

Informática Gráfica

Photon Mapping

Mayo Peribáñez, Carlos; NIP 799083

Bielsa Uche, Jaime; NIP 819033

13 de enero de 2024

Ingeniería Informática

Escuela de Ingeniería y Arquitectura

Universidad de Zaragoza

Índice

1. Ecuación de render	3
2. Comparación next-event estimation y fotones almacenados en el mapa	4
3. Efectos de iluminación	5
4. Análisis de resultados	7
4.1 Fijamos número de vecinos a 10	7
4.2 Fijamos número de fotones a 100000	9
5. Extras	12

1. Ecuación de render

$$L_o(x, \omega_o) = L_e(x, \omega_o) + \int_{\Omega} L_i(x, \omega_i) f_r(x, \omega_i, \omega_o) |n \cdot \omega_i| d\omega_i$$

El término $L_e(x, \omega_o)$ es la luz que emite el material, $L_i(x, \omega_i)$ la luz que le llega al punto en el que se calcula, $f_r(x, \omega_i, \omega_o)$ las propiedades del material y $|n \cdot \omega_i|$ su geometría.

En photon mapping no consideramos luces de área, por lo que para las figuras, $L_e(x, \omega_o) = 0$. Al igual que en el path tracer, el cálculo de la luz se realiza a partir de la suma de las luces directa e indirecta. El cálculo de la luz indirecta se basa en un algoritmo de dos pasos en los que se confecciona y utiliza un mapa de fotones: desde cada fuente de luz, se lanzan rayos que rebotan con superficies hasta ser absorbidos. Conforme rebotan en superficies (total o parcialmente) difusas, generan fotones que guardan un flujo asociado a la luz incidente $L_i(x, \omega_i)$ de las fuentes de luz y al ángulo de incidencia $|n \cdot \omega_i|$. Se acumula el ángulo y la BRDF $f_r(x, \omega_i, \omega_o)$ para realizar scattering del resto de fotones del rayo.

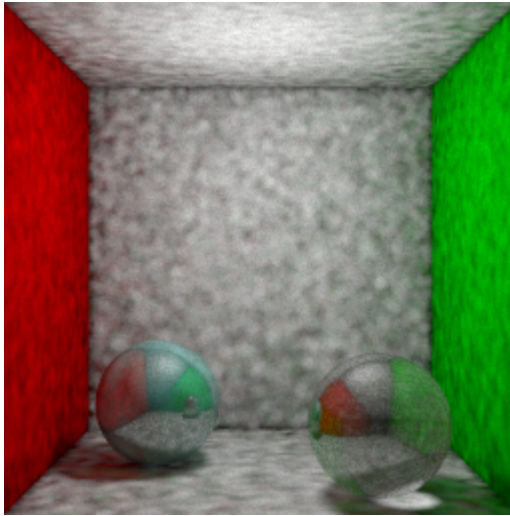
Luego, desde la cámara se lanzan otros rayos de la forma habitual. Al encontrar una superficie difusa, obtenemos nuestra estimación del color en base al componente difuso de la superficie $f_r(x, \omega_i, \omega_o)$ y al flujo de los fotones más cercanos.

Para calcular la luz directa de una manera más certera, utilizamos next-event estimation: descartamos el primer fotón que genera cada rayo si impacta de forma directa con una superficie difusa. Luego, al lanzar rayos desde la cámara, además de buscar los fotones, buscamos la luz directa de cada píxel con las fuentes de luz, trazando rayos de sombra.

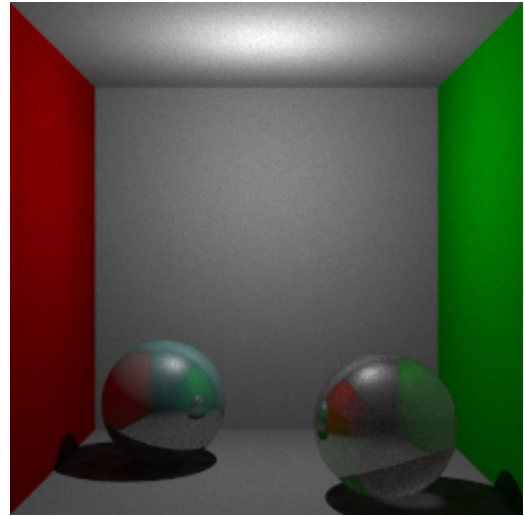
2. Comparación next-event estimation y fotones almacenados en el mapa

Inicialmente el programa almacenaba fotones en cada uno de los rebotes que estos daban, más tarde se hizo que el primer rebote que hace un fotón sobre una superficie plana no se almacene y cuando se lanzan los rayos desde la cámara se utiliza next-event estimation para calcular la luz directa y el mapa de fotones para calcular la luz indirecta.

