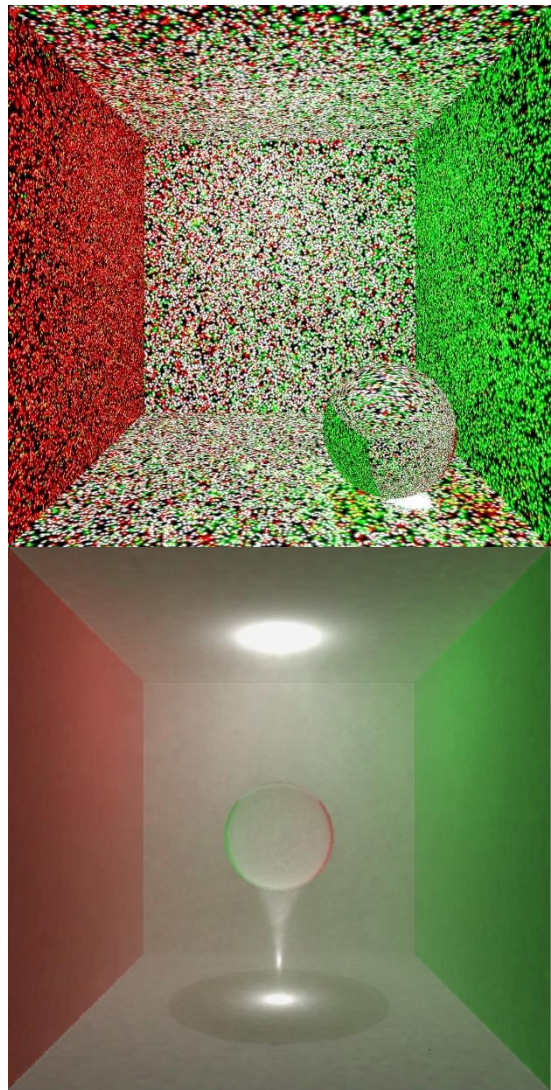
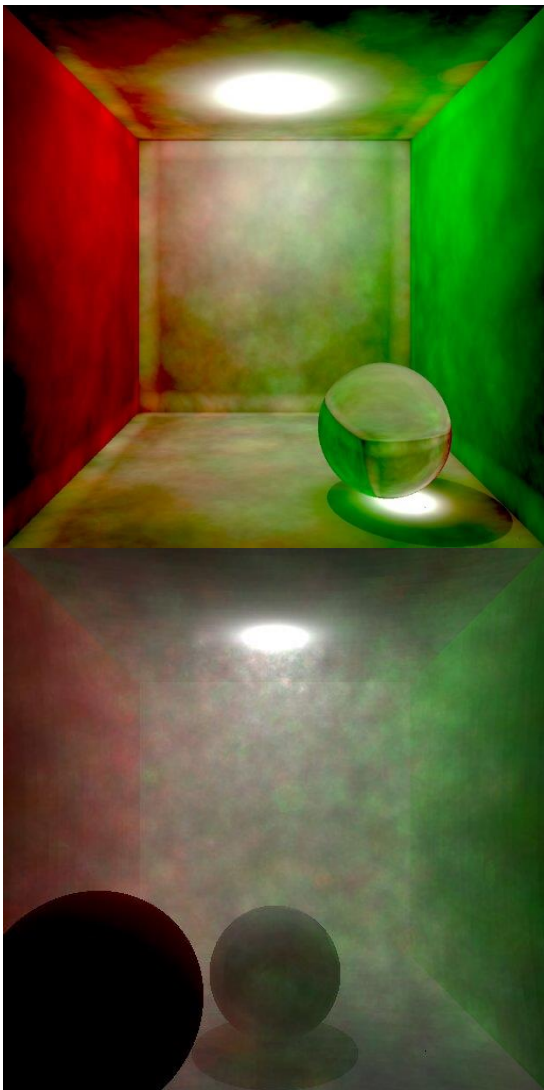


PHOTON MAPPING



Álvaro Fuentemilla Martínez, 699678
Víctor Labrador Ortíga, 701658
David Márquez Calavia, 700940

INTRODUCCIÓN

Photon mapping es un algoritmo en dos pasos de iluminación global, desacoplando representación de la iluminación de la geometría y almacenando la información de la luz indirecta en una estructura de datos denominada photon map, o mapa de fotones.

De este modo, se almacena en el mapa de fotones la iluminación indirecta de la escena, y posteriormente se calcula la luz global mediante la adición de la luz directa y la indirecta. Este algoritmo permite obtener efectos, como cáusticas, de una manera más sencilla que con path tracing.

GENERACION MAPA DE FOTONES

Se ha empleado el código proporcionado en Moodle para la realización del algoritmo.

El primer paso es trazar los fotones desde las fuentes de luz hacia la escena. Para ello hay que determinar cuantos fotones van a ser emitidos en base al número de luces en la escena y sus intensidades. Dicho número se ha calculado obteniendo primero la intensidad total de la escena, sumando todas las intensidades de las luces y posteriormente lanzando por cada luz el número de fotones totales por la intensidad de la luz a muestrear entre la intensidad total, es decir:

$$numFotonesLuz_i = numFotonesTotales \times \frac{intensidadLuz_i}{intensidadTotal}$$

Una vez calculado el número de fotones a lanzar por cada luz, hay que darles una dirección, dado que solo se trabaja con luces puntuales, esto se soluciona muestreando una esfera.

Esta parte del algoritmo de photon mapping consiste en guardar los fotones que se obtienen después de lanzar rayos desde las fuentes de luz al resto de la escena, en dos mapas de fotones distintos. Uno para las causticas y otro para la luz indirecta.

