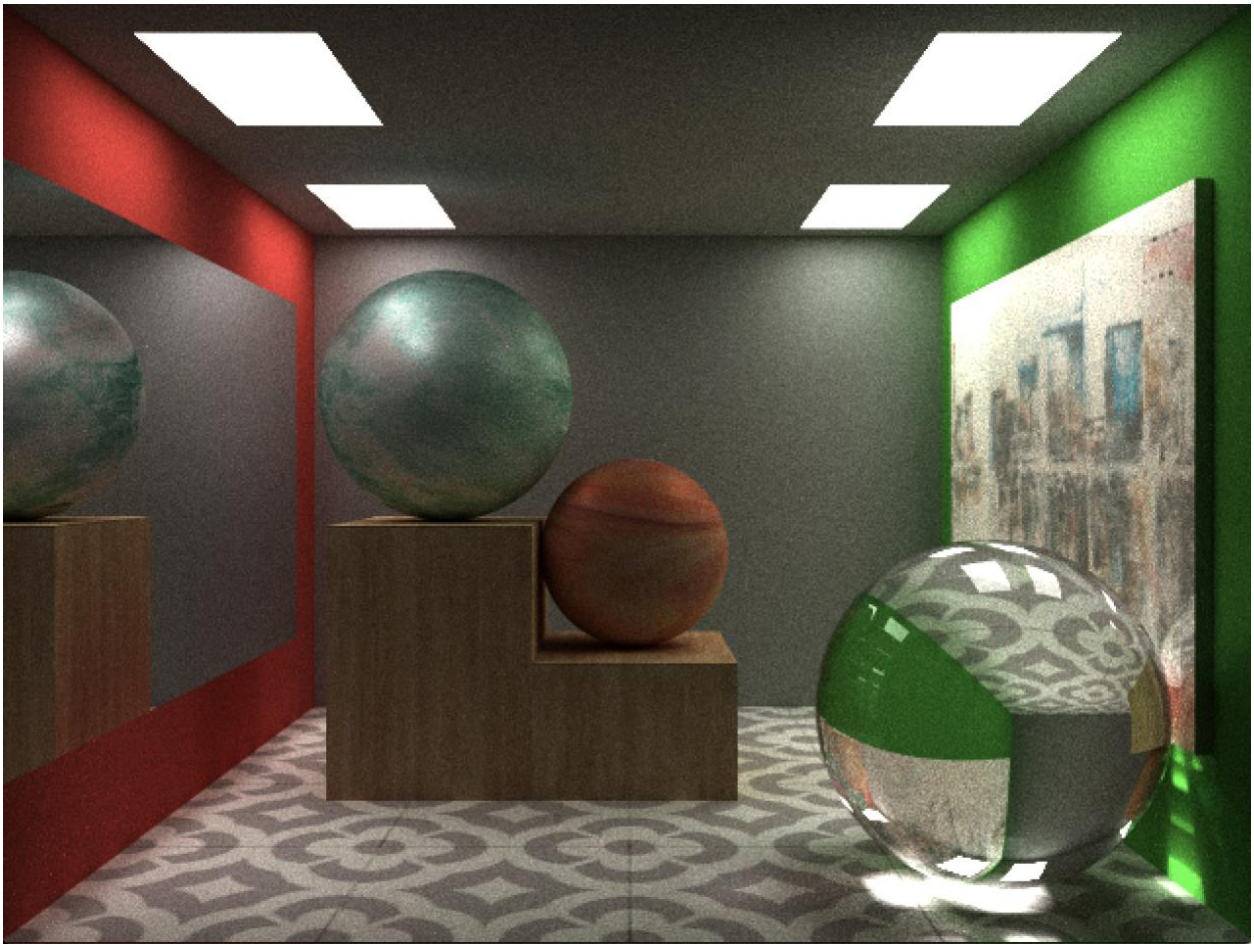


Path Tracing

Alejandro Gutiérrez Bolea 735089

Daniel Cay Delgado 741066

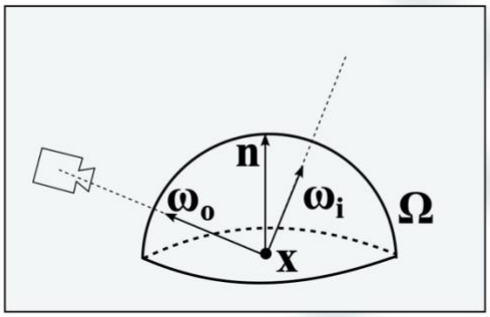
Informática Gráfica 2019-2020

1. Introducción

Esta memoria explica la implementación del algoritmo de Path tracing que se ha explicado en la asignatura, haciendo uso del lenguaje C++. Se han ido añadiendo las funcionalidades requeridas en las distintas entregas intermedias, logrando así un Path Tracer que genera imágenes HDR, y que gracias a la implementación de un Tone mapper son convertidas a LDR.

2. Implementación

2.1. Ecuación de Render



$$L_o(\mathbf{x}, \omega_\mathbf{o}) = L_e(\mathbf{x}, \omega_\mathbf{o}) + \int_\Omega L_i(\mathbf{x}, \omega_\mathbf{i}) f_r(\mathbf{x}, \omega_\mathbf{i}, \omega_\mathbf{o}) |\mathbf{n} \cdot \omega_\mathbf{i}| d\omega_\mathbf{i}$$

La ecuación de render tiene por objetivo devolver la cantidad total de luz emitida desde un punto *x* en la dirección *wo* (*Lo*). El término *Le* indica la cantidad de luz que emite el punto *x*. En esta implementación se ha ignorado ya que se considera que un objeto que refleja luz no puede emitir y viceversa. *Li* representa la luz que llega a *x* desde la dirección *wi*. La función de distribución de reflectancia bidireccional (*fr*) expresa la luz reflejada en dirección *wo* proveniente de la dirección *wi*. Determina las propiedades del material. Por último, el término del coseno *|n*wi|*, que atenúa la luz en función del ángulo entre la normal en el punto *x* y la dirección en la que llega la luz a *x* (*wi*). Cuando el ángulo entre la normal y *wi* es 0, incide la máxima luz en *x*, por lo que el término del coseno vale 1.

Esta ecuación se integra en toda la hemiesfera para obtener la luz total que llega a *x* desde todas las direcciones posibles. Como esto es imposible de realizar debido a su alto coste computacional,

se usa la técnica probabilista de Monte Carlo [3], que consiste en lanzar *N* caminos por cada píxel. Cuantos más se lancen, mayor coste computacional, pero mejores resultados.

2.2. Algoritmo

1. Se lanzan *N* caminos desde la cámara a través de cada píxel con una dirección aleatoria.

2. Con un camino que va en una dirección pueden ocurrir 3 fenómenos:
   2.1. No intersecta con ninguna figura de la escena, por lo que el color devuelto por ese camino es el color por defecto de fondo, que es el negro *RGB(0,0,0)*.
   2.2. Intersecta con una luz de área, por lo que camino devuelve la intensidad asociada a dicha luz (valor *RGB* entre 0 e infinito).
   2.3. Intersecta a una figura que refleja la luz. A partir del *PUNTO 3* se explica qué ocurre en esta situación.

3. Puesto que se está implementando iluminación global, para la situación *2.3* se debe calcular la luz directa e indirecta del punto intersectado *P*. La suma de toda esa luz será el color *RGB* que devolverá el camino.

4. **Luz indirecta**: Se calcula recursivamente volviendo al *PUNTO 2*, pero usando como inicio del camino la dirección en la que salga rebotado el camino al intersectar con el punto *P*, que dependerá de las propiedades del material intersectado (muestreo por importancia):

   4.1.

   -Materiales con BRDF lambertiana o de Phong: Los lambertianos son los que poseen *kd* y los de phong los que poseen *ks* y kd. La nueva dirección es aleatoria dentro de la hemiesfera creada entorno a la normal del punto intersectado (muestreo por coseno). Para calcular esta dirección se realiza un cambio de base a coordenadas locales (normal=(0,1,0)), y se calculan los ángulos theta y phi. Para ello se generan dos números aleatorios (random1, random2) comprendidos entre 0 y 1, y a continuación se calcula:

$$theta = acos\sqrt{1 - random1}$$
$$phi = 2 * \pi * \text{random2}$$

   La dirección aleatoria es igual a [1]:

$$\sin(theta) * \cos(phi), \cos(theta), \sin(theta) * \sin(phi)$$

Por último, se vuelve a coordenadas globales multiplicando la dirección por la matriz de cambio de base.

