



Universidad
Zaragoza

Trabajo Fin de Grado

Ingeniería Informática

Modelado y simulación del transporte de la luz en la
atmósfera

Modeling and Simulation of Light Transport in the
Atmosphere

Autor

Pedro Andrés Gavín Murillo

Director

Adrián Jarabo Torrijos

ESCUELA DE INGENIERÍA Y ARQUITECTURA
Diciembre 2020

Modelado y simulación del transporte de la luz en la atmósfera

RESUMEN

La apariencia de la atmósfera está definida por las interacciones entre la luz y los gases que forman parte de esta. Estas interacciones son complejas, y están determinadas por las propiedades ópticas de dichos gases. Modelar correctamente la atmósfera y su interacción con la luz es un tema muy importante en gráficos por computador, y en otros ámbitos relacionados que tratan de resolver problemas inversos, incluyendo visión por computador basada en física, *remote sensing*, ciencias atmosféricas, o astrofísica. El objetivo de este proyecto es desarrollar un modelo físico del transporte de la luz en el medio atmosférico terrestre para el software de simulación de transporte de luz Mitsuba 2. Este software presenta un conjunto de características que lo hacen muy adecuado para su uso en problemas inversos de transporte de luz en medios atmosféricos, incluyendo diferenciación automática para su uso en problemas de optimización, soporte hiperespectral y de polarización de la luz, o implementación eficiente, vectorizada y adaptada a hardware gráfico.

A partir de trabajos científicos y mediciones experimentales de la atmósfera terrestre se ha desarrollado un modelo atmosférico parametrizable, en función de la posición del observador (tanto dentro como fuera de la atmósfera), de la turbidez de la atmósfera, de la fecha y del paisaje bioclimático (bioma). El modelo desarrollado incluye fenómenos complejos de interacción luz-materia, como la absorción debida a la capa de ozono, o la dispersión en base a moléculas o aerosoles, modelados de forma rigurosa desde las Ecuaciones de Maxwell.

Finalmente, se demuestra el modelo desarrollado mostrando las variaciones en la apariencia de la atmósfera dependiendo de los parámetros introducidos. Este proyecto es un primer paso para el desarrollo de modelos atmosféricos generales, capaces de predecir de forma precisa y sin simplificaciones arbitrarias el transporte de luz en la atmósfera para su uso en problemas inversos.

Índice

1. Introducción	1
2. Transporte de luz en medios participativos	3
2.1. <i>Path tracing</i> volumétrico	6
3. Modelo físico atmosférico	7
3.1. Modelos existentes	8
3.2. Propiedades ópticas de la atmósfera	9
3.3. Modelando la atmósfera	11
3.3.1. Dispersión molecular	11
3.3.2. Absorción molecular	12
3.3.3. Aerosoles	12
4. Diseño e implementación	14
4.1. Diseño de Mitsuba 2	14
4.2. Implementación	15
4.2.1. Medio participativo atmosférico	15
4.2.2. Funciones de fase	17
4.2.3. Cámara ojo de pez	17
5. Análisis del modelo	18
6. Resultados	22
7. Conclusiones	28
Bibliografía	32
Lista de Figuras	35
A. Guía de uso	36
B. Enlaces de interés	41

Capítulo 1

Introducción

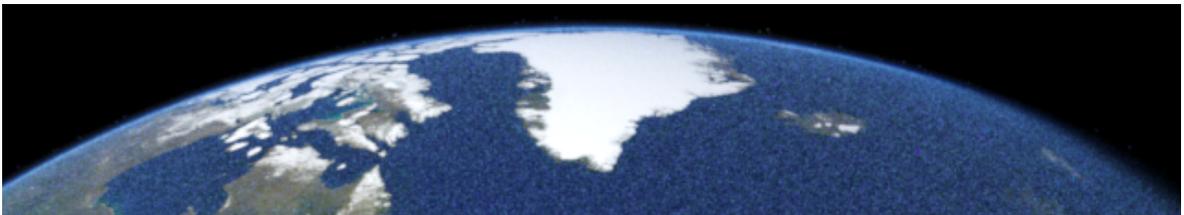


Figura 1.1: *Render* de la atmósfera terrestre vista desde el espacio.

La apariencia de la atmósfera es el resultado de complejas interacciones entre la luz incidente desde el Sol, y otras luminarias celestiales (p.e.: la Luna, galaxias, radiación cósmica) [Jen+01]. En concreto, la atmósfera está formada por un conjunto de gases, incluyendo moléculas y aerosoles suspendidos, que interactúan con la luz en cada punto diferencial de la atmósfera de forma no trivial. Simular la atmósfera terrestre y recrear la apariencia del cielo es importante para las aplicaciones que requieren mundos en los que el tiempo transcurra y poder observarlos desde el interior del planeta y desde el espacio. Estas aplicaciones pueden ir desde películas de animación y videojuegos, hasta la simulación científica de fenómenos naturales, visualización arquitectónica, estudios de ergonomía visual, o simuladores de conducción o de vuelo para pilotos de aeronaves. Así mismo, el desarrollo de modelos precisos de transporte de luz en la atmósfera tiene una gran importancia en campos donde interesa el *problema inverso*, en el que se intentan obtener los parámetros que definen la atmósfera en base a observaciones. Esto incluye campos como *remote sensing*, ciencias atmosféricas y ambientales, o incluso *image forensics* [LEN12], donde en base a observaciones del cielo se puede extraer información de la imagen como la fecha o la hora.

El reciente desarrollo de modelos de rendering diferencial [Zha+20], orientados a resolver problemas inversos utilizando modelos complejos de transporte de luz, se ha mostrado como una alternativa muy prometedora para resolver estos problemas inversos en base a modelos físicos complejos. Desafortunadamente, pese a que existen

en la literatura un buen número de modelos e implementaciones *open-source* de [Bru17], ninguna cumple con las características requeridas para aplicaciones a problemas inversos. En concreto, para resolver estos problemas en un marco de computación científica se requiere:

- Un modelo espectral de la atmósfera en base a medidas ópticas de la misma.
- Sin limitaciones comunes en modelos de ciencias atmosféricas y *remote sensing* [Cha50], incluyendo la asunción de atmósfera plana, luz incidente colimada, o superficie puramente absorbente.
- Adaptable a problemas de optimización complejos, eficiente y diferenciable.

En este proyecto se trata de solventar estos problemas, implementando un modelo general basado en mediciones de la atmósfera dentro del software de simulación de transporte de luz basado en física Mitsuba 2 [Nim+19]. Mitsuba 2 es *open-source*, hiperespectral, vectorizable con soporte para CPU y GPU, y, mediante diferenciación automática, orientado a su uso en problemas inversos. En concreto, en este trabajo se implementa el modelo de Guimera y col. [GGJ18], que parametriza la atmósfera en base a la fecha de observación, la situación bioclimática y la turbiedad del ambiente. El modelo es general, permitiendo condiciones de observación arbitrarias en términos de posición (dentro y fuera de la atmósfera), o en el dominio temporal (hora y fecha, incluyendo condiciones de anochecer con el Sol debajo del horizonte).

El trabajo realizado es un primer paso en la adopción del *render* diferenciable para solucionar problemas inversos basados en observaciones de la atmósfera. Para ello, toda la implementación está disponible *open-source* en¹. Además, es fácilmente extendible a otras atmósferas, requiriendo únicamente incluir los parámetros ópticos que definen la atmósfera en cuestión.

¹<https://github.com/agavinm/atmosphere-renderer>

Capítulo 2

Transporte de luz en medios participativos

Un medio participativo es una mezcla de partículas que interactúan con la luz (p.e.: humo, arena en suspensión, aire, etc.) [Figura 2.1]. Este tipo de interacciones se clasifican en absorción, emisión, *in-scattering* y *out-scattering*. Se produce un evento de **absorción** cuando una partícula absorbe parte de la radiancia espectral visible del rayo de luz y la libera en forma de radiación no visible (p.e.: calor), mientras que se produce un evento de **emisión** cuando una partícula absorbe parte de la radiación no visible del rayo y la libera como luz visible (luminiscencia). En cambio, un evento de ***out-scattering*** ocurre cuando una partícula dispersa parte de la radiancia espectral del rayo de luz en diferentes direcciones y un evento de ***in-scattering*** ocurre cuando a una partícula le llegan rayos de luz de distintas direcciones y los dispersa en dirección al rayo de luz principal. Formalmente, estas interacciones se explican con las Ecuaciones de Maxwell, que modelan cómo un campo electromagnético incidente (la luz) interactúa con las partículas suspendidas, generando un nuevo campo. Modelar esta problemática utilizando electromagnetismo en sistemas como la atmósfera, sin embargo, es imposible computacionalmente. En su lugar, se adopta un enfoque probabilista, en el que se modela estadísticamente cada uno de esos eventos mediante los coeficientes de absorción, emisión y *scattering*. Estos modelan la probabilidad diferencial de interacción con el medio, con unidad m^{-1} . Además, la pérdida de radiancia espectral del rayo de luz se mide mediante el coeficiente de extinción, que es la suma del coeficiente de absorción y del coeficiente de *scattering* $\mu_t(\mathbf{x}, \lambda) = \mu_a(\mathbf{x}, \lambda) + \mu_s(\mathbf{x}, \lambda)$.

Dependiendo de las características del medio participativo se distinguen los medios participativos homogéneos y los heterogéneos. Un **medio participativo homogéneo** es aquel que tiene la misma densidad y composición en cualquier punto \mathbf{x} , por lo que su interacción con la luz no varía aunque cambie la posición desde donde se mida (p.e.: nube con la misma densidad y composición en todo su volumen). Por tanto, los

coeficientes de *scattering* μ_s y extinción μ_t son constantes. Mientras que un **medio participativo heterogéneo** es aquel cuya densidad y/o composición son distintas en algún punto del medio, y por tanto su interacción con la luz puede cambiar dependiendo de donde se mida (p.e.: humo que se disipa con la altura). Así, los coeficientes de *scattering* μ_s y extinción μ_t dependen de la posición \mathbf{x} y de la longitud de onda λ .

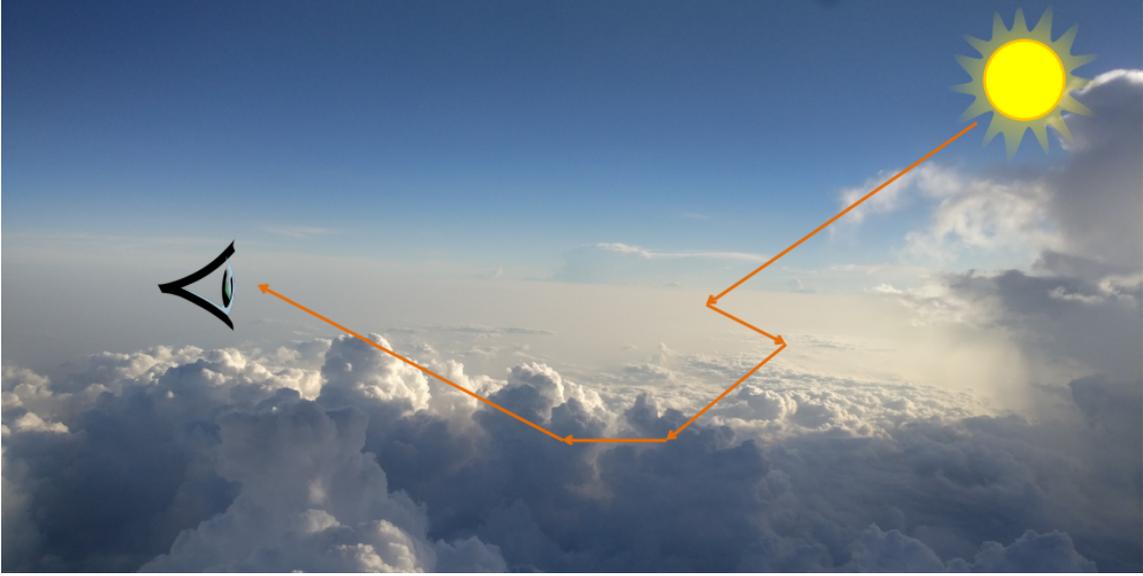


Figura 2.1: Transporte de la luz en un medio participativo para una fotón. En cada interacción con el medio, la partícula puede ser absorbida o dispersada en otra dirección. La apariencia del medio es el resultado de la interacción de muchas de estas partículas al llegar al sensor (p.ej. el ojo). [Imagen cortesía de Adrián Jarabo]

Basado en esos coeficientes, podemos modelar el flujo neto de radiancia al avanzar una distancia diferencial en la dirección $\vec{\omega}_o$ como:

$$(\vec{\omega}_o \cdot \nabla)L(\mathbf{x}, \vec{\omega}_o, \lambda) = -\mu_t(\mathbf{x}, \lambda)L(\mathbf{x}, \vec{\omega}_o, \lambda) + \mu_s(\mathbf{x}, \lambda) \int_{S^2} f(\mathbf{x}, \vec{\omega}_i \rightarrow \vec{\omega}_o, \lambda)L(\mathbf{x}, \vec{\omega}_i, \lambda)d\vec{\omega}_i \quad (2.1)$$

Esta ecuación es una ecuación diferencial sobre la radiancia espectral L para una longitud de onda λ en el punto \mathbf{x} y con dirección $\vec{\omega}_o$, y define su variación como la suma del coeficiente de extinción negativo $-\mu_t(\mathbf{x}, \lambda)$ multiplicado por la radiancia espectral emitida $L(\mathbf{x}, \vec{\omega}_o, \lambda)$ más el coeficiente de *scattering* $\mu_s(\mathbf{x}, \lambda)$ multiplicado por la radiancia espectral incidente $L(\mathbf{x}, \vec{\omega}_i, \lambda)$, dada la función de fase $f(\mathbf{x}, \vec{\omega}_i \rightarrow \vec{\omega}_o, \lambda)$ y para toda la esfera unitaria S^2 . No se consideran los eventos de emisión debido a que la atmósfera terrestre no emite luz en el espectro visible.