Respondiendo a la **Question 1.1**, este problema se resolvió de la siguiente manera. Siendo ξ un número real aleatorio entre 0 y 1.

$$\begin{aligned}\theta &= 2 \times \pi \times \xi \\ \Phi &= \arccos(1 - 2 \times \xi) \\ direccion &= \{\sin(\Phi) \times \cos(\theta), \sin(\Phi) \times \sin(\theta), \sin(\Phi)\} \\ direction &= \frac{direccion}{||direccion||}\end{aligned}$$

Esta manera distribuye uniformemente la dirección que seguirá un foton emitido por una fuente de luz, solucionando el problema de otros métodos para muestrear una esfera, que tienden a acumular más muestras en los polos.

Tras instanciar el fotón, hay que trazar el rayo que este va a seguir, lo cual esta implementado en la función *PhotonMapping::trace_ray* del código proporcionado. La siguiente explicación responde a **Question 1.2**.

A la función se le pasa el rayo trazado, el cual contiene la posición donde se encuentra la luz (y por consiguiente el inicio del camino del fotón) y la dirección aleatoria obtenida anteriormente, así como la energía de fotón y las dos listas que guardan los fotones globales y los que representan las cáusticas. Primero se comprueba que no se hayan realizado previamente el máximo de trazados permitidos en la escena, y en caso de haberlos hecho, se devuelve falso. Ahora hay que iterar el camino que sigue el fotón hasta que este muere. Primero se calcula la intersección con la escena, y

en caso de no colisionar se termina el bucle. Si ha colisionado, y el objeto tiene un material de tipo delta, es decir que redirige el rayo, implica que el fotón representa una partícula cáustica, por lo que hay que guardarse esa información, en este caso en un booleano.

Si no es un material delta, y no es su primer choque, lo cual implicaría iluminación directa, el fotón hay que guardarlo en su correspondiente mapa de fotones, por lo que si el booleano representando si es una partícula cáustica está a *true*, éste se almacena en dicho mapa de fotones, y en caso contrario, en el de fotones globales, almacenando en ambos casos el punto de la colisión, la dirección del fotón y la energía que este almacena.

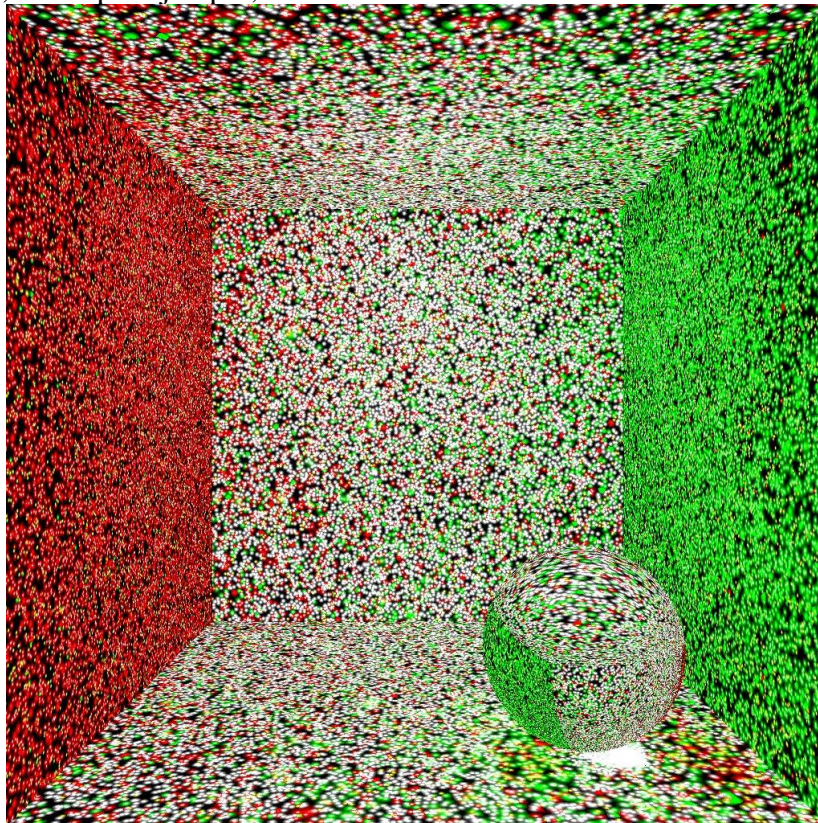
A continuación, se comprueba si el fotón se extingue, ya sea porque ha realizado el máximo número de rebotes, por ruleta rusa, o porque no es una partícula cáustica y representa iluminación directa.

En caso de no cumplir las anteriores especificaciones, se obtiene el rayo que representa el camino que sigue el fotón tras el rebote con el objeto, y se actualiza su energía y se vuelve a ejecutar lo anterior.

Si el tamaño de las dos listas de fotones es mayor del máximo permitido se devuelve *false* y termina la ejecución, y en caso contrario, se devuelve *true*.

En respuesta a la **Question 1.3**, teniendo que lanzar 1000 fotones, y habiendo 3 fuentes de luz, cada una con 1 Watt por ángulo sólido, se lanzarían $1000 \times \frac{1}{3} = 333$, y la energía de los fotones sería $\frac{1}{333}$ Watt, y posteriormente se multiplicaría por $\frac{4}{\pi}$, obteniendo así la energía final del fotón.

El tener un mapa de fotones permite posteriormente visionar una imagen con los fotones que han sido almacenados, como por ejemplo, en este caso.



ESTIMACION DE RADIANCIA

Para estimar la radiancia, hay que lanzar rayos desde la cámara hacia la escena, por todo el plano de proyección. A partir de este rayo se calcula la colisión del mismo con la escena y se almacena la luz directa en ese punto, y posteriormente se añade la iluminación global y la de las cáusticas, empleando ambos mapas de fotones.

