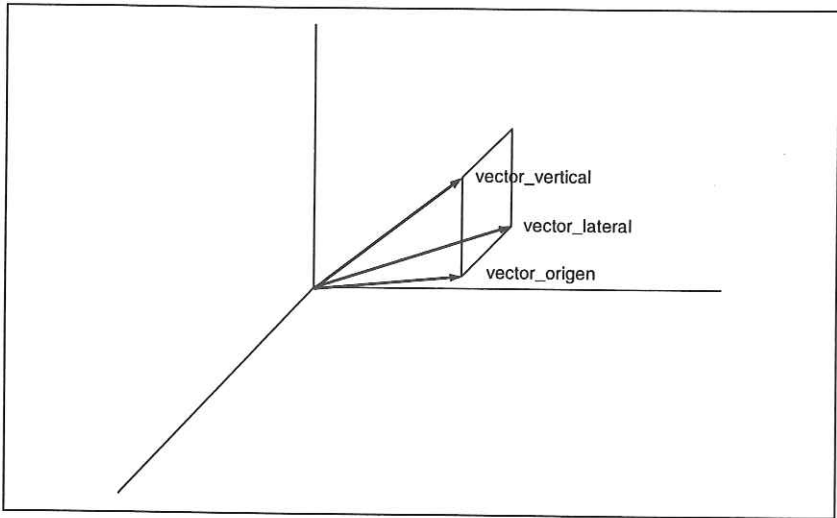


FIGURA 2.
Estructura de un haz 3D.

3.2. Trazado de haces de rayos primarios en escenas descompuestas mediante árboles octales

Como ya hemos apuntado nuestro algoritmo sólo contempla el trazado de haces para rayos primarios y por tanto el resto de rayos secundarios se trazan de modo individual. El tamaño y la forma de los haces está en función de la estructura de descomposición que define la escena. Los haces iniciales se establecen teniendo en cuenta los límites de la pantalla y de los octantes que componen la cara del árbol octal visible desde el observador (figura 1).

Tras la creación de los haces iniciales, la definición del resto de haces es dinámica. Esta estará en función del tamaño y la forma del haz que llega al nodo y de los nodos a los que se dirigirá cuando abandone el nodo actual. De este modo un haz puede descomponerse en otros de salida de menor tamaño que se enviarán a nodos vecinos (ver figura 3).

Para obtener los haces de salida asociados a un haz de entrada primeramente debemos hallar, en función de la dirección del haz inicial, cuál es la dirección o direcciones de salida del mismo. Para cada una de dichas direcciones se creará al menos un haz de salida. Tras ello, debemos recorrer el árbol octal para determinar el nodo o nodos vecinos en una dirección dada, pues el tamaño de dichos nodos vecinos también influirá en el número y tamaño de los haces de salida.

El algoritmo de búsqueda de nodos vecinos que vamos a utilizar se apoya en los trabajos de Samet [SAM89] que permiten, mediante el enlace al padre, recorrer el árbol y determinar el nodo vecino de igual