El coeficiente de extinción es negativo debido a que cada partícula que interactúe con el rayo de luz [Figura 2.3] hace que pierda radiancia espectral en los eventos de

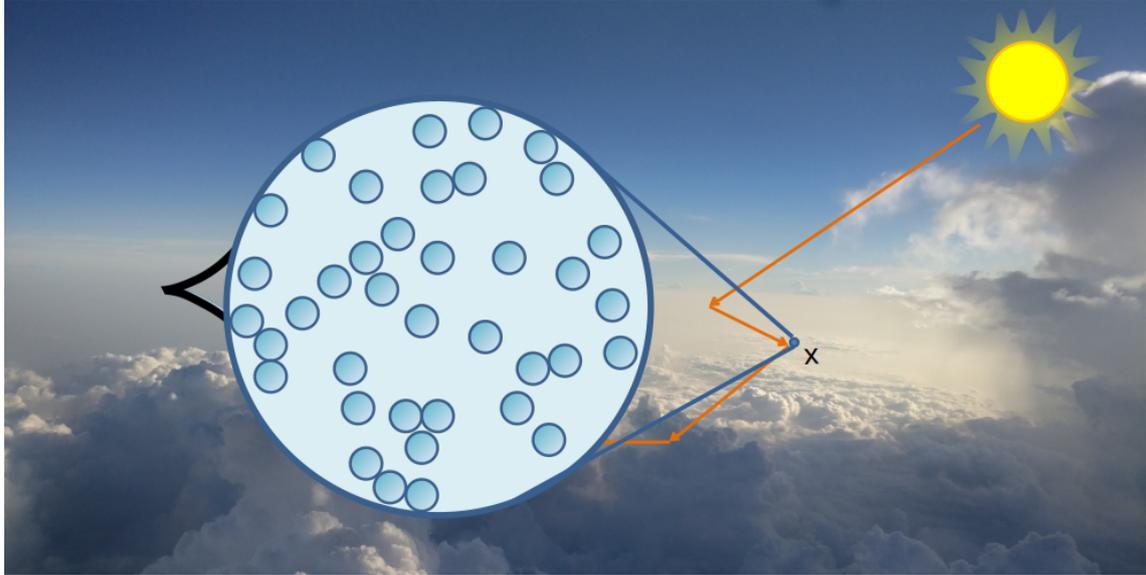


Figura 2.3: Ampliación de las partículas que conforman el medio participativo para el punto \mathbf{x} . [Imagen cortesía de Adrián Jarabo]

absorción y *scattering*. Dando como resultado un rayo de luz emitido en dirección $\vec{\omega}_o$ con menor radiancia espectral como se muestra en la Figura 2.4.

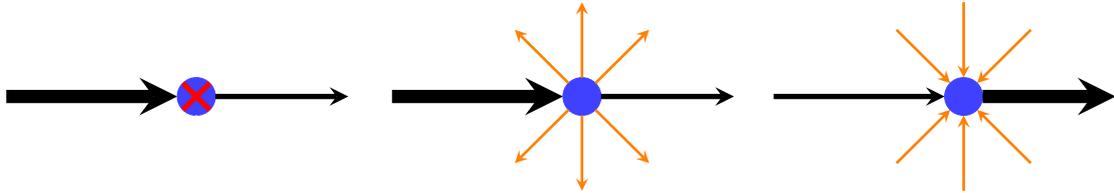


Figura 2.4: A la izquierda, una partícula absorbe parte de la radiancia espectral del rayo de luz (evento de absorción). En el centro, una partícula dispersa parte de la radiancia espectral en diferentes direcciones (evento de *out-scattering*). A la derecha, una partícula dispersa parte de la radiancia espectral proveniente de otros rayos de luz en la misma dirección que la del rayo de luz emitido (evento de *in-scattering*).

A su vez, el coeficiente de *scattering* $\mu_s(\mathbf{x}, \lambda)$ de la ecuación de *render* volumétrica es positivo debido a que cada partícula que interactúe con el rayo de luz dispersa parte de la radiancia espectral que le llega de otros rayos de luz en la misma dirección que el rayo de luz incidente. Para calcular la dispersión entrante se realiza la integral en toda la esfera unitaria S^2 desde todas las direcciones $\vec{\omega}_i$. Esto da como resultado que el rayo de luz emitido tenga mayor radiancia espectral como se muestra en la Figura 2.4.

Los coeficientes de extinción, absorción y *scattering* se definen en base a la concentración de partículas del medio por su sección transversal media. Por ejemplo, el coeficiente de absorción se calcula como $\mu_a(\mathbf{x}, \lambda) = C_a(\mathbf{x})\sigma_a(\mathbf{x}, \lambda)$, donde la concentración $C_a(\mathbf{x})$ es el número de partículas por volumen en m^{-3} y la sección

transversal media $\sigma_a(\mathbf{x}, \lambda)$ en m^2 . De manera análoga se obtiene el coeficiente de *scattering*.

Finalmente, la función de fase $f(\mathbf{x}, \vec{\omega}_i \rightarrow \vec{\omega}_o, \lambda)$ es una función de probabilidad que se utiliza para calcular cómo la radiación espectral proveniente de todas las direcciones $\vec{\omega}_i$ dentro de la esfera unitaria S^2 es dispersada hacia la dirección $\vec{\omega}_o$ en el punto \mathbf{x} .

2.1. Path tracing volumétrico

Las escenas normales no contienen únicamente medios participativos, sino que también contienen objetos que interactúan con la luz. Para calcular la radiancia espectral L que le llega a un punto \mathbf{x} , hay que considerar no sólo las interacciones de la luz en el medio participativo, sino también las interacciones con los objetos, y por ello se utiliza *path tracing* volumétrico, donde un mismo camino puede tener interacciones con un medio participativo y con un objeto como se muestra en la Figura 2.5.

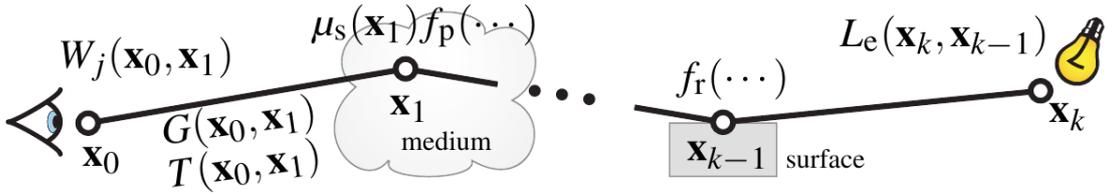


Figura 2.5: Camino completo desde un emisor hasta la cámara atravesando distintos medios participativos e interactuando con objetos [Nov+18].

Se puede calcular el valor de un píxel j en una imagen I como:

$$I_j = \int_P f_j(\bar{x}) d\bar{x} \quad (2.2)$$

donde $\bar{x} = (x_0, x_1, \dots, x_k) \in P$ es un camino de la luz de longitud $k \in [1, \infty)$, P es el espacio de todos los caminos posibles en la escena, $d\bar{x} = \prod_{i=0}^k dx_i$ y $f_j(\bar{x})$ es la contribución para los caminos de longitud k definida como:

$$f_j(\bar{x}) = W_j(x_0, x_1) L_e(x_k, x_{k-1}) G(x_0, x_1) T(x_0, x_1) \prod_{i=1}^{k-1} f_s(x_{i-1}, x_i, x_{i+1}) G(x_i, x_{i+1}) T(x_i, x_{i+1}), \quad (2.3)$$

Esta ecuación está desarrollada en [Nov+18] y, como se explica en dicho documento, se tiene en cuenta si los distintos valores de x_i pertenecen a una interacción con el medio participativo o con un objeto.

Capítulo 3

Modelo físico atmosférico

La atmósfera terrestre es la parte más externa de la Tierra, que está compuesta por una mezcla de gases y partículas en suspensión. Los gases predominantes son el nitrógeno (78,09 %) y el oxígeno (20,95 %), seguidos del argón, del dióxido de carbono y del ozono entre otros. Las partículas en suspensión pueden ser procedentes de procesos naturales como restos de arena y sales marinas transportadas por el viento y el agua condensada de las nubes, o procedentes de procesos artificiales a causa de la acción del ser humano en forma de humo y gases condensados.

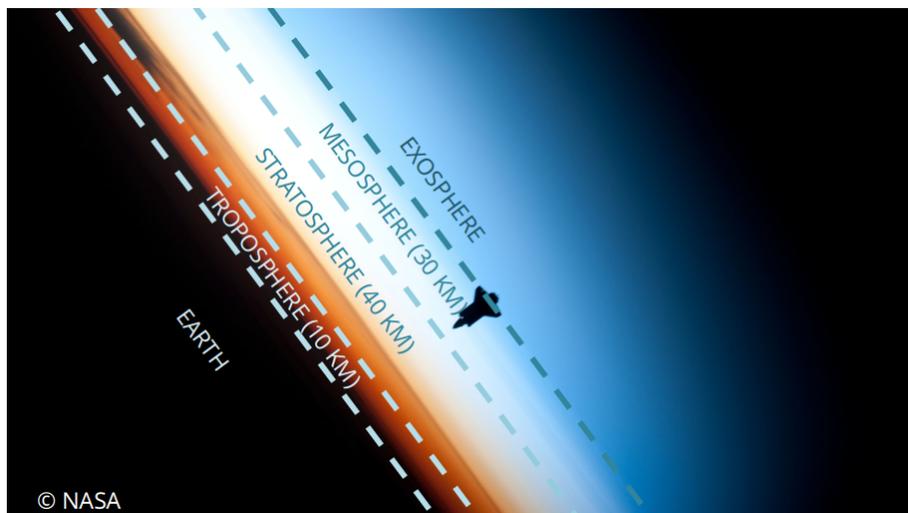


Figura 3.1: Distribución de las capas de la atmósfera [GGJ18].

La atmósfera terrestre está formada por capas superpuestas con diferentes concentraciones de **gases** y **aerosoles** (partículas en suspensión), como se puede observar en la Figura 3.1. En concreto se divide en: la troposfera, que abarca los primeros 12 km desde el nivel del mar y que contiene la mayor parte de la masa atmosférica, la estratosfera, que continúa hasta los 50 km y que contiene la capa de ozono, la mesosfera, que transcurre desde los 50 km hasta los 80 km de elevación respecto del nivel del mar, la termosfera o ionosfera, que abarca desde los 80 km hasta

los 700 km de altitud y donde se encuentra la Línea de Karman, que es el límite imaginario entre la atmósfera terrestre y el espacio exterior, y la exosfera que es la capa más alta de la atmósfera con una altitud de hasta 10.000 km.

3.1. Modelos existentes

Simular el comportamiento de la luz en un medio atmosférico puede resultar útil para observar diferentes efectos ópticos en planetas o condiciones atípicas, pero principalmente el mayor uso conocido es en los videojuegos y en las películas e imágenes generadas por ordenador. Esto ha promovido muchos trabajos previos donde se estudia y modela el transporte de la luz en la atmósfera, como los trabajos de Bruneton y Neyret [BN08], y Hosek y Wilkie [HW12]. Como se explica en el trabajo de Guimera y col. [GGJ18], se pueden distinguir entre todos los modelos atmosféricos desarrollados los explícitos y los analíticos.

Los modelos explícitos parten de mediciones precisas de la distribución y composición de moléculas (gases) y partículas en suspensión (aerosoles), y simulan mediante fuerza bruta cómo la luz interacciona con dichas partículas. El primer modelo fue propuesto por Nishita y col. [Nis+93] en el que se utilizaban moléculas y aerosoles, pero sólo se tenía en cuenta el *scattering* individual. Para acelerar el renderizado Bruneton y Neyret [BN08] y Elek y Kmoch [EK10] propusieron precalcular la radiancia espectral en la atmósfera y utilizar sus modelos en GPUs. Basado en el modelo de Nishita y col., O’Neil [ONe05] desarrolló un modelo en tiempo real limitado al *scattering* individual. Jensen y col. [Jen+01] utilizaron también el modelo de Nishita y col. para desarrollar un cielo nocturno que incluía luz de la Luna y las estrellas. Basándose en mediciones reales, Haber y col. [HMS05] propuso un modelo atmosférico muy preciso que tenía en cuenta tanto la humedad como la temperatura para calcular el color durante el atardecer. Gutierrez y col. [Gut+04] también utilizaron rayos curvos para simular efectos como los rayos verdes producidos segundos después del amanecer y del atardecer. Kutz [Kut12] también utilizó mediciones del ozono para representar un modelo del cielo más preciso. En este trabajo se utiliza el modelo explícito propuesto en [GGJ18] para representar la atmósfera teniendo en cuenta la posición geográfica y la fecha.

