

Universidad Nacional Mayor de San Marcos

Universidad del Perú. Decana de América

Facultad de Ciencias Matemáticas

Escuela Profesional de Computación Científica

Limitaciones de un algoritmo de renderización volumétrica

TESIS

Para optar el Título Profesional de Licenciado en Computación Científica

AUTOR

Miguel Alejandro ESTELA BEJARANO

ASESOR

Mg. Jhelly Reynaluz PÉREZ NUÑEZ

Lima, Perú

2021



Reconocimiento - No Comercial - Compartir Igual - Sin restricciones adicionales

https://creativecommons.org/licenses/by-nc-sa/4.0/

Usted puede distribuir, remezclar, retocar, y crear a partir del documento original de modo no comercial, siempre y cuando se dé crédito al autor del documento y se licencien las nuevas creaciones bajo las mismas condiciones. No se permite aplicar términos legales o medidas tecnológicas que restrinjan legalmente a otros a hacer cualquier cosa que permita esta licencia.

Referencia bibliográfica

Estela, M. (2021). *Limitaciones de un algoritmo de renderización volumétrica*. [Tesis de pregrado, Universidad Nacional Mayor de San Marcos, Facultad de Ciencias Matemáticas, Escuela Profesional de Computación Científica]. Repositorio institucional Cybertesis UNMSM.

Metadatos complementarios

Datos de autor	
Nombres y apellidos	Miguel Alejandro Estela Bejarano
Tipo de documento de identidad	DNI
Número de documento de identidad	77298381
URL de ORCID	
Datos de asesor	
Nombres y apellidos	Jelly Reynaluz Pérez Núñez
Tipo de documento de identidad	DNI
Número de documento de identidad	44076412
URL de ORCID	https://orcid.org/0000-0003-0717-8277
Datos del jurado	
Presidente del jurado	
Nombres y apellidos	María Natividad Zegarra Garay
Tipo de documento	DNI
Número de documento de identidad	09206994
Miembro del jurado 1	
Nombres y apellidos	Carlos Augusto Ruiz De La Cruz Melo
Tipo de documento	DNI
Número de documento de identidad	08249640
Datos de investigación	
Línea de investigación	No aplica
Grupo de investigación	No aplica
Agencia de financiamiento	Sin financiamiento
Ubicación geográfica de la investigación	Edificio: Universidad Nacional Mayor de San Marcos País: Perú Departamento: Lima Provincia: Lima Distrito: Cercado de Lima Latitud: -12.06041 Longitud: -77.08228

Año o rango de años en que se realizó la investigación	Julio 2021 - Setiembre 2021
URL de disciplinas OCDE	Ciencias de la computación https://purl.org/pe-repo/ocde/ford#1.02.01 Óptica https://purl.org/pe-repo/ocde/ford#1.03.06

ACTA DE SUSTENTACIÓN DE TESIS EN LA MODALIDAD VIRTUAL PARA LA OBTENCIÓN DEL TÍTULO PROFESIONAL DE LICENCIADO(A) EN COMPUTACIÓN CIENTÍFICA (PROGRAMA DE TITULACIÓN PROFESIONAL 2021-I)

En la UNMSM – Ciudad Universitaria – Facultad de Ciencias Matemáticas, siendo las 08:40 horas del jueves 21 de octubre del 2021, se reunieron los docentes designados como Miembros del Jurado Evaluador (PROGRAMA DE TITULACIÓN PROFESIONAL 2021-I): Dra. María Natividad Zegarra Garay (PRESIDENTE), Mg. Carlos Augusto Ruiz De La Cruz Melo (MIEMBRO) y la Mg. Jhelly Reynaluz Pérez Núñez (MIEMBRO ASESOR), para la sustentación de la Tesis titulada: "**LIMITACIONES DE UN ALGORITMO DE RENDERIZACIÓN VOLUMÉTRICA**", presentado por el señor **Bachiller Miguel Alejandro Estela Bejarano**, para optar el Título Profesional de Licenciado en Computación Científica.

Luego de la exposición de la Tesis, la Presidente invitó al expositor a dar respuesta a las preguntas formuladas.

Realizada la evaluación correspondiente por los Miembros del Jurado Evaluador, el expositor mereció la aprobación **bueno,** con un calificativo promedio de **dieciséis. (16).**

A continuación, los Miembros del Jurado Evaluador dan manifiesto que el participante **Bachiller Miguel Alejandro Estela Bejarano** en vista de haber aprobado la sustentación de su Tesis, será propuesto para que se le otorgue el Título Profesional de Licenciado en Computación Científica.

Siendo las 09:20 horas se levantó la sesión firmando para constancia la presente Acta.

UNMSM
Firmado digitalmente por ZEGARRA GARAY María Natividad FAU 20148092282 soft
Motivo: Soy el autor del documento
Fecha: 03.11.2021 17:08:59 -05:00

Dra. María Natividad Zegarra Garay
PRESIDENTE

Mg. Carlos Augusto Ruiz De La Cruz Melo
MIEMBRO

Mg. Jhelly Reynaluz Pérez Núñez
MIEMBRO ASESOR

Resumen

Sintetizar imágenes fotorrealistas por computadora es un proceso complejo, debido a que para alcanzar el realismo debemos simular el comportamiento de la luz. En el mundo real, la luz interactúa con todo lo que nos rodea, sin embargo en un programa debemos definir como es que este interactúa con la materia, puesto que en algunos casos puede interactuar con su superficie o el volumen del mismo. Los medios participativos son responsables de una gran parte de fenómenos ópticos que podemos observar en el mundo, algunos ejemplos son la atmósfera, las nubes, el humo, puesto que la luz interactúa en todo su volumen.

Este trabajo se enfoca en la interacción de la luz en medios participativos en un algoritmo de trazado de rayos en *Mitsuba 2*, para analizar las limitaciones de este, además para comprender todo este proceso se muestra las herramientas físicas y matemáticas que tenemos para describir este fenómeno.

Palabras Clave: Renderización, volumen, transporte de luz, Mitsuba, limitaciones

Abstract

Creating computer generated images is a complex process, due to achieve a high level of detail we need to simulate light behavior. In real life, light interacts with everything that surround us, however, when developing a computer program, we need to define how light interacts with matter, since light can interact with its surface or volume. Participating media is responsible for many observable optical phenomena in our world, some examples are atmosphere, clouds, and smoke, because light interacts inside them.

This work focus at participating media light interaction in a ray tracing algorithm using *Mitsuba 2*, to analyze their limitations, also to comprehend this process, we present physics and math tools that describe this phenomena.

Keywords: Rendering, volume, light transport, Mitsuba, limitations

Acta de sustentación

Índice general

I Introducción . . . 6
II Información del lugar donde se realizó la actividad . . . 7
III Descripción de la actividad . . . 8
3.1 Presentación . . . 8
3.2 Marco teórico . . . 9
3.2.1 Naturaleza de la Luz . . . 9
3.2.2 Radiometría . . . 10
3.2.3 Iluminación global . . . 12
3.2.4 Transferencia Radiativa . . . 15
3.3 Trabajo Relacionado . . . 21
3.3.1 Volúmenes . . . 21
3.3.2 Propagación de la luz . . . 22
3.3.3 Ecuación de renderización para volúmenes . . . 24
3.3.4 Métodos para resolver la VRE . . . 25
3.4 Metodología . . . 25
3.4.1 Eligiendo un sistema de renderización . . . 26
3.4.2 Path tracer . . . 26
3.4.3 Volume path tracer . . . 27
3.5 Resultados . . . 28
IV Conclusiones . . . 32
V Recomendaciones . . . 33
Bibliografía . . . 34
Lista de Símbolos . . . 36
Terminología . . . 38
A Renderización . . . 40
1.1 Renderización basada en físicas . . . 40
1.1.1 Sistemas de Renderización . . . 41
B Monte Carlo . . . 43
2.1 Probabilidades . . . 43
2.1.1 Variables aleatorias . . . 43
2.1.2 Función de distribución acumulada . . . 43
2.1.3 Función de densidad . . . 44
2.1.4 Esperanza . . . 44
2.1.5 Varianza . . . 44
2.2 Estimador de Monte Carlo . . . 44
2.2.1 Esperanza y convergencia . . . 45
2.3 Reducción Varianza . . . 46
2.3.1 Muestreo por importancia . . . 46

2.3.2 Ruleta rusa 46
C Scripts 47

Capítulo I

Introducción

La ecuación de renderización por Kajiya (1986) es una forma reintepretada del modelo de transferencia radiativad desarrollada por Chandrasekhar (1960) y deriva en parte de los modelos de transferencia de calor radiante para poder ser simulada, en otra palabras, describe el transferencia de luz para ser simulada computacionalmente. Esta ecuación tiene diferentes aproximaciones, todas resolviendo el problema de transferencia de luz, sin embargo todas tienen ventajas y desventajas; algunas debido al alto costo computacional, otras al interpretar un fenómeno óptico debidamente.

En este trabajo se mostrarán los resultados de implementar la transferencia radiativa en un algoritmo de trazado de rayos basado en las técnicas de *Monte Carlo* de forma que podremos saber cuales son los límites del programa para interpretar este fenómeno físico. Primero haremos una revisión a los conceptos más básicos para comprender que es lo que se mide y como se mide, revisaremos el trabajo relacionado hasta la fecha con respecto a este problema puesto que al ser la renderización basada en físicas un campo multidisciplinario el proceso es complejo. Se continuará con los resultados obtenidos del algoritmo implementado en el programa de *Mitsuba*. En el apartado de resultado mostraremos las escenas que se han creado para reconocer que tipos de fenómenos están al alcance de este algoritmo. Finalmente llegaremos a las conclusiones que este trabajo a encontrado.

Capítulo II

Información del lugar donde se realizó la actividad

Institución donde se desarrolló la actividad: *Universidad Nacional Mayor de San Marcos*

Periodo de duración de la actividad: *02/07/2021 - 15/10/2021*

Razón social: *Universidad Nacional Mayor de San Marcos*

Dirección Postal: *Calle Germán Amezaga N° 375*

Capítulo III

Descripción de la actividad

3.1. Presentación

En este capítulo se revisará los conceptos básicos necesarios para comprender las herramientas que se utilizarán para la renderización de un volumen usando un algoritmo de trazado de rayos en Mitsuba 2. Las secciones a revisar de este capítulo son:

- Marco teórico, se revisará la teoría necesaria para comprender el fenómeno de la luz, como se mide y como la formulación física-matemática de este fenómeno es interpretado en los algoritmos de Iluminación global para simular la interacción de la luz con la materia, además se mencionarán dos técnicas importantes de las múltiples que han sido desarrolladas a lo largo de estos años para la iluminación global, finalmente se revisará la teoría de la transferencia radiativa que explica las la interacción de la luz en un medio participativo y como esta contribuye a la medida de radiancia en un punto en el espacio.
- Trabajo relacionado, se mostrará la interpretación actual de este fenómeno de interacción de luz en medios participativos así como los trabajos relacionados al desarrollo de algoritmos para la iluminación global y los límites de cada uno.
- Metodología, se describirá el software que se ha utilizado en este trabajo y la razón principal por que se elegido tal, así como una descripción del algoritmo utilizado para representar este fenómeno.
- Resultados, mostraremos las imágenes obtenidas después de muestrar la RTE para computar la VRE y observar el resultado final en la escena sintetizada, observaremos su varianza con otro algoritmo que hace uso del muestreo por importancia para la VRE.
- Recomendaciones, se mencionará la razón por la cual es de suma importancia considerar técnicas para la reducción de varianza en este tipo de trabajo.

