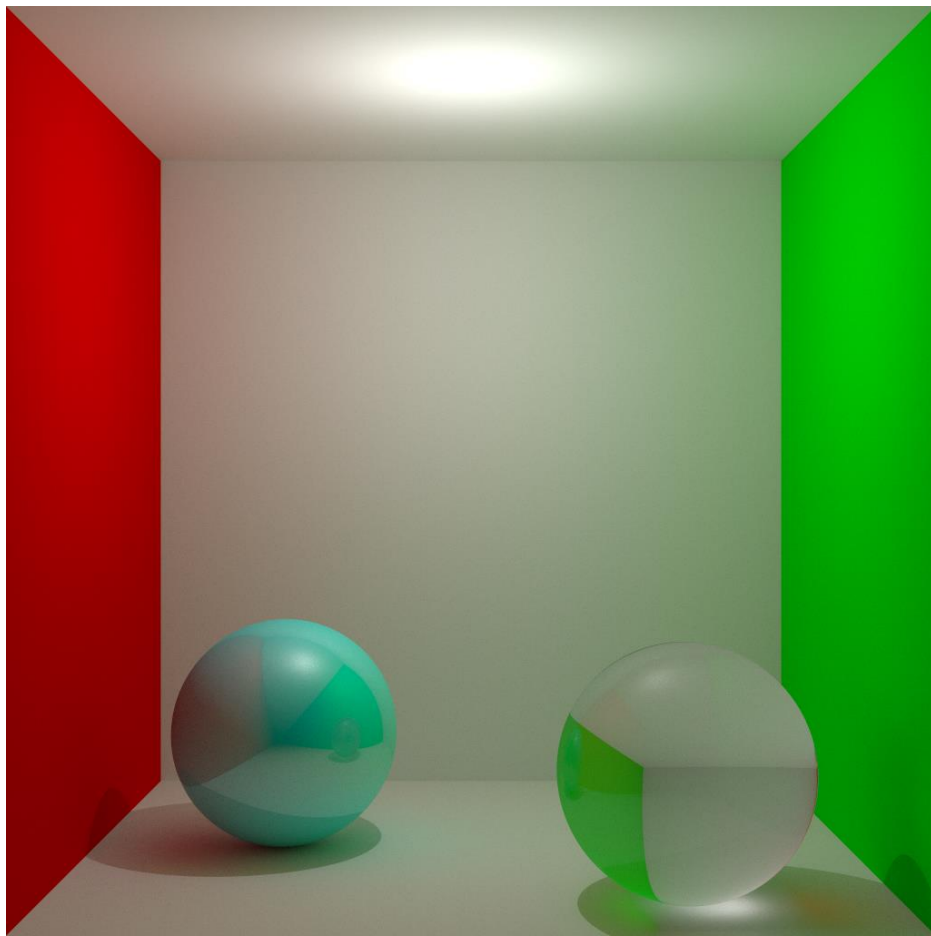


Path Tracer

INFORAMTICA GRÁFICA (2024-2025)



Alex Petrov

NIP: 812019 | 18/06/2025

Contenido

1. Introducción	2
2. Ecuación de render	2
2.1. Iluminación directa	3
2.2. Iluminación indirecta	3
3. Escenas.....	5
3.1. Cornell box con bola de plástico y bola de cristal	5
3.2. Cornell box con bolas difusas.....	6
3.3. Convergencia del algoritmo	6
4. Efectos de iluminación.....	7
4.1. Sombras duras	8
4.2. Sombras suaves	8
4.3. Color bleeding.....	9
4.4. Causticas	9
5. Extensiones Implementadas	10
5.1. Otras geometrías.....	10
5.2. Parelelizacion.....	10
5.3. Utilidades	11
6. Renders Adicionales.....	12
7. Referencias.....	13

1. Introducción

El presente trabajo tiene como propósito principal diseñar e implementar desde los fundamentos un sistema de Path Tracing en lenguaje C++, enmarcado dentro de la asignatura Informática Gráfica.

A través de esta práctica se pretende adquirir una comprensión sólida tanto teórica como técnica sobre la generación realista de imágenes mediante la simulación física de la interacción de la luz en escenarios tridimensionales.

El Path Tracing destaca frente a otras técnicas convencionales como el Ray Tracing debido a su capacidad superior para reproducir fenómenos lumínicos complejos como la reflexión múltiple, refracción y otros efectos característicos de la iluminación global.

A lo largo de la memoria se explicarán los conceptos esenciales, haciendo énfasis en la ecuación de render, junto con demostraciones visuales de escenas producidas y un análisis de los resultados obtenidos, incluyendo las características avanzadas que se han añadido opcionalmente.

