

INFORMÁTICA GRÁFICA

Photon Mapping

Curso 2019-2020

Autores:

Víctor Miguel Peñasco Estívaléz	741294
Rubén Rodríguez Esteban	737215

Tabla de contenidos

Introducción	2
Generación del mapa de fotones	2
Muestreo de luz	2
Camino aleatorio del fotón	3
Almacenamiento de fotones	5
Estimación de radiancia	5
Iluminación directa	6
Iluminación global	8
Decisiones de diseño	14
Conclusiones	16
Organización del trabajo	17
Referencias	18

Introducción

Este informe presenta un motor de renderizado basado en el algoritmo de mapeado de fotones o *Photon mapping*, introducido por Henrik Wann Jensen y que aproxima la Ecuación de Render. La idea básica principal consiste en disparar fotones desde la fuente de luz en lugar de desde la cámara y almacenarlos en un mapa de fotones, de ahí su nombre. El segundo paso traza los rayos de la cámara y estima la iluminación global con una estimación de la radiación basada en los vecinos más cercanos del mapa. El algoritmo de Photon Mapping es especialmente interesante porque puede simular difíciles efectos para capturar con Ray Tracing y Path Tracing, como las cáusticas, o efectos que suelen ser computacionalmente muy costosos como los medios participativos.

Para la implementación del algoritmo de Photon Mapping se ha usado el código facilitado por el profesorado de la asignatura. Para poder incluir las imágenes HDR generadas por el programa en este documento, se ha hecho uso de software de terceros como Photoshop y GIMP para hacer tone mapping y transformar las imágenes a LDR.

Finalmente se ha optado por proporcionar tanto un apartado de las conclusiones a las que se ha llegado durante la elaboración del proyecto como una sección donde figura el trabajo realizado por cada integrante del equipo.

Generación del mapa de fotones

El proceso dedicado a la generación del mapa de fotones se ha subdividido en dos etapas diferentes que se van a explicar de forma detallada a continuación:

Muestreo de luz

El primer paso para almacenar los fotones en el mapa de fotones es muestrear un rayo que inicie el camino aleatorio del fotón (Photon Random Walk). Para esto se ha utilizado la técnica de **Rejection Sampling**. El objetivo es muestrear de manera uniforme en una esfera unitaria cuyo centro es la localización del punto de luz.

De forma iterativa, se lanza un fotón desde cada una de las luces puntuales (no se ha implementado iluminación de área). La base de este método ha consistido en incluir la esfera de proyección en un cubo también unitario de tal manera que el muestreo se ha efectuado por medio de tres valores aleatorios, uno para cada eje, cuyos rangos oscilan en el intervalo $[-1, 1]$, lo cual genera un punto dentro del cubo. Las muestras no válidas son aquellas cuya distancia euclídea al centro de la esfera es superior a 1. De esta forma, si la muestra se encuentra dentro de la esfera inscrita en el cubo, es válida, si no lo es, se rechaza y se pasa a intentar con la siguiente luz. Como en cada intento las probabilidades de generar una muestra que se rechace son las mismas, la diferencia de número de fotones iniciados desde dos puntos de luz cualesquiera de la escena converge a cero.

Esto funciona bien cuando la potencia de todas las luces de la escena es la misma. En otro caso, por razones de eficiencia, se podrían emitir fotones con la misma energía inicial

(independientemente de la potencia de la luz desde la que se lanzan), y que el número de fotones emitidos desde cada luz fuera proporcional a la potencia de la luz. Además, tener almacenados en el mapa fotones con energías muy dispares dificulta la convergencia del algoritmo.

Si el punto generado por rejection sampling es aceptado, se genera una dirección restando al punto la localización de la luz puntual. Esta dirección, junto con la localización de la luz, forma el rayo que se utiliza como inicio en el camino del photon, el cual se describe a continuación.

Camino aleatorio del fotón

Una vez se tiene el rayo inicial, se utiliza la función ***trace_ray*** implementada por los profesores para realizar el camino del fotón y guardar en listas los fotones que posteriormente se almacenarán en el mapa de fotones. A esta función en cuestión hay que pasarle como parámetros:

- el rayo generado por el método de rejection sampling.
- la potencia de la fuente de luz: energía inicial del fotón.
- la lista de fotones globales.
- la lista de fotones cáusticos.
- booleano *direct*: almacenar también el primer rebote.
- booleano *direct_only*: almacenar únicamente el primer rebote.

Un hecho importante a enfatizar es el empleo de un flag denominado *flagDI* con el fin de determinar cómo calcular la estimación de radiancia. Dicho flag puede tomar valores cuyos significados son los siguientes:

- (0) Iluminación directa con Ray Tracing e iluminación global con Photon Mapping (*direct* = false, *direct_only* = false).
- (1) Iluminación directa y global con Photon Mapping (*direct* = true, *direct_only* = false).
- (2) Sólo iluminación directa con Ray Tracing (*direct* = false, *direct_only* = false).
- (3) Sólo iluminación directa con Photon Mapping (*direct* = false, *direct_only* = true).
- (4) Sólo iluminación global con Photon Mapping (*direct* = false, *direct_only* = false).

La **energía inicial del fotón** se corresponde con la intensidad de la luz (una para cada canal RGB). Para que la energía esté **normalizada** es necesario dividir la intensidad entre el número de fotones emitidos desde el punto de luz concreto. Además es necesario tener en cuenta la integración sobre el ángulo sólido de la esfera (geometría sobre la que se ha muestreado), lo que supone multiplicar por 4π .

En primer lugar, se ha controlado el número caminos de fotón que se van a lanzar con el objetivo de garantizar que como máximo se van a lanzar esos caminos y no más. Posteriormente, con la función *shift()* se desplaza el punto de origen del rayo ligeramente para evitar errores numéricos en caso de intersectar con la superficie de un objeto que esté en sentido opuesto al del rayo, ya que el punto de luz podría estar pegado o muy cerca de un objeto.

Seguidamente se ha procedido a calcular los caminos aleatorios de los fotones tal y como se va a explicar a continuación. El cálculo de una intersección y su posible rebote posterior se efectúa de forma iterativa, comprobando inicialmente si ha sobrepasado la cota máxima de fotones a emitir. En caso negativo, dado el rayo generado por rejection sampling, se calcula su siguiente intersección con algún objeto de la escena (el más cercano de ellos), fenómeno para el cual se han contemplado los siguientes casos:

- Si el objeto con el que intersecta el fotón es un objeto que genera cáusticas, no se almacena ningún fotón, pero se actualiza una variable booleana llamada *is_caustic_particle*. La importancia de esta variable estriba en que para las iteraciones posteriores se usa para determinar si guardar el siguiente fotón en la lista cáustica o en la global.
- Si el objeto intersectado no es cáustico, se comprueba primero si el fotón viene directamente de la fuente de luz o ha rebotado únicamente con algún objeto delta antes, porque en ese caso se trata de iluminación directa que puede ser simulada con ray tracing. Si se decide utilizar photon mapping en vez de ray tracing para calcular la iluminación directa, (variable booleana *direct* o *direct_only* es *true*) entonces el fotón de esa primera intersección no delta sí se guarda. Cuando el fotón tiene que ser guardado se usa la variable *is_caustic_particle* para determinar en qué lista, cáustica o global, almacenarlo. Adicionalmente, antes de almacenar, se comprueba si el número de fotones guardados en la listas no sobrepasan los umbrales de capacidad máxima, *m_nb_caustic_photons* y *m_nb_global_photons*, respectivamente.