Los modelos analíticos representan la apariencia atmosférica como una función dependiente únicamente de la dirección de la cámara, es decir, sin utilizar mediciones explícitas. Estos modelos resultan eficaces como muestran CIE [CIE04] y Perez

y col. [PSM93] partiendo de una aproximación de la luminosidad del cielo en diferentes condiciones meteorológicas. Preetham y col. [PSS99] propusieron un modelo parametrizado por la turbiedad de la atmósfera. Wilkie y col. [Wil+04] partió del modelo de Preetham y col. y añadió la polarización de la luz. Finalmente, Hosek y Wilkie [HW12] incrementó la precisión bajo condiciones de alta turbiedad del cielo utilizando *rendering* hiperspectral en lugar de RGB, e incluyendo una descripción más precisa del Sol en su trabajo posterior [HW13]. Estos modelos utilizan una atmósfera parametrizable limitada únicamente a la turbiedad, por lo que estos modelos no se utilizan en este proyecto.

Existen modelos más recientes como el de Hillaire [Hil20] que propone una técnica físicamente correcta para simular el transporte de la luz en la atmósfera en tiempo real (por medio de GPU), teniendo en cuenta el *scattering* múltiple y una composición dinámica de la atmósfera que puede cambiar dependiendo de las condiciones climatológicas. Todos los modelos se pueden extender y añadir otros tipos de atmósferas, como en el trabajo de Costa y col. [Cos+20] que, además de la atmósfera terrestre, utilizan la atmósfera marciana. Dada su flexibilidad, se ha decidido implementar el modelo de Guimera y col., que se detalla en la Sección 3.3 [GGJ18].

3.2. Propiedades ópticas de la atmósfera

Para definir las propiedades ópticas de la atmósfera terrestre, y como se ha hecho en los trabajos relacionados, la atmósfera se considera un medio participativo heterogéneo compuesto por dos tipos de partículas elementales: moléculas y aerosoles. Una **molécula** es aquella partícula que tiene un tamaño menor que las longitudes de onda de la luz (gases). Por el contrario, un **aerosol** es aquella partícula que tiene un tamaño mayor (partículas líquidas o sólidas en suspensión).

Se asume que el *scattering* de estas partículas es inelástico, es decir, no hay transferencia de energía entre diferentes longitudes de onda en forma de luminiscencia, y que las partículas son independientes entre sí (en caso de correlación, la ecuación 2.2 no sería válida y habría que usar su forma generalizada [JAG18]). Esto quiere decir que para los coeficientes de *scattering*, absorción y extinción, su concentración $C(\mathbf{x})$ se puede calcular como el sumatorio de las concentraciones $C^j(\mathbf{x})$ para cada tipo de partícula independiente j del medio M , y su sección transversal $\sigma(\mathbf{x}, \lambda)$ como la proporción de la concentración de la partícula independiente j del medio M multiplicada por el sumatorio de su sección transversal $\sigma^j(\mathbf{x}, \lambda)$ como se muestra en las siguientes ecuaciones.

$$C(\mathbf{x}) = \sum_{j \in M} C^j(\mathbf{x}) \quad (3.1)$$

$$\sigma(\mathbf{x}, \lambda) = \sum_{j \in M} \frac{C^j(\mathbf{x})}{C(\mathbf{x})} \sigma^j(\mathbf{x}, \lambda) \quad (3.2)$$

De la misma manera, se puede definir la siguiente función de fase para el medio M

$$f(\mathbf{x}, \vec{\omega}_i \rightarrow \vec{\omega}_o, \lambda) = \sum_{j \in M} \frac{\mu_s^j(\mathbf{x}, \lambda)}{\mu_s(\mathbf{x}, \lambda)} f^j(\mathbf{x}, \vec{\omega}_i \rightarrow \vec{\omega}_o, \lambda), \quad (3.3)$$

donde $\mu_s^j(\mathbf{x}, \lambda)$ es el coeficiente de *scattering* para la partícula independiente j y $f^j(\mathbf{x}, \vec{\omega}_i \rightarrow \vec{\omega}_o, \lambda)$ es su función de fase.

Las moléculas son las responsables del color de la atmósfera (incluidos los tonos rojizos del atardecer), que presentan una distribución espacial más homogénea. Principalmente producen eventos de *scattering* al interactuar con la luz, por lo que se considera su coeficiente de absorción $\mu_a^m(\mathbf{x}, \lambda) = 0$, a excepción del ozono que absorbe toda la luz y, de la misma manera, se considera su coeficiente de *scattering* $\mu_s^o(\mathbf{x}, \lambda) = 0$. Por contra, los aerosoles están distribuidos en la parte más baja de la atmósfera y son dependientes del entorno geográfico en el que están. Esto permite distinguir entre las moléculas, el ozono y los aerosoles para calcular los coeficientes de absorción y de *scattering* como se muestra a continuación.

$$\mu_a(\mathbf{x}, \lambda) = \mu_a^o(\mathbf{x}, \lambda) + \mu_a^a(\mathbf{x}, \lambda), \quad (3.4)$$

$$\mu_s(\mathbf{x}, \lambda) = \mu_s^m(\mathbf{x}, \lambda) + \mu_s^a(\mathbf{x}, \lambda), \quad (3.5)$$

donde los coeficientes con superíndices m , o y a representan, respectivamente, los coeficientes para las moléculas, el ozono y los aerosoles. De esta manera, se puede definir la función de fase para la atmósfera como la suma ponderada de las funciones de fase de las moléculas $f^m(\mathbf{x}, \vec{\omega}_i \rightarrow \vec{\omega}_o, \lambda)$ y de los aerosoles $f^a(\mathbf{x}, \vec{\omega}_i \rightarrow \vec{\omega}_o, \lambda)$.

$$f(\mathbf{x}, \vec{\omega}_i \rightarrow \vec{\omega}_o, \lambda) = \frac{\mu_s^m(\mathbf{x}, \lambda)}{\mu_s(\mathbf{x}, \lambda)} f^m(\mathbf{x}, \vec{\omega}_i \rightarrow \vec{\omega}_o, \lambda) + \frac{\mu_s^a(\mathbf{x}, \lambda)}{\mu_s(\mathbf{x}, \lambda)} f^a(\mathbf{x}, \vec{\omega}_i \rightarrow \vec{\omega}_o, \lambda) \quad (3.6)$$

3.3. Modelando la atmósfera

Para modelar la atmósfera se ha establecido el límite en la mesosfera (a 86 km de altitud) debido a que hasta ese punto se concentran la gran mayoría de gases y aerosoles, y a nivel computacional resulta más eficiente descartar la termosfera y la exosfera, que apenas tienen masa atmosférica pero abarcan una gran altitud (hasta los 10.000 km). Además, se ha considerado a la Tierra como una esfera perfecta y difusa, cuyo albedo varía en función de la posición geográfica del observador.

3.3.1. Dispersión molecular

En el *scattering* molecular $\mu_s^m(\mathbf{x}, \lambda)$ predomina el *Rayleigh scattering* como estableció el físico Strutt [Str71]. Este tipo de *scattering* presenta una dispersión de la luz de forma difusa, y depende principalmente de la longitud de onda del rayo de luz y de la altitud de la interacción en el punto \mathbf{x} , ya que la concentración de las moléculas varía en función de la presión y de la temperatura, que dependen de la altitud en la atmósfera. De esta manera, se puede definir el *scattering* molecular como $\mu_s^m(z, \lambda) = C_s^m(z) \sigma_s^m(\lambda)$, donde $C_s^m(z)$ es la concentración de moléculas para la altitud z y $\sigma_s^m(\lambda)$ es la sección transversal molecular para la longitud de onda λ . Los valores medios de $C_s^m(z)$ para distintas altitudes z se han extraído de *U.S. Standard Atmosphere* [Atm76], mientras que los valores medios de $\sigma_s^m(\lambda)$ para distintas longitudes de onda λ se han extraído del documento *Rayleigh-scattering calculations for the terrestrial atmosphere* propuesto por Bucholtz [Buc95].

La función de fase para las moléculas de *Rayleigh* sólo depende de las direcciones de los rayos de entrada $\vec{\omega}_i$ y de salida $\vec{\omega}_o$ en la interacción de las moléculas con la luz.

$$f^m(\vec{\omega}_i \rightarrow \vec{\omega}_o) = \frac{3}{16\pi}(1 + \cos^2 \theta) \quad (3.7)$$

Mientras que la función de fase de Chandrasekhar [Cha50] también depende de la longitud de onda λ , por lo que es más precisa y es la que se ha utilizado.

$$f^m(\vec{\omega}_i \rightarrow \vec{\omega}_o, \lambda) = \frac{3}{16\pi(1 + 2\gamma(\lambda))} [1 + 3\gamma(\lambda) + (1 - \gamma(\lambda)) \cos^2 \theta], \quad (3.8)$$

donde $\cos \theta$ es el producto escalar entre $\vec{\omega}_i$ y $\vec{\omega}_o$, y los valores medios de $\gamma(\lambda)$ para distintas longitudes de onda λ se han obtenido del documento *Rayleigh-scattering calculations for the terrestrial atmosphere* [Buc95].

3.3.2. Absorción molecular

La **absorción molecular** se debe a la absorción del ozono $\mu_a^o(\mathbf{x}, \lambda)$, que se acumula mayoritariamente en la capa de ozono y está ubicada en la estratosfera. Al igual que para las demás moléculas, el ozono depende principalmente de la longitud de onda del rayo de luz y de la altitud de la interacción en el punto \mathbf{x} . Además, como menciona Dütsch [Düt74], la concentración de ozono también varía en función del mes en el que se mida. Por lo que se puede definir la absorción del ozono como $\mu_a^o(z, t, \lambda) = C_a^o(z, t) \sigma_a^o(\lambda)$. Los valores medios de $C_a^o(z, t)$ se han extraído de [RK53] aplicando la corrección definida en [Düt74], mientras que los valores medios de $\sigma_a^o(\lambda)$ se han extraído del trabajo de Gorshlev y col. [Gor+14].

3.3.3. Aerosoles

A diferencia de los gases, los aerosoles presentan una distribución y composición diferentes en función del terreno geográfico desde donde se midan. Como propusieron Zimmermann y col. [Zim+89] y con la ampliación de Jaenicke [Jae93], se han modelado los siguientes entornos geográficos diferentes: *Background*, *Desert-Dust*, *Maritime-Clean*, *Maritime-Mineral*, *Polar-Antartic*, *Polar-Artic*, *Remote-Continental*, *Rural* y *Urban*. De manera similar a las moléculas, el **scattering y la absorción de los aerosoles** para cada tipo de entorno dependen de la longitud de onda del rayo de luz y de la altitud de la interacción en el punto \mathbf{x} , por lo que se pueden definir como $\mu_s^a(z, \lambda) = C^a(z) \sigma_s^a(\lambda)$ y $\mu_a^a(z, \lambda) = C^a(z) \sigma_a^a(\lambda)$ respectivamente. Donde las secciones transversales medias $\sigma_s^a(\lambda)$ y $\sigma_a^a(\lambda)$ para distintas longitudes de onda λ se han calculado usando teoría de Mie [Hul57]: La teoría de Mie describe la solución de las Ecuaciones de Maxwell cuando un campo electromagnético plano interactúa con una partícula esférica dieléctrica, parametrizado por el índice de refracción complejo de la esfera y su tamaño. Los parámetros de cada tipo de aerosol provienen del trabajo de D’Almeida y col. [DSK91]. Finalmente, la concentración $C^a(z)$ para cada tipo de aerosol se ha calculado como se explica en [AIA99],

$$C^a(z) = C^a(0) \left(\exp\left(\frac{-z}{H_p}\right) + \left(\frac{C_b}{C^a(0)}\right) \right), \quad (3.9)$$

donde $C^a(0)$ es la concentración en el suelo ($z = 0$), y las constantes H_p y C_b son dependientes de cada tipo de aerosol y se han extraído del mismo documento.

La función de fase elegida para los aerosoles es la función de fase de Henyey-Greenstein [HG41] que es una aproximación empírica de la función de fase

de Mie, y resulta significativamente más simple de calcular como:

$$f^a(\vec{\omega}_i \rightarrow \vec{\omega}_o) = \frac{1 - k^2}{4\pi(1 + k \cos \theta)^2}, \quad (3.10)$$

donde $\cos \theta$ es el producto escalar entre $\vec{\omega}_i$ y $\vec{\omega}_o$, y $k \in [-1, 1]$ controla la dirección de

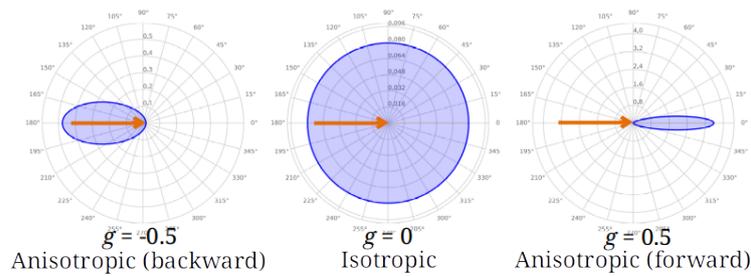


Figura 3.12: Tipos de *scattering* anisotrópico e isotrópico [$g = k$]. [Imagen cortesía de Adrián Jarabo]

scattering más probable, es decir, si $k = -1$ la dirección de *scattering* más probable será $\vec{\omega}_i$ (*scattering* anisotrópico (*backward*)), si $k = 1$ será $\vec{\omega}_o$ (*scattering* anisotrópico (*forward*)) y si $k = 0$ todas las direcciones serán equiprobables (*scattering* isotrópico), como se muestra en la Figura 3.12.

