Informática Gráfica

Path Tracer

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

Para comenzar se va a escribir la ecuación de render y describir cada una de sus partes.

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

El término $Le(x, wo)$ es la luz que emite el material, $Li(x, wi)$ la luz que le llega al punto en el que se calcula, $fr(x, wi, wo)$ las propiedades del material y $|n \cdot wi|$ su geometría.

En el caso del path tracer la emisión del material solo se tiene en cuenta en las luces de área.

La integral en fenómenos especulares se aproxima utilizando la estimación de Monte Carlo, eligiendo una sola muestra, es decir, en cada choque de un rayo con un objeto tan sólo se genera un rayo, formando un único camino. En el caso de fenómenos especulares, transmisivos la integral es determinista pues todos los rayos que inciden sobre el material.

$$\int_a^b f(x)dx \approx \frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)}{p(x_i)}$$

Para elegir la dirección del siguiente rayo se ha utilizado el muestreo uniforme del coseno, que, además, acaba simplificando los cálculos (elimina el término $|n \cdot wi|$).

$$L_o(\mathbf{x}, \omega_\mathbf{o}) \approx \sum_{i=1}^{N} \frac{2\pi L_i(\mathbf{x}, \omega_\mathbf{i}) f_r(\mathbf{x}, \omega_\mathbf{i}, \omega_\mathbf{o}) \cos\theta_i \sin\theta_i}{2\sin\theta_i \cos\theta_i}$$

Cada vez que un rayo intersecta con una figura se lanzan rayos de sombra para calcular la iluminación directa en el punto ($Li(x, wi)$), utilizando la siguiente fórmula (next-event estimation):

$$\sum_{i=1}^{n} \frac{p_i}{|\mathbf{c_i} - \mathbf{x}|^2} f_r\left(\mathbf{x}, \frac{\mathbf{c_i} - \mathbf{x}}{|\mathbf{c_i} - \mathbf{x}|}, \omega_\mathbf{o}\right) \left|\mathbf{n} \cdot \frac{\mathbf{c_i} - \mathbf{x}}{|\mathbf{c_i} - \mathbf{x}|}\right|$$

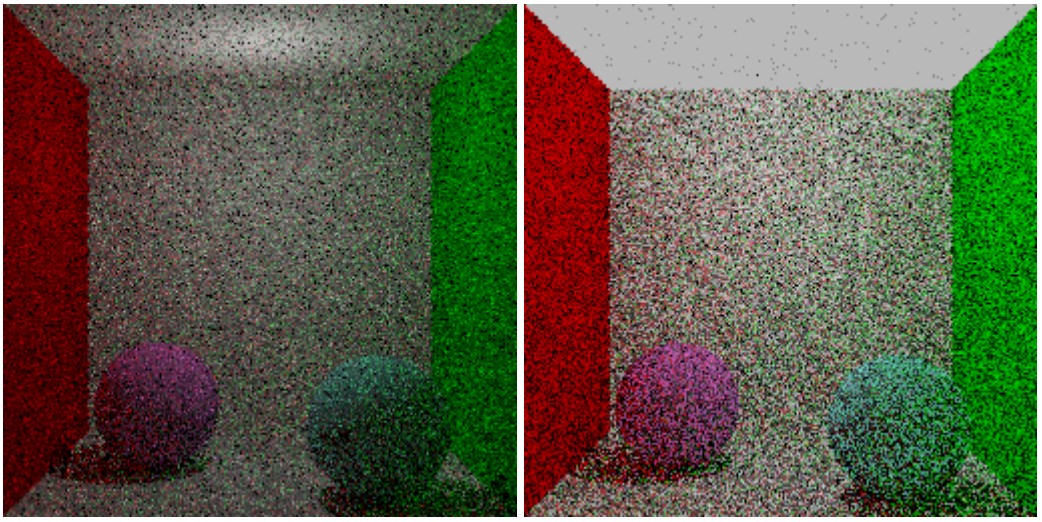
siendo n el número de fuentes de luz puntuales y c_i el punto en el que se ubica cada una de ellas.

Una vez calculada la luz directa, se multiplica por π y por la $fr(x, wi, wo)$ del material con el que ha intersectado y se guarda en una variable *scatter* que la función iterativa con la que se ha implementado usa para almacenar la pérdida de energía.