Los resultados tras aplicar tone mapping gamma 2.2 son los siguientes:



Luz directa mapa de fotones



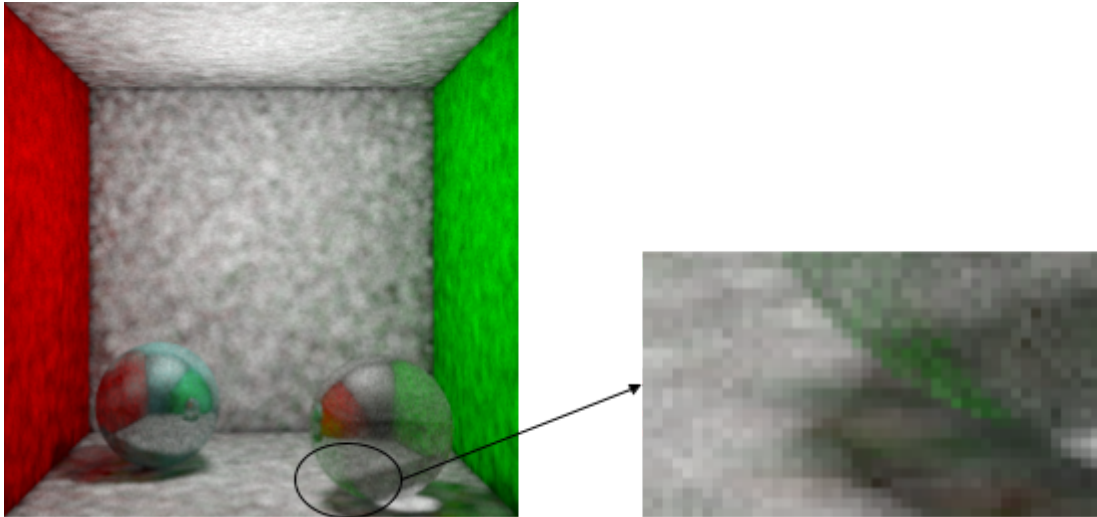
Luz directa next-event estimation

Ambas imágenes están generadas con 100000 fotones y 50 vecinos.

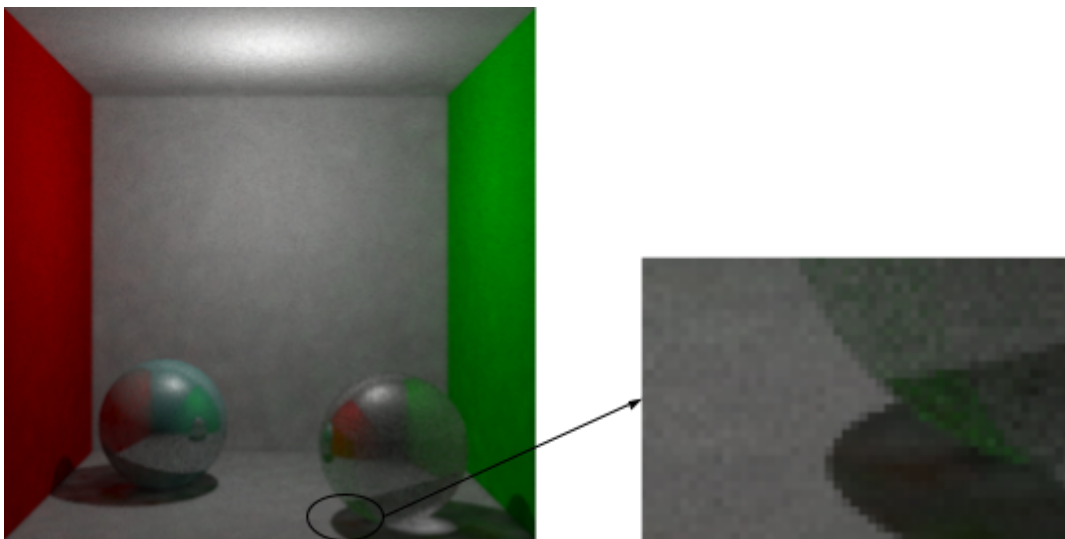
Cómo se puede ver, el resultado mejora mucho cuando se utiliza next-event estimation.