3.2. Marco teórico

3.2.1. Naturaleza de la Luz

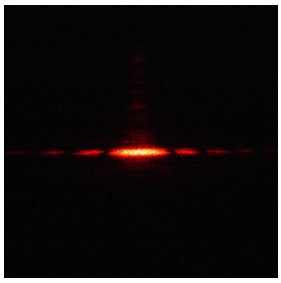
La luz es una onda electromagnética; la luz es una corriente de fotones que transportan paquetes de energía; la luz, a pesar de ser una onda, viaja a lo largo de lineas rectas o rayos, este comportamiento nos permiten estudiar lentes y espejos.

Entonces ¿Qué es la luz? La luz es todo lo mencionado anteriormente, pero estas descripciones físicas que la define y explica su comportamiento dependen mucho de la escala en la que se la estudie.

En Bahaa E. A. Saleh (2007) mencionan que las teorías ópticas que describen su comportamiento se han desarrollado históricamente en el siguiente orden: óptica geométrica, óptica ondulatoria, óptica electromagnética y óptica cuántica; cada una explicando fenómenos ópticos que la anterior no describía.



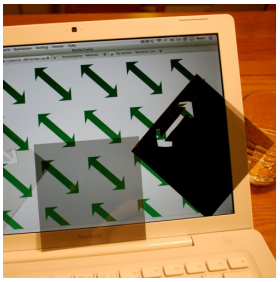
(a) Burbuja de jabón
por Raspopova (2020)



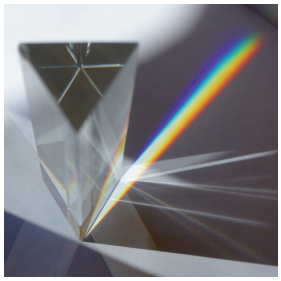
(b) Experimento de *Fresnel*
por Sauerbronn (2008)

Figura 3.1: Interferencia y difracción

La figura 3.1a nos muestra el fenómeno de interferencia por el grosor de la película de la burbuja de jabón ya que los colores que vemos son debidos a la luz reflejada por su superficie exterior e interior. La figura 3.1b es el experimento de difracción de Fresnel que nos muestra la naturaleza de onda que tienen la luz.



(a) Polarización LCD
por Kuiper (2008)

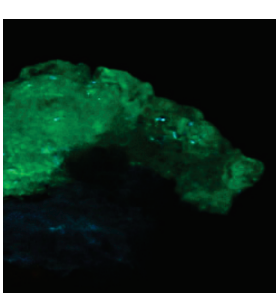


(b) Dispersión en un prisma
por Ma (2013)

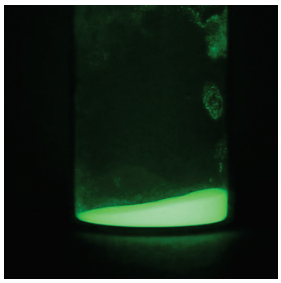
Figura 3.2: Polarización y dispersión

La polarización de la luz viene dada por su naturaleza como onda electromagnética, en la figura 3.2a podemos observar como el filtro polarizador de la pantalla LCD transmite

luz, con un vector de campo eléctrico paralelo a las flechas, y como este no trasmite luz cuando los filtros están cruzados. En la figura 3.2b podemos observar como el prisma hace que la luz se ralentice y curve su camino separándola en ondas de distinta frecuencia.



(a) Hialita
por Dulyan (2005)



(b) Pigmento de sulfuro de zinc
por Michicgn (2008)

Figura 3.3: Fluorescencia y fosforescencia

En una escala más pequeña, existen sistemas moleculares que poseen la capacidad de emitir radiación, pero su interacción con la radiación electromagnética depende de la velocidad de absorción y emisión. La figura 3.3a muestra la fluorescencia de una hialita irradiada por luz ultravioleta y la figura 3.3b muestra un pigmento de sulfuro de zinc emitiendo radiación en la oscuridad.

La Luz en computación gráfica

Es la luz lo que nos permite ver cosas y le da perspectiva a nuestra existencia, entonces si podemos simular su comportamiento podemos darle perspectiva a mundos inexistentes.

En el campo de computación gráfica se investiga métodos para sintetizar y manipular imágenes, donde la renderización (apéndice A) es un proceso para generar imágenes fotorrealistas, y para lograr este resultado se suele simular el comportamiento de la luz utilizando herramientas matemáticas.

3.2.2. Radiometría

La luz es un tipo de radiación electromagnética y la radiometría es la ciencia que estudia la medida de la radiación electromagnética sobre todas las longitudes de onda del espectro electromagnético. Existe también la fotometría, pero esta solo estudia el espectro visible que percibe el ser humano.

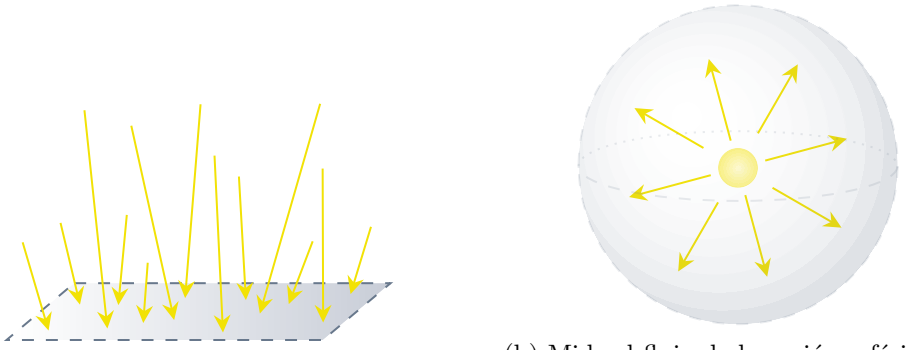
En Pharr y col. (2016) se menciona que la radiometría proporciona un conjunto de ideas y herramientas matemáticas que describen la propagación y reflexión de la luz, además forman las bases de varios algoritmos de renderización. Sin embargo, originariamente fue desarrollada usando óptica geométrica.

Medidas radiométricas

Energía. Consideremos un haz de luz que viaja en el espacio como en la figura 3.4, podemos decir que este transporta energía en una dirección, esta es llamada energía radiante, se simboliza con Q y es medida en J Joules.

Figura 3.4: Rayos de luz en el espacio.

Flujo. Flujo radiante, o potencia, es el total de energía que atraviesa una superficie o región del espacio por unidad de tiempo.



(a) Mide el flujo de una región del espacio.

(b) Mide el flujo de la región esférica que envuelve a la fuente de luz.

Figura 3.5: Rayos de luz que atraviesan una región en el espacio.

El flujo radiante Φ puede ser encontrado tomando el limite diferencial de energía Q por diferencial de tiempo t.

$$\Phi = \lim_{\Delta t \to 0} \frac{\Delta Q}{\Delta t} = \frac{dQ}{dt} \tag{3.1}$$

Su unidad es J/s o watts (W).

Densidad de flujo radiante. La densidad del flujo radiante viene dada cuando consideramos el área A sobre la cual pasan estos fotones.

$$\frac{d\Phi}{dA} \tag{3.2}$$

La *irradiancia* es el flujo radiante incidente, y la *exitancia radiante*, el emitido; cada uno por unidad de área de la superficie.

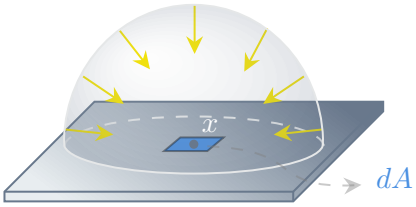


Figura 3.6: Irradiancia, flujo radiante incidente por unidad de área en un punto de una superficie.

Radiancia. Hemos visto que la irradiancia y la exitancia radiante es el diferencial del flujo radiante sobre una diferencial de área, pero estos no distinguen la distribución direccional del flujo; esta es llamada *Radiancia* L, el cual mide la irradiancia o exitancia radiante con respecto a un ángulo sólido Ω.

$$L = \frac{d\Phi}{cos\theta\, dA\, d\Omega} = \frac{d\Phi}{dA^{\perp}\, d\Omega} \tag{3.3}$$

donde $dA^{\perp} = dA\, cos\theta$ es llamada área proyectada.

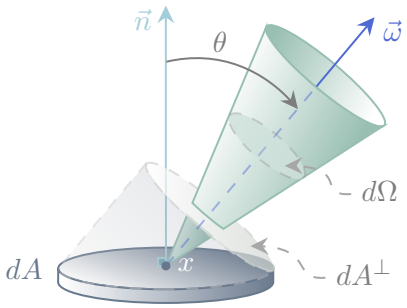


Figura 3.7: Geometría de la definición de radiancia.

Radiancia en renderización. La radiancia es una de las medidas métricas más importantes en los algoritmos de iluminación global y es por eso que se suele definir respecto a una dirección diferencial $d\vec{\omega}$ en un punto x.

$$L(x, \vec{\omega}) = \frac{d\Phi(x, \vec{\omega})}{d\vec{\omega}\, dA^{\perp}(x)} = \frac{d\Phi(x, \vec{\omega})}{(\vec{n} \cdot \vec{\omega})\, d\vec{\omega}\, dA(x)} \tag{3.4}$$

Fotometría

Como se dijo al principio, la fotometría solo se encarga del espectro visible percibido por el ojo humano, además cada medida radiométrica espectral puede ser convertida a su equivalente fotométrico, ver cuadro 3.1, integrándola con la curva de respuesta espectral, la cual describe la sensibilidad del ojo humano a las diferentes longitudes de onda.

McCluney (2014) presenta más a detalle lo descrito en este apartado.

3.2.3. Iluminación global

Tenemos una idea de qué es la luz y como se mide, pero para poder generar una imagen fotorrealista por computadora de una escena virtual debemos entender como se comporta,

Radiometría	Unidad	Fotometría	Unidad
Energía radiante	joule (Q)	Energía lumínica	talbot (T)
Flujo radiante	watt (W)	Flujo luminoso	lumen (lm)
Intensidad radiante	$W\,sr^{-1}$	Intensidad luminosa	$lm\,sr^{-1}$ (cd candela)
Irradiancia	$W\,m^{-2}$	Iluminancia	$lm\,m^{-2}$ (lx lux)
Radiancia	$W\,m^{-2}\,sr^{-1}$	Luminancia	$lm\,m^{-2}\,sr^{-1}$ (nit)

Cuadro 3.1: Medidas radiométricas y sus equivalentes fotométricos

i.e., como interactúa con la materia. *Iluminación global* o GI se refiere a los algoritmos que, cuando determinan la luz que impacta en una superficie, no solo toman en cuenta las fuentes de luz (*iluminación local*), sino también la contribución por los objetos que la rodean, en Akenine-Möller y col. (2018) se presenta más a detalle la diferencia entre ambos.



(a) *Direct ilumination* con MIS, *samples* 128

(b) *Path tracer* *samples* 128, *depth* 5

Figura 3.8: Diferencias de la iluminación local y global.
Imágenes generadas con Nimier-David y col. (2019)

La Ecuación de renderización

Sabemos que la luz interactúa con la materia y una parte del resultado de este fenómeno llegará a nuestros ojos. Podríamos decir también que la cantidad de luz que llega a nuestros ojos desde la superficie de un objeto viene dada por la suma de la luz emitida del objeto, si fuera una fuente de luz, y la cantidad de luz reflejada.