Tras el almacenamiento del fotón, se hace Ruleta Rusa para poder evaluar si el camino del fotón termina o no en función del albedo del objeto intersectado (se utiliza la media de los tres canales RGB). En este punto, si la condición *epsilon2 > avg_surf_albedo* se evalúa como verdad, la Ruleta Rusa marca evento de absorción del fotón. De este modo, si el fotón es absorbido o si ha rebotado ya un total de 20 veces, el camino termina. En caso contrario, se calcula el nuevo rayo desde el punto de intersección actual haciendo uso de la función *get_outgoing_sample_ray*, actualizando la energía saliente del fotón en base a las propiedades del material con el que ha intersectado. Esta actualización consiste en multiplicar la energía por el albedo. Luego, en el caso de que el material intersectado no sea delta, la energía se multiplica por el coseno entre la normal del objeto y la dirección del rayo, y se divide por π . Finalmente, independientemente de las propiedades del material, la energía se divide entre la distribución de probabilidad y la media del albedo para los tres canales RGB.

Si las listas de fotones están llenas termina el camino dado que no hay posibilidad de seguir almacenando fotones nuevos. En caso contrario comienza una iteración con el nuevo rayo calculado y su energía actualizada.

Almacenamiento de fotones

El almacenamiento de los fotones en los mapas de fotones es un proceso que se lleva a cabo tras efectuar el muestreo de luz (rejection sampling) y calcular el camino aleatorio de los fotones (Photon Random Walk) lanzados. Es importante enfatizar el hecho de que se han usado dos mapas distintos, uno para fotones cáusticos y otro para fotones globales, usando como estructura de datos el Kd tree facilitado por el profesorado de la asignatura dado que es una estructura de datos ya implementada que permite tanto guardar mucha información como realizar búsquedas eficientes de los vecinos más cercanos.

La razón por la que se ha decidido usar dos Kd trees diferentes estriba en que así se puede decidir cuántos fotones de cada tipo, globales y cáusticos, almacenar. Para llenar el mapa global simplemente se ha recorrido la lista de fotones globales y guardarlos uno a uno. Sin embargo, para el relleno de fotones cáusticos ha sido necesario hacer una comprobación adicional: verificar la existencia de objetos cáusticos en la escena. El relleno del mapa de fotones cáustico se ha efectuado de la misma forma que el relleno del mapa global.

Como se dispone de un mapa de fotones exclusivo para las cáusticas, se puede aumentar el número de fotones a almacenar en este, de forma que posteriormente en la estimación de radiancia se puede conseguir representar las cáusticas de forma más precisa.

Tras llevar a cabo el relleno de los mapas de fotones se ha procedido a balancearlos con el fin de poder realizar más tarde las búsquedas. De esta manera, el tiempo necesario para encontrar los k vecinos más cercanos a un punto dado en un Kd tree con N fotones almacenados, es del orden de $O(k \log N)$.

Estimación de radiancia

Para llevar a cabo la segunda parte del proceso de Photon Mapping, la estimación de radiancia o densidad, se establece una diferenciación dependiendo de la modalidad de iluminación escogida por medio de un flag denominado *flagID*. Estas modalidades son las siguientes:

- (0) Iluminación directa con Ray Tracing e iluminación global con Photon Mapping.
- (1) Iluminación directa y global con Photon Mapping.
- (2) Sólo iluminación directa con Ray Tracing.
- (3) Sólo iluminación directa con Photon Mapping.
- (4) Sólo iluminación global con Photon Mapping

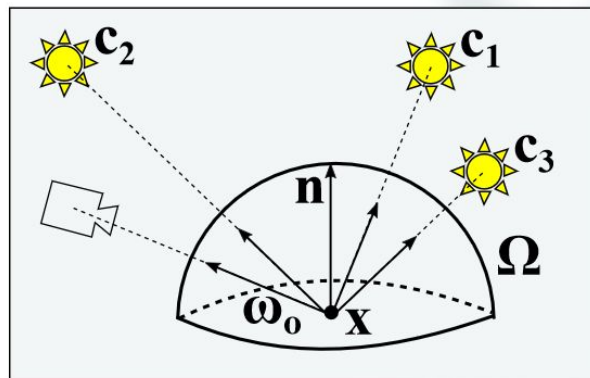
Independientemente de la modalidad elegida, todo comienza trazando un rayo desde la cámara hasta cada píxel. Hay que tener en cuenta que esto puede generar problemas de aliasing. Para mejorar esto se podría aumentar el número de muestras por píxel (muestreando en un punto aleatorio dentro de éste), pero no se ha implementado nada al respecto.

El rayo lanzado puede intersectar con un objeto delta. En ese caso, se genera un nuevo rebote hasta que se intersecte con un objeto difuso. Una vez se ha intersectado con un objeto difuso, se procede a calcular la iluminación directa y global tal y como se describe en los siguientes apartados.

Iluminación directa

Se han utilizado dos métodos para calcular la iluminación directa: Ray Tracing y Photon Mapping.

Si se utiliza Ray Tracing, por cada rayo trazado desde la cámara que ha intersectado sobre un objeto, se calcula la contribución de cada una de las fuentes de luz de la escena (sólo se han tenido en consideración luces puntuales). Para esto, si el punto intersectado es visible desde una fuente de luz, se suma la contribución de ésta de acuerdo con la ecuación de presente en la Figura 1.



$$L_o(\mathbf{x}, \omega_o) = \sum_{i=1}^n \frac{p_i}{|\mathbf{c}_i - \mathbf{x}|^2} f_r \left(\mathbf{x}, \frac{\mathbf{c}_i - \mathbf{x}}{|\mathbf{c}_i - \mathbf{x}|}, \omega_o \right) \left| \mathbf{n} \cdot \frac{\mathbf{c}_i - \mathbf{x}}{|\mathbf{c}_i - \mathbf{x}|} \right|$$

Figura 1

Contribución de fuentes de luz sobre punto intersectado (ecuación de render)

La contribución de cada luz sobre el punto (en el caso ser visible) consiste en la potencia de la luz dividida por la distancia al cuadrado, multiplicado por la geometría (producto escalar entre la normal y la dirección normalizada del punto a la luz), y por la BRDF del material.