2. Ecuación de render

La ecuación de render se define como:

$$L_0(x, \vec{\omega}_0) = L_e(x, \vec{\omega}_0) + \int_{\Omega} L_i(x, \vec{\omega}_i) f_r(x, \vec{\omega}_i, \vec{\omega}_0) |\vec{n} \cdot \vec{\omega}_i| d\vec{\omega}_i$$

Donde:

- L_0 es la luz saliente en un punto x en una dirección $\vec{\omega}_0$
- L_e es la radiación emitida en un punto x en una dirección $\vec{\omega}_0$
- \int_{Ω} es la integral sobre todas las direcciones posibles ($\vec{\omega}_i$)
- L_i es la luz incidente en un punto x desde una dirección $\vec{\omega}_i$
- $f_r(x, \vec{\omega}_i, \vec{\omega}_0)$ es la función de reflectancia de la superficie, o BRDF (*Bidirectional Reflectance Distribution Function*). Esta define como una superficie refleja la luz incidente, relacionando la dirección de entrada $\vec{\omega}_0$ con la dirección de salida $\vec{\omega}_i$.
- $|\vec{n} \cdot \vec{\omega}_i|$ es el producto escalar entre la dirección de llegada de la luz ($\vec{\omega}_i$) y la normal de la superficie (\vec{n}).

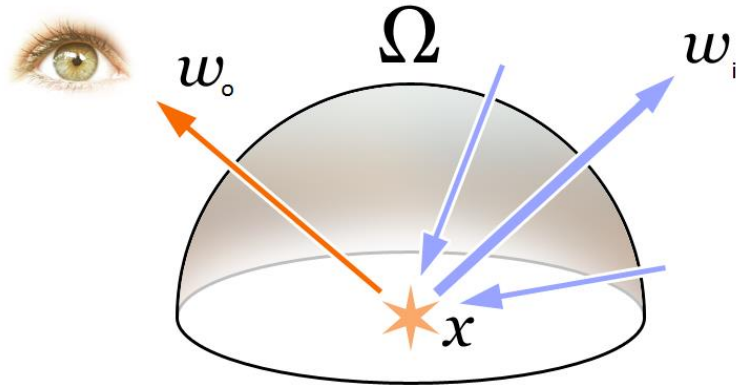


Figura 2.1 - Representación visual de la ecuación de render (Wikipedia)

El path tracer desarrollado calcula la radiancia saliente considerando tanto la radiancia emitida directamente por fuentes luminosas (L_e) como la radiancia incidente indirecta (L_i). Se usan métodos de Monte Carlo para integrar numéricamente sobre las direcciones entrantes (Ω), emplea modelos BRDF Lambertianos y especulares para f_r , y considera explícitamente el factor coseno ($|\vec{n} \cdot \vec{\omega}_i|$) en el cálculo de la iluminación en cada intersección rayo-superficie.

2.1. Iluminación directa

El path tracer desarrollado calcula la iluminación directa con la siguiente expresión:

$$L_0(x, \vec{\omega}_0) = \sum_{i=1}^N \frac{p_i}{|c_i - x|^2} f_r(x, \vec{\omega}_i, \vec{\omega}_0) \left| \vec{n} \cdot \frac{c_i - x}{|c_i - x|} \right|$$

Donde:

- p_i es la potencia de la fuente de luz i
- c_i es el centro de la fuente de luz i

Es decir, la luz saliente en un punto, es la suma de todas las luces incidentes en ese punto.

Esta expresión se utiliza en *next-event estimation*, para mejorar significativamente la eficiencia del path tracing. En cada intersección con un material, también se lanzan rayos de luz a cada fuente de luz, para estimar de forma precisa la luz directa que incide en cada lugar. Mientras haya un rayo directo haya una fuente de luz, sin que intersecte con algún otro objeto, se sumara la contribución de next-event estimation a la luz final obtenida en un punto.

2.2. Iluminación indirecta

En la Ecuación de Render, la iluminación global queda representada mediante una integral sobre todas las direcciones posibles ($\vec{\omega}_i$). En este trabajo, dicha integral se estima numéricamente utilizando el método Monte Carlo.

Dado que la integral de la ecuación de render es un cálculo extremadamente costoso, aproximamos dicha integral con el método de Monte Carlo.

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

N es el número de muestras contenidas en el intervalo $[a, b]$. Dado que este método genera hay que tener en cuenta la importancia cada muestra (probabilidad de haberla generado), obteniendo finalmente la siguiente expresión:

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

Donde:

- $p(x_i)$ es la probabilidad de que pase $f(x_i)$

Además, se implementa una técnica de ruleta rusa para modelar la variedad de rebotes según el tipo de material.