-Materiales especulares perfectos (delta BRDF): Poseen *ksp* y la nueva dirección del camino no se selecciona de forma aleatoria, viene determinada por la ley de reflexión [2].

-Materiales refractores perfectos (delta BTDF): Poseen *kr* y la nueva dirección del camino viene determinada por la ley de Snell [2].

4.2. Ruleta rusa [4]: Un material puede poseer valor distinto a 0 en más de uno de los coeficientes comentados (*kd, ks, ksp y kr*). Por ello, para determinar según qué evento va a fijarse la nueva dirección del camino se utiliza la llamada ruleta rusa. Esta va a establecer una relación entre el valor de los coeficientes y las probabilidades de que se produzca un evento u otro. Los coeficientes son tuplas *RGB*, por lo que su probabilidad en la ruleta rusa se ha determinado por el mayor valor de las 3 componentes *RGB* para cada coeficiente. Además, siempre existe un 10% de probabilidad de matar el camino actual (de que haya absorción), asegurando así el final de la recursión en algún momento para todos los caminos. Puesto que se debe asegurar ese 10% de absorción, si la suma de las probabilidades entre todos los eventos es mayor al 90% se deberán normalizar para que su suma nunca supere dicho valor.

4.3. Por último, el valor devuelto por la recursividad (*Li*) se debe multiplicar por la función de reflectancia bidireccional (*fr*) y por el término del coseno, además de dividirlo por la *pdf* que surge del uso de monte carlo y la ruleta rusa.

Además, si la superficie del material es lamertiana o phong, debido al muestreo por coseno aparecen unos términos en el cálculo de la luz indirecta que se han simplificado a mano para evitar errores de redondeo y ganar limpieza de código:

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

Ejemplo del cálculo de luz indirecta con material Lambertiano:

Li=valor devuelto por la llamada recursiva al *PUNTO 2*

$$LuzIndirecta = \frac{Li*(Kd/\pi)*\cos\theta*sin\theta}{pdf*(1/2\pi)*2*\cos\theta*sin\theta} = \frac{Li*kd}{pdf}$$

5. **Luz directa**: Solo se computa para aquellos objetos que poseen kd o ks, y si existen luces puntuales en la escena (*next event estimation*). En este algoritmo, los caminos hacen de rayos de sombra, pero como las luces puntuales no se pueden intersectar, si no se realiza el cálculo de la luz directa la escena quedaría totalmente negra ya que ningún camino terminaría por intersección con una fuente de luz.

   Por cada fuente de luz puntual se calcula:

   $$L_i(\mathbf{x}, \omega_\mathbf{i}) f_r(\mathbf{x}, \omega_\mathbf{i}, \omega_\mathbf{o}) |\mathbf{n} \cdot \omega_\mathbf{i}|$$

   Se suma el valor obtenido por cada una ya que la luz es aditiva.

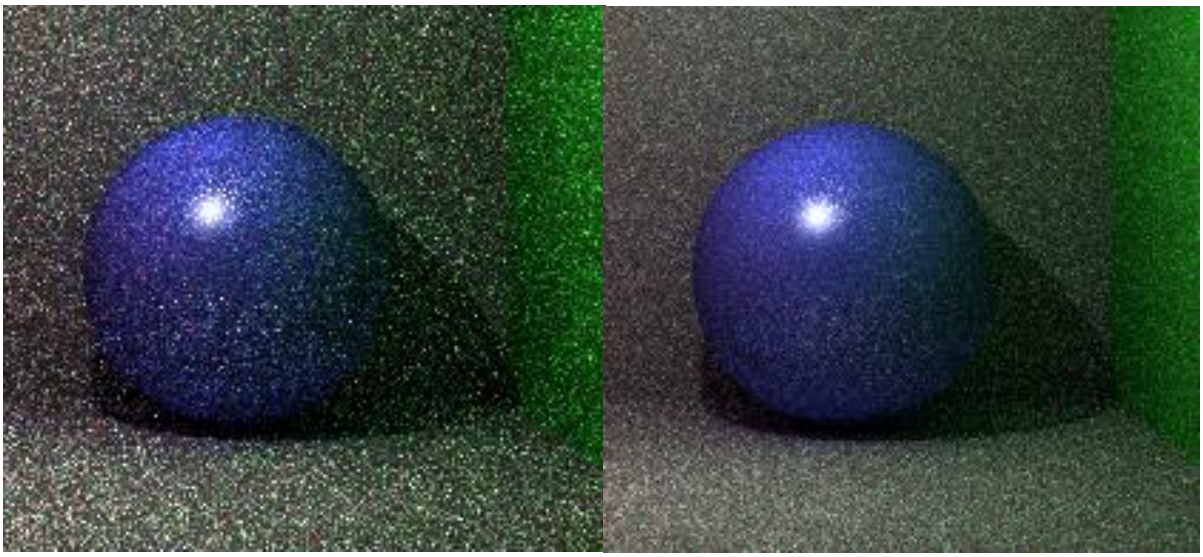
6. El número de caminos lanzados por cada píxel es especificado por el usuario. El color de un píxel vendrá determinado por la suma del color devuelto por los N caminos lanzados a través de ese píxel, dividido por N.

7. Sobre el valor resultante en cada píxel se ha aplicado *clamping* con el fin de que finalmente esté comprendido en el rango [0,255]. De esta manera se pasa de la imagen HDR generada a la LDR que es la que se puede visualizar.

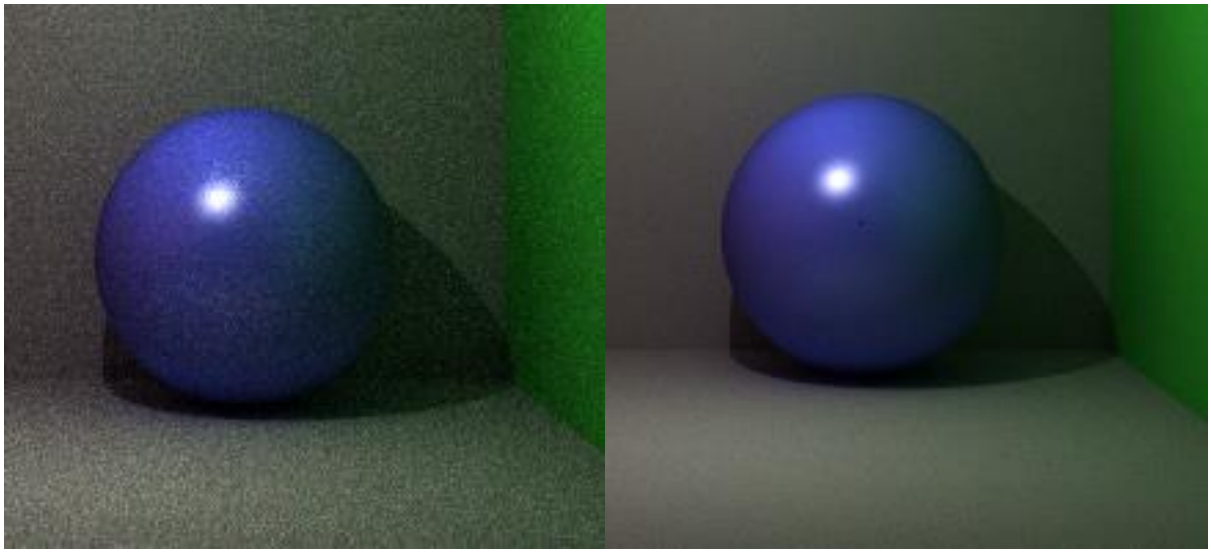
3. Convergencia

3.1. Caminos por píxel

Se puede observar como conforme se aumenta el número de rayos lanzados, disminuye el ruido del resultado final y aumenta el tiempo de ejecución. Se ha observado experimentalmente que la reducción de ruido no crece de forma lineal. Conforme más ppp se lanzan, más diferencia se necesita con respecto a los ppp lanzados anteriormente para notar una reducción del ruido.