En caso de que el rayo colisione con un objeto con material delta, se ha de trazar el rayo de salida hasta la siguiente colisión. Este procedimiento se sigue hasta que el rayo no encuentre una colisión con un objeto de material delta, donde se procederá a aplicar la iluminación directa de las luces puntuales empleando la ecuación de render.

$$L_o(x, w_i) = \int_{\Omega} L_i(x, w_i) f_r(x, w_o, w_i) |n \cdot w_i| dw_i$$

Siendo L_i el aporte de la luz, calculado mediante la potencia de la luz entre la distancia en el punto x , f_r las propiedades del material, y el producto escalar de la norma l_y la dirección del punto hacia la luz. Dada la propiedad aditiva de la luz, se suman los resultados de cada luz puntual y se obtiene el aporte total de la luz directa.

Ahora se calcula la estimación de radiancia con los mapas de fotones generados tanto para la luz indirecta como para las causticas. Mediante la ecuación de la radiancia reflejada:

$$L_o(x, \omega_o) = \int_{\Omega} f_r(\omega_i, \omega_r) \frac{d^2 \Delta \phi(x, \omega_i)}{dA_i}$$

La integral se aproxima con el sumatorio del flujo de los n fotones más cercanos dividido por el área de la geometría que los contiene.

Al principio se utilizó una circunferencia para buscar los fotones más cercanos pero posteriormente se cambió al área de un cono.

$$L_o(x, \omega_o) = \frac{1}{\Delta A} \sum_{p=1}^n f_r(\omega_p, \omega_r) \Delta \phi(x, \omega_i)$$

Esta aportación se suma a la luz directa y se multiplica por el coeficiente W , que en el caso de que no haya pasado el rayo por ninguna superficie delta será 1, si para por alguna tendrá el valor correspondiente a su coeficiente de reflexión correspondiente.

Para calcular la radiancia final hay que tener en cuenta la luz directa, la luz indirecta y las causticas.

La luz indirecta se busca en el mapa de fotones de la luz indirecta, se buscan los fotones vecinos, se suma el flujo de los fotones y se divide por el area de la circunferencia que lo contiene. Primero se probó usando una circunferencia y posteriormente un cono.

Para sumar la luz de las causticas se sigue el mismo procedimiento que para la luz indirecta pero buscando los fotones vecinos en el mapa de fotones de las causticas.

La luz directa se calcula con el primer rebote de los fotones.

Question 2.1.

$$L_0(x, w_i) = \int_{\Omega} L_i(x, w_i) f_r(x, w_o, w_i) |n \cdot w_i| dw_i$$
$$L_o(x, \omega_o) = \int_{\Omega} f_r(\omega_i, \omega_r) \frac{d^2 \Delta \phi(x, \omega_i)}{dA_i}$$

El término que cambia es la luz incidente, para photon mapping se aproxima mediante el mapa de fotones, que guarda el flujo. Dividiendo por el área que contiene los fotones más cercanos, el sumatorio de dichos fotones.

Para medios participativos se tiene que dividir el flujo por el área de una esfera. También se tiene en cuenta la transmitancia y los coeficientes de absorción y scattering.

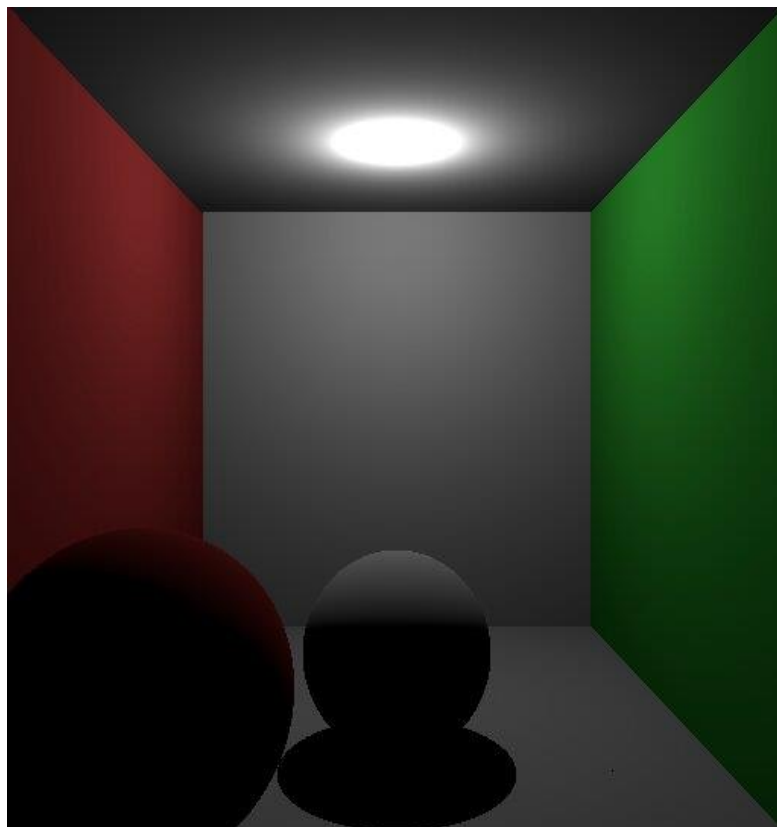
$$L_i(x \mapsto \varpi) \approx \sum_{p=1}^k p(x, \varpi_p, \varpi) \frac{\phi(x, \varpi_p)}{\frac{4}{3} \pi r^3}$$

