

Proyecto Fin de Carrera

Clustering Bidireccional en Iluminación Global Basada en Puntos

Raúl Buisán Esporrín

Director: Adrián Jarabo Torrijos
Ponente: Dr. Diego Gutiérrez Pérez

Escuela de Ingeniería y Arquitectura
2014

Clustering Bidireccional en Iluminación Global Basada en Puntos

Resumen

La síntesis de imágenes fotorrealistas por ordenador, requiere modelar y simular de forma precisa las interacciones entre luz y materia. Para conseguir este realismo, no sólo se debe calcular la iluminación que proviene de las distintas fuentes de luz que iluminan la escena, sino que también se debe tener en cuenta la energía reflejada entre las distintas superficies, denominada iluminación global. A pesar de los grandes avances tecnológicos, la generación de este tipo de imágenes requiere de una gran cantidad de tiempo y recursos. En producciones cinematográficas este coste tiene un importante impacto debido a la complejidad de las escenas modeladas además de la necesidad de generar miles de fotogramas. Por ello, las productoras invierten millones de dólares en potentes *clusters* de cálculo para la generación de contenido digital de sus películas, siendo la investigación y desarrollo de nuevas técnicas un tema de gran interés.

En los últimos años, han surgido técnicas capaces de calcular la iluminación global de forma aproximada. A pesar de ofrecer resultados aproximados, son lo suficientemente convincentes como para que sus errores pasen desapercibidos. Algunos de estos métodos simulan la luz reflejada por las superficies como un conjunto de luces virtuales (VPLs), calculando su contribución mediante una evaluación jerárquica de las mismas, agrupando VPLs similares como una única fuente de energía. Debido a su eficiencia, estas técnicas han sido extensamente utilizadas en multitud de producciones cinematográficas.

A pesar de su eficiencia, estos métodos no escalan bien con el número de píxeles a iluminar. Este hecho se agrava con las enormes resoluciones necesarias en la creación de contenido digital. La alta definición se ha convertido en un estándar y el contenido 3D se está implantando progresivamente, donde la generación de imágenes desde múltiples puntos de vista es necesaria. Sin embargo, las muestras generadas desde la cámara presentan una coherencia que puede ser explotada para aproximar la luz que reciben.

En este proyecto se aborda el desarrollo de una nueva técnica para el cálculo de la iluminación global basada en VPLs, que explota las similitudes entre las muestras desde la cámara y las VPLs para reducir de forma adaptativa la cantidad de computación, mediante la evaluación jerárquica de las contribuciones de las luces sobre los puntos a iluminar. De esta manera, podemos desarrollar un algoritmo capaz de generar imágenes o grupos de imágenes de alta resolución con gran cantidad de elementos de iluminación con costes sublineales, tanto en el número de luces virtuales como en el número de píxeles.

Para llevar a cabo esta tarea, implementamos el algoritmo escalable *Point Based Color Bleeding*, adaptándolo a un *framework* basado en trazado de rayos, y planteando mejoras para evitar algunos de sus problemas más comunes. Así mismo, desarrollamos una nueva técnica capaz de aproximar la iluminación global jerarquizando el conjunto de muestras generadas desde la cámara para finalmente combinar ambos algoritmos en una técnica que minimiza el coste del renderizado, con órdenes de complejidad sublineales con respecto del número de luces y muestras.

Agradecimientos

Quiero dar gracias a Adrián por su implicación y ayuda a lo largo de la realización del proyecto. A él y a Diego por darme la oportunidad de trabajar en otros proyectos del grupo. Y por supuesto a todos los compañeros del *Graphics and Imaging Lab*. También a mi familia y amigos por su apoyo a lo largo de estos años.

Índice general

1. Introducción	1
2. Conocimiento previo	3
2.1. Iluminación indirecta con luces virtuales	4
2.2. Trabajo previo	5
3. Point-Based Global Illumination	7
3.1. Preproceso	7
3.2. Cálculo de la iluminación	8
3.2.1. Condiciones de parada	8
3.2.2. Singularidades	10
3.2.3. Test de visibilidad	11
4. <i>Clustering</i> de muestras de la cámara	13
4.1. Preproceso	13
4.2. Cálculo de la iluminación	14
4.2.1. Condiciones de parada	15
5. Clustering bidireccional	17
5.1. Preproceso	17
5.2. Cálculo de la iluminación	17
5.2.1. Visibilidad	18
5.2.2. Condiciones de parada y refinado de pares	19
6. Resultados	21
6.1. Análisis de rendimiento	21
6.1.1. Análisis de rendimiento de PBGI	21
6.1.2. Análisis de rendimiento del método de <i>clustering</i> de muestras	22
6.1.3. Análisis de rendimiento del método bidireccional	23
6.2. Análisis de calidad y parámetros	24
6.2.1. Algoritmo de <i>clustering</i> de VPLs	24
6.2.2. Algoritmo de <i>clustering</i> de muestras	24
6.2.3. Algoritmo de <i>clustering</i> de VPLs y muestras	24

6.2.4. Análisis de error	24
6.3. Resultados adicionales	27
7. Conclusiones	31
7.1. Discusión y trabajo futuro	31
7.2. Conclusión personal	32
7.3. Diagrama temporal del proyecto	32
A. Métodos auxiliares	39
A.1. Muestreo uniforme de superficies	39
B. Resultados adicionales	41
B.1. Tablas de rendimiento	41

Capítulo 1

Introducción

La síntesis de imágenes fotorrealistas por ordenador, requiere modelar y simular de forma precisa las interacciones entre luz y materia. Para conseguir este realismo, no sólo se debe calcular la iluminación que proviene de las distintas fuentes de luz que iluminan la escena, sino que se debe tener en cuenta la energía reflejada entre las distintas superficies, denominada iluminación global. Sin embargo, su cálculo requiere de una ingente cantidad de tiempo y recursos. En producciones cinematográficas este coste tiene un especial impacto: debido a la complejidad de las escenas generadas, un único fotograma requiere horas de cálculo; por ejemplo, en *Cars 2* cada fotograma necesitó una media de 11.5 horas de render ¹ ². Este coste por fotograma se multiplica por la cantidad de *frames* que son necesarios renderizar (129600 para 90 minutos a 24 fps). A pesar de su elevado coste, prescindir de la iluminación global ofrece resultados poco realistas como se puede observar en la Figura 1.1, y por ello las productoras invierten millones de dólares en grandes *clusters* de cálculo para la generación del contenido digital de sus películas, siendo el estudio y desarrollo de nuevas técnicas eficientes de renderizado un tema de gran interés.



Figura 1.1: La iluminación global es imprescindible para dotar realismo a una imagen. Como se puede observar, incluso con modelos geométricos detallados, la ausencia de la luz reflejada entre las superficies (**derecha**) oculta muchos de los detalles y da un resultado antinatural. Por el contrario, la iluminación global permite generar imágenes más creíbles y fieles a la realidad (**derecha**). Fuente: [TL04]. ©2011 Dreamworks Animation LLC.

Diversos trabajos resuelven la iluminación global modelándola como un conjunto de *luces virtuales* (VPLs) que simulan la energía reflejada por las superficies. La contribución de estas luces

¹<http://www.cnet.com/news/new-technology-revs-up-pixars-cars-2>

²En Pixar, el coste estándar es de alrededor de 10 horas por fotograma. Fuente: Henrik Wann Jensen.

virtuales es aproximada mediante una evaluación jerárquica de las mismas, agrupando VPLs con una emisión similar como una única fuente de energía, [WFA⁺05, Chr08]. Aunque la solución no es exacta, los errores generados son virtualmente imperceptibles para el observador humano. Este tipo de técnicas son eficientes, y dan excelentes resultados hasta el punto de que han sido utilizadas en un gran número de producciones cinematográficas (por ejemplo *Pirates of the Caribbean 2* o *Up*), y habiendo sido galardonadas con un *Scientific & Technical Award* de la Academia.

Aunque eficientes, estos algoritmos no escalan bien con el número de píxels a iluminar. Este factor es especialmente problemático debido a las enormes resoluciones necesarias en la generación de contenido digital (la alta definición es un estándar en la imagen doméstica, y el formato cine requiere de incluso más resolución). Asimismo, el contenido de imagen 3D se está implantando progresivamente; para su generación tanto en formato estereoscópico como automultiescópico [MWA⁺13] (pantallas 3D sin gafas), se requiere renderizar las escenas desde múltiples puntos de vista (*light fields* [LH96]), lo cual multiplica a su vez el número de píxels. En estas situaciones, el número de muestras es muy grande, y su renderizado se convierte en una tarea especialmente costosa.

Sin embargo, las muestras generadas desde la cámara, presentan gran coherencia que se puede explotar con el objetivo de acelerar los cálculos. En este trabajo presentamos nuevos algoritmos que permiten la simulación de la iluminación global en escenas tridimensionales de forma eficiente y escalable con el número de píxels a renderizar. Para ello, se analiza el estado del arte en técnicas que aproximan la iluminación global mediante una ordenación y evaluación jerárquica de las luces en la escena, y se extrapolan dichas técnicas al dominio de los píxels de la imagen. Esto nos permite desarrollar un algoritmo capaz de generar imágenes o grupos de imágenes de alta resolución con gran cantidad de elementos de iluminación con costes sublineales, tanto en el número de luces virtuales como en número de píxels. Para ello, desarrollamos un conjunto de nuevas heurísticas que tienen en cuenta la mayor saliencia de los errores en las aproximaciones en espacio de imagen, así como discutimos las aplicaciones de nuestro sistema a un amplio conjunto de efectos (e.g. *depth of field*, *antialiasing*).

Las **contribuciones** de este proyecto son:

1. Implementación del PBGI [Chr08], algoritmo de iluminación escalable, analizando sus deficiencias y virtudes, y proponiendo mejoras algorítmicas para evitar algunos de sus problemas más comunes, e.g. *Virtual Spherical Lights* [HKWB09]. Esta técnica que sirve como punto de partida para el desarrollo de nuevos algoritmos, ha sido elegida por ser general, relativamente fácil de implementar, y su amplia utilización en producción.
2. Desarrollo de un nuevo algoritmo de renderizado capaz de agrupar muestras generadas desde todos los posibles puntos de vista, aproximando la iluminación que reciben, y obteniendo rendimientos sublineales con respecto de su número. Esta técnica nos permite obtener una primera aproximación de la idea de agrupar muestras de la cámara para simular de una forma más eficiente la iluminación global.
3. Desarrollo de un nuevo algoritmo de cálculo de la iluminación global, capaz de aproximar de forma adaptativa tanto la contribución de las muestras de luz, como la energía recibida por las muestras de la cámara, minimizando el coste de renderizado para un error dado.

Capítulo 2

Conocimiento previo

La radiancia saliente desde un punto x hacia una dirección $\vec{\omega}_o$, se modela con la Ecuación de Render [Kaj86] como la suma de la radiancia emitida L_e y la radiancia reflejada:

$$L_o(x, \vec{\omega}_o) = L_e(x, \vec{\omega}_o) + L_r(x, \vec{\omega}_o). \quad (2.1)$$

$$L_r(x, \vec{\omega}_o) = \int_{\Omega^+} L_i(x, \vec{\omega}_i) f_r(x, \vec{\omega}_i, \vec{\omega}_o) \langle \vec{n}_x, \vec{\omega}_i \rangle^+ d\vec{\omega}_i. \quad (2.2)$$

siendo Ω^+ la semiesfera orientada hacia la normal del punto \vec{n}_x , $\langle \vec{n}_x, \vec{\omega}_i \rangle^+$ es la atenuación geométrica modelada como el producto escalar positivo entre la normal \vec{n}_x y la dirección de incidencia $\vec{\omega}_i$, y f_r es la *Bidirectional Reflectance Distribution Function* o *BRDF*, relaciona la energía entrante y saliente en un punto dependiendo de las direcciones de entrada y salida. El comportamiento de la BRDFs dependerá de las propiedades de la superficie, y siempre se cumple que $\int_{\Omega^+} f_r(x, \vec{\omega}_i, \vec{\omega}_o) d\vec{\omega}_i \leq 1$. La Figura 2.1 muestra ejemplos de distintas *BRDFs*. L_i es la radiancia incidente en el punto x en la dirección ω_i . Esta radiancia puede provenir de una fuente de luz, o haber sido reflejada por otra superficie ¹. Esta radiancia reflejada se denomina iluminación indirecta o global. Dado que no es posible resolver este tipo de integrales de forma analítica, en general se recurre a métodos estocásticos (integración por Monte Carlo [Owe13]) que aproximan la integral en base a un conjunto de muestras aleatorias. Así, para cada punto se evalúan un conjunto de muestras direccionales, y se evalúa recursivamente la Ecuación 2.2.

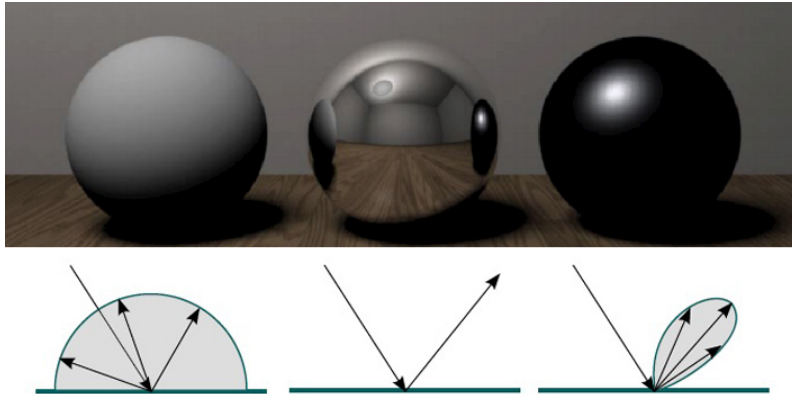


Figura 2.1: Ejemplos de BRDFs modelando materiales con diferentes propiedades ópticas. **(Izquierda)**: Reflexión Lambertiana. **(Centro)**: Reflexión especular. **(Derecha)**: Reflexión de una superficie brillante (*glossy*).

¹Nótese que, aunque en este trabajo nos centramos en transporte de luz entre superficies, la radiancia también puede haber sido reflejada por un medio participativo (e.g. niebla).

Teniendo en cuenta que radiancia incidente en un punto x proviene de los distintos puntos de la escena, la Ecuación 2.2 se puede reformular modificando el dominio de integración desde las posibles direcciones en la semiesfera, al conjunto de las superficies visibles desde el punto:

$$L_r(x, \vec{\omega}_o) = \int_{\mathcal{S}} L_i(s, \vec{\omega}_i) f_r(x, \vec{\omega}_i, \vec{\omega}_o) G(x, s) V(x, s) ds. \quad (2.3)$$

donde \mathcal{S} es el conjunto de todas las superficies de la escena, V una función binaria cuyo valor es uno si la superficie s es visible desde x y G es el término geométrico, que atenúa por distancia y por área proyectada del rayo incidente:

$$G(x, s) = \frac{\langle \vec{n}_s, -\vec{\omega}_i \rangle^+ \langle \vec{n}_x, \vec{\omega}_i \rangle^+}{\|s - x\|^2}. \quad (2.4)$$

2.1. Iluminación indirecta con luces virtuales

Intuitivamente, la Ecuación 2.3 modela cada punto $s \in \mathcal{S}$ como una luz virtual (VPL) que emite una cantidad de energía equivalente a la luz que refleja [Kel97]. Las VPLs son generadas en una etapa de preproceso y posteriormente son utilizadas para iluminar todos los puntos generados desde la cámara como muestra la Figura 2.2. De esta forma, la energía reflejada desde un punto x en una dirección $\vec{\omega}_o$ se calcula como:

$$L_r(x, \vec{\omega}_o) = \sum_{j=0}^N f_r(x, \vec{\omega}_{i_j}, \vec{\omega}_o) G(x, l_j) V(x, l_j) L_o(l_j, -\omega_{i_j}). \quad (2.5)$$

siendo N el número de VPLs, l_j la posición de la VPL j y $L_o(l_j, -\omega_{i_j})$ la energía que emite la VPL j hacia el punto x , calculada mediante la Ecuación 2.1.