(a) *(b)*



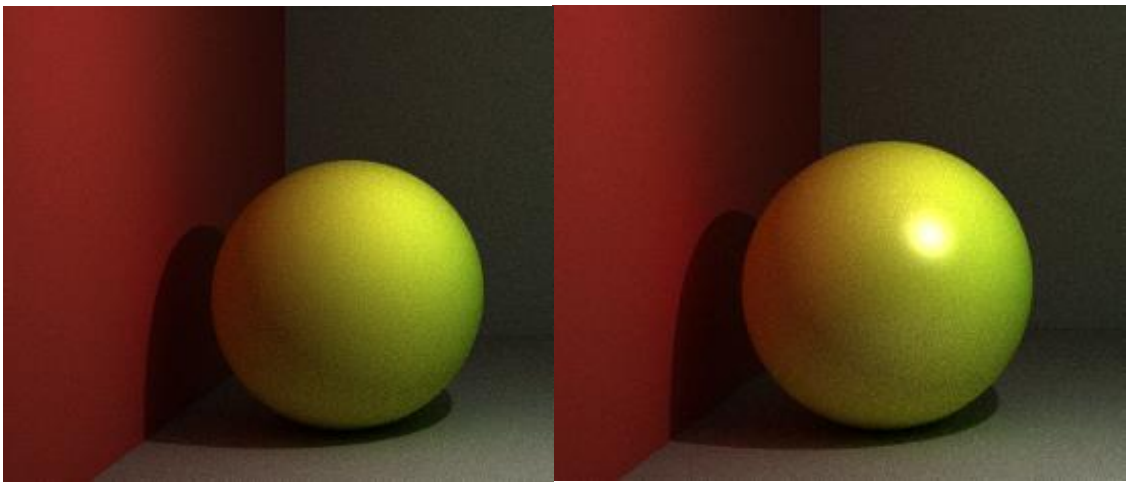
(c) *(d)*

Figura 1: Caja de Cornell con una luz puntual. Las paredes son difusas y la esfera es Phong con kd(0.26,0.3,0.78), ks(0.1,0.1,0.1) y alfa=30. (a) 4ppp/1 min. (b) 8ppp/2 min. (c) 64ppp/20 min. (d) 1000ppp/311 min.

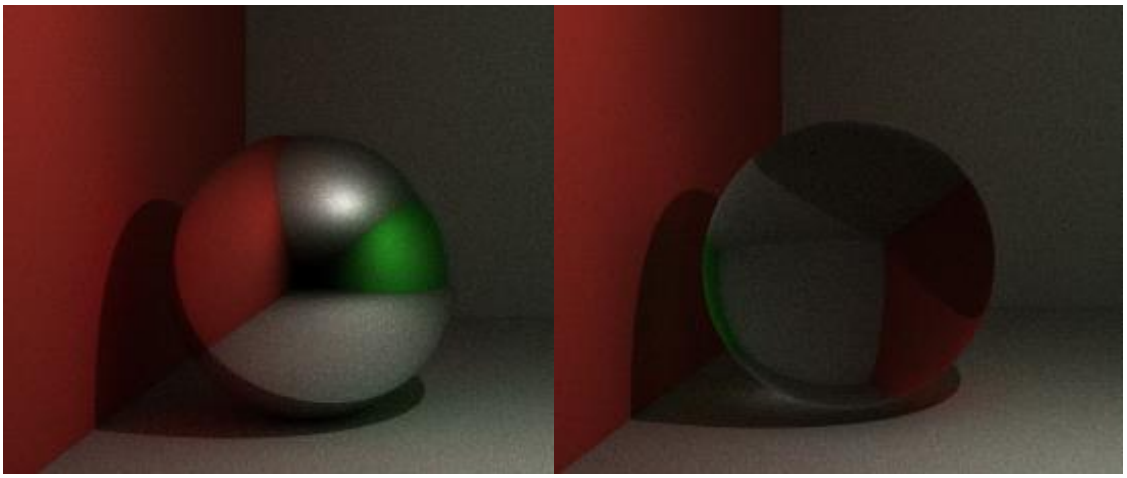
Se observa que entre la figura 1(a) y 1(b) hay una reducción de ruido considerable habiendo solo 4ppp de diferencia. Entre la 1(c) y 1(d) se observa reducción de ruido pero hay que tener en cuenta la gran diferencia de ppp entre una y otra.

3.2. Materiales

El algoritmo converge antes con los materiales difusos y de Phong. Esto es debido a la *next event estimation*, la cual no es calculada cuando el rayo golpea un material especular o refractor perfecto, siendo entonces más probable que el camino en cuestión termine sin aportar nada de luz al resultado final.



(a) *(b)*



(c) *(d)*

Figura 2: 200ppp. Caja de Cornell con una luz puntual. Las paredes son difusas y la esfera es (a) kd(0.8,0.8,0.2)/58 min. (b) kd(0.8,0.8,0.8),ks(0.1,0.1,0.1),alfa=30 /60 min. (c) ksp(0.9,0.9,0.9)/59 min (d) kr(0.9,0.9,0.9),ior=2/ 57 min.

La diferencia entre las figuras 2(a), 2(b), 2(c) y 2(d) tanto en coste temporal como de convergencia es despreciable. Esto se debe a que al ser una única esfera que está dentro de una escena más grande, su peso en la composición no es el suficiente como para que se perciba una variación en el coste temporal. También influye que solo se hayan usado 200ppp en vez de un valor más grande (*Más ppp, más cálculos de next event estimation*).

3.3. Fuentes de luz

El algoritmo converge antes con las fuentes de luz puntuales, pero su coste computacional es mayor. Esto se debe nuevamente a que usando luces de área no hay *next event estimation*, por lo que es más probable que un camino termine sin aportar nada de luz al resultado final. Con luces de área, si el camino termina en absorción o por no intersectar nada, no aportará nada de luz, si termina chocando con una luz de área lo hará. En cambio, con las luces puntuales, al ir calculando la luz directa en cada intersección del camino, es mucho más fácil que un camino aporte información al resultado final.

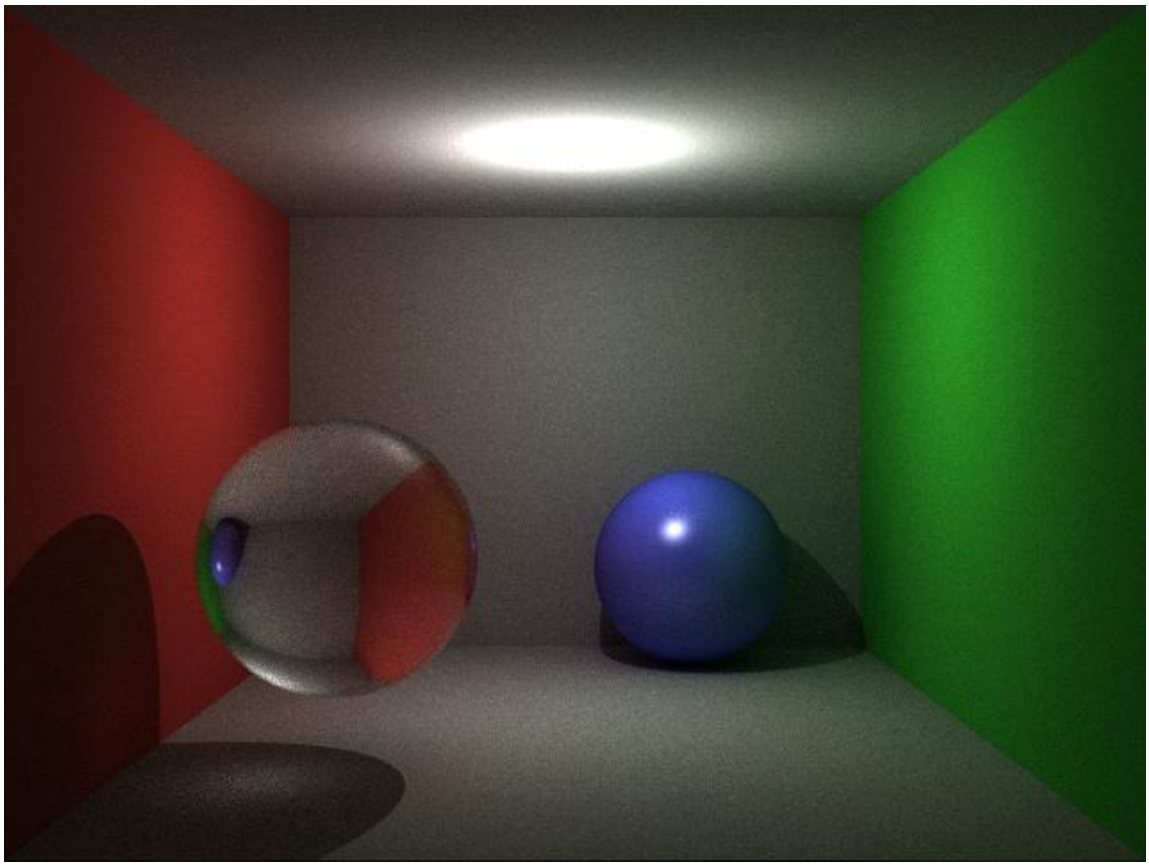


Figura 3: 200 ppp/65 min. Caja de Cornell con una luz puntual. La esfera de la izquierda es Fresnel con ior=2 y la esfera de la derecha es Phong con kd(0.26,0.3,0.78), ks(0.1,0.1,0.1) y alfa=30. Todas las paredes son difusas. La pared izquierda con kd(0.8,0.2,0.2), la derecha con kd(0.2,0.8,0.2), y el techo, suelo y pared del fondo con kd(0.6,0.6,0.6).