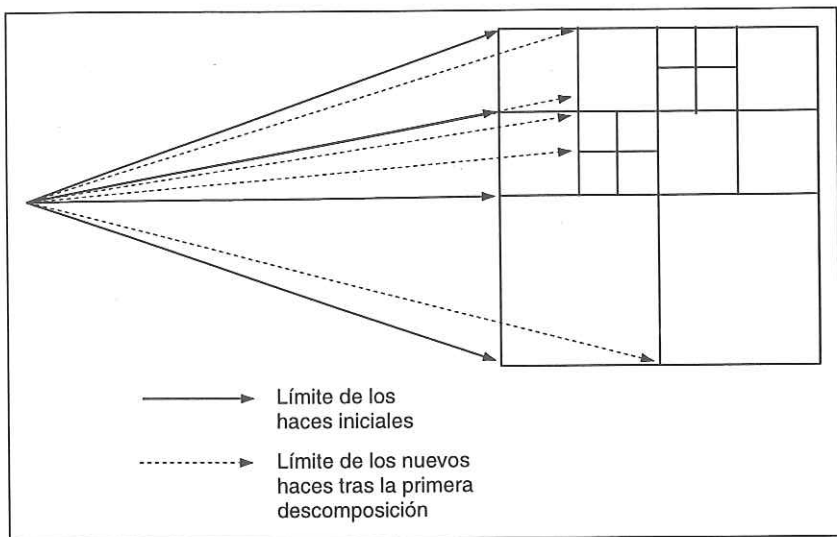


FIGURA 3.
Descomposición de los haces en una escena 2D.

o mayor tamaño en una dirección dada. Dicho algoritmo realiza una ascensión en el árbol hasta encontrar el primer antecesor común tanto al nodo en el que estamos como al nodo al que se dirigirá nuestro haz. Tras ello realiza un descenso tomando como referencia el camino seguido hasta entonces para encontrar dicho antecesor común. De este modo nos situamos en el nodo vecino de mayor o igual tamaño. Si el nodo de igual tamaño encontrado es no terminal (tiene asociado un subárbol de descendientes) debemos seguir la búsqueda de los nodos terminales en la dirección deseada dentro del subárbol.

Cada haz al llegar a un nodo deberá realizar un conjunto de tareas que finalmente nos permitirán determinar el color de los diferentes rayos contenidos en él. Las tareas a realizar dependerán básicamente de las características del nodo al que ha llegado el haz. De este modo:

1. Si el nodo contiene objetos.
 - Se deberán seleccionar, de todos los rayos lanzados desde el observador, aquellos que pertenecen al haz.
 - Calcular para los rayos seleccionados y que en etapas previas no han chocado con ningún objeto, su posible intersección con los objetos contenidos en el nodo. Si un rayo intersecciona con al menos un objeto, su tratamiento coincide con el algoritmo de trazado de rayos básico, es decir, se determina el punto de intersección más próximo al observador y se estudia la necesidad de generar nuevos rayos secundarios (transmitidos y/o reflejados) y rayos de sombra, realizándose los cálculos.

los oportunos para determinar el color del rayo primario. Junto a este tratamiento se etiqueta el rayo como intersectado.

- Si existe algún rayo del haz que no ha chocado con ninguno de los objetos contenidos en el nodo, se calculan los haces de salida en función del haz de entrada y de los nodos vecinos en las direcciones de salida. En este caso los haces de salida pueden contener rayos primarios que no han chocado con ningún objeto y otros que sí lo han hecho y que por tanto están etiquetados.
 - Si todos los rayos del haz están etiquetados como intersectados se finaliza el trazado de dicho haz.
2. Si el nodo está vacío, es decir, no contiene objetos.
 - Se calculan los haces de salida en función de los nodos vecinos y del haz de entrada.
 3. Si dentro de la estructura de árbol en la dirección de salida no existe ningún nodo vecino, esto es, si el haz sale del cubo que contiene a la escena.
 - Se deberán seleccionar, de todos los rayos lanzados desde el observador, aquellos que pertenecen al haz.
 - Asociar a los rayos seleccionados y que en etapas previas no han chocado con ningún objeto, un valor de iluminación ambiente a no ser que la dirección del rayo coincida con la de una fuente de luz. En este último caso se asociará al rayo primario el color de la fuente y una intensidad que puede estar atenuada en función de la distancia.
 - Finalizar el trazado del haz.

Como vemos el trazado de haces de rayos primarios finalizará cuando todos los haces han sido trazados, es decir, cuando todos los rayos en ellos contenidos han sido etiquetados (bien al intersectar con un objeto o al abandonar la escena).

Es necesario apuntar que el resto de rayos secundarios o de sombra se trazan a través del árbol de modo individual. El trazado del resto de rayos secundarios o de sombra se realizará del modo propuesto por Sament [SAM89].