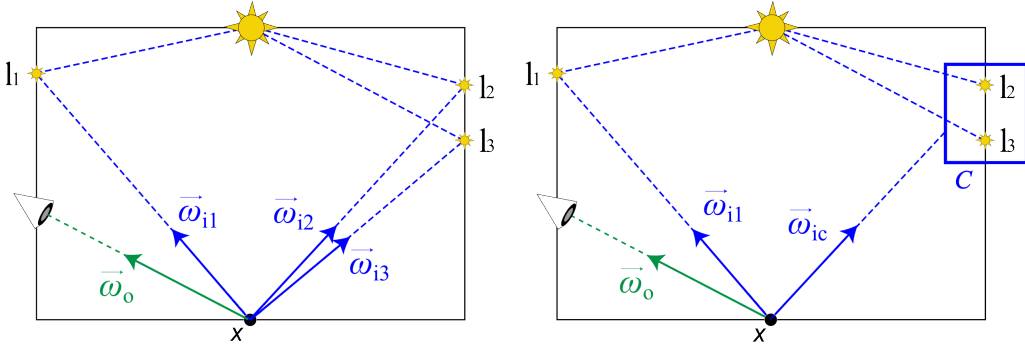


Figura 2.2: (a) La iluminación global sobre el punto x es calculada a partir de la contribución de tres luces virtuales. (b) La iluminación global es calculada de forma aproximada agrupando las luces l_2 y l_3 .

Desafortunadamente, el coste del cálculo de la Ecuación 2.5 es lineal con respecto del número de VPLs, dado que cada término debe ser evaluado para cada VPL. Definiendo un *cluster* C como un subconjunto de las luces virtuales su contribución puede ser aproximada como:

$$L_{rC}(x, \vec{\omega}_o) = \sum_{j \in C} f_r(x, \vec{\omega}_{i_j}, \vec{\omega}_o) G(x, l_j) V(x, l_j) L_o(l_j, -\omega_{i_j}) \approx f_r(x, \vec{\omega}_{iC}, \vec{\omega}_o) G(x, C) V(x, C) L_o(C, -\omega_{iC}). \quad (2.6)$$

siendo $\vec{\omega}_{iC}$ la dirección de x al *cluster* C . Nótese que tanto el método para agrupar luces virtuales, así como para evaluarlas, varía con respecto del algoritmo utilizado. En el resto de sección revisamos los diferentes métodos existentes en la literatura. Más tarde, en el Capítulo 3 describimos en profundidad el utilizado en este trabajo

2.2. Trabajo previo

El problema de la iluminación global ha sido extensamente estudiado, existiendo actualmente un gran número de técnicas capaces de calcularla [DBB06]. *Bidirectional Path Tracing* (BDPT) [LW93, VG95a], calcula la iluminación global generando de forma estocástica subcaminos desde la cámara y desde las fuentes de luz, que son conectados de forma determinista para formar un camino completo ². *Photon mapping* [Jen01] es un método basado en trazado de partículas, que crea un mapa de fotones que representa la distribución de luz en la escena. Para cada píxel a sombrear, la energía que recibe es estimada a partir del mapa, permitiendo calcular caminos difícilmente muestreables, como caústicas. Sin embargo, en grandes escenas se necesita una gran cantidad de fotones para obtener imágenes sin ruido. Los métodos basados en luces virtuales [DKH⁺14, Kel97] modelan la irradiancia (energía reflejada) por las superficies como un conjunto de luces puntuales. Estos métodos necesitan del orden de cientos de miles de VPLs para modelar la iluminación indirecta de forma compleja.

Dentro de la familia de algoritmos basados en luces virtuales, numerosas técnicas tratan de reducir el coste computacional aproximando la contribución de las luces virtuales en la escena. Walter et al. [WFA⁺05] proponen *Lightcuts* en el que organizan las VPLs en un árbol de luces, seleccionando de forma adaptativa los cortes óptimos en función de una métrica perceptual. *Multidimensional Lightcuts* (MDLC) [WABG06] extiende *Lightcuts* [WFA⁺05] para simular de forma eficiente efectos distribuidos (*motion-blur*, *depth-of-field*, *antialiasing* ...) [CPC84] agrupando tanto luces virtuales como muestras generadas desde la cámara; sin embargo, solo es capaz de agrupar muestras generadas dentro del dominio de cada píxel de la imagen. Walter et al. [WKB12] combinan MDLC con BDPT utilizando *multiple importance sampling* (MIS) [VG95b] para minimizar el error seleccionando la técnica de muestreo más apropiada. Otros trabajos [HPB07, OP11, HVAPB08], modelan el problema como una gran matriz que relaciona las VPLs y los puntos de la cámara, calculando la imagen como la suma de un subconjunto de las columnas de dicha matriz.

Otras técnicas crean las luces virtuales descomponiendo la escena en pequeños discos que almacenan la energía que reciben de las fuentes de luz y los utilizan posteriormente para calcular la iluminación global difusa o la oclusión ambiental. Por ejemplo, Bunnell [Bun05] utiliza esta aproximación acelerando el proceso de renderizado agrupando discos lejanos en un único disco que aproxima su contribución. Christensen [Chr08] extiende este procedimiento mediante la jerarquización de los discos en un *octree* que permite calcular de una forma aproximada y eficiente la iluminación global. Ritschel et al. [REG⁺09] proponen un método basado en GPU utilizando una jerarquización de las luces y muestreo por importancia de las BRDFs. Wang et al. [WHB⁺13], extienden PBGI [Chr08] bajo la idea de que píxeles cercanos tendrán cortes similares en el árbol de luces, particionando el espacio de pantalla para reducir la cantidad de cálculos. Con el objetivo de disminuir el consumo de memoria de los métodos basados en puntos, Buchholz y Boubekeur [BB12] realizan una compresión de los armónicos esféricos empleados, mientras que Kontkanen et al. [KTO11] utilizan técnicas *out-of-core* trabajando con subconjuntos de los árboles de luces virtuales y mejorando el paralelismo del proceso de renderizado.

Para simular materiales translúcidos, donde la luz penetra la superficie de objetos translúcidos por un punto, es dispersada en su interior para salir por un punto distinto de la superficie [JMLH01] (*Subsurface Scattering*, SSS), Jensen y Buhler [JB02] proponen un método de dos etapas, calculando de forma eficiente la energía dispersada en el interior de los objetos a partir de una evaluación jerárquica de la irradiancia sobre las superficies, calculada en una etapa de preproceso. Arbree et al. [AWB08] aceleran el cálculo del SSS aplicando *clustering* de VPLs y de las irradiancias sobre los objetos de un modo similar a MDLC.

Estos métodos permiten costes sublineales con respecto del número de luces, aunque son lineales respecto del número de puntos a sombrear. Por contra, nosotros proponemos algoritmos que escalan sublinealmente tanto el número de píxeles como con el de luces en la escena.

²Definimos un camino, como la sucesión de las interacciones que experimenta la luz desde que es emitida por las fuentes hasta que alcanza el sensor de la cámara.

Capítulo 3

Point-Based Global Illumination

Point-Based Global Illumination (PBGI) [Chr08] es una técnica que permite el cálculo aproximado de la iluminación global debido a interreflexiones difusas de forma eficiente, habiendo sido utilizada desde su aparición en más de una docena de títulos además de haber sido galardonada con un *Scientific and Technical Award* de la Academia.

El algoritmo trabaja en dos etapas. En una fase de preproceso (Sección 3.1), la irradiancia de las superficies de la escena es descompuesta en un conjunto de luces virtuales, que son ordenadas en una estructura jerárquica basadas en su posición. En la segunda fase de renderizado (3.2), para calcular la iluminación indirecta en cada punto x visible desde la cámara, recorreremos la jerarquía de luces, refinando el corte en el árbol hasta alcanzar el nivel de precisión mínimo en la aproximación de la energía incidente en x . De esta manera, se evita la necesidad de evaluar el total de las luces virtuales, obteniendo órdenes de complejidad sublineales ($O(\log(N))$) con respecto del número de VPLs. En la Figura 3.1 se muestra un ejemplo de este proceso.

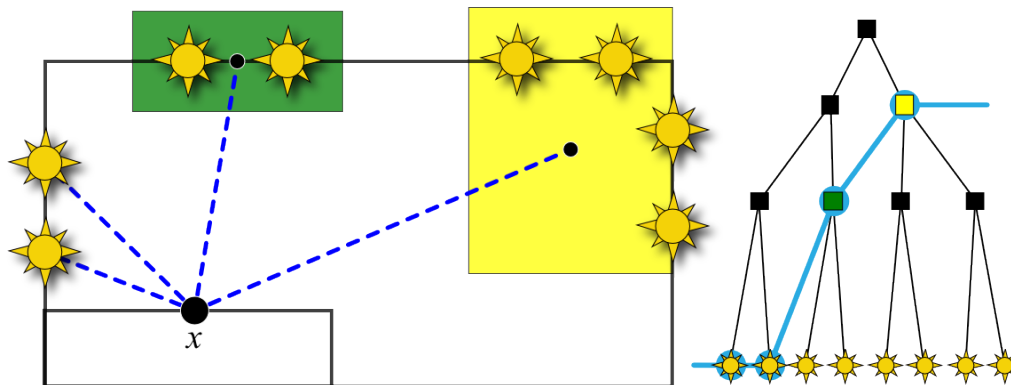


Figura 3.1: **Izquierda** Escena iluminada por 8 VPLs. La luz recibida por el punto x es aproximada mediante el agrupamiento de algunas de las VPLs generadas. **Derecha:** Representación simplificada del árbol de luces. La línea azul representa las VPLs y *clusters* utilizados para aproximar la iluminación, formando un corte sobre el árbol.

3.1. Preproceso

Datos de entrada Los datos de entrada para PBGI son el conjunto de VPLs en los que se descompone la iluminación de la escena. Para la generación de esta nube de puntos utilizamos una técnica simple basada en el muestreo uniforme de las superficies de la escena (consultar el Anexo A.1 para más detalles).

Para cada VPL almacenamos su posición, su área, su normal y la energía que emiten. A pesar

de tener un área asociada, se puede realizar una analogía con el concepto de VPL descrito en el Capítulo 2. El área A se calcula como la superficie total que abarcan las primitivas de la escena dividida por el número de muestras generadas, la normal del disco es la normal de la superficie en la posición de la VPL. Finalmente, asumiendo que las superficies tienen una reflectancia Lambertiana, la luz emitida depende únicamente de la energía recibida desde las fuentes de luz, siendo calculada como:

$$L_e(\vec{\omega}_o) = A \frac{\rho}{\pi} L_i \langle \vec{n}, \vec{\omega}_o \rangle^+. \quad (3.1)$$

siendo ρ el albedo¹ de la superficie. Aunque este método permite generar millones de puntos de forma eficiente, se ignora si están ocluidos con respecto de las luces de la escena, por lo que se pueden generar muestras que no contribuirán a la iluminación global. Nótese que el resto del algoritmo es independiente de cómo se generan las luces virtuales pudiendo utilizar otras alternativas (e.g. Transporte de partículas [Kel97]).

El árbol de luces Con el fin de evaluar de una forma eficiente la contribución de las VPLs generadas, son introducidas en una jerarquía creando *clusters* de luces virtuales. Basándonos en [Chr08], utilizamos un árbol de tipo *octree*. Aunque se podría haber utilizado otro tipo de estructuras de particionamiento del espacio, un *octree* nos permite construir la jerarquía de forma rápida. Durante la generación del árbol, los nodos se subdividen hasta que las hojas contienen un número máximo de VPLs. En la implementación realizada, para una gestión óptima de la memoria se utiliza un máximo de ocho VPLs por hoja.

3.2. Cálculo de la iluminación

Para calcular la iluminación global de forma eficiente, para cada punto a iluminar x recorreremos el árbol desde su nodo raíz, refinando el corte del árbol hasta que se alcanza la condición de parada (ver Sección 3.2.1). La iluminación en x se calcula como la suma de la contribución de cada uno de los nodos del corte (Ecuación 2.5). Esto significa que es necesario evaluar, además de la BRDF y los términos de atenuación geométrica, la visibilidad de x desde los nodos (ver Sección 3.2.3), y la radiancia emitida desde cada nodo.

En el caso de que el nodo es una hoja (una VPL individual), calculamos su contribución evaluando la Ecuación 2.5. Nótese que el término geométrico puede producir singularidades en forma de puntos brillantes, en puntos cercanos a la VPL; en la Sección 3.2.2 describimos cómo gestionarlas.

En caso de que el nodo a evaluar sea un *cluster* de VPLs, se aproxima su contribución como la suma de todas las VPLs que contiene. Ésta es dependiente de la dirección de emisión, pero de baja frecuencia, por lo que proyectamos la suma de las contribuciones de las luces virtuales en armónicos esféricos. Más detalles sobre armónicos esféricos se pueden encontrar en [Gre03]. Para almacenar iluminación difusa, se considera que 3 niveles de armónicos esféricos (9 coeficientes) ofrecen suficiente precisión [RH01, Chr08].

3.2.1. Condiciones de parada

Para determinar si un *cluster* es lo suficientemente preciso se utiliza el valor de su ángulo sólido Ω_L con respecto del punto a iluminar. El ángulo sólido, definido como la proyección de un área sobre una esfera de radio unidad, mide el tamaño aparente de una superficie S desde un punto x y es calculado como:

¹El albedo es la cantidad de radiación que cualquier superficie refleja respecto a la radiación que incide sobre la misma.

$$\Omega_L = \int_A \frac{\langle \vec{n}_s, \vec{\omega}_s \rangle}{d(s, x)^2} ds \approx \frac{\|A\| \langle \vec{n}_A, \vec{\omega}_A \rangle}{d(A, x)^2}. \quad (3.2)$$

con A la superficie de S visible desde x con área $\|A\|$, \vec{n}_s la normal en un punto diferencial $s \in A$, $\vec{\omega}_s$ y $d(s, x)$ dirección y distancia desde s a x , y \vec{n}_A , $\vec{\omega}_A$ y $d(A, x)$ la normal, dirección y distancia desde un punto representativo en A hasta x .

El área proyectada de un nodo, se puede aproximar como la suma de las áreas proyectadas de las VPLs que contiene. Al igual que la energía emitida, el área proyectada de una VPL varía en función de su orientación. Por ello, se procede de manera análoga almacenando su valor en una base de armónicos esféricos. De esta forma, es posible obtener el valor aproximado del área proyectada de un *cluster* y calcular su ángulo sólido. Sin embargo, en ocasiones cuando los *clusters* son muy grandes, el error introducido por esta aproximación provoca artefactos en la imagen. Para evitarlos, calculamos un segundo valor del área proyectada Ω_S a partir de la esfera de radio r que envuelve el volumen de un *cluster* como:

$$\Omega_S = \frac{\pi r^2}{d^2}. \quad (3.3)$$

Para acelerar el proceso de renderizado, se impone una condición adicional. A menudo, los puntos a iluminar están orientados de tal forma que nunca recibirán energía de muchas de las VPLs. Esta situación puede ser detectada permitiendo evitar el recorrido de ramas completas que nunca contribuirán a la iluminación. Para calcular esta condición de corte, se calcula el producto escalar entre la normal del punto x , \vec{n}_x y los vectores desde x hasta las ocho esquinas de la *bounding box* del *cluster* como se muestra en la Figura 3.3. Si alguno de los productos escalares es mayor que cero, indica que puede haber VPLs dentro del *cluster* que pueden irradiar el punto x ; si todos los productos escalares son menores que cero, se finaliza el recorrido para ese nodo y sus hijos. Denominamos a esta condición de parada 'Test del horizonte'.

Por tanto, las condiciones de parada del algoritmo de *clustering* de VPLs son:

1. Ω_L Ángulo sólido de un nodo calculado de acuerdo a la Ecuación 3.2.
2. Ω_S Ángulo sólido de un nodo calculado de acuerdo a la Ecuación 3.3.
3. Test del horizonte.

