

Optimizaciones en estimación de densidades para iluminación global

R. J. García, C. Ureña, M. Lastra, R. Montes y J. Revelles

Departamento de Lenguajes y Sistemas Informáticos. Universidad of Granada

{ruben,almagro,mلاstral,rosana,jrevelle}@ugr.es

Resumen

Este artículo compara empírica y teóricamente varias optimizaciones de estimación de densidades en el plano tangente, que es una técnica para iluminación global. Esta técnica se basa en Photon Maps y da más exactitud si las superficies no son localmente suaves o continuas. La primera de las optimizaciones crea un conjunto de rayos candidatos para cada cálculo de irradiancia. La segunda es una técnica nueva que se presenta en este artículo y que usa indexación espacial de los discos alrededor de los puntos en que se calcula la irradiancia. Se ha realizado un estudio analítico del orden de complejidad de los algoritmos y uno práctico del tiempo de cálculo para distintos valores de los parámetros de los algoritmos. Por último, se dan reglas que permiten elegir la optimización más apropiada para cada caso.

1. Introducción

El rendimiento de la estimación de densidades depende de calcular rápido qué rayos contribuyen a una zona de la escena. Las técnicas actuales usan esquemas de partición del espacio para conseguir velocidades interactivas. La mayoría de las técnicas indexan los rayos para calcular estimación de densidades en los puntos de la escena. Nuestra aportación, basada en estimación de densidades en el plano tangente, permite usar cualquier técnica de partición espacial para indexar las zonas que afectan a cada punto en el que se estima la densidad de energía, obteniendo mejor rendimiento que las técnicas existentes cuando las zonas son relativamente pequeñas.

En estimación de densidades los rayos normalmente se incluyen en una indexación espacial para poder encontrar de forma rápida qué rayos afectan a una posición dada. La técnica Photon Maps[11] indexa los impactos de los fotones usando un kd-tree.

La caché de esferas[13] crea una lista de esferas que contienen rayos. Havran usa un kd-tree construido bajo demanda para indexar los rayos[7].

En ray tracing la mayoría de las aplicaciones crean un índice espacial para los objetos de la escena[3, 6].

Nuestra aportación consiste en aplicar una indexación espacial de los discos usados en Estimación de densidades en el plano tangente.

Existen trabajos que aplican métodos analíticos a algoritmos de iluminación global. En [8] se usan métodos estadísticos para estudiar diferentes técnicas de indexación espacial para ray tracing y presenta un marco que facilita la comparación de las diferentes técnicas. La medida de las líneas usada en el estudio teórico de este artículo se usó también en [2].

El artículo se desglosa en las siguientes secciones: la sección 2 describe las técnicas de estimación de densidades más comunes, comenzando con técnicas simples cuyas limitaciones se resuelven con algoritmos más elaborados. Finalmente nuestros algoritmos se describen con detalle. Se presenta una nueva optimización para estimación de densidades en el plano tangente basada en indexación de discos que consigue mejorar el rendimiento cuando la distancia entre puntos en los que se calcula la irradiancia es del orden de magnitud del núcleo de la función de estimación de densidades. Sin embargo, la mayor ventaja de la técnica es que cualquier método de indexación espacial puede aplicarse trivialmente en este esquema. La sección 3 contiene una comparación empírica de la eficiencia en tiempo entre las distintas optimizaciones de estimación de densidades en el plano tangente, usando una escena de prueba. La sección 4 muestra un estudio de la eficiencia en tiempo de los distintos algoritmos. Las probabilidades de intersección rayo-objeto se usan para calcular el va-

lor esperado del tiempo de los distintos algoritmos, y se deriva un orden de eficiencia en notación $O()$ para cada algoritmo. Se asume una distribución uniforme de los rayos. Finalmente se resumen las conclusiones más importantes y el trabajo futuro.

2. Métodos de estimación de densidades

Para obtener un valor de irradiancia en un punto, hay que integrar la radiancia incidente para todas las direcciones de entrada.

El método más básico se describe en [4] y en [16] donde se encuentran los impactos de los fotones en los triángulos y se calcula la densidad de energía en el triángulo. Los vértices tienen una irradiancia que es el promedio de la de los triángulos a los que pertenecen, y se interpola linealmente en el triángulo.

Un método útil para estimar la integral es el método de estimación de densidades, popularizado por [23]. Tiene tres fases. La primera fase se basa en el modelo de partículas de la luz, y simula fotones desde las fuentes de luz. La segunda (estimación de densidades propiamente) estima la irradiancia. La tercera fase simplifica la geometría tras el cálculo de iluminación. Esta última fase a menudo se obvia porque se considera fuera del ámbito de estimación de densidades.

Jensen [12] diseñó el conocido método de Photon Maps. Consiste en encontrar los n fotones más cercanos al punto donde se estima la irradiancia, sumar su energía y dividir por el área del círculo máximo de la esfera que contiene los n fotones (n está predefinido).

Algunos excelentes algoritmos para iluminación global en tiempo real, como [5] y [22], son ray-tracers rápidos distribuidos que usan Photon Maps para terminar la recursividad.

La limitación más conocida de Photon Maps es que cuando se calcula la irradiancia en un punto, los fotones cercanos deberían estar en el mismo plano y en un área circular alrededor del punto. [10] presenta un algoritmo que resuelve esta limitación usando información geométrica en las cercanías del punto.