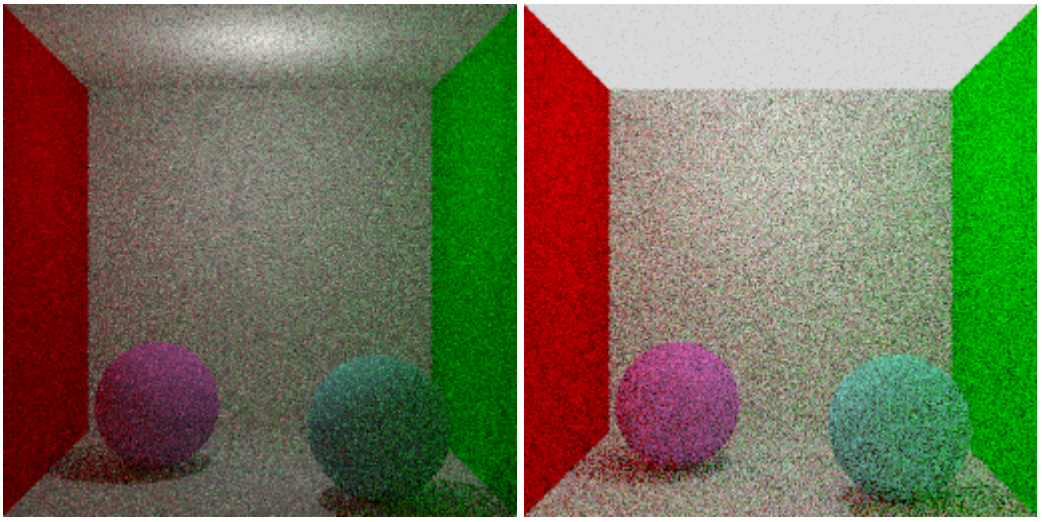
2. Ejemplos Cornell Box con diferentes muestras por píxel

Todas las imágenes son de 256x256 píxeles y a todas ellas se les ha aplicado tone mapping gamma 2,2.

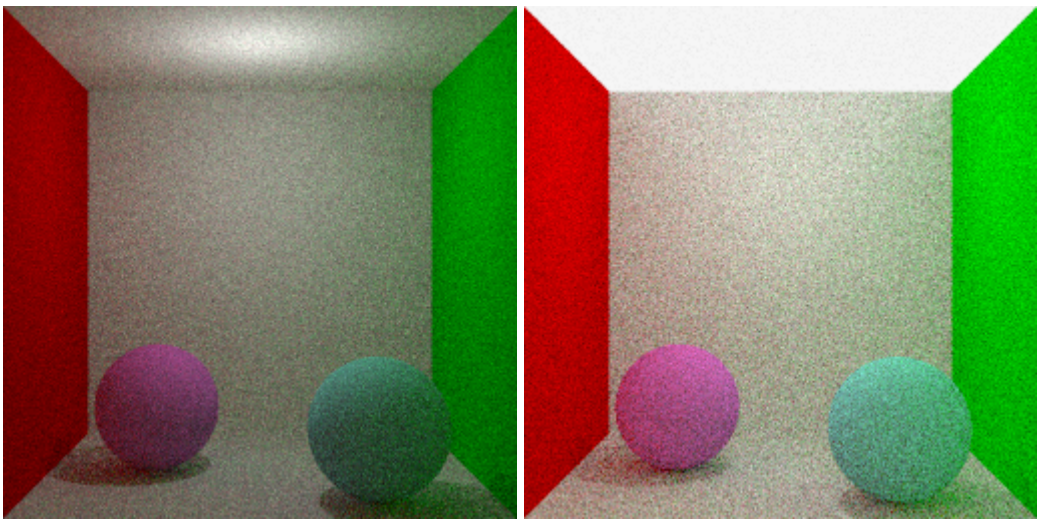
En primer lugar se va a comparar el efecto de las luces puntuales y las luces de área. La escena está formada por dos esferas difusas, en las imágenes de la izquierda hay una luz puntual y en las de la derecha el plano de arriba es una luz de área.