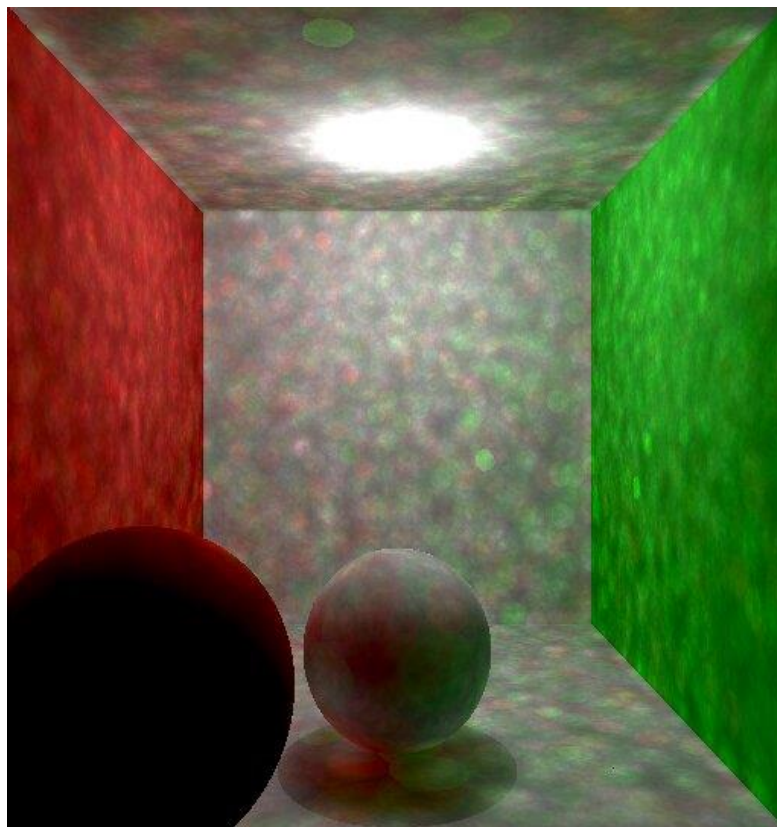
Question 2.2

Como se ha comentado en el apartado anterior, esto se consigue al dividir la suma del flujo de los n fotones más cercano entre el área de la circunferencia que los contiene o el área de la esfera para el caso de medio participativos.

Question 2.3

La luz directa con ray tracing se calcula directamente con la ecuación de render sin embargo con photon mapping se hace estimación de radiancia. Por lo tanto es mas correcta la forma en que la calcula ray tracing.





Question 2.4

Si se hace la media de renders de photon mapping no se puede llegar a la solución correcta como se llegaría utilizando path tracer. Habría que lanzar infinitos fotones para obtener la solución correcta, lo que significa que cuanto mayor es el número de fotones mayor calidad y precisión de imagen se consigue.





Question 2.5

El resultado obtenido es parecido ya que la luz que ilumina el objeto no es directa, sin embargo en el resto de la imagen sí que se aprecia diferencia.

El resultado obtenido es similar, ya que la luz que ilumina ese objeto no es directa, luego no hay mucha diferencia entre ambas. Sin embargo, para el resto de la escena, el resultado de ray tracing sigue siendo más preciso. Se puede apreciar por ejemplo en las sombras de la imagen de ray tracing.

Question 2.6

Photon mapping permite generar causticas más fácilmente que path tracing ya que en path tracing habría que implementar path tracing bidireccional. Sin embargo para conseguir buena calidad de la imagen es necesario lanzar un número muy alto de fotones lo cual realentiza mucho el render de la imagen.

MEDIOS PARTICIPATIVOS

Se ha seguido el mismo procedimiento utilizado en path tracer para implementar medios participativos en photon mapping. Adaptando los coeficientes de absorción y de scattering al tamaño de la imagen.

A lo largo de la explicación se va a emplear la siguiente notación con respecto a las propiedades del medio:

$$\sigma_s = \text{coeficiente de scattering}, \sigma_a = \text{coeficiente de absorción},$$

$$\sigma_t = \sigma_s + \sigma_a = \text{coeficiente de extinción}$$

Los coeficientes se han de elegir acorde con el tamaño de la escena, de modo que σ_t sea aproximadamente $\frac{1}{m}$, siendo m el tamaño máximo al que se puede producir una colisión en una escena. Para simplificar, se aplica un medio participativo a toda la escena en vez de añadir una geometría con las propiedades del medio. No obstante, el concepto es el mismo.

Primero se ha creado un nuevo photon map, el cual se encargará de almacenar los fotones que han interactuado con el medio.

A continuación, se ha tenido que modificar la función de *PhotonMapping::trace_ray*. Por simplicidad se ha dejado un caso que responde a cuando no hay medios participativos ($\sigma_t = 0$), y otro que se ejecuta cuando si lo hay ($\sigma_t > 0$).

Al llamar a *PhotonMapping::trace_ray* se calcula la intersección con la escena, lo cual proporciona la distancia a la que colisiona el fotón. Se ha de calcular a que distancia se va a producir la interacción con el medio, si es que se produce, ignorando el medio antes de que suceda dicha interacción, y eso se calcula de la siguiente forma:

$$\text{distancia} = \frac{-\log(1 - \xi)}{\sigma_t}$$

Siendo ξ un número real aleatorio entre 0 y 1.

En caso de que la distancia a la que se produce la interacción sea menor que la distancia a la que ocurre la colisión con la escena, hay que calcular que evento ha ocurrido, y almacenar la energía necesaria.

Después se calcula que tipo de evento ha sucedido siguiendo la siguiente regla:

$$\xi < \frac{\sigma_s}{\sigma_t} \quad : \text{Evento de scatter} \qquad \xi > \frac{\sigma_s}{\sigma_t} \quad : \text{Evento de absorción}$$

$$L_i(x, \varpi) \approx \sum_{t=0}^{S-1} T_r(x \leftrightarrow x_t) \sigma_s(x_t) L_i(x, \varpi) \Delta_t + T_r(x \leftrightarrow x_t) L_i(x_s, \varpi)$$

En el evento de scatter se calcula la energía del fotón multiplicando la energía actual por el coeficiente de scatter y por el albedo del medio, y se guarda en la lista de fotones del medio participativo, poniendo su posición en el lugar donde se ha dado la interacción con el medio según la distancia explicada anteriormente. Acto seguido, hay que volver a lanzar un fotón en otra dirección con la energía restante (energía del fotón original menos la que se ha quedado en el medio).