Otra limitación menos conocida de Photon Maps y [10] es que si hay superficies en la escena relativamente muy pequeñas, estas zonas tienen comparativamente una alta varianza, y tienden a aparecer muy brillantes o muy oscuras, si el número de fotones no es lo suficientemente alto. Véase la figura 1 (izquierda).

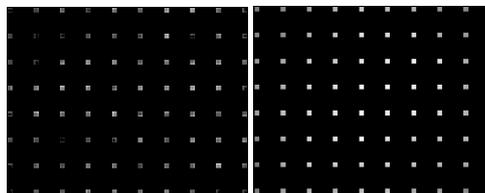


Figura 1: Photon Maps (izquierda) y DETP (derecha)

2.1. Estimación de densidades en el plano tangente

Un método que evita la alta varianza de Photon Maps mencionada en la sección anterior, consiste en guardar los rayos y usar un disco de tamaño fijo centrado en el punto donde se calcula la irradiancia y que está contenido en el plano tangente a la superficie. Los rayos que intersecten un disco dado se usan para calcular la irradiancia en el punto central [13]. El algoritmo se llama estimación de densidades en el plano tangente (DETP). Nótese que este algoritmo guarda la trayectoria de los fotones (origen, dirección y punto de impacto), a diferencia de Photon Maps. Véase la Figura 2. Para evitar sombras propias en superficies cóncavas, se usa la segunda intersección del rayo con la escena en lugar de la primera. Este método usa discos de radio fijo [13]

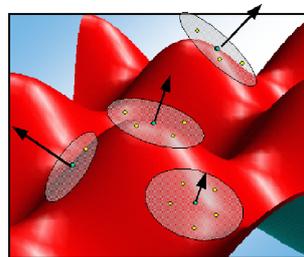


Figura 2: Estimación de densidades en el plano tangente

El algoritmo compensa de forma óptima entre exactitud y varianza cuando el radio del disco es del orden de magnitud de la mitad de la distancia entre muestras de irradiancia. Si el radio del disco fuera más pequeño, los rayos que intersecan el plano tangente cerca del punto medio entre dos muestras de irradiancia se ignorarían. Si fuera más grande, las intersecciones se usarían para varios cálculos, creando un suavizado artificial.

2.2. Caché de esferas

La limitación de DETP es que el número de intersecciones rayo-disco es alto, por lo que aumenta el tiempo de cómputo. Para evitar esto, es posible usar la caché de esferas [13].

La caché de esferas consiste en crear una jerarquía de esferas de radio decreciente y guardar los rayos que intersecan cada esfera para disminuir el número de tests de intersección rayo-disco.

Primero, se crea una esfera tangente a la caja englobante de la esfera. Esta esfera interseca todos los rayos.

Después, como se ve en la figura 3, se crean esferas de radio decreciente unas dentro de otras (la razón entre el radio de dos esferas consecutivas es un parámetro llamado Q), hasta que el radio está justo por encima del radio del disco.

Cada esfera tiene una estructura de datos asociada que contiene los rayos que interseca dicha esfera. Estos rayos se calculan intersecando la esfera con los rayos de la esfera inmediatamente superior.

El primer punto en el que se ha de calcular la irradiancia se usa como el centro de las esferas. Por tanto, el primer disco está en la esfera interna. La irradiancia se puede calcular comprobando qué rayos de la esfera interna intersecan el disco, y sumando su energía. El número de intersecciones rayo-disco se reduce claramente.

Para el resto de puntos, si el disco centrado en el punto está dentro de la esfera interna, el disco se interseca con los rayos de esta esfera. En caso contrario, se descarta la esfera y se comprueba el resto de esferas hasta que una contenga el disco. En ese momento se recalcula la jerarquía de esferas, usando el punto como centro. Véase la figura 3 derecha.

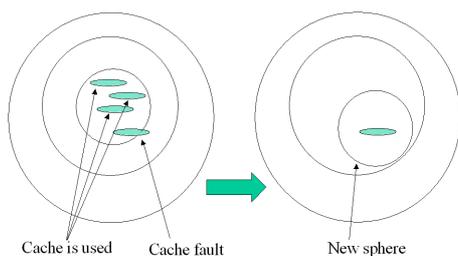


Figura 3: Caché de esferas

Finalmente, el disco se interseca con los rayos de la esfera interior, del mismo modo que si no se necesita recalculer ninguna esfera.

Ordenación de puntos. El rendimiento de la caché de esferas depende de la coherencia espacial de los discos, ya que esta coherencia permite reusar las esferas. [13] contiene un algoritmo que reordena los puntos para incrementar la coherencia espacial.

El algoritmo transforma las coordenadas de los puntos en tres enteros sin signo de 16 bits, de modo que la caja englobante de la escena va de 0 al máximo valor representable en cada coordenada. Los puntos se ordenan con la siguiente función de comparación: Para dos puntos A y B, se comprueba el bit más significativo (MSB) de la coordenada X de A con la de B, después el MSB en Y y el Z. Después se comprueban los siguientes bits, en orden. En cuanto uno de los bits sea menor, el punto se considera menor.