3. *Efectos de iluminación*

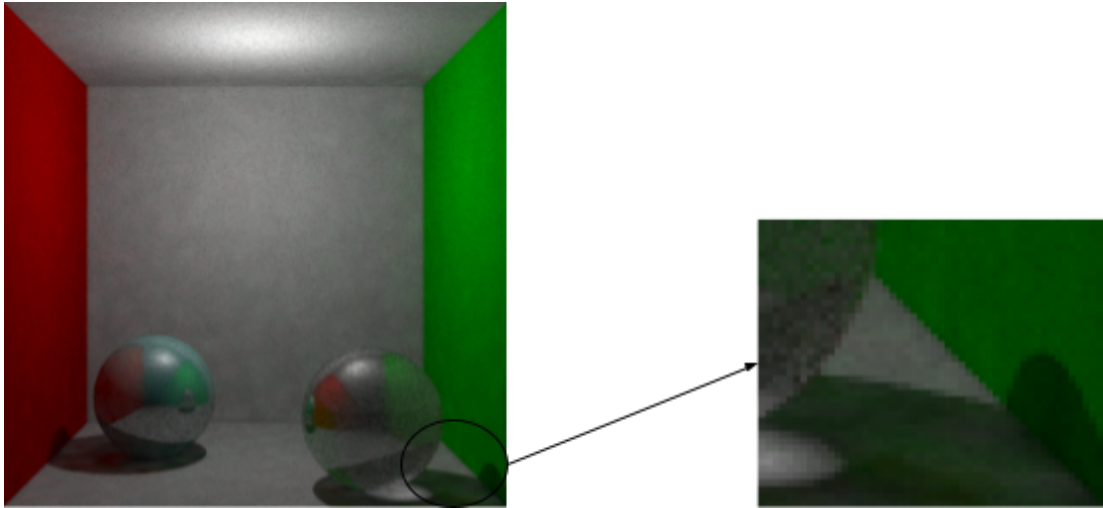
El photon mapping consigue sombras blandas cuando usas el mapa de fotones para calcular la luz directa, ya que no busca intersecciones entre el punto en el que ha hecho hit el rayo y las luces puntuales, además de añadir intensidad de fotones vecinos que pueden no estar dentro de la sombra.



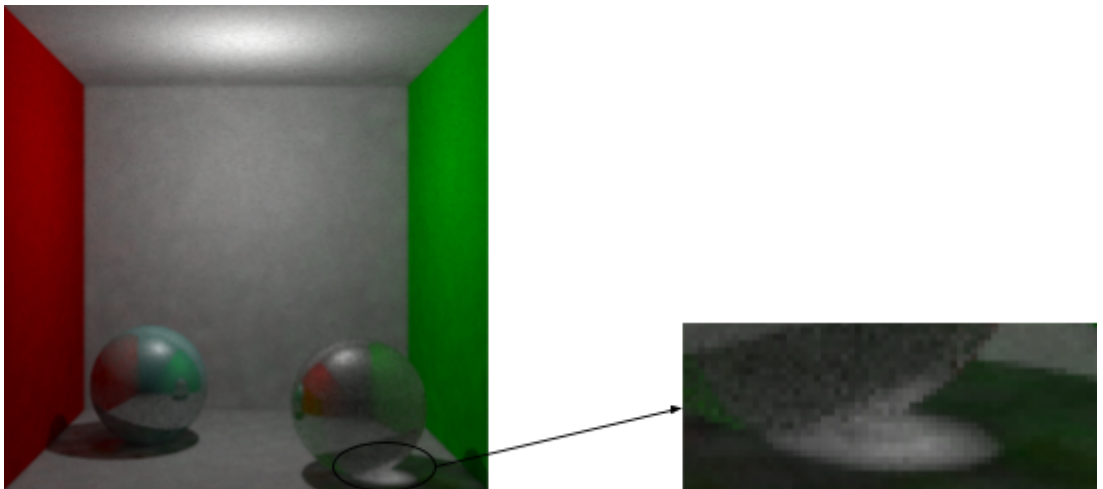
Al usar next-event estimation se consiguen sombras duras.



El color bleeding se puede apreciar claramente en el verde que se genera en la sombra de la esfera derecha. Podría apreciarse más todavía, pero se ha restringido la suma de la contribución de los fotones cercanos a sólo los que pertenezcan a la misma figura.



Las cáusticas se pueden observar en la esfera de la derecha y son más fáciles de conseguir que en el path tracer.