Esta idea es descrita en Kajiya (1986) que formalmente dice que la radiancia saliente $L_o(x, \vec{\omega})$ de una superficie un punto x en dirección $\vec{\omega}$ es la suma de la radiancia emitida L_e y radiancia reflejada L_r.

$$L_o(x, \vec{\omega}) = L_e(x, \vec{\omega}) + L_r(x, \vec{\omega}) \tag{3.5}$$

Siendo la radiancia reflejada, computada como

$$L_r(x, \vec{\omega}) = \int_H L_i(x, \vec{\omega}_i)\, f_r(x, \vec{\omega}_i, \vec{\omega})\, (\vec{n} \cdot \vec{\omega}_i)\, d\vec{\omega}_i \tag{3.6}$$

Donde H es el hemisferio superior alrededor de la superficie normal $\vec{n}$ en x, f_r la *Bidirectional Reflectance Distribution Function* BRDF, vid. figura 3.9.

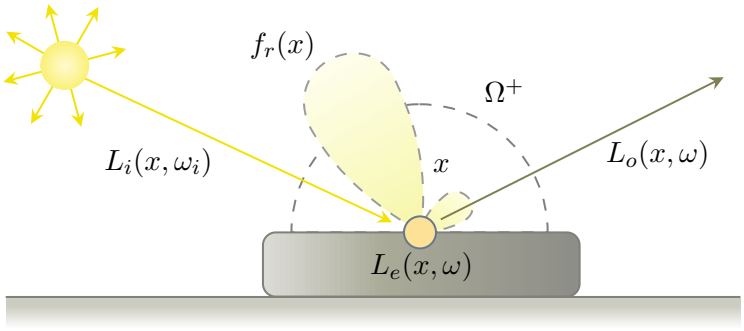
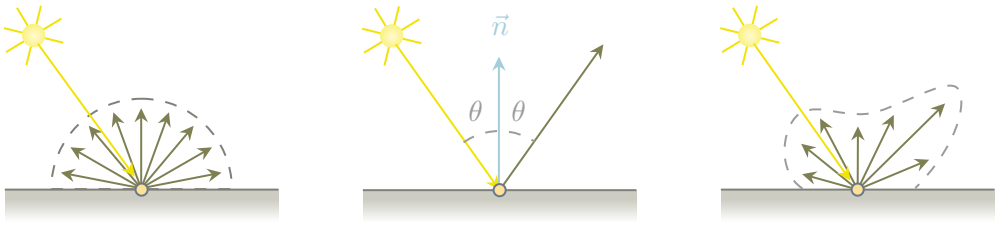


Figura 3.9: Diagrama esquemático de la ecuación de renderización

Cuando la luz interactúa con un objeto en la escena, esta interactúa con su superficie y puede ser reflejada, refractada o incluso absorbida. La BRDF nos dice como veríamos ese punto desde una dirección particular $\vec{\omega}$ mientras es iluminada desde otra dirección $\vec{\omega}_i$, vid. figura 3.10.



(a) Lambertiana (b) Especular perfecta (c) Reflexión arbitraria

Figura 3.10: Reflexión de la luz modelada por la f_r.

La figura 3.10a muestra la reflexión lambertiana o difusa ideal la cual refleja la luz incidente en todas las direcciones, la figura 3.10b refleja la luz con un ángulo igual a la luz incidente, como un espejo; y la figura 3.10c muestra una forma general de las BRDFs que definen una distribución de reflectancia por un rayo incidente.

Métodos para el cálculo de GI

Existen muchos métodos que intentan computar la interacción de la luz, en Ritschel y col. (2012) se presenta una recopilación de estos, sin embargo, solo mencionaré 2 de ellos.

Elementos finitos. O también conocido como de Radiosidad, descrito en Goral y col. (1984), es un método clásico para computar las soluciones a los problemas de transporte de luz. La idea principal de este método se basa en discretizar las superficies de la escena en elementos de superficie finita, llamados parches, y computar el transporte de luz entre ellos, vid. figura 3.11.

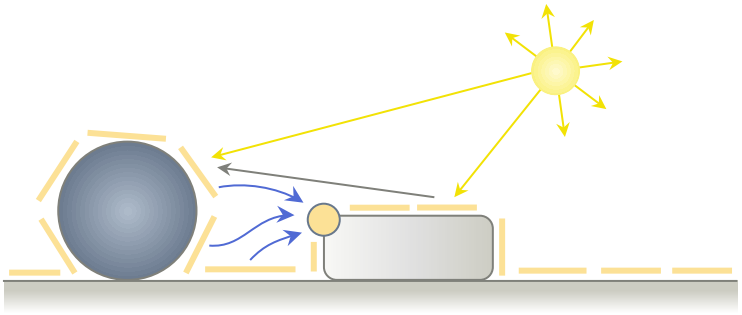


Figura 3.11: Diagrama esquemático del método de radiosidad.

Trazado de rayos. La ecuación de renderización puede ser resuelta usando técnicas de Monte Carlo (apéndice B), Kajiya (1986) también presenta estas técnicas en su publicación. Para resolverla, se utiliza un gran número de muestras direccionales que son evaluadas en la función a aproximar y el promedio de todos los resultados converge a la solución real, ver la figura 3.12.

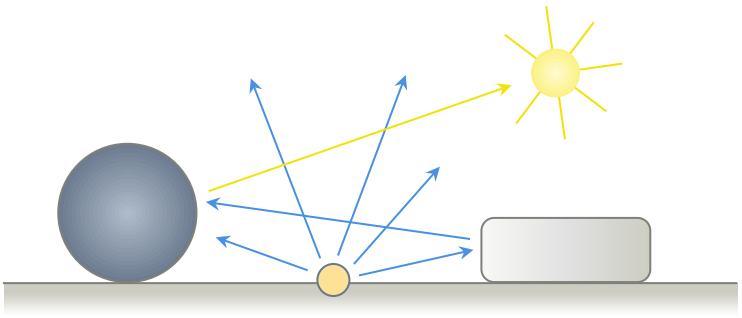


Figura 3.12: Diagrama esquemático del método de trazado de rayos.

3.2.4. Transferencia Radiativa

La radiometría nos da herramientas para medir la luz, pero si queremos entender su interacción con la materia recurriremos a la teoría del transporte radiativo en Chandrasekhar (1960), esta nos ayudará para entender con mayor facilidad el desarrollo de este trabajo.

Definiciones

Intensidad específica. La intensidad específica se define similarmente a la radiancia (3.7), donde Q_v es la cantidad de energía radiante en el rango de frecuencia dv, la cual es transportada a través del área dA y que sale en una dirección confinada al ángulo sólido $d\Omega$ durante un intervalo de tiempo dt.

$$I_v = \frac{dQ_v}{cos\theta \, dA \, dv \, d\Omega \, dt} \tag{3.7}$$

Siendo θ el ángulo de inclinación con respecto a la normal $\vec{n}$ de la superficie A.

La intensidad cambia en el espacio, dirección, tiempo y frecuencia en un medio que absorbe, emite y esparce radiación. I_v puede ser escrita en coordenadas cartesianas

$$I_v = I_v(x, y, z; \alpha, \beta, \gamma; t) \tag{3.8}$$

Donde x, y, z son los ejes de las coordenadas cartesianas y α, β, γ son los cosenos de dirección que definen el punto y la dirección de I_v.

Si el medio con el que vamos a interactuar está estratificado en capas planas paralelas, entonces:

$$I_v = I_v(z, \theta, \phi; t) \tag{3.9}$$

Si I_v es independiente de ϕ, entonces tenemos un campo de radiación con una simetría axial sobre el eje z.

En una simetría esférica , I_v:

$$I_v = I_v(r, \theta; t) \tag{3.10}$$

Donde r es el radio de la esfera y θ es el ángulo entre la dirección del rayo y el vector radio.

Se dice que el campo de radiación es isotrópico en un punto, si la intensidad es independiente de la dirección a la que apunta, entonces:

$$I_v = I_v(r; t) \tag{3.11}$$

Si la intensidad es independiente de las coordenadas espaciales y la dirección, entonces dicho campo es homogéneo e isotrópico.

Si la intensidad I_v es integrada sobre todas las frecuencias, es llamada la *intensidad integrada* y es dada por:

$$I = \int_0^\infty I_v dv \tag{3.12}$$

Para esta definición no se han tomado en cuenta los parámetros que caracterizan el estado de polarización de la intensidad.

Coeficiente de absorción. La intensidad especifica puede ser atenuada debido a la interacción con la materia. Si I_v se convierte en $I_v + dI_v$ después de atravesar un espesor ds en dirección de su propagación entonces:

$$dI_v = -\kappa_v p I_v ds \tag{3.13}$$

Donde p es la densidad de la materia y κ_v es el *coeficiente de absorción de la masa.*

Esta reducción de intensidad no es necesariamente dada solo por la interacción con la materia, ya que puede pasar que la energía perdida de la intensidad incidente puede aparecer en otras direcciones como *radiación dispersada* y que la restante habrá sido "verdaderamente" absorbida. Por lo que debemos diferenciar entre *verdadera absorción* y *dispersión.*

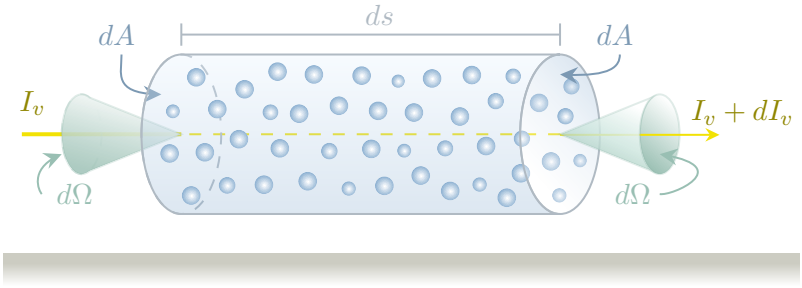


Figura 3.13: Diagrama esquemático de la transferencia radiativa

Dispersión. Diremos que la energía dispersada de I_v en todas las direcciones viene dada por:

$$\kappa_v \, p \, ds \; \times I_v \, cos\theta \, dv \, dA \, d\Omega \tag{3.14}$$

Siendo la masa del elemento:

$$dm = p \, cos\theta \, dA \, ds \tag{3.15}$$

Entonces la energía dispersada puede escribirse como:

$$\kappa_v \, I_v \, dm \, dv \, d\Omega \tag{3.16}$$

Función de fase. Notamos que para formular cuantitativamente el concepto de dispersión, debemos especificar la distribución angular de la *radiación dispersada* (3.16). Por lo tanto introducimos la *función de fase* $\rho(cos\theta)$ tal que

$$\kappa_v \, I_v \, \rho(cos\theta) \, \frac{d\Omega'}{4\pi} dm \, dv \, d\Omega \tag{3.17}$$

da la razón de la energía que está siendo dispersada en un elemento de ángulo sólido $d\Omega'$ y en una dirección inclinada al ángulo θ a la dirección de la intensidad especifica incidente sobre un elemento de masa dm.

De la ecuación (3.17) la razón de la energía incidente perdida debido al dispersión en todas las direcciones es:

$$\kappa_v \, I_v \, dm \, dv \, d\Omega \int_S p(cos\theta) \, \frac{d\Omega'}{4\pi} \tag{3.18}$$

Donde S es el dominio esférico que abarca todo el ángulo sólido. Notamos que (3.18) es igual a (3.16) si:

$$\int_S p(cos\theta) \frac{d\Omega'}{4\pi} = 1 \tag{3.19}$$

i.e., si la función de fase está normalizada a la unidad.

Regresando al caso que donde la verdadera absorción y dispersión se encuentran presentes, la energía dispersada que considera la distribución angular con la función de fase (3.17) será menor que (3.16) si:

$$\int_S p(cos\theta) \frac{d\Omega'}{4\pi} = \varpi_0 \leq 1 \tag{3.20}$$

Por ende el caso general difiere del caso de puro dispersión solo por el hecho de que la función de fase no está normalizada a la unidad. Es evidente de las definiciones dadas que

ϖ_0 representa la fracción de luz perdida debida a la dispersión, mientras que $(1 - \varpi_0)$ representa la fracción restante que ha sido transformada en otras formas de energía.