2.3. Indexación de discos

La técnica de indexación de discos crea una indexación espacial con los discos de la escena. Posteriormente, los rayos atraviesan el índice espacial añadiendo su contribución a los discos que intersecan. Véase la figura 4. El rayo sólo ha de seguirse hasta la primera intersección con la escena real (o la segunda si hay superficies cóncavas). El índice es-

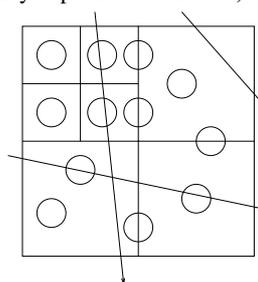


Figura 4: Indexación de discos

El índice espacial debe ser capaz de guardar discos y calcular eficientemente todas las intersecciones con un segmento (los extremos del segmento son el origen del rayo y la intersección con la escena real). Todos los algoritmos publicados pueden hacerlo.

Hay estudios de las características de la escena que hacen que unas indexaciones espaciales sean más eficientes que otras[8]. Otros estudios usan una simulación rápida con pocos rayos para elegir el método de indexación más apropiado[9, 18]. Ya que la posición de los discos sigue la superficie de los objetos, esta investigación es aplicable a esta técnica también.

Este método tiene más rendimiento que la caché de esferas cuando los discos tienen un radio que es del orden de magnitud de la distancia media entre puntos en que se calcula la irradiancia. En otras situaciones, la caché de esferas es mejor. Los detalles están en la sección 3.

La profundidad máxima de la indexación espacial debe ajustarse al cambiar el tamaño del disco, ya que al crecer los discos, se cortan entre sí. Dividir un voxel crea voxels donde la mayoría de los discos pertenecen a todos los voxels hijos. Esto hace que el tiempo de intersección con los hijos sea mayor que con el voxel original.

Para ilustrar el problema, imagine que queremos calcular la irradiancia en cada uno de los vértices de la malla de la figura 5 (recuerde sin embargo que DETP es independiente de geometría). Hay un disco centrado en cada vértice de la escena. Al crecer el disco, se nota que una partición del espacio no ayuda.

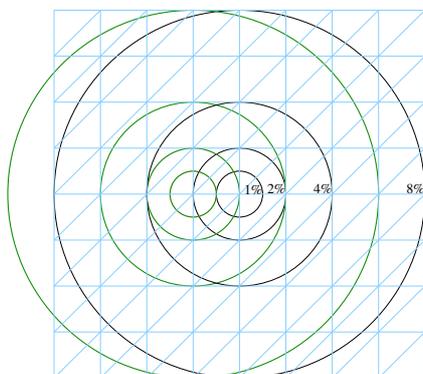


Figura 5: Discos en una malla típica

3. Comparación empírica entre la caché de esferas e indexación de discos

Todos los algoritmos descritos en las secciones anteriores han sido implementados en nuestro sistema de rendering. El sistema calcula la irradiancia en los vértices de la escena, aunque DETP puede calcular la irradiancia en cualquier punto de la superficie.

La primera escena de prueba puede verse en la figura 6. Tiene 72 500 triángulos, y la llamaremos la primera escena con árbol.

La segunda escena con árbol es un tipo distinto de árbol y suelo, con triángulos mayores (Figura 8).

Usaremos un árbol BSP alineado a los ejes[20] en los primeros ejemplos de la sección, y un octree en la comparación siguiente, más exhaustiva. Se eligieron esas técnicas porque dan buen rendimiento en la fase de fotosimulación.

Los resultados de tiempos para el árbol BSP se ven en la figura 7. En este artículo, todos los tiempos están en segundos.



Figura 6: Primera escena con árbol, 10 000 fotones

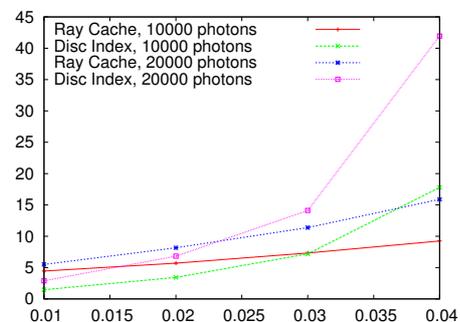


Figura 7: Tiempo en segundos del cálculo de irradiancia usando la caché de esferas y la indexación de discos para la primera escena, en función del radio del disco

La longitud de las aristas de los triángulos del suelo es el 2 % de la longitud de la escena, y las del árbol el 1 %. Es fácil ver que para tamaños de disco por debajo del 4 % de la escena, la indexación de discos es más rápida que la caché de esferas.

Para 20 000 fotones, los resultados sólo son mejores para tamaños del 1 % y 2 %.

Esta gráfica muestra cómo el rendimiento de la indexación de discos disminuye al aumentar el área de los discos. En la segunda escena, los resultados son mejores usando discos más grandes ya que el tamaño de los triángulos ha aumentado.



Figura 8: Segunda escena con árbol, 20 000 fotones

Para discos grandes, una división espacial muy fina no es útil, ya que hace que los rayos intersequen repetidamente con cada disco al atravesar la estructura, como se vio en la figura 5.

Para comparar las técnicas con más detenimiento, se hicieron una serie de pruebas con la primera escena. Se midió el tiempo usando las distintas combinaciones de radio del disco de DETP (entre 1% y 16%), profundidad máxima del octree (entre 5 y 8 niveles) y número de fotones (entre 100 y 409 600).