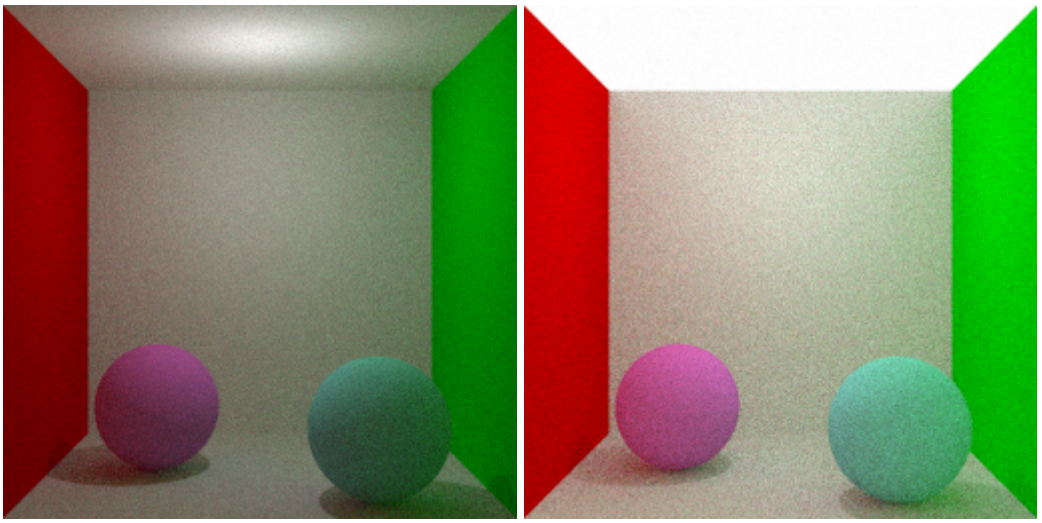
2 rayos por píxel



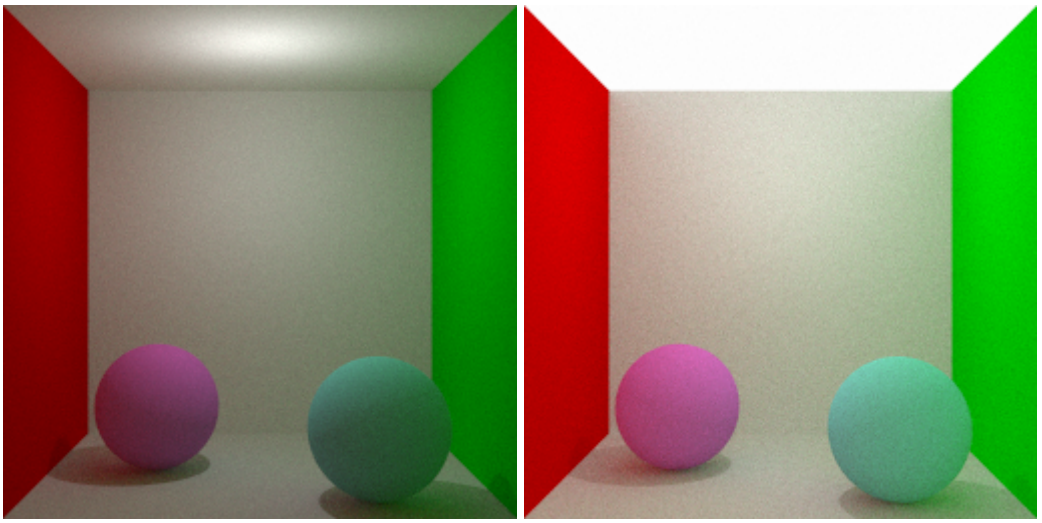
8 rayos por píxel



32 rayos por píxel



128 rayos por píxel

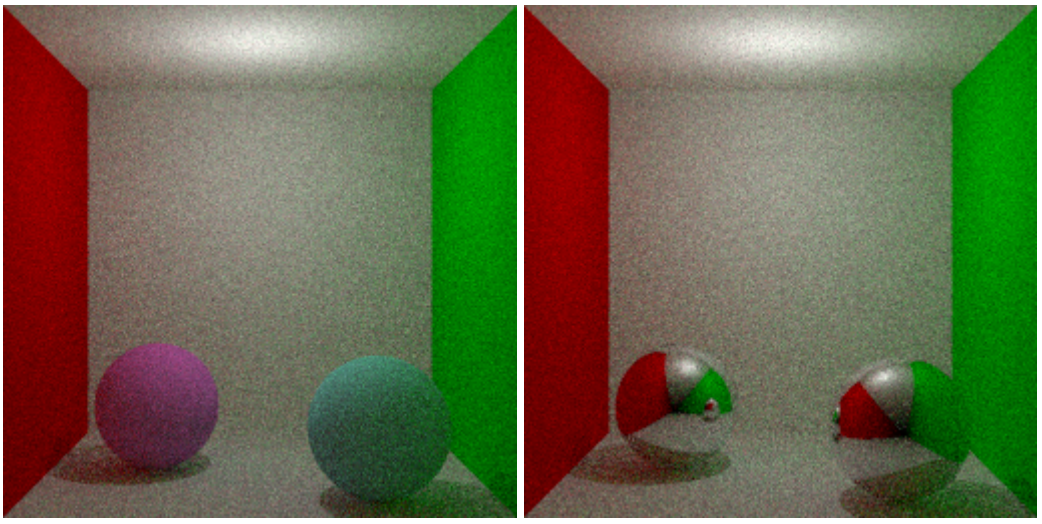


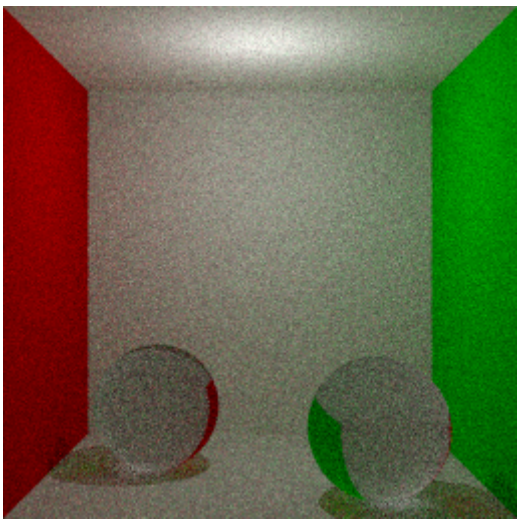
512 rayos por píxel

El número de rayos afecta significativamente a la convergencia de la imágen. Cómo se puede observar, las imágenes generadas con pocos rayos por píxel tienen demasiado ruido, mientras que conforme aumentamos los rayos se va reduciendo, hasta conseguir una convergencia “perfecta” con 