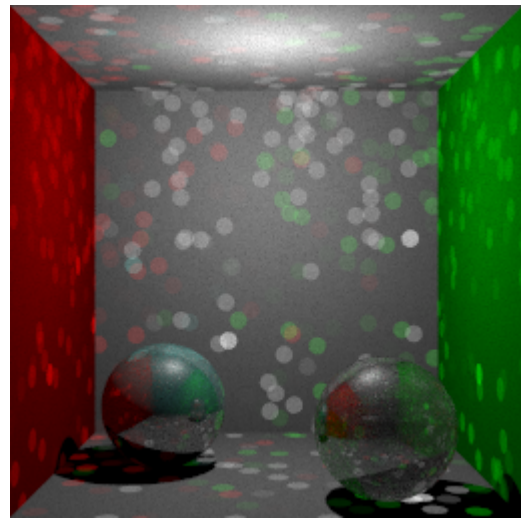
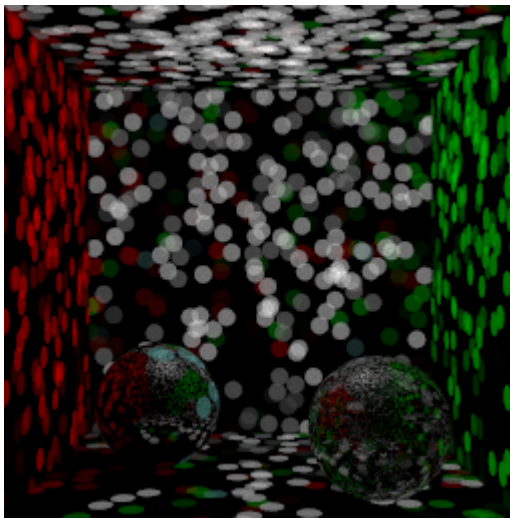
4. *Análisis de resultados*

En este apartado se van a mostrar y analizar diferentes pruebas sobre la misma escena, modificando varios valores. A todas las imágenes se les ha realizado tone mapping gamma 2.2 y en todas ellas se han lanzado 32 rayos por píxel.

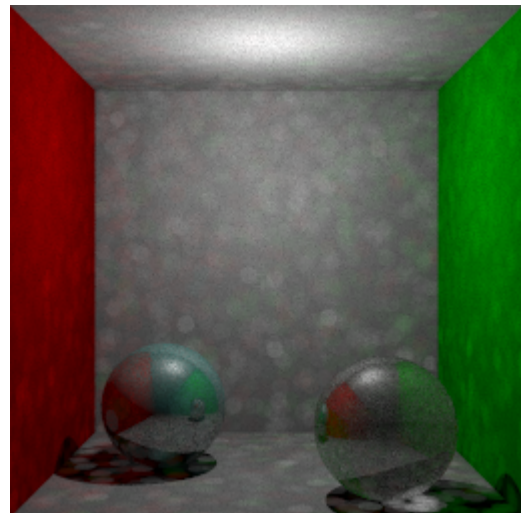
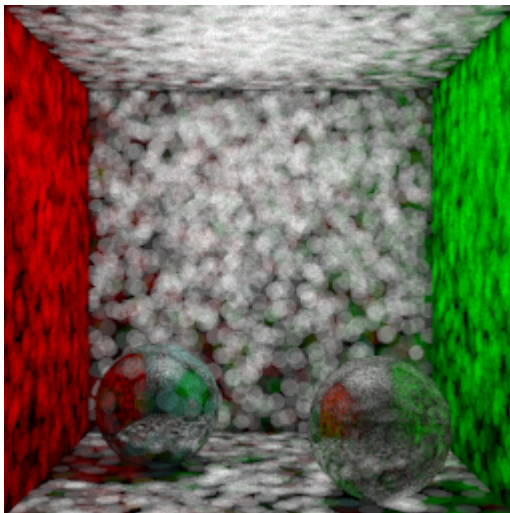
En la columna de la izquierda se encuentran las imágenes calculando la luz directa mediante el mapa de fotones, en la de la derecha las que se calcula mediante next-event estimation.