Los tiempos son lineales para la indexación de discos con octrees de distintas profundidades. La caché de esferas también es lineal, aunque el tiempo es ligeramente mejor que lineal cuando hay pocos rayos, debido a que los rayos de la esfera interna caben en la caché del procesador. De acuerdo con los experimentos realizados, la diferencia en rendimiento entre la caché de esferas y la indexación de discos depende básicamente de la razón de la distancia entre discos ($\overline{\Delta x}$) al radio del disco (d). Si llamamos X a este número, $X = \frac{\overline{\Delta x}}{d}$ y recordando que el rendimiento de los algoritmos cambia suavemente con este número y por tanto las divisiones son difusas, se pueden reconocer tres regiones.

- La primera corresponde a $X \leq 2$, en la que la indexación de discos es mejor que la caché de esferas. La diferencia es más importante cuanto más pequeño sea el radio del disco.
- La segunda corresponde a $X \geq 6$, en la que la situación es la contraria.
- La tercera es la situación intermedia $2 < X < 6$. En este caso, si los rayos de la esfera interna caben en la caché del procesador, la caché de esferas es más rápida por la coherencia de este algoritmo. En caso contrario, la indexación de discos es más rápida, ya que el rendimiento de la caché de esferas para cada rayo disminuye.

La figura 9 da el rendimiento de las técnicas para cada caso. En una tercera escena más compleja, que puede verse en la figura 10, la indexación de discos obtuvo mejoras en tiempo hasta de un 50%. En esta escena, la distancia media entre vértices es de $0.01\hat{3}$ y el radio del disco 0.01 ; $\overline{\Delta x}/d=1.\hat{3}$, por tanto estamos en la primera zona.

Como resumen, es de destacar que el rendimiento de la indexación de discos baja más rápido que el de la caché de esferas cuando crece el tamaño del disco

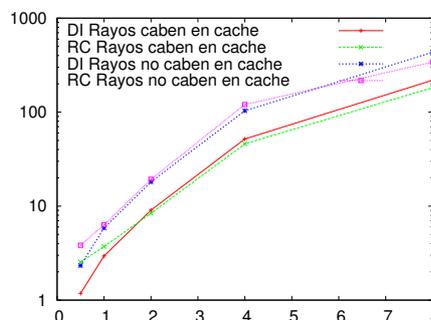


Figura 9: Zonas en que cada técnica es óptima. Tiempo en función de $\frac{\overline{\Delta x}}{d}$



Figura 10: Patio grande (122 318 triángulos)

y que la memoria usada por la indexación de discos crece muy rápido al crecer el tamaño del disco, lo que impide usarlo para tamaños grandes de disco y que para radios muy pequeños, la indexación de discos con la mayor profundidad siempre es mejor que la caché de esferas. También hay que tener en cuenta que se debe disminuir la profundidad de la indexación espacial al crecer el radio del disco.

4. Estudio teórico de la caché de esferas y la indexación de discos

Estos algoritmos tienen la misma técnica de estimación de densidades (DETP). Por tanto, para cualquier escena y conjunto de rayos, calcularán la misma solución (aunque tardarán tiempos diferentes). En este caso, los algoritmos pueden compararse calculando el tiempo de cómputo en función del tamaño de la escena (número de discos y tamaño) y del número de rayos. De los cuatro algoritmos (DETP básica, caché de esferas, ordenación de puntos y indexación de discos), sólo se usan la ordenación de puntos y la indexación de discos ya que tienen mayor rendimiento. La notación usada se resume en la Tabla 1. La eficiencia de DETP básica

claramente es $O(n_R n_P)$. Para la caché de esferas sin ordenación de puntos hemos de suponer una distribución uniforme de puntos para calcular la irradiancia. Esto hace que la caché de esferas no obtenga mejoras en el tiempo. Se puede demostrar que el algoritmo es $O(n_R n_P)$, con una constante oculta ligeramente superior a la unidad. La eficiencia de los otros dos algoritmos se discute en detalle en las siguientes secciones.

Para extender la comparación a otras técnicas de estimación de densidades (como Photon Maps), se debe añadir un estudio de varianza y error, y comparar el tiempo de cómputo en función del error o la varianza.

Un estudio asintótico del rendimiento de los algoritmos en el caso promedio es útil para probar la escalabilidad. Las variables del estudio son el número de cálculos de irradiancia (complejidad de la escena) y el número de rayos (complejidad de la iluminación).

Para calcular el tiempo promedio de los algoritmos mencionados en las secciones anteriores es necesario una estimación del número de intersecciones rayo-disco, que dependen de la distribución de los rayos. Supondremos una distribución uniforme de rayos en el espacio, y calcularemos la medida del conjunto de rayos que intersecan un volumen finito. Esta suposición no realista es esencial para obtener fórmulas matemáticas para la probabilidad de intersección. Es una suposición común [1, 2], también usada implícitamente en [17], que funciona bien en la práctica.

Para integrar, necesitamos una medida para el espacio de rayos que sea invariante bajo rotación y translación. Esto significa que todas las posiciones y direcciones son igualmente importantes. La medida se llama μ_ℓ en [2] y permite definir la densidad de los rayos; se usa implícitamente en las siguientes secciones.