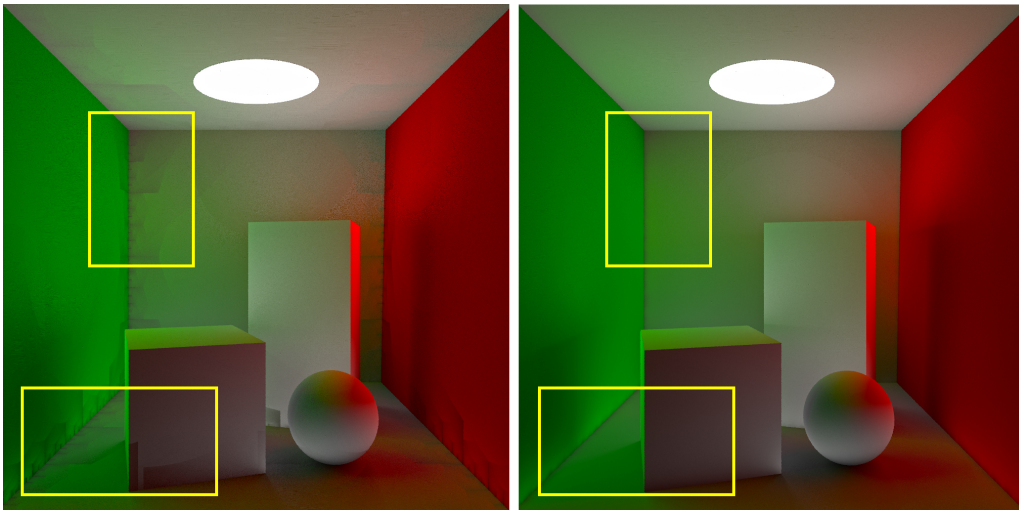


Figura 3.2: El cálculo del valor Ω_S permite evitar las imprecisiones introducidas por los armónicos esféricos para calcular el área proyectada. **Izquierda:** Resultado obtenido aplicando Ω_S . **Derecha:** Resultado obtenido sin utilizar Ω_S .

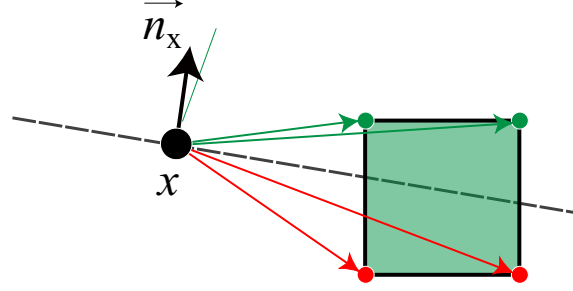


Figura 3.3: Muchos de los *clusters* del árbol no contribuyen a la iluminación debido a la orientación de los puntos a iluminar. Esta situación se detecta calculando el producto escalar entre la normal las muestras, y los vectores que apuntan a las esquinas de la *bounding box* del *cluster*. Si todos los productos escalares son menores que cero, se descartarán todas las ramas que dependen de ese nodo.

3.2.2. Singularidades

Las singularidades consisten en picos de energía producidos por VPLs a distancias próximas a cero provocando que el término geométrico (Ecuación 2.4) tienda a infinito.

Una forma de solventar este problema consiste en establecer un valor máximo para el término geométrico. Sin embargo, esta aproximación produce una pérdida en la iluminación. En escenas difusas esta pérdida de energía solo es visible en ciertas áreas (esquinas o geometría compleja), pero es significativo en materiales brillantes donde estos picos de energía proceden de las propiedades de reflectancia de los objetos [KFB10].

La solución adoptada para evitar estas singularidades consiste en modelar las VPLs como luces virtuales esféricas (*Virtual Spherical Lights* o VSLs) [HKWB09]. Conceptualmente, la VSL expande la luz puntual 0D en una esfera de radio finito 3D como muestra la Figura 3.4. De esta forma la energía recibida se calcula integrando la radiancia saliente sobre el ángulo sólido de la VSL, eliminando el término geométrico.

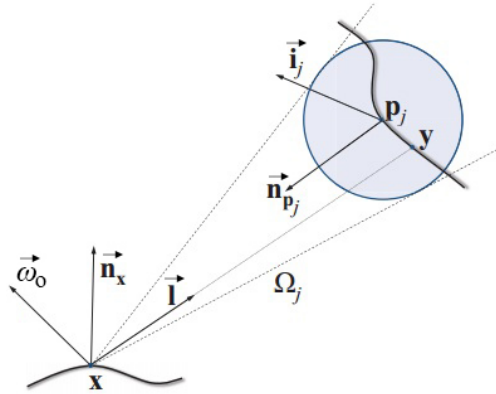


Figura 3.4: Luz esférica virtual. Un punto de una superficie x con normal \vec{n}_x y vista desde una dirección $\vec{\omega}_o$ recibe energía de una VSL desde todo el ángulo sólido Ω_j , localizada en p_j , con normal \vec{n}_{p_j} . Fuente: [HKWB09].

Calculamos la energía reflejada en un punto x desde una VPL modelada como una VSL j de radio r_j , integrando su radiancia en el ángulo sólido Ω_j como:

$$L_r(x, \vec{\omega}_o) = \frac{L_{VPL}}{\pi r_j^2} \int_{\Omega_j} f_r(x, \vec{l}, \vec{\omega}_o) \langle \vec{n}_x, \vec{l} \rangle^+ f_r(y, \vec{l}_j, -\vec{l}) \langle \vec{n}_{p_j}, -\vec{l} \rangle^+ V(x, y) d\vec{l}. \quad (3.4)$$

siendo L_{VPL} la radiancia recibida de todas las fuentes de luz por la VPL modelada como VSL.

Para simplificar el cálculo de la Ecuación 3.4 hacemos algunas aproximaciones:

1. Calculamos la visibilidad entre el punto x y el centro de la VSL, p_j . Asumiendo un número suficiente de luces virtuales, este test es preciso.
2. El vector normal y la BRDF de los puntos j en las superficies contenidas en la VSL son aproximados por la normal \vec{n}_{p_j} y la BRDF de la posición de la VSL p_j .
3. Teniendo en cuenta que modelamos la iluminación global difusa, la BRDF de la VSL es constante y puede ser extraída de la integral.
4. Utilizamos un radio fijo para las VSLs calculado multiplicando el radio de la VPL, r_{VPL} , por una constante k .
5. La radiancia recibida por la VSL de las fuentes de luz es igual a $L_{VPL} = L_i \pi r_{VPL}^2$, siendo L_i la radiancia recibida en la posición p_j desde las fuentes de luz.

Con estas asunciones, simplificamos la Ecuación 3.4 como:

$$L_r(x, \vec{\omega}_o) = \frac{L_e}{k^2} \frac{\rho_j}{\pi} V(x, p_j) \int_{\Omega_i} f_r(x, \vec{l}, \vec{\omega}_o) \langle \vec{n}_x, \vec{l} \rangle^+ \langle \vec{n}_{p_j}, -\vec{l} \rangle^+ d\vec{l}. \quad (3.5)$$

siendo ρ_j el albedo del punto p_j . Utilizando este método, se elimina el término $1/d^2$ de la ecuación, evitando las singularidades cuando una VPL está muy cerca de una superficie. Para resolver la integral de la Ecuación 3.5 utilizamos integración por Monte Carlo aproximando su valor como:

$$L_r(x, \vec{\omega}_o) \approx \frac{L_e}{k^2} \frac{\rho_j}{\pi} V(x, p_j) \sum_{i=1}^N f_r(x, \vec{l}_i, \vec{\omega}_o) \langle \vec{n}_x, \vec{l}_i \rangle^+ \langle \vec{n}_{p_j}, -\vec{l}_i \rangle^+ \frac{1}{p(\vec{l}_i)}. \quad (3.6)$$

siendo N el número de muestras, y $p(\vec{l}_i)$ la probabilidad de generar la muestra \vec{l}_i , en nuestro caso uniforme en el ángulo sólido de la VSL en x .

Los valores escogidos para el multiplicador del radio k varían entre escenas. En las pruebas realizadas se ha encontrado que valores entre tres y seis ofrecen buenos resultados. El número de muestras es proporcional al ángulo sólido de la VSL [HKWB09], utilizando un máximo de 100 muestras para $\Omega_i = 2\pi$. Esta situación se da sin embargo de una forma extraordinaria y en la mayoría de las ocasiones se utilizará un número mucho menor. En la Figura 3.5 se compara el resultado obtenido entre la utilización de VPLs y VSLs.

3.2.3. Test de visibilidad

A la hora de calcular la energía recibida sobre un punto, se debe determinar el término de visibilidad V (Ecuación 2.3). En la implementación realizada del algoritmo, esta operación se basa en trazado de rayos en dirección a la VPL o *cluster*. Consideramos que existe una oclusión si la intersección más cercana se da a una distancia menor que la existente entre el punto a iluminar y la VPL o *cluster*. En el caso de las VPLs, se trazan rayos desde el punto x hasta su posición. Sin embargo, orientar los rayos al centro un *cluster* ofrece malos resultados, obteniendo artefactos en forma de ruido estructurado (*banding*). Esto es debido a la correlación en las muestras *entre píxels*.

Para solucionar este problema, la estrategia empleada es similar a la aproximación descrita en la Sección 3.2.2. En lugar de trazar el rayo al centro del *cluster*, se muestrea estocásticamente la dirección del rayo en el ángulo sólido del *cluster*, de forma similar a muestrear una VSL. De esta forma, pese a que aumenta la varianza, reducimos la correlación entre muestras, lo que elimina el *banding* produciendo mejores resultados [RAMN12]. En nuestra implementación, utilizamos una única muestra por *cluster*, ofreciendo buenos resultados. En la Figura 3.6 se comparan los resultados obtenidos aplicando este método de cálculo de la visibilidad con el trazado de rayos al centro del *cluster*.

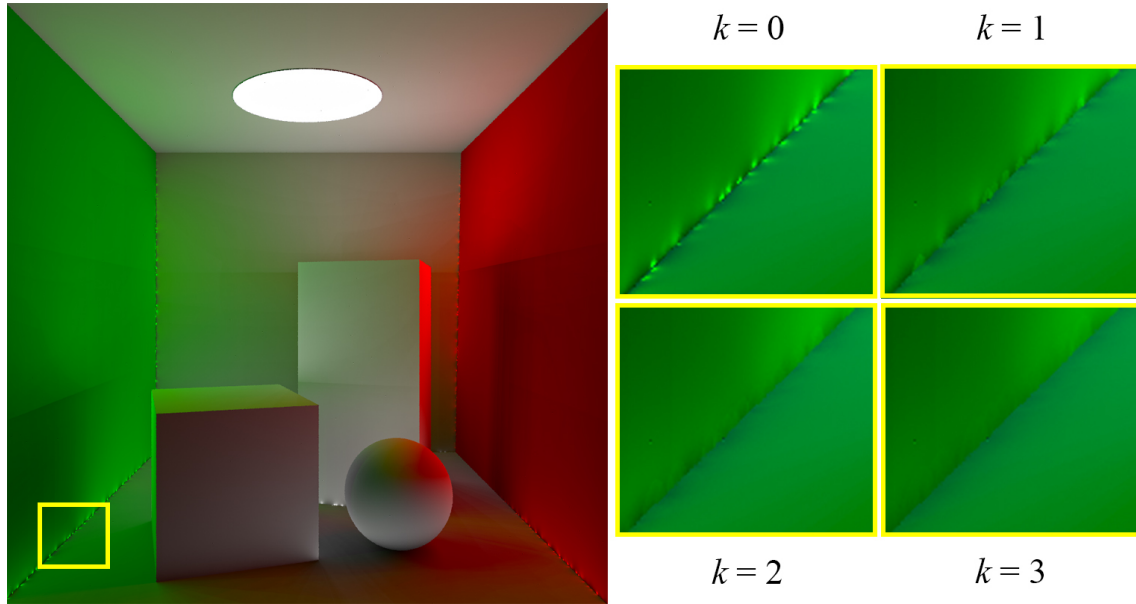


Figura 3.5: Comparación de los resultados para diferentes valores de k en una Cornell-Box difusa (**izquierda**), para $k = 0, 1, 2, 3$. Nótese cómo al aumentar el valor de k las singularidades presentes como ruido de alta frecuencia son suavizadas, mientras que la cantidad global de energía en la escena es preservada.

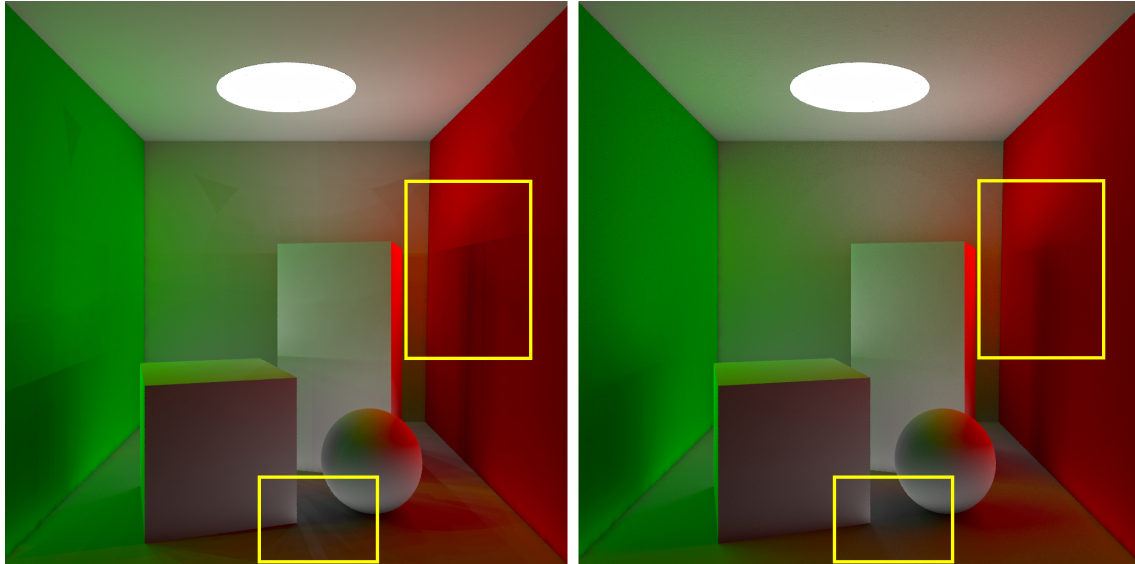


Figura 3.6: Comparación entre el muestreo de la visibilidad de los *clusters* de forma determinista al centro del *cluster* (**izquierda**), o estocásticamente en el ángulo sólido comprendido por el *cluster* (**derecha**), para el mismo número de muestras. Nótese cómo usando muestreo estocástico el *banding* es significativamente reducido, ofreciendo mucho mejores resultados con el mismo coste. En ambos casos sólo la iluminación indirecta es calculada.

Capítulo 4

Clustering de muestras de la cámara

A pesar de que las técnicas que aproximan la iluminación global jerarquizando las luces virtuales en *clusters* trabajan de forma eficiente, escalan linealmente con la cantidad de puntos que deben ser iluminados, lo cual afecta a su rendimiento al renderizar imágenes de alta resolución, o multi-vista (e.g. estéreo o *light fields*). La coherencia que presentan estos puntos puede ser aprovechada para aproximar la energía que reciben con complejidades sublineales. Aunque existen métodos capaces de agrupar tanto las luces virtuales como los puntos a iluminar, no trabajan con todo el posible conjunto de muestras que pueden ser generadas, perdiendo generalidad y la posibilidad de aproximar un mayor número de muestras.

En el presente capítulo, presentamos nuestra técnica para el cálculo de la iluminación global basada en puntos que permite agrupar las muestras de todos los posibles puntos de vista en el espacio de la escena trabajando con todo su conjunto. Este método servirá como punto de partida para explorar nuevos algoritmos para el cálculo de la iluminación global de forma escalable respecto del número de luces y muestras, que presentamos en el Capítulo 5.

De forma similar al algoritmo desarrollado en el Capítulo 3, este método trabaja en dos etapas. En la fase de preproceso, se generan tanto las VPLs como los puntos visibles desde la cámara; estas últimas son introducidas en una estructura jerárquica que se utiliza después para aproximar la iluminación indirecta en grupos de píxeles. En la etapa de renderizado, la iluminación indirecta se calcula recorriendo el árbol de muestras creando un corte que aproxima la visibilidad y la radiancia recibida por las muestras. La Figura 4.1 muestra un ejemplo del funcionamiento del algoritmo.