4.1 *Fijamos número de vecinos a 10*

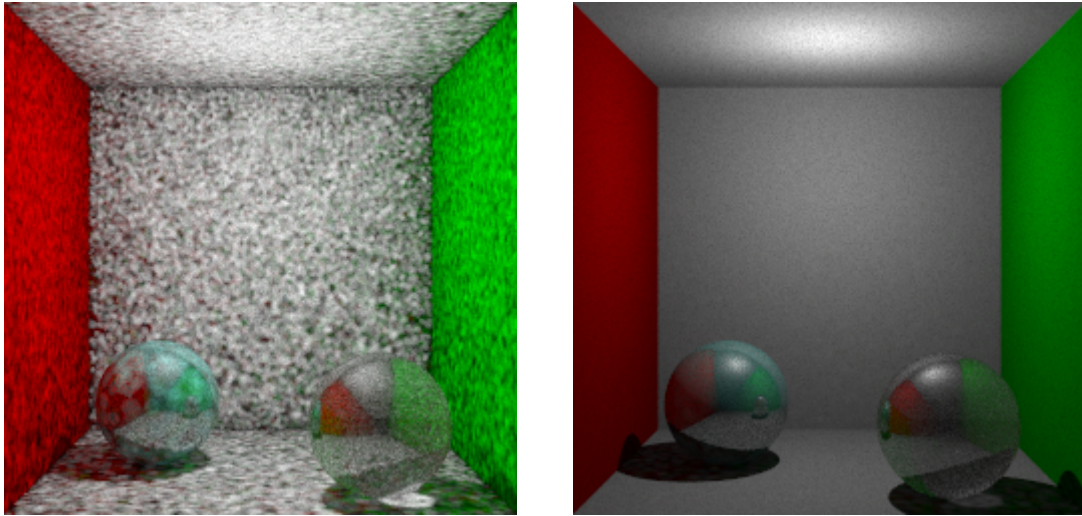
1k fotones



10k fotones



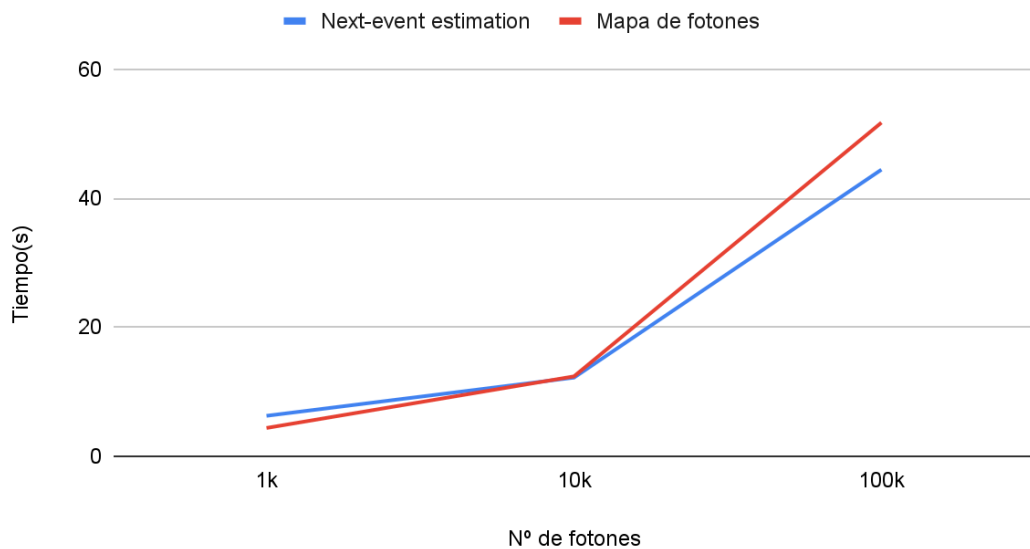
100k fotones



En ambos casos el número de fotones que se lanzan influye considerablemente en la calidad de la imagen generada. Conforme aumentamos el número de fotones, la energía que tiene cada uno de ellos es menor, ya que se reparten la energía de la fuente de luz, lo que acaba mejorando la convergencia de la imagen final.

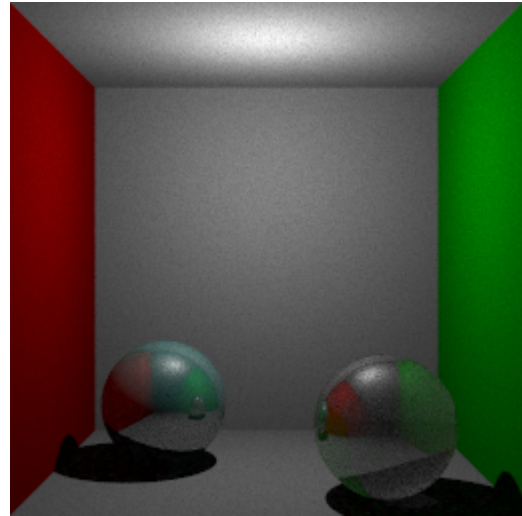
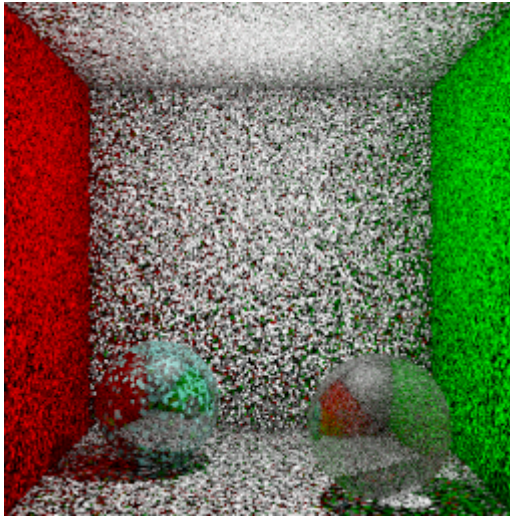
En cuanto al tiempo de ejecución, en el siguiente gráfico se va aumentando.