4.1. Tiempo promedio usando la caché de esferas con ordenación de puntos

Para calcular la fracción promedio de rayos que intersecan una esfera, usaremos algunos resultados de geometría integral. La sección III.14.2 del libro de Santaló [19] tiene fórmulas útiles.

Sean K_0, K_1 conjuntos convexos en E_n (el espacio euclídeo de dimensión n) tal que $K_1 \subset K_0$. La probabilidad de que un r -plano L_r ($r=1, \dots, n-1$), elegido de acuerdo con una distribución aleatoria

Cantidades relacionadas con rayos

R	Conjunto de rayos
$n_R = \#R$	Número de rayos

Cantidades relacionadas con esferas

$S = \{S_i\}$	Conjunto de esferas
r_i	Radio de S_i
$r_i = r_{i-1}Q = r_0Q^i$	
$V_i = \frac{4}{3}\pi r_i^3$	Volumen de S_i
m_i	Número de recálculos de la esfera S_i con ordenación de puntos.
k	Número de esferas

Cantidades relacionadas con el tiempo

u	Tiempo de intersección Rayo-Disco
t	Tiempo de intersección Rayo-Esfera
T_R	Tiempo para recalcular las esferas con ordenación de puntos
T_I	Tiempo de intersección disco-esfera interna

Otros símbolos

$0 < Q < 1$	Razón de radios entre dos esferas
$P = \{P_i\}$	Conjunto de muestras de irradiancia
$n_P = \#P$	Número de muestras de irradiancia
d	Radio del disco

Cuadro 1: Símbolos usados en el artículo invariante bajo rotación y translación, que corta a K_0 también corte a K_1 es:

$$p(L_r \cap K_1 \neq \emptyset) = \frac{M_{r-1}(\partial K_1)}{M_{r-1}(\partial K_0)} \quad (1)$$

donde ∂K es el contorno de K y M_{r-1} son las integrales de curvatura media.

Especializando para $n = 3$ (espacio tridimensional), siendo K_0 la primera esfera, K_1 la esfera interior y $r = 1$ (hiperplanos de una dimensión: líneas) la probabilidad resultante es:

$$p(L_0 \cap K_1 \neq \emptyset) = \frac{M_0(\partial K_1)}{M_0(\partial K_0)} \quad (2)$$

Para cuerpos convexos K de E_3 , las integrales de curvatura media son $M_0 = F$ (área de ∂K), como dice la sección III.13.6.(g) de [19]. Por tanto,

$$p(L_0 \cap K_1 \neq \emptyset) = \frac{F_1}{F_0} = \frac{4\pi r_1^2}{4\pi r_0^2} \quad (3)$$

Si la distribución de rayos es uniforme, el número de rayos que interseca la esfera interior es independiente de la posición de la misma, y es proporcional al cociente del cuadrado del radio de las esferas.

El número de rayos que intersecan S_i es entonces:

$$n_i = n_R \frac{r_i^2}{r_0^2} = n_R Q^{2i} \quad (4)$$

El coste de recalculer la esfera S_i una vez es el número de rayos en la esfera envolvente por el tiempo de intersección rayo-esfera.

$$t_i = tn_{i-1} \quad (5)$$

donde t es el tiempo de intersección rayo-esfera. La esfera 0 nunca se recalcula, porque todos los discos están contenidos dentro.

Ahora averigüemos cuántas esferas hay. Las esferas se crean con un radio decreciente hasta que el radio de la esfera está justo por encima del radio del disco (i.e. el radio de la siguiente esfera sería menor que el del disco).

Sea k el número de esferas. Para calcular k , usaremos d como el radio del disco. Recordemos (Sección 2.2) que el cociente entre radios de dos esferas adyacentes es Q , y que las esferas se construyen hasta que su radio está justo por encima del radio del disco d . k debería por tanto satisfacer las siguientes ecuaciones:

$$r_k = Q^k r_0 \geq d \quad ; \quad r_{k+1} < d \quad (6)$$

Por lo tanto k se puede calcular así:

$$k = \left\lceil \log_Q \left(\frac{d}{r_0} \right) \right\rceil \quad (7)$$

El coste de intersecar el disco con los rayos de la esfera interna, T_I , es:

$$T_I = un_k n_P = un_R Q^{2k} n_P = \quad (8)$$

$$un_P n_R Q^{2 \lceil \log_Q(\frac{d}{r_0}) \rceil} \lesssim un_P n_R \frac{d^2}{r_0^2} \quad (9)$$

Dado que la caché de esferas no es útil a no ser que la localización de los puntos en que se calcula la irradiancia sean coherentes, una curva que rellena el espacio se usa para ordenar los puntos. Este algoritmo usa la curva de Lebesgue.