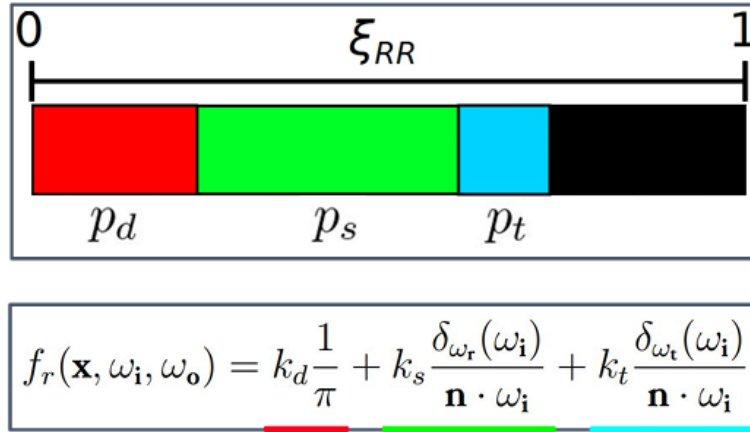


Figura 2.2 - probabilidades de los eventos de la Ruleta Rusa

Tras lanzar un rayo desde la cámara y producirse una intersección con alguna superficie, se generan recursivamente nuevos rayos. La dirección de estos rayos secundarios depende directamente del tipo de material presente:

- Para superficies difusas se emplea muestreo coseno,
- Para superficies especulares se calcula la reflexión perfecta
- En superficies refractantes se determina la dirección exacta de refracción.

Este proceso recursivo continúa hasta que el rayo es absorbido, sale del escenario o impacta contra una luz explícita. De este modo, en el trabajo desarrollado se calcula la iluminación global combinando la contribución indirecta (resultado de las interacciones recursivas del rayo) y la contribución directa proporcionada por las fuentes luminosas.

3. Escenas

Para mostrar los resultados que produce el path tracer desarrollado, se mostraran distintos rendes sobre dos escenas:

- Cornell box con paredes difusas, pared roja a la izquierda y verde a la derecha, con dos bolas difusas, una azul y otra rosa.
- Cornell box con paredes difusas, pared roja a la izquierda y verde a la derecha, con dos bolas, una de plástico azul (mix de difuso y especular), y otra bola de cristal (mix de especular y refracción)

Cada imagen se renderiza a una resolución de 256x256 pixeles, y 6 rebotes para cada path del algoritmo de path tracing.