En cuanto a la función BRDF utilizada, se ha usado la parte lambertiana de Phong, lo cual se resume en el albedo (k_d) dividido por π . Se ha optado por no tener en cuenta la parte especular de Phong debido a que no se encontraba implementada en el código proporcionado y la integración de nuevo código con las bibliotecas compiladas podría resultar complicado. La luminancia obtenida mediante el proceso de Ray Tracing simplemente se suma posteriormente a la luminancia correspondiente a la iluminación global. Esto es posible gracias a la propiedad aditiva de la luz.

Si se decide utilizar Photon Mapping para computar la iluminación directa, es necesario haber guardado en el mapa de fotones el fotón generado por el primer rebote difuso durante la parte de preproceso de Photon Mapping, es decir, la generación de dicho mapa. Si estos fotones han sido guardados, para obtener la iluminación directa simplemente hay que realizar el proceso relativo a la iluminación global que se explica a continuación.

Cuestión 2.1. Renderizar dos imágenes de la escena 2 con solo iluminación directa, cada imagen donde la iluminación directa es calculada con Ray Tracing y Photon Mapping, respectivamente. ¿Cuáles son las diferencias entre la iluminación directa calculada con Ray Tracing y con Photon Mapping? ¿Cuál de las dos crees que es más exacta y por qué?

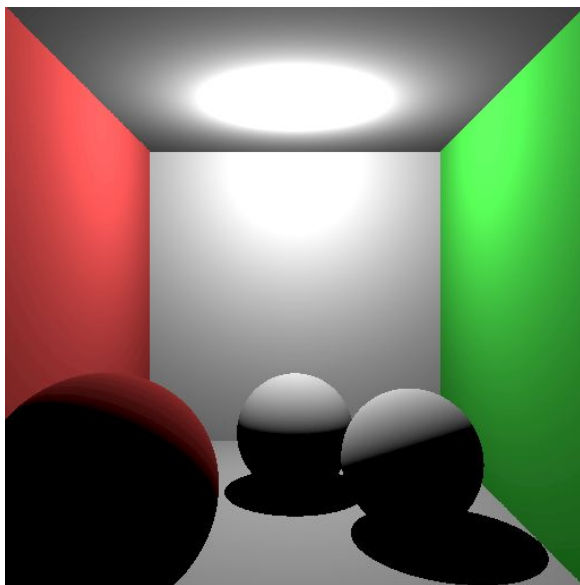


Figura 2.1

Escena 2 con solo iluminación directa
calculada con Ray Tracing

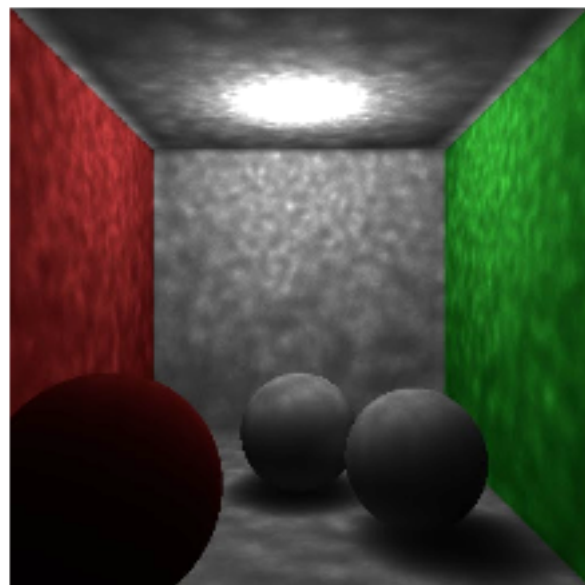


Figura 2.2

Escena 2 con solo iluminación directa
calculada con Photon Mapping

La imagen generada mediante la técnica de Ray Tracing es mucho más exacta que la generada por Photon Mapping. En teoría, la imagen generada mediante Photon Mapping, con un número de fotones que tiende a infinito, y un kernel cuyo tamaño tiende a 0, genera imágenes igual de exactas. En la práctica, la técnica de Ray Tracing ofrece buenos resultados de iluminación directa con un tiempo de cálculo despreciable en comparación con el que necesitaría Photon Mapping.

Otro defecto de Photon Mapping es el efecto que se puede observar en las esquinas, que aparecen más iluminadas que las zonas contiguas. Esto se debe a que se asume que las superficies son planares (se ha aplicado un filtro de cono sobre superficie teóricamente plana). Por esta razón, en la estimación de radiancia de las esquinas contribuyen fotones cercanos que se encuentran en ambos planos. En Photon Mapping, pese a los posibles kernels, se asume que los fotones se encuentran en el mismo punto sobre el que se ha intersectado.

Sin embargo, la ventaja de Photon Mapping es clara: renderizar efectos que sería imposibles o muy costosos de simular con Ray Tracing o Path Tracing. En las dos imágenes que se muestran a continuación, se puede observar la diferencia al renderizar la escena 1 con sólo iluminación directa utilizando Ray Tracing frente a utilizar Photon Mapping. Pese a la exactitud de Ray Tracing, no es capaz de simular las cáusticas que sí se generan utilizando sólo iluminación directa con Photon Mapping.