El algoritmo de ordenación divide el cubo en 2^{48} celdas. Cada celda tiene una coordenada X , Y y Z con 16 bits cada una. Estas 3 coordenadas se concatenan para formar un número de 48 bits. Una función reordena los bits del número de la siguiente forma: los 3 bits menos significativos de cada coordenada corresponde con los 3 bits menos significativos del nuevo número, y se repite el proceso hasta llegar al bit más significativo. Un ejemplo con 3 bits por coordenada sería:

- Original: $x_2 x_1 x_0 y_2 y_1 y_0 z_2 z_1 z_0$
- Reordenado: $x_2 y_2 z_2 x_1 y_1 z_1 x_0 y_0 z_0$

Ahora las celdas se visitan desde 0 hasta $2^{48} - 1$. Cambiar los tres bits menos significativos significa mover el objeto como máximo la diagonal de las celdas, $2 * \sqrt{3} * 2^{-16}$, y como mínimo la arista de las celdas, $2 * 2^{-16}$. Una esfera con un radio de ese orden de magnitud se recalcularía aproximadamente 2^{48} veces. Es fácil ver que una esfera de radio $2^{-m} r_0$ se recalcula 2^{3m} veces. Por tanto, ya que:

$$r_i = Q^i r_0 = 2^{-m} r_0 \quad (10)$$

$$m = -\log_2 Q^i \quad (11)$$

la esfera S_i se recalcula $m_i = 2^{3 * (-\log_2 Q^i)} = Q^{-3i}$ veces.

Este es el máximo de veces que se reconstruye la esfera, y es independiente del número de puntos en que se calcula la estimación de densidades. Hay un límite independiente del máximo número de veces, dado por el número de puntos de irradiancia. La esfera S_i se recalcula como máximo n_P veces. El coste de recálculo en este caso es:

$$T_R = \sum_{i=1}^k \min(n_P, m_i) t_i \quad (12)$$

y el coste total es:

$$T = T_R + T_I \quad (13)$$

T puede usarse para averiguar qué Q debe usarse. Vea las figuras 11, 12 y 13. La figura 11 muestra el coste de recalculer las esferas por rayo. (i.e. 1 significa n_R intersecciones, equivalente a no optimizar), en función del radio del disco y Q . Un Q pequeño crea pocas esferas, así que el tiempo es pequeño. El radio del disco influye en la longitud de la lista de esferas, pero las esferas más pequeñas tienen muy pocos rayos y por tanto añaden poco tiempo, así que las diferencias en tiempo son pequeñas.

La figura 12 da una gráfica del tiempo de intersección de los discos con los rayos de la esfera interna. Este tiempo sólo depende del radio de la esfera, que depende de Q : el radio es r_k tal que $r_k = r_0 Q^k \geq d$ y $r_{k+1} < d$. La gráfica muestra que hay un infinito número de Q óptimos. Sin embargo, si la inestabilidad numérica hace que sea difícil obtener los valores mínimos, Q debería estar tan cerca como sea posible de 1.

Un Q alto da esferas internas que envuelven firmemente a los discos, aunque el número de recálculos es alto. Un Q bajo hace que los discos tengan más espacio alrededor; habrá muchos rayos en

la esfera interior y aumentará el tiempo de cálculo. Los picos de la gráfica corresponden a esferas internas que envuelven perfectamente a los discos, i.e., $d = r_k$. Los mínimos locales de la función tienen el mismo valor, que se corresponde con el tiempo para intersecar a los rayos en una esfera de radio d : $un_R \frac{d^2}{r_0^2}$.

La figura 13 muestra la integración de las dos gráficas anteriores. Tiene forma de U, como era de esperar de las gráficas componentes, y muestra cómo cambiar el radio del disco hace que el valor óptimo de Q cambie suavemente para crear una esfera interna que envuelva el disco. Cuando Q se acerca a 1, el número de esferas aumenta exponencialmente y el tiempo sube asintóticamente a infinito en $Q = 1$, que significa un número infinito de esferas.

La figura muestra que 0,6 – 0,7 es un buen compromiso, porque está por debajo de 1 en la figura 11 y cerca del punto donde la pendiente se vuelve importante, y el valor en la gráfica de la figura 12 también es bastante bajo. El resultado es coherente con experimentos prácticos[13].

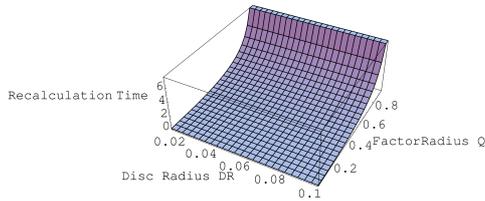


Figura 11: Número de recálculos de la caché de esferas en función de Q y el radio del disco

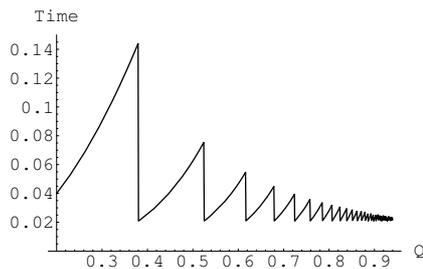


Figura 12: Tiempo de intersección del disco con los rayos de la esfera interna, en función de Q

Se vio antes que para un n_P lo suficientemente grande, el número de recálculos es fijo y el tiempo sólo depende de n_R .

El tiempo para intersecar los discos en la esfera interna, por el contrario, depende tanto de n_R como

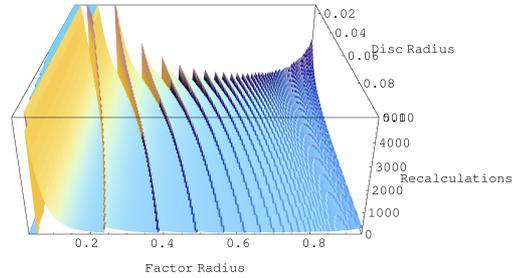


Figura 13: Tiempo de recálculo de la caché de esferas en función de Q y el radio del disco

de n_P y es por tanto $O(n_R n_P)$. La constante oculta, $\frac{d^2}{r_0^2}$, puede hacer que este algoritmo sea bastante eficiente en la práctica.