Una vez que se ha determinado el color de los rayos que estaban contenidos en los distintos haces es necesario comprobar si todos los rayos primarios han sido trazados. Debemos fijarnos en aquellos rayos que no pertenecen a ningún haz, ya que no se dirigen al cubo que contiene los objetos de la escena. El tratamiento de estos rayos es similar al propuesto para aquellos rayos contenidos en haces que abandonan la escena.

3.3. Análisis de resultados

En este apartado vamos a mostrar algunos resultados obtenidos con este algoritmo en el trazado de haces sobre escenas 2D, que nos permitan mostrar su potencia. Para analizar el comportamiento de nuestro algoritmo lo hemos comparado con otro que realiza un trazado individual de los rayos (propuesto por Samet [SAM89]). La elección de este algoritmo viene motivada por ser éste uno de los que más frecuentemente se toma como referencia al plantear alternativas al trazado de rayos en escenas estructuradas mediante técnicas de descomposición no uniforme. A su vez, como ya hemos indicado, las ideas de Samet [SAM89] sobre recorrido de árboles octales son utilizadas también dentro del algoritmo que nosotros proponemos.

Hemos utilizado para realizar el análisis de nuestro algoritmo escenas que permitan simular un comportamiento lo más general posible. Para ello primeramente hemos utilizado una escena en la que los objetos se distribuyen aleatoriamente y, por tanto, no encontramos grandes áreas con una alta concentración de objetos. Por otra parte se ha utilizado una versión 2D de las denominadas «SphereFlake» derivadas de una librería de dominio público [HAI87] frecuentemente utilizada para comprobar la eficiencia de las diferentes alternativas de aceleración del trazado de rayos. En este caso, como podemos ver en la figura 4,

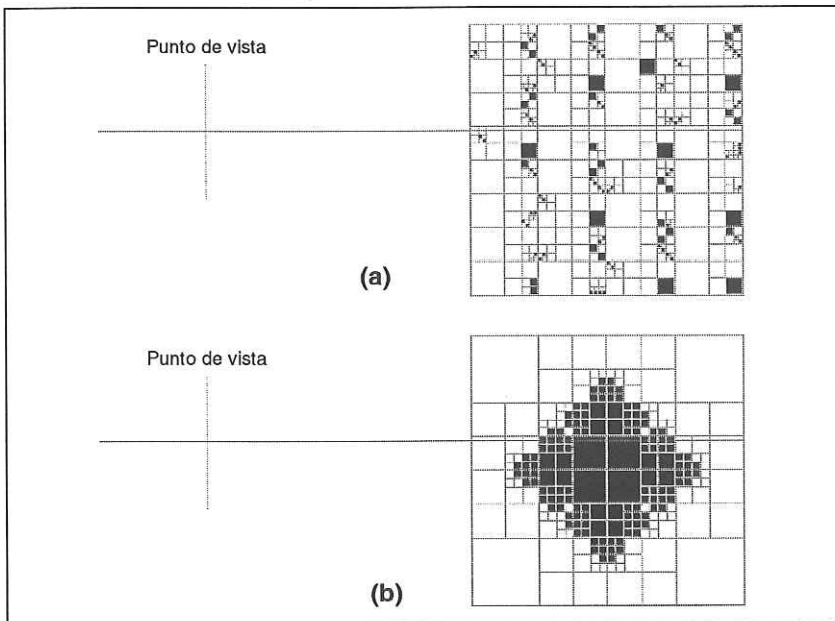


FIGURA 4.
Escenas 2D.

dentro de la escena los objetos están fuertemente concentrados apareciendo grandes zonas totalmente vacías. La primera de las escenas tiene 106 nodos ocupados por objetos y en la segunda dicho número aumenta hasta 140 nodos.