Se harán muestras con 2, 8, 32, 128, 512 y 2048 muestras por pixel

3.1. Cornell box con bola de plástico y bola de cristal

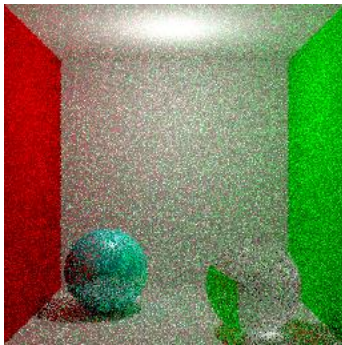


Figura 3.2 - 2 muestras/pixel

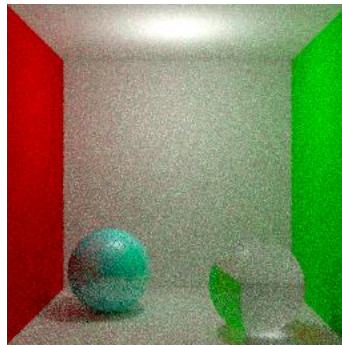


Figura 3.2 - 8 muestras/pixel

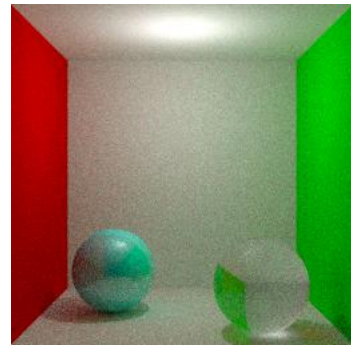


Figura 3.3 - 32 muestras/pixel

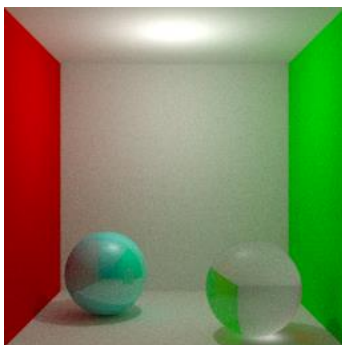


Figura 3.4 - 128 muestras/pixel

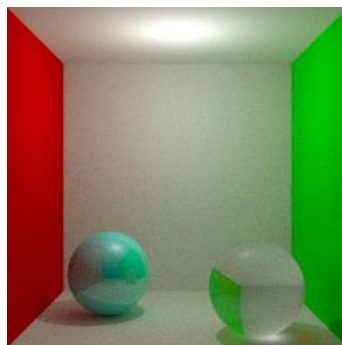


Figura 3.5 - 512 muestras/pixel

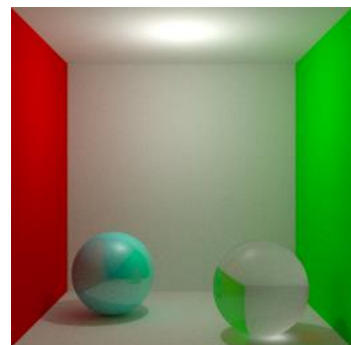


Figura 3.6 - 2048 muestras/pixel

La figura muestra claramente cómo afecta el número de muestras por píxel en la calidad visual de las imágenes generadas mediante el Path Tracing. Con pocas muestras por píxel (2 o 8), se observa un ruido significativo debido a la alta varianza del método de Monte Carlo. A medida que aumentamos el número de muestras (32, 128, 512, hasta 2048), el ruido se reduce progresivamente, logrando imágenes más suaves, nítidas y realistas. Este fenómeno se debe a que el promedio estadístico de múltiples muestras por píxel disminuye la variabilidad inherente al muestreo aleatorio. Así, un mayor número

de muestras mejora sustancialmente la calidad del resultado final a costa de un aumento del tiempo de renderizado.

3.2. Cornell box con bolas difusas

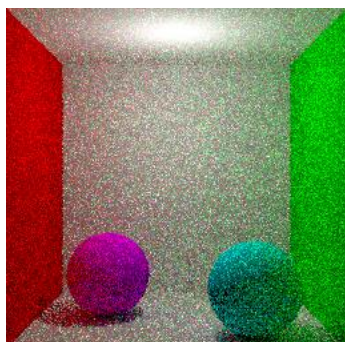


Figura 3.3 - 2 muestras/pixel

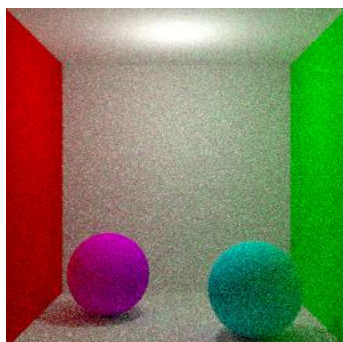


Figura 3.8 - 8 muestras/pixel

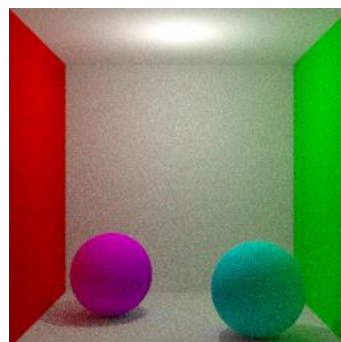


Figura 3.9 - 32 muestras/pixel

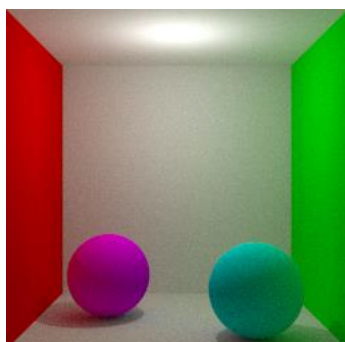


Figura 3.10 - 128
muestras/pixel

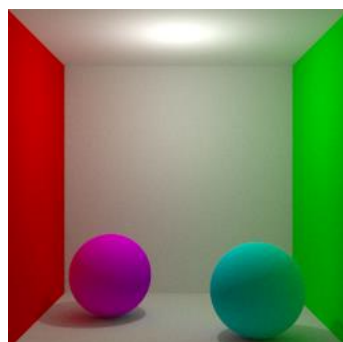


Figura 3.11 - 512
muestras/pixel

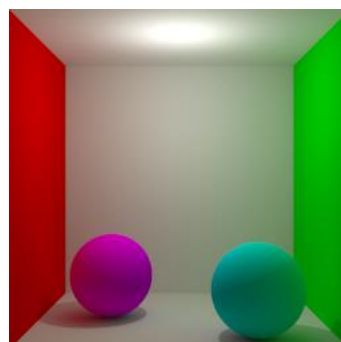


Figura 3.12 - 2024
muestras/pixel

Al igual que la escena anterior, a medida que se incrementa el número de muestras por píxel, se aprecia una notable reducción del ruido y una mejora progresiva en la definición de los colores y la iluminación.

En comparación con la escena anterior (con materiales reflectantes y refractivos), el proceso de convergencia es más uniforme y estable, ya que los materiales difusos distribuyen la energía de manera más homogénea, sin concentraciones de luz intensas. Esto permite al path tracer obtener resultados visualmente aceptables con menos muestras, aunque el ruido sigue presente en configuraciones de baja calidad.

3.3. Convergencia del algoritmo

En cuanto a la escena del apartado 3.1, estos son los resultados:

Muestras/pixel	Tiempo
2	0.331769s
8	0.97382s
32	3.49526s
128	13.4607s
512	53.4414s
2024	3m 44.8331s

En cuanto a la escena del apartado 3.2, estos son los resultados:

Muestras/pixel	Tiempo
2	0.325316s
8	0.902757s
32	3.61953s
128	13.488s
512	54.6833s
2024	3m 46.7738s

Para más detalles, se ha renderizado la escena 3.1, con una resolución de 1024x1024, con distintas muestras por pixel con 6 rebotes por path.