Debemos referirnos a ϖ_0 como el *albedo para el dispersión simple*. Además, cuando $\varpi_0 = 1$ decimos que tenemos un *caso de conservativo* de *dispersión perfecta*: la dispersión perfecta es una analogía del concepto conservativo en dinámica.

El ejemplo más sencillo de la función fase es

$$\rho(cos\theta) = constante = \varpi_0 \tag{3.21}$$

En este caso la radiación dispersada es isotrópica para cada elemento con masa. Otro caso de mucho interés es el de *la función de fase de Rayleigh*.

$$\rho(cos\theta) = \frac{3}{4}(1 + cos^2\theta) \tag{3.22}$$

Coeficiente de emisión. El *coeficiente de emisión* j_v se define tal que un elemento de masa dm emite en dirección confinada a un elemento de un ángulo solido $d\Omega$ en el intervalo de frecuencia dv y en un tiempo dt, una cantidad de energía radiante:

$$j_v \, dm \, d\Omega \, dv \, dt \tag{3.23}$$

En el caso de un medio que esparce radiación habrá una contribución al coeficiente de emisión de parte de la radiación dispersada de todas las direcciones que van en la dirección de nuestra intensidad especifica.

Sigue que de la ecuación (3.17) que el dispersión de la intensidad especifica desde una dirección (θ', ϕ') contribuye a la intesidad especifica en la dirección (θ, ϕ), siendo la energía:

$$\kappa_v \, dm \, dv \, d\Omega \, \rho(\theta, \phi; \theta', \phi') \, I_v(\theta', \phi') \, \frac{sin\theta' \, d\theta' \, d\phi'}{4\pi} \tag{3.24}$$

donde hemos escrito $p(\theta, \phi; \theta', \phi')$ para denotar la función de fase para el ángulo entre las direcciones especificadas por (θ, ϕ) y (θ', ϕ'). Por eso la contribución, $j_v^{(s)}$, al coeficiente de emisión por dispersión es:

$$j_v^{(s)}(\theta, \phi) = \kappa_v \frac{1}{4\pi} \int_0^{\pi} \int_0^{2\pi} \rho(\theta, \phi; \theta', \phi') \, I_v(\theta', \phi') sin\theta' \, d\theta' \, d\phi' \tag{3.25}$$

Debemos esperar que en general habrá contribuciones al coeficiente de emisión por otras causas que no sea el dispersión. Cuando este no es el caso, podemos decir que tenemos una *atmósfera dispersada*. Es decir

$$j_v \equiv j_v^{(s)} \tag{3.26}$$

Tengamos en cuenta que una atmósfera dispersada no implica el caso de dispersión perfecto.

Un caso que de alguna forma es opuesto al de la atmósfera dispersada es en el que la atmósfera está en *equilibrio termodinámico local*, en el que se asume que las circunstancias son tales que podemos definir en cada punto de la atmósfera una temperatura local T tal que la emisión del coeficiente en ese punto es dado en términos del coeficiente de absorción por la ley de Kirchhoff, i.e., que en cada punto tenemos una relación

$$j_v = \kappa_v B_v(T), \tag{3.27}$$

donde

$$B_v(T) = \frac{2hv^3}{c^2}\frac{1}{e^{hv/kT}-1} \tag{3.28}$$

es la función de Planck (k y h son las constantes de Boltzmann y Planck respectivamente).

La función fuente. La razón de la emisión con respecto al coeficiente de absorción se denota de la siguiente manera:

$$\Im_v = \frac{j_v}{\kappa_v} \tag{3.29}$$

De acuerdo a ecuación (3.25) y ecuación (3.26) para una atmósfera dispersada

$$\Im_v(\theta,\phi) = \frac{1}{4\pi}\int_0^{\pi}\int_0^{2\pi} \rho(\theta,\phi;\theta',\phi')\, I_v(\theta',\phi') sin\theta'\, d\theta' d\phi' \tag{3.30}$$

Mientras que para una atmósfera en equilibrio termodinámico local

$$\Im_v = B_v(T) \tag{3.31}$$

Ecuación de transferencia

Desde la definición de intensidad en la ecuación (3.7), sigue que la diferencia de energía radiante en un intervalo de frecuencia dv que cruzan dos caras normales, en un tiempo dt y confinados al mismo ángulo es:

$$\frac{dI_v}{ds}\, ds\, dv\, dA\, d\Omega\, dt \tag{3.32}$$

La diferencia en energía debe provenir del exceso de emisión sobre el de absorción en intervalo de frecuencia y el ángulo sólido considerado. Entonces la energía absorbida (3.14) es:

$$\kappa_v\, p\, ds \times I_v\, dv\, dA\, d\Omega\, dt \tag{3.33}$$

Mientras que el monto emitido es:

$$j_v\, p\, dA\, ds\, dv\, d\Omega\, dt \tag{3.34}$$

Contando la ganancia y pérdida de la intensidad específica durante su trayecto transversal del cilindro se tiene:

$$\frac{dI_v}{ds} = -\kappa_v p I_v + j_v p \tag{3.35}$$

En términos de la función fuente $\Im$ se puede rescribir la ecuación (3.35) de la forma:

$$-\frac{1}{\kappa_v p}\frac{dI_v}{ds} = I_v - \Im_v \tag{3.36}$$

Siendo está la *ecuación de transferencia radiativa* RTE.

En una atmósfera dispersada. Para una atmósfera dispersada y una atmósfera en equilibrio termodinámico las funciones fuente vienen dadas por las ecuaciones (3.30) y (3.31).

En un sistema de coordenadas cartesianas la ecuación de transferencia puede ser escrita como

$$-\frac{1}{\kappa_v p}\left(l\frac{\partial}{\partial x}+m\frac{\partial}{\partial y}+n\frac{\partial}{\partial z}\right)I_v(x,y,z;l,m,n) = I_v(x,y,z;l,m,n)-\Im_v(x,y,z;l,m,n) \tag{3.37}$$

Siendo la función fuente funcionalmente dependiente de la intensidad en un punto, la ecuación de transferencia es generalmente una *ecuación integro diferencial*.

Una solución formal de la ecuación de transferencia radiativa

Por conveniencia removeremos el sufijo v que representa la frecuencia especifica.

$$-\frac{1}{kp}\frac{dI}{ds} = I-\Im \tag{3.38}$$

Multiplicando a ambos lados de (3.38) por $e^{\kappa ps}$:

$$\begin{aligned} -e^{\kappa ps}\frac{1}{\kappa p}\frac{dI(s)}{ds} &= e^{\kappa ps}(I(s)-\Im(s)) \\ e^{\kappa ps}\frac{dI(s)}{ds}+kpe^{\kappa ps}I(s) &= kpe^{\kappa ps}\Im(s) \\ \frac{d}{ds}[e^{\kappa ps}I(s)] &= kpe^{\kappa ps}\Im(s) \end{aligned} \tag{3.39}$$

Integramos de 0 a s y hacemos un cambio de notación en el lado derecho de la ecuación para diferenciar del punto s y del diferencial ds'

$$\begin{aligned} [e^{\kappa ps}I(s)]_0^s &= \int_0^s e^{\kappa ps'}\Im(s')kp\,ds' \\ e^{\kappa ps}I(s) &= I(0)+\int_0^s e^{\kappa ps'}\Im(s')kp\,ds' \\ I(s) &= e^{-\kappa ps}\,I(0)+\int_0^s e^{kp(s'-s)}\Im(s')kp\,ds' \end{aligned}$$

Esta es una solución a la ecuación de transferencia.

$$I(s) = I(0)e^{-\tau(s,0)}+\int_0^s \Im(s')e^{-\tau(s,s')}kp\,ds' \tag{3.40}$$

donde $\tau(s,s')$ es el espesor óptico del material entre los puntos s y s':

$$\tau(s,s') = \int_{s'}^{s} kp\,ds \tag{3.41}$$

El significado físico de la ecuación (3.40) expresa el hecho de que la intensidad en cualquier punto y en una dirección dada resulta de la emisión de todos los puntos anteriores a s', reducidos por el factor $e^{\tau(s,s')}$ que es la absorción de la energía por la materia con la que interactúa.

3.3. Trabajo Relacionado

En el apartado 3.2.3 el comportamiento de la luz es descrito por la ecuación de renderización, pero esta solo describe su comportamiento al interactuar con el límite de la materia, i.e., su interacción con la superficie de los objetos en la escena. Sin embargo, el mundo real no es tan sencillo, pues la luz puede interactuar con un medio, el cual atraviesa, antes de llegar a la superficie de un objeto.

Los elementos en la escena que interactúan con la luz, suelen ser llamados *medios no participativos* en los que la interacción solo ocurre en la superficie y *medios participativos* donde la interacción ocurre en todo su *volumen*.

En este capítulo revisaremos la literatura con respecto a este problema de interacción con medios participativos, que un en una escena virtual, pueden ser llamados *volúmenes*.

3.3.1. Volúmenes

Son los objetos en la escena que actúan como medios participativos, ya que la luz al atravesarlos se somete a tres tipos de fenómenos que son la absorción, dispersión y emisión. Además, en este tipo de trabajos, suelen ser descritos como una colección de partículas de cualquier tamaño en cual se toma en cuenta la densidad del mismo, ya que su tamaño debería ser despreciable comparándolo con el promedio de la distancia entre partículas.

La creación de estos modelos en algunos casos vienen dadas por simulaciones fluidos (e.g. la formación de las nubes, el humo) que describen su comportamiento como partículas a través del tiempo, que luego son representados en estructuras de datos bastantes complejas que permiten la definición de las siguientes propiedades.

Propiedades

Consideremos un fotón que viaja a través de un volumen (humo, niebla, nube), este puede colisionar con las partículas que forman el volumen y estas colisiones definen la distribución de radiancia a través del volumen, pero como no es factible simular cada una de las partículas en un volumen, estas colisiones se manejaran como probabilidades.

La probabilidad de que un fotón colisione es definida por el coeficiente $\sigma(x)$, que es la densidad de probabilidad de colisión por unidad de distancia recorrida dentro del volumen.

Absorción. Se expresada como σ_a y describe la colisión cuando un fotón es absorbido por el volumen. Físicamente, decimos que la energía del fotón fue transferida a la energía de la partícula del volumen en cuestión, o de un forma más general, se convirtió en calor.

Dispersión. Se expresada como σ_s, dispersa el fotón en una dirección definida por la función de fase. El valor de la radiancia que lleva el fotón no cambia con este coeficiente, cualquier cambio de radiancia que lleva el fotón se modela con la absorción y emisión. Pero es importante notar que afecta a la contribución total de la radiancia computada.

Función de fase. Expresada como $\rho(x, \vec{\omega}, \vec{\omega}')$, es la distribución angular de la radiancia esparcida y es usualmente modelada como una función unidimensional del ángulo θ entre las dos direcciones $\vec{\omega}$ y $\vec{\omega}'$.

Emisión. Los volúmenes emiten radiancia de forma una forma única, podríamos decir que se comportan de la misma forma que una fuente de luz. La radiancia emitida es expresada como $L_e(x, \vec{\omega})$, esta puede ser absorbida o esparcida como lo haría cualquier radiancia proveniente de una fuente de luz no volumétrica. En el mundo real, los volúmenes no emiten radiancia direccionalmente y $L_e(x, \vec{\omega})$ es la misma en todas las direcciones.

Extinción. Existen casos, en los cuales es mejor tener una parametrización diferente a la de coeficiente de absorción y esparcimiento. Podemos definir al coeficiente de extinción como $\sigma_t = \sigma_a + \sigma_s$ (Algunas veces llamado atenuación) como la suma de los coeficientes de absorción y dispersión.