En la figura 5 se muestra el comportamiento de ambos algoritmos para las dos escenas seleccionadas y para distinto número de rayos a trazar. En ella se puede ver como el tiempo de procesamiento con el algoritmo de trazado de haces se mantiene casi invariable independientemente del número de rayos a trazar, mientras que en el otro caso (algoritmo de Samet) se produce un crecimiento constante al aumentar el número de rayos. Por tanto, podemos ver que al incrementar-se el número de rayos a trazar las ventajas de nuestro algoritmo con

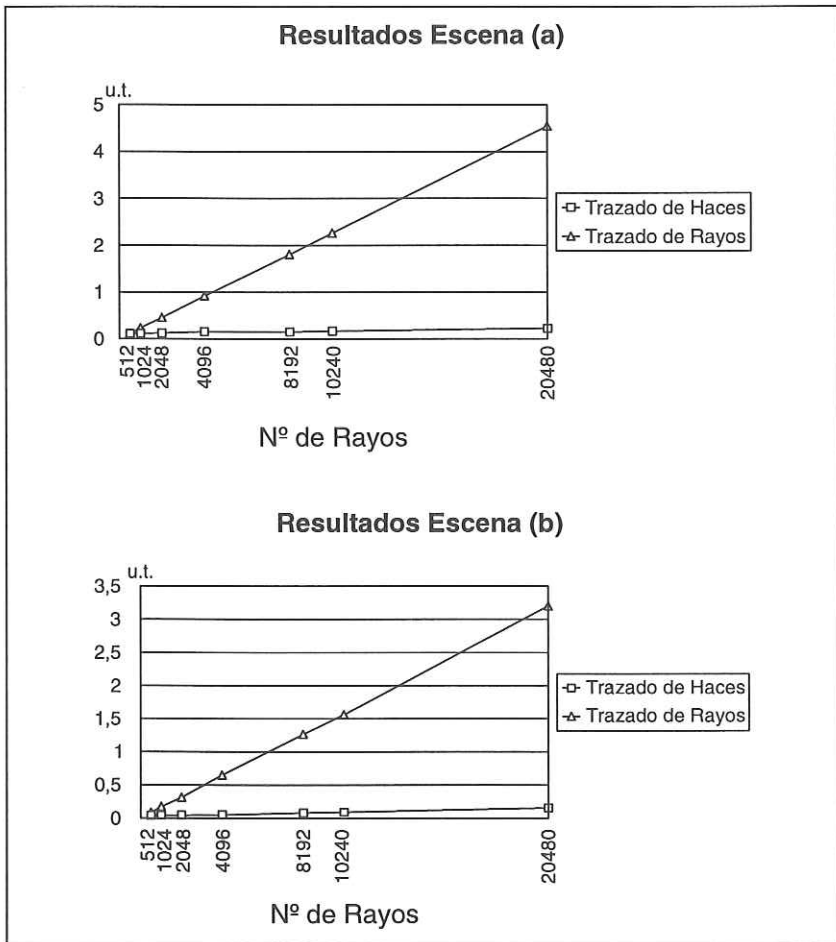


FIGURA 5.
Resultados comparativos.

respecto al otro crecen de forma notoria. Esto es debido a que aumenta el número de rayos que están contenidos en los diferentes haces y que, por tanto, se desplazan juntos dentro de la escena.

Los beneficios de nuestro algoritmo se derivan de que no tenemos que analizar individualmente rayos primarios cada vez que un haz llega a un nodo. Si el nodo que encuentra el haz en su camino no contiene objetos, simplemente debemos calcular el o los haces de salida en función de sus nodos vecinos, sin preocuparnos de determinar los rayos contenidos en los diferentes haces tanto de entrada como de salida. Por tanto, todos los rayos asociados a un haz se mueven dentro de la estructura de árbol octal al mismo tiempo. De ese modo, no es necesario para los rayos que llegan a un nodo determinar los puntos de salida individuales de cada uno, ni tampoco buscar los vecinos en dicha dirección.