Muestras/pixel	Tiempo
2	3.80654s
4	7.17262s
8	13.9734s
16	27.3706s
32	55.961s
64	1m 49.9717s
128	3m 46.4263s
256	7m 37.4052s
512	16m 25.7574s

4. Efectos de iluminación

Para mostrar los efectos de las imágenes, se ha renderizado una imagen de la escena del apartado 3.1 con una resolución 1024x1024 con 512 muestras por pixel y 6 rebotes para los paths del algoritmo de path tracing.

4.1. Sombras duras

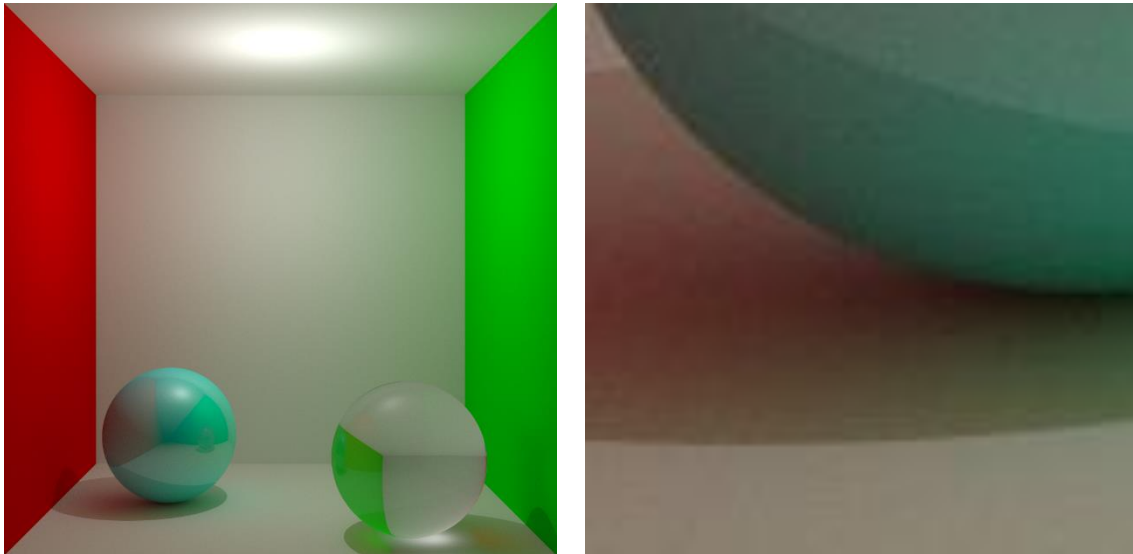


Figura 4.1 - Visualización de las sombras duras de las bolas

La imagen ilustra claramente el fenómeno conocido como "sombras duras", típicas en fuentes de luz puntuales. Se observa que las sombras proyectadas por los objetos presentan contornos bien definidos y sin degradados suaves.

Al analizar de cerca la ampliación, puede apreciarse claramente el borde nítido de la sombra proyectada sobre la superficie, lo que indica que la fuente de iluminación utilizada es puntual, generando una transición abrupta entre las zonas iluminadas y las que permanecen en sombra.

4.2. Sombras suaves

Las sombras suaves ocurren cuando la fuente de luz es más extensa o difusa. En este caso, la luz se dispersa alrededor de los objetos, creando transiciones más suaves entre las áreas iluminadas y las sombras. De esta manera, se puede considerar que se pueden obtener sombras suaves fácilmente con luces de área en path tracer. Este efecto también es independiente de los materiales de la escena.