4.1. Preproceso

Datos de entrada El algoritmo desarrollado, necesita del conjunto de muestras generadas desde la cámara, que son creadas trazando rayos que intersectan con la geometría de la escena. Todos los puntos a iluminar deben ser almacenados para poder crear una jerarquía que permita aproximar la energía que reciben desde las VPLs. Para cada punto generado se almacena su posición, su normal, la dirección del rayo a partir del que ha sido generada, la BRDF de la superficie en la que residen, su peso, siendo este último la probabilidad con que se genera la muestra y el píxel desde el que se ha trazado el rayo.

El árbol de muestras El conjunto de muestras generado es introducido en una jerarquía. De forma análoga al trabajo desarrollado en el Capítulo 3, utilizamos un árbol de tipo *octree*, permitiendo crear una jerarquía de forma eficiente. Sin embargo, esta estructura particiona el espacio sin tener en cuenta la orientación de las muestras. Por ello, en un trabajo futuro se evaluará la utilización de otro tipo de estructuras que permitan agrupar puntos con normales similares.

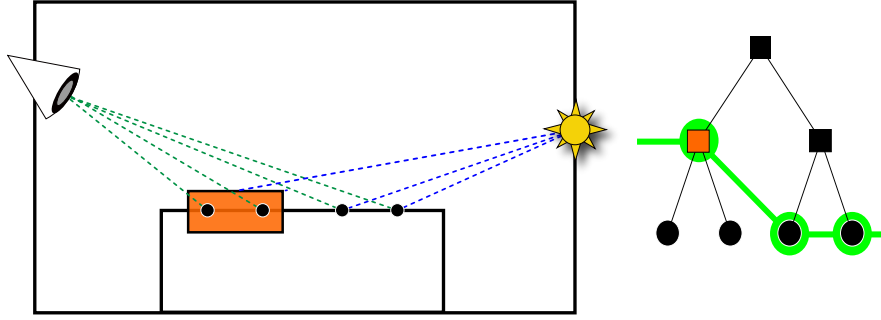


Figura 4.1: **Izquierda:** Escena para la que se generan 4 muestras desde la cámara e iluminada por una VPL. La energía reflejada desde la VPL puede ser aproximada para algunas de las muestras generadas. **Derecha:** Representación simplificada del árbol de muestras. La línea verde representa las hojas y nodos utilizados para aproximar la iluminación, formando un corte en el árbol. En este ejemplo, la iluminación en los nodos agrupados en el *cluster* naranja es calculada sólo para el *cluster*, y propagada a sus hijos al final del proceso.

4.2. Cálculo de la iluminación

Para calcular la iluminación global, este método recorre el árbol de muestras para cada VPL comenzando por el nodo raíz, creando un corte hasta alcanzar la precisión necesaria (ver Sección 4.2.1). Si un nodo es una hoja del árbol (una muestra) la energía reflejada se calcula de acuerdo a la Ecuación 2.5. En caso contrario, el *cluster* aproximará la visibilidad y la energía recibida para todas las muestras que contiene.

El test de visibilidad se calcula trazando un rayo desde la VPL al centro del *cluster*. Sin embargo, en superficies no alineadas con los ejes se obtienen malos resultados. Para solucionar este problema, durante la inicialización del árbol de muestras se refina la posición del centro de cada nodo mediando las posiciones de las muestras que contienen, mejorando notablemente la calidad del test como muestra la Figura 4.2.

Para aproximar la radiancia recibida mediante los *clusters* y evitar su cálculo en las hojas del árbol, hay que tener en cuenta algunos aspectos. Los nodos deben de almacenar la energía recibida por todas las luces virtuales. A su vez, la atenuación geométrica y la BRDF deben ser calculados para las hojas, siendo estos valores dependientes de la dirección de incidencia y de su orientación.

Para llevar a cabo esta tarea, se evaluaron distintas alternativas; en base al trabajo de Walter et al. [WABG06], es posible aproximar la radiancia reflejada a partir de una muestra elegida de forma aleatoria dentro del conjunto de puntos contenidos por cada *cluster*. También se valoró aproximar estadísticamente la contribución en todo el *cluster*, mediante el modelado de forma probabilística del conjunto de puntos discretos que forman el *cluster* [JHY⁺14, BN12]. No obstante, es difícil valorar cómo se vería afectada la precisión de la radiancia incidente utilizando este método.

Finalmente, se opta por utilizar una aproximación basada en los trabajos de Sloan et al. [SKS02], y Ramamoorthi y Hanrahan [RH01], almacenando la radiancia incidente sobre los *clusters* en armónicos esféricos. Este tipo de bases nos permiten almacenar de una forma eficiente la energía recibida manteniendo la componente direccional de la misma, de modo que permite su uso sobre BRDFs arbitrarias. Además, el uso de armónicos esféricos nos permite calcular la integral de la radiancia reflejada como el producto escalar entre la proyección en armónicos esféricos de la BRDF y de la radiancia incidente. Mediante esta propiedad, tras atravesar el árbol sumamos la energía almacenada en los nodos hasta las hojas, y calculamos la radiancia reflejada como la integral entre esta suma y la atenuación geométrica de cada muestra proyectada en armónicos esféricos. Para más detalles de esta operación consultar [Gre03].

La limitación de esta aproximación es que sólo resulta útil con materiales de baja frecuencia

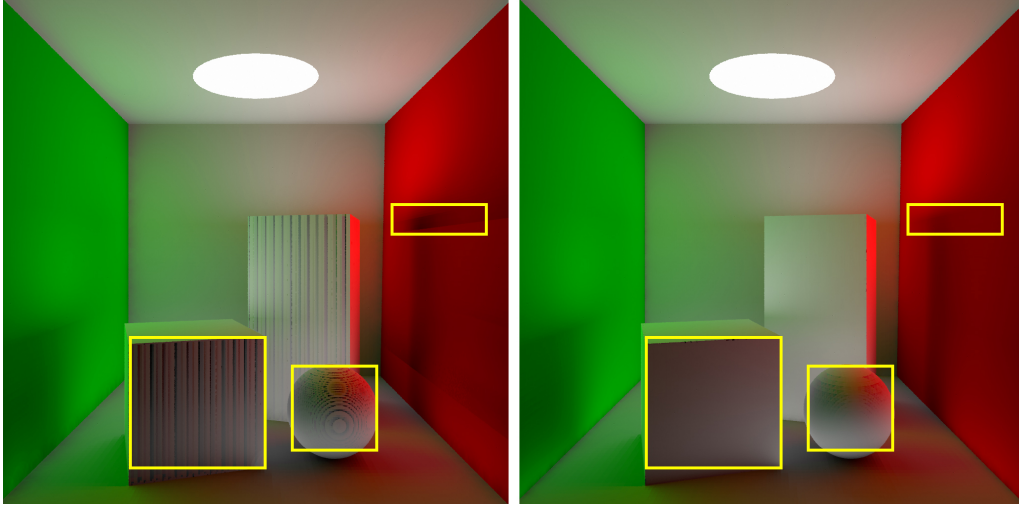


Figura 4.2: Comparación de los resultados obtenidos mediante el refinamiento de los centros de los *clusters* en una Cornell-Box difusa sin la componente directa. **Izquierda:** Resultado obtenido sin refinar el centro de los *clusters*. Las superficies no alineadas con los ejes y las zonas con oclusiones presentan artefactos que degradan la calidad de la imagen. **Derecha:** Resultado obtenido refinando el centro de los *clusters* como la media de las posiciones de las muestras que contienen. La precisión del test de visibilidad aumenta, eliminando la mayoría de los artefactos.

(en nuestro caso Lambertianos). Por ello, se implementa una versión alternativa que utiliza los *clusters* de muestras para aproximar únicamente la visibilidad, de forma que cuando un nodo es lo suficientemente preciso y es visible, la radiancia reflejada es calculada en las hojas. A pesar de ser menos eficiente, teniendo en cuenta que el cálculo de la visibilidad es la operación más costosa durante el cálculo de la iluminación, este método mantiene órdenes de complejidad sublineales con respecto del número de muestras.

4.2.1. Condiciones de parada

De forma similar al algoritmo presentado en el Capítulo 3, para determinar la precisión de un *cluster* se utiliza el valor de su ángulo sólido con respecto de las VPLs. Para calcular este valor, es necesario conocer el área que proyecta cada nodo hacia la VPL que se esté tratando. Dado que las muestras son puntos 0D y no disponen de área, no es posible aproximar este valor de la forma en que trabaja el algoritmo del Capítulo 3 mediante armónicos esféricos. Por ello, este valor se calcula a partir el área proyectada de la esfera que envuelve a cada *cluster* de acuerdo a la Ecuación 3.3.

Con el propósito de obtener más precisión en los *clusters* cercanos al punto de vista, el ángulo sólido es calculado también desde la posición de la cámara, de manera que se obtenga un nivel adicional de control en la calidad final de la solución generada. Sin embargo, en general la utilización del ángulo sólido desde las VPLs ofrece la precisión suficiente, dejando este parámetro como una condición de parada adicional, pero no indispensable.

Teniendo en cuenta que la jerarquización en un *octree* crea *clusters* subdividiendo el espacio, se ignora la orientación que pueden tener las muestras contenidas. En ciertas zonas de la escena como esquinas o superficies curvas, el test de visibilidad no es lo suficientemente preciso, obteniendo resultados con poca calidad sino se utilizan umbrales del ángulo sólido extremadamente pequeños. Para poder obtener un nivel de precisión extra en estas situaciones, se impone una condición adicional que permite refinar más los *clusters* con mucha variación geométrica. Estos casos son detectados mediante la utilización de una estructura en forma de cono que acota las normales contenidas. Estableciendo un valor máximo para el ángulo de apertura de dicho cono, se evita la utilización de *clusters* que contengan muestras que violen esta restricción como mostramos en la Figura 4.3.

Una técnica utilizada habitualmente es la utilización de mapas de normales para simular geometrías complejas en las superficies mediante la variación del cálculo de la iluminación. La utilización del cono de normales con estos materiales puede tener un gran impacto en el rendimiento. Por ello, esta condición se aplica sobre las normales de la geometría, dado que el cálculo de la visibilidad depende únicamente de la geometría.

Este método escala bien con el número de píxels (ver Capítulo 6). Sin embargo, tiene un coste lineal respecto del número de VPLs. En la siguiente sección presentamos un algoritmo escalable que combina el *clustering* jerárquico de luces virtuales (Capítulo 3) y de muestras de la cámara.

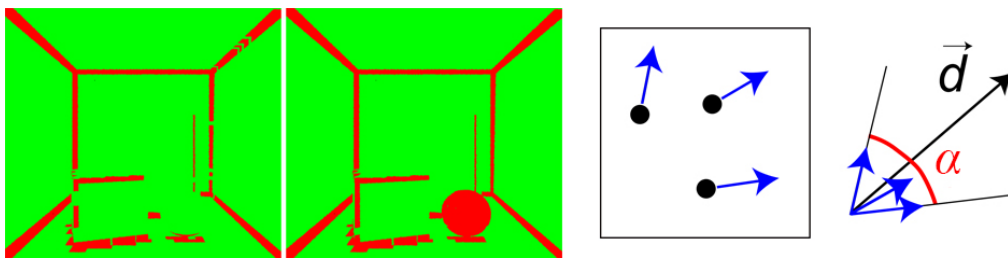


Figura 4.3: Efecto del test del cono de normales en la escena Cornell-Box. En verde se muestran las muestras que satisfacen esta restricción, y en rojo las que la violan, para ángulos máximos de 1.5 radianes (**Izquierda**) y 0.1 radianes (**Centro**). (**Derecha**) Ejemplo de cono de normales para tres puntos contenidos en un *cluster*. El cono queda definido por su dirección \vec{d} y su ángulo de apertura α .

Capítulo 5

Clustering bidireccional

Las técnicas descritas hasta ahora, se limitan a agrupar luces virtuales o muestras generadas desde la cámara de forma independiente. Aunque se consigue calcular la iluminación global de una escena de una forma más eficiente, si se combinan ambas estrategias es posible obtener costes sublineales con respecto de la cantidad de VPLs y muestras.

En este capítulo, se propone un nuevo algoritmo que combina el *clustering* jerárquico de luces virtuales aproximando la iluminación indirecta [WFA⁺05, Chr08] (Capítulo 3), y el de muestras desde la cámara descrito en el Capítulo 4, de modo que el coste del algoritmo sea de orden ($O(\log(M + N))$).

Partiendo de los árboles de VPLs y muestras, el algoritmo trabaja con una jerarquía implícita que representa todas las posibles relaciones entre nodos de luces y nodos de puntos a iluminar como muestra la Figura 5.1. Cada nodo de esta jerarquía empareja elementos del árbol de VPLs y del árbol de muestras formando pares de nodos. Para aproximar la iluminación global, comenzando a partir del par formado por los nodos raíz de ambos árboles, éstos son refinados creando nuevos pares, hasta cumplir las condiciones de parada. El resultado es un corte en la jerarquía formado por el conjunto de pares utilizados para calcular la iluminación global, aproximando tanto la radiancia reflejada por las superficies como la recibida por las muestras.

5.1. Preproceso

Los datos de entrada del algoritmo son el conjunto de VPLs que modelan la iluminación indirecta y las muestras generadas desde la cámara. Crear un árbol a partir de todos los posibles pares de VPLs y muestras es un proceso costoso tanto en tiempo como en memoria. Por ello, se utiliza una jerarquía implícita sobre el espacio de los posibles pares, formada por el producto cartesiano de los árboles independientes de luces y muestras. El nodo raíz representa el espacio de todos los posibles pares de nodos, mientras que las hojas representan pares individuales de VPLs y muestras. Mediante la relación de los dos árboles, es posible disponer de una jerarquía, sin la necesidad de construirla explícitamente como muestra la Figura 5.1.

5.2. Cálculo de la iluminación

Para calcular la iluminación global, el algoritmo selecciona de forma adaptativa un conjunto de pares de nodos que permiten aproximar la iluminación global. Este conjunto forma un corte en la jerarquía donde, para cada nodo hoja, el conjunto de caminos desde la raíz hasta dicha hoja contiene un único nodo del corte. Para crear el corte, el algoritmo parte del par formado por las raíces de los árboles de luces y muestras, y crea nuevos pares hasta alcanzar las condiciones de parada (Sección 5.2.2). A la hora de calcular la iluminación global, se procede de forma análoga a

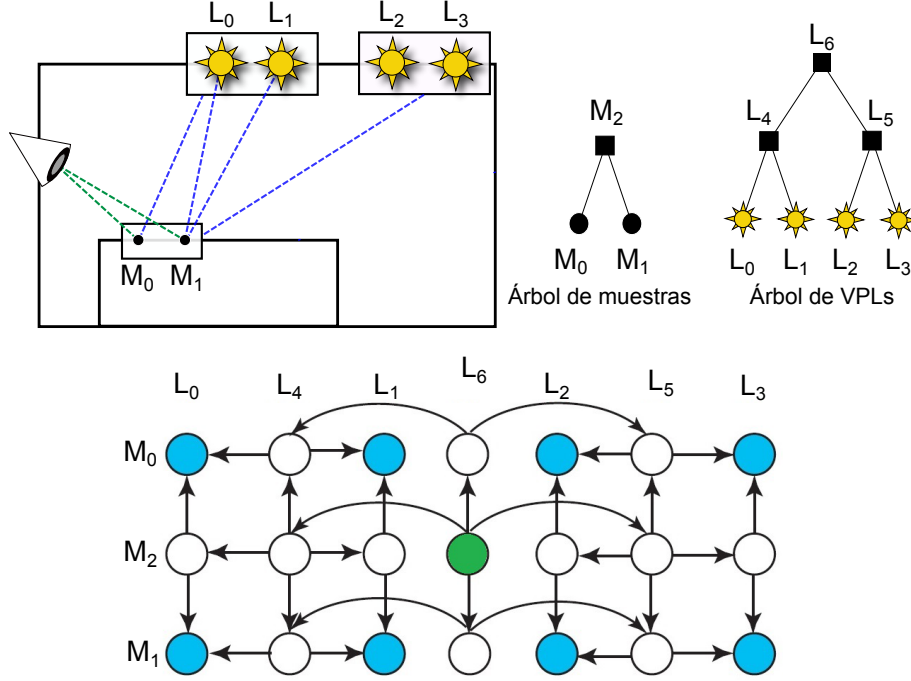


Figura 5.1: Figura tomada de [WABG06] **Arriba izquierda:** Escena con dos muestras y cuatro VPLs. La iluminación global es calculada utilizando los pares (M_0, L_4) , (M_1, L_0) , (M_1, L_1) y (M_2, L_5) . **Arriba derecha:** Árboles independientes de VPLs y muestras de la escena. **Abajo:** Árbol implícito de VPLs y muestras. Cada nodo representa un posible par de *clusters* de VPLs y muestras. Los nodos azules representan las hojas de la jerarquía mientras que el nodo verde representa la raíz.

la descrita en los Capítulos 3 y 4. Sin embargo, a la hora de refinar se pueden dar cuatro situaciones distintas; una muestra puede recibir la radiancia de un *cluster* de VPLs o de una única VPL, y un *cluster* de muestras puede ser iluminado por un *cluster* de VPLs o una única VPL. Mostramos estas situaciones en la Figura 5.2. En la Sección 5.2.2 describimos el proceso de refinamiento del árbol en cada una de estas situaciones.