Dado que el medio es homogéneo, su función de fase es isotrópica, por lo que para calcular la nueva dirección del fotón se muestrea la esfera uniformemente como se ha hecho al muestrear las luces para trazar los fotones.

Para calcular la energía con la que continua el fotón se resta la energía original del fotón con la energía consumida en el evento de scatter.

Como la función es iterativa en la siguiente iteración el fotón tendrá la dirección y energía calculadas anteriormente.

En el evento de absorción la idea es similar pero utilizando el coeficiente de absorción para calcular la energía, es decir, se multiplica la energía del fotón por el coeficiente de absorción y por el albedo del medio y se guarda en la lista de fotones del medio participativo. No hay que calcular ninguna nueva dirección ya que al tratarse de un evento de absorción la dirección del fotón es la misma, por lo tanto solo hay que calcular la nueva energía restando la energía original del fotón menos la energía producida por el evento de absorción.

Si no ocurre ningún evento de scatter la energía almacenada en las listas de fotones para la luz indirecta como para las causticas se tiene que tener en cuenta el albedo del medio por lo que antes de guardarse se multiplica la energía por el albedo del medio.

En la función *PhotonMapping::preprocess* se guardan los fotones que han interactuado con el medio participativo en un kd-tree, que será el que será utilizado posteriormente para la estimación de radiancia.

Para la estimación de radiancia se ha utilizado la técnica de ray marching.

Hay que tener en cuenta la transmitancia para la luz indirecta y la luz de las cáusticas, se multiplica la transmitancia.

$$T_r(x \leftrightarrow x_t) = e^{-(x-x_t)\sigma_t}$$

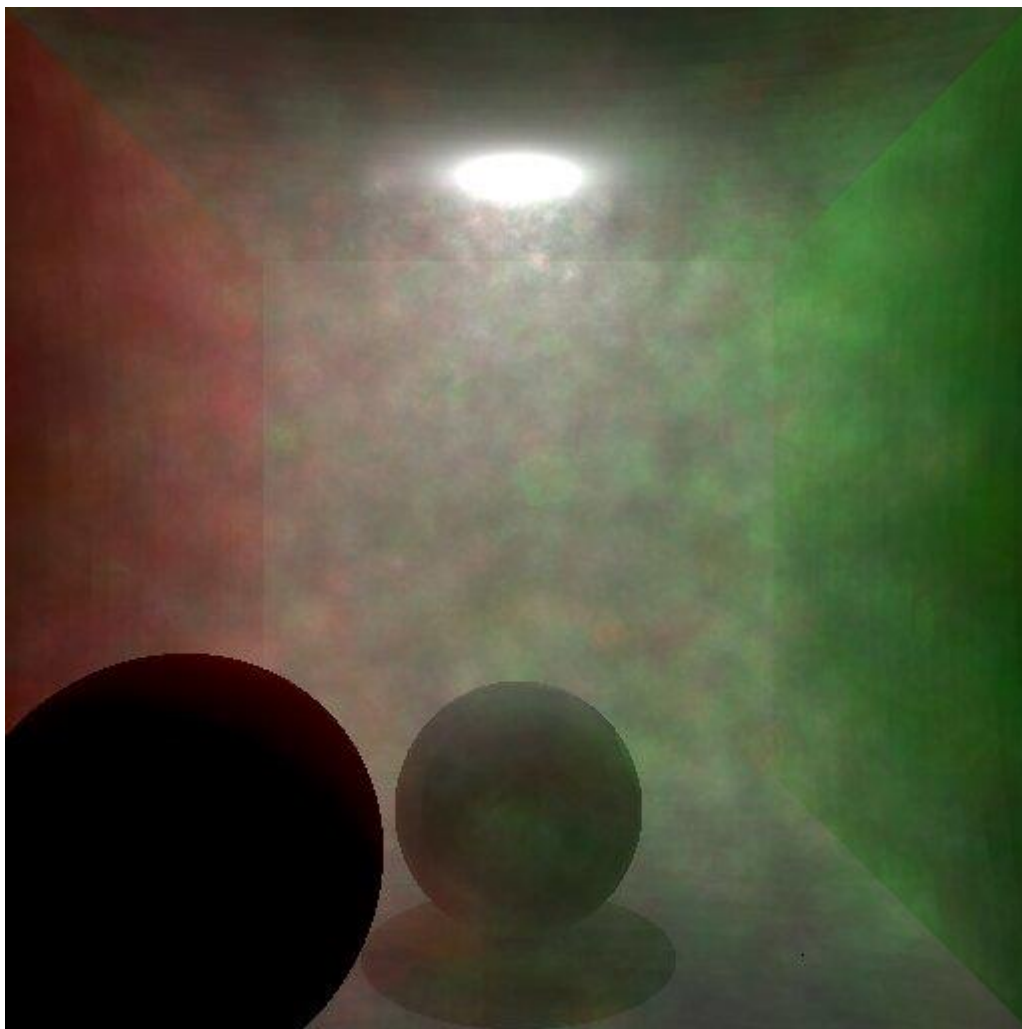
Para hacer ray marching, se trazan rayos desde la cámara hasta la escena a lo largo del plano de proyección, y se buscan fotones en el mapa del medio participativo cada 0.1 de distancia a lo largo del rayo. En cada punto se buscan n fotones más cercanos, se suma el flujo de los fotones y se divide el sumatorio entre el área de una esfera.

$$L_i(x \mapsto \varpi) \approx \sum_{p=1}^k p(x, \varpi_p, \varpi) \frac{\phi(x, \varpi_p)}{\frac{4}{3} \Pi r^3}$$