4.3. Color bleeding

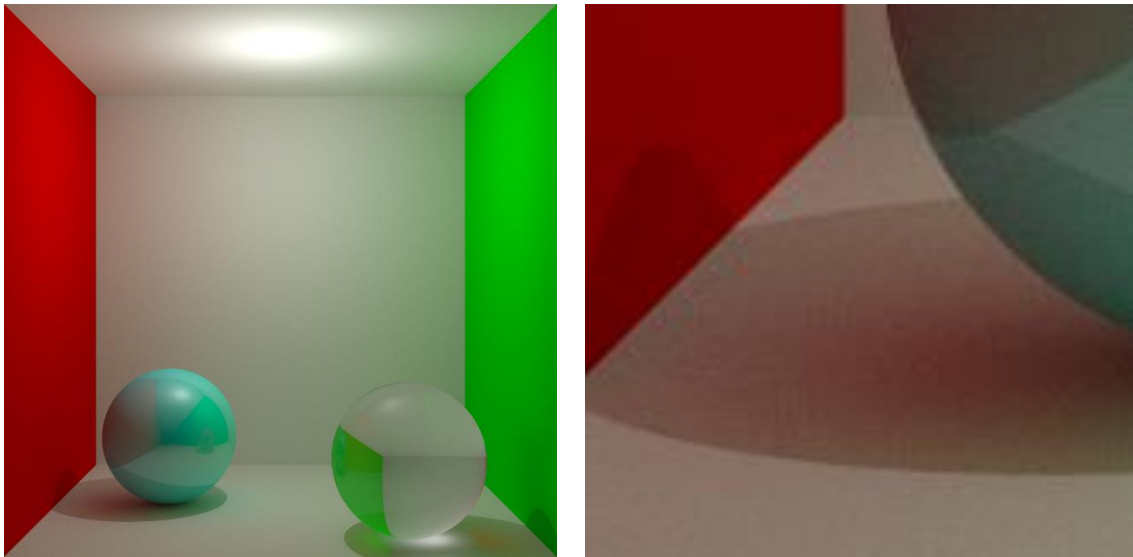


Figura 4.2 - Visualización del efecto de color bleeding

La figura muestra un claro ejemplo del efecto conocido como color bleeding, o "sangrado de color". Este fenómeno ocurre cuando la luz rebota en superficies coloreadas y transporta parte de ese color hacia otras zonas de la escena.

En la imagen ampliada puede observarse cómo el tono rojizo de la pared izquierda se refleja sobre el plano cercano y sobre la esfera, tiñendo suavemente las áreas de sombra. Este tipo de interacción sólo puede capturarse mediante técnicas de iluminación global como el path tracing, ya que requiere simular múltiples rebotes de luz que transportan información espectral a través de la escena.

4.4. Causticas

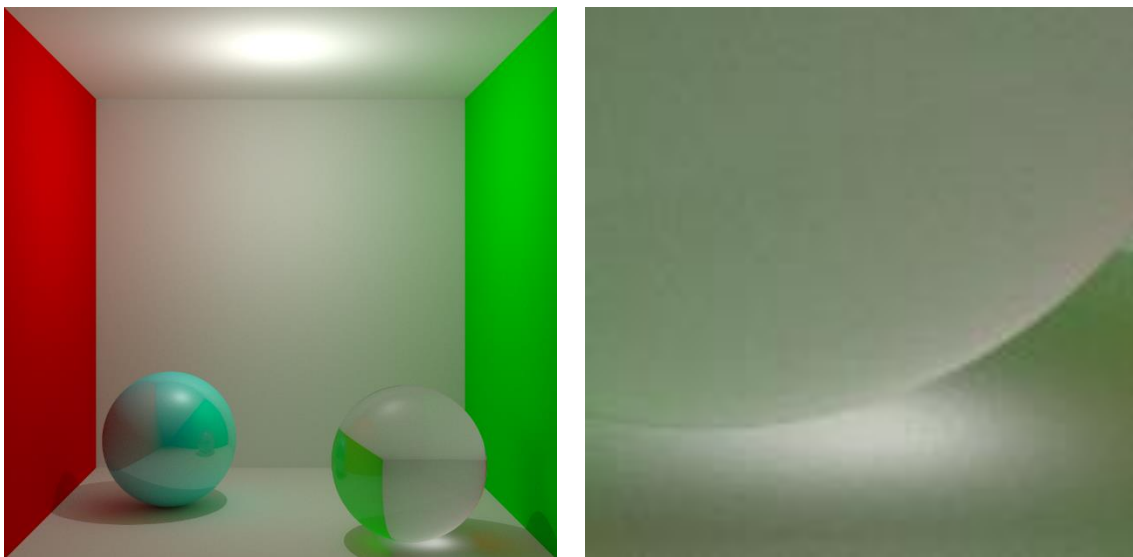


Figura 4.3 - Visualización del efecto de Causticas

La figura representa el fenómeno de las cáusticas, un efecto óptico característico que ocurre cuando la luz es concentrada por reflexión o refracción sobre superficies específicas.

En este caso, se observa cómo la esfera transparente refracta los rayos de luz, generando una zona iluminada más intensa sobre el plano inferior. La ampliación muestra claramente esta concentración de luz en forma de patrones curvos y brillantes, que no podrían reproducirse sin simular correctamente los múltiples rebotes y trayectorias de los rayos. Este tipo de efectos resalta la capacidad del path tracing para capturar comportamientos complejos de la luz en materiales dieléctricos.

En Path Tracing, las cáusticas generadas por luces puntuales son difíciles de capturar, ya que requieren una gran cantidad de muestras para que los rayos aleatorios coincidan con los caminos de concentración luminosa. En cambio, técnicas como Photon Mapping son más eficientes para este tipo de efectos, ya que depositan explícitamente fotones en las zonas iluminadas, resolviéndolas con mayor precisión y menor ruido.