512 rayos. Aumentarlos más podría incrementar mucho el tiempo de ejecución por la poca mejora que se obtendría.

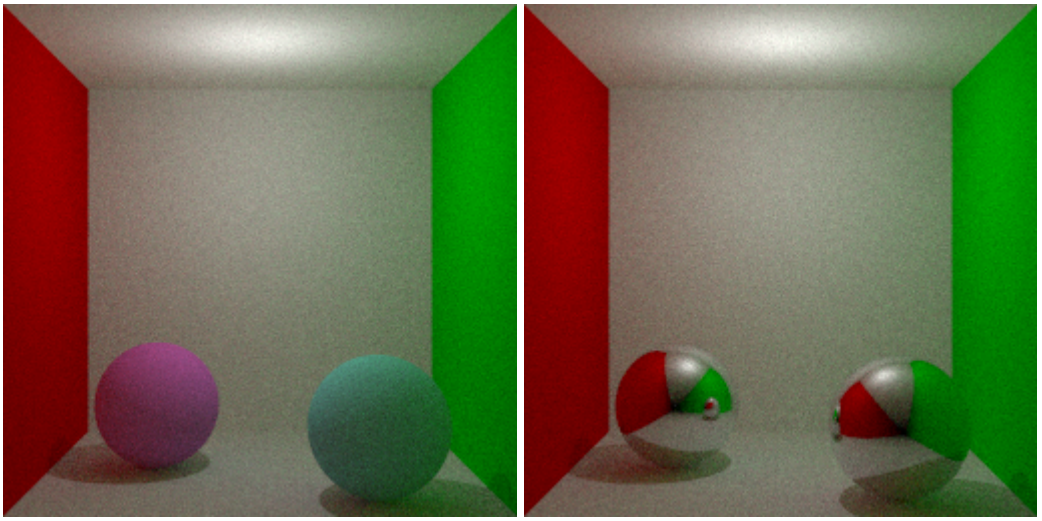
En cuanto al tipo de luz, se puede ver que las imágenes con una luz puntual convergen más rápidamente que las que tienen la luz de área. Esto se debe a que cuando un rayo intersecta con una luz de área ya no rebota más, por lo que pierde información de los que serían los siguientes rebotes. Además, si un rayo es absorbido antes de intersectar con la luz de área, no la va a tener en cuenta.