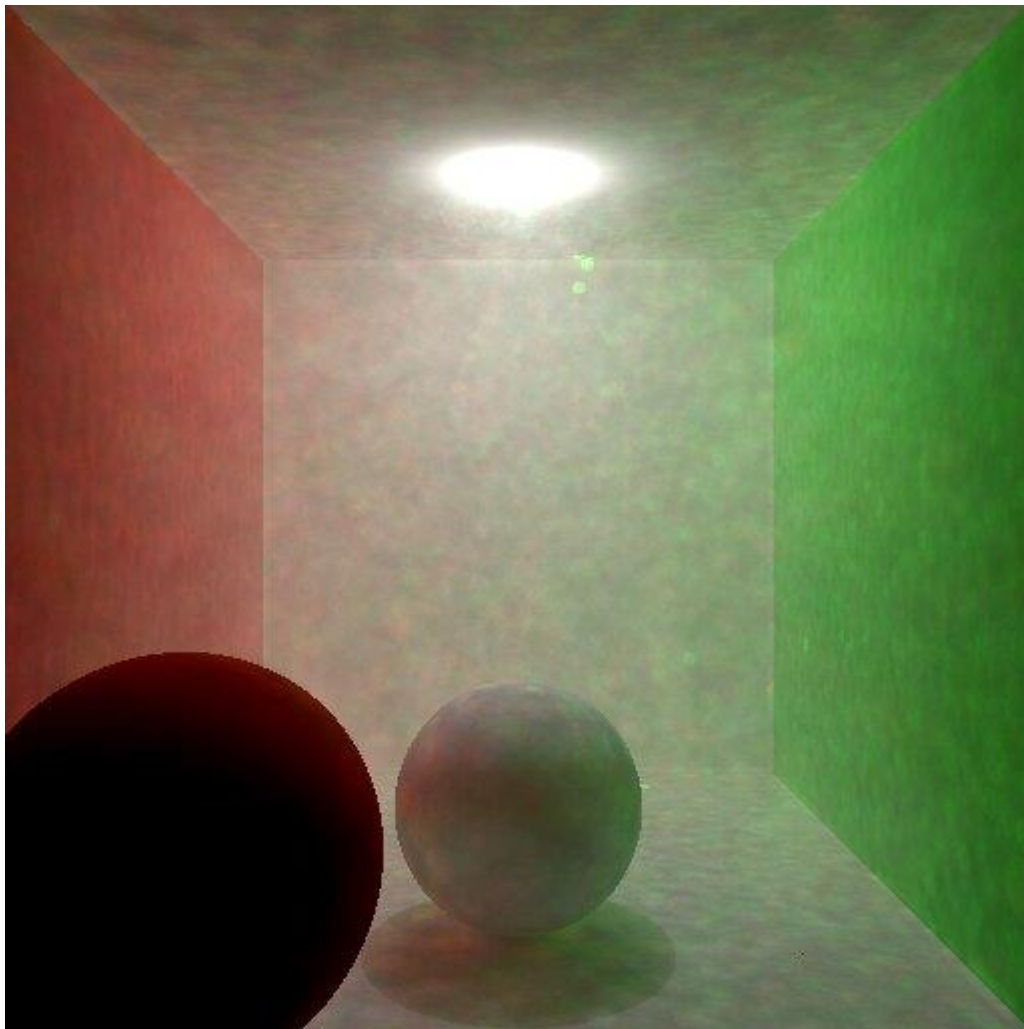
Con esta técnica es probable que se este sumando varias veces el flujo de un mismo fotón en dos puntos diferentes. Para arreglar este fallo se comprueba que la luminancia no sea superior a 1 en alguno de los 3 canales, y en caso de que este sea superior se actualiza su valor a 1.

Se intentó implementar un ray marching adaptativo para corregir este error en el que la distancia que se avanza en el rayo para cada iteración fuera dos veces el radio de la esfera en la que se había buscado los n vecinos más cercanos para la iteración anterior, pero provocaba mucho ruido y se decidió utilizar la técnica comentada anteriormente.