5. Extensiones Implementadas

5.1. Otras geometrías

En este trabajo se ha ampliado el conjunto de primitivas geométricas soportadas, incorporando nuevas clases que heredan de la interfaz base Figure. Además de esferas y planos, se han implementado cilindros (Cylinder), triángulos individuales (Triangle) y mallas triangulares completas (TriangleMesh), lo que permite representar modelos más complejos y detallados. Estas nuevas figuras han sido integradas dentro del sistema de intersección de escenas (FigureCollection), lo que permite gestionar múltiples tipos de geometrías de forma unificada y eficiente en el proceso de renderizado.

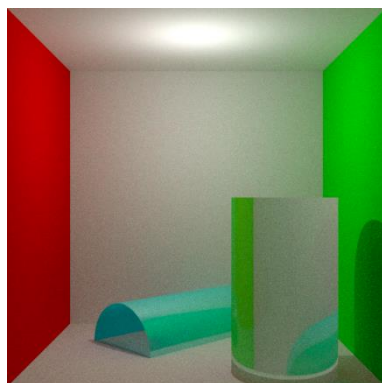


Figura 5.2 - Cilindros

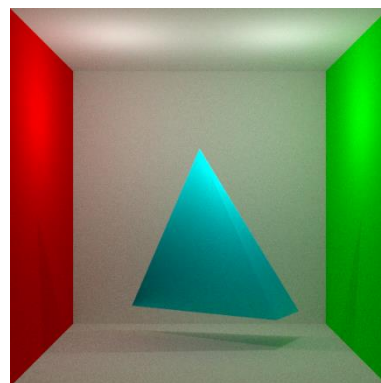


Figura 5.2 - Mapa de triángulos

5.2. Parellelizacion

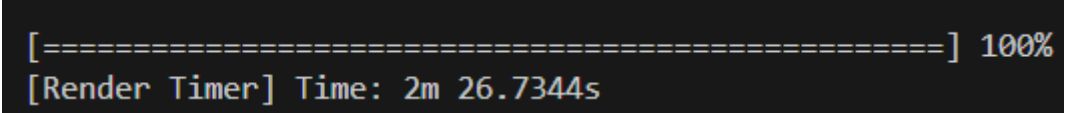
Este trabajo implementa la paralelización del renderizado utilizando una clase ThreadPool, que distribuye dinámicamente tareas de píxeles individuales entre múltiples hilos según el número de núcleos disponibles. Cada tarea calcula el color de un píxel y se encola de forma segura para ser ejecutada en paralelo, mejorando notablemente el rendimiento sin introducir bloqueo entre hilos.

Además, se utiliza un hilo adicional para actualizar una barra de progreso mediante una variable atómica compartida. Esta estrategia dinámica permite un reparto equilibrado de la carga, cumple con los requisitos del enunciado y mantiene una estructura modular y eficiente.

5.3. Utilidades

Con el objetivo de facilitar el análisis de rendimiento, se ha implementado la clase `ScopedTimer`, que permite medir automáticamente el tiempo de ejecución de un bloque de código mediante RAII. Al instanciarse, inicia un cronómetro, y al destruirse muestra el tiempo transcurrido en consola, sin necesidad de llamadas manuales.

Además, se ha incorporado una barra de progreso (progressbar) que informa en tiempo real del avance del renderizado, especialmente útil en ejecuciones largas o multihilo. Ambas herramientas han resultado fundamentales para el desarrollo, pruebas y comparación de mejoras.



```
[=====] 100%  
[Render Timer] Time: 2m 26.7344s
```