La extinción define la radiancia total perdida por la absorción y el esparcimiento en un solo coeficiente, pero se debe asegurar la inclusión de la radiancia dispersada está modelada en la formulación integral en el caso de la albedo para dispersión simple.

Homogeneidad y heterogeneidad

Es importante notar que la distribución de partículas en un medio participativo puede variar a lo largo del espacio. Por ejemplo la interacción de luz en la atmósfera es muy diferente a la interacción de la luz en una nube. Dependiendo de que tan uniforme sea un medio podremos distinguirlos, de forma que un *medio homogéneo* es aquel en que la luz interactúa de manera uniforme en todo su volumen, pero uno *medio heterogéneo* tendrá alguna dependencia espacial y se comportará de forma diferente en todos los puntos dentro de su volumen.

3.3.2. Propagación de la luz

La distribución de radiancia en volúmenes es definida por la RTE descrita en apartado 3.2.4, esta describe el equilibrio de la intensidad específica, i.e., radiancia, de modo que se procederá a describir la RTE en términos de radiancia con sus correspondientes coeficientes. Esta interpretación es descrita en Fong y col. (2017).

Absorción

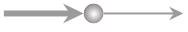


Figura 3.14: Diagrama esquemático de la absorción

$$-\sigma_a(x)\, L(x, \vec{\omega}) \tag{3.42}$$

Dispersión saliente

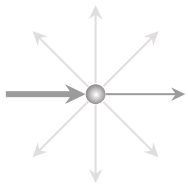


Figura 3.15: Diagrama esquemático de la dispersión saliente

Es importante notar que la radiancia $L(x, \vec{\omega})$ pierde valor debido a la interacción de los fotones en el medio, i.e., la dispersión de aquellos que se van una dirección diferente a la de $\vec{\omega}$ y se esta se calcula como:

$$-\sigma_s(x)\, L(x, \vec{\omega}) \tag{3.43}$$

Emisión

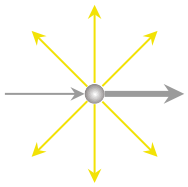


Figura 3.16: Diagrama esquemático de la emisión

$$\sigma_a(x)\, L_e(x, \vec{\omega}) \tag{3.44}$$

Dispersión entrante

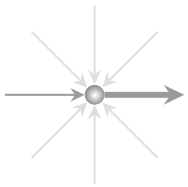


Figura 3.17: Diagrama esquemático de la dispersión entrante

Es la contribución de la dispersión saliente de todas las otras direcciones $\vec{\omega}$

$$\sigma_s(x) \int_S \rho(x, \vec{\omega}, \vec{\omega}')\, L(x, \vec{\omega}')\, d\vec{\omega}' \tag{3.45}$$

Donde S es el dominio esférico al rededor del punto x.

Formando la RTE

El equilibrio de la transferencia radiativa en términos de radiancia viene dada por ecuaciones (3.42) a (3.45)

$$\begin{aligned}\frac{d}{dx}L(x,\vec{\omega}) = &-\sigma_t(x)\,L(x,\vec{\omega}) \\ &+\sigma_a(x)\,L_e(x,\vec{\omega}) \\ &+\sigma_s(x)\int_S \rho(x,\vec{\omega},\vec{\omega}')\,L(x,\vec{\omega}')\,d\vec{\omega}'\end{aligned} \tag{3.46}$$

Donde $\sigma_t = \sigma_a + \sigma_s$.

3.3.3. Ecuación de renderización para volúmenes

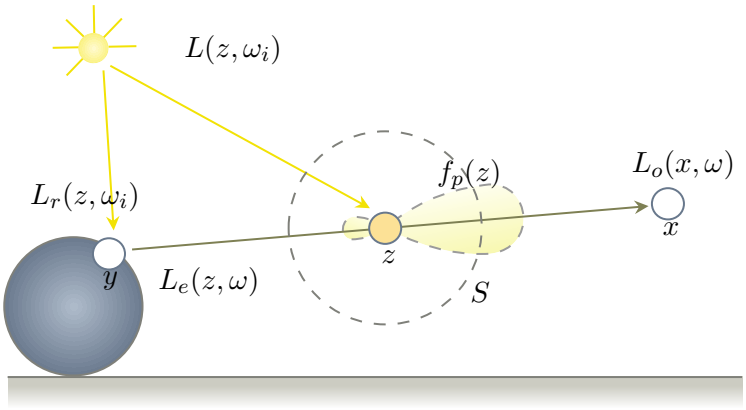


Figura 3.18: Diagrama esquemático de VRE

Esta ecuación viene de integrar ecuación (3.46), que expresa lo mismo que en la apartado 3.2.4. La radiancia de la superficie atenuada L_{at} desde una superficie en y y la radiancia dispersada L_{dis} acumulada en el medio entre x e y.

$$L_o(x,\vec{\omega}) = L_{at}(x,\vec{\omega}) + L_{dis}(x,\vec{\omega}) \tag{3.47}$$

Siendo la radiancia atenuada L_{at} y la dispersada L_{dis}

$$L_{at}(x,\vec{\omega}) = T(d)L_r(x,\vec{\omega}) \tag{3.48}$$

$$L_{dis}(x,\vec{\omega}) = \int_0^d T(s)\left[\sigma_a L_e(z,\vec{\omega}) + \sigma_s\int_S \rho(z,\vec{\omega},\vec{\omega}')\,L(z,\vec{\omega}')\,d\vec{\omega}'\right]ds \tag{3.49}$$

Donde $d = ||x-y||$ es la distancia, $z = x + d\cdot\vec{\omega}$ un punto en el espacio y la transmitancia T definida como:

$$T(s) = e^{-\int_0^s \sigma_t(z)ds'} \tag{3.50}$$

3.3.4. Métodos para resolver la VRE

Simular la interacción de la luz en medios participativos ha sido estudiado extensivamente. Muchas de las técnicas utilizadas hasta el día de hoy han sido desarrolladas para estudiar la transferencia radiativa de calor, J. (1954) y Chandrasekhar (1960).

Cerezo y col. (2005) presenta una vista general sobre las técnicas para resolver la *VTE* y Fong y col. (2017) presenta el avance actual de estás técnicas utilizadas en sistemas de grandes industrias del entretenimiento.

Métodos analíticos

Blinn (1982) una de los primeras publicaciones en interesarse en la visualización de los medios participativos, en especial los anillos de Saturno y la atmósfera, resolvió analíticamente la RTE para el caso de dispersión simple.

Métodos deterministas

Basados en soluciones numéricas, varios de estos métodos son extensiones de los métodos radiosidad para tratar la dispersión isotrópica volumétrica, en Rushmeier y Torrance (1987) utiliza un método el cual dicretiza la VRE y la resuelve sobre una colección finita de elementos. La mayoría de estos métodos en el principio eran imprácticos por la complejidad computacional y los costos de almacenamiento tan altos que tenían para las más simples escenas.

Si bien los métodos de radiosidad computan la solución para todo el espacio de la escena, tenemos que tener en cuenta las desventajas de estos métodos radican en la interacción de las superficies con los medios, pues una escena con bastantes objetos con geometrías complejas no representaría bien el transporte de la luz.

Métodos estocásticos

Basados en el muestreo aleatorio, la mayoría de técnicas actuales se basan en el *sampling* y la integración de Monte Carlo. En Hanrahan y Krueger (1993) y Pattanaik y Mudur (1993) se propusieron métodos para extender el algoritmo de trazado de rayos para simular la dispersión en medios participativos. Los métodos basados en la integración de Monte Carlo son interesantes por que son usualmente imparciales y garantizan una convergencia a la solución real, sin embargo estos sufren de bastante al generar la imagen y solo puede mejorar esto con un mayor esfuerzo computacional.

3.4. Metodología

Como se ha mencionado en el apartado 3.3.4, existen diferentes métodos para aproximarse a este problema, en este trabajo se ha escogido un método estocástico para aproximarnos a la VRE, el cual hace uso de la técnica de integración de Monte Carlo, pues es una de las más predominantes en la literatura actual. Debido a la complejidad de implementar un sistema que considere todo lo explicado en explicado en apéndice A, se ha decido utilizar un sistema de renderización con un algoritmo de trazado de rayos que abarca la RTE, i.e., soluciona parcialmente la VRE, puesto que existen diferentes interacciones que simular dependiendo del medio con el que trabajemos, además la brevedad en la que se desarrollo este trabajo es nuestra principal razón por la cual escogimos un sistema de renderización.

3.4.1. Eligiendo un sistema de renderización

En este trabajo se ha elegido *Mitsuba 2* por Nimier-David y col. (2019) por la facilidad de implementar algoritmos bajo su sistema de complementos, lo que nos permite enfocarnos solamente en el analisis de un Path tracer para volúmenes, ya que el desarrollo completo de este tipo de software es bastante complejo.

Mitsuba 2

Mitsuba 2 es un sistema de renderizado orientado a la investigación, escrito en *C++ 17 portable* usando la librería *Enoki*. Es desarrollado por la *Realistic Graphics Lab* en *EPFL*.

Este consiste de un pequeño grupo de librerías principales y complementos que implementan toda la funcionalidad de este tipo de software, que van desde los materiales, las fuentes de luz hasta los algoritmos de renderización.

3.4.2. Path tracer

Path tracer es un método de Monte Carlo para solucionar la RE, debido a la sencillez y arbitrariedad del algoritmo es usado como la base para la implementación de los sistemas de renderización, que luego se enfocaran en interpretar fenómenos ópticos que son dependientes de la naturaleza de la luz, ya sea como onda, onda electromagnética o fotón. Existe mucha literatura al respecto de este tipo de algoritmo en los cuales se basan en técnicas para la reducción de varianza generada por la generación aleatoria de las muestras direccionales al integrar la RE. En este trabajo se utilizará una versión expandida de este algoritmo que considera la RTE, i.e., la VRE.

Algoritmo

Como *Mitsuba 2* incluye una serie de algoritmos implementados, los cuales se usarán como referencia para familiarizarnos con el programa. También se ha hecho uso de lo descrito en Pharr y col. (2016) para conocer a un poco la implementación de este programa.

Primero veremos el pseudocódigo del componente de integración del algoritmo de ilumina-ción directa en *Mitsuba 2* basado en un Path tracer, para luego solo mostrar la integración de la RTE.

Direct

Algoritmo 1: Una perspectiva general de **direct.cpp**

Datos: $\vec{v}$ rayo, Ω escena
Resultado: L espectro de luz computado
$L \leftarrow 0$
$SI \leftarrow$ interacciones de Ω con $\vec{v}$

`/* Interacción directa con la fuente de luz */`
si SI *interactúa con* fuentes de luz **entonces**
  $L \leftarrow L +$ contribución de la fuente de luz SI
fin
si SI *no interactúa con otro objeto* **entonces**
  devolver L
fin

`/* Sampling de las fuentes de luz */`
para $i \leftarrow 0$ **a** *cantidad de fuentes de luz en* Ω **hacer**
  $\vec{d}, l \leftarrow$ contribución de la fuente de luz i de Ω
  $p_{val} \leftarrow$ contribución de la $BSDF$ en SI por $\vec{d}$
  $p_{pdf} \leftarrow$ *pdf* de la $BSDF$ en SI por $\vec{d}$
  $mis \leftarrow$ peso de la contribución de l usando p_{pdf}
  $L \leftarrow L + mis \times p_{val} \times l$
fin

`/* Sampling de la` $BSDF$ `*/`
para $i \leftarrow 0$ **a** *cantidad de muestras de la* $BSDF$ Ω **hacer**
  $\vec{d}, l \leftarrow$ contribución de la $BSDF$
  $SI_p \leftarrow$ interacción de $\vec{d}$ en Ω
  $e \leftarrow$ solo rayos que interactúan con una fuente de luz
  $e_{val} \leftarrow$ contribución de la fuente de luz SI_p
  $e_{pdf} \leftarrow$ *pdf* de la interacción SI con $\vec{d}$
  $L \leftarrow L + l \times e_{val} \times$ peso y MIS de la *bsdf*
fin

3.4.3. Volume path tracer

En esta sección solo se mencionará la parte de la integración de la RTE que se encuentra en **volpath.cpp**.