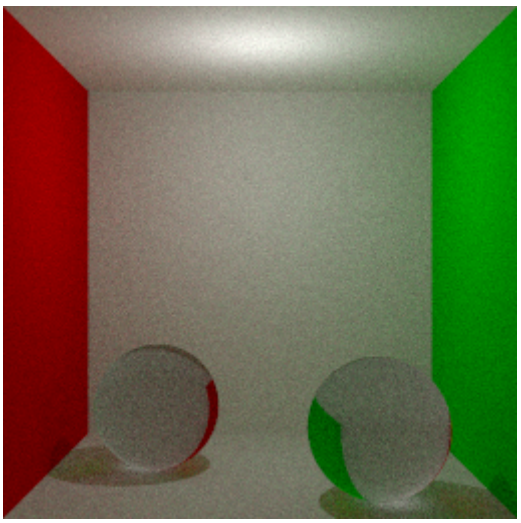
Se puede ver cómo el tiempo de ejecución del programa es mucho menor para las luces de área, por el primer motivo comentado anteriormente.





32 rayos por píxel

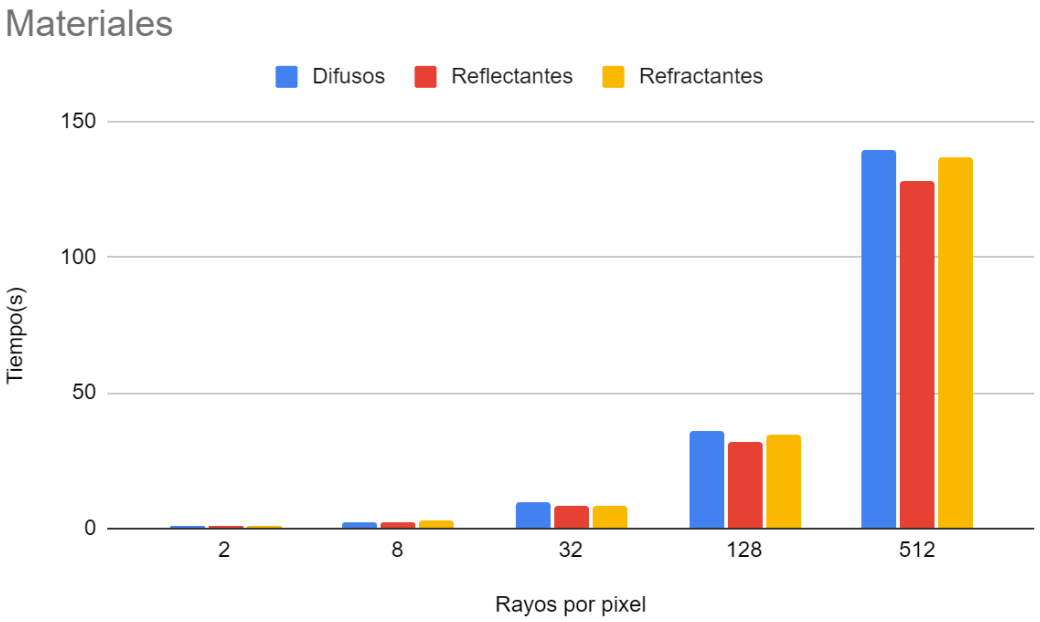




128 rayos por píxel

Los materiales reflectantes o refractarios convergen más lento que los difusos. Esto se debe a que cuando un rayo intersecta con ese material.

En cuanto al coste temporal dependiendo del tipo de material, se ha observado que es el mismo para todos ellos, ya que no importa el tipo de material de las figuras, si no la probabilidad de absorción que en este caso era muy parecida para todos los materiales.