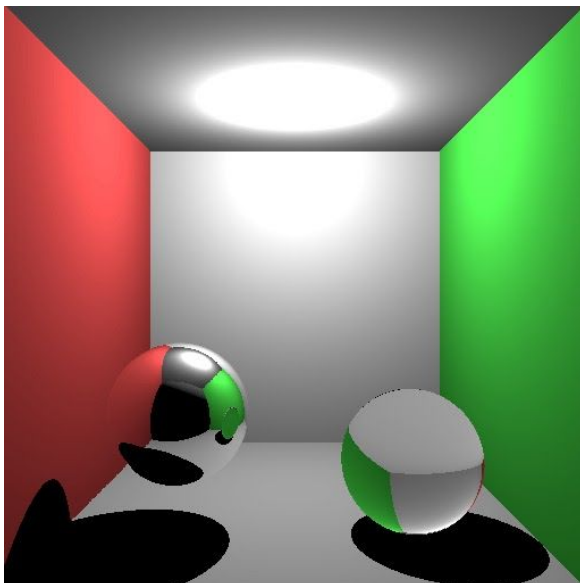


Figura 3.1

Escena 1 con solo iluminación directa
calculada con Ray Tracing

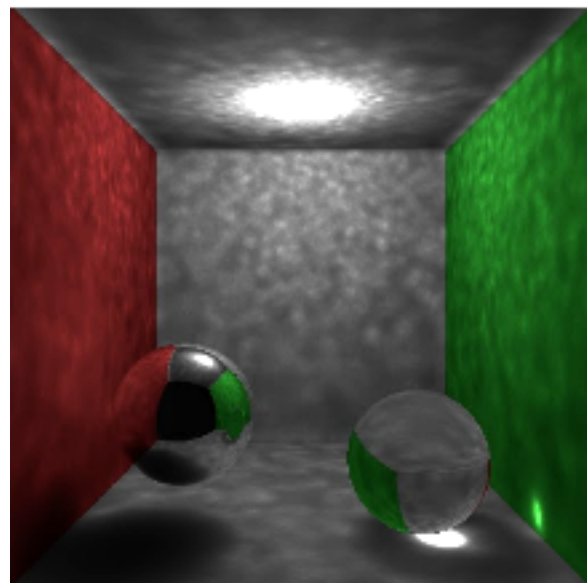


Figura 3.2

Escena 1 con solo iluminación directa
calculada con Photon Mapping

Las dos escenas anteriores han sido renderizadas con un total de 100.000 fotones en cada mapa y 30 vecinos a buscar.

Iluminación global

Para el cómputo de la iluminación global se hace uso del mapa de fotones almacenado en una estructura de datos Kd tree. Sobre el punto intersectado, el mismo que se ha utilizado para la iluminación directa mediante Ray Tracing, se hace una búsqueda en el Kd tree para encontrar los k fotones más cercanos a ese punto. Esto se hace tanto sobre el mapa de fotones globales (los procedentes de un rebote previo sobre material difuso) como sobre el mapa de fotones cáusticos (los procedentes de un rebote previo sobre material delta).

Esta búsqueda sobre el mapa devuelve una lista de fotones de longitud k . Posteriormente se itera sobre ella para tener en cuenta la contribución de cada uno de los fotones encontrados.

La contribución de cada uno de los fotones se ha estimado mediante una ecuación de radiancia (véase Figura 4). Para esto se ha utilizado un filtro de cono, de forma que los fotones encontrados que estén más cerca del punto intersectado serán tenidos más en cuenta que los fotones más lejanos.

$$L_r(\mathbf{x}, \omega_r) \approx \frac{1}{(1 - \frac{2}{3k})\pi r^2} \sum_{p=1}^n \left(1 - \frac{\|\mathbf{x} - \mathbf{x}_p\|}{kr} \right) f_r(\omega_p, \omega_r) \Delta\Phi_p(\mathbf{x}_p, \omega_p)$$

Figura 4

Estimación de radiancia mediante filtro de cono (aproximación de ecuación de render)

Para cada fotón encontrado, se hace el producto de su flujo (energía del fotón) por la BRDF (componente difusa dividida por π) y por un peso, que se corresponde con 1 menos la distancia euclídea entre el punto y el fotón, dividida entre k por el radio del kernel (distancia desde el punto hasta el vecino más lejano encontrado). En la práctica el valor de filtrado k que se ha utilizado ha sido 2.

Estos valores de cada fotón se acumulan, para ser finalmente multiplicados por un factor de filtrado de cono, que depende de los parámetros k y r , siendo este último el radio del kernel.

Cuestión 2.2. Renderizar la escena 1 usando Photon Mapping. Esta escena tiene dos esferas con material delta BSDF. Describir cómo se computan las sombras en cada una de las esferas y por qué.

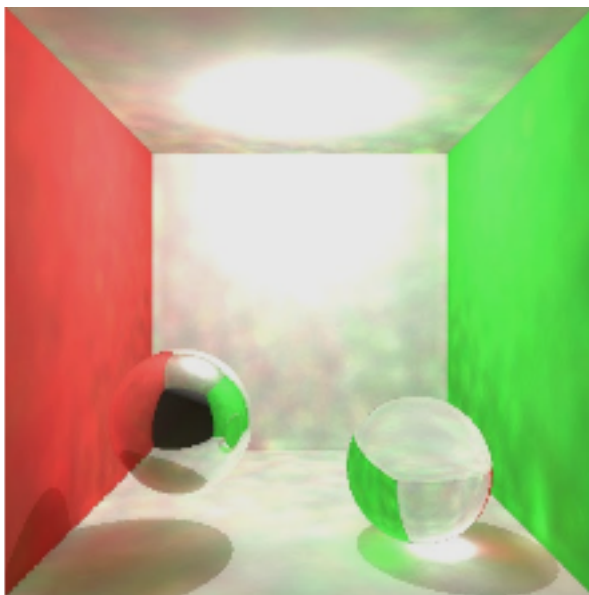


Figura 5.1

Escena 1 con iluminación directa calculada con Ray Tracing

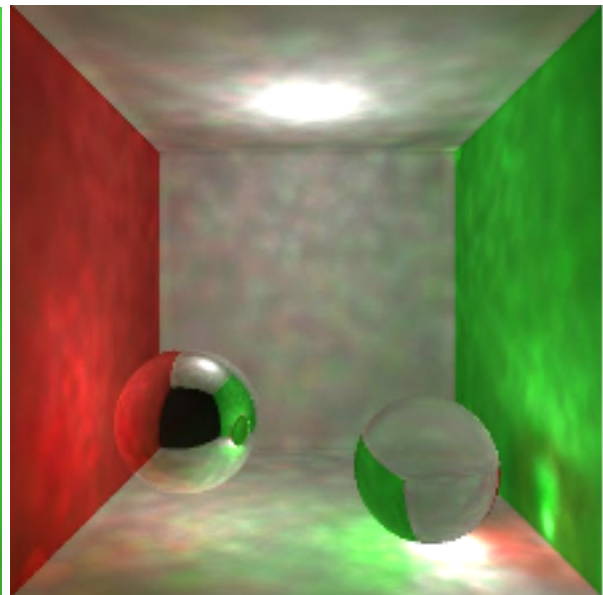


Figura 5.2

Escena 1 con iluminación directa calculada con Photon Mapping

Las dos escenas anteriores han sido renderizadas con un total de 100.000 fotones en cada mapa y 100 vecinos a buscar.

Las sombras se computan en ambas escenas con Photon Mapping. Para sombrear cada una de estas escenas, se han seguido los pasos explicados previamente para obtener la iluminación global. Al trazar un rayo desde la cámara, si se intersecta con un material delta, se sigue su camino (nuevo rebote) hasta llegar a un material difuso. Es así como se consiguen los efectos perfecto especular y refracción que muestran cada una de las esferas. Al intersectar con un material difuso, ya sea al trazar un rayo desde la cámara o después de haber pasado por objetos delta, se realiza la estimación de radiancia, buscando en ambos mapas. Al lanzar los caminos de los fotones desde las fuentes de luz (en la parte de preprocesado), se consiguen generar los efectos de cáusticas que serían prácticamente imposibles de conseguir si se tuvieran que encontrar los caminos desde la cámara hasta las luces (como en Path Tracing).