Nº de vecinos = 10

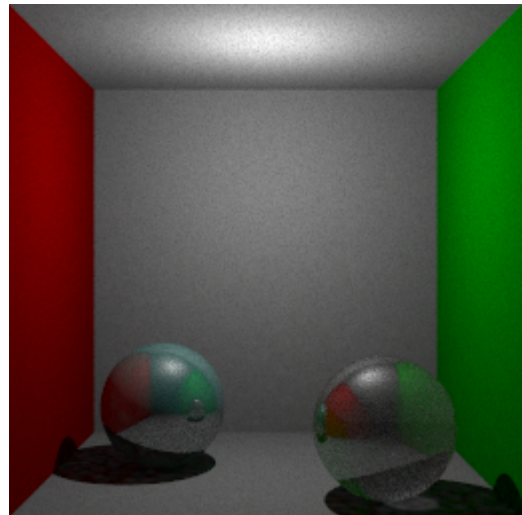
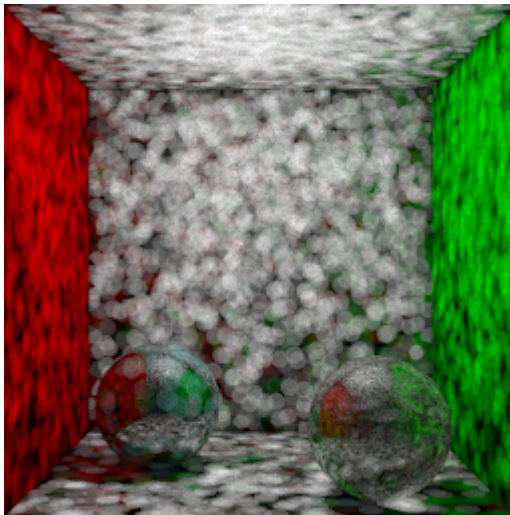


4.2 Fijamos número de fotones a 100000

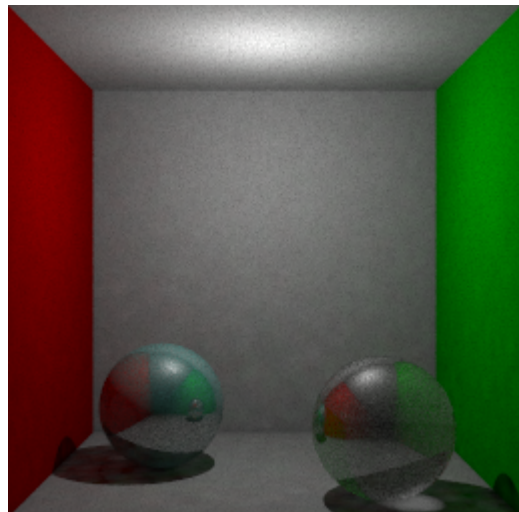
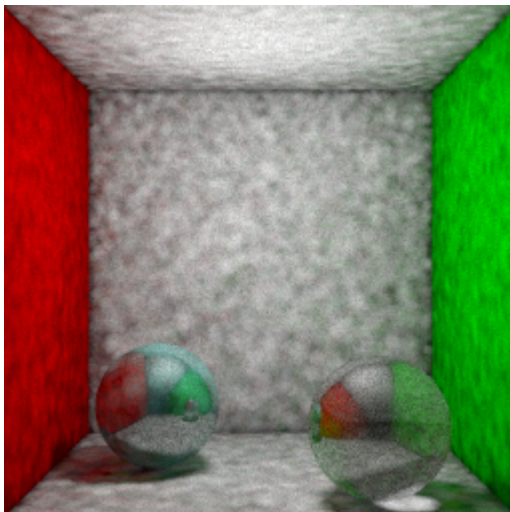
1 vecino