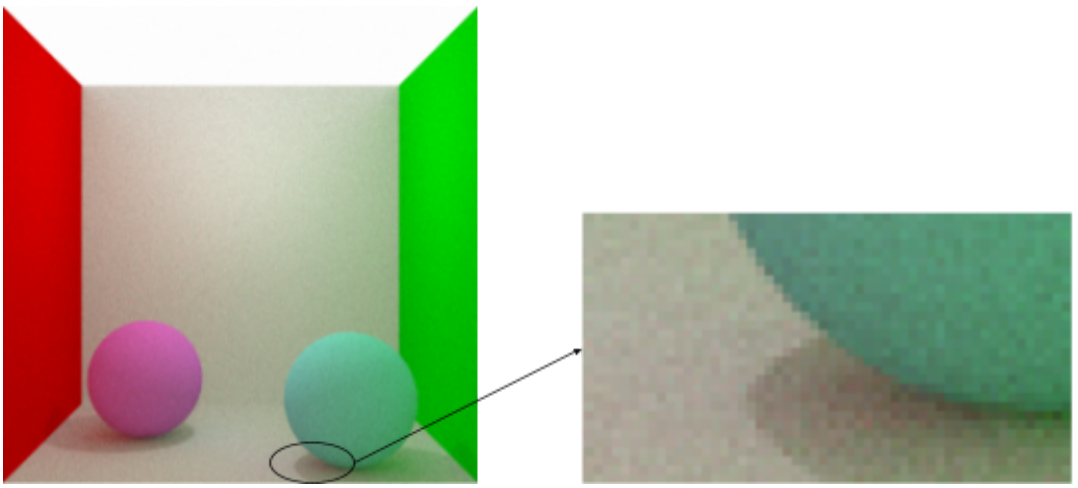
3. Efectos de iluminación global

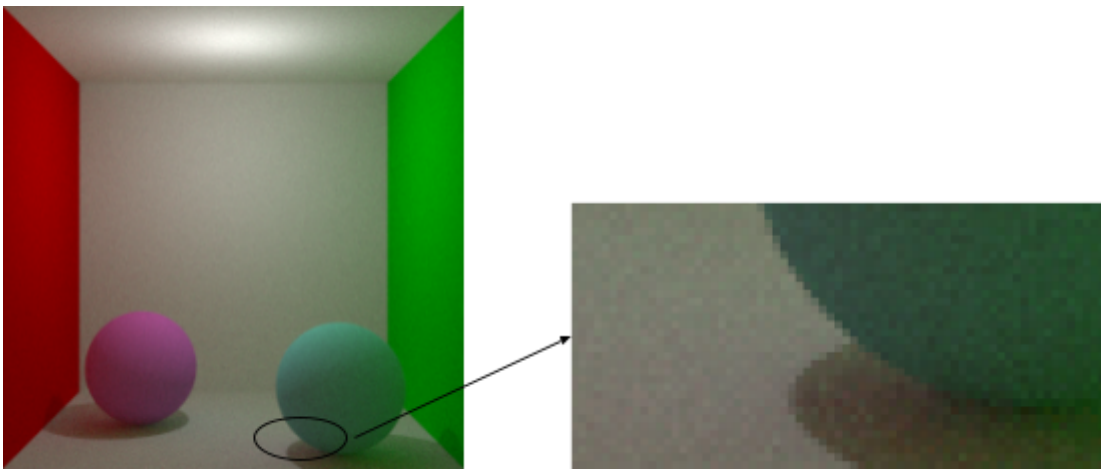
El path tracer sigue el camino de un rayo hasta que es absorbido. Cada uno de los rebotes del rayo contribuye a la iluminación del primer punto en el que ha impactado. Esto suaviza la diferencia entre las zonas sombreadas y sus adyacentes. Por ello permite obtener fácilmente sombras suaves y color bleeding, pero dificulta obtener sombras duras.

Para conseguir sombras más duras se deben usar luces puntuales. Otra opción sería limitar el número de rebotes, pero la imagen final perdería calidad.

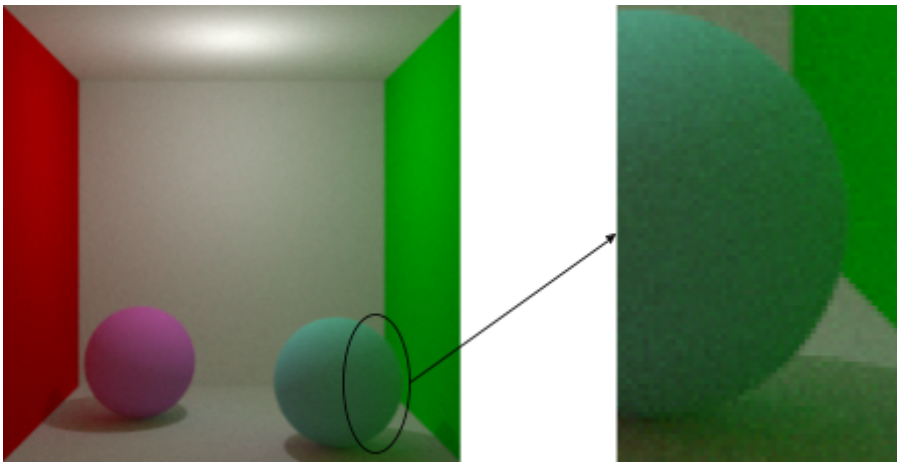
Las cáusticas son más difíciles de conseguir, para ello son necesarios materiales refractarios, con un índice lo más elevado posible.