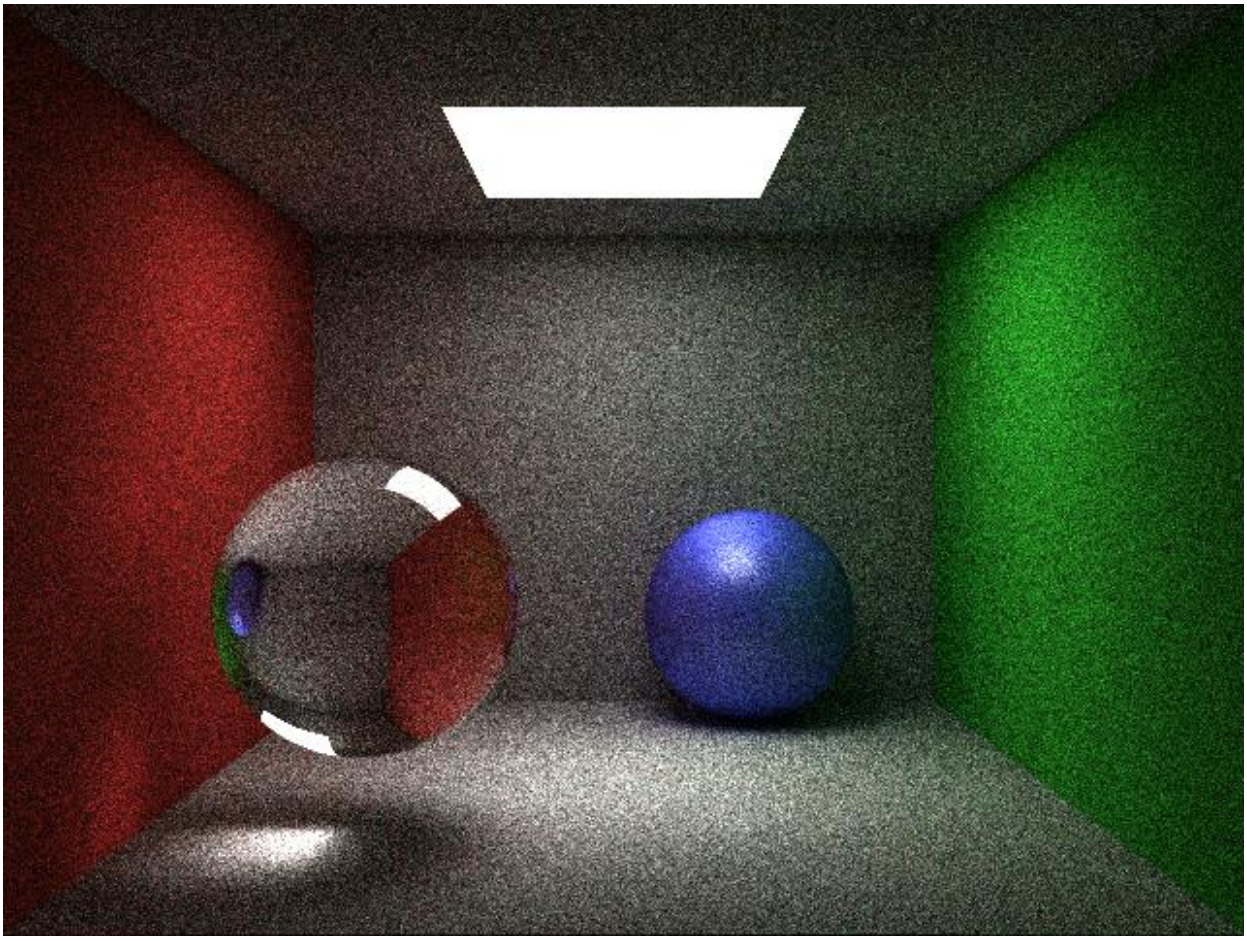


Figura 4: 200ppp/60 min. Caja de Cornell con una luz de área. La esfera de la izquierda es Fresnel con ior=2 y la esfera de la derecha es Phong con kd(0.26,0.3,0.78), ks(0.1,0.1,0.1) y alfa=30. Todas las paredes son difusas. La pared izquierda con kd(0.8,0.2,0.2), la derecha con kd(0.2,0.8,0.2), y el techo, suelo y pared del fondo con kd(0.6,0.6,0.6).

De acuerdo a lo comentado anteriormente, se puede observar cómo la figura 3 presenta menos ruido que la figura 4 pero ha tardado 5 minutos más.

4. Iluminación global

4.1. Sombras

Se pueden distinguir dos tipos de sombras, las *hard* y *soft shadows*. Las primeras son producidas por las luces puntuales ya que al computar la *next event estimation,* hay mucha diferencia de luz entre un punto al que le llega luz directa y otro punto al que no le llega. En las segundas, al no realizar el cálculo de la luz directa, la sombra está computada exclusivamente por la luz indirecta, lo que provoca una sombra mucho más suave y degradada.

Figura 5: Soft shadow

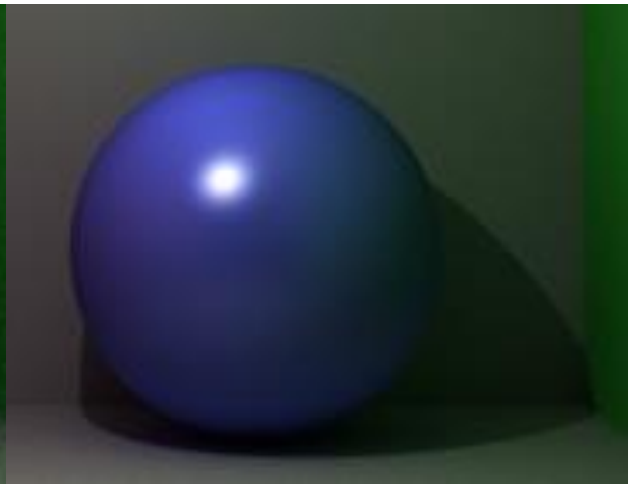


Figura 6: Hard shadow

4.2. Color bleeding

Es el fenómeno por el cual los objetos o superficies son coloreados por el reflejo de la luz de color de las superficies cercanas. Para producirlo es necesario que el material del objeto tenga una componente difusa para poder emitir en el color correspondiente.

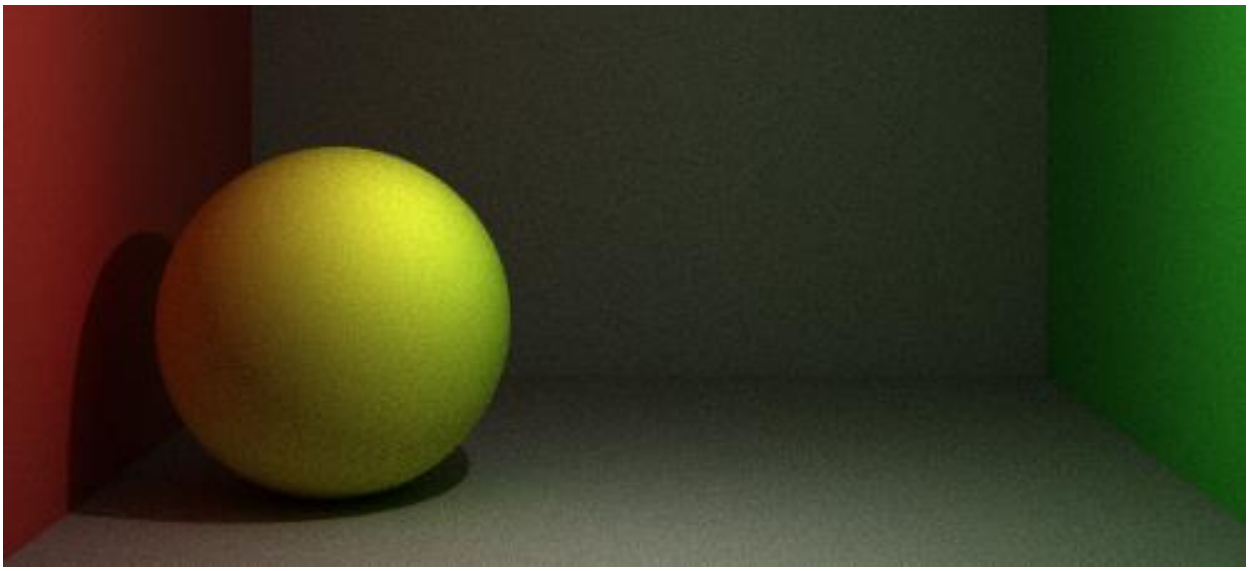


Figura 7: Caja de Cornell con una luz puntual. La esfera es difusa con kd(0.8,0.8,0.2). Todas las paredes son difusas. La pared izquierda con kd(0.8,0.2,0.2), la derecha con kd(0.2,0.8,0.2), y el techo, suelo y pared del fondo con kd(0.6,0.6,0.6).