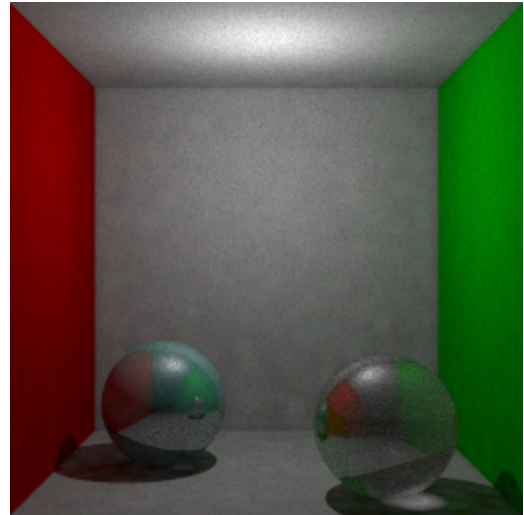
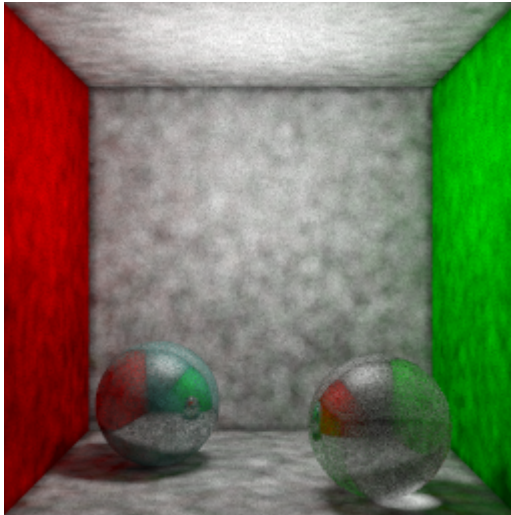
10 vecinos



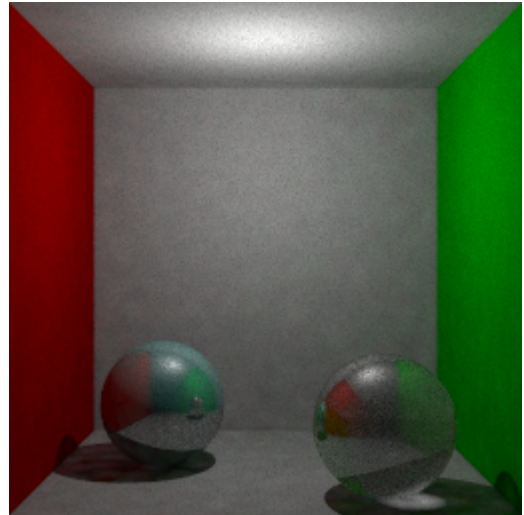
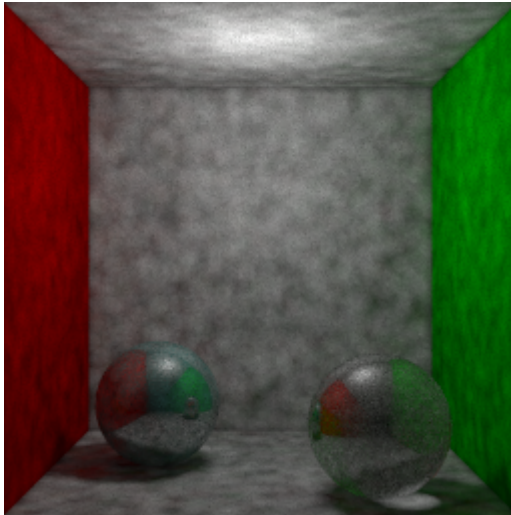
50 vecinos