Algoritmo 2: Muestrando la RTE

Datos: $\vec{v}$ rayo, Ω escena
Resultado: L espectro de luz computado
$L \leftarrow 0$
$T \leftarrow 0$
$NI \leftarrow True$
$MI \leftarrow$ interacciones de Ω con $\vec{v}$

si *existe* MI **entonces**
  si *existe dispersión* **entonces**
    $T \leftarrow T * \frac{\sigma_s}{\sigma_t}$ de MI
  fin
  $\rho \leftarrow$ función de fase del medio en MI
  `/* Sampling de las fuentes de luz */`
  $e \leftarrow$ solo rayos que interactúan con una fuente de luz
  $\rho_{val} \leftarrow$ evaluando la ρ para e
  $L \leftarrow L * T * \rho_{val} * e$
  `/* Sampling de las función de fase */`
  $\rho_{pdf} \leftarrow$ evaluando ρ
  $NI \leftarrow$ define en base a lo obtenido en ρ_{pdf} si esta interacción requerirá interacción con una superficie
fin

Cuando se hace uso del muestreo de la función de fase como el muestreo de las fuentes de luz, se utiliza la integración de Monte Carlo para evaluar estos valores, los cuales son utilizados para actualizar la radiancia computada o tomar decisiones para manejar las interacciones con las superficies, lo que hace algoritmo 1.

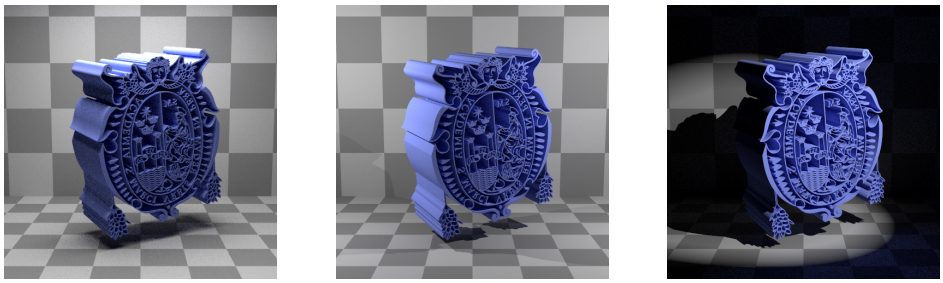
3.5. Resultados

Para analizar los resultados obtenidos, en general se hace una comparación visual de los resultados generados por otro algoritmo de trazado de rayos más sencillo, pero el cual ha sido evaluado por una mayor cantidad de tiempo. En nuestro caso por el algoritmo de la VRE con MIS, implementado en *Mitsuba 2*, ya que no disponemos de una imagen de referencia con la que podamos comparar que el fenómeno físico ha sido simulado correctamente.

Varios de los objetos, a excepción de los volúmenes, han sido creados en *Blender*. El hardware sobre el cual se han computado estos resultados es una computadora de escritorio con un procesador AMD Ryze 9 3900X 12-Core 3.79GHz, 32Gb RAM en Ubuntu 20.04 LTS, la versión compilada de *Mitsuba 2* variante con el *computational backend scalar* y representación de color *RGB*, puesto que la tarjeta gráfica que dispongo en el momento no es compatible para la renderización con *GPU*. Además, se ha utilizado una escena simple con diferentes fuentes de luz, para comparar resultados ver figura 3.19 .

Como se esta utilizando una versión compilada de *Mitsuba* se ha utilizado un script en python para comparar los resultados revisar apéndice C.

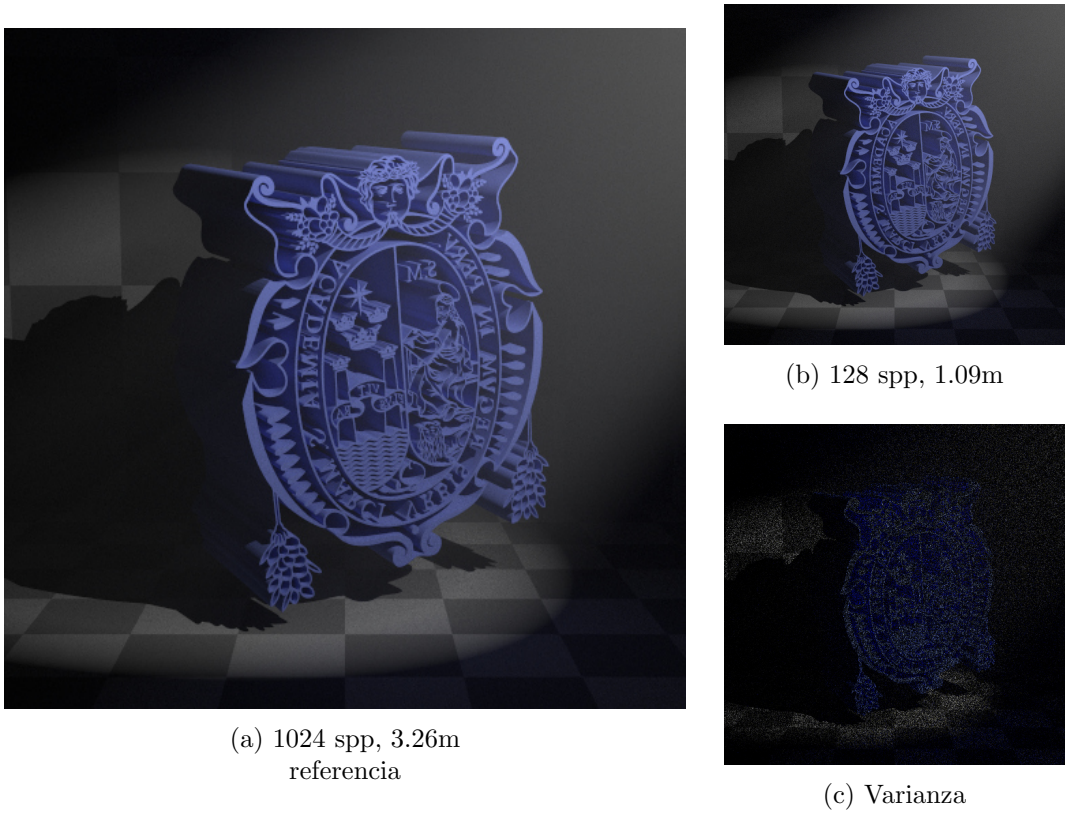
Se puede apreciar en la figura 3.20, la contribución de un medio homogéneo, el resultado obtenido muestra la contribución de la luz en el espacio, algo que no se puede apreciar



(a) *Area*, 14.9s (b) *Point*, 13.7s (c) *Spot*, 15s

Figura 3.19: Path tracer, 500 × 500 64 spp, 151538 *primitives*, 5 *depth*

en figura 3.19c y como esta también atenúa lo que impacta sobre la superficie del objeto hasta llegar a la cámara.



(a) 1024 spp, 3.26m referencia

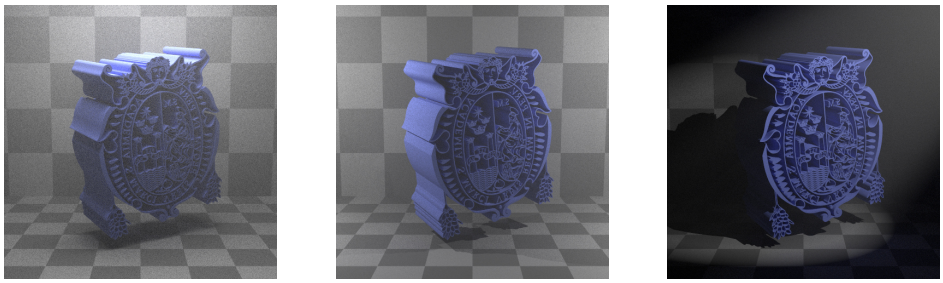
(b) 128 spp, 1.09m

(c) Varianza

Figura 3.20: Spot, Medio homogéneo

Usualmente el tipo de análisis para estos resultados viene dado por la varianza, pero un análisis complejo de la varianza es dependiente de las técnicas de reducción de varianza para el modelo en cuestión, donde se considera que se debe computar o no, sin embargo solo haremos un análisis visual de nuestros resultados.

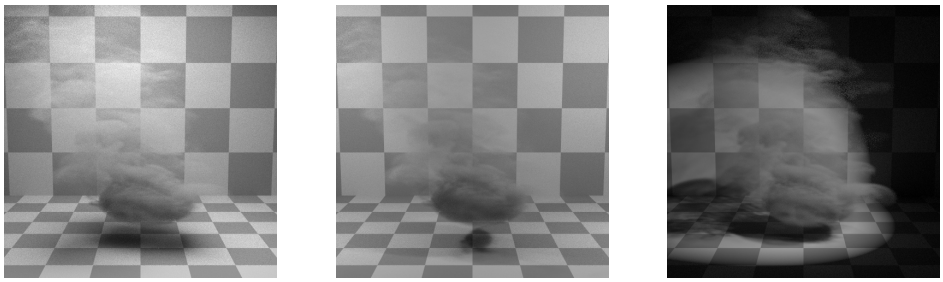
Los resultados de la figura 3.21, muestran la contribución del luz en todo el medio, para las tres fuentes de luz donde se puede observar claramente la atenuación de la luz reflejada en la superficie del logo de San Marcos. Las modelos utilizados a continuación son parte



(a) *Area* (b) *Point* (c) *Spot*

Figura 3.21: Medio homogéneo, 500x500, 256spp

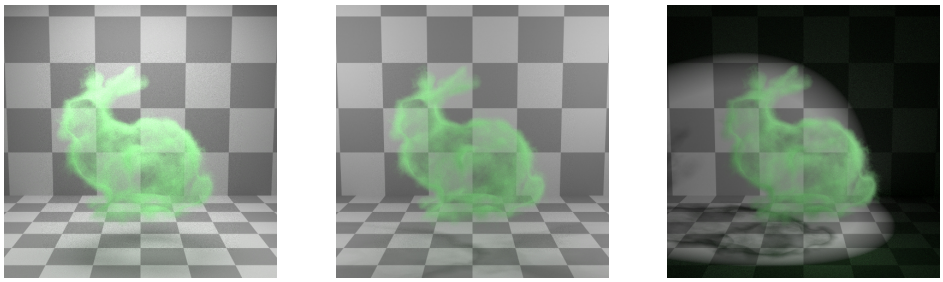
de los recursos de descarga de *OpenVDB* que han sido procesados para que puedan ser utilizados en *Mitsuba*.



(a) *Area* (b) *Point* (c) *Spot*

Figura 3.22: Humo, 500x500, 256 spp

En la figura 3.22, podemos observar la contribución dentro del volumen y como esta llega a generar sombra en algunos casos, puesto que esta atenúa la contribución directa de las fuentes de luz.



(a) *Area* (b) *Point* (c) *Spot*

Figura 3.23: Conejo, 500x500, 256spp, albedo $(0{,}5, 1, 05)$

Finalmente, la figura 3.23 muestra que el Albedo es el encargado de la dispersión de la luz, para este caso podemos observar un color verde debido a la definición del albedo ejecutar el programa.