6. Renders Adicionales

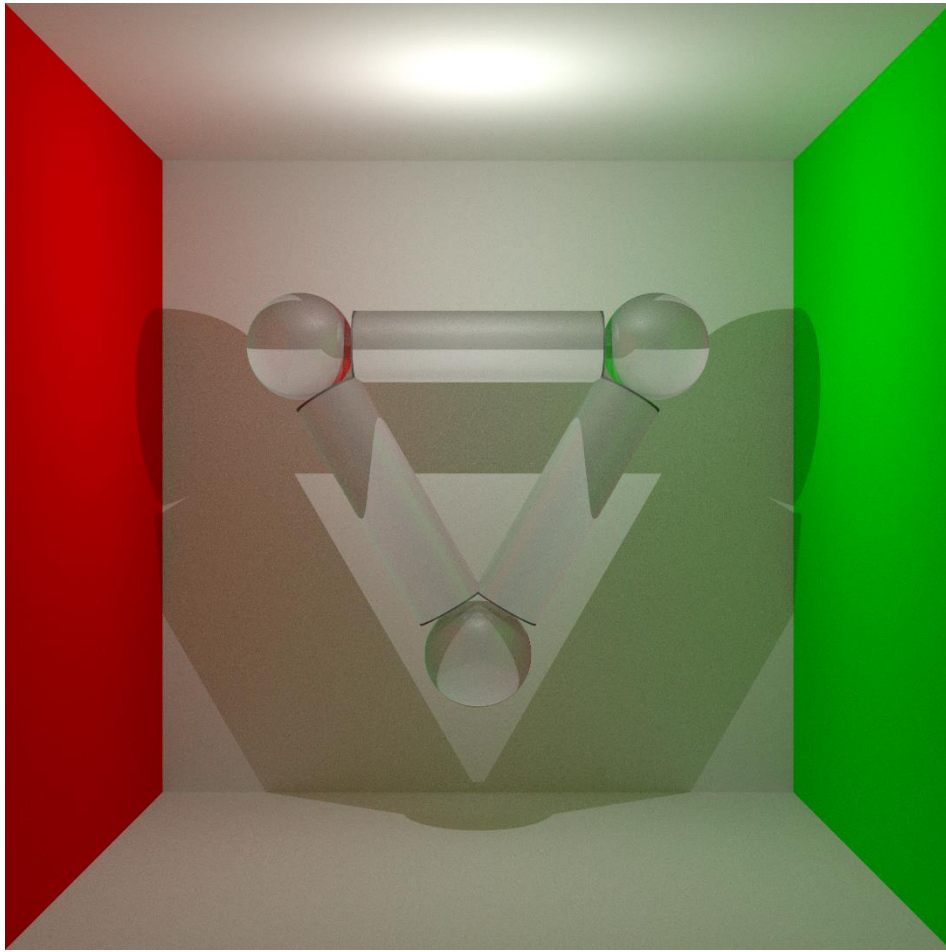


Figura 5.1 - Triangulo invertido con esferas y cilindros

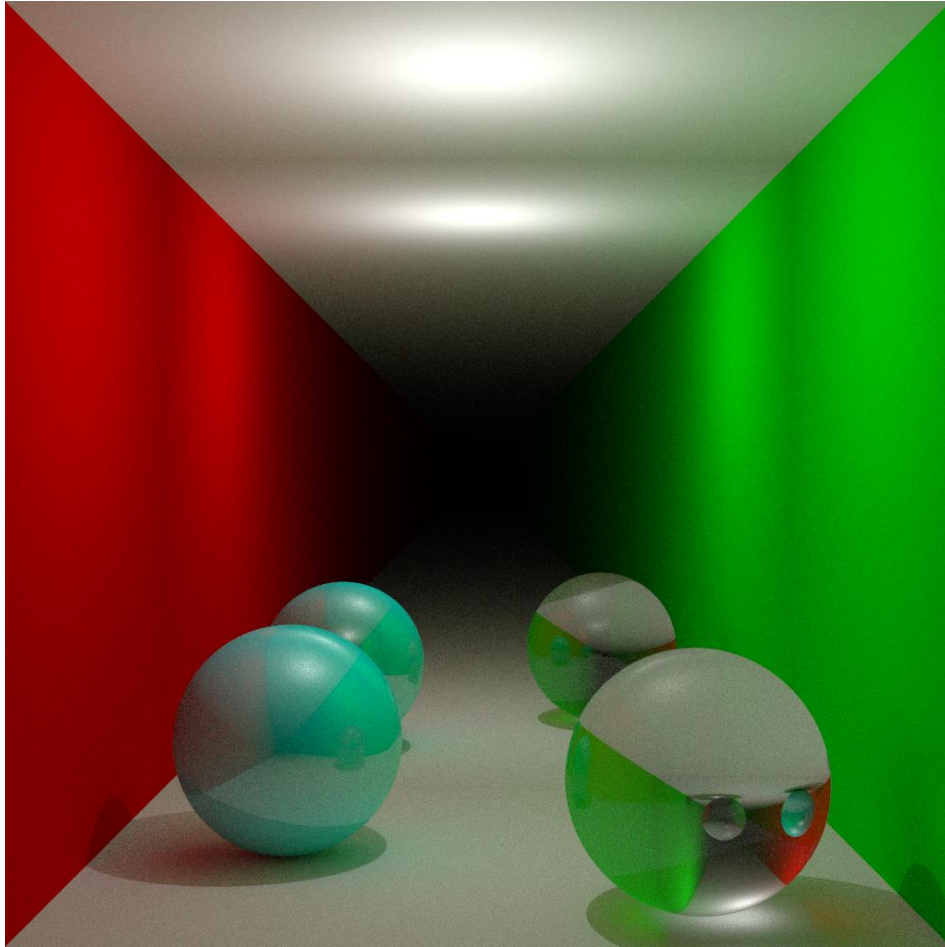


Figura 6.2 - Cornell box con pared de fondo espejo

7. Referencias

- [Ray Tracing in a Weekend.pdf](#)
- [Ray Tracing in One Weekend](#)
- <https://youtube.com/playlist?list=PLVotA8ycjnCuOGBTriaHdw80lpr1r33IM&si=bBXvteJsRxzSEa91>
- [Path tracing - Wikipedia](#)
- Material de clase