Capítulo 4

Diseño e implementación

Mitsuba 2 es un motor de *render open-source* escrito en C++²⁰, y programado siguiendo un paradigma de orientación a objetos. Implementa una serie de algoritmos de simulación de transporte de luz, basados en métodos de Monte Carlo, de entre los que destaca *path tracing* [Sección 2.1]. Dichos algoritmos operan sobre la unidad mínima de energía lumínica (radiancia espectral), sobre todo el espectro visible, que luego es transformada a RGB para generar la imagen resultante. Además, incluye por defecto una serie de luminarias y cámaras, así como modelos de apariencia complejos, y diferentes tipos de medios generales (homogéneos y heterogéneos).

4.1. Diseño de Mitsuba 2

Mitsuba 2 sigue un paradigma de orientación a objetos, donde se implementan unas clases virtuales base para definir luminarias, cámaras, materiales, medios participativos, y funciones de fase. A cada uno de estos elementos se accede desde una clase proxy (Scene), sobre la cual el algoritmo de simulación de transporte de luz (Integrator) accede. Cada una de las clases que heredan de las clases virtuales están definidas como plugins, donde se implementan cada material, luminaria, cámara, etc.

Mitsuba 2 está construido sobre la biblioteca Enoki¹, que permite vectorizar de forma transparente al programador el código, adoptando modelos escalares, vectoriales, o de paralelismo masivo para p.ej. procesadores gráficos (GPU). Así mismo, esta biblioteca implementa nativamente funciones de autodiferenciación, que permiten computar de forma automática la derivada de cada interacción luz-materia.

Durante la realización de este trabajo se han desarrollado los siguientes *plugins*:

¹Enoki: <https://enoki.readthedocs.io/en/master/>

- El medio participativo atmósfera, que implementa el cálculo de los coeficientes de *scattering* y absorción para las moléculas y los aerosoles.
- Las funciones de fase de Chandrasekhar, Rayleigh y Henyey-Greenstein.
- Una cámara de ojo de pez, que permite una apertura de hasta 360°.

4.2. Implementación

4.2.1. Medio participativo atmosférico

Los medios participativos heredan de la clase *Medium* que expone los siguientes métodos.

- El método *get_scattering_coefficients(...)* devuelve los coeficientes de *scattering* μ_s y extinción μ_t .
- El método *phase_function(...)* devuelve la función de fase que se utiliza en el medio participativo.

Todos ellos están parametrizados por la posición en la que se realiza el evento de *scattering*/absorción.

Se ha desarrollado el medio participativo atmosférico partiendo de la implementación del medio participativo heterogéneo, y se han añadido los siguientes parámetros:

- El parámetro *molecular_phase_function* establece la función de fase molecular que se va a utilizar, entre la función de fase de Chandrasekhar y Rayleigh.
- El parámetro *month* define el mes del año en la escena, que variará la concentración del ozono.
- El parámetro *turbidity* establece la proporción de turbidez de los aerosoles respecto de su densidad base.
- El parámetro *aerosol_model* indica el modelo de aerosol que se va a utilizar, es decir, el entorno bioclimático (*Background*, *Desert-Dust*, *Maritime-Clean*, *Maritime-Mineral*, *Polar-Antartic*, *Polar-Artic*, *Remote-Continental*, *Rural* o *Urban*). Por defecto no utiliza ningún modelo de aerosol y sólo utiliza el modelo molecular (gases).
- Los parámetros *earth_radius* y *earth_center* establecen el radio de la Tierra y la posición de su centro en las unidades que se deseen.

En primer lugar, se han tabulado los valores extraídos en el Capítulo 3 para las secciones transversales ($\sigma_s^m(\lambda)$ y $\sigma_s^a(\lambda)$) y concentraciones ($C_s^m(z)$ y $C_s^a(z, t)$ [para cada mes t]) moleculares respecto de z y λ respectivamente. De la misma manera, se han tabulado los valores para las secciones transversales de los aerosoles ($\sigma_s^a(\lambda)$ y $\sigma_a^a(\lambda)$) obtenidas mediante las ecuaciones de Mie y a partir de sus índices de refracción con el programa MiePlot², y sus concentraciones ($C^a(z)$) extraídas en el Capítulo 3.

Los límites de las longitudes de onda λ están establecidos por Mitsuba 2 como 360 nm en el límite inferior y 830 nm en el límite superior, mientras que los límites de las altitudes z se han establecido como 0 km en el límite inferior y 86 km en el límite superior, por lo que si una altitud z supera estos límites los coeficientes devueltos valdrán 0. Después, se calcula la extinción máxima $max(\mu_t)$ desde $z = 0$ km hasta $z = 87$ km calculando las concentraciones C cada 0.01 km y utilizando los valores máximos de sección transversal σ .

get_scattering_coefficients(...) Para calcular la posición \mathbf{x} de la interacción, en primer lugar, se calcula la escala a la que está la Tierra, ya que los cálculos internos se hacen en kilómetros, como $earth_scale = \frac{6356.766}{earth_radius}$. A partir de aquí, la altitud z es la distancia entre el punto donde ha ocurrido una interacción y el centro de la Tierra, menos el radio de la Tierra y multiplicado por la escala, $z = (|\vec{d}| - earth_radius) \times earth_scale$. Para obtener el valor de cada coeficiente, los valores tabulados en precómputo se interpolan en run-time para obtener $\mu(z, \lambda) = C(z) * \sigma(\lambda)$. Nótese que C y σ son separables, de modo que realizamos dos interpolaciones 1D en lugar de una sola interpolación bilinear.

phase_function(...) Para elegir la función de fase en cada interacción se selecciona de forma probabilista entre dispersión molecular o por aerosoles, mediante la técnica de ruleta rusa con dos eventos. La ruleta rusa con dos eventos consiste en generar un número aleatorio mayor o igual que 0 y menor que 1 donde, si la probabilidad del primer evento es mayor al número aleatorio se realiza el primer evento, y en caso contrario se realiza el segundo evento. Así, se asigna como probabilidad de utilizar la función de fase molecular f^m la proporción de *scattering* molecular $\mu_s^m(\mathbf{x}, \lambda)$ sobre el *scattering* total $\mu_s(\mathbf{x}, \lambda) = \mu_s^m(\mathbf{x}, \lambda) + \mu_s^a(\mathbf{x}, \lambda)$, es decir,

²MiePlot: <http://www.philiplaven.com/mieplot.htm>

$$p(f^m) = \frac{\mu_s^m(\mathbf{x}, \lambda)}{\mu_s(\mathbf{x}, \lambda)}, \quad (4.1)$$

mientras que la probabilidad de *scattering* por aerosoles es:

$$P(f^a) = 1 - p(f^m). \quad (4.2)$$

4.2.2. Funciones de fase

Las funciones de fase implementadas presentan diferentes direcciones de muestreo como las que se muestran en la Figura 3.12. La función de fase de Chandrasekhar [Ecuación 3.8] y la función de fase de Rayleigh [Ecuación 3.7] tienen una distribución de muestreo isotrópica, es decir, todas las direcciones de muestreo son equiprobables. Mientras que en la función de fase de Henyey-Greenstein [Ecuación 3.10] se ha utilizado una distribución anisotrópica con coeficiente $k = 0.76$ (*forward*), es decir, la dirección de muestreo elegida variará poco respecto de la dirección original.

4.2.3. Cámara ojo de pez

Finalmente, se ha implementado una cámara de ojo de pez para obtener imágenes con una apertura desde 0° hasta 360° , es decir, de toda la esfera unitaria S^2 . Para su implementación se han seguido las indicaciones de Bourke [Bou01].

Capítulo 5

Análisis del modelo

En este capítulo se analizan los distintos valores y coeficientes tanto del modelo molecular como de los aerosoles del modelo desarrollado. En primer lugar, se muestran las gráficas correspondientes a la parte molecular.

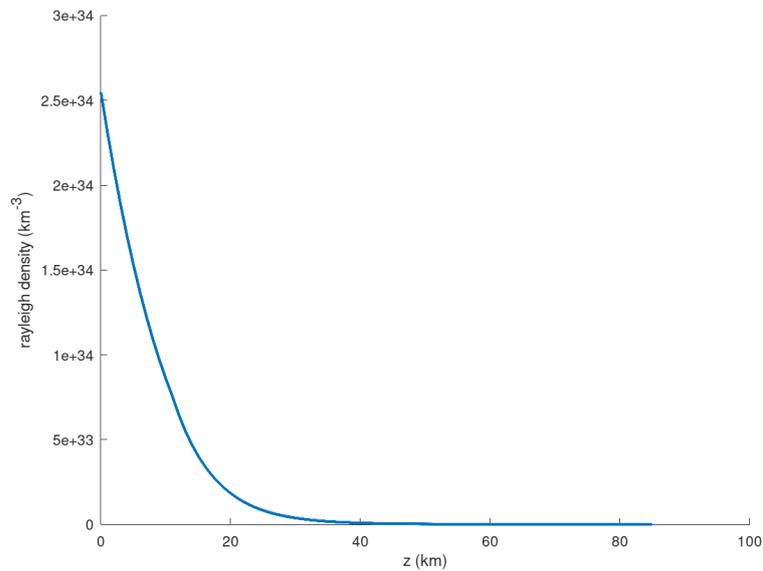


Figura 5.1: En el eje y aparecen las concentraciones del *scattering* molecular $C_s^m(z)$ en km^{-3} para las altitudes z en km del eje x .

Como se puede observar en la Figura 5.1 y como ya se había comentado al analizar el modelo en capítulos anteriores, la mayor parte de los gases responsables del *scattering* molecular se encuentra en los primeros 20 km de altitud, y como se aprecia en la Figura 5.2 la mayor parte del ozono responsable de la absorción molecular reside en la capa de ozono.

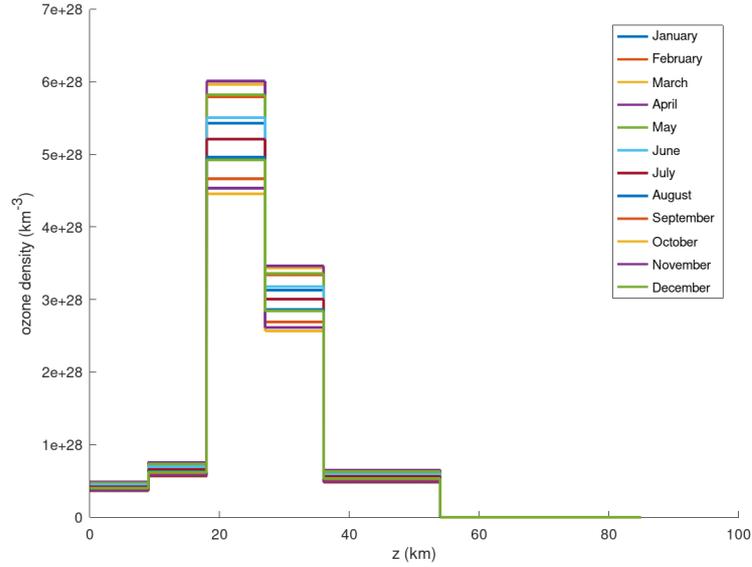


Figura 5.2: En el eje y aparecen las concentraciones de la absorción molecular $C_a^o(z, t)$ en km^{-3} para cada mes del año y dadas las altitudes z en km del eje x .

A continuación, se muestran las gráficas correspondientes a las secciones transversales del modelo molecular en función de las longitudes de onda λ .

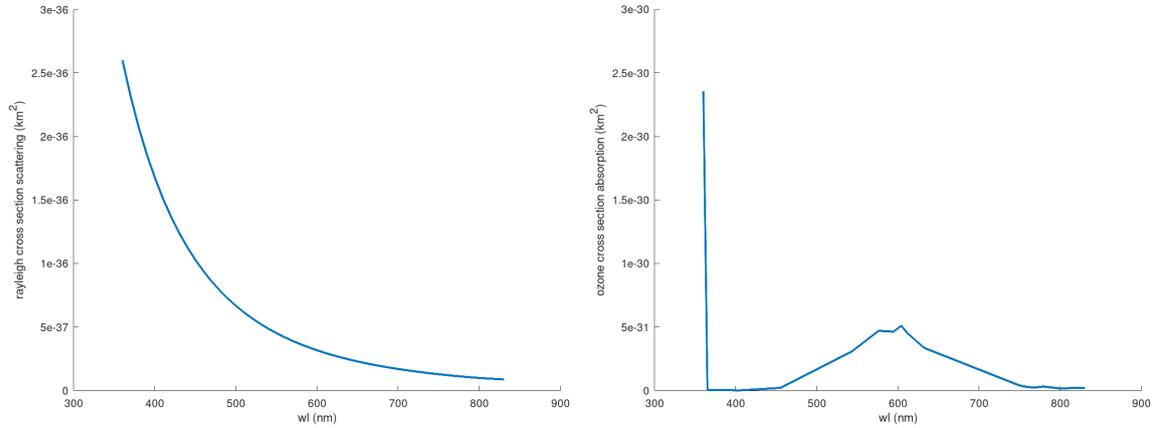


Figura 5.3: La gráfica de la izquierda corresponde a la sección transversal del *scattering* molecular $\sigma_s^m(\lambda)$ y la de la derecha a la sección transversal de la absorción molecular $\sigma_a^o(\lambda)$. En el eje y aparecen las secciones transversales en km^2 y en el eje x las longitudes de onda λ en nm .