5.2.1. Visibilidad

Mientras que la función de visibilidad de punto a *cluster* está descrita en los Capítulos 3 y 4, estas aproximaciones no se pueden realizar entre *clusters* donde el valor de la función de visibilidad entre *clusters* $V(l, m)$ está definido como:

$$V(l, m) = \int_{\Omega_l} \int_{\Omega_m} V(m, l) dm dl. \quad (5.1)$$

Para evitar el trazado de múltiples rayos, en esta versión del algoritmo se procede de manera análoga a la descrita en el Capítulo 3 realizando un muestreo aleatorio del *cluster* de luces virtuales desde el centro del *cluster* de muestras con una único rayo. Aunque en general se consiguen buenos resultados, en un trabajo futuro se buscarán alternativas para resolver la integral de la visibilidad de forma eficiente [RAMN12], con el fin de mejorar los resultados de este test y permitir la utilización de condiciones de corte menos restrictivas.

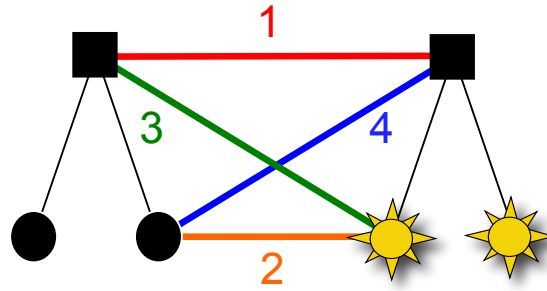


Figura 5.2: Representación de los posibles pares que puede ser formados durante el recorrido de la jerarquía de luces y muestras. **(1)** Par formado por un *cluster* de VPLs y un *cluster* de muestras. En esta situación, la energía recibida por el nodo de muestras se calcula a partir de los coeficientes de los armónicos esféricos del *cluster* de VPLs. Posteriormente, la energía reflejada por cada muestra hacia la cámara se determina de la misma manera en que trabaja el algoritmo de *clustering* de muestras. **(2)** Par formado por nodos hoja de VPLs y muestras. Se calcula la energía de la VPL modelada como VSL de acuerdo a la Ecuación 3.5. **(3)** Par formado por un nodo hoja del árbol de VPLs y un *cluster* de muestras. Para determinar la luz recibida por el conjunto de muestras, se aplica el algoritmo de *clustering* de muestras para cada uno de las luces virtuales del *cluster* de VPLs. **(4)** Par formado por un nodo hoja del árbol de muestras y un *cluster* de VPLs. Para determinar la contribución sobre las muestras individuales, se aplica el algoritmo de *clustering* de muestras.

5.2.2. Condiciones de parada y refinado de pares

Las condiciones de parada utilizadas en este método, son equivalentes a las expuestas en los Capítulos 3 y 4, utilizando el valor del ángulo sólido de los *clusters* calculado con respecto de los centros de los nodos, además de la restricción del cono de normales para *clusters* de muestras. Para que un par pueda ser utilizado para aproximar la iluminación, se deben de cumplir todas las restricciones impuestas. En caso contrario, el par se reemplaza por nuevos pares, creados refinando uno de los dos miembros, seleccionado de acuerdo a una sencilla heurística:

1. Si el nodo de muestras viola la condición del ángulo máximo del cono de normales, se crean nuevos pares a partir del *cluster* de VPLs y los hijos del *cluster* de muestras.
2. En caso contrario, si se violan las condiciones para el ángulo sólido máximo, la heurística trata de refinar aquel nodo que minimice en mayor grado el error introducido. Definiendo Ω_L como el ángulo sólido máximo de un *cluster* de VPLs, Ω_S el ángulo sólido de la envolvente de un *cluster* de VPLs, Ω_m el ángulo sólido de un *cluster* de muestras desde la posición de un nodo del árbol de VPLs, y Ω_{mc} el ángulo sólido de un *cluster* de muestras desde el punto de vista, la heurística se define como:

$$h_L = \max\left(\frac{\Omega_L}{\Omega_{Lmax}}, \frac{\Omega_S}{\Omega_{Smax}}\right) \quad (5.2)$$

$$h_m = \max\left(\frac{\Omega_m}{\Omega_{mmax}}, \frac{\Omega_{mc}}{\Omega_{mcmax}}\right) \quad (5.3)$$

$$Nodo = \begin{cases} \text{Nodo VPLs} & \text{si } h_L > h_m \\ \text{Nodo muestras} & \text{si } h_L \leq h_m \end{cases} \quad (5.4)$$

El algoritmo desarrollado es capaz de combinar los métodos de *clustering* de muestras y de VPLs, obteniendo complejidades sublineales ($O(\log(M + N))$). En el Capítulo 6 comparamos los resultados de rendimiento y calidad obtenidos con respecto de los otros dos métodos desarrollados.

Capítulo 6

Resultados

La implementación realizada de los distintos algoritmos presentados en este proyecto ha sido realizada bajo el *framework* PBRT [PH10], implementando tanto versiones monohilo como multi-hilo.

6.1. Análisis de rendimiento

En esta sección, mostramos los resultados de rendimiento obtenidos para los tres algoritmos, en la escena *Cornell-Box* [CG85], variando las condiciones de corte y comparándolos con el coste de una evaluación lineal de todos los pares de muestras y VPLs. Para la realización de las pruebas, se utilizan versiones que no paralelizan el proceso con el fin de poder comparar de forma efectiva el rendimiento de los distintos métodos.

Las tablas con los resultados del coste en segundos pueden encontrarse en el Anexo B.

6.1.1. Análisis de rendimiento de PBGI

A continuación comparamos el rendimiento del algoritmo de *clustering* de VPLs presentado en el Capítulo 3 con respecto de una evaluación lineal de todo el conjunto de VPLs, fijando una resolución de 256x256 y una muestra por píxel (256x256x1), un valor $\Omega_S = 0,1$ sr y variando tanto la cantidad de VPLs como el valor de Ω_L . La Figura 6.1 y la Tabla B.1 muestran los resultados obtenidos.

Como se puede observar, el algoritmo de *clustering* de VPLs es capaz de calcular la iluminación global de una forma mucho más eficiente en comparación a calcular todas las VPLs individualmente. En los resultados obtenidos, observamos que incrementos del número de VPLs mantienen un tiempo de renderizado estable, llegando incluso a disminuir. Esto es debido a que un incremento en el número de VPLs permite la creación de un mayor número de *clusters*, siendo el cálculo de la energía a partir de un nodo una operación más rápida. A partir de cierto número de VPLs, los cortes creados en la jerarquía tienen una profundidad similar, por lo que el tiempo de renderizado se mantiene estable al no alcanzar ramas más profundas para un valor Ω_L dado. Con valores grandes $\Omega_L = 0,1$ sr, el algoritmo alcanza rápidamente la profundidad máxima del corte, mientras que con valores más pequeños $\Omega_L = 0,001$ sr, podemos apreciar mejor su comportamiento. Inicialmente aumenta el tiempo de renderizado al aumentar tanto el número de *clusters* utilizados como el número de hojas. A partir de cierto número, el número de hojas alcanzadas disminuye, y la cantidad de nodos aumenta provocando que el rendimiento del algoritmo mejore. La Figura 6.2 muestra la utilización de hojas y nodos, reflejando el comportamiento del método.

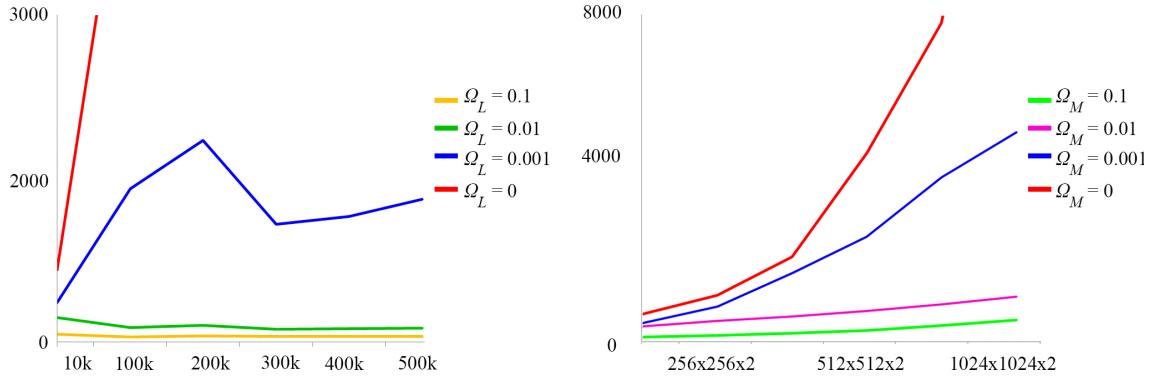


Figura 6.1: **Izquierda** Gráfica comparativa del tiempo en segundos para el algoritmo de *clustering* de VPLs y el cálculo de la iluminación evaluando todo el conjunto de VPLs ($\Omega_L = 0$ sr). **Derecha** Gráfica comparativa del tiempo en segundos para el algoritmo de *clustering* de muestras con $\alpha = 0,1$ rad y el cálculo de la iluminación con todo el conjunto de VPLs y muestras ($\Omega_L = 0$ sr).

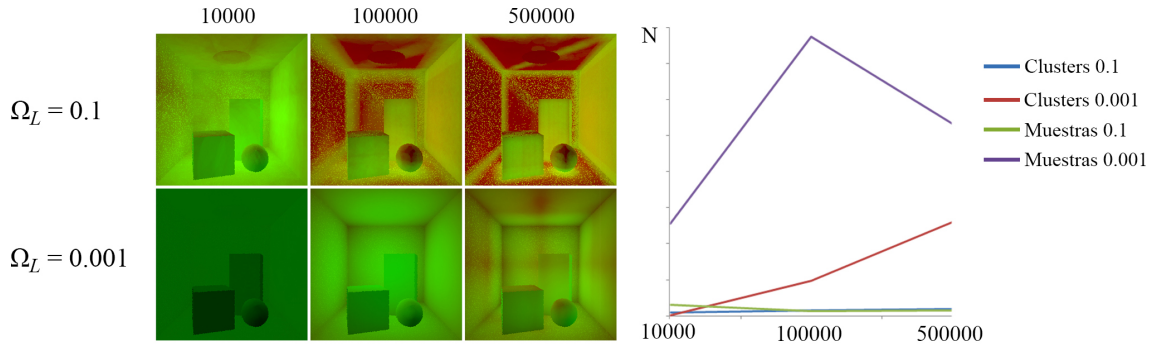


Figura 6.2: **Izquierda**: Representación de la cantidad de *clusters* utilizados para $\Omega_L = 0,1$ sr y $\Omega_L = 0,01$ sr y un número de VPLs de 10k, 100k y 500k. Colores rojos representan la cantidad de *clusters* utilizados, mientras que colores verdes representan zonas donde se alcanzan las hojas del árbol. **Derecha**: Gráfica comparativa de la cantidad de *clusters* y VPLs utilizados.

6.1.2. Análisis de rendimiento del método de *clustering* de muestras

Para comparar la eficiencia del algoritmo desarrollado en el Capítulo 4 con respecto del cálculo de la iluminación con todo el conjunto de VPLs y muestras, fijamos una cantidad de 10000 VPLs, un valor $\Omega_C = 0,1$ sr y variamos la resolución de la imagen, el ángulo sólido Ω_M y el valor de α . La Figura 6.1 y la Tabla B.2 muestran los resultados obtenidos.

Podemos observar como el algoritmo de *clustering* de muestras es capaz de reducir el tiempo de cálculo a medida que aumenta el número de muestras trazadas desde la cámara. Cabe destacar el aumento del coste en la versión que se limita a aproximar el cálculo de visibilidad dado que se ha de propagar hasta las hojas la VPL tratada en cada momento, teniendo un impacto mayor a medida que se incrementa la profundidad del árbol. A partir de 512x512x2 muestras la aproximación mediante armónicos esféricos con $\Omega_M = 0,001$ sr es capaz de obtener resultados superiores con respecto de la técnica que aproxima la visibilidad con $\Omega_M = 0,01$ sr. En cuanto a la escalabilidad del algoritmo, al aumentar la resolución, al contrario que el método de *clustering* de VPLs, el tiempo de cálculo se incrementa. Esto se debe a que al aumentar la cantidad de muestras, tanto el número de evaluaciones de *clusters* como de hojas crece, como mostramos en la Figura 6.3. Este aumento es producido por el parámetro α , produciendo que en ciertas geometrías se mantenga una alta tasa de utilización de hojas. Aunque en la escena de pruebas no tiene un gran impacto en el rendimiento, con geometrías complejas puede afectar negativamente al tiempo de cálculo.

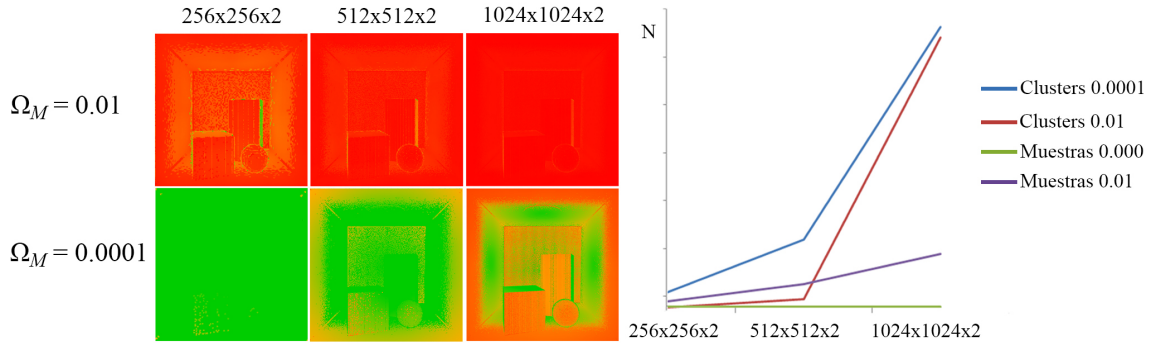


Figura 6.3: **Izquierda:** Representación de la cantidad de *clusters* utilizados para $\Omega_M = 0,01$ sr y $\Omega_M = 0,0001$ sr y una cantidad de muestras de 256x256x2, 512x512x2 y 1024x1024x2. Colores rojos codifican la cantidad de *clusters* incrementándose su tonalidad a medida que se profundiza más en el árbol, mientras que colores verdes representan zonas donde se alcanzan las hojas del árbol. **Derecha:** Gráfica comparativa de la cantidad de *clusters* y hojas utilizados con valores $\Omega_M = 0,01$ sr, y $\Omega_M = 0,0001$ sr

6.1.3. Análisis de rendimiento del método bidireccional

A continuación analizamos el comportamiento del algoritmo de *clustering* de luces virtuales y muestras, comparándolo con PBGI y el método de *clustering* de muestras. En el primer caso, comparamos el escalado de ambos algoritmos a medida que aumentamos la resolución de la imagen, fijando $\Omega_L = 0,01$ sr para ambas técnicas y con valores $\Omega_M = 0,001$ sr y $\Omega_M = 0,0001$ sr. En el segundo caso, comparamos el rendimiento de los métodos aumentando la cantidad de VPLs con un valor $\Omega_M = 0,0001$ sr para el método de *clustering* de muestras. La Figura 6.4 recoge los resultados con valores $\Omega_M = 0,0001$ sr mientras que en la Tabla B.3 se pueden encontrar medidas adicionales.