Cuestión 2.3 Renderizar la escena 1 incrementando el número de fotones a almacenar en los mapas ($n = \{1K, 10K, 50K, 100K\}$) fijando el número de fotones vecinos para estimar la radiancia con $k = 10$. Ahora, renderizar la misma imagen con $n = 100K$ fotones e ir incrementando el número de fotones vecinos para hacer la estimación de radiancia con valor $k = \{1, 10, 50, 100\}$. Discutir los resultados en términos coste y calidad de imagen.

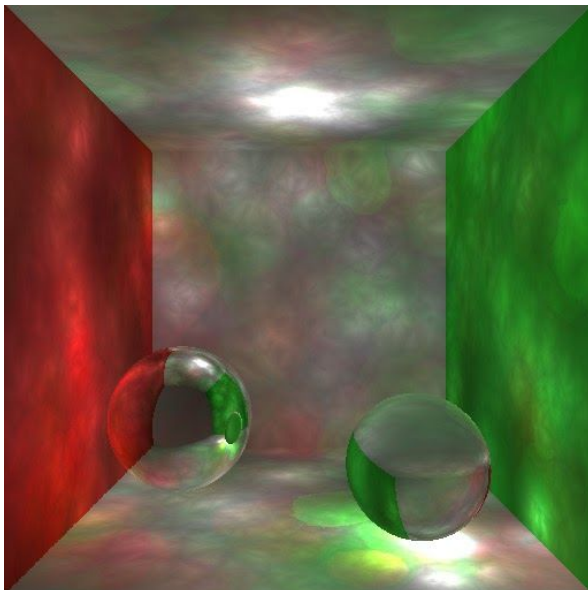


Figura 6.1
Escena 1 con iluminación directa y
global con Photon Mapping
 $n = 1K$ y $k = 10$

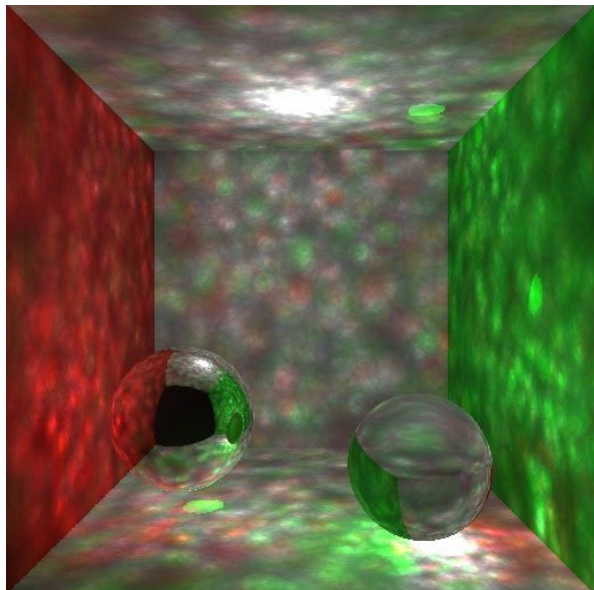


Figura 6.2
Escena 1 con iluminación directa y
global con Photon Mapping
 $n = 10K$ y $k = 10$

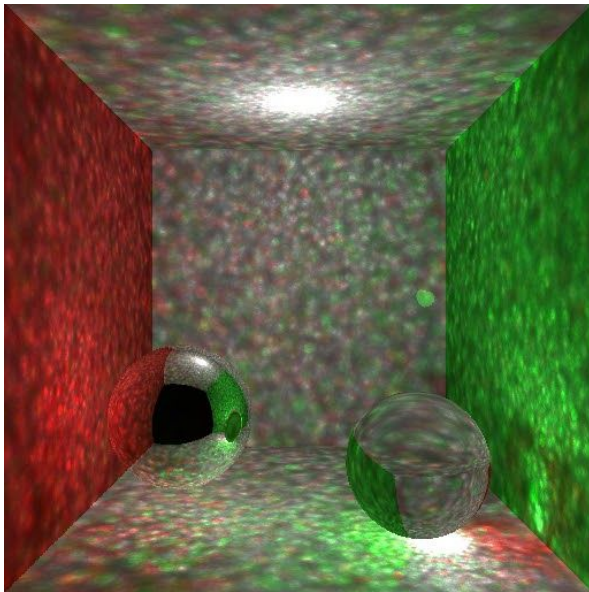


Figura 6.3
Escena 1 con iluminación directa y
global con Photon Mapping
 $n = 50K$ y $k = 10$

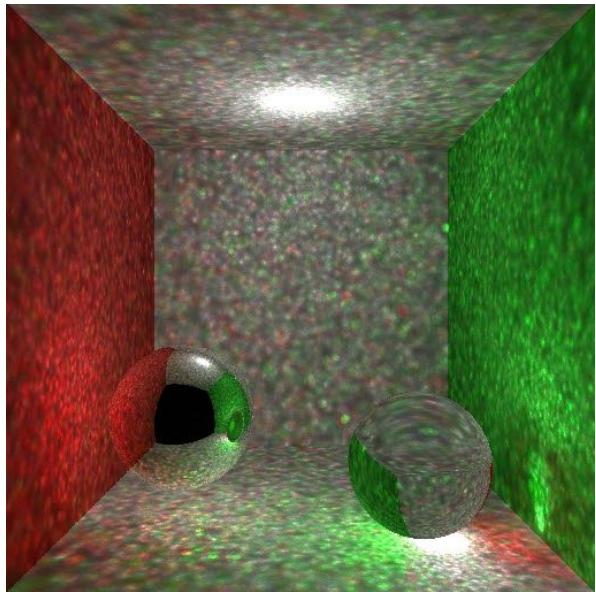


Figura 6.4
Escena 1 con iluminación directa y
global con Photon Mapping
 $n = 100K$ y $k = 10$

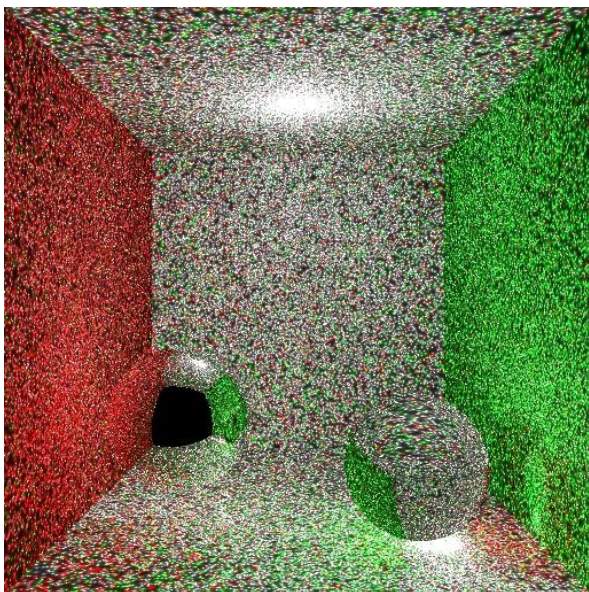


Figura 7.1
Escena 1 con iluminación directa y
global con Photon Mapping
 $n = 100K$ y $k = 1$