Se puede deducir de las gráficas de la Figura 5.3 que la mayor parte de la dispersión producida por el *scattering* molecular ocurre en las longitudes de onda λ correspondientes al color azul, y la mayor parte de la absorción producida por el ozono ocurre en los colores ultravioleta.

En el análisis siguiente se utiliza el modelo de aerosoles *Desert-Dust* con una turbididad media. Las gráficas corresponden a las concentraciones en función de las altitudes z y a las secciones transversales en función de las longitudes de onda λ . Como

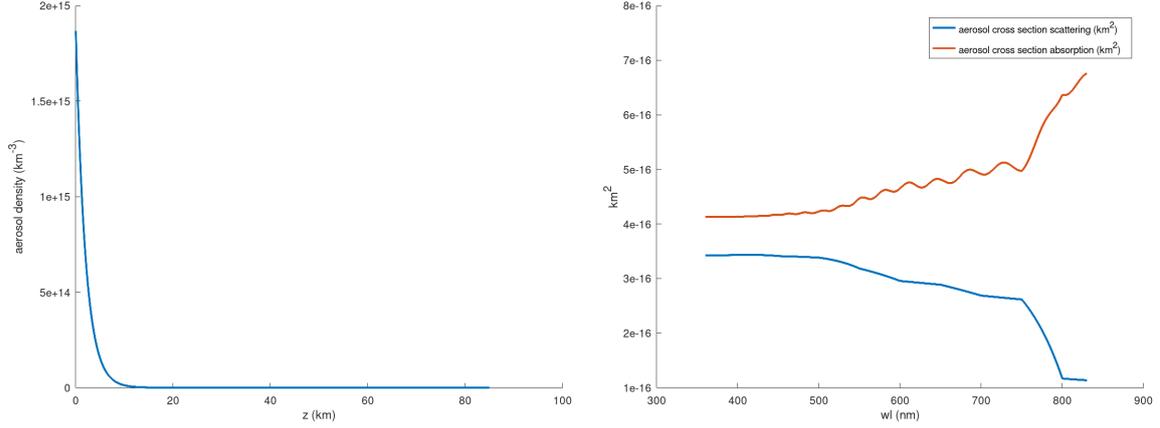


Figura 5.4: La gráfica de la izquierda corresponde a las concentraciones para el modelo de aerosoles *Desert-Dust* $C^a(z)$ en km^{-3} en el eje y para las altitudes z en km del eje x . La gráfica de la derecha corresponde a las secciones transversales de los aerosoles en km^2 en el eje y para las longitudes de onda λ en nm del eje x [en azul las correspondientes al *scattering* $\sigma_s^a(\lambda)$ y en rojo las correspondientes a la absorción $\sigma_a^a(\lambda)$].

se puede observar en la Figura 5.4, la concentración de los aerosoles *Desert-Dust* baja considerablemente en los primeros 10 km de altitud. A continuación, se muestra una gráfica con todos los coeficientes del modelo anterior y el molecular para el mes de enero, con longitud de onda $\lambda = 500 nm$ y en función de las altitudes z .

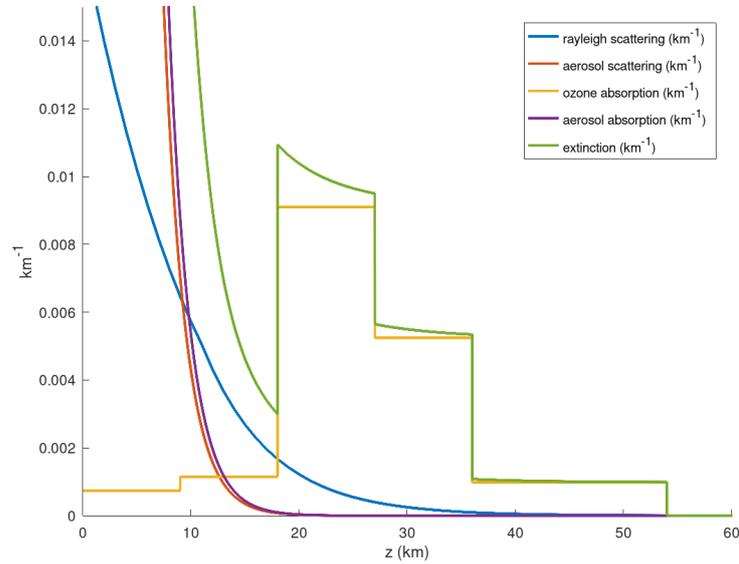


Figura 5.5: En el eje y aparecen, en km^{-1} , los coeficientes de *scattering* molecular $\mu_s^m(z, \lambda = 500nm)$ en azul, *scattering* de los aerosoles $\mu_s^a(z, \lambda = 500nm)$ en rojo, absorción molecular $\mu_a^o(z, t = January, \lambda = 500nm)$ en amarillo, absorción de los aerosoles $\mu_a^a(z, \lambda = 500nm)$ en morado y extinción total $\mu_t(z, t = January, \lambda = 500nm)$ en verde. En el eje x aparecen las altitudes z en km .

Como se puede apreciar en la Figura 5.5, en los primeros 10 km de altitud tienen más peso los coeficientes de los aerosoles, mientras que a partir de ese punto predominan

los coeficientes moleculares. Todos los aerosoles tienen este tipo de comportamiento, pero en mayor o menor medida dependiendo de la turbidez y de la altitud máxima que alcanzan sus concentraciones.

Capítulo 6

Resultados

La ejecución de Mitsuba 2 con el modelo atmosférico desarrollado se ha ejecutado sobre un ordenador con sistema operativo Windows 10 con dos procesadores Intel Xeon Gold 6140 de 16 núcleos y 32 hilos cada uno, y con 256 GB de memoria RAM. Cada *render* ha tardado entre 4 y 8 horas en ser generado, con 128 interacciones máximas en cada camino y 121 muestras por píxel utilizando la técnica de muestreo *stratified*.

Se han probado diferentes situaciones para comprobar el correcto funcionamiento del modelo, tanto desde el espacio como desde dentro de la atmósfera. En primer lugar, se muestra una imagen desde el espacio tomada con una cámara *pinhole* convencional y utilizando únicamente el modelo molecular (sin aerosoles), donde se puede apreciar el color azulado de la atmósfera.

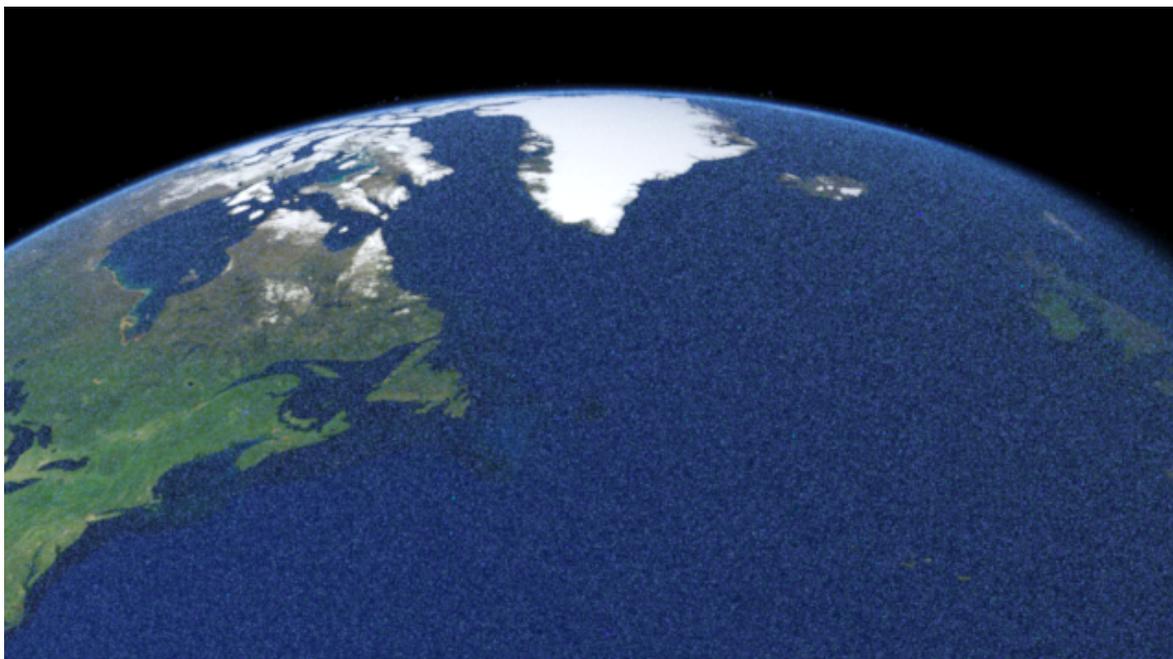


Figura 6.1: *Render* desde el espacio y sin aerosoles.

De la Figura 6.1 se puede comprobar que el color azulado de la atmósfera está

determinado por los gases que la componen. La textura utilizada para envolver a la esfera de la Tierra se ha descargado de la página web de la NASA¹.

A continuación, se muestra una comparación de *renders* realizados desde el espacio captando toda la Tierra y con una cámara *pinhole* convencional. En estas imágenes se puede comprobar como varía la turbiedad de la atmósfera con el modelo de aerosoles *Desert-Dust*.



Figura 6.2: *Renders* desde el espacio, con el modelo de aerosoles *Desert-Dust* y diferentes niveles de turbiedad. La turbiedad aumenta de izquierda a derecha.

Como se puede observar en la Figura 6.2, el grosor de la atmósfera es muy pequeño comparado con el de la Tierra, por lo que las demás comparativas se han hecho desde dentro de la atmósfera para obtener un detalle mayor. A continuación, se muestran *renders* realizados desde dentro de la atmósfera y con una cámara *pinhole* convencional. Estas imágenes no contienen aerosoles y se puede apreciar cómo el color de la atmósfera está definido principalmente por los gases, con colores azules claros cuando los rayos solares inciden con un ángulo de 90° sobre la superficie terrestre y con colores más oscuros y ligeramente anaranjados cuando el Sol se desvanece.

¹NASA - Visible Earth: <https://visibleearth.nasa.gov/collection/1484/blue-marble>

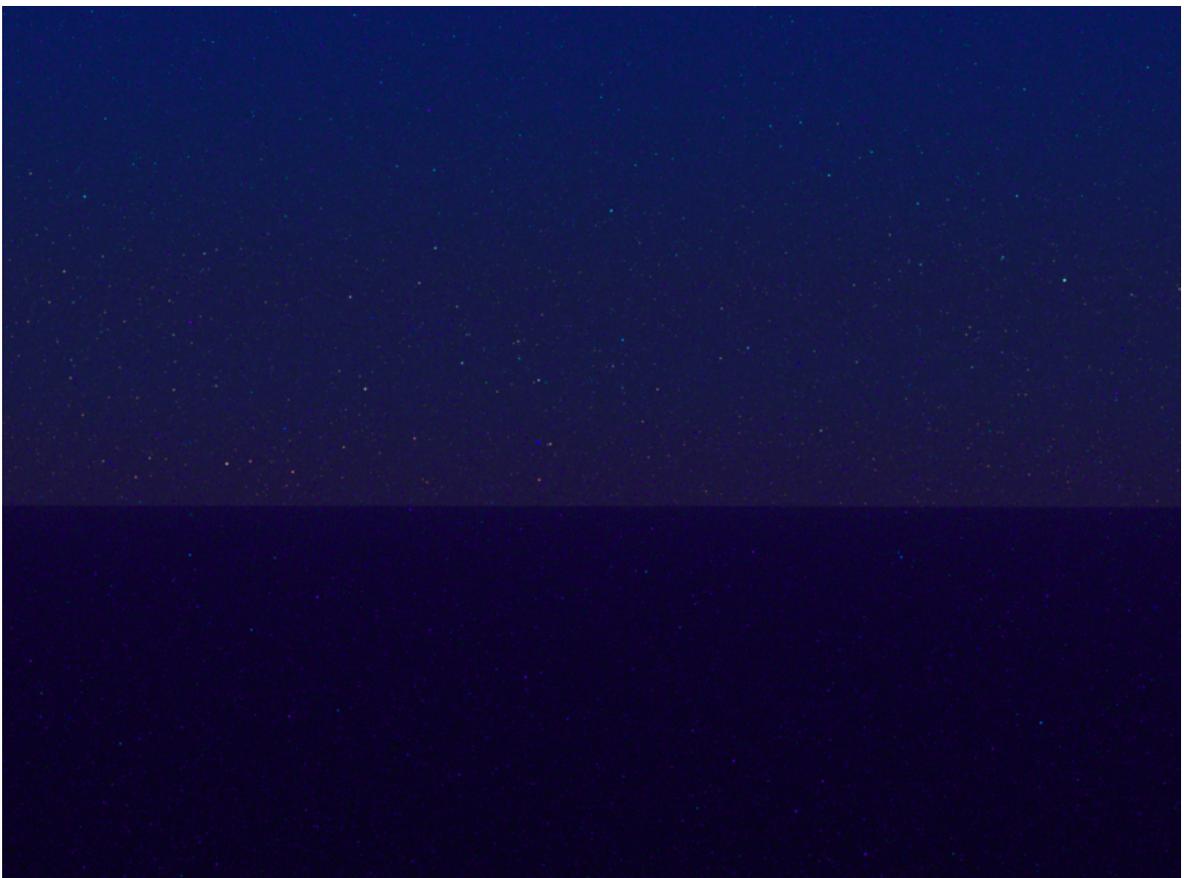


Figura 6.3: *Renders* realizados sin aerosoles y en el mes de febrero. En la imagen superior el Sol está a 90° respecto del horizonte, y en la imagen inferior está a 180° .