En las pruebas realizadas, se puede observar como la técnica bidireccional obtiene rendimientos superiores con respecto a PBGI y el método de *clustering* de muestras. En el caso de PBGI, a bajas resoluciones obtiene peores resultados debido a que la cantidad de muestras que pueden ser agrupadas en *clusters* es pequeña y el coste de atravesar la jerarquía de luces y muestras es mayor. A medida que aumenta la cantidad de muestras, el algoritmo bidireccional mejora los tiempos de cálculo con respecto de la técnica de *clustering* de VPLs. En cuanto al método de *clustering* de muestras, somos capaces de obtener un amplio margen de mejora.

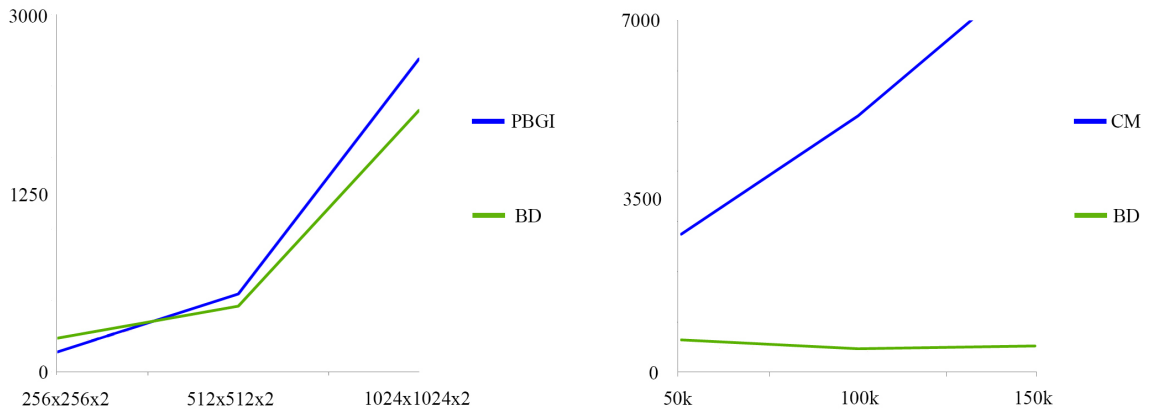


Figura 6.4: **Izquierda:** Gráfica comparativa del tiempo en segundos entre PBGI con $\Omega_L = 0,01$ sr y el algoritmo de *clustering* de VPLs y muestras con $\Omega_L = 0,01$ sr y $\Omega_M = 0,0001$ sr. **Derecha:** Gráfica comparativa del tiempo medido en segundos entre el algoritmo de *clustering* de muestras(CM) para $\Omega_M = 0,0001$ sr y el algoritmo de *clustering* de VPLs y muestras con $\Omega_M = 0,0001$ sr y $\Omega_V = 0,01$ sr.

6.2. Análisis de calidad y parámetros

A continuación, presentamos el análisis de la calidad de imagen obtenida con los tres algoritmos desarrollados con el objetivo de determinar los parámetros óptimos para sus condiciones de parada. Para la realización de las pruebas, se utiliza la escena Cornell-Box sin la componente de iluminación directa, con 100000 VPLs y una resolución de 1024x1024 y 2 muestras por píxel.

6.2.1. Algoritmo de *clustering* de VPLs

En la Figura 6.5 mostramos los resultados obtenidos para el método PBGI con ángulos sólidos máximos $\Omega_L = 0,1$ sr y $\Omega_L = 0,01$ sr fijando un valor $\Omega_S = 0,1$ sr. Un umbral de $\Omega_L = 0,1$ sr ofrece resultados con más ruido debido a las oclusiones, el cual se hace más notable conforme aumenta la complejidad de la escena. Por el contrario, un valor $\Omega_L = 0,01$ sr permiten minimizar este efecto, manteniendo buenos niveles de escalabilidad con el número de VPLs.

6.2.2. Algoritmo de *clustering* de muestras

Para el algoritmo de *clustering* de muestras, utilizamos valores $\Omega_M = 0,01$ sr, $\Omega_M = 0,001$ sr y $\Omega_M = 0,0001$ sr para el ángulo sólido y valores $\alpha = 1,5$ rad y $\alpha = 0,1$ rad. La Figura 6.6 muestra los resultados obtenidos. Nótese la necesidad de utilizar umbrales menores con respecto al método PBGI para obtener buenos resultados. Valores de $\Omega_M = 0,01$ sr y $\Omega_M = 0,001$ sr presentan ruido estructurado, especialmente en zonas con oclusiones, mientras que con un valor máximo $\Omega_M = 0,0001$ sr este efecto es prácticamente inapreciable.

Adicionalmente, el parámetro α tiene consecuencias más notables con valores altos Ω_M , sin embargo en escenas con geometría detallada, resulta de utilidad con el fin de obtener un mayor grado de precisión incluso con ángulos sólidos pequeños.

6.2.3. Algoritmo de *clustering* de VPLs y muestras

Para la técnica bidireccional, analizamos la calidad de los resultados estableciendo valores $\Omega_L = 0,1$ sr y $\Omega_L = 0,01$ sr para los *clusters* de VPLs y valores para los *clusters* de muestras $\Omega_M = 0,001$ sr y $\Omega_M = 0,0001$ sr, mostrando en qué medida afecta cada uno a la calidad de imagen. Los resultados obtenidos se muestran en la Figura 6.7. Minimizando el parámetro Ω_M obtenemos mejores resultados que minimizando el valor de Ω_L . Sin embargo dada la poca precisión del test de visibilidad entre *clusters*, necesitaremos ser restrictivos y utilizar valores de $\Omega_L = 0,01$ sr y $\Omega_M = 0,0001$ sr.

Cabe destacar como, al contrario que la técnica de *clustering* de muestras, el efecto del ángulo máximo entre las normales de las muestras es mayor, aunque las superficies con normales muy dispares en general siguen presentando artefactos.

6.2.4. Análisis de error

Para analizar el error introducido por cada técnica, calculamos el MSE en base a un *ground truth* calculado evaluando todas las muestras y VPLs. En la Figura 6.8 mostramos los resultados obtenidos.

Para PBGI y el método bidireccional obtenemos un mayor error. Este es introducido por la aproximación de la energía mediante armónicos esféricos. Sin embargo, estos métodos generan imágenes con menos artefactos. Por el contrario, los algoritmos de *clustering* de muestras presentan un valor menor para el error, debido a la utilización de las VPLs de forma individual y el hecho de necesitar ángulos sólidos máximos más bajos para obtener resultados aceptables.

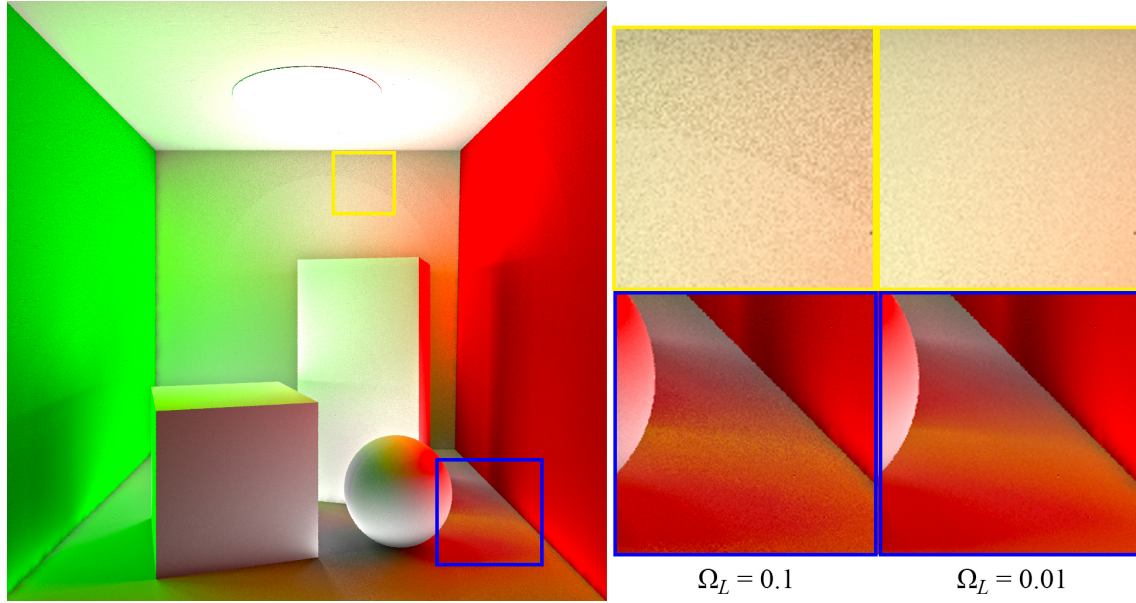


Figura 6.5: Comparación de la calidad de imagen para el algoritmo de *clustering* de VPLs con $\Omega_L = 0,1$ sr y $\Omega_L = 0,01$ sr.

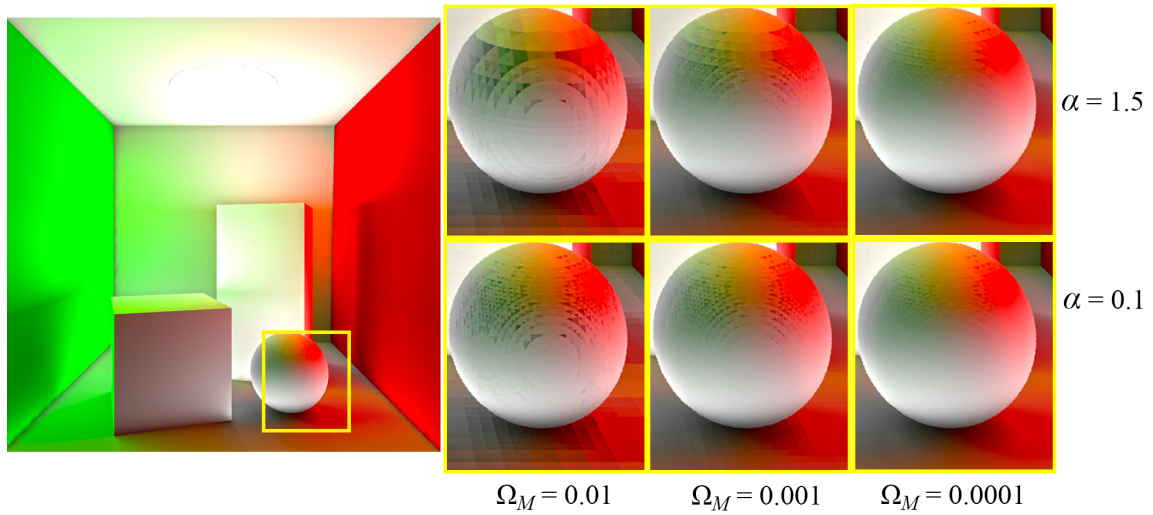


Figura 6.6: Comparación de la calidad de imagen para el algoritmo de *clustering* de muestras con valores $\Omega_M = 0,01$ sr, $\Omega_M = 0,001$ sr, $\Omega_M = 0,0001$ sr, y ángulos máximos para el cono de normales $\alpha = 1,5$ rad y $\alpha = 0,1$ rad.

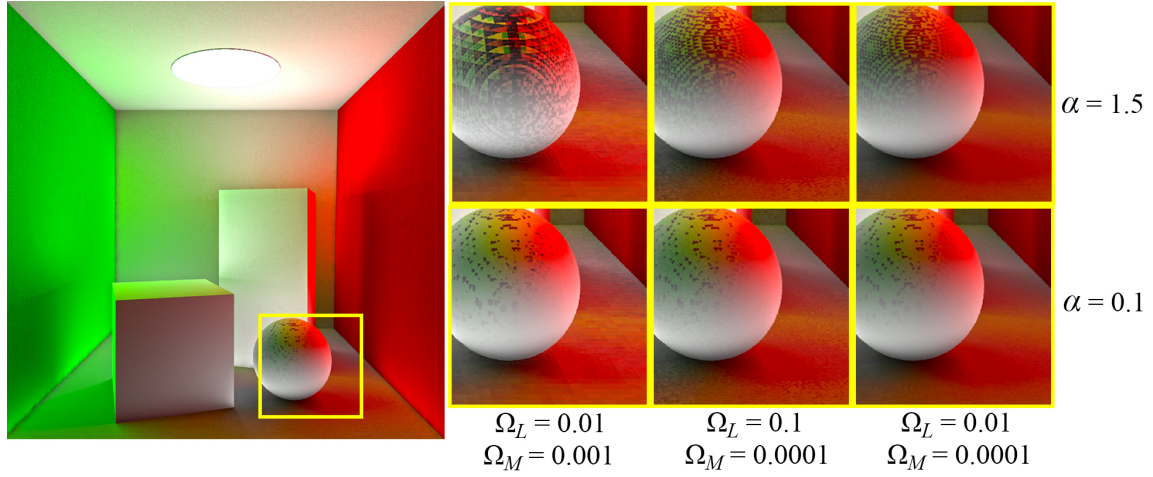


Figura 6.7: Comparación de la calidad de imagen para el algoritmo de *clustering* de VPLs y muestras variando los parámetros Ω_L , Ω_M y α .

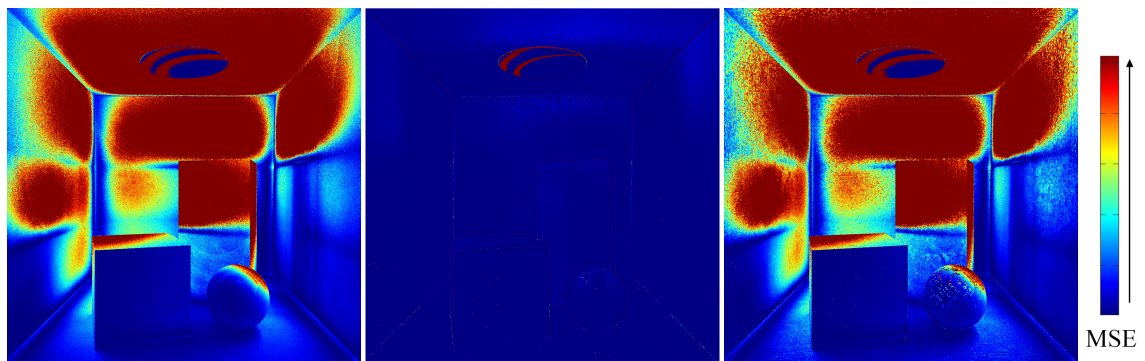


Figura 6.8: MSE para el algoritmo de *clustering* de VPLs (**Izquierda**), *clustering* de muestras (**Centro**) y *clustering* de VPLs y muestras (**Derecha**).

6.3. Resultados adicionales

A continuación se presentan resultados adicionales en escenas con una complejidad superior. Teniendo en cuenta la escalabilidad lineal con respecto del número de VPLs del método de *clustering* de muestras, se utilizará PBGI y el método bidireccional. En el Anexo B se recogen las estadísticas del rendimiento obtenido para las distintas escenas calculadas.

En los resultados obtenidos, a la izquierda presentamos la imagen completa renderizada con el método bidireccional, a excepción de la escena *Bunny*. A la derecha comparamos la calidad entre los dos métodos en algunas zonas de las escenas, siendo la columna izquierda el método de *clustering* de VPLs, y la columna derecha el método de *clustering* de VPLs y muestras.