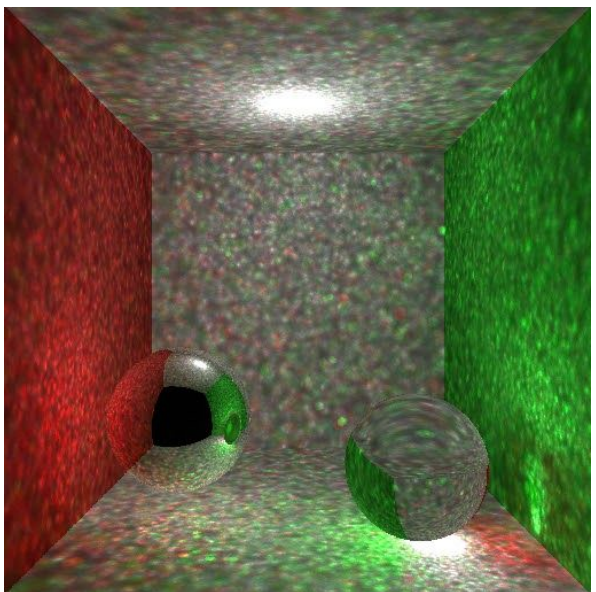


Figura 7.2
Escena 1 con iluminación directa y
global con Photon Mapping
 $n = 100K$ y $k = 10$

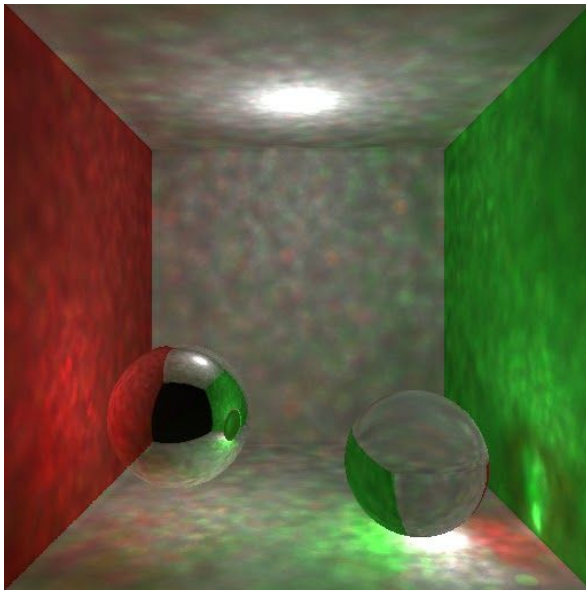


Figura 7.3

Escena 1 con iluminación directa y
global con Photon Mapping
 $n = 100K$ y $k = 50$

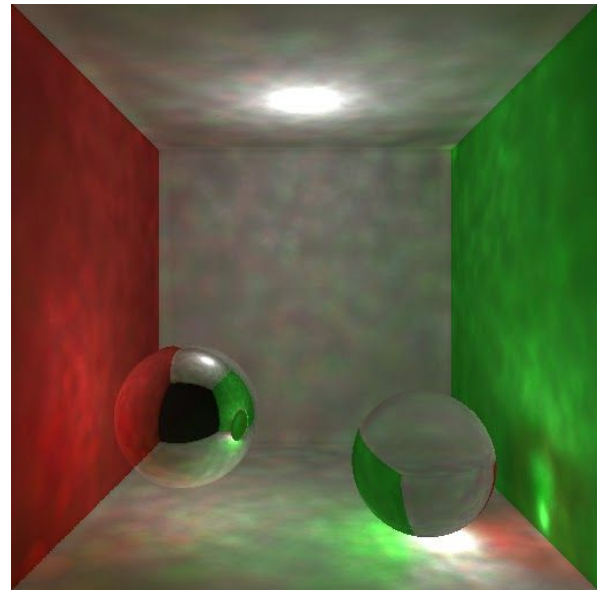


Figura 7.4

Escena 1 con iluminación directa y
global con Photon Mapping
 $n = 100K$ y $k = 100$

De forma previa a la valoración de la calidad y coste de cada una de las imágenes, es importante recalcar como se aprecia una correcta normalización de la energía de los fotones, de forma que en cada una de las imágenes, pese a cambiar el valor de uno de los parámetros habiendo fijado el otro, la energía total es aproximadamente la misma.

En base a las imágenes anteriores se puede observar que lo mejor es mantener una proporcionalidad entre ambos parámetros ya que, por ejemplo, la escena 1 generada con 1K fotones almacenados y 10 vecinos es una imagen que aunque pierde mucha precisión no es tan incómoda a nivel visual como otras imágenes como la escena 1 generada con 50K y 10 vecinos o la escena generada con 10K y 10 vecinos en las que se puede ver una peor visualización precisamente por esta descompensación entre los parámetros.

Otro factor importante es el número de fotones vecinos k que se emplean para realizar el cálculo de la estimación de radiancia. De acuerdo a los resultados obtenidos cuánto más pequeño es ese parámetro más inexacto es el cálculo de la estimación dado que se está perdiendo mucha cantidad de información. Además, en relación a lo comentado en el párrafo anterior se puede observar que cuanto más pequeño es este valor y más grande es el número de fotones a lanzar (mayor desproporcionalidad) mucho mayor es la cantidad de información pérdida. Por ello, la imagen generada con 100K fotones a almacenar y un 1 vecino, así como la generada con 100K fotones y 10 vecinos están marcadas por la presencia de un alto nivel de ruido frente a las imágenes generadas con 100K fotones y 50 vecinos y 100K fotones con 100 vecinos donde la calidad es mucho mayor. Por tanto, dado un número fijo de fotones almacenados, un número mayor de vecinos en la estimación de radiancia reduce el ruido a cambio de un mayor sesgo (bias), ya que aumenta el número de fotones que contribuyen a iluminar más de un punto de la escena al mismo tiempo.