En las imágenes tomadas con una cámara *pinhole* convencional [Figura 6.3] no se puede apreciar todo el cielo dentro de la hemiesfera, por lo que en las siguientes imágenes se muestran *renders* realizados desde dentro de la atmósfera, con la cámara de ojo de pez y sin aerosoles. En esta comparativa se puede ver cómo varía el color de la atmósfera al cambiar la posición del Sol, desde un azul claro cuando el Sol está alineado con el eje vertical hasta colores más oscuros cuando está a punto de anochecer, e incluso tonos morados justo cuando ha anochecido.

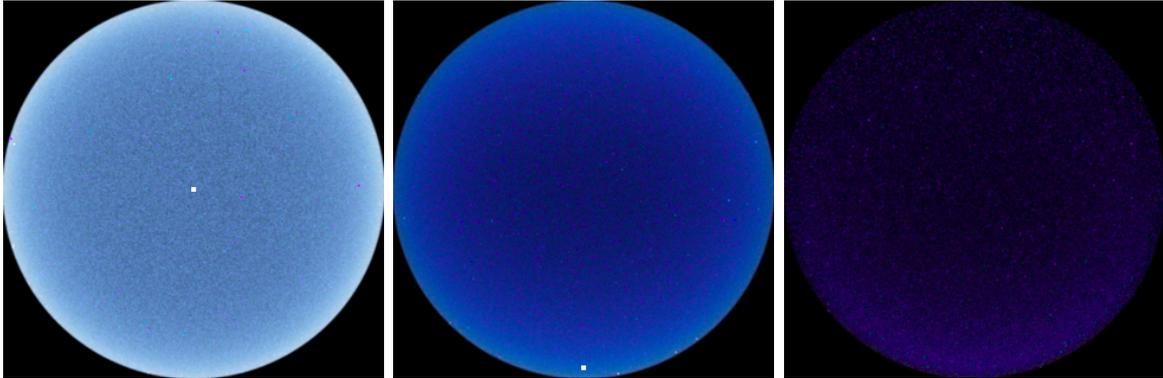


Figura 6.4: *Renders* realizados sin aerosoles y en el mes de enero. En la imagen de la izquierda el Sol está a 90° respecto del horizonte, en la del centro está a 5° y en la de la derecha a -5° .

En la Figura 6.4 se muestran cielos sin aerosoles, por lo que no se llega a apreciar un cielo del todo realista. A continuación, se muestran *renders* desde el entorno bioclimático *Desert-Dust*, desde donde se puede observar cómo el Sol queda más difuminado y según varía la turbididad de los aerosoles en la atmósfera, la luz que llega a la cámara se reduce considerablemente [Figura 6.5].

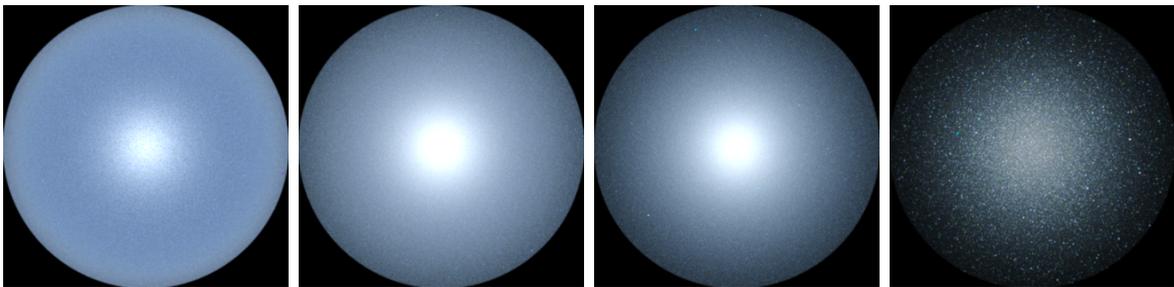


Figura 6.5: *Renders* realizados con el modelo de aerosoles *Desert-Dust*, en el mes de enero y con el Sol a 90° respecto del horizonte. La turbididad de la atmósfera aumenta de izquierda a derecha.

En las imágenes siguientes se muestran *renders* desde diferentes entornos bioclimáticos con diferente turbididad de los aerosoles.

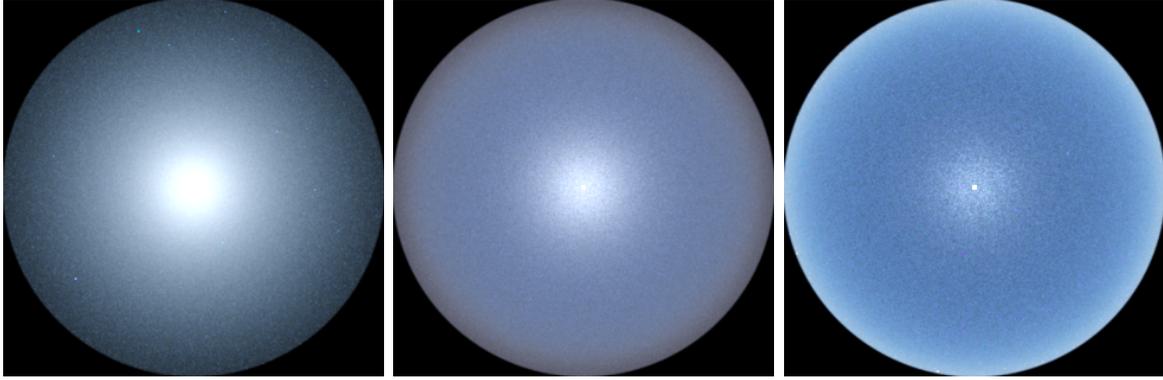


Figura 6.6: *Renders* realizados con el Sol a 90° respecto del horizonte y en el mes de enero. La imagen de la izquierda está realizada con el modelo de aerosoles *Desert-Dust*, la del centro con *Maritime-Mineral* y la de la derecha con *Polar-Artic*.

Como se puede observar en la Figura 6.6, dependiendo de la composición de los aerosoles, y su concentración en la atmósfera, varían los colores del cielo. A continuación, se muestra una comparativa con diferentes *renders* desde diferentes entornos bioclimáticos y en diferentes meses del año.

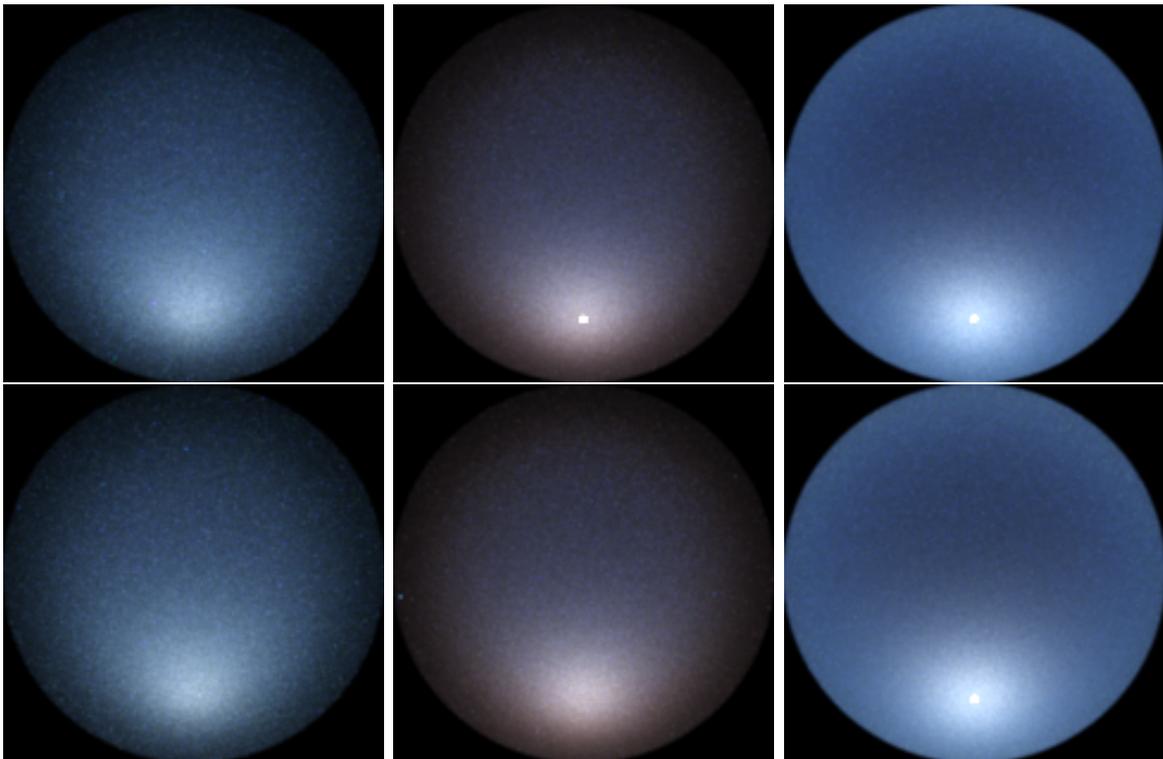


Figura 6.7: *Renders* realizados con el Sol a 30° respecto del horizonte. Las imágenes de la primera fila están tomadas en el mes de enero y las de la segunda fila en el mes de noviembre. Las imágenes de la izquierda corresponden con el modelo de aerosoles *Desert-Dust*, las del centro con *Maritime-Mineral* y las de la derecha con *Polar-Artic*.

Como las imágenes de la Figura 6.7 presentan una atmósfera con bastantes aerosoles, no se puede apreciar con claridad la diferencia entre las concentraciones de

ozono. A continuación, se muestran dos imágenes con el modelo de aerosoles *Background* y con baja turbididad.

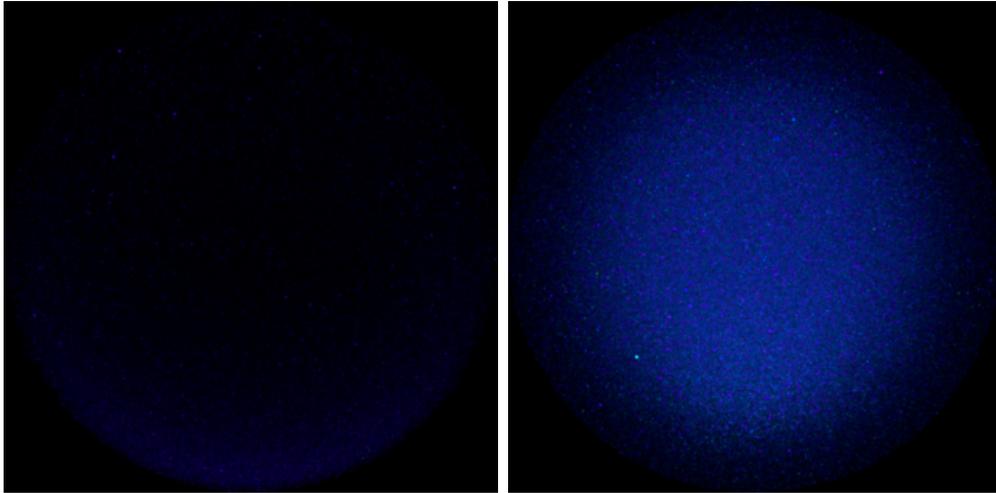


Figura 6.8: *Renders* realizados con el Sol a -5° respecto del horizonte. La imagen de la izquierda está tomada en el mes de enero y la de la derecha en el mes de noviembre.

Como se puede observar en los *renders* de la Figura 6.8, la imagen tomada en el mes de enero tiene mayor densidad de ozono, que absorbe mayor cantidad de luz, por lo que está más oscura que la imagen tomada en el mes de noviembre que tiene una cantidad de ozono menor.

Finalmente, se muestra una comparación entre una fotografía real extraída de Hosek y Wilkie [HW12] y un *render* con modelo de aerosoles *Desert-Dust* y baja turbididad [Figura 6.9].