Todas las escenas han sido evaluadas considerando un medio homogéneo, puesto que existe una complejidad aún más grande para representar fenómenos ópticos en medios heterogéneos, tanto física como matemática y computacionalmente, puesto por el corto tiempo

que hemos tenido solo hemos tratado de mostrar la interacción más básica de los medios participativos en un algoritmo de trazado de rayos.

Capítulo IV

Conclusiones

Hemos visto que modificar un algoritmo de trazado de rayos para que interactúe con medios participativos genera resultados suficientemente realistas, pero podemos decir que no son físicamente correctos. Esto es debido a la interpretación que realizamos para poder computar este fenómeno.

Las técnicas para reducir la varianzia son muy influyentes en el resultado obtenido, por ende es importante conocer su descripción matemática y su implementación en el programa, pues computar en la dirección correcta nos ayudaría a obtener una imagen sin mucho ruido.

Es importante mencionar que existen muchas formas de abarcar esta interacción para obtener un resultado más preciso o más rápido. Diferentes sistemas utilizan implementaciones más optimizadas, ya que el proceso de transferencia de luz es uno infinitamente recursivo, y se tienen que tomar decisiones para detener este proceso y lo computado realmente contribuya al resultado.

Si bien no se ha desarrollado nada innovador en este trabajo debido a la brevedad de este trabajo, es satisfactorio comprender este fenómeno físico, su descripción matemática y las propuestas para simular este comportamiento computacional mente.

Capítulo V

Recomendaciones

Si bien hemos visto como implementar la interacción de la luz como medios participativos, no hemos hablado de la formación de estos, puesto que conocer su formación podría ayudarnos a implementar optimizaciones para reducir los tiempos de renderización, ya que estos suelen ser representados en diferentes estructuras de datos bastante complejas, que facilitan el acceso a los datos encontrado en ellos.

Bibliografía

Metropolis, N. & Ulam, S. (1949). The monte carlo method. *Journal of the American statistical association*, *44*(247), 335-341 (vid. pág. 43).

J., H. S. (1954). *Science Progress (1933-)*, *42*(165), 128-129. http://www.jstor.org/stable/43415215 (vid. pág. 25)

Chandrasekhar, S. (1960). *Radiative Transfer*. Dover Publications, INC. (Vid. págs. 6, 15, 25).

Blinn, J. F. (1982). Light Reflection Functions for Simulation of Clouds and Dusty Surfaces. *SIGGRAPH Comput. Graph.*, *16*(3), 21-29. https://doi.org/10.1145/965145.801255 (vid. pág. 25)

Goral, C. M., Torrance, K. E., Greenberg, D. P. & Battaile, B. (1984). Modeling the Interaction of Light between Diffuse Surfaces. *SIGGRAPH Comput. Graph.*, *18*(3), 213-222. https://doi.org/10.1145/964965.808601 (vid. pág. 14)

Kajiya, J. T. (1986). The Rendering Equation. *ACM SIGGRAPH Computer Graphics*, *20*(4), 143-150. https://doi.org/10.1145/15886.15902 (vid. págs. 6, 13, 15)

Rushmeier, H. E. & Torrance, K. E. (1987). The Zonal Method for Calculating Light Intensities in the Presence of a Participating Medium. *SIGGRAPH Comput. Graph.*, *21*(4), 293-302. https://doi.org/10.1145/37402.37436 (vid. pág. 25)

Glassner, A. S. (1989). *An introduction to ray tracing*. Morgan Kaufmann. (Vid. pág. 41).

Hanrahan, P. & Krueger, W. (1993). Reflection from Layered Surfaces Due to Subsurface Scattering. *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*, 165-174. https://doi.org/10.1145/166117.166139 (vid. pág. 25)

Pattanaik, S. & Mudur, S. (1993). Computation of global illumination in a participating medium by monte carlo simulation. *Comput. Animat. Virtual Worlds*, *4*, 133-152 (vid. pág. 25).

Foley, J. D., Van, F. D., Van Dam, A., Feiner, S. K., Hughes, J. F. & Hughes, J. (1996). *Computer graphics: principles and practice* (Vol. 12110). Addison-Wesley Professional. (Vid. pág. 41).

Cerezo, E., Pérez, F., Pueyo, X., Seron, F. J. & Sillion, F. X. (2005). A survey on participating media rendering techniques. *The Visual Computer*, *21*(5), 303-328. https://doi.org/10.1007/s00371-005-0287-1 (vid. pág. 25)

Dulyan, A. (2005). Hyalite fluorescence. https://commons.wikimedia.org/wiki/File:Hyalite_fluorescence.jpg. (Vid. pág. 10)

Bahaa E. A. Saleh, M. C. T. (2007). *Fundamentals of Photonics*. Wiley. (Vid. pág. 9).

Kuiper, P. (2008). LCD Polarization. https://commons.wikimedia.org/wiki/File:LCD_polarization.JPG. (Vid. pág. 9)

Michicgn. (2008). Phosphoreszierende Pigmente (ZnS). https://commons.wikimedia.org/wiki/File:Phosphoreszenz.jpg. (Vid. pág. 10)

Sauerbronn, L. (2008). Fresnel single slit diffraction experiment. https://commons.wikimedia.org/wiki/File:Fresnel_Diffraction_experiment_DSC04570.JPG. (Vid. pág. 9)

Dunn, W. L. & Shultis, J. K. (2011). *Exploring monte carlo methods*. Elsevier. (Vid. pág. 43).

Ritschel, T., Dachsbacher, C., Grosch, T. & Kautz, J. (2012). The State of the Art in Interactive Global Illumination. *ACM SIGGRAPH Computer Graphics*, *31*(1), 160-188. https://doi.org/gfzwbm (vid. pág. 14)

Ma, K. (2013). Prism outdoor. https://commons.wikimedia.org/wiki/File:Prism_outdoor.jpg. (Vid. pág. 9)

McCluney, W. R. (2014). *Introduction to radiometry and photometry*. Artech House. (Vid. pág. 12).

Pharr, M., Jakob, W. & Humphreys, G. (2016). *Physically Based Rendering: From Theory to Implementation* (3rd). Morgan Kaufmann Publishers Inc. (Vid. págs. 10, 26, 41, 43).

Fong, J., Kulla, M. W. C. & Habel, R. (2017). Production volume rendering: SIGGRAPH 2017 course. *ACM SIGGRAPH Computer Graphics*, *2*. https://doi.org/10.1145/3084873.3084907 (vid. págs. 22, 25)

Akenine-Möller, T., Haines, E., Hoffman, N., Pesce, A., Iwanicki, M. & Hillaire, S. (2018). *Real-Time Rendering 4th Edition*. A K Peters/CRC Press. (Vid. pág. 13).

Community, B. O. (2018). *Blender - a 3D modelling and rendering package*. Blender Foundation. Stichting Blender Foundation, Amsterdam. http://www.blender.org. (Vid. pág. 41)

Nimier-David, M., Vicini, D., Zeltner, T. & Jakob, W. (2019). Mitsuba 2: A Retargetable Forward and Inverse Renderer. *ACM Trans. Graph.*, *38*(6). https://doi.org/10.1145/3355089.3356498 (vid. págs. 13, 26, 41)

Raspopova, M. (2020). Burbuja. https://unsplash.com/photos/HvkmgKaw1R4. (Vid. pág. 9)

Lista de Símbolos

Muchos los símbolos en este trabajo tienen algún sufijo que explican una dependencia con respecto a una variable, para mantener una lectura uniforme con respecto a la literatura en este tipo de trabajos, se han mantenido sus sufijos en inglés.

Símbolo	**Descripción**
Q	Energía
t	Tiempo
Φ	Flujo de energía
A	Área
v	Frecuencia de onda electromagnética[1]
x	Punto en el espacio
$\vec{\omega}$	Vector dirección[2]
$\vec{n}$	Vector normal a una superficie
Ω	Ángulo sólido
θ	Ángulo
S	Dominio esférico
H	Dominio hemisférico
L	Radiancia
L_o	Radiancia saliente
L_e	Radiancia emitida por una superficie o medio
L_r	Radiancia reflejada por una superficie
L_i	Radiancia incidente por dispersión en un medio
ρ	Función de fase
f_r	Bidirectional Reflectance Distribution Function
σ_a	Coeficiente de absorción
σ_s	Coeficiente de dispersión
σ_t	Coeficiente de extinción
Q_v	Energía de una frecuencia específica
I_v	Intensidad específica
κ	Coeficiente de absorción de masa
p	Coeficiente de densidad de la materia
s	Espesor de un medio
m	Masa de un elemento

[1] Los símbolos con el sufijo v denotan que se trabaja una frecuencia específica.

[2] En este tipo de trabajos se suele asociar ω con un vector dirección normalizado a la unidad que está confiado a un ángulo sólido, así que una notación correcta sería $\vec{\omega}_\Omega$, sin embargo en renderización solo se suele usar la dirección.

Símbolo	**Descripción**
ϖ_0	Albedo para dispersión simple o *single scattering*
j	Coeficiente de emisión
$\Im$	Función fuente
τ	Función para promediar el espesor óptico del material
T	Transmitancia

Terminología

Debido a que muchos de los términos utilizados en este trabajo provienen del inglés y varios de estos carecen de una traducción contextual, he decidido dejarlos como tal para que puedan ser fácilmente relacionados con los que se encuentran en la literatura actual.

Término	**Descripción**
2D	Bidimensional
3D	Tridimensional
Albedo	Representa la reflectividad de la luz al interactuar con un particula en un medio participativo.
BRDF	*Bidirectional Reflectance Distribution Function*
Direct ilumination	*Iluminación directa*, algoritmo que computa solo la contribución de luz que impacta sobre una superficie, sin considerar la contribución del reflejo de la misma sobre otros objetos en la escena.
GI	*Global Ilumination*, conjunto de algoritmos que computan la contribución de luz sobre todos los objetos en la escena.
MIS	*Multiple Importance Sampling*, técnica para minimizar la variancia de los métodos de Monte Carlo.
Monte Carlo	Metodología que usa el muestro para estimar el promedio de un resultado.
Path tracer	*Trazador de caminos*, es un algoritmo de GI basado en las técnicas de Monte Carlo para computar la contribución de luz que impacta sobre una superficie considerando la contribución de los objetos que lo rodean.
Radiosidad	Conjunto de algoritmos que simula distribución de luz en un espacio discretizado.
Ray casting	*Casteo de rayos*, se refiere a la interacción de un rayo un objeto, es un algoritmo para
Ray tracing	*Trazado de rayos*, algoritmo para computar la ecuación de la renderización usando métodos de Monte Carlo.
RE	*Rendering Equation*, ecuación de la renderización introducida por Kajija en su publicación original titulada de la misma forma.

Término	**Descripción**
RTE	*Radiative Transfer Equation*, ecuación diferencial de la transferencia radiativa que modela el intercambio de energía en un medio participativo.
spp	*Samples Per Pixel*, muestras direccionales por pixel.
VRE	*Volume Rendering Equation*, ecuación de la renderización para volumenes en este caso se considera la contribución de los medios participativos para el computo de radiancia.

Apéndice A

Renderización

La renderización o sintetización de imágenes es el proceso de generar imágenes a partir de la descripción de una escena virtual. Existen diferentes razones para querer renderizar una escena virtual y dependiendo de estas, la arquitectura del programa que renderizará esta imagen cambiará, ya que ha veces queremos generar dichas imágenes de manera casi inmediata ,e.g., renderización en tiempo real como utilizan los videojuegos; o obtener una imagen casi indistinguible de una foto real, e.g., renderización basada en físicas, como lo que vemos en las películas; o tal vez quisiéramos darle otra perspectiva a nuestra vida, e.g., renderización no realista.