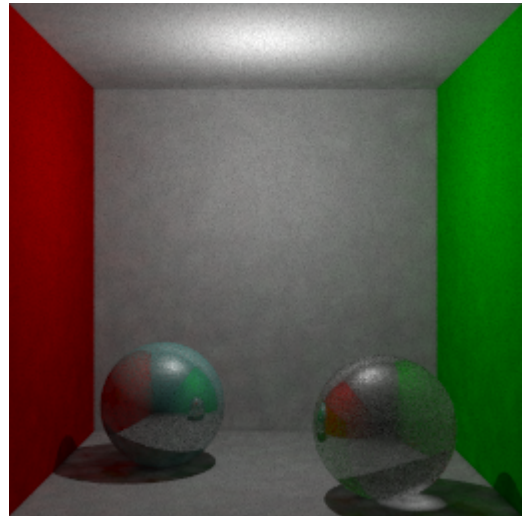
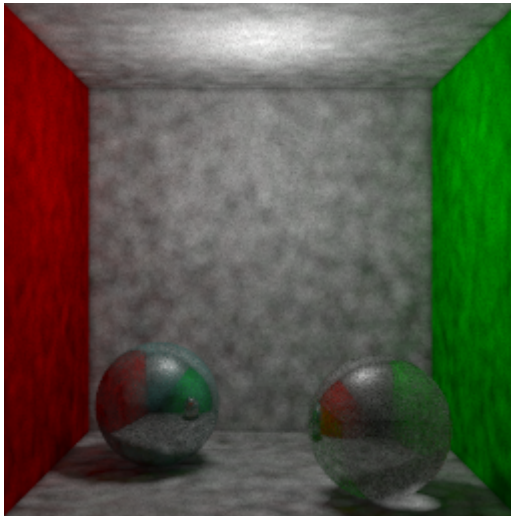
100 vecinos



500 vecinos



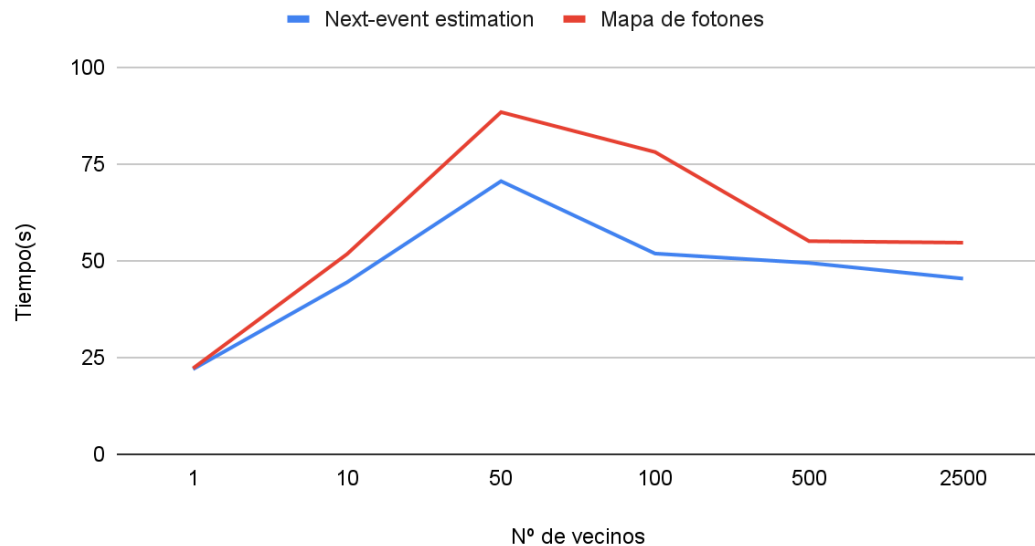
2500 vecinos



Las imágenes procesadas usando solamente el mapa de fotones van convergiendo conforme aumentamos el número de vecinos. Las generadas usando next-event estimation con pocos vecinos es prácticamente solo la luz directa de la imagen, al subir el número de vecinos se muestra más luz indirecta y aparecen los efectos asociados a ella, cómo las cáusticas.

En cuanto al tiempo de ejecución, en el siguiente gráfico se va aumentando.

Nº de fotones = 100.000



Aunque el menor tiempo se obtiene al buscar solo un fotón, el mayor tiempo no se encuentra al buscar muchos fotones sino al buscar una cantidad moderada, siendo el peor caso 50 fotones. Los resultados de esta gráfica se pueden deber al coste de la función `nearest_neighbors` del KDTree, que al hacer una búsqueda espacial con intervalos puede ser que tarde menos si se busca un conjunto más grande.

5. *Extras*

Todos los extras comentados en el path tracer han sido implementados también en el photon mapping: definición de nuevas figuras y optimización por hilos.

Además, para la carga de escenas y distintas configuraciones de los renderizadores, se ha creado un nuevo formato de escena **.objx** derivado del formato de definición de objetos **.obj**, donde se añaden los parámetros de configuración, cámara, fuentes de luz y figuras, estas últimas con su apariencia.

Una descripción más detallada de este formato de archivo para generar nuevas escenas se puede encontrar en OBJX.md.

En la carpeta scenes hay algunos ejemplos sencillos de escena.