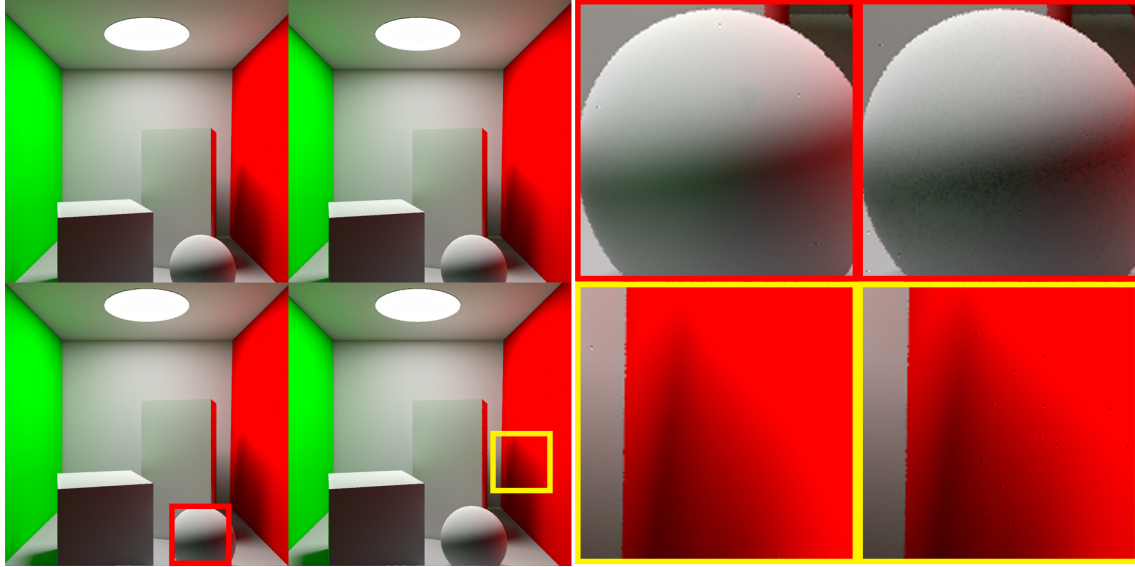


Figura 6.9: Escena *light field*.

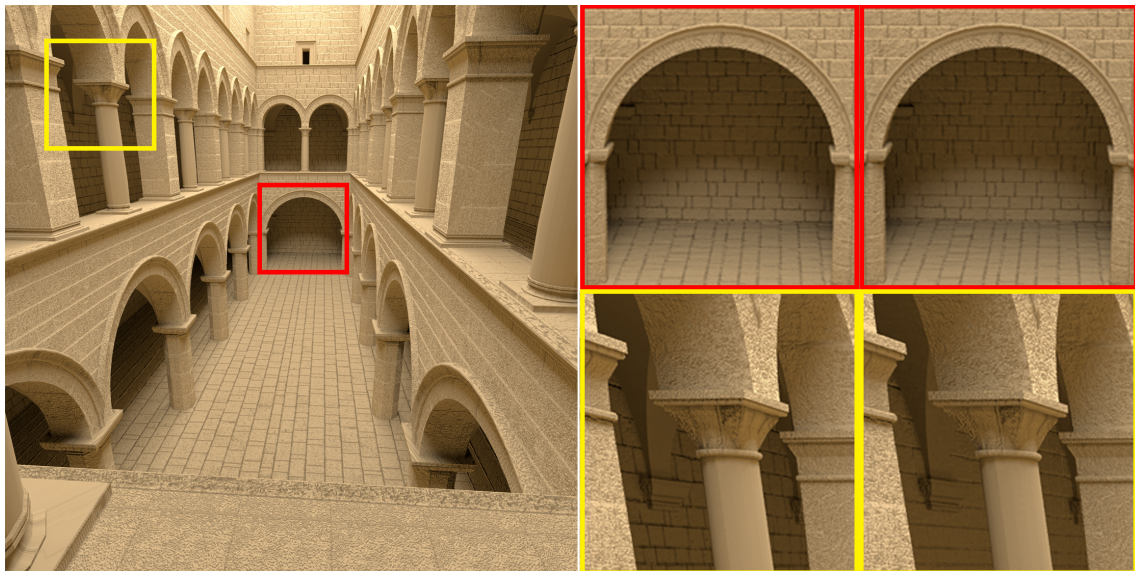


Figura 6.10: Escena *Sponza*.

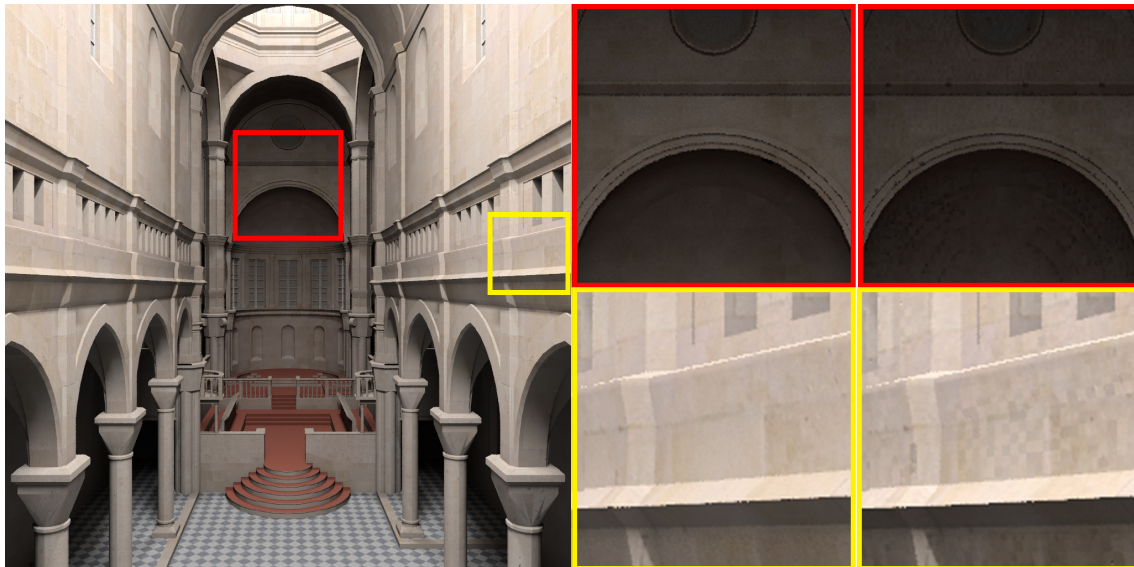


Figura 6.11: Escena *Sibenik*.

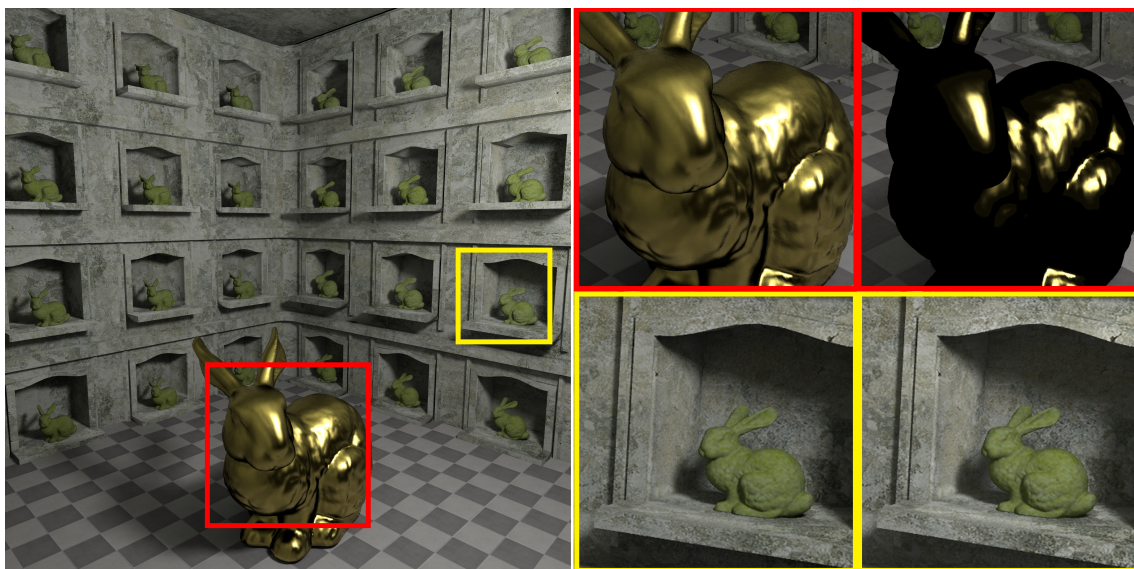


Figura 6.12: Escena *Bunny*. **(Izquierda)**: Resultado obtenido con PBGI. **(Derecha)**: Nótese como el algoritmo bidireccional no es capaz de calcular la iluminación global sobre materiales de alta frecuencia (Conejo central).

Figura 6.13: Escena *San Miguel*.

Capítulo 7

Conclusiones

En este proyecto, hemos desarrollado un nuevo algoritmo que permite el cálculo eficiente de la iluminación global (difusa) utilizando técnicas de clustering en espacio de luces y de muestras de cámara. Como punto de partida, hemos implementado PBGI [Chr08], un algoritmo de iluminación global basado en *clustering* de VPLs utilizado en producción cinematográfica (Capítulo 3). Para ello, lo hemos adaptado a un sistema basado en trazado de rayos [PH10] y hemos resuelto los problemas comunes a las técnicas basadas en luces virtuales como las singularidades y la presencia de ruido estructurado, para lo cual hemos adaptado las VSLs [HKWB09] a PBGI, además de otros problemas derivados de la aproximación introducida por la utilización de *clusters*.

Posteriormente, hemos diseñado una técnica de renderizado capaz de agrupar las muestras generadas desde uno o múltiples de vista, obteniendo costes sublineales con respecto de su número. A diferencia de trabajos previos [WABG06, WHB⁺13], nuestra técnica trabaja con el conjunto total de muestras en la escena, lo que nos permite aprovechar mejor la coherencia presente en las muestras desde la cámara.

Finalmente, hemos propuesto un nuevo método capaz de combinar el algoritmo de *clustering* de luces virtuales implementado, y el nuevo método de *clustering* de muestras de la cámara, permitiendo obtener órdenes de complejidad sublineales ($(O(\log(M + N)))$). Sus resultados han sido estudiados en varios escenarios, observando que en la mayoría de casos es posible obtener calidades similares a PBGI en un tiempo menor.

7.1. Discusión y trabajo futuro

Pese a que la técnica presentada es capaz de calcular la iluminación global de una forma eficiente, con órdenes de complejidad sublineales con respecto del número de luces virtuales y de muestras, presenta algunas limitaciones que deberán ser tratadas en trabajos futuros.

En la implementación llevada a cabo, únicamente calculamos la iluminación global proveniente del primer rebote de la luz en las superficies. Para extenderlo a múltiples rebotes sería necesario repetir el proceso de iluminación iterativamente. Nótese que nuestro método, al ser sublineal tanto en las luces como en los puntos de sombreado, permite que este proceso sea muy eficiente. Otra alternativa sería la generación de las luces virtuales mediante trazado estocástico de partículas [Kel97]; en este caso, la extensión sería trivial.

Asimismo, la utilización de armónicos esféricos funciona bien para aproximar la iluminación de baja frecuencia, sin embargo necesitamos mayor precisión para poder tratar con materiales de mayor frecuencia. La solución trivial para manejar estos materiales es resolver la integral estocásticamente, muestreando la BRDF e introduciendo las muestras generadas dentro de la jerarquía de muestras desde la cámara.

Otra opción sería aumentar el número de bandas de armónicos esféricos a almacenar, basado

en la frecuencia de la función de reflectancia [DHS⁺05, LALD12]; sin embargo, para escenas con muchas superficies especulares esta aproximación sería muy ineficiente en memoria, y además podría dar problemas de *ringing* [SHHS03].

Adicionalmente, el uso de técnicas basadas en VPLs es muy prometedor para renderizar jerárquicamente materiales translúcidos [JMLH01]; explorar cómo incorporar este tipo de materiales a nuestro algoritmo bidireccional es un interesante trabajo futuro a explorar [AWB08].

Atendiendo a los resultados, en la mayoría de las imágenes renderizadas, conseguimos reducir el tiempo de renderizado. Sin embargo, en algunas escenas (*Bunny*), el algoritmo de *clustering* de VPLs resulta más eficaz que el método bidireccional. Esto es debido a las diferencias de normales dentro de los *clusters* de muestras, que hacen que el algoritmo tenga que refinar significativamente el árbol, de modo que los costes fijos de la técnica se vuelven dominantes. Por ello, evaluaremos el uso de otro tipo de jerarquías que agrupen las muestras teniendo en cuenta su orientación con el propósito de generar *cluster* de mayor calidad. La utilización de técnicas de prefiltrado que nos permitan aproximar el contenido de un *cluster* de muestras mediante una distribución estadística, puede resultar interesante para calcular la iluminación en un *cluster* con un menor grado de error [BN12].

Finalmente, el algoritmo está limitado por la memoria consumida, debido a que almacenamos los árboles de VPLs y muestras desde la cámara. Para solucionar este problema en escenas de gran complejidad, limitadas por la memoria del sistema, se pueden adaptar técnicas de compresión [BB12] y de almacenamiento *out-of-core* [KTO11] existentes para VPLs al árbol de muestras de cámara. Nótese que los resultados mostrados en la Sección 6.3 han sido renderizados con versiones multihilo, donde el algoritmo de *clustering* de VPLs es capaz de repartir mejor la carga de trabajo entre los procesadores, por lo que la mejora en el *scheduling* del método bidireccional es un aspecto importante a mejorar.

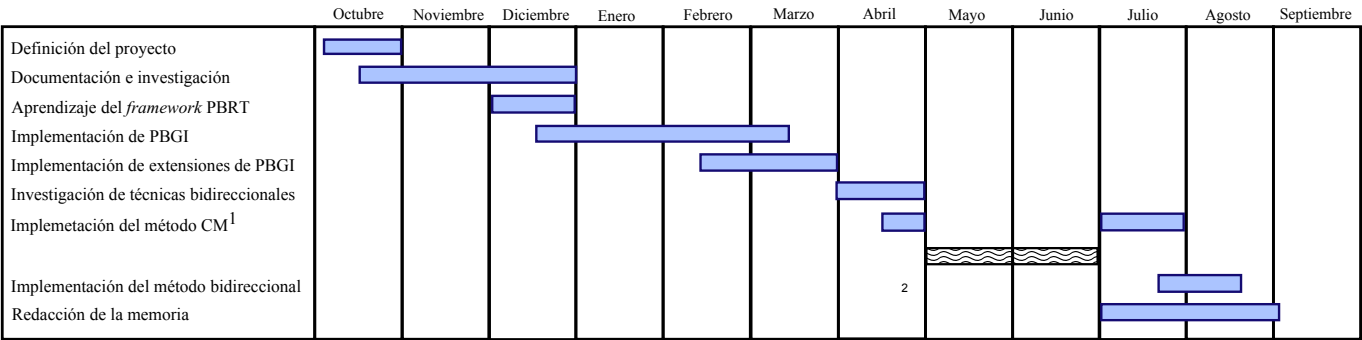
Las heurísticas de error de los algoritmos presentados están basadas en el ángulo sólido entre puntos y *clusters*. Aunque efectivas, estas técnicas son demasiado conservativas, y no son intuitivas a la hora de ser definidas por el usuario. El desarrollo de nuevas heurísticas que utilicen información basada en la frecuencia de las superficies a iluminar [FPG01, DBD⁺07], de las BRDFs en la escena [LALD12], o haciendo uso de las limitaciones del sistema visual humano [RFBW07, HČA⁺12], es una línea de trabajo futuro muy interesante.

7.2. Conclusión personal

A nivel personal, la realización de este proyecto ha supuesto un reto que me ha permitido iniciarme en el campo de los gráficos por ordenador, y profundizar en las bases físicas en las que se fundamentan. Trabajar junto a la gente del *Graphics and Imaging Lab* me ha permitido conocer de primera mano la forma de trabajo en investigación, dándome la oportunidad de colaborar con el artículo *A Framework for Transient Rendering*, aceptado en la conferencia internacional SIGGRAPH Asia 2014, desarrollado en colaboración con Jaroz Wojciech investigador jefe del grupo de gráficos en Disney Research, Zürich. La relación con Adrián y Diego ha sido realmente buena además de con el resto de compañeros del laboratorio, siendo grandes profesionales y mejores personas, dispuestas a ayudar en todo momento.

7.3. Diagrama temporal del proyecto

A continuación presentamos la evolución temporal del proyecto, iniciado en octubre de 2013, hasta su finalización en septiembre de 2014.