En la figura 7 se puede observar cómo la esfera de color amarillo posee una tonalidad rojiza por la izquierda y una verdosa por la derecha, ambas producidas por el color de las paredes.

4.3. Cáusticas

Se pueden formar debido a dos situaciones:

(a) Usando luces de área, cuando un rebote indirecto atraviesa un material delta y choca con la luz de área.
(b) Con un rebote indirecto que atraviesa un material delta y choca con una superficie a la que le llega mucha luz procedente de una luz puntual.

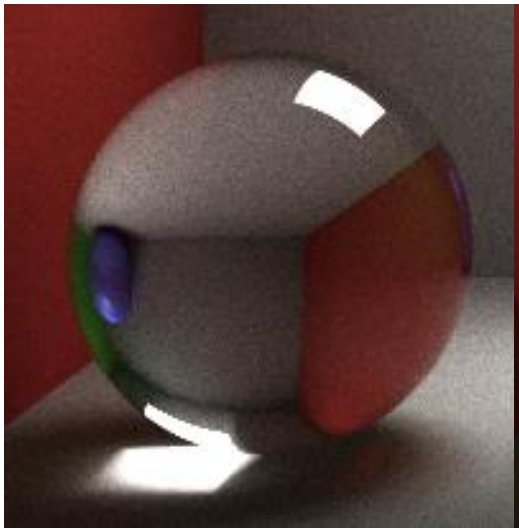


Figura 8: (a)

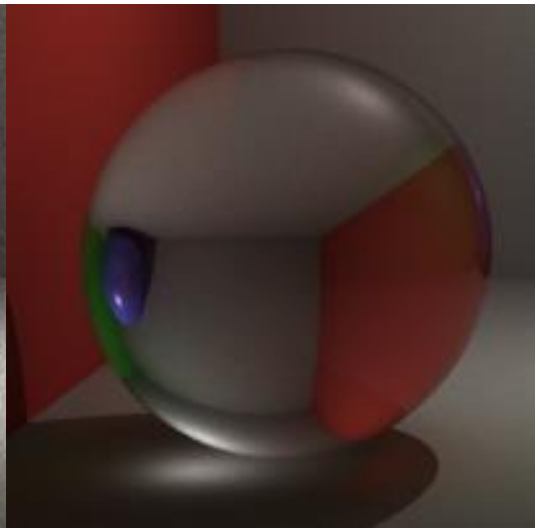


Figura 9: (b)

Es el efecto más complejo ya que es difícil reconstruir el camino de una cáustica (se necesita "mucha suerte"). Por ello, para lograr dicho efecto hay que lanzar un número alto de caminos por píxel.

5. Extensiones

5.1. Paralelización

Con el fin de reducir el tiempo de ejecución, se ha dividido el trabajo entre el número máximo de threads que permite el ordenador en el que se está ejecutando el programa con el fin de lograr el mínimo tiempo de ejecución posible. Inicialmente, se calculaba el color de cada píxel uno tras otro. Si la resolución de la imagen resultante es de 400x400, debía calcularse uno tras otro el color de 160.000 pixeles. Con esta mejora, la pantalla de píxeles se reparte entre distintos threads. Para el caso anterior, con 4 threads, cada uno de ellos calcularía una porción de 400x100, es decir, 40.000 píxeles, ocurriendo los cálculos de cada thread de forma simultánea, reduciendo así en gran medida el coste temporal.

Para desarrollar esta mejora se han usado los conocimientos adquiridos en la asignatura de *Programación de Sistemas Concurrentes y Distribuidos*.

5.2. Texturas

El proceso de añadir una textura a una superficie consiste en saber qué píxel de la textura se corresponde con qué punto de la figura de la escena sobre la que se quiere colocar dicha textura. Se han añadido sobre 2 tipos de figuras:

Plano finito:

Se debe calcular el porcentaje hacia la derecha (eje x) y hacia arriba (eje y) en el que está el punto de intersección *P* dentro del plano finito. Después, obtener el color de la textura que se encuentra en la posición determinada por dichos porcentajes y colocarlo en P.

Figura 10: Textura madera

Figura 11: Textura madera sobre plano finito

Esfera:

Se ha usado la técnica denominada UV mapping [5]. La idea es la misma que para los planos finitos, calcular el porcentaje (eje x, eje y) dentro de la textura, es decir, el punto de la textura del cual se quiere obtener el color, pero en este caso obtener estos porcentajes es un poco más complejo:

$$u = 0.5 + \frac{\arctan2(d_z, d_x)}{2\pi},$$
$$v = 0.5 - \frac{\arcsin(d_y)}{\pi}.$$

u=eje x, v=eje y, normal del punto P de la esfera intersectado =(dx,dy,dz)

Figura 12: Textura mármol

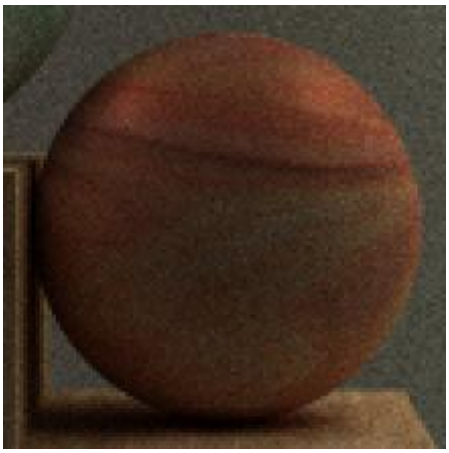


Figura 13: Textura mármol sobre esfera

5.3. Fresnel

Los materiales transparentes como el vidrio o el agua son tanto refractivos como reflectantes. Cuánta luz reflejan y cuánta refractan viene determinado por el ángulo de incidencia de la luz. Con las ecuaciones de Fresnel [2] se obtienen los valores del *ksp* y *kr* dado un *ior*. De esta manera se pueden simular materiales reales ya que no se da un valor arbitrario a estos coeficientes.

Las fórmulas aplicadas son las siguientes:

$$F_{R\parallel} = \left(\frac{\eta_2 \cos\theta_1 - \eta_1 \cos\theta_2}{\eta_2 \cos\theta_1 + \eta_1 \cos\theta_2} \right)^2,$$

$$F_{R\perp} = \left(\frac{\eta_1 \cos\theta_2 - \eta_2 \cos\theta_1}{\eta_1 \cos\theta_2 + \eta_2 \cos\theta_1} \right)^2.$$

$\eta_{1\,y}\,\eta_2$ son los índices de refracción de los 2 medios

cos θ_1 y cos θ_2 son los ángulos de reflexión y refracción respectivamente

$$F_R = \frac{1}{2}(F_{R\parallel} + F_{R\perp}).$$

$$F_T = 1 - F_R.$$

ksp = F_R

kr = F_T

Se debe comentar el caso de la reflexión interna total en el que ksp=1 y kr=0, que se da cuando el seno del ángulo de refracción excede la unidad.