Tan solo debemos determinar y analizar individualmente los rayos asociados a un haz cuando deseamos obtener el color final asociado a cada uno. Esto será necesario si el haz llega a un nodo ocupado por objetos o abandona el cubo que contiene a la totalidad de la escena.

Por otra parte, la determinación de los rayos contenidos en un haz se beneficia de la estructura en que almacenamos dichos rayos. Estos se almacenan de modo ordenado en una matriz lo que facilita la utilización de algoritmos de búsqueda eficientes. En nuestro programa se utiliza algoritmos de búsqueda dicotómica que consiguen buenos resultados a la hora de seleccionar dentro de la matriz de rayos aquellos contenidos en un determinado haz.

4. CONCLUSIONES

En este trabajo hemos mostrado una nueva alternativa de aceleración del trazado de rayos que nos permite recorrer más rápidamente una escena descompuesta mediante métodos de descomposición espacial no uniforme. Con este algoritmo conseguimos que los rayos primarios sólo se analicen individualmente cuando debemos comprobar su posible intersección con los objetos almacenados en un octante o cuando salen de la escena. De este modo, al evitar el trazado individual de rayos primarios eliminamos, para cada uno de los rayos, las tareas del cálculo del punto de salida (dirección de salida) y de la búsqueda del nodo vecino en dicha dirección que, a su vez, contenga al punto de salida.

El grado de optimización que se consigue con este algoritmo depende en gran medida del número de rayos que podemos procesar simultáneamente, es decir, del tamaño de los haces que trazamos en la escena y por tanto está fuertemente ligado a la estructura de la misma (nivel de coherencia que presenten los objetos de ésta). Por otra parte,

debido a que el trazado de un haz generalmente implica un proceso de descomposición de éste en otros de menor tamaño, su eficiencia se ve afectada por el tamaño de los octantes que el haz vaya encontrándose en su camino.

Tras esta primera versión del algoritmo los siguientes trabajos que nos planteamos van en dos líneas un tanto diferentes. Por un parte se pretende seguir profundizando en la optimización de los algoritmos propuestos e incorporar el trazado de haces para el resto de rayos generados al chocar un rayo con un objeto. La segunda línea va dirigida a obtener una versión paralela de este algoritmo ya que, al igual que ocurre con el trazado de rayos, nuestra propuesta de procesamiento de los diferentes haces permite que estos puedan ser trazados de forma independiente ofreciendo, por tanto, amplias posibilidades para su paralelización.

5. BIBLIOGRAFÍA

- [ARV89] ARVO, J., KIRK, D.: «A survey of Ray Tracing Acceleration Techniques. En *Introduction to Ray Tracing*, A.S. GLASSNER A.S. ed. Academic Press, 1989, pp. 201-262.
- [END94] ENDL, R., SOMMER, M.: «Classification of Ray-Generators in Uniform Subdivisions and Octrees for Ray Tracing». *Computer Graphics Forum*, vol. 13, nº 1, 1994, pp. 3-19.
- [HAI87] HAINES, E.: «A proposal for Standard Graphics Environments». *IEEE Computer Graphics & Applications*, vol. 7, nº 11, Nov. 1987, pp. 3-5.
- [MCN92] McNEILL, M. D. et al.: «Performance of Space Subdivision Techniques in Ray Tracing». *Computer Graphics Forum*, vol. 11, nº 4, 1992, pp. 213-220.
- [SAM89] SAMET, H.: «Implementing Ray Tracing with Octrees and Neighbor Finding». *Computer & Graphics*, vol. 13, nº 4, 1989, pp. 445-460.
- [SPE86] SPEER, L. R. et al.: «A Theoretical and Empirical Analysis of Coherent Ray-Tracing». *Computer-Generated Images (Proc. Of Graphics Interface '85)*, 1986, pp. 11-25.
- [WHI80] WHITTED, T.: «An Improved Illumination Model for Shaded Display». *Comm of the ACM*, vol 23, nº 6, 1980, pp. 343-349.