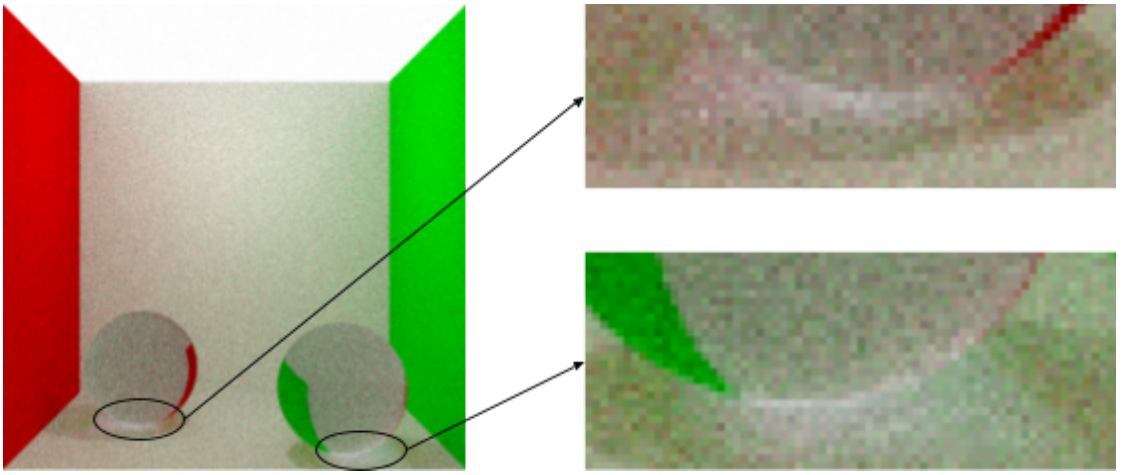
En las siguientes imágenes se pueden ver los dos tipos de sombra, aunque se observa que con una luz puntual la sombra se endurece, pero no llega a ser totalmente dura.





En las siguientes imágenes se puede apreciar el color bleeding en una esfera difusa y las causticas en una esfera que refracta. Para poder ver las cáusticas es necesario que haya luces de área.





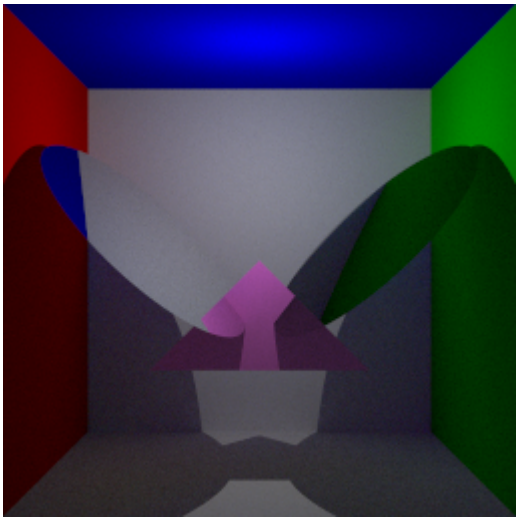
4. Extensiones

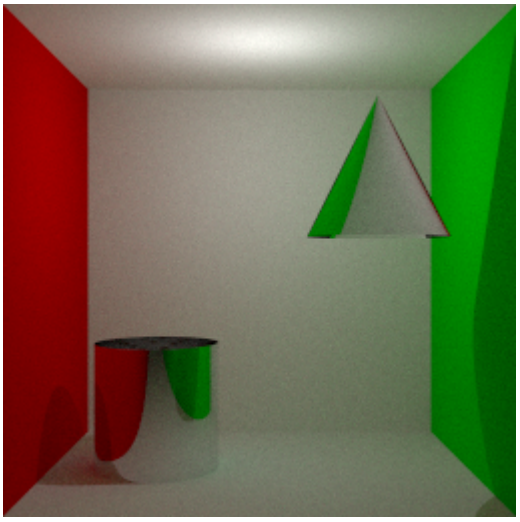
4.1 Otras geometrías

Además de los planos y esferas (geometrías obligatorias), se han incluido triángulos, discos, discos perforados, cilindros y conos.