Si el radio del disco es aproximadamente igual a la distancia entre muestras de irradiancia, que es deseable de acuerdo con la sección 2.1, se puede probar que la eficiencia del algoritmo aumenta.

Supongamos que el radio del disco es aproximadamente igual a la distancia entre muestras.

Para n_P pequeño (para no llegar al límite de recálculos), ya que el radio del disco es aproximadamente la distancia entre muestras,

$$k \approx \log_Q \left(\frac{1}{\sqrt[3]{n_P}} \right) = \log_{\bar{Q}} \sqrt[3]{n_P} \quad (14)$$

donde $\bar{Q} = \frac{1}{Q}$, ya que $0 < Q < 1$.

Ahora, para calcular el orden de eficiencia del algoritmo, se muestra el valor de algunas cantidades importantes, y finalmente se expande el tiempo del algoritmo completo y se calcula la eficiencia. El valor de n_i , el número de rayos en la esfera i :

$$n_i = n_R Q^{2i} \quad (15)$$

El tiempo para recalcular la esfera i es:

$$t_i = t n_R Q^{2i-2} \quad (16)$$

El número de recálculos de la esfera i :

$$m_i = Q^{-3i} \quad (17)$$

El tiempo para recalcular la esfera i para todo el algoritmo es:

$$m_i t_i = t n_R Q^{-i-2} \quad (18)$$

Finalmente, el tiempo del algoritmo completo es por tanto:

$$\sum_{i=1}^k m_i t_i = \frac{t n_R}{Q^2} \left(\frac{\sqrt[3]{n_P} - 1}{1 - Q} \right) = O(n_R \sqrt[3]{n_P}) \quad (19)$$

4.2. Indexación de discos

El diámetro del disco es un tamaño mínimo útil para los voxels en la técnica de indexación de discos. Ya que la indexación de discos se puede usar con distintas técnicas de indexación espacial, el orden de complejidad depende de la técnica usada. Sin embargo, en [17] se prueba que el orden de complejidad es el mismo para un grid, un árbol binario y un octree. En [21] se estudia la complejidad media de ray tracing para otras técnicas. La evidencia sugiere que la mayoría de las técnicas de indexación tienen el mismo orden de eficiencia en el caso promedio. Usaremos un octree como técnica de indexación en nuestro análisis dadas sus prestaciones en escenarios genéricos.

El tamaño relativo del lado de un voxel a profundidad k con respecto al lado de la escena completa es 2^{1-k} . Si fijamos el diámetro del disco como lado del voxel más pequeño, obtenemos:

$$2^{1-k} = 2d \quad (20)$$

$$k = \lfloor -\log_2 d \rfloor \quad (21)$$

Hay por tanto 8^k voxels. Una distribución uniforme de las muestras de irradiancia significa que hay $n_P/8^k$ muestras por voxel, y por tanto, el tiempo de intersección entre un rayo y un voxel es $n_P u/8^k$. Como cada rayo atraviesa una línea de voxels (2^k voxels) y el origen se encuentra recorriendo el árbol (k pasos), el tiempo promedio para este método es:

$$T = u k n_R n_P / 4^k \quad (22)$$

Si el tamaño de voxel es similar al del disco, $k = O(\log_2 \sqrt[3]{n_P})$. El algoritmo es por tanto $O(n_R \sqrt[3]{n_P} \log n_P)$.

Si usamos una estructura de partición espacial balanceada, de acuerdo con [1], llegar a un nodo vecino puede hacerse en $O(1)$. Entonces el rendimiento de la indexación de discos queda $O(n_R \sqrt[3]{n_P})$.

La eficiencia es mayor que el $O(n_R n_P)$ de la caché de esferas para n_P altos. Para n_P pequeños, en que la distancia entre muestras de irradiancia es similar al radio del disco, la eficiencia es la misma que la de la caché de esferas.

5. Conclusiones

De acuerdo con el estudio empírico de las técnicas, la razón entre la distancia media entre vértices y el radio de los discos define zonas en las que los algoritmos son adecuados. La indexación de discos es óptima si esta razón es menor que 2; la caché de

esferas es óptima si la razón es mayor que 6. En la zona intermedia la caché de esferas es mejor si los rayos de la esfera interna caben en la caché del procesador. Un algoritmo óptimo universal calcularía esta razón y el número medio de rayos en la esfera interna y elegiría la mejor técnica en función de estos datos.

Los resultados del estudio teórico son la demostración de que DETP básica y la caché de esferas son $O(n_R n_P)$, y que la caché de esferas con ordenación de puntos, aunque $O(n_R n_P)$, tiene una constante oculta proporcional a $\frac{d^2}{r_0^2}$ (la fracción de rayos en una esfera cuyo radio es el de los discos), que hace que el algoritmo sea bastante rápido en la práctica. Para escenas pequeñas, este último algoritmo puede probarse que es $O(n_R \sqrt[3]{n_P})$. También se ha demostrado que el factor de radios debe estar entre 0.6 y 0.7.