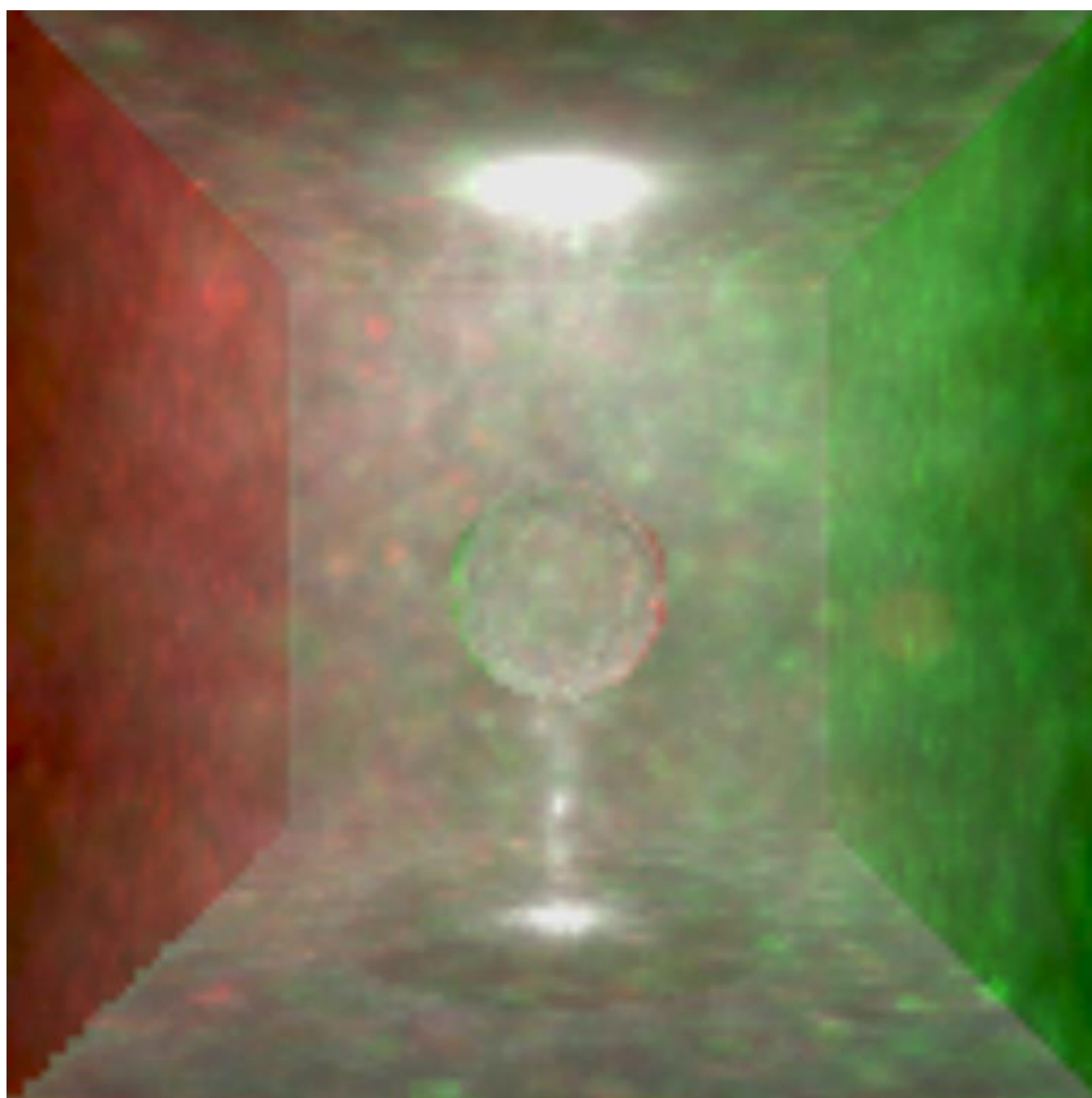
A continuación, se muestran algunas imágenes con medio participativo.



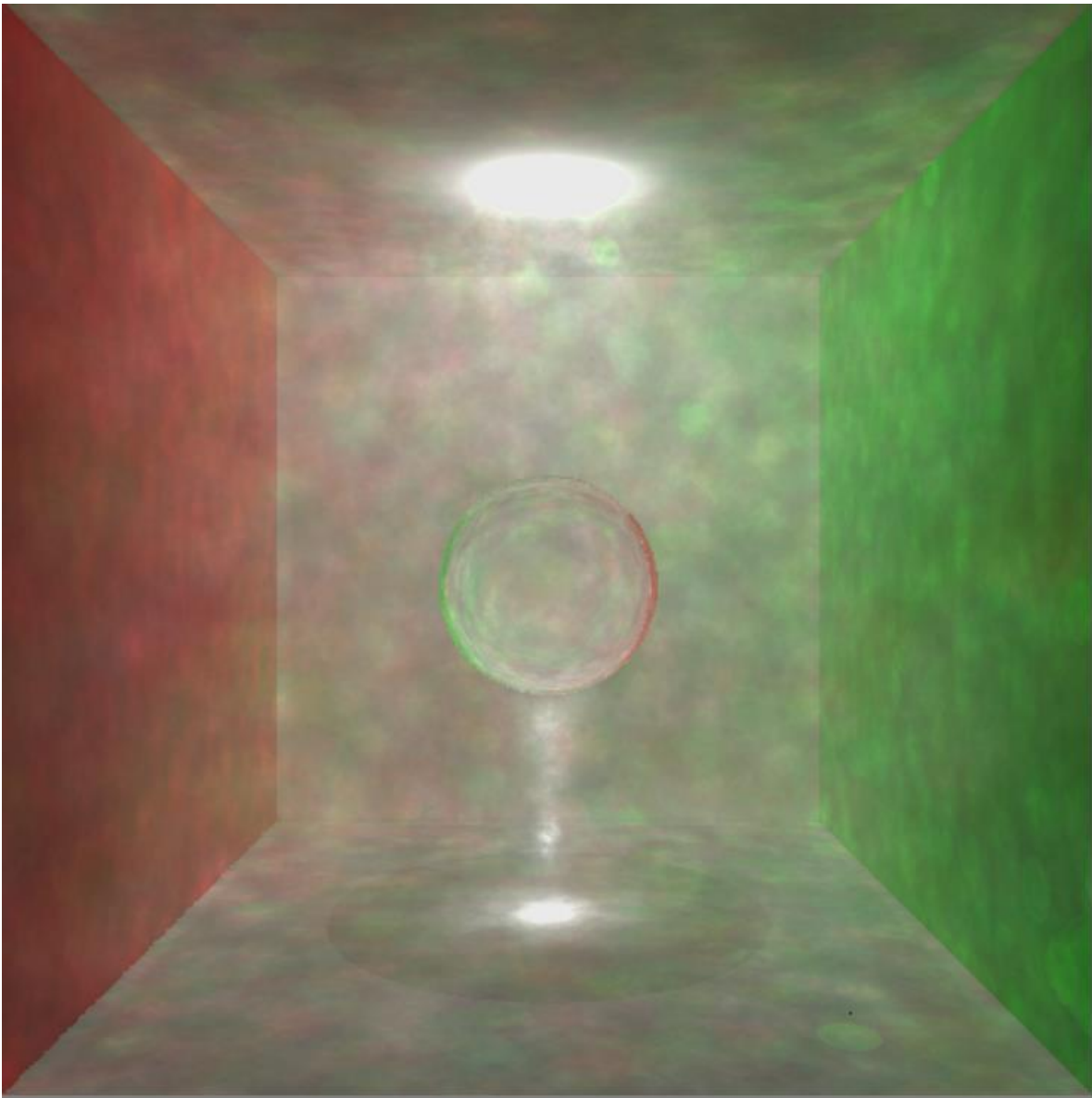
100.000 fotones lanzados, 50 fotones vecinos para estimación de radiancia.