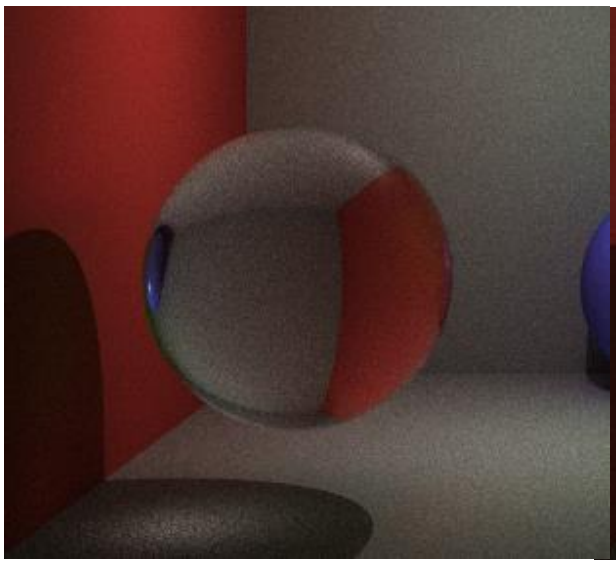


Figura 14: ior=1.5 (Vidrio)

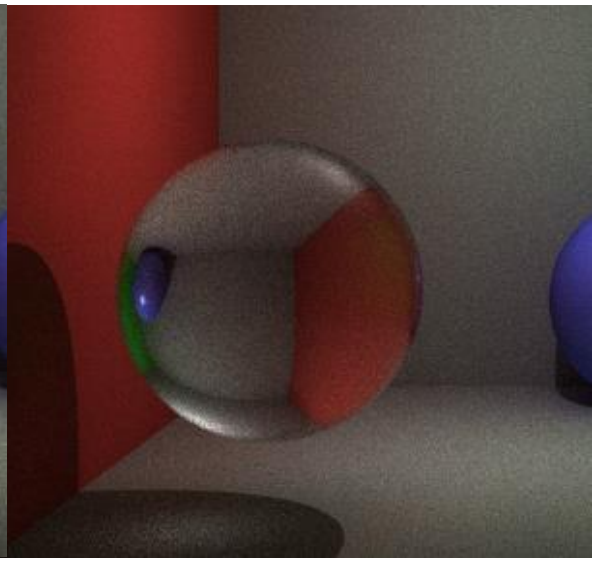


Figura 15: ior=2 (Cristal)

6. Dificultades

Las dificultades se han centrado en la luz indirecta, puesto que su elaboración implica entender muy bien algunos conceptos matemáticos, como el cálculo de un rayo en dirección aleatoria para materiales lambertianos y phong, y la gestión de la ruleta rusa.

7. Metodología

Se ha seguido el modelo de desarrollo en espiral, modelo iterativo en el que en cada una de sus iteraciones se debe: determinar objetivos, analizar riesgos, desarrollar software y verificarlo.

8. Imágenes

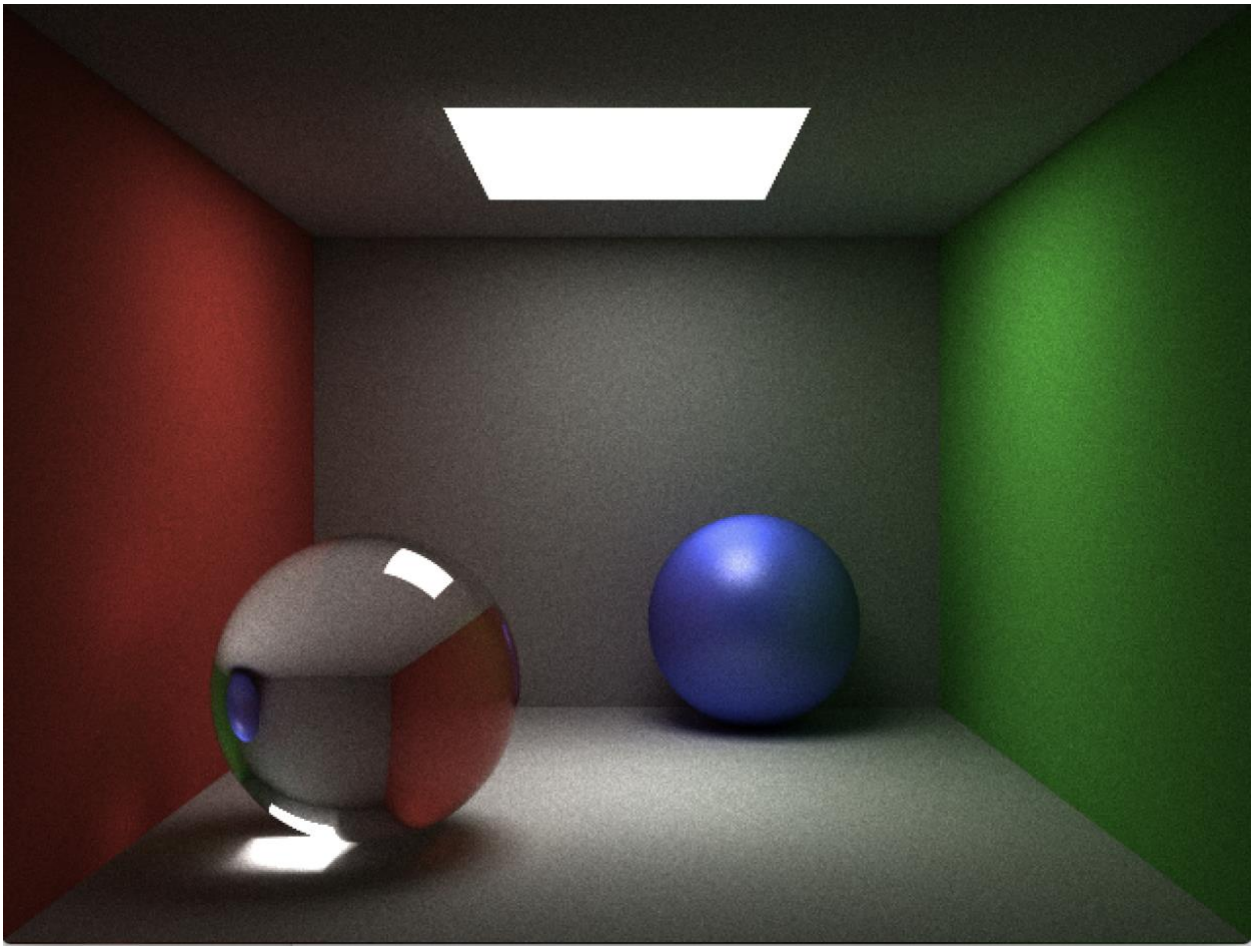


Figura 16: 640x480/4000ppp/19 horas y media. Caja de Cornell con una luz de área. La esfera de la izquierda es Fresnel con ior=2 y la de la derecha es Phong con kd(0.26,0.3,0.78), ks(0.1,0.1,0.1) y alfa=30. Todas las paredes son difusas. La pared izquierda con kd(0.8,0.2,0.2), la derecha con kd(0.2,0.8,0.2), y el techo, suelo y pared del fondo con kd(0.6,0.6,0.6).