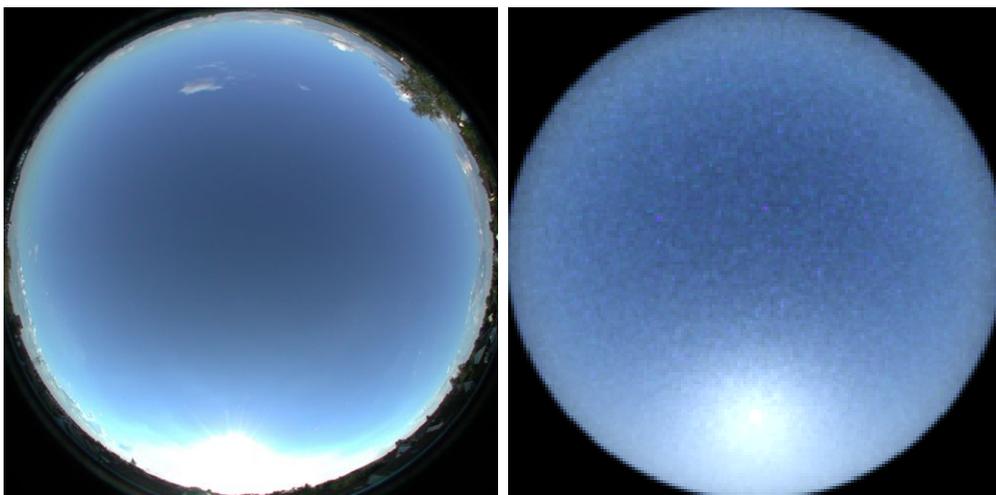


Figura 6.9: Comparación entre una fotografía real (derecha) y un *render* (izquierda).

Capítulo 7

Conclusiones

En este trabajo se ha conseguido simular el comportamiento del transporte de la luz en la atmósfera terrestre, tanto desde dentro de la atmósfera como desde el espacio, a partir de una gran colección de trabajos previos y del código de un motor de *render* profesional. Ha sido todo un reto comprender las bases físicas que sostienen el transporte de la luz en medios atmosféricos y desarrollarlas partiendo del código de Mitsuba 2, que es completamente genérico. Esto se debe a las diferentes características que incluye en su compilación y permiten su ejecución de manera vectorizada o escalar, tanto en CPU como en GPU, por lo que utiliza la biblioteca Enoki¹ para la mayoría de los tipos de dato y funciones. Implementar el modelo atmosférico utilizando los tipos de dato y sus métodos genéricos ha supuesto la mayoría de los errores debido a que, dependiendo de los parámetros establecidos en la compilación, cambian las funciones en las que se pueden utilizar estos tipos de dato (p.e.: si el tipo de dato base es *float* su comparación con otro *float* devuelve un *booleano*, en cambio si está vectorizado devuelve un vector de *booleanos*). Esto, sumado a la documentación incompleta para el desarrollo de Mitsuba 2² supone una gran curva de aprendizaje inicial. Además, se ha intentado acceder a más artículos de los mencionados en la bibliografía, pero, a pesar de que se trata del desarrollo de un trabajo académico, no estaban disponibles en las bibliotecas o había que pagar por ellos.

Independientemente de estas dificultades, me ha resultado muy interesante poder llevarlo a cabo y, especialmente, comprender los resultados de manera visual. Ya que en la mayoría de los trabajos que he hecho hasta la fecha los resultados se mostraban por consola o en un fichero, y simular el comportamiento del transporte de la luz en la atmósfera no tendría sentido si no se pudieran visualizar los resultados en forma de imágenes. A pesar de que me haya parecido interesante, me ha resultado complicado,

¹Enoki: <https://enoki.readthedocs.io/en/master/index.html>

²Mitsuba 2 - Render API Reference: https://mitsuba2.readthedocs.io/en/latest/generated/render_api.html#medium

sobre todo al principio, ya que cuando las imágenes no salen como esperas (o incluso son negras) resulta desesperante descubrir si el problema es que has entendido mal el modelo físico, has utilizado unidades incorrectas o simplemente has puesto la cámara mirando a la nada (menos mal que Adrián me ha explicado varias veces los principios físicos y ayudado cuando los resultados no tenían sentido). Este trabajo lo he realizado a tiempo completo desde el último examen que tuve a principios de septiembre de 2020 y he seguido el cronograma de la Figura 7.1 durante el desarrollo del mismo (además he tenido reuniones y correos electrónicos con Adrián todas las semanas, y charlas del grupo de investigación *Graphics and Imaging Lab*).



Figura 7.1: Cronograma.

Este proyecto se podría continuar de dos maneras: la primera desde el punto de vista físico y la segunda desde el punto de vista comercial.

El siguiente paso para continuar con este proyecto desde el punto de vista físico sería, en primer lugar, utilizar las funciones de fase de Mie en lugar de la de Henyey-Greenstein para el modelo de aerosoles, ya que Henyey-Greenstein no depende de la longitud de onda λ del rayo de luz. Además, se podrían añadir otros efectos atmosféricos como los arcoíris o las auroras boreales, y poder modelar nubes con una mayor turbididad en lugares concretos.

Mientras que el siguiente paso para continuar con este proyecto desde el punto de vista comercial sería optimizarlo para su uso en tiempo real y así poder utilizarlo en un motor de juego. De esta manera se podrían explorar videojuegos donde haya una atmósfera realista, se permita el transcurso del tiempo y se vea cómo el color del cielo varía en función de los biomas y la fecha. También sería interesante utilizar los datos atmosféricos de otros planetas, tanto del Sistema Solar como inventados, para simular el transporte de la luz en entornos completamente diferentes a la Tierra.

Bibliografía

- [AIA99] AIAA. *Guide to Global Aerosol Models*. 1801 Alexander Bell Drive, Reston, VA 22091: American Institute of Aeronautics y Astronautics, 1999.
- [Atm76] US Standard Atmosphere. “NOAA, NASA, USAF”. En: *Washington DC* (1976).
- [BN08] Eric Bruneton y Fabrice Neyret. “Precomputed atmospheric scattering”. En: *Computer Graphics Forum*. Vol. 27. 4. Wiley Online Library. 2008, págs. 1079-1086.
- [Bou01] Paul Bourke. “Computer Generated Angular Fisheye Projections”. En: (2001). URL: <http://paulbourke.net/dome/fisheye/>.
- [Bru17] Eric Bruneton. “A Qualitative and Quantitative Evaluation of 8 Clear Sky Models”. En: *IEEE transactions on visualization and computer graphics* 23.12 (2017), págs. 2641-2655.
- [Buc95] Anthony Bucholtz. “Rayleigh-scattering calculations for the terrestrial atmosphere”. En: *Applied Optics* 34.15 (1995), págs. 2765-2773.
- [Cha50] S Chandrasekhar. “Radiative transfer”. En: *Chandrasekhar, S* (1950).
- [CIE04] S CIE. *011/E: 2003 Spatial distribution of daylight-CIE standard general sky*. 2004.
- [Cos+20] Jonathas Costa y col. *Interactive Visualization of Atmospheric Effects for Celestial Bodies*. 2020. arXiv: 2010.03534 [cs.HC].
- [DSK91] Guillaume A. D’Almeida, Eric P. Shettle y Peter. Koepke. *Atmospheric aerosols : global climatology and radiative characteristics*. A. Deepak Pub Hampton, Va., USA, 1991. ISBN: 0937194220.
- [Düt74] HU Dütsch. “The ozone distribution in the atmosphere”. En: *Canadian journal of chemistry* 52.8 (1974), págs. 1491-1504.
- [EK10] Oskar Elek y Petr Kmoch. “Real-time spectral scattering in large-scale natural participating media”. En: *Proceedings of the 26th Spring Conference on Computer Graphics*. ACM. 2010, págs. 77-84.
- [GGJ18] David Guimera, Diego Gutierrez y Adrian Jarabo. “A Spatio-Temporal Physically-Based Sky Model”. En: *Proceedings of CEIG ’18*. 2018.
- [Gor+14] V Gorshchev y col. “High spectral resolution ozone absorption cross-sections—Part 1: Measurements, data analysis and comparison with previous measurements around 293 K”. En: *Atmospheric Measurement Techniques* 7.2 (2014), págs. 609-624.

- [Gut+04] Diego Gutierrez y col. “Chasing the green flash: a global illumination solution for inhomogeneous media”. En: *Proceedings of the 20th spring conference on Computer graphics*. ACM. 2004, págs. 97-105.
- [HG41] L. G. Henyey y J. L. Greenstein. “Diffuse radiation in the Galaxy”. En: *Astrophysical Journal* 93 (ene. de 1941), págs. 70-83. DOI: 10.1086/144246.
- [Hil20] Sébastien Hillaire. “A Scalable and Production Ready Sky and Atmosphere Rendering Technique”. En: *Computer Graphics Forum* 39 (jul. de 2020), págs. 13-22. DOI: 10.1111/cgf.14050.
- [HMS05] Jörg Haber, Marcus Magnor y Hans-Peter Seidel. “Physically-based simulation of twilight phenomena”. En: *ACM Transactions on Graphics (TOG)* 24.4 (2005), págs. 1353-1373.
- [Hul57] Hendrik Christoffel Hulst. *Light scattering by small particles*. Courier Corporation, 1957.
- [HW12] Lukas Hosek y Alexander Wilkie. “An analytic model for full spectral sky-dome radiance”. En: *ACM Transactions on Graphics (TOG)* 31.4 (2012), pág. 95.
- [HW13] Lukas Hosek y Alexander Wilkie. “Adding a Solar-Radiance Function to the Hosek-Wilkie Skylight Model”. En: *Computer Graphics and Applications, IEEE* 33.3 (2013), págs. 44-52.
- [Jae93] R. Jaenicke. *Tropospheric Aerosols in Aerosol-Cloud-Climate Interactions*, P.V. Hobbs, Academic Press, 1993.
- [JAG18] Adrian Jarabo, Carlos Aliaga y Diego Gutierrez. “A Radiative Transfer Framework for Spatially-Correlated Materials”. En: *ACM Transactions on Graphics* 37.4 (2018).
- [Jen+01] Henrik Wann Jensen y col. “A physically-based night sky model”. En: *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. ACM. 2001, págs. 399-408.
- [Kut12] Peter Z Kutz. “Physically-Based Atmosphere Rendering”. En: (2012).
- [LEN12] Jean-François Lalonde, Alexei Efros y Srinivasa Narasimhan. “Estimating Natural Illumination from a Single Outdoor Image”. En: sep. de 2012, págs. 183-190.
- [Nim+19] Merlin Nimier-David y col. “Mitsuba 2: a retargetable forward and inverse renderer”. En: *ACM Transactions on Graphics* 38 (nov. de 2019), págs. 1-17. DOI: 10.1145/3355089.3356498.
- [Nis+93] Tomoyuki Nishita y col. “Display of the earth taking into account atmospheric scattering”. En: *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*. ACM. 1993, págs. 175-182.
- [Nov+18] Jan Novák y col. “Monte Carlo Methods for Volumetric Light Transport Simulation”. En: *Computer Graphics Forum (Proceedings of Eurographics - State of the Art Reports)* 37.2 (mayo de 2018). To appear.
- [ONe05] Sean O’Neil. “Accurate atmospheric scattering”. En: *GPU Gems* 2 (2005), págs. 253-268.

- [PSM93] Richard Perez, Robert Seals y Joseph Michalsky. “All-weather model for sky luminance distribution—preliminary configuration and validation”. En: *Solar energy* 50.3 (1993), págs. 235-245.
- [PSS99] Arcot J Preetham, Peter Shirley y Brian Smits. “A practical analytic model for daylight”. En: *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co. 1999, págs. 91-100.
- [RK53] KR Ramanathan y RN Kulkarni. “Height distribution of atmospheric ozone”. En: *Proceedings of the Indian Academy of Sciences-Section A*. Vol. 37. 2. Springer. 1953, págs. 321-331.
- [Str71] John William Strutt. “On the scattering of light by small particles”. En: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 41.275 (1871).
- [Wil+04] Alexander Wilkie y col. “An analytical model for skylight polarisation”. En: *Rendering Techniques*. 2004, págs. 387-398.
- [Zha+20] Cheng Zhang y col. “Path-space differentiable rendering”. En: *ACM Transactions on Graphics* 39 (jul. de 2020). DOI: 10.1145/3386569.3392383.
- [Zim+89] PH Zimmermann y col. “A global three-dimensional source-receptor model investigation using ^{85}Kr ”. En: *Atmospheric Environment* 23.1 (1989), págs. 25-35.