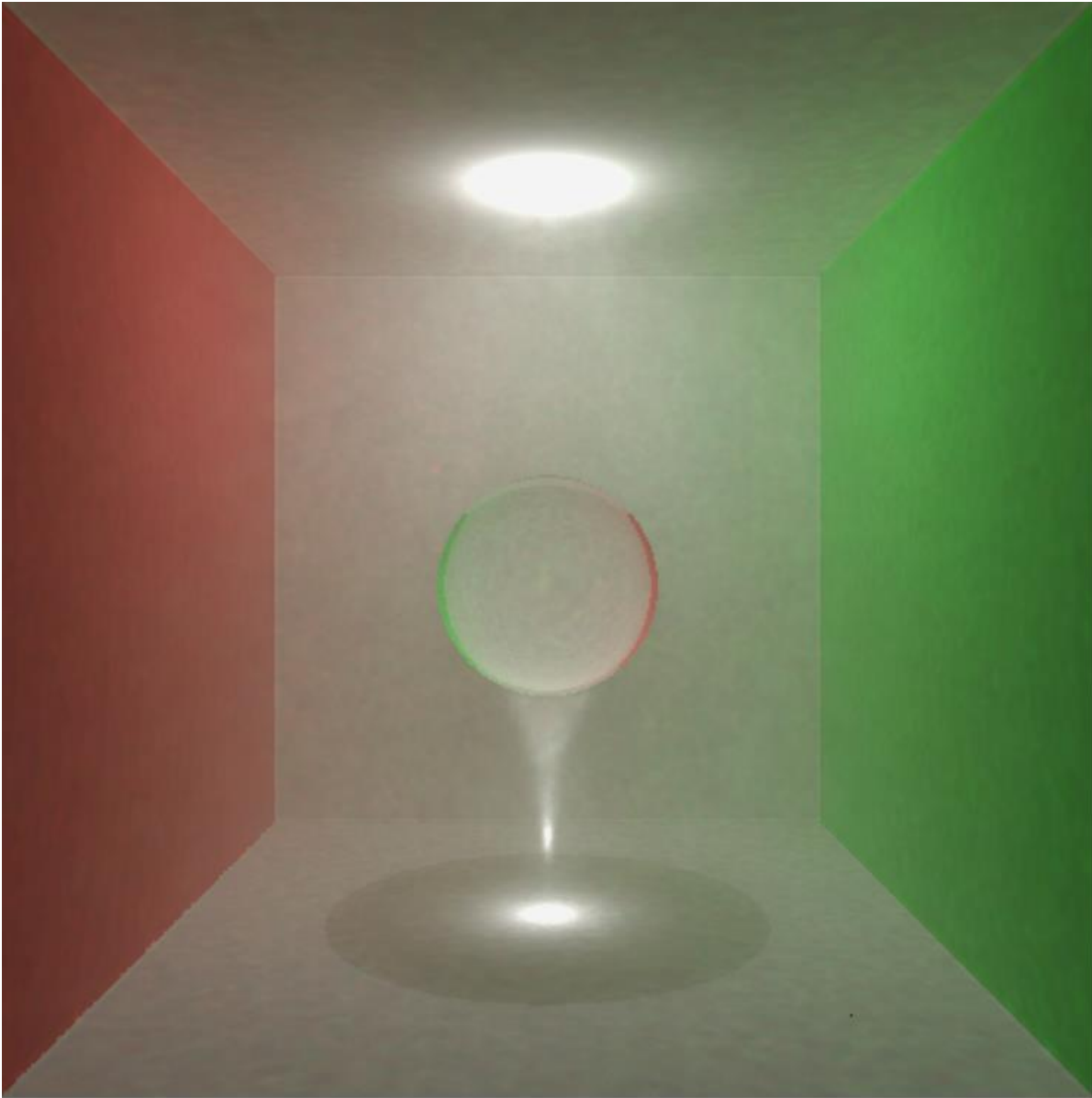
1.000.000 fotones lanzados, 100 fotones vecinos para estimación de radiancia.



100.000 fotones lanzados, 10 fotones vecinos para estimación de radiancia.



100.000 fotones lanzados, 50 fotones vecinos para estimación de radiancia.



10.000.000 fotones lanzados, 300 fotones vecinos para estimación de radiancia

En las tres imágenes anteriores se puede apreciar una cáustica volumétrica, lo cual se debe a que los fotones que atraviesan la esfera de cristal interactúan con el medio, dando lugar a este efecto.

BIBLIOGRAFÍA

- Apuntes de la asignatura
- <https://cs.dartmouth.edu/wjarosz/publications/dissertation/chapter7.pdf>
- <http://corysimon.github.io/articles/uniformdistn-on-sphere/>
- <https://www.sciencedirect.com/science/article/pii/S0022407303001146>
- https://www.fing.edu.uy/inco/cursos/cga/Clases/2018/Photon_mapping.pdf
- <http://graphics.ucsd.edu/~henrik/papers/book/>
- <http://mathworld.wolfram.com/SpherePointPicking.html>
- <https://graphics.stanford.edu/courses/cs348b-00/course8.pdf>
- https://web.cs.wpi.edu/~emmanuel/courses/cs563/write_ups/zackw/photon_mapping/Photon-Mapping.html
- https://www.fing.edu.uy/inco/cursos/cga/Clases/2017/Photon_mapping.pdf