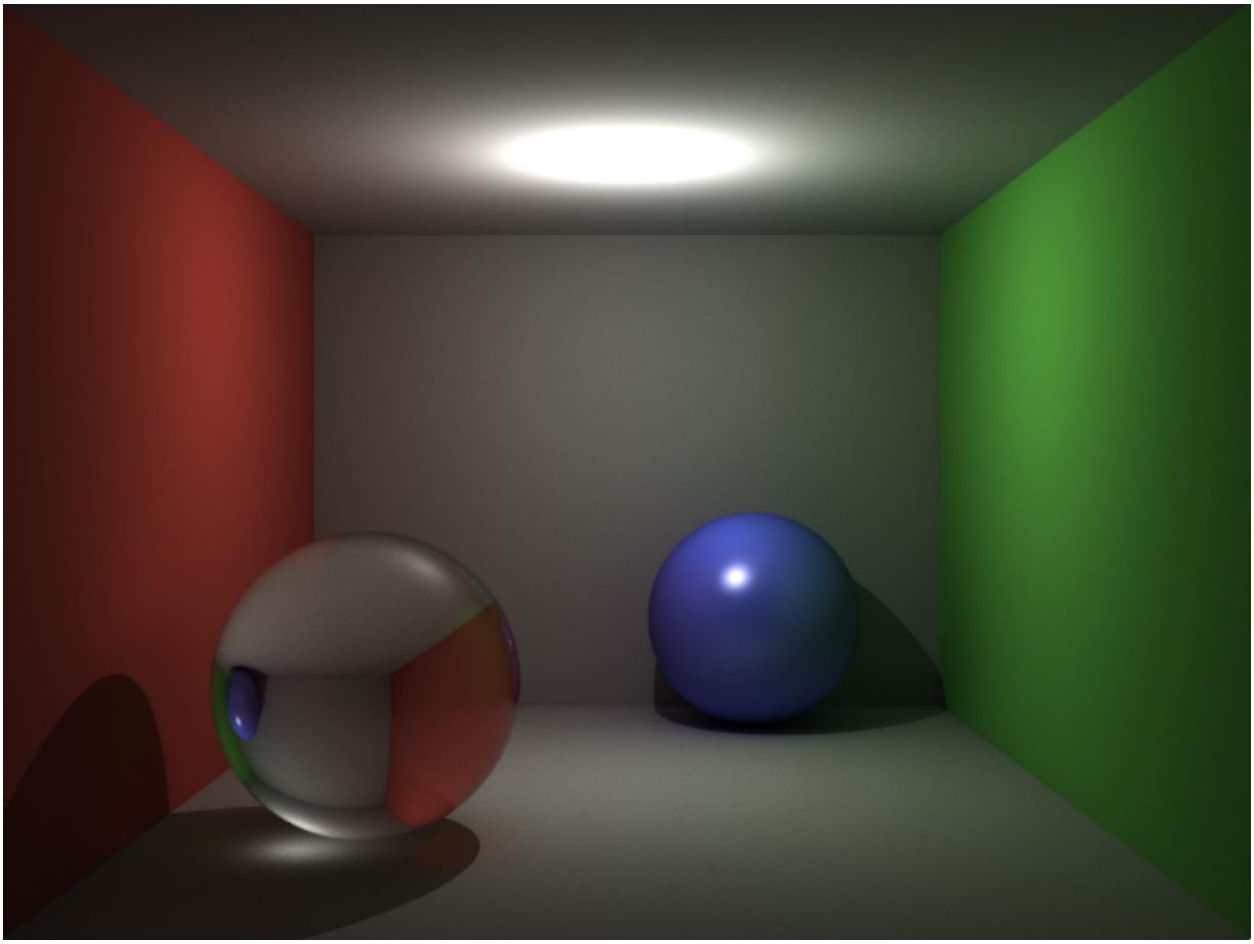


Figura 17: 640x480/4000ppp/21 horas. Caja de Cornell con una luz puntual. La esfera de la izquierda es Fresnel con ior=2 y la de la derecha es Phong con kd(0.26,0.3,0.78), ks(0.1,0.1,0.1) y alfa=30. Todas las paredes son difusas. La pared izquierda con kd(0.8,0.2,0.2), la derecha con kd(0.2,0.8,0.2), y el techo, suelo y pared del fondo con kd(0.6,0.6,0.6).

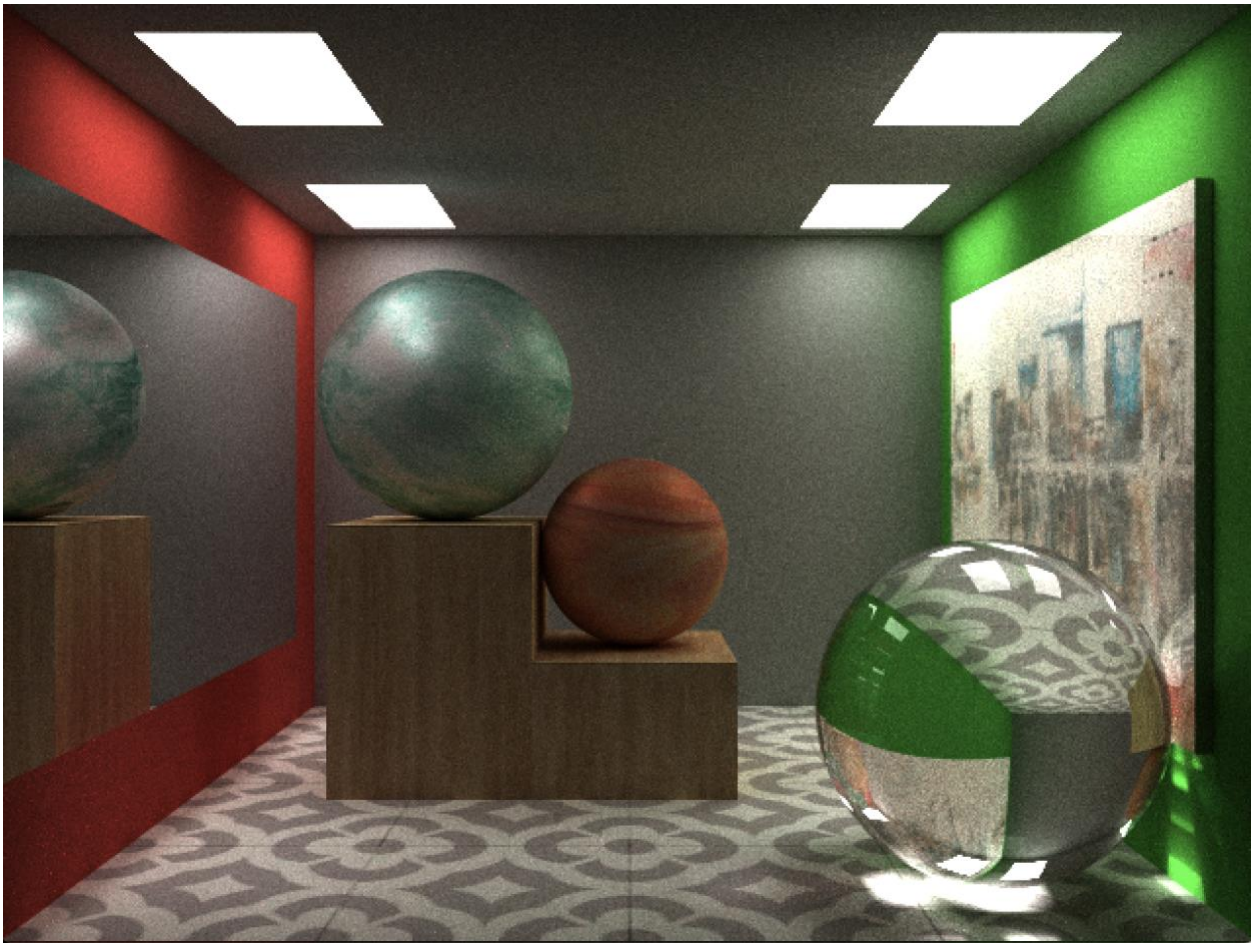


Figura 18: 640x480/4000ppp/22 horas. La esfera de la izquierda es de Phong con kd=textura, ks(0.1,0.1,0.1) y alfa=30, la del centro es difusa con kd=textura y la de la derecha es Fresnel con ior=2. Los planos que conforman las paredes son difusos. El de la izquierda con kd(0.8,0.2,0.2), el de la derecha con kd(0.2,0.8,0.2), el techo y la pared del fondo con kd(0.6,0.6,0.6) y el suelo con kd=textura. En la pared de la izquierda hay un plano finito con ksp(0.9,0.9,0.9) que actúa de espejo y en la pared derecha hay otro plano finito difuso con kd=textura que imita a un cuadro. En las 4 esquinas del techo hay un plano finito que se comporta como luz de área.