Indexación de discos es $O(n_R \sqrt[3]{n_P} \log n_P)$ para árboles no balanceados y $O(n_R \sqrt[3]{n_P})$ para balanceados.

6. Trabajo futuro

Planeamos usar el benchmarking BART [15] en el estudio empírico para estudiar la frontera entre las distintas zonas de utilidad de cada algoritmo. Hay que hacer más estudios de otros algoritmos relacionados con estimación de densidades en el plano tangente, como Radio Variable [14], que permite que el tamaño de los discos cambie en función del número de rayos en las cercanías (como hace Photon Maps). Los mapas de rayos para iluminación global propuestos por Havran et al [7] es otra optimización independiente que merece la pena estudiar. El estudio que se propone en este trabajo debería usarse como base para estimación automática de parámetros en DETP. Finalmente se deberían estudiar modelos de distribución de muestras de irradiancia y rayos más realistas.

7. Agradecimientos

Este trabajo ha sido financiado dentro del proyecto TIN2004-07672-C03-02, cofinanciado con fondos FEDER. Agradecemos a Miguel Vega su ayuda en el estudio teórico.

Referencias

- [1] B. Aronov, H. Brönnimann, A. Y. Chang, and Y. Chiang. Cost prediction for ray shooting. In

- SCG '02, pages 293–302. ACM Press, 2002.
- [2] B. Aronov, H. Brönnimann, A. Y. Chang, and Y. Chiang. Cost-driven octree construction schemes: An experimental study. In *SCG '03*, pages 227–236. ACM Press, 2003.
- [3] J. Arvo and D. Kirk. *A Survey of Acceleration Techniques*, chapter 6, pages 201–262. Academic Press, San Diego, 1989.
- [4] James R. Arvo. Backward Ray Tracing. In *ACM SIGGRAPH '86 Course Notes - Developments in Ray Tracing*, volume 12, pages 259–263, 1986.
- [5] Johannes Guenther, Ingo Wald, and Philipp Slusallek. Realtime Caustics using Distributed Photon Mapping. In *Proceedings of the Eurographics Symposium on Rendering*, 2004.
- [6] V. Havran. *Heuristic Ray Shooting Algorithms*. Ph.d. thesis, Department of Computer Science and Engineering, Czech Technical University in Prague, 2001.
- [7] V. Havran, Jiri Bittner, and H.P. Seidel. Ray maps for global illumination. *SIGGRAPH 2004 Sketches Programme*, 2004.
- [8] V. Havran and W. Purgathofer. On comparing ray shooting algorithms. *Computer and Graphics*, 27, Issue 4:593–604, August 2003.
- [9] V. Havran, J. Pšikryl, and W. Purgathofer. Statistical comparison of ray-shooting efficiency schemes. Technical Report TR-186-2-00-14, Institute of Computer Graphics and Algorithms, Vienna University of Technology, may 2000.
- [10] Heinrich Hey and Werner Purgathofer. Advanced radiance estimation for photon map global illumination. *Computer Graphics Forum*, 21(3):541–546, 2002.
- [11] Henrik Wann Jensen. *Realistic Image Synthesis using Photon Mapping*. AK Peters, 2001.
- [12] H.W. Jensen. Global illumination using photon maps. In *Rendering Techniques '96*, pages 21–30. Springer-Verlag, 1996.
- [13] M. Lastra, C. Ureña, J. Revelles, and R. Montes. A particle-path based method for Monte-Carlo density estimation. *Poster at: 13th EUROGRAPHICS Workshop on Rendering*, 2002.
- [14] Miguel Lastra Leidinger. *Stochastic Rendering Techniques for Complex Environments*. PhD thesis, University of Granada, 2004.
- [15] J. Lext, U. Assarsson, and T. Moeller. Bart: A benchmark for animated ray tracing. Technical report, Dept. of Computer Engineering, Chalmers University of Technology, Goeteborg, 2000.
- [16] S.N. Pattanaik and S.P. Mudur. Computation of global illumination by Monte Carlo simulation of the particle model of light. *Proceedings of 3rd Eurographics Rendering Workshop, Bristol*, 1992.
- [17] Erik Reinhard, Arjan J. F. Kok, and Frederik W. Jansen. Cost prediction in ray tracing. In *Rendering Techniques '96*, pages 41–50. Springer-Verlag, June 1996.
- [18] J. Revelles, M. Lastra, R.J. García, and R. Montes. A formal framework approach for ray-scene intersection test improvement. In *WSCG'2003*, ISBN: 80-903100-2-8, 2003.
- [19] L. Santalo. *Integral Geometry and Geometric Probability*. Cambridge University Press, 2 edition, October 2002.
- [20] K. Sung and P. Shirley. Ray tracing with the BSP tree. In *Graphics Gems III*, pages 271–274.
- [21] L. Szirmay-Kalos and G. Márton. Worst-case versus average case complexity of ray-shooting. *Computing*, 61(2):103–131, 1998.
- [22] I. Wald, J. Guenther, and P. Slusallek. Balancing Considered Harmful – Faster Photon Mapping using the Voxel Volume Heuristic. *Computer Graphics Forum*, 22(3), 2004.
- [23] B. Walter, P. M. Hubbard, P. Shirley, and D. P. Greenberg. Global illumination using local linear density estimation. *ACM Transactions on Graphics*, 16(3):217–259, July 1997.