En base a los siguientes experimentos también se ha podido comprobar que entre las dos partes que conforman el algoritmo de Photon Mapping, trazado de fotones y estimación de radiancia, la fase más costosa con diferencia es la segunda, dado que para efectuar el cálculo de la radiancia es necesario buscar en los mapas de fotones los k vecinos más próximos. Dicha operación tiene un coste del orden de $O(k \log N)$, siendo k el número de vecinos y N el número de fotones en el mapa. Por tanto, aunque incrementar el número de fotones almacenados contribuye a aumentar el tiempo de cálculo, esto se hace (en la parte de estimación de radiancia) de forma logarítmica, mientras que aumentar el número k de fotones más cercanos tiempo un gran impacto directo en el coste.

En definitiva, la imagen con más calidad es la generada con 100K fotones almacenados y 100 vecinos para la estimación de radiancia. Como contrapartida, existe un sesgo que hace que la imagen no sea realmente correcta de acuerdo a la física, y el tiempo de cálculo es mayor que el de los otros renders de la misma escena. Por otra parte, como se ha comentado en varias ocasiones, esta técnica permite obtener efectos que no sería posible obtener con otros métodos.

Cuestión 2.4 Renderizar la escena 4 computando la iluminación directa usando Ray Tracing y usando Photon Mapping. Describe cuáles son las diferencias entre computar la iluminación directa de la imagen con ambas técnicas.

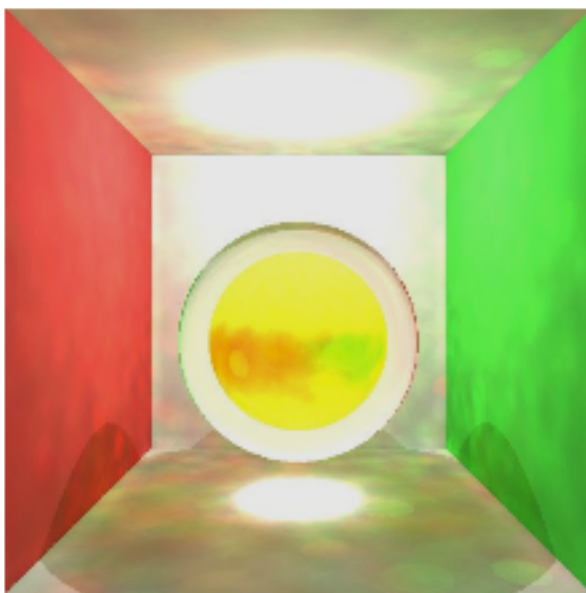


Figura 8.1
Escena 4 con iluminación directa
usando Ray Tracing y Photon Mapping
 $n = 100K$ y $k = 100$

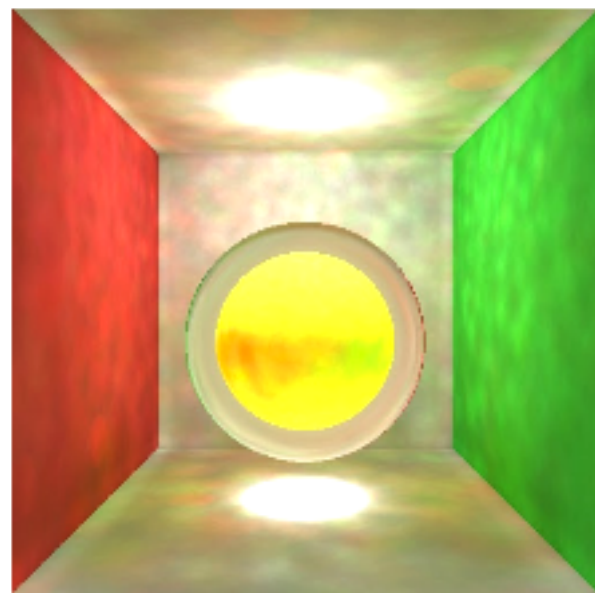


Figura 8.2
Escena 4 con sólo iluminación
directa con Photon Mapping
 $n = 100K$ y $k = 100$



Figura 9.1
Escena 4 con sólo iluminación
directa con Ray Tracing

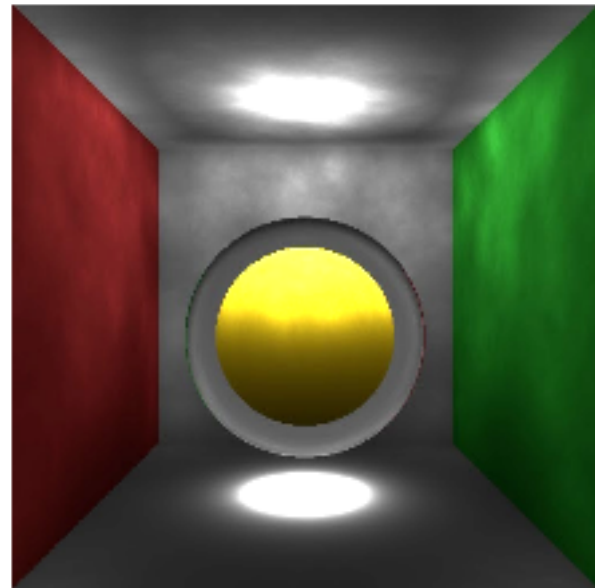


Figura 9.2
Escena 4 con sólo iluminación
directa con Photon Mapping
 $n = 10K$ y $k = 50$

Se han realizado renders de la escena 4 tal y como se ha indicado. Dos de las escenas incluyen iluminación directa y global y las otras dos sólo directa. La respuesta a la pregunta se puede responder de forma mucho más clara atendiendo a los renders en los que solo se simula la iluminación directa.

En la escena con sólo Ray Tracing, no se consigue iluminar la esfera difusa ya que se trata de iluminación global (indirecta), debido a que antes tiene que atravesar material refractante, es decir, toda la iluminación de esta esfera difusa, se consigue estimar con los fotones almacenados en el mapa de fotones cáusticos. Al utilizar sólo Ray Tracing, cuando los rayos trazados desde la cámara llegan a la esfera difusa tras atravesar la refractante, los rayos de sombra utilizados para calcular la iluminación directa chocan con la esfera exterior, de manera que no llega ninguna luz directa.

Decisiones de diseño

A lo largo del documento, se ha comentado algunas decisiones de diseño relativas a la implementación del Photon Mapping, como puede ser el *flagDI* que permite cambiar fácilmente la modalidad de la iluminación a renderizar, o la igualación del número de fotones lanzados desde cada luz (asumiendo idealmente que tienen la misma potencia). A continuación se describen más decisiones de diseño que no han sido mencionadas anteriormente.

A la hora de generar los mapas de fotones, se han utilizado 3 parámetros:

- Número máximo de fotones a disparar.
- Número máximo de fotones a guardar en el mapa global.

- Número máximo de fotones a guardar en el mapa cáustico.