Lista de Figuras

1.1.	<i>Render</i> de la atmósfera terrestre vista desde el espacio.	1
2.1.	Transporte de la luz en un medio participativo para una fotón. En cada interacción con el medio, la partícula puede ser absorbida o dispersada en otra dirección. La apariencia del medio es el resultado de la interacción de muchas de estas partículas al llegar al sensor (p.ej. el ojo). [Imagen cortesía de Adrián Jarabo]	4
2.2.	Ecuación de <i>render</i> volumétrica.	4
2.3.	Ampliación de las partículas que conforman el medio participativo para el punto \mathbf{x} . [Imagen cortesía de Adrián Jarabo]	5
2.4.	A la izquierda, una partícula absorbe parte de la radiancia espectral del rayo de luz (evento de absorción). En el centro, una partícula dispersa parte de la radiancia espectral en diferentes direcciones (evento de <i>out-scattering</i>). A la derecha, una partícula dispersa parte de la radiancia espectral proveniente de otros rayos de luz en la misma dirección que la del rayo de luz emitido (evento de <i>in-scattering</i>).	5
2.5.	Camino completo desde un emisor hasta la cámara atravesando distintos medios participativos e interaccionando con objetos [Nov+18].	6
2.6.	Valor de un píxel j en una imagen I	6
2.7.	Ecuación de radiancia espectral en un camino de longitud k para un píxel j	6
3.1.	Distribución de las capas de la atmósfera [GGJ18].	7
3.2.	Ecuación para calcular la concentración en el medio M	10
3.3.	Ecuación para calcular la sección transversal en el medio M	10
3.4.	Función de fase del medio M	10
3.5.	Coefficiente de absorción en la atmósfera.	10
3.6.	Coefficiente de <i>scattering</i> en la atmósfera.	10
3.7.	Función de fase atmosférica.	10
3.8.	Función de fase de <i>Rayleigh</i>	11

3.9. Función de fase de Chandrasekhar.	11
3.10. Función de concentración de los aerosoles.	12
3.11. Función de fase de Henyey-Greenstein.	13
3.12. Tipos de <i>scattering</i> anisotrópico e isotrópico [$g = k$]. [Imagen cortesía de Adrián Jarabo]	13
4.1. Probabilidad de utilizar la función de fase molecular f^m en la ruleta rusa.	17
4.2. Probabilidad de utilizar la función de fase de los aerosoles f^a en la ruleta rusa.	17
5.1. En el eje y aparecen las concentraciones del <i>scattering</i> molecular $C_s^m(z)$ en km^{-3} para las altitudes z en km del eje x	18
5.2. En el eje y aparecen las concentraciones de la absorción molecular $C_a^o(z, t)$ en km^{-3} para cada mes del año y dadas las altitudes z en km del eje x	19
5.3. La gráfica de la izquierda corresponde a la sección transversal del <i>scattering</i> molecular $\sigma_s^m(\lambda)$ y la de la derecha a la sección transversal de la absorción molecular $\sigma_a^o(\lambda)$. En el eje y aparecen las secciones transversales en km^2 y en el eje x las longitudes de onda λ en nm	19
5.4. La gráfica de la izquierda corresponde a las concentraciones para el modelo de aerosoles <i>Desert-Dust</i> $C^a(z)$ en km^{-3} en el eje y para las altitudes z en km del eje x . La gráfica de la derecha corresponde a las secciones transversales de los aerosoles en km^2 en el eje y para las longitudes de onda λ en nm del eje x [en azul las correspondientes al <i>scattering</i> $\sigma_s^a(\lambda)$ y en rojo las correspondientes a la absorción $\sigma_a^a(\lambda)$].	20
5.5. En el eje y aparecen, en km^{-1} , los coeficientes de <i>scattering</i> molecular $\mu_s^m(z, \lambda = 500nm)$ en azul, <i>scattering</i> de los aerosoles $\mu_s^a(z, \lambda = 500nm)$ en rojo, absorción molecular $\mu_a^o(z, t = January, \lambda = 500nm)$ en amarillo, absorción de los aerosoles $\mu_a^a(z, \lambda = 500nm)$ en morado y extinción total $\mu_t(z, t = January, \lambda = 500nm)$ en verde. En el eje x aparecen las altitudes z en km	20
6.1. <i>Render</i> desde el espacio y sin aerosoles.	22
6.2. <i>Renders</i> desde el espacio, con el modelo de aerosoles <i>Desert-Dust</i> y diferentes niveles de turbididad. La turbididad aumenta de izquierda a derecha.	23

6.3.	<i>Renders</i> realizados sin aerosoles y en el mes de febrero. En la imagen superior el Sol está a 90° respecto del horizonte, y en la imagen inferior está a 180°	24
6.4.	<i>Renders</i> realizados sin aerosoles y en el mes de enero. En la imagen de la izquierda el Sol está a 90° respecto del horizonte, en la del centro está a 5° y en la de la derecha a -5°	25
6.5.	<i>Renders</i> realizados con el modelo de aerosoles <i>Desert-Dust</i> , en el mes de enero y con el Sol a 90° respecto del horizonte. La turbiedad de la atmósfera aumenta de izquierda a derecha.	25
6.6.	<i>Renders</i> realizados con el Sol a 90° respecto del horizonte y en el mes de enero. La imagen de la izquierda está realizada con el modelo de aerosoles <i>Desert-Dust</i> , la del centro con <i>Maritime-Mineral</i> y la de la derecha con <i>Polar-Artic</i>	26
6.7.	<i>Renders</i> realizados con el Sol a 30° respecto del horizonte. Las imágenes de la primera fila están tomadas en el mes de enero y las de la segunda fila en el mes de noviembre. Las imágenes de la izquierda corresponden con el modelo de aerosoles <i>Desert-Dust</i> , las del centro con <i>Maritime-Mineral</i> y las de la derecha con <i>Polar-Artic</i>	26
6.8.	<i>Renders</i> realizados con el Sol a -5° respecto del horizonte. La imagen de la izquierda está tomada en el mes de enero y la de la derecha en el mes de noviembre.	27
6.9.	Comparación entre una fotografía real (derecha) y un <i>render</i> (izquierda).	27
7.1.	Cronograma.	29

Anexo A

Guía de uso

En este apartado se explica cómo compilar y utilizar Mitsuba 2 con el modelo atmosférico desarrollado.

A.1. Cómo compilar el proyecto

En primer lugar se debe descargar el código fuente del proyecto alojado en el repositorio de GitHub indicado en la Sección B.3 y abrir una terminal dentro del directorio del repositorio descargado. A continuación, se resume la manera de compilar el proyecto explicada con más detalle en la guía oficial de Mitsuba 2¹.

A.1.1. Linux

Para compilar el proyecto en Linux se deben tener instalados los paquetes siguientes. El gestor de paquetes que se utiliza es apt sobre Ubuntu.

```
# Install recent versions build tools, including Clang and libc++
↳ (Clang's C++ library)
sudo apt install -y clang-9 libc++-9-dev libc++abi-9-dev cmake
↳ ninja-build

# Install libraries for image I/O and the graphical user interface
sudo apt install -y libz-dev libpng-dev libjpeg-dev libxrandr-dev
↳ libxinerama-dev libxcursor-dev

# Install required Python packages
sudo apt install -y python3-dev python3-distutils python3-setuptools
```

Una vez instalados los paquetes necesarios, se compila el proyecto como se muestra a continuación.

¹Mitsuba 2 - Compiling the system: https://mitsuba2.readthedocs.io/en/latest/src/getting_started/compiling.html

```
# Ensure that two environment variables CC and CXX are exported
export CC=clang-9
export CXX=clang++-9

# Create a directory where build products are stored
mkdir build
cd build
cmake -GNinja ..
ninja
```

A.1.2. Windows

Para compilar el proyecto en Windows es necesario tener instalado Visual Studio 2019² con CMake y Python ≥ 3.6 .

```
# To be safe, explicitly ask for the 64 bit version of Visual Studio
cmake -G "Visual Studio 16 2019" -A x64
```

A continuación, hay que abrir el archivo *mitsuba.sln* generado y compilarlo con Visual Studio 2019 de la manera habitual.

A.2. Cómo utilizar el programa

Para utilizar Mitsuba 2 primero hay que configurar las variables de entorno ejecutando en una terminal lo siguiente.

```
# On Linux
source setpath.sh

# On Windows
C:/.../mitsuba2> setpath
```

Tras ejecutar el comando anterior, se puede lanzar el programa *mitsuba* para renderizar una escena definida en XML como se explica en la guía oficial³.

```
mitsuba scene.xml
```

Para utilizar el medio atmosférico desarrollado, se debe añadir el código siguiente a una escena.

²Visual Studio 2019: <https://visualstudio.microsoft.com/vs/>

³Mitsuba 2 - Scene file format: https://mitsuba2.readthedocs.io/en/latest/src/getting_started/file_format.html

```

<medium type="atmosphere" id="atmosphere">
  <float name="earth_radius" value="..."/>
  <point name="earth_center" x="..." y="..." z="..."/>
  <string name="molecular_phase_function" value="..."/>
  <integer name="month" value="..."/>
  <string name="aerosol_model" value="..."/>
  <float name="turbidity" value="..."/>
</medium>

<shape type="sphere">
  <float name="radius" value="..."/>
  <point name="center" x="..." y="..." z="..."/>

  <bsdf type="null"/>

  <ref id="atmosphere" name="interior"/>
</shape>

```

Donde radio de la esfera [*radius*] tiene que ser al menos 87 km mayor que el radio de la Tierra y el centro de la esfera [*center*] tiene que ser el mismo que el de la Tierra. Los parámetros de la atmósfera tienen que ser como se explica a continuación. El parámetro [*earth_radius*] es el radio de la Tierra. El parámetro [*earth_center*] es el centro de la esfera y el de la Tierra. El parámetro [*molecular_phase_function*] puede ser *cha* (por defecto) para utilizar como función de fase molecular la función de fase de Chandrasekhar y *rayleigh* para la de Rayleigh. El parámetro [*month*] corresponde al mes, entre 1 (enero, por defecto) y 12 (diciembre). El parámetro [*aerosol_model*] corresponde al modelo de aerosoles (entorno bioclimático), que por defecto no utiliza ninguno, y puede ser *BackgroundAerosol*, *DesertDustAerosol*, *MaritimeCleanAerosol*, *MaritimeMineralAerosol*, *PolarAntarticAerosol*, *PolarArticAerosol*, *RemoteContinentalAerosol*, *RuralAerosol* o *UrbanAerosol*. El parámetro [*turbidity*] corresponde a la turbididad de los aerosoles en forma de factor multiplicador de su concentración mayor o igual que 0 (por defecto es 1).

A continuación, se muestra un ejemplo XML de una escena donde el Sol está alineado con el eje Z, la cámara está 1 metro por encima del Polo Norte, se utilizan como unidades los kilómetros, se emplea la cámara de ojo de pez con una apertura de 180° alineada con el eje Z+ y el entorno bioclimático es *Desert-Dust*.

```

<scene version="2.2.1">
  <integrator type="volpath">
    <!-- Instantiate a path tracer with a max. path length of 128 -->
    <integer name="max_depth" value="128"/>
  </integrator>

```

```

</integrator>

<!-- Instantiate a fisheye camera with 180 degrees aperture -->
<sensor type="fisheye">
  <transform name="to_world">
    <!-- Translate the camera 1 meter above the ground at the north
    → pole -->
    <translate x="0" y="0" z="6356.767"/>
  </transform>
  <float name="aperture" value="180"/>

  <!-- Render with 121 samples per pixel using a stratified sampling
  → strategy -->
  <sampler type="stratified">
    <integer name="sample_count" value="121"/>
  </sampler>

  <!-- Generate an EXR image -->
  <film type="hdrfilm">
    <integer name="width" value="320"/>
    <integer name="height" value="320"/>
  </film>
</sensor>

<!-- Distance unit: km -->

<!-- Sun -->
<shape type="sphere">
  <float name="radius" value="696340"/>
  <point name="center" x="0" y="0" z="149597870.700"/> <!-- Distance
  → = 149597870700 meters -->

  <emitter type="area">
    <spectrum name="radiance" value="770000"/> <!-- 770000 W per
    → square unit per steradian -->
  </emitter>
</shape>

<medium type="atmosphere" id="atmosphere">
  <float name="earth_radius" value="6356.766"/>
  <point name="earth_center" x="0" y="0" z="0"/>
  <string name="aerosol_model" value="DesertDustAerosol"/>
  <float name="turbidity" value="0.001"/>
</medium>

<!-- Earth -->
<shape type="sphere">
  <float name="radius" value="6356.766"/>

```

```
<bsdf type="diffuse">
  <rgb name="reflectance" value="0, 0, 0"/>
</bsdf>
</shape>

<!-- Earth atmosphere -->
<shape type="sphere">
  <float name="radius" value="6443.77"/>
  <point name="center" x="0" y="0" z="0"/>

  <bsdf type="null"/>

  <ref id="atmosphere" name="interior"/>
</shape>

<shape type="sphere">
  <float name="radius" value="0.001"/>
  <point name="center" x="0" y="0" z="6356.767"/>
  <boolean name="flip_normals" value="true"/>

  <bsdf type="null"/>

  <ref id="atmosphere" name="interior"/>
</shape>
</scene>
```

Anexo B

Enlaces de interés

En este apartado se detallan los enlaces de interés respectivos al código fuente y a la licencia del mismo.

B.1. Mitsuba 2

El *commit* de GitHub del motor Mitsuba 2 del que se ha partido para realizar este trabajo:

<https://github.com/mitsuba-renderer/mitsuba2/tree/2fb0c9634e696887fae3921a60818a3a503c892e>

B.2. Licencia de uso

La licencia de uso del motor Mitsuba 2 para el *commit* anterior: <https://github.com/mitsuba-renderer/mitsuba2/blob/2fb0c9634e696887fae3921a60818a3a503c892e/LICENSE>

B.3. Código fuente

El código fuente de este proyecto está alojado en un repositorio público de GitHub con la licencia de Mitsuba 2 anterior:

<https://github.com/agavinm/atmosphere-renderer>