¹Algoritmo de *clustering* de muestras

Figura 7.1: Distribución temporal de las tareas realizadas en este proyecto.

Bibliografía

- [AWB08] Adam Arbree, Bruce Walter, and Kavita Bala, *Single-pass scalable subsurface rendering with lightcuts*, Computer Graphics Forum, vol. 27, Wiley Online Library, 2008, pp. 507–516.
- [BB12] Bert Buchholz and Tamy Boubekeur, *Quantized point-based global illumination*, Computer Graphics Forum, vol. 31, Wiley Online Library, 2012, pp. 1399–1405.
- [BN12] Eric Bruneton and Fabrice Neyret, *A survey of nonlinear prefiltering methods for efficient and accurate surface shading*, Visualization and Computer Graphics, IEEE Transactions on **18** (2012), no. 2, 242–260.
- [Bun05] Michael Bunnell, *Dynamic ambient occlusion and indirect lighting*, Gpu gems **2** (2005), no. 2, 223–233.
- [CG85] Michael F Cohen and Donald P Greenberg, *The hemi-cube: A radiosity solution for complex environments*, ACM SIGGRAPH Computer Graphics, vol. 19, ACM, 1985, pp. 31–40.
- [Chr08] Per Christensen, *Point-based approximate color bleeding*, Pixar Technical Notes **2** (2008), no. 5, 6.
- [CPC84] Robert L Cook, Thomas Porter, and Loren Carpenter, *Distributed ray tracing*, ACM SIGGRAPH Computer Graphics, vol. 18, ACM, 1984, pp. 137–145.
- [DBB06] Philip Dutre, Kavita Bala, and Philippe Bekaert, *Advanced global illumination*.
- [DBD⁺07] George Drettakis, Nicolas Bonneel, Carsten Dachsbacher, Sylvain Lefebvre, Michael Schwarz, and Isabelle Viaud-Delmon, *An interactive perceptual rendering pipeline using contrast and spatial masking*, Proceedings of the 18th Eurographics conference on Rendering Techniques, Eurographics Association, 2007, pp. 297–308.
- [DHS⁺05] Frédo Durand, Nicolas Holzschuch, Cyril Soler, Eric Chan, and François X Sillion, *A frequency analysis of light transport*, ACM Transactions on Graphics (TOG) **24** (2005), no. 3, 1115–1126.
- [DKH⁺14] Carsten Dachsbacher, Jaroslav Krivánek, Miloš Hašan, Adam Arbree, Bruce Walter, and Jan Novák, *Scalable realistic rendering with many-light methods*, Computer Graphics Forum, vol. 33, Wiley Online Library, 2014, pp. 88–104.
- [FPG01] James A Ferwerda, Fabio Pellacini, and Donald P Greenberg, *Psychophysically based model of surface gloss perception*, Photonics West 2001-Electronic Imaging, International Society for Optics and Photonics, 2001, pp. 291–301.
- [Gre03] Robin Green, *Spherical harmonic lighting: The gritty details*, Archives of the Game Developers Conference, vol. 2, 2003, pp. 2–3.
- [HČA⁺12] Robert Herzog, Martin Čadík, Tunç O Aydıncı, Kwang In Kim, Karol Myszkowski, and Hans-P Seidel, *Norm: No-reference image quality metric for realistic image synthesis*, Computer Graphics Forum, vol. 31, Wiley Online Library, 2012, pp. 545–554.

- [HKWB09] Miloš Hašan, Jaroslav Krivánek, Bruce Walter, and Kavita Bala, *Virtual spherical lights for many-light rendering of glossy scenes*, ACM Transactions on Graphics (TOG), vol. 28, ACM, 2009, p. 143.
- [HPB07] Miloš Hašan, Fabio Pellacini, and Kavita Bala, *Matrix row-column sampling for the many-light problem*, ACM Transactions on Graphics (TOG), vol. 26, ACM, 2007, p. 26.
- [HVAPB08] Miloš Hašan, Edgar Velázquez-Armendáriz, Fabio Pellacini, and Kavita Bala, *Tensor clustering for rendering many-light animations*, Computer Graphics Forum, vol. 27, Wiley Online Library, 2008, pp. 1105–1114.
- [JB02] Henrik Wann Jensen and Juan Buhler, *A rapid hierarchical rendering technique for translucent materials*, ACM Transactions on Graphics (TOG), vol. 21, ACM, 2002, pp. 576–581.
- [Jen01] Henrik Wann Jensen, *Realistic image synthesis using photon mapping*, AK Peters, Ltd., 2001.
- [JHY⁺14] Wenzel Jakob, Miloš Hašan, Ling-Qi Yan, Jason Lawrence, Ravi Ramamoorthi, and Steve Marschner, *Discrete stochastic microfacet models*, ACM Transactions on Graphics (Proceedings of SIGGRAPH 2014) **33** (2014), no. 4.
- [JMLH01] Henrik Wann Jensen, Stephen R Marschner, Marc Levoy, and Pat Hanrahan, *A practical model for subsurface light transport*, Proceedings of the 28th annual conference on Computer graphics and interactive techniques, ACM, 2001, pp. 511–518.
- [Kaj86] James T Kajiya, *The rendering equation*, ACM Siggraph Computer Graphics, vol. 20, ACM, 1986, pp. 143–150.
- [Kel97] Alexander Keller, *Instant radiosity*, Proceedings of the 24th annual conference on Computer graphics and interactive techniques, ACM Press/Addison-Wesley Publishing Co., 1997, pp. 49–56.
- [KFB10] Jaroslav Krivánek, James A Ferwerda, and Kavita Bala, *Effects of global illumination approximations on material appearance*, ACM Transactions on Graphics (TOG) **29** (2010), no. 4, 112.
- [KTO11] Janne Kontkanen, Eric Tabellion, and Ryan S Overbeck, *Coherent out-of-core point-based global illumination*, Computer Graphics Forum, vol. 30, Wiley Online Library, 2011, pp. 1353–1360.
- [LALD12] Jaakko Lehtinen, Timo Aila, Samuli Laine, and Frédo Durand, *Reconstructing the indirect light field for global illumination*, ACM Transactions on Graphics (TOG) **31** (2012), no. 4, 51.
- [LH96] Marc Levoy and Pat Hanrahan, *Light field rendering*, Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, ACM, 1996, pp. 31–42.
- [LW93] Eric P Lafortune and Yves D Willems, *Bi-directional path tracing*, Proceedings of CompuGraphics, vol. 93, 1993, pp. 145–153.
- [MWA⁺13] Belen Masia, Gordon Wetzstein, Carlos Aliaga, Ramesh Raskar, and Diego Gutierrez, *Display adaptive 3d content remapping*, Computers & Graphics **37** (2013), no. 8, 983–996.
- [OP11] Jiawei Ou and Fabio Pellacini, *Lightslice: matrix slice sampling for the many-lights problem.*, ACM Trans. Graph. **30** (2011), no. 6, 179.
- [Owe13] Art B. Owen, *Monte carlo theory, methods and examples*, 2013.

- [PH10] Matt Pharr and Greg Humphreys, *Physically based rendering: From theory to implementation*, Morgan Kaufmann, 2010.
- [RAMN12] Ravi Ramamoorthi, John Anderson, Mark Meyer, and Derek Nowrouzezahrai, *A theory of monte carlo visibility sampling*, ACM Transactions on Graphics (TOG) **31** (2012), no. 5, 121.
- [REG⁺09] Tobias Ritschel, Thomas Engelhardt, Thorsten Grosch, H-P Seidel, Jan Kautz, and Carsten Dachsbacher, *Micro-rendering for scalable, parallel final gathering*, ACM Transactions on Graphics (TOG), vol. 28, ACM, 2009, p. 132.
- [RFWB07] Ganesh Ramanarayanan, James Ferwerda, Bruce Walter, and Kavita Bala, *Visual equivalence: towards a new standard for image fidelity*, ACM Transactions on Graphics (TOG), vol. 26, ACM, 2007, p. 76.
- [RH01] Ravi Ramamoorthi and Pat Hanrahan, *An efficient representation for irradiance environment maps*, Proceedings of the 28th annual conference on Computer graphics and interactive techniques, ACM, 2001, pp. 497–500.
- [SHHS03] Peter-Pike Sloan, Jesse Hall, John Hart, and John Snyder, *Clustered principal components for precomputed radiance transfer*, ACM Transactions on Graphics (TOG) **22** (2003), no. 3, 382–391.
- [SKS02] Peter-Pike Sloan, Jan Kautz, and John Snyder, *Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments*, ACM Transactions on Graphics (TOG), vol. 21, ACM, 2002, pp. 527–536.
- [TL04] Eric Tabellion and Arnauld Lamorlette, *An approximate global illumination system for computer generated films*, ACM Transactions on Graphics (TOG), vol. 23, ACM, 2004, pp. 469–476.
- [VG95a] Eric Veach and Leonidas Guibas, *Bidirectional estimators for light transport*, Photorealistic Rendering Techniques, Springer, 1995, pp. 145–167.
- [VG95b] Eric Veach and Leonidas J Guibas, *Optimally combining sampling techniques for monte carlo rendering*, Proceedings of the 22nd annual conference on Computer graphics and interactive techniques, ACM, 1995, pp. 419–428.
- [WABG06] Bruce Walter, Adam Arbree, Kavita Bala, and Donald P Greenberg, *Multidimensional lightcuts*, ACM Transactions on Graphics (TOG), vol. 25, ACM, 2006, pp. 1081–1088.
- [WFA⁺05] Bruce Walter, Sebastian Fernandez, Adam Arbree, Kavita Bala, Michael Donikian, and Donald P Greenberg, *Lightcuts: a scalable approach to illumination*, ACM Transactions on Graphics (TOG), vol. 24, ACM, 2005, pp. 1098–1107.
- [WHB⁺13] Beibei Wang, Jing Huang, Bert Buchholz, Xiangxu Meng, and Tamy Boubekeur, *Factorized point based global illumination*, Computer Graphics Forum, vol. 32, Wiley Online Library, 2013, pp. 117–123.
- [WKB12] Bruce Walter, Pramook Khungurn, and Kavita Bala, *Bidirectional lightcuts*, ACM Transactions on Graphics (TOG) **31** (2012), no. 4, 59.

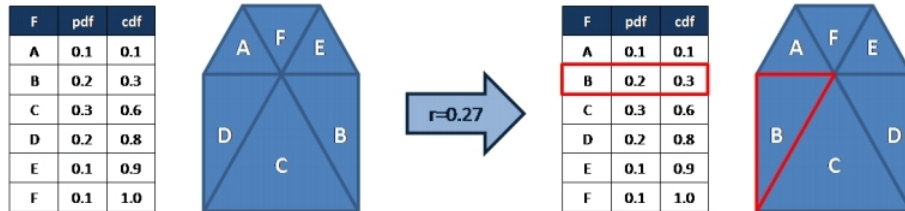
Apéndice A

Métodos auxiliares

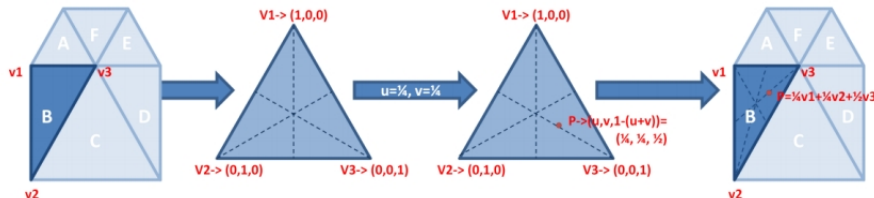
A.1. Muestreo uniforme de superficies

Para la generación de la nube de puntos a partir de la geometría de una escena, el primer paso consiste en generar posiciones uniformes dentro de los triángulos. Para ello, se realizan dos tipos de muestreos; uno por importancia que permite seleccionar triángulos dentro de la malla de la escena, y otro uniforme que genera puntos dentro de dicho triángulo.

Para llevar a cabo el muestreo por importancia, se utiliza el área de cada triángulo para pesar la probabilidad de elegir dicho triángulo. Para ello se almacenan en un vector todos los candidatos, junto con su probabilidad $pdf_i = \frac{area_i}{\sum area_i}$, así como su probabilidad acumulativa $cdf_i = \sum_{j \leq i} pdf_j$. Para determinar el triángulo a muestrear se genera una variable aleatoria $r \in [0, 1]$, de modo que se selecciona el elemento $i/r \in (cdf_{i-1}, cdf_i]$. Para encontrar el elemento que cumple con estas características se realiza una búsqueda dicotómica en el vector.



Para el muestreo de los triángulos, se utiliza el método explicado en [Pha04], por el cual se muestrea el triángulo de forma uniforme, de forma que se obtiene el punto según sus coordenadas baricéntricas. De esta forma, el proceso es independiente del área o de los ángulos del triángulo.



Apéndice B

Resultados adicionales

B.1. Tablas de rendimiento

Algoritmo	Ω_L	10k	100k	200k	300k	400k	500k
PBGI	0.1	71	50	56	51	53	53
PBGI	0.01	225	134	155	119	122	127
PBGI	0.001	365	1407	1851	1079	1153	1308
Lineal		662	4986	9480	13841	18227	22468

Tabla B.1: Coste en segundos del algoritmo descrito en el Capítulo 3 en comparación a calcular linealmente las luces virtuales. Para las pruebas realizadas se utiliza una resolución de 256x256x1 y $\Omega_S = 0,1$.

Algoritmo	Ω_M	α	256x256x1	256x256x2	512x512x1	512x512x2	1024x1024x1	1024x1024x2
CM SH	0.01	1.5	90	112	156	204	305	419
CM SH	0.01	0.1	106	141	198	268	389	524
CM SH	0.001	1.5	347	459	572	673	831	1007
CM SH	0.001	0.1	365	495	614	745	910	1095
CM SH	0.0001	1.5	437	829	1636	2508	3955	5086
CM SH	0.0001	0.1	451	852	1681	2565	4037	5140
CM V	0.01	1.5	139	207	382	603	1225	2030
CM V	0.01	0.1	154	232	415	662	1303	2175
CM V	0.001	1.5	453	617	854	1129	1979	2897
CM V	0.001	0.1	429	598	859	1162	1961	2910
CM V	0.0001	1.5	429	964	1828	2753	4749	6377
CM V	0.0001	0.1	498	905	1793	2732	4705	6336
Lineal			673	1133	2075	4634	7825	16184

Tabla B.2: Tabla de tiempos en segundos para el algoritmo de *clustering* de muestras. Las pruebas realizadas utilizan 10000 VPLs, $\Omega_C = 0,1$. CM atiende a *Clustering* de Muestras, SH a la implementación basada en armónicos esféricos y V a la versión que aproxima únicamente la visibilidad.

Algoritmo	Ω_L	Ω_M	256x256x1	512x512x1	1024x1024x1
PBGI	0.01		138	544	2189
BD	0.01	0.001	205	377	1154
BD	0.01	0.0001	235	460	1829

Tabla B.3: Tabla tiempos en segundos para método bidireccional con respecto de PBGI. En los test realizados se han utilizado 100 mil VPLs, $\alpha = 0,1$, $\Omega_S = 0,1$ y $\Omega_C = 0,1$

Algoritmo	Ω_L	Ω_M	50k	100k	150k
CM		0.001	2745	5112	8086
BD	0.01	0.001	536	377	445
BD	0.01	0.0001	640	460	522

Tabla B.4: Tabla comparativa del rendimiento del método bidireccional con el algoritmo de *clustering* de muestras. En los test realizados se ha fijado una resolución de 512x512x1, $\alpha = 0,1$, $\Omega_S = 0,1$ y $\Omega_C = 0,1$.

Escena	Resolución	N VPLs	PBGI	Bidireccional
<i>light field</i>	4096x4096x2	100000	12936	5056
<i>Sponza</i>	2048x2048x2	5000000	15134	7690
<i>Sibenik</i>	1024x1024x2	1000000	4905	3769
<i>Bunny</i>	2048x2048x2	5000000	3515	5858
<i>San Miguel</i>	1024x1024x2	8000000	10678	8299

Tabla B.5: Tabla de tiempos para las escenas adicionales presentadas en la Sección 6.3