Para ello se ha implementado la función de intersección de cada nueva geometría con un rayo, utilizando la ecuación implícita de las geometrías, además de otras clases requeridas para la implementación de las figuras: obtener una normal en un punto específico de la superficie.

En las siguientes imágenes se muestran las geometrías implementadas.





4.2 Paralelización

Se ha paralelizado la ejecución del algoritmo en varios hilos. El número de hilos se puede modificar desde la declaración de la escena.

Una vez el programa lee cuantos hilos va a utilizar divide los píxeles de la imagen en nº hilos^2 cuadrados. Todos estos cuadrados están a la espera de que un hilo los seleccione para procesar esa parte de la imagen. El acceso a ellos está protegido mediante exclusión mutua por un monitor para que dos hilos no puedan acceder al mismo.

Una vez un hilo ha terminado de procesar todos los píxeles del cuadrado que se le ha asignado rellena la parte de la imagen correspondiente al cuadrado que ha calculado y le pide al monitor que le asigne otro, así hasta finalizar toda la imagen.

Se ha realizado una prueba con 128 rayos por píxel, el programa con un sólo hilo ha tardado 28.568 segundos, mientras que usando 16 hilos le ha costado 7.742, por lo que el tiempo se ha reducido en un 73% aproximadamente.

4.3 Ficheros .objx

Para la carga de escenas y distintas configuraciones de los renderizadores, se ha creado un nuevo formato de escena 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.

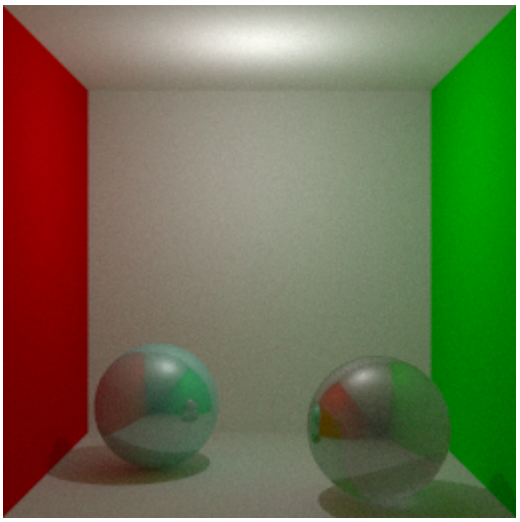
Una descripción más detallada de este formato de archivo para generar nuevas escenas se puede encontrar en OBJX.md.

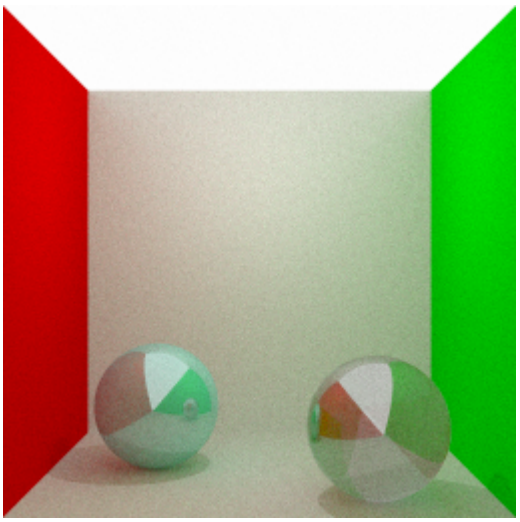
En la carpeta scenes hay algunos ejemplos sencillos de escena.

5. Organización

En cuanto a la organización que se ha seguido, se ha utilizado GitHub para tener un registro del código y todas sus versiones, ambos integrantes han ido desarrollando el código a la vez, cuando uno de ellos tenía tiempo y juntos en las sesiones de prácticas. Los avances realizados fuera de las sesiones de prácticas se han subido a GitHub y se ha avisado al compañero, para ponerle al día y explicarle las nuevas funcionalidades o correcciones que se han desarrollado.

6. Renders finales





Ambas figuras se han generado con 512 rayos por píxel.

La escena tiene una esfera, a la izquierda, difusa que refleja y una que refleja y refracta a la derecha.