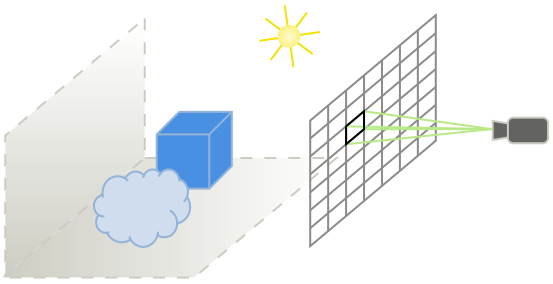


Figura 1.1: Descripción de una escena virtual.

1.1. Renderización basada en físicas

Para obtener una imagen fotorrealista de una escena virtual es necesario interpretar nuestro entorno y como funciona físicamente para que nuestro programa genere dicho resultado. Casi todos los programas renderización basada en físicas están basados en los algoritmos de Ray tracing. Este algoritmo es muy simple, se basa en seguir el camino de un rayo de luz a través de una escena mientras interactúa y se refleja con los objetos en su entorno. Existen muchas formas de escribir un Ray tracing, pero casi todos algunos de los objetos y fenómenos:

- Cámaras
- Fuentes de Luz
- Intersecciones de los rayos de luz y los objetos

- Dispersión de luz en la superficie
- Transporte de luz indirecta
- Propagación de rayos

Existe mucha literatura respecto al tema, Glassner (1989) es uno de los primeros libros en hablar sobre Ray tracing. Foley y col. (1996) presenta una vista general sobre lo que hay que tener en cuenta al renderizar, cosas como:

- Percepción visual humana
- Geometría 2D y 3D, y sus transformaciones
- Representación y manipulación de imágenes
- Ray casting y las rasterización
- Texturas y su mapeo
- Transporte de luz y su dispersión
- Meshes
- Shaders
- Estructuras de datos
- Hardware para gráficos

1.1.1. Sistemas de Renderización

Un sistema de renderización o motor de render es un programa especializado que sintetiza una imagen 2D a partir de la información de una escena virtual, dicha escena contiene elementos que la describen como: geometrías, texturas, fuentes de luz, etc. Actualmente existen diversos sistemas basados en físicas siendo algunos propietarios como: Octane-Render desarrollado por Otoy, Renderman por Pixar, Hyperion por Disney, Arnold por Autodesk; otros de código abierto como Cycles por Community (2018), PBRT-v3 por Pharr y col. (2016) o Mitsuba 2 por Nimier-David y col. (2019).

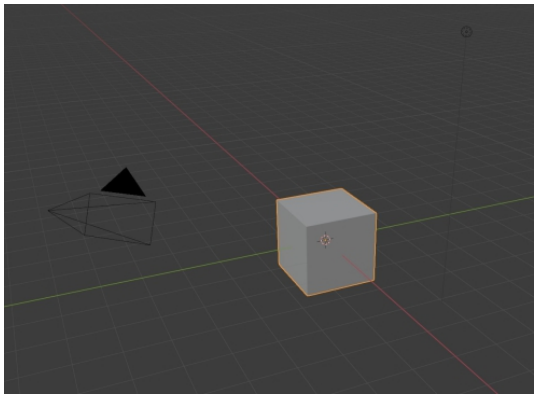


Figura 1.2: Escena básica de Blender

La mayoría de sistemas propietarios son usados en la industria pues ofrecen implementaciones optimizadas en hardware específico, en la descripción del comportamiento de la

luz, en el muestreo de rayos, etc. Estos sistemas por el simple hecho de ser propietarios, necesitan de una licencia para uso, la cual no te garantiza su modificación.

Los sistemas de código abierto suelen ser usados por estudiantes para aprender y por investigadores para desarrollar nuevos métodos para abarcar diferentes problemas que se encuentran en este campo.

Apéndice B

Monte Carlo

La método de análisis llamado *Monte Carlo* es una metodología que usa el promedio de una muestra para estimar el promedio de la población. La integración de *Monte Carlo* aplica este proceso a la estimación numérica de las integrales, en Pharr y col. (2016) se presenta su uso en computación gráfica. En general, *Monte Carlo* es un proceso numérico usado para estimar respuestas a una gran variedad de problemas, debida a su flexibilidad, los investigadores pueden experimentar de diferentes formas.

Su nombre viene del casino de Monte Carlo ubicado en Mónaco debido a los juegos de azar y se originó durante el desarrollo de la bomba atómica en Metropolis y Ulam (1949). La precisión de los *métodos de Monte Carlo* solo puede ser analizada desde un punto de vista estadístico.

2.1. Probabilidades

El método de *Monte Carlo* se basa en dos teoremas matemáticos: la ley de los grandes números y el teorema del límite central, un desarrollo más completo de este método puede encontrarse en Dunn y Shultis (2011).

2.1.1. Variables aleatorias

Una variable aleatoria X es una función que que mapea el resultado de un proceso aleatorio. Una variable aleatoria puede ser discreta (e.g. lanzar un dado) o continua (e.g. la altura de una persona). En Computación gráfica se lidia usualmente con variables aleatorias continuas las cuales toman valores sobre un rango de dominios continuos (e.g. las direcciones de una esfera).

2.1.2. Función de distribución acumulada

La función de distribución acumulada de una variable X es la probabilidad de que un valor sea elegido de la distribución de la variable sea menor o igual que x:

$$F(x) = P[X \leq x] \tag{2.1}$$

Caso Discreto

Siendo f la función de probabilidad de X.

$$F(x) = P[X \leq x] = \sum_{u<=x} f(u) \tag{2.2}$$

Caso Continuo

Siendo *pdf* la función de densidad de X.

$$F(x) = P[X \leq x] = \int_{-\infty}^{x} pdf(u)\, du \tag{2.3}$$

2.1.3. Función de densidad

La variable aleatoria X tiene la función de densidad *pdf*.

$$P[a \leq X \leq b] = \int_{a}^{b} pdf(x)\, dx \tag{2.4}$$

Si F es la función de distribución de X, entonces

$$F(x) = \int_{-\infty}^{x} pdf(u)\, du \tag{2.5}$$

y si la *pdf* es continua:

$$pdf(x) = \frac{d}{dx} F(x) \tag{2.6}$$

2.1.4. Esperanza

$$\langle x \rangle \equiv E(x) = \int_{a}^{b} x\, pdf(x)\, dx \tag{2.7}$$

2.1.5. Varianza

$$\sigma^2(x) = \int_{a}^{b} [x - \langle x \rangle]^2\, pdf(x)\, dx \tag{2.8}$$

2.2. Estimador de Monte Carlo

La integración de *Monte Carlo* usa un muestreo aleatorio de una función para computar númericamente un estiamdo de su integral. Supongamos que queremos integrar la función unidimensional $f(x)$ desde a a b

$$F = \int_{a}^{b} f(x) dx \tag{2.9}$$

Podemos aproximar esta integral promediando resultados de la función f en puntos aleatorios dentro del intervalo. Dado un conjunto de N variables aleatorias $X_i \in [a, b)$ con

función de densidad $1/(b-a)$ de modo que el estimador de *Monte Carlo* para computar F es:

$$\langle F \rangle = (b-a)\frac{1}{N-1}\sum_{i=0}^{N} f(X_i) \tag{2.10}$$

La variable aleatoria X_i puede ser construida utilizando un número aleatorio distribuido uniformemente entre cero y uno, $\xi_i \in [0,1)$

$$X_i = a + \xi_i(b-a) \tag{2.11}$$

de forma que podemos expandir el estimador

$$\langle F \rangle = (b-a)\frac{1}{N-1}\sum_{i=0}^{N} f(a+\xi_i(b-a)) \tag{2.12}$$

Es importante notar que el resultado obtenido es una aproximación a F utilizando N muestras.

2.2.1. Esperanza y convergencia

Es relativamente fácil mostrar que $\langle F \rangle$ es F, desde la definición de Esperanza y siendo la función de distribución $1/(b-a)$:

$$\begin{aligned}
E[\langle F \rangle] &= E[(b-a)\frac{1}{N-1}\sum_{i=0}^{N} f(X_i)] \\
&= (b-a)\frac{1}{N}\sum_{i=0}^{N-1} E[f(X_i)] \\
&= (b-a)\frac{1}{N}\sum_{i=0}^{N-1}\int_a^b f(x)\, pdf(x)\, dx \\
&= \frac{1}{N}\sum_{i=0}^{N-1}\int_a^b f(x)\, dx \\
&= \int_a^b f(x)\, dx \\
&= F
\end{aligned} \tag{2.13}$$

2.3. Reducción Varianza

Para mejorar el resultado de la integración de *Monte Carlo* necesitamos reducir la varianza. Desde que sus muestras son independientes, el cálculo de la varianza viene dado por:

$$\begin{aligned}\sigma^2[\langle F\rangle] &= \sigma^2[\frac{1}{N}\sum_{i=0}^{N-1}\frac{f(X_i)}{pdf(X_i)}] \\ &= \frac{1}{N^2}\sum_{i=0}^{N-1}\sigma^2[\frac{f(X_i)}{pdf(X_i)}] \\ &= \frac{1}{N^2}\sum_{i=0}^{N-1}\sigma^2[Y_i] \\ &= \frac{1}{N}\sigma^2[Y]\end{aligned} \tag{2.14}$$

Donde $Y_i = f(X_i)/pdf(X_i)$ y Y representa el resultado de cualquier Y_i, e.g., $Y = Y_i$.

2.3.1. Muestreo por importancia

El muestreo por importancia reduce la varianza observando que tenemos la libertad para escoger la función de densidad usada durante la integración. Escogiendo muestras desde una distribución $pdf(x)$, que tiene forma similares a la función que es integrada $f(x)$, la varianza es reducida.

2.3.2. Ruleta rusa

El método de la ruleta rusa es utilizado para poder evaluar muestras específicas que son costosas de computar y contribuyen muy poco al resultado final. Esta técnica divide el problema en subregiones y clasifica cada subregión como atractiva o inatractiva computacionalmente.

Apéndice C

Scripts

Script para convertir el formato XYZ generado por Mitsuba a uno RGB, pues estas imágenes necesitaban el formato correcto para ser compilados por latex.

```
from PIL import Image, ImageChops, ImageMath
import OpenEXR
import Imath
import numpy as np
import math, operator
from functools import reduce

def xyz_to_rgb(exrfile):
    file = OpenEXR.InputFile(exrfile)
    dw = file.header()["dataWindow"]
    size = (dw.max.x - dw.min.x + 1, dw.max.y - dw.min.y + 1)
    data = [
        np.frombuffer(c, np.float32).reshape(size)
        for c in file.channels("RGB", Imath.PixelType(Imath.PixelType.FLOAT))
    ]
    img = np.dstack(data)
    img = np.clip(img, 0, 1)
    # Gamma correction
    img = np.where(
        img <= 0.0031308, 12.92 * img, 1.055 * np.power(img, 1 / 2.4) - 0.055
    )
    return Image.fromarray((img * 255).astype(np.uint8), "RGB")
```

Calculo del Relative Square Error para su visualización, utilizando un módulo de PIL

```
out = ImageMath.eval(
    "((int(a)-int(b))**2)/(int(a)**2)",
    a=image_1,
    b=image_2,
)
```

Calculo del Relative Error Square en valor númerico

```
h = out.histogram()
print(
    math.sqrt(
        reduce(operator.add, map(lambda h, i: h * (i ** 2), h, range(256)))
        / (float(out.size[0]) * out.size[1])
    )
)
```