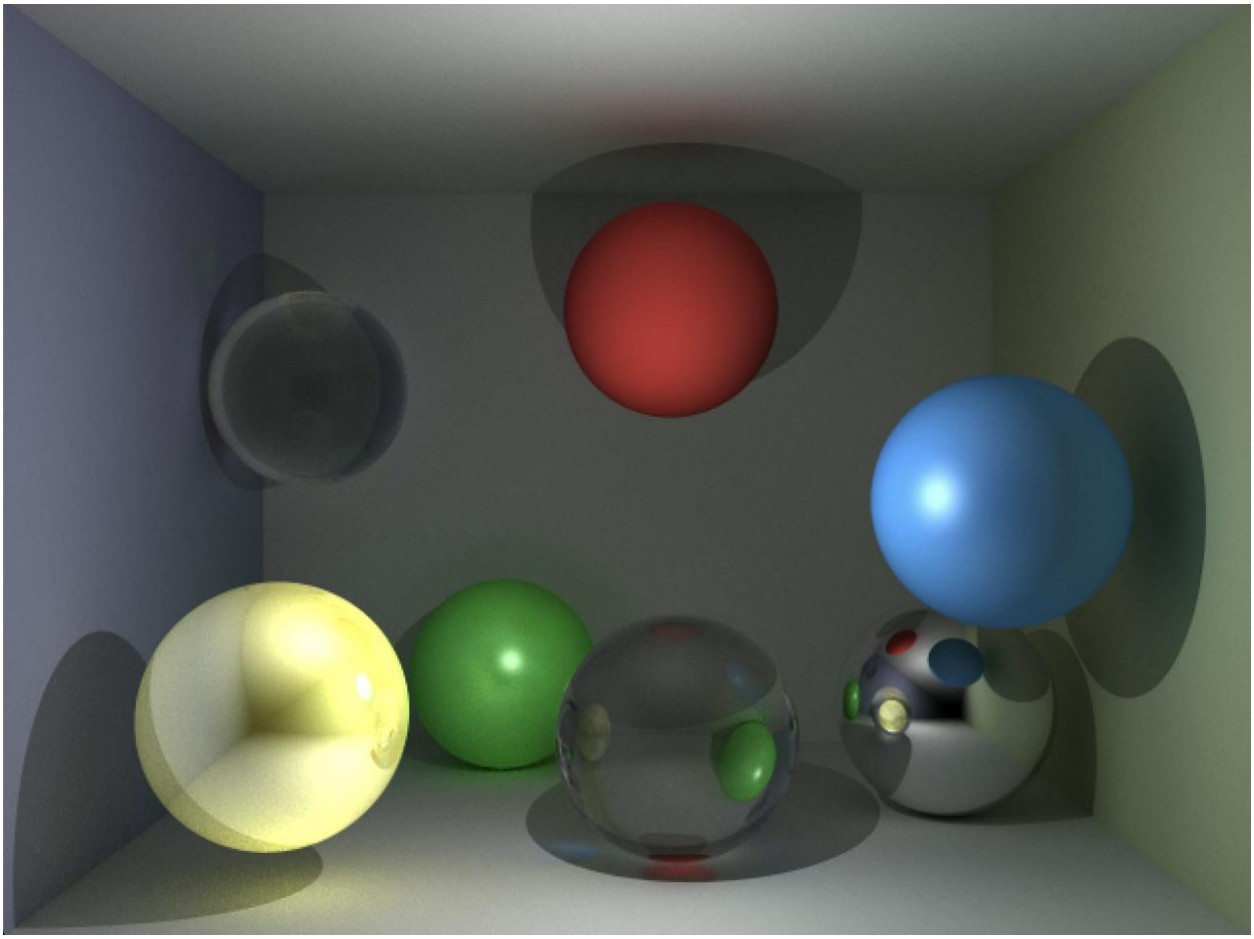


Figura 19: 640x480/4000ppp/35 horas.

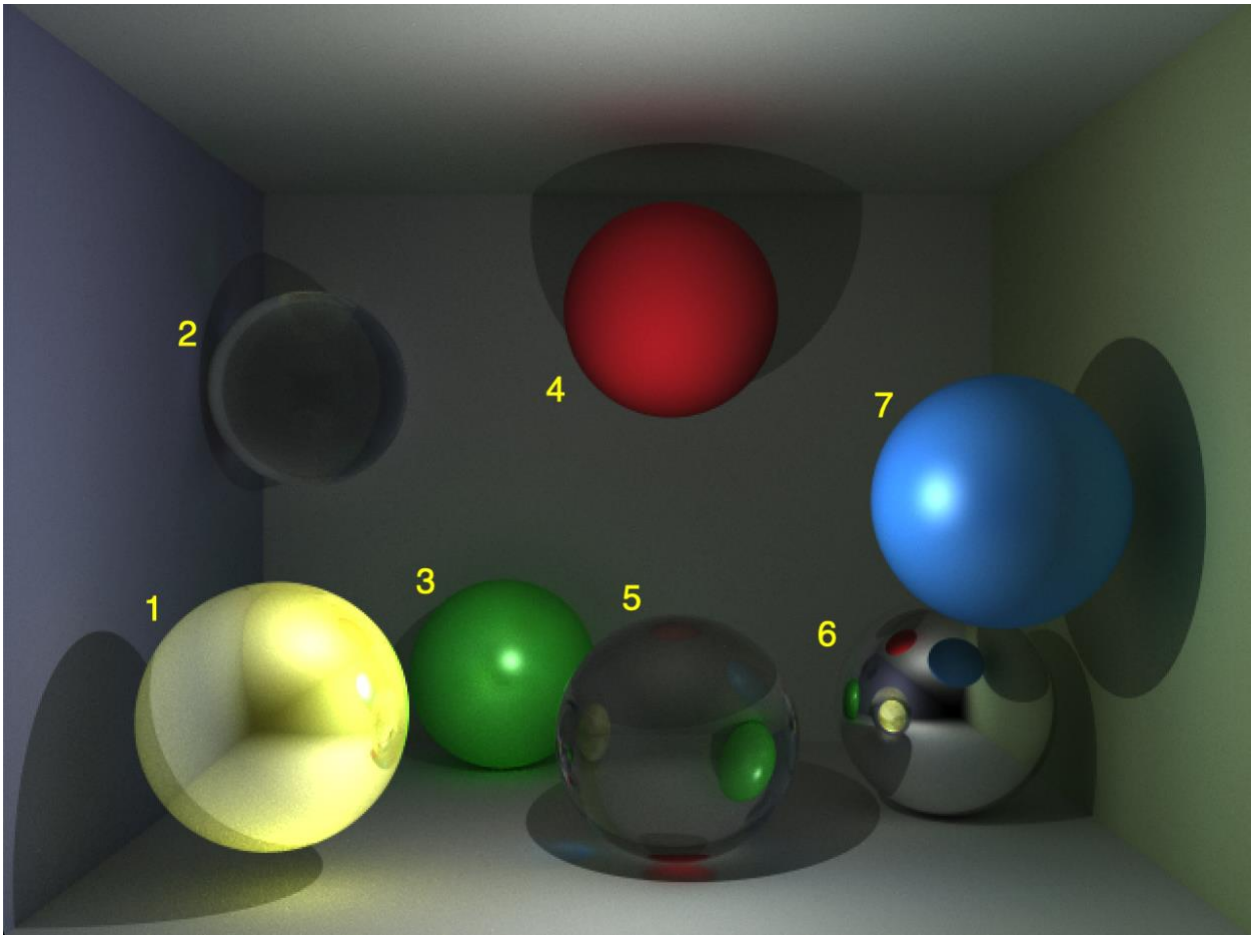


Figura 20: Explicación de la figura 20. (1) kd(0.78,0.75,0.24), ksp(0.9,0.9,0.9). (2) Fresnel con ior=1.5. (3) kd(0.2,0.8,0.2), ks(0.2,0.2,0.2), alfa=30. (4) kd(0.8,0.2,0.2). (5) Fresnel con ior=2. (6) ksp(0.9,0.9,0.9). (7) kd(0.23,0.51,0.8), ks(0.1,0.1,0.1), alfa=10.

9. Conclusión

Se ha sido capaz de desarrollar un *path tracer* al cual se le ha ido añadido de forma progresiva mayor complejidad con el fin de crear imágenes más artísticas, realistas, y en definitiva, más bonitas al ojo humano. Además, se han incorporado una serie de extensiones centradas en facilitar el trabajo (paralelización) y dar aún más autenticidad a las imágenes elaboradas (Fresnel y texturas).

Se es consciente de que se puede añadir mucha más complejidad al algoritmo, pero se considera que los conocimientos adquiridos van más allá de los básicos.

El cuello de botella a la hora de realizar el trabajo ha estado en los conocimientos matemáticos necesarios para realizar los retos que se iban presentando. Pese a los abundantes problemas que han surgido durante el proceso se ha disfrutado mucho ya que el visualizar los progresos siendo estos tan llamativos ha servido de motivación para seguir trabajando con el fin de lograr mejores resultados.

10. Bibliografía

[1] Mathworld: https://mathworld.wolfram.com/SphericalCoordinates.html
[2] Scratchapixel: https://www.scratchapixel.com/lessons/3d-basic-rendering/introduction-to-shading/reflection-refraction-fresnel
[3] Asignatura informática gráfica (diap 20 en adelante): https://moodle.unizar.es/add/pluginfile.php/2303601/course/section/356530/07-montecarlo-handout.pdf
[4] Asignatura informática gráfica (diap 52 en adelante): https://moodle.unizar.es/add/pluginfile.php/2303601/course/section/356530/07-montecarlo-handout.pdf
[5] Wikipedia: https://en.wikipedia.org/wiki/UV_mapping