En base a los valores de estos parámetros y de acuerdo a la composición de la escena, la fase de preprocesado (generación de los mapas) tendrá una condición de terminación u otra, tal y como se ha visto en la descripción de la función *trace_ray*. Este hecho significa que el número de fotones emitidos puede ser menor que el número máximo de fotones a disparar. Por esta razón, la normalización de la energía del fotón, en lo que se refiere a la división entre el número de fotones emitidos, se hace a posteriori. Durante la estimación de radiancia, se divide el flujo del fotón entre el número de fotones emitidos, ya que en ese momento se conoce exactamente su valor.

Otra decisión de diseño tomada ha sido, a la hora de trazar los fotones en la generación del mapa de fotones, multiplicar la energía inicial del fotón por el número de luces de la escena. Si, por ejemplo, se van a emitir 100 fotones en total en una escena con una única luz puntual, y el flujo inicial de cada fotón emitido, después de normalizar, tiene valor 1. Si a la misma escena se le añaden otras dos luces iguales (3 luces en la escena), emitiendo 100 fotones en total, el flujo inicial de cada fotón tendrá, después de la normalizar, valor 3. Gracias a esta decisión, se consigue obtener resultados similares (triple de luz en la escena) sin necesidad de triplicar el número de fotones emitidos, lo cual aumentaría el tiempo de cálculo.

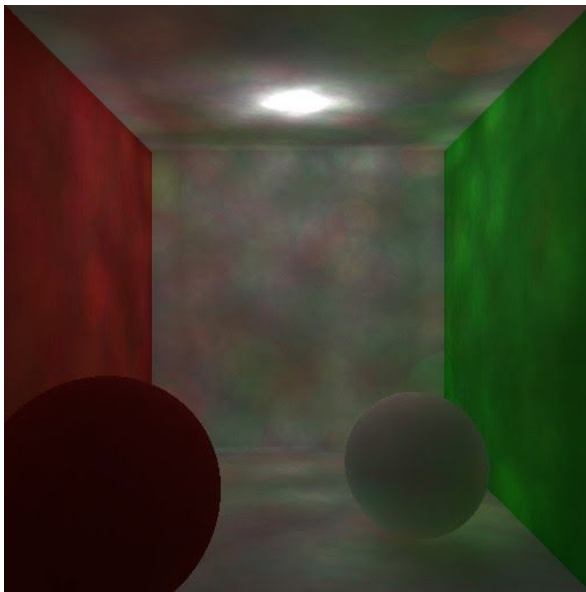


Figura 10.1

Escena con una sola luz puntual
con iluminación directa y global
con Photon Mapping
 $n = 10K$ y $k = 50$

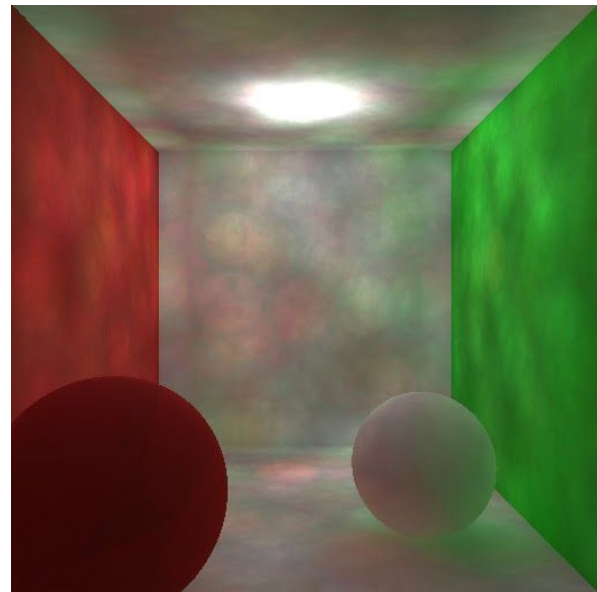


Figura 10.2

Escena con tres luces puntuales
idénticas con iluminación directa
y global con Photon Mapping
 $n = 10K$ y $k = 50$

Conclusiones

A lo largo de este informe se ha podido tomar conciencia de una serie de hechos importantes expuestos a continuación. El algoritmo de Photon Mapping se ha implementado con el objetivo de poder renderizar escenas con presencia de iluminación global. Mediante este algoritmo se pueden conseguir efectos de luz que son muy difíciles de conseguir con otras técnicas como por ejemplo el Path Tracing. De este modo, Photon Mapping es una técnica que permite generar de manera muy aproximada a la realidad, a pesar de la presencia del sesgo (bias), efectos luminosos de gran complejidad como son las cáusticas.

Este algoritmo tiene la ventaja de que a nivel promedio es más rápido que el algoritmo del Path Tracing. De hecho, con una cantidad de fotones a disparar razonable y un número de fotones vecinos a buscar no muy exagerado se pueden obtener escenas con un alto grado de realismo, aunque siempre están marcadas por la huella de ciertas inconsistencias como la presencia del sesgo. El algoritmo del Path Tracing por su parte requiere un alto número de rayos a lanzar para poder renderizar escenas con buena calidad y sin ruido.

Este proyecto ha sido realizado sin ningún opcional dado el poco margen de tiempo del que se ha dispuesto para poder implementarlo, aunque inicialmente se había considerado incorporar ciertas extensiones como los medios participativos o kernels adicionales para poder probar el comportamiento del algoritmo con cada uno de ellos.

En definitiva, se considera que después de la implementación, se han comprendido mejor la forma con la que Photon Mapping trabaja para poder computar escenas con iluminación global.

Organización del trabajo

El desarrollo del Photon Mapping se ha llevado a cabo en todo momento con el trabajo simultáneo de los dos miembros del equipo. Se ha hecho esto con el fin de equiparar las horas trabajadas y, más importante, que todas las decisiones de diseño e implementación tuvieran el respaldo de ambos miembros.

En la siguiente tabla, se muestra el número de horas dedicadas a cada una de las fases del proyecto.

Tarea realizada	Horas dedicadas	
	Víctor Peñasco	Rubén Rodríguez
Comprensión y diseño del algoritmo	4	4
Comprensión del código proporcionado	2	2
Implementación del preprocesado	4	4
Implementación de la estimación de radiancia	5	5
Depuración y pruebas	20	20
Memoria (incluyendo renders finales)	20	20
Total	55	55

Referencias

[1] "Iluminación global"

https://moodle.unizar.es/add/pluginfile.php/2303601/course/section/356530/30_global_models.pdf

[2] "Photon Mapping"

https://moodle.unizar.es/add/pluginfile.php/2303601/course/section/356530/40_photon_mapping.pdf?time=1574916009098

[3] "Una Guía Práctica de Iluminación Global usando Mapas de Fotones"

<https://graphics.stanford.edu/courses/cs348b-00/course8.pdf>

[4] "Descripción del algoritmo de Photon Mapping"

https://en.wikipedia.org/wiki/Photon_mapping