



Trabajo Fin de Grado

Modelo físico de apariencia para piel de reptiles
A physically-based appearance model for reptile skin

Autor

Diego Bielsa Monterde

Directores

Juan Raúl Padrón Griffé
Adolfo Muñoz Orbañanos

ESCUELA DE INGENIERÍA Y ARQUITECTURA
2022



DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD

(Este documento debe acompañar al Trabajo Fin de Grado (TFG)/Trabajo Fin de Máster (TFM) cuando sea depositado para su evaluación).

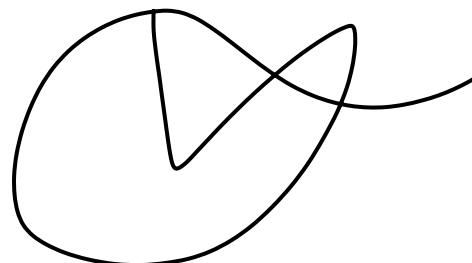
D./Dª. Diego Bielsa Monterde,

con nº de DNI 26058212V en aplicación de lo dispuesto en el art. 14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de (Grado/Máster)
Grado, (Título del Trabajo)
Modelo físico de apariencia para piel de reptiles

, es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada debidamente.

Zaragoza, 20 de septiembre de 2022



Fdo: Diego Bielsa Monterde

Resumen

En la computación gráfica es esencial el estudio de las propiedades ópticas de los materiales, es decir, la interacción de los mismos con las partículas de luz tanto a nivel ondulatorio como geométrico. En este proyecto se estudia más a fondo el primero de ellos concretamente la óptica de ondas, pues esta puede llegar a desencadenar procesos físicos como la interferencia que genera coloraciones de alto atractivo visual.

Es por ello que este estudio se centra en la reproducción de apariencia de reptiles teniendo en cuenta procesos físicos, más concretamente de la serpiente *Xenopeltis unicolor* caracterizada por la iridiscencia que producen sus escamas gracias a la interferencia de ondas. Para su modelado de apariencia se ha implementado un modelo multicapa, concretamente de tres capas; la más externa aportará la coloración característica de esta serpiente obtenida mediante iridiscencia; la capa intermedia modelará la absorción de la luz debido al paso por la superficie de las escamas; por último, la capa interna modelará el punto en el cual la luz ya no puede seguir su camino hacia el interior y que da el color natural a la serpiente. Es esencial para la apariencia a su vez el patrón de escamas seguido por las de esta serpiente, por lo que se verá modelada una textura que simule tal patrón.

Los datos para la generación de dicho modelo han sido obtenidos a base del estudio de diferentes documentos referentes tanto a las serpientes en su contexto biológico como al proceso de obtención de colores mediante coloración estructural. Además se mostrarán imágenes de los resultados obtenidos y será validado visualmente gracias a la comparación con imágenes reales.

Índice

1. Introducción	1
1.1. Motivación	1
1.2. Objetivos y Alcance	2
1.3. Desarrollo del Trabajo	3
1.4. Estructura de la Memoria	4
2. Fundamentos teóricos	7
2.1. Coloración Estructural	7
2.2. Coloración y Características Estructurales en Serpientes	8
2.3. Computación Gráfica	10
2.4. Tipos de Materiales	12
2.4.1. Material Multicapa	12
2.5. Bump Mapping	13
2.6. Algoritmos de Simulación	14
2.6.1. Path Tracing	14
2.6.2. Position-Free Monte Carlo	15
3. Desarrollo e Implementación	17
3.1. Entorno de Programación	17
3.1.1. Mitsuba	17
3.2. BSDF Final	18
3.3. Modelado de la Textura	21
3.4. Interferencia de Capa Fina	21
3.5. Capa de Absorción	23
3.6. Position-free Monte Carlo	26
4. Resultados y Validación	29
4.1. Resultados	29
4.2. Validación	34

5. Conclusiones y Trabajo Futuro	37
5.1. Conclusiones del Trabajo	37
5.2. Trabajo Futuro	38
5.3. Conclusiones Personales	38
6. Bibliografía	39
Lista de Figuras	41
Anexos	44
A. Materiales Básicos	47
B. Coloración y Características Estructurales en Serpientes	51
B.1. Características Generales	51
B.2. Xenopeltis Unicolor	52
C. Interferencia de Capa Fina	55
D. Path Tracing y Position-Free Monte Carlo	61
D.1. Path Tracing	61
D.1.1. Path Integral	62
D.1.2. Monte Carlo	62
D.1.3. Path Tracing	63
D.1.4. Luz Directa e Indirecta	64
D.2. Position-Free Monte Carlo	65

Capítulo 1

Introducción

*A lo largo de este primer capítulo se pretende exponer la motivación dada la cual se ha realizado este estudio de modelado de apariencia de la serpiente *Xenopeltis unicolor*, además se van a tratar los temas referentes a los objetivos y el desarrollo del trabajo.*

1.1. Motivación

El modelado de apariencia de pieles de reptiles de manera fiel a la realidad es una idea que interesa a varios campos de estudio y uso de los mismos. Entre ellos destacan las diferentes áreas de entretenimiento relacionadas con simulación por ordenador como el cine y la industria del videojuego. Con respecto a la simulación de su apariencia por ordenador se busca obtener el mayor realismo posible a las imágenes generadas, es por ello que aplicando conocimientos físicos sobre coloración se obtienen imágenes con colores altamente llamativos y cercanos a la realidad. Este método de obtención de apariencias basado en la física dota a las imágenes de colores difícilmente alcanzables sin la aplicación del mismo, por lo que ciertas industrias se llegarían a beneficiar de su uso; vease que este se podría transladar a películas como *Godzilla*, *Jurassic Park*, *El libro de la selva -live action-* y multitud de simulaciones por ordenador que requieran del uso de apariencias de pieles de reptiles como los videojuegos en los que estas aparezcan poniendo como ejemplo la saga *Far Cry*. Además, de conseguir un modelo de apariencia perfectamente realista, se podrían también llegar a entrenar sistemas de inteligencia artificial de reconocimiento de reptiles sin la necesidad de que estos sean reales, sino meras simulaciones.

El presente estudio se va a centrar en la apariencia de la serpiente *Xenopeltis unicolor* cuya principal característica se basa en la producción de colores ampliamente llamativos cuando la luz solar incide sobre sus escamas. Esto sucede gracias al fenómeno denominado iridiscencia a grandes rasgos definido como el cambio de color de una superficie con la variación de ángulo de visión y/o de incidencia de la luz sobre la misma,

debido a óptica de ondas. La Figura 1.1 muestra la especie de serpiente mencionada.



Figura 1.1: Serpiente de la especie *Xenopeltis unicolor* [1]

1.2. Objetivos y Alcance

El principal objetivo de este trabajo es simular la apariencia de la superficie de la serpiente *Xenopeltis unicolor* principalmente a nivel de coloración pero teniendo en cuenta a su vez el patrón hexagonal de sus escamas. El alcance de este proyecto incluye:

- Estudiar la coloración estructural en la naturaleza.
- Estudiar la estructura de las pieles de reptiles.
- Estudiar la estructura específica de la especie de serpiente *Xenopeltis unicolor* y su coloración estructural.
- Desarrollar un modelo multicapa que represente la apariencia de la serpiente *Xenopeltis unicolor*.
- Generar imágenes sintéticas del modelo de apariencia aplicado a un modelo 3D representativo.

- Validar las imágenes obtenidas visualmente con imágenes reales de la especie *Xenopeltis unicolor*.

1.3. Desarrollo del Trabajo

Con la finalidad de obtener los mejores resultados posibles y de entender los diversos mecanismos de coloración en la naturaleza este trabajo comienza con un estudio bibliográfico de la coloración estructural que se da en la misma. Posteriormente, se enfoca más concretamente en los mecanismos de coloración de la *Xenopeltis unicolor*.

Al estar el entorno teórico establecido, se pasa a la implementación propiamente dicha. Para ello se ha usado el renderizador *Mitsuba 0.6.0* [2] escrito en el lenguaje de programación C++, el cual aporta multitud de herramientas para el renderizado a la vez que optimizaciones y facilidades de acoplamiento de código sobre él, por lo que el primer paso de esta implementación es su instalación y posterior estudio a fondo tanto de la documentación como de la extensión del código base. Una vez se domina dicha herramienta ya se está en disposición de implementar el código que resuelva el problema tratado, por lo que se llevan a cabo las implementaciones necesarias para la obtención de colores obtenidos por razones físicas. Posteriormente se desarrolla un modelo multicapa realizando un proceso iterativo de adición y eliminación de posibles fenómenos que mejoren el modelo a la capa física previamente implementada.

Finalmente, una vez se ha obtenido el modelo asumido como mejor se procede a la validación del mismo de manera visual, es decir, se comparan las imágenes obtenidas con las pertenecientes a una *Xenopeltis unicolor* real.

En las Figuras 1.2 y 1.3 se pueden ver un diagrama de Gantt y un recuento de las horas por tarea (respectivamente) con el fin poder visualizar de una mejor manera el reparto tanto a nivel de fechas como de horas de las diferentes labores a realizar comentadas durante esta Sección.

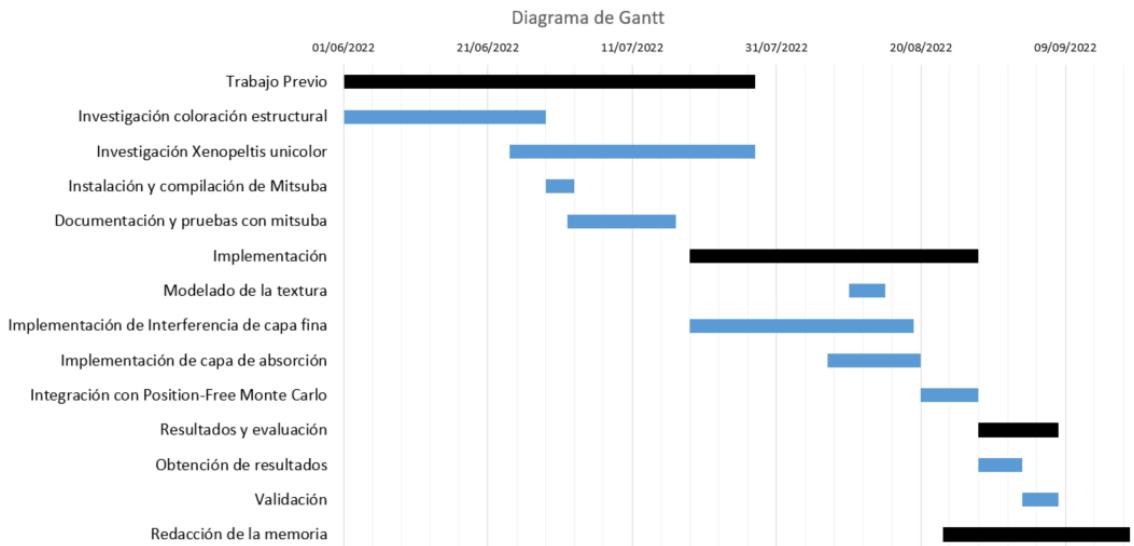


Figura 1.2: Diagrama de Gantt del proyecto

Tareas		Horas Dedicadas
Trabajo Previo	Investigación coloración estructural	33
	Investigación Xenopeltis unicolor	41
	Instalación y compilación de Mitsuba	7
	Documentación y pruebas con Mitsuba	18
Implementación	Modelado de la textura	13
	Implementación de interferencia de capa fina	60
	Implementación de capa de absorción	34
	Integración con Position-Free Monte Carlo	17
Resultados y Evaluación	Resultados	14
	Validación	4
Redacción	Memoria	50
Total		291

Figura 1.3: Horas dedicadas por tarea del proyecto

1.4. Estructura de la Memoria

El resto del documento posee una estructura concreta y consta de cuatro capítulos adicionales que exponen tanto conceptos necesarios a saber como el desarrollo e implementación llevados a cabo durante la realización de este trabajo. En primer lugar se encuentra el Capítulo 2, este es la base del estudio llevado a cabo y expone todo lo necesario a saber como son los temas biológicos necesarios, las bases de la computación gráfica y las propiedades físicas que se usan como base para la implementación del modelo de apariencia. Seguido a este se encuentra el Capítulo 3 en el cual se va a

poder ver el diagrama de clases que representa al código implementado y acoplado al renderizador *Mitsuba* [2] así como los diferentes pasos llevados a cabo en tema de implementación para la obtención del resultado final del trabajo. El Capítulo 4 se encarga de mostrar lo conseguido, enseñando multitud de imágenes de diferentes modelos geométricos con el modelo de apariencia final aplicado, modificando algunos parámetros para la obtención de variaciones en la coloración; eventualmente compara estas imágenes con otras de la especie *Xenopeltis unicolor* para la validación de las mismas, es decir, se valida si estas se asemejan o por el contrario no lo hacen. Para culminar el documento se encuentra el Capítulo 5, este recoge las conclusiones obtenidas una vez el trabajo se ha finalizado.

Capítulo 2

Fundamentos teóricos

A lo largo de este capítulo se explican los conceptos teóricos necesarios para la comprensión del trabajo en cuestión, centrándose en su importancia para la realización del trabajo, el problema que resuelven y los fundamentos básicos para su entendimiento. La estructura de este Capítulo 2 se centra en un principio en el estudio biológico realizado para posteriormente focalizarse en los fundamentos necesarios a conocer de la Computación Gráfica, tanto los básicos como los más avanzados utilizados en este trabajo.

2.1. Coloración Estructural

Para llegar a conocer las razones de la apariencia de la serpiente *Xenopeltis unicolor* y por ende conseguir un modelo de apariencia correcto es necesario como primer paso entender la coloración en la naturaleza y más concretamente la vinculada a óptica de ondas. Existe también la óptica geométrica que se centra en la explicación de la propagación de la luz concibiendo esta como rayos con trayectorias rectilíneas, pero no explica por completo todos los fenómenos de coloración que suceden en la naturaleza. Es por ello que la *Coloración Estructural* juega un gran papel en este trabajo, dando los conocimientos necesarios sobre diversos tipos de coloración y asentando las bases teóricas para el comienzo del modelado de apariencia en cuestión.

Estos fenómenos de *Coloración Estructural* suceden por la interacción de la luz con micro-estructuras en las pieles/plumas/escamas de los animales y fundamentalmente se producen por *interferencia de capa fina*, *interferencia multicapa*, *rejillas de difracción*, *cristales fotónicos* y *dispersión de la luz* [3].

Durante este proyecto es esencial el tipo de coloración por *interferencia de capa fina* que en resumen se produce cuando tenemos una capa fina de anchura comparable a la longitud de onda de la onda incidente y con un índice de refracción (cociente de la velocidad de la luz en el vacío y la velocidad de la luz en el medio sobre el que se

quiera calcular) diferente del medio externo e interno. Lo que va a suceder es que al ser capas que permiten el paso de la luz pero también su rebote, va a haber cantidades de luz que por ende se transmitan y reflejen periódicamente hasta que ya no le quede energía al rayo para seguir propagandose, lo que va a generar un patrón de reflexiones y transmisiones como el que se ve en la Figura 2.1a. Además, al ser la anchura de la capa comparable a longitud de onda, las reflexiones y transmisiones son tan consecutivas que se genera interferencia de ondas entre ellas lo que implica variaciones en la energía de los rayos y por tanto del color percibido. En la Figura 2.1b se puede ver un ejemplo de coloración por *interferencia de capa fina* en una burbuja, concretamente en los colores azulados y morados producidos. Para más detalle sobre el proceso físico que les ocurre a las ondas así como sobre las matemáticas que resuelven el fenómeno de *interferencia de capa fina* sobre las que se ha basado su implementación ver el Anexo C.

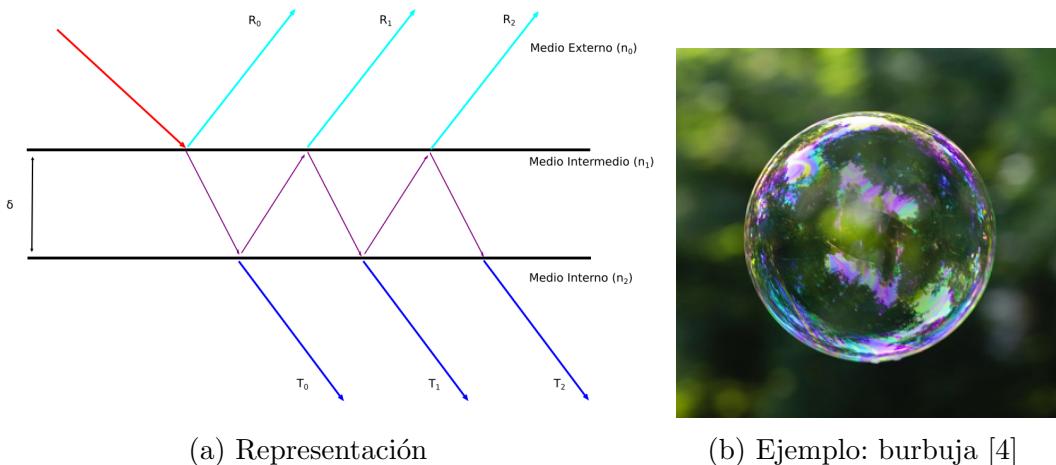


Figura 2.1: Interferencia de capa fina, representación del transporte de luz por las capas y ejemplo de la iridiscencia producida

A continuación, el estudio biológico se centra en la denominada *Coloración Estructural* que se da en serpientes así como en la estructura de su superficie tanto a gran escala como a pequeña escala, centrándose en primera instancia en lo común a la gran mayoría de las serpientes para despues concretar lo que sucede en la especie sujeta a estudio *Xenopeltis unicolor*.

2.2. Coloración y Características Estructurales en Serpientes

La *Coloración Estructural* mencionada anteriormente sucede en parte de la familia reptil [5, 6], interesando a lo largo de este estudio sobre todo la referente a serpientes.

Es por ello que para realizar un modelado de apariencia que se asemeje a la realidad es necesario el estudio de la superficie (escamas y su organización) en la familia de las serpientes con la finalidad de conocer la estructura y la causa de los colores llamativos que se pueden observar en algunas especies de ellas para su posterior simulación.

Estas superficies poseen características muy diferentes a las de otros seres vivos basadas en la existencia de escamas, desde su estructura estudiada a macro-escala a la estudiada a micro-escala.

Lo respectivo a la macro-estructura se centra en el patrón de escamas, que aunque similar en las diferentes especies de serpientes, no es idéntico en su totalidad sino que varía de una a otra. El patrón de escamas de la serpiente objeto de estudio *Xenopeltis unicolor* muestra escamas hexagonales con pequeñas irregularidades formando filas perfectamente encajadas con mínimas separaciones entre una escama y otra, tal y como se puede ver en la Figura 2.2.

Con respecto a la micro-estructura, es necesario centrarse en la capa más exterior de la superficie de las serpientes denominada *Oberhäutchen*. En lo más superior de esta es que se encuentran los denominados iridóforos encargados de aportar la iridiscencia mostrada en la parte derecha de la Figura 2.2 mediante *interferencia de capa fina* pues están formados por pequeñas láminas cristalinas que reflejan y refractan la luz. Justo debajo de estos iridóforos es que se encuentra una capa de pigmentos oscura, por la cual la luz penetra durante un corto recorrido hasta intersectar con el fin de la misma y comenzar su camino de vuelta a la superficie; por lo que esta capa de pigmentos se puede dividir en dos, una superficial que modele la absorción de la luz producida por su recorrido por la capa y una interior que modela el fondo de la capa.



Figura 2.2: Sección aumentada de la Figura 1.1

Tanto la macro-estructura como la micro-estructura se encuentran más en detalle en el Anexo B.

2.3. Computación Gráfica

La computación gráfica es la rama de la informática encargada de generar gráficos por ordenador, esto es, la generación de imágenes de objetos reales o inventados cuya apariencia se asemeje lo máximo posible a la realidad (esto no es siempre cierto pues a veces se deja de lado la realidad con el fin de buscar el atractivo visual) [7].

Para obtener una imagen se han de tener diversos componentes que hagan posible conseguirla, estos son la geometría que conformarán los objetos de la imagen con su material que modele su comportamiento; fuentes de luz que iluminen la geometría y permitan su visualización; cámara desde la cual vamos a ver la imagen generada; un plano sobre el que se van a proyectar las geometrías 3D a una pantalla en 2D; y por último un sistemas de coordenadas de 3 dimensiones sobre el que establecer las geometrías y diferentes objetos.

Con esto ya se tienen todos los componentes pero ahora hace falta que con ellos se obtenga la imagen requerida. Para obtener las geometrías 3D en el plano 2D comentado la solución usada en este proyecto se basa en la técnica conocida como trazado de rayos, en este método desde la cámara se lanzan rayos por cada punto del plano -conocido como pixel- que van intersectando con las diferentes geometrías obteniendo su posición.

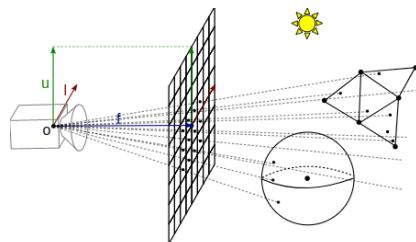


Figura 2.3: Componentes para trazado de rayos; se pueden ver las diferentes geometrías, la fuente de luz, la cámara, el plano, el sistema de coordenadas y los rayos trazados

Llegados a este punto se tiene que la geometría ya se encuentra en el plano, pero aún falta obtener el color de cada pixel. Para ello hay que resolver la conocida como *Ecuación de render* (2.1) pero antes hay que establecer una nomenclatura previa:

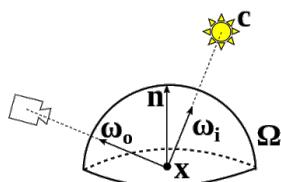


Figura 2.4: Elementos de la *Ecuación de Render*

- w_o es la dirección de observación negada (dirección contraria).

- x es la posición en el espacio.
- n es la normal de la superficie en x .
- Ω es la hemiesfera con radio unidad cuyo centro se situa en el punto observado x .
- w_i es la dirección del rayo de salida.

La *Ecuación de Render* se expresa como:

$$L_o(x, w_o) = L_e(x, w_o) + \int_{\Omega} L_i(x, w_i) f_r(x, w_i, w_o) |n \cdot w_i| dw_i \quad (2.1)$$

Donde:

- $L_o(x, w_o)$ determina la radiancia que le llega al observador cuando observa el punto x en la dirección w_o , siendo la radiancia la cantidad de luz por longitud de onda, es decir, el color.
- $L_e(x, w_o)$ determina la radiancia emitida por el punto x en la dirección w_o por un objeto emisor de luz, de no serlo, el valor será cero.
- La operación es integrada en el rango Ω , es decir, se evalua la integral en toda la hemiesfera orientada con respecto a la normal de la superficie.
- $L_i(x, w_i)$ es la radiancia que recibe el punto x desde la dirección w_i .
- $f_r(x, w_i, w_o)$ es el factor conocido como *BSDF* cuyas siglas significan *Bidirectional Scattering Distribution Function* (Función de Distribución de Dispersión Bidireccional). Este es el encargado de modular la manera en la que se distribuye la luz dadas la dirección incidente w_i y la de salida w_o y viene dado por cada tipo de material. El objetivo de este proyecto se centra en este factor de la ecuación, pues lo que se desea es la obtención de una *BSDF* que simule el comportamiento de la luz al contacto con la superficie de la serpiente a modelar que se puede ver en la imagen 1.1.
- $|n \cdot w_i|$ calcula el coseno del ángulo que forma la dirección a la fuente de luz desde x , nombrada como w_i , con la normal de la superficie n . Esto implica que contra menor sea el ángulo más cercano a 1 va a ser ese factor, por tanto mayor será la luz recibida, mientras que si el ángulo crece, el coseno se acerca a 0, disminuyendo por tanto la luz recibida. Este factor se denomina *Ley del Coseno de Lambert*.

2.4. Tipos de Materiales

Los tipos de materiales son esenciales en la computación gráfica, estos se encargan de modelar el comportamiento de la luz al intersectar el objeto formado por el material determinado; concretamente su objetivo es dar solución al segundo factor de la *Ecuación de Render* (2.1) - f_r - denominado *BSDF*. Como se ha comentado con anterioridad, el objetivo de este estudio se centra en la obtención de un material -y por ende de una *BSDF* que resuelva dicho factor- que modele la apariencia de la serpiente *Xenopeltis Unicolor* que se puede ver en la Figura 1.1.

Hay 3 materiales básicos cuya comprensión es esencial para entender el resto de ellos. Estos son los siguientes:

- Difuso: al intersectar la luz con él, esta es transmitida a partes iguales por toda la hemiesfera.
- Especular: al intersectar la luz con él, esta se refleja de manera perfecta.
- Dieléctrico: al intersectar la luz con él, esta o bien es reflejada al igual que un especular o bien se transmite al interior del material.

Para una descripción más detallada de los mismos así como para observar ejemplos de dichos materiales se puede ver el Anexo A.

2.4.1. Material Multicapa

Como se ha visto en la sección referente a la estructura de la serpiente *Xenopeltis unicolor*, esta está basada en la superposición de capas, por lo que la simulación de materiales multicapa es esencial para este trabajo. Estos materiales son los más comunes en la realidad por lo que su simulación es de gran importancia.

Los modelos de apariencia multicapa tratan de computar una *BSDF* la cual está compuesta por capas de otras *BSDF* que dejan pasar la luz a través de ellas hasta llegar a la más interna la cual solo permite reflexión, empemando la luz por tanto su camino de salida del material. Es por ello que el cómputo de la *BSDF* global implica el cómputo de las *BSDF* de todas las capas así como del camino entre capas seguido por la luz. Gracias a este problema surgen multitudes de aproximaciones que simulan el transporte de la luz entre capas para resolverlo [8, 9] centrando este estudio en el uso de la aproximación *Position-Free Monte Carlo* la cual se explica más adelante en la Sección 2.6.2. Estos algoritmos de aproximación del transporte de luz entre capas se encargan de modelar las funciones de evaluación y muestreo de manera que se tiene en cuenta que la posibilidad de reflexión entre ellas genera tanto cambios en la energía resultante como en las direcciones de los rayos.

2.5. Bump Mapping

Bump mapping (cuya traducción poco usual al castellano sería *mapeo de relieves*) es una técnica que aplica rugosidad a un material dada una textura. Esta técnica va a ser esencial para aplicar al modelo de apariencia las escamas, pues va a permitir que con una sencilla textura plana en 2D que modele su patrón se consiga desde un modelo geométrico liso uno cuya apariencia sea la de estar cubierto de las escamas dadas en la textura. Se basa en el cálculo del cambio de color de una textura para aplicarle a cada punto el gradiente -relieve- correspondiente.

Esta rugosidad es conseguida gracias a la perturbación de las normales de la superficie del objeto en cuestión durante el cálculo de la luminosidad, lo cual aplica tal sombreado al material que aparenta que este contiene relieve sin haber sido realmente modificado.

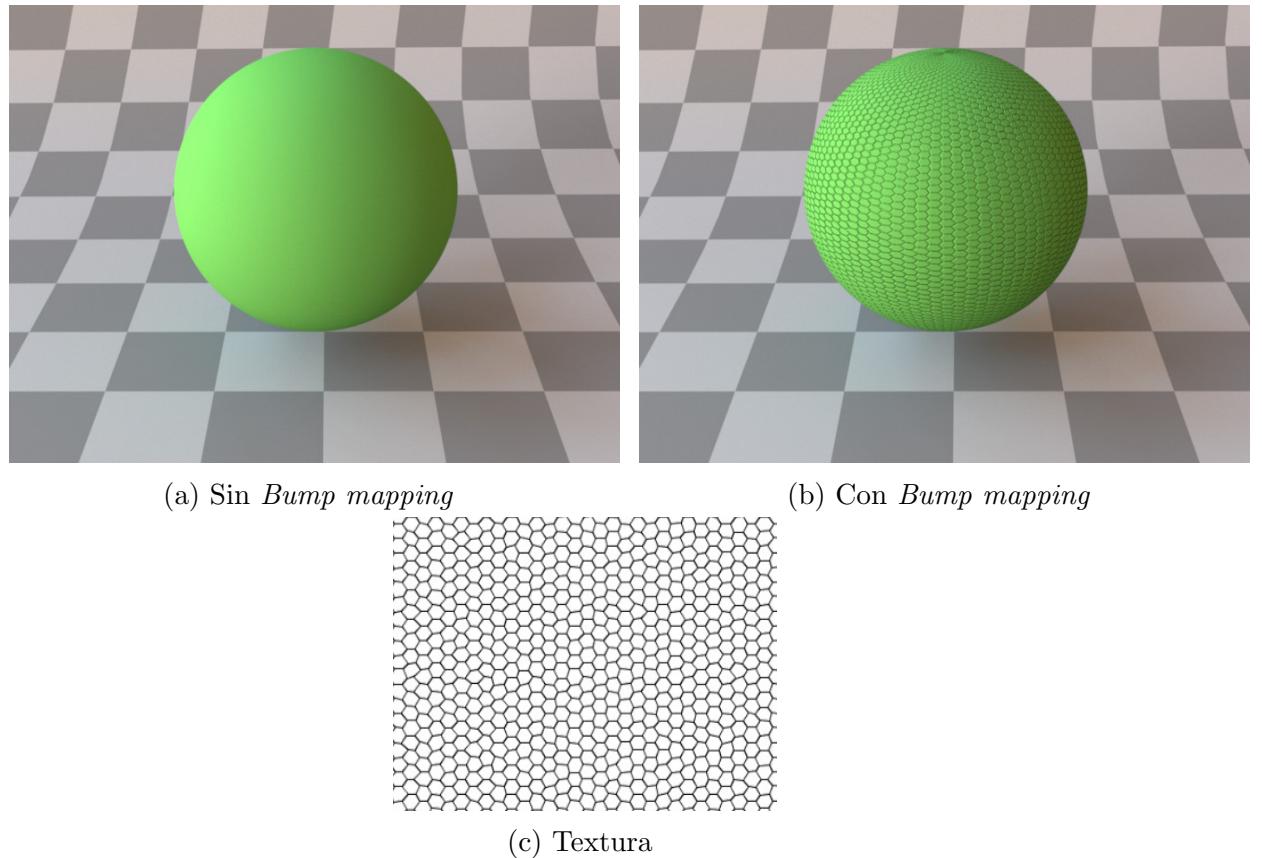


Figura 2.5: Ejemplo de aplicación de *Bump mapping* mediante textura

2.6. Algoritmos de Simulación

Para conseguir que todo lo anterior funcione, se ha de tener en cuenta los algoritmos de simulación que permiten obtener los colores correspondientes a cada pixel y por ende la imagen final. Existen multitud de tipos de simulación como el algoritmo *Path Tracing* usado en este trabajo.

2.6.1. Path Tracing

Path Tracing es una técnica basada en trazado de rayos anteriormente comentado en la Sección 2.3. Concretamente se basa en la resolución de la *Ecuación de Render* (2.1) de manera característica, pues en vez de integrar para toda la hemisferia, es decir, para todos los rayos posibles en tal situación, se centra en resolver solo una de esas posibilidades y por ende un solo camino posible del rayo (denominado *path*). En la Figura 2.6 se muestra gráficamente como *Path Tracing* es una técnica basada en trazado de rayos simplificado.

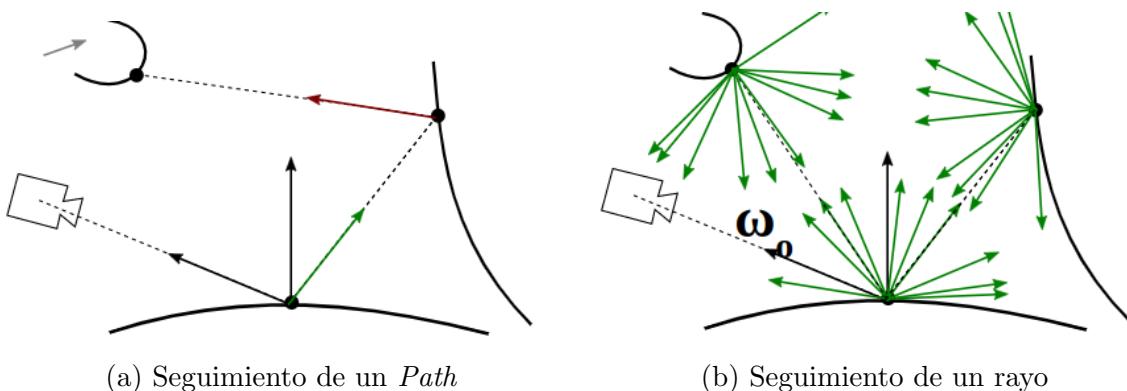


Figura 2.6: A la izquierda un ejemplo de un *Path*; se puede ver como este sigue un solo camino rebotando en los materiales sin dividirse en todos los posibles rayos hasta una última intersección. A la derecha ejemplo de seguimiento de trazado de rayos, donde se ve como se exploran todas las posibilidades.

Con esto, lo que se va a hacer es propagar multitud de *paths* con el fin de obtener el renderizado de la imagen completa pero con un costo mucho menor al que un trazado de rayos tendría debido al comentado seguimiento de un solo camino en vez de la multitud de posibilidades que este tiene. Se puede consultar de manera más detallada y concisa la explicación de este tipo de algoritmo de simulación en el Anexo D.

2.6.2. Position-Free Monte Carlo

En la Sección 2.4.1 se hace referencia a los materiales multicapa y a que existen diferentes formas de simular el transporte de la luz por ellos. *Position-Free Monte Carlo* [9] es un método de simulación del transporte de la luz en materiales multicapa, modelando desde las intersecciones con las capas hasta el camino que sigue la luz entre ellas.

Este algoritmo se basa en el uso de *Path Tracing* (2.6.1) para simular el transporte de luz entre capas con el fin de que quede reflejada la interacción de la luz con el medio que se encuentra entre las mismas.

Es caracterizado también por asumir que en el recorrido de un *path* por entre capas no hay desplazamiento en el eje horizontal, sino solo en el vertical. Esta asunción viene dada porque el desplazamiento en el eje horizontal es tan mínimo en una superficie infinita que ignorarlo es una aproximación razonable, por lo que lo interesante es simular el transporte en profundidad por el medio ya que es este el que afectará de manera más notable a la luz y además simplificará cuantiosamente su resolución.

Vease un ejemplo en la Figura 2.7.

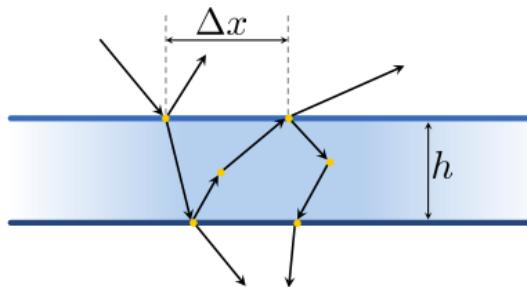


Figura 2.7: Imagen ejemplo de la asunción de mismo punto de salida y entrada pues Δx se asume 0; también muestra el camino simulado -*path*- seguido por un rayo, siendo los puntos amarillos los de cálculo del color y las flechas el camino que sigue [9]

Para ver este algoritmo descrito de manera más detallada y relacionado de manera directa con *Path Tracing*, se puede ver el Anexo D.

Capítulo 3

Desarrollo e Implementación

Durante este capítulo se va a hablar del trabajo realizado para la implementación del modelo de apariencia final. Se comenzará con la mención del entorno de programación usado para seguidamente hablar sobre el modelo creado. La descripción del modelo de apariencia que ha sido creado será llevada a cabo de manera estructurada, en un primer lugar se describirá tal modelo relacionandolo con todo lo visto en Fundamentos Teóricos (Capítulo 2) para seguidamente describir cada pieza del mismo por separado para que este se entienda correctamente.

3.1. Entorno de Programación

Para llevar a cabo una implementación del modelo de apariencia final de manera correcta es necesario tener un código base que una las piezas a implementar de manera sencilla, es por ello que es necesario el uso de un renderizador de imágenes previamente programado sobre el cual se acoplen los diferentes módulos implementados para su correcto funcionamiento. El renderizador elegido es el conocido como *Mitsuba* en su versión 0.6.0 [2], por lo que a continuación se explica sus principales características y el por qué de su elección.

3.1.1. Mitsuba

Mitsuba [2] es un *software* de renderizado de imágenes orientado a investigación. Está implementado en el lenguaje de programación C++ y otorga la posibilidad de realizar renderizados de manera tanto concurrente como distribuida para su optimización. Además su estructura es extremadamente modular lo que facilita de gran manera su entendimiento y uso así como la adición de nuevos módulos con fines de investigación o meramente de ocio.

Gracias a esto es que es perfecto para su uso en este trabajo, pues a parte de dar la facilidad de la adición de módulos como los que se van a comentar en secciones

posteriores, permite tener una base sólida sobre la que empezar a trabajar.

3.2. BSDF Final

Para el modelado de apariencia de la serpiente *Xenopeltis unicolor* hay que tener en cuenta, al igual que en el estudio de la misma en la Sección B.2, tanto su macro-estructura a nivel de patrón de escamas como su micro-estructura a nivel de coloración.

Con respecto a la macro-estructura es necesaria la creación del patrón de escamas para su posterior adición al modelo, para ello se va a modelar una textura con respecto al patrón expuesto en la Sección B.2, a continuación tal textura será aplicada al modelo usando la técnica *Bump Mapping* (2.5) para dar a la superficie la apariencia de escamas deseada. Se puede ver paso a paso la creación de la textura en la Sección 3.3 posterior a esta.

Seguidamente hay que darle al modelo de apariencia en cuestión la coloración que este ha de poseer, para ello hay que fijarse en la micro-estructura (ver Sección B.2) dividiéndose esta en dos partes (cuya explicación se puede ver así mismo de manera gráfica en la Figura 3.1):

1. La primera consta de la capa mencionada de iridóforos la cual otorga iridiscencia a la serpiente. Para la simulación de la misma es que se ha implementado el fenómeno de *Interferencia de Capa Fina* (C) sobre un material dieléctrico que simula los iridóforos (debido a que son láminas de cristales y estos a su vez materiales dieléctricos) cuya implementación se explica en la Sección 3.4. Concretamente los parámetros usados para la instanciación de la *Interferencia de Capa Fina* son:

- Índice de refracción exterior: 1.0 al ser aire.
- Índice de refracción intermedio: 1.56 pues los iridóforos poseen tal índice que es el de la queratina.
- Índice de refracción interior: 1.0 pues entre la capa interior y la de absorción se encuentra un espacio de muy pequeño tamaño relleno por aire (es posible que en ocasiones este espacio no exista por lo que pueda variar este índice de refracción).
- Anchura de la capa: suele variar entre los 800 y 1000 nanómetros, por lo que se establecerá la media de 900nm.

2. La segunda consta de la capa de pigmentos oscuros, que tal y como se ha expuesto en la Sección B.2 está formada por una capa superficial de absorción con un color

difuso oscuro como capa interior. Su implementación se puede ver en la Sección 3.5; además, tal y como se expone en tal Sección hacen falta dos parámetros obtenidos de manera empírica:

- Coeficiente de absorción: 0.7.
- Anchura: 2.

Al haberse hablado de capas, la simulación del transporte de la luz por ellas va a ser llevada a cabo por el método de *Position-free Monte Carlo* (ver Anexo D.2).

Aplicando todo esto se tiene un material basado en *Position-free Monte Carlo* con diferentes capas añadidas como ilustra la Figura 3.1. Esto quiere decir que el factor f_r de la *Ecuación de Render* 2.1 queda resuelto de manera que simula la apariencia de la serpiente *Xenopeltis unicolor* y por tanto el objetivo del trabajo se ha conseguido.

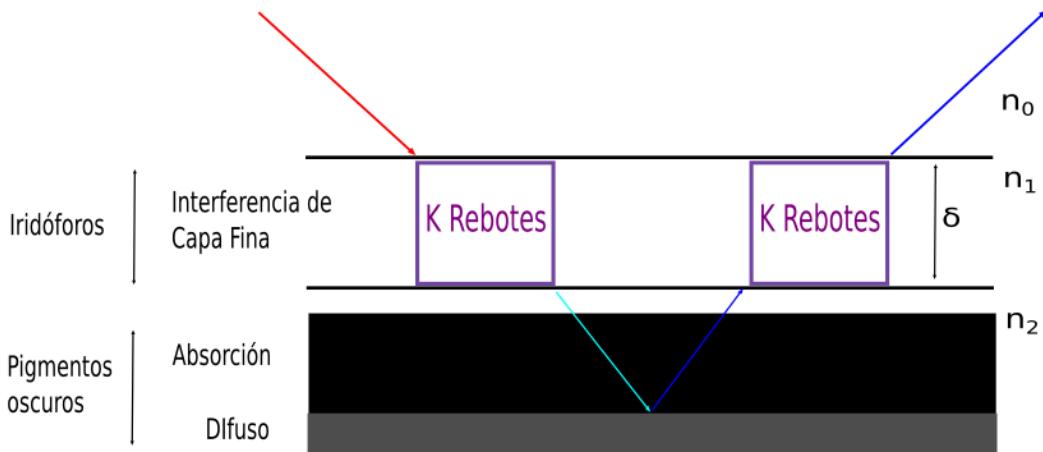


Figura 3.1: Modelo de apariencia multicapa final desarrollado

Con todo esto el resultado de los módulos implementados va a quedar tal y como se ve en el diagrama de clases 3.2, siendo las clases marcadas en verde las implementadas desde cero, las marcadas en rojo las que han tenido que ser debidamente acopladas a *Mitsuba* (3.1) y el resto las pertenecientes a *Mitsuba* y que hacen uso de lo implementado para su correcto funcionamiento. Cada una de ellas tiene un fin:

- *TexturaEscamas*: representa al objeto creado como textura, se puede ver más detalladamente en la Sección 3.3.
- *InterferenciaCapaFina*: se encarga de implementar los cálculos necesarios para la simulación del fenómeno de *interferencia de capa fina* (C). Se puede ver más en

detalle en la descripción de la primera fase de implementación de *interferencia de capa fina* en la Sección 3.4.

- *DieléctricoConInterferenciaCapaFina*: se encarga de la creación de un material dieléctrico al que se le ha añadido *interferencia de capa fina* para simular la capa de iridóforos. Representa una *BSDF* por lo que se encarga de la implementación de su evaluación y muestreo así como de su construcción, destrucción y evaluación de la *pdf*. Se puede ver más en detalle en la descripción de la segunda fase de implementación de *interferencia de capa fina* en la Sección 3.4.
- *Absorción*: se encarga de modelar la capa de absorción que simula la capa de pigmentos oscuros. Representa una *BSDF* por lo que se encarga de la implementación de su evaluación y muestreo así como de su construcción, destrucción y evaluación de la *pdf*. Se puede ver más detalladamente en la Sección 3.5.
- *Position-Free Monte Carlo*: se encarga de implementar el transporte de la luz por modelos multicapa usando *Position-Free Monte Carlo* D.2. Representa una *BSDF* por lo que se encarga de la implementación de su evaluación y muestreo así como de su construcción, destrucción y evaluación de la *pdf*. Se puede ver más en detalle en la Sección 3.6.

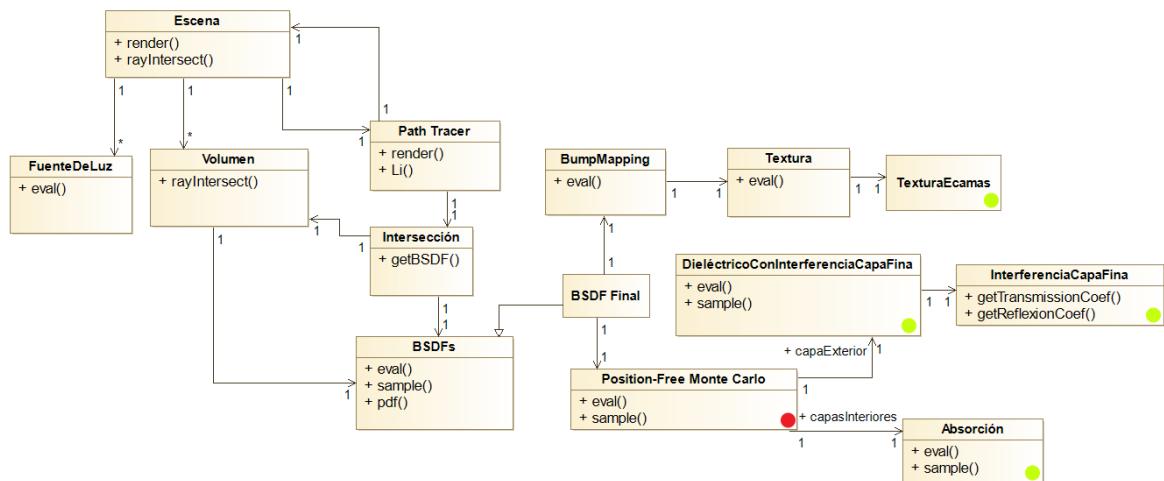


Figura 3.2: Diagrama de clases del modelo de apariencia implementado y su acoplamiento a *Mitsuba*

Cabe destacar que la clase *Escena* no ha sido implementada como tal, sin embargo sí que es necesaria su instanciación mediante la escritura de ficheros XML en los que se ha de modelar la escena tal y como se puede ver en la documentación de *Mitsuba* [2].

3.3. Modelado de la Textura

Para la recreación del patrón de escamas y por tanto la modelación de apariencia a macro-escala, es necesario modelar una textura que lo simule. Como se comentó en la sección referente a *Bump Mapping* (2.5), es posible aplicar una textura a un modelo geométrico de manera que se simulan desplazamientos sobre la superficie del mismo. Es por ello que esta técnica es esencial para la simulación de escamas.

Lo primero a tener en cuenta en la modelación de la textura es el resultado que se quiere obtener, que tal y como se ha mencionado en el apartado B.2 la textura habría de quedar como filas de hexágonos (con pequeñas irregularidades) perfectamente encajadas entre ellas. Para crear dicha textura se ha usado el editor de imágenes libre y de código abierto conocido como Gimp2.10 [10] cuyas herramientas dan la posibilidad de generar la textura definida previamente.

Los pasos a seguir para la modelación de la textura se basan en la aplicación sobre un lienzo en blanco de un filtro de **Mosaico** y su posterior ajuste de la regularidad de los hexágonos que lo forman mediante los parámetros de regularidad que la herramienta otorga.

Esta es la textura conseguida mediante los pasos especificados:

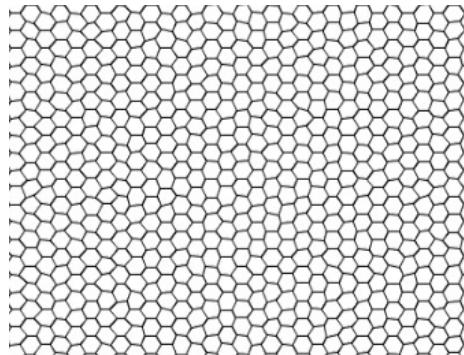


Figura 3.3: Textura final

Una vez se tiene esta textura, ya es posible añadirla al modelo de apariencia para que gracias a *Bump Mapping* simule la apariencia de las escamas, de manera que se añade al modelo final tal y como se ve en el diagrama 3.2.

3.4. Interferencia de Capa Fina

Tal y como se ha mencionado en la sección B.2, la serpiente *Xenopeltis unicolor* posee sobre sus escamas los denominados iridóforos, los cuales generan iridiscencia debido a *Interferencia de Capa Fina*. Por lo tanto, es esencial la implementación de tal fenómeno con el fin de replicar el comportamiento de las escamas de este reptil.

Dados los fundamentos teóricos de *Interferencia de Capa Fina* descritos en la sección C, es posible llevar a cabo un cálculo numérico para la obtención de este fenómeno, basándose este en el cálculo de la ecuación C.5 y por conservación de energía también de la ecuación C.6, para conseguir los coeficientes de reflexión y refracción -transmisión-.

La idea de la implementación se basa en que una vez se obtienen tales coeficientes, se han de sustituir por el resultado de la ecuación de *Fresnel* (A.5) del material deseado, con el fin de que la cantidad de energía no venga dada por Fresnel sino por el cálculo de los coeficientes teniendo en cuenta *Interferencia de Capa Fina*. Esto hará que las funciones que lo usan que son *eval* y *sample* en vez de usar el coeficiente dado por *Fresnel* usen el dado por *Interferencia de Capa Fina*, lo que ocasionará que la amplitud y fase de onda de cada color cambien de tal manera que se obtenga iridiscencia. La implementación se divide en 2 fases:

1. La primera consta de la implementación de las funciones que resuelvan la ecuación final expuesta con anterioridad (C.5) para la obtención de los coeficientes, estas funciones son las vistas en el diagrama 3.2 como las funciones *getTransmissionCoef* y *getReflexionCoef* pertenecientes a la clase *InterferenciaDeCapaFina*. Como se puede ver dicha ecuación se divide en dos factores, el primero de ellos formado por los índices de refracción y los cosenos del rayo incidente con la normal en el medio exterior e interior. Los índices de refracción así como el coseno del rayo incidente con la normal del medio exterior vienen dados por lo que se habrá de calcular únicamente el coseno del rayo incidente con la normal de la capa interior; para ello se usará la *ley de Snell* (A.4) con la que se calculará el seno y con las razones trigonométricas una vez se tiene el seno se puede calcular el coseno. Vease la ecuación trigonométrica:

$$\sin^2 \theta + \cos^2 \theta = 1 \quad (3.1)$$

Con el primer factor calculado, solo falta el cálculo del segundo. Este se va a calcular dos veces, una con las ecuaciones de Fresnel con luz s-polarizada y otra con luz p-polarizada, pues se asume que se obtiene la misma de ambos tipos por lo que se hace una media de los resultados eliminando así la polarización. Para el cálculo de este factor bastará con calcular de manera estricta las ecuaciones que se han planteado en el Anexo C (ver ecuación C.5) y ya se habrá obtenido el total de energía refractada por la longitud de onda dada; además, por el principio de conservación de energía (ver C.6) también se tiene la cantidad de energía reflejada.

2. La segunda parte consta de integrar dicha implementación del cálculo de coeficientes de transmisión y reflexión en la *BSDF* a la que se le quiera aplicar

este fenómeno. Para ello hay que modificar tanto su evaluación como su muestreo (vistos en D.1) pues el cálculo de radiancia se va a ver afectado por los coeficientes tanto de refracción como de reflexión obtenidos en el apartado anterior y el muestreo del rayo de salida o bien se verá perfectamente reflejado o bien refractará pero teniendo en cuenta los índices de refracción que la *Capa Fina* posea. Para su integración tanto para el muestreo como para la evaluación (cuyo pseudo-código de implementación genérico se puede ver en el Algoritmo 1) hay que tener en cuenta diferentes aspectos:

- En el paso anterior se ha calculado el coeficiente para una longitud de onda dada, por tanto debido a que se usa un espectro RGB habrá que calcularlo para cada color con su longitud de onda en correspondiente (rojo: 650nm; verde: 510nm; azul: 475nm). Dicho vector se multiplicará al de radiancia resultante con el fin de quedarse con la cantidad de energía requerida por tal coeficiente. Antes de la multiplicación el vector habrá de ser normalizado.
- Si el material contiene reflexión total el cambio con *Fresnel* es trivial, se ha únicamente de sustituir y multiplicar el espectro resultante por el obtenido.
- Si el material contiene refracción se ha de tener en cuenta el lado de incidencia del rayo, es decir, si se incide por el interior del material o por el exterior. Cuando el rayo incida por el exterior del material no habrá que hacer nada simplemente invocar a la función. Sin embargo cuando este haya refractado e incida por el interior se tendrá que poner signo positivo al coseno de incidencia (pues en la implementación de las funciones *getTransmissionCoef* y *getReflexionCoef* se asume como tal) e invertir los índices de refracción al invocar al algoritmo los algoritmos *getTransmissionCoef* y *getReflexionCoef* debido a que ahora el rayo viaja del medio interno al externo, es decir, a la inversa.

Con esto ya se tiene implementada *Interferencia de Capa Fina* y se puede añadir a cualquier material que use *Fresnel*. Por tanto se tiene que con respecto al diagrama 3.2 se tienen implementadas las clases *InterferenciaDeCapaFina* y *DieléctricoConInterferenciaCapaFina* con sus respectivas funciones que permiten el modelado de la capa de iridóforos que aporta iridiscencia al modelo de apariencia.

3.5. Capa de Absorción

Como se comenta en la Sección 2.1 debajo de la capa de iridóforos la serpiente *Xenopeltis unicolor* posee una capa de pigmentos oscura que la dota de ese color oscuro

Algorithm 1 Integración del Algoritmo de Interferencia de Capa Fina a un Material

Require: $radiancia, \cos\theta, anchura, n_0, n_1, n_2$

Ensure: $radiancia$

```
1:  $\{R, G, B\} \leftarrow 0$ 
2: if interseccion.esExterior() then
3:    $R \leftarrow getXCoeff(\cos\theta, 650, anchura, n_0, n_1, n_2)$ 
4:    $G \leftarrow getXCoeff(\cos\theta, 510, anchura, n_0, n_1, n_2)$ 
5:    $B \leftarrow getXCoeff(\cos\theta, 475, anchura, n_0, n_1, n_2)$ 
6: else
7:    $R \leftarrow getXCoeff(|\cos\theta|, 650, anchura, n_2, n_1, n_0)$ 
8:    $G \leftarrow getXCoeff(|\cos\theta|, 510, anchura, n_2, n_1, n_0)$ 
9:    $B \leftarrow getXCoeff(|\cos\theta|, 475, anchura, n_2, n_1, n_0)$ 
10: end if
11:  $radiancia \leftarrow radiancia * normalizar(\{R, G, B\})$ 
12: return :  $radiancia$ 
```

característico. Modelar esta capa se puede hacer de manera sencilla, es decir, modelarla simplemente con un color negro que la simule. Sin embargo existe una manera más precisa y usada en renderizados de pieles humanas basada en el uso de una capa de absorción.

La finalidad de la capa de absorción es simular el recorrido de la luz por la capa de pigmentos oscura hasta su incidencia con el interior de la misma formado por un sustrato difuso, esto es, contra más perpendicular a la superficie incida el rayo, menos recorrido va a tener este por dicha capa y por tanto menor absorción de color se va producir, mientras que a menor perpendicularidad se tiene que el recorrido por la capa es mayor por lo tanto se tendrá una mayor absorción. Gráficamente se puede ver un ejemplo del comportamiento de este fenómeno en la Figura 3.4, en la cual el rayo viaja por la capa de absorción hasta incidir con el sustrato difuso interior e iniciar su viaje de salida; se puede ver como el rayo de salida se encuentra ennegrecido, esto es debido a que se ha absorbido el color durante el paso por la capa.

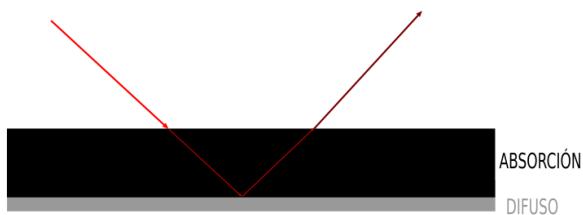


Figura 3.4: Representación gráfica de un ejemplo de absorción

Es necesario dividir la implementación de esta capa en dos fases, una que implemente el sustrato difuso interno y otra la absorción propiamente dicha.

En primer lugar se encuentra la capa interior difusa. Esta actúa tal y como se

describe este material en la Sección 2.4 habiéndosele aplicado un color grisaceo como base.

En segundo lugar se encuentra la capa de absorción que consta de dos parámetros esenciales para su implementación:

- *Coeficiente de absorción*: este coeficiente establece la energía absorbida tras el paso por la capa.
- *Anchura*: este parámetro define la anchura de la capa de absorción, a mayor anchura mayor es el recorrido y por tanto habrá más absorción.

Con ambas capas establecidas su acoplamiento es llevado a cabo con la implementación tanto de la evaluación de la radiancia como del muestreo. Para la evaluación es necesario tener en cuenta la absorción obtenida por los coeficientes y el color de la capa difusa pues el resultado de la radiancia va a radicar de la solución de aplicar la absorción sobre el color difuso tal y como se ve en el algoritmo 2. En este algoritmo la absorción queda calculada primero multiplicando el coeficiente de absorción por la anchura para tener esta en cuenta y seguidamente elevando e a un número negativo (para obtener como absorción un dígito decimal y que por tanto disminuya la radiancia) calculado como el resultado de multiplicar el coeficiente obtenido con anterioridad con la distancia recorrida por la luz a través de la capa de absorción (con el fin de que a mayor recorrido, mayor sea el número negativo y por ende menor el resultado del cálculo de absorción por lo que su multiplicación disminuirá en mayor medida la radiancia). Con respecto al muestreo se ha de tener únicamente en cuenta la capa interior difusa pues la de absorción no supone un desvío en la dirección del rayo de entrada; es por ello que se ha de muestrear con respecto al comportamiento de un material difuso de manera aleatoria por la hemiesfera tal y como se ve en la Sección 2.4.

Algorithm 2 Algoritmo de Absorción sobre Difuso

Require: *interseccion, coeficiente, anchura, radiancia*

Ensure: *absolucion*

- 1: $coeficiente \leftarrow coeficiente * anchura$
 - 2: $absolucion \leftarrow e^{-coeficiente*interseccion.recorrido}$
 - 3: **return** : $absolucion * radiancia$
-

Al ser un tipo de material, en la *Ecuación de Render* 2.1 resuelve el factor f_r y lo hace de la manera que se ve en la ecuación 3.2 donde el resultado viene dado por la multiplicación de la f_r del material difuso interno por la absorción como ilustra el algoritmo 2.

$$f_r = \frac{k_d}{\pi} * absolucion \quad (3.2)$$

Esta capa encaja con un tipo de material *BSDF*, por lo que con respecto al diagrama de clases de la imagen 3.2 implementa la clase denominada *Absorción* modelando su evaluación y muestreo.

3.6. Position-free Monte Carlo

En la sección D.2 se ve la importancia que cobra *Position-free Monte Carlo* en las simulaciones de materiales multicapa y por consiguiente en el modelo de apariencia que se desea simular. Es por ello que ha sido necesaria la acoplación del código de este tipo de simulación en *Mitsuba* gracias a la existencia del mismo [9].

La finalidad de *Position-free Monte Carlo* es crear un tipo de material multicapa cuyas capas puedan ser arbitrariamente establecidas con los tipos de materiales que se deseen. Esto implica que se pueden establecer las capas que se quieran de los materiales que se quieran y simular el transporte de la luz entre capas con la técnica *Path Tracing*. Con respecto al diagrama 3.2 hace referencia a la clase *Position-Free Monte Carlo* teniendo por ende las funciones correspondientes. Lo peculiar en la implementación de este tipo de material es la existencia de la función *generatePath*, función llamada desde las establecidas en el diagrama 3.2 *eval* y *sample*, que se encarga de llevar a cabo *Path Tracing* entre las capas.

Esta función presenta dos posibilidades que se siguen en bucle:

- La primera posibilidad se encuentra en que se intersecte con una capa, en ese momento se identifica si esta puede refractar o no. De no hacerlo el rayo se verá reflejado siguiendo las propiedades del material de la propia capa intersectada, habiendo aquí otras dos posibilidades; la primera es que el rayo se encuentre en su camino de entrada y al reflejar sobre el material indicado comience su camino de salida (o lo termine si refleja en la capa externa), en este caso todos los materiales más internos se verán anulados pues la luz no podrá llegar hasta ellos; la segunda es que el rayo se encuentre en su camino de salida y por lo tanto el reflejo implique que este vuelva a dirigir su camino hacia los materiales internos. De estar la posibilidad de refractar y que esta sea la elegida por probabilidad se refractará siguiendo las propiedades del material de dicha capa, además, también se tienen dos posibilidades; la primera es que el rayo se encuentre en su camino de entrada y por tanto al refractar pase al espacio entre capas contiguo más interno; la segunda es que el rayo se encuentre en su camino de salida y por tanto al refractar este continue tal camino de manera que si se encontraba intersectando con la capa exterior este terminaría, de no ser así se pasaría al espacio entre capas contiguo más externo. En todas las intersecciones se ha de propagar la radiancia

que se haya muestreado al intersectar con el material de cada capa en cuestión.

- La segunda es que el camino seguido sea entre capas y por tanto se haya intersectado con un punto arbitrario del medio entre capas. De haber sucedido esto se calculará una nueva dirección y se propagará un determinado término de absorción al haber intersectado con tal medio.

En el algoritmo 3 se muestra en pseudocódigo la implementación de la función *generatePath*.

Algorithm 3 Algoritmo representativo de la función *generatePath*

Require: w_i

Ensure: $radiancia, w_o$

```

1: salida  $\leftarrow$  falso
2: while not salida do
3:   muestrearPath()
4:   if intersecta.capa() then
5:     radiancia  $\leftarrow$  radiancia * capa.evalucion
6:     if intersecta.capaExterior() then
7:       if intersecta.capaExteriorPorMedioExterior() and esReflexion() then
8:         w_o  $\leftarrow$  capa.muestrearReflexion
9:         salida  $\leftarrow$  cierto
10:        break
11:      end if
12:      if intersecta.capaExteriorPorMedioInterior() and esRefraccion() then
13:        w_o  $\leftarrow$  capa.muestrearRefraccion
14:        salida  $\leftarrow$  cierto
15:        break
16:      end if
17:    end if
18:    w_o  $\leftarrow$  capa.muestrear
19:    salida  $\leftarrow$  falso
20:   else if not intersecta.capa() then
21:     radiancia  $\leftarrow$  radiancia * medio.absorcion
22:     salida  $\leftarrow$  falso
23:   end if
24: end while
25: return : radiancia, w_o

```

Para el acoplamiento de este material a mitsuba es necesario su adición al código con su respectiva compilación. Con respecto al uso basta con entender la función *generatePath* y sus limitaciones para añadir los materiales de exterior a interior y que el transporte de luz pueda ser llevado a cabo.

Capítulo 4

Resultados y Validación

A lo largo de este capítulo se muestran los resultados de la implementación llevada a cabo vista a paso; esto es, se verá un proceso de creación del modelo desde la visión de los componentes por separado hasta la unión progresiva de los mismos hasta la obtención del resultado final. Se van a ver diferentes modelos geométricos, siendo estos una esfera, un toroide y una serpiente [11].

4.1. Resultados

A continuación se presentan los resultados obtenidos durante la realización de este trabajo tanto de manera descompuesta en piezas como estas unidas entre ellas para formar el modelo de apariencia final.

En primer lugar se van a presentar los modelos geométricos usados, que tal y como se han comentado son una esfera, un toroide y una serpiente a los que se le aplicará un color difuso grisaceo; estos se pueden ver en la imagen 4.1.

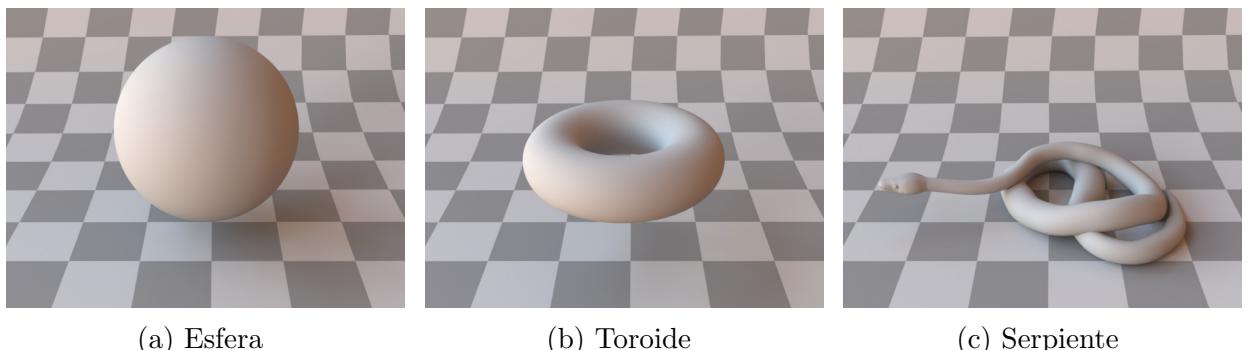


Figura 4.1: Modelos geométricos usados para el modelado de apariencia, con un material difuso gris

Para continuar con la creación del modelo de apariencia hay que centrarse en primer lugar en la macro-estructura, es decir, en el patrón de escamas. Para ello hay que aplicar a los modelos geométricos de la imagen 4.1 la textura desarrollada en la sección 3.3 empleando *Bump Mapping* 2.5 para dotar de relieve a las escamas y por ende un mayor realismo a las imágenes. El resultado se puede apreciar en la imagen 4.2.

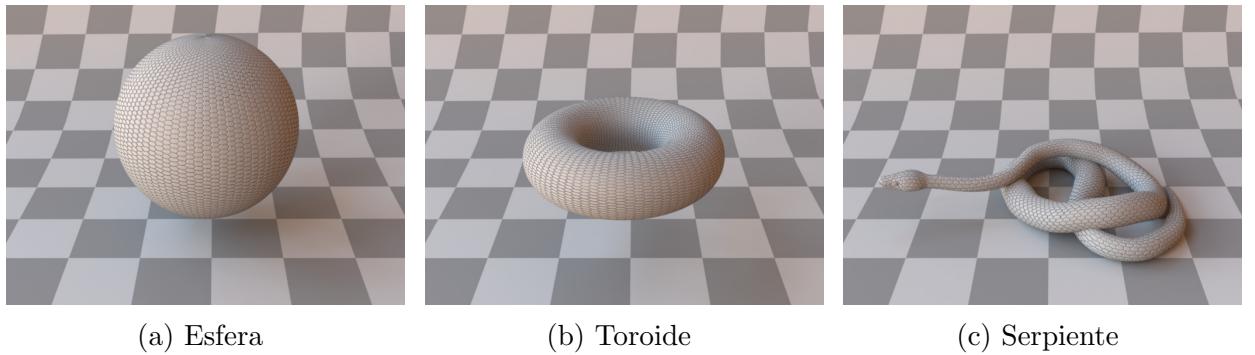


Figura 4.2: Textura aplicada a los modelos geométricos con *Bump Mapping*

Al igual que en anteriores apartados, una vez se tiene la macro-estructura hay que centrarse en la coloración marcada por la micro-estructura de la serpiente, por lo que en primer lugar se va a mostrar la simulación de la capa de iridóforos por *Interferencia de Capa Fina* que se puede ver en la imagen 4.3. Vease que lo simulado en esta imagen 4.3 es meramente *Interferencia de Capa Fina* habiendo sido los parámetros ajustados a los establecidos para el modelo de apariencia final en la Sección 3.2.

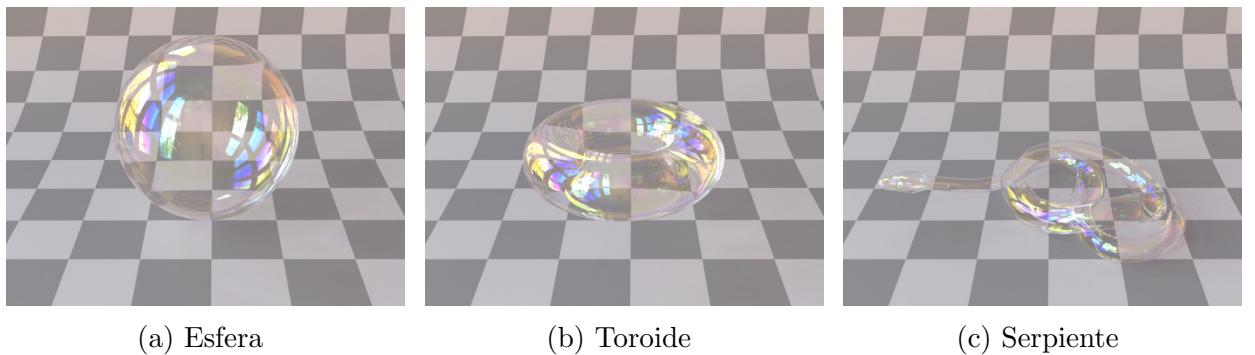


Figura 4.3: *Interferencia de Capa Fina* aplicada a los modelos 3D simulando una burbuja. Parámetros: $n_0 = 1,0$; $n_1 = 1,56$; $n_2 = 1,0$; $\delta = 900$

A continuación se va a mostrar en la imagen 4.4 la adición de los resultados de aplicar la textura con *Bump Mapping* vistos en la imagen 4.2 al material con *Interferencia de Capa Fina* visto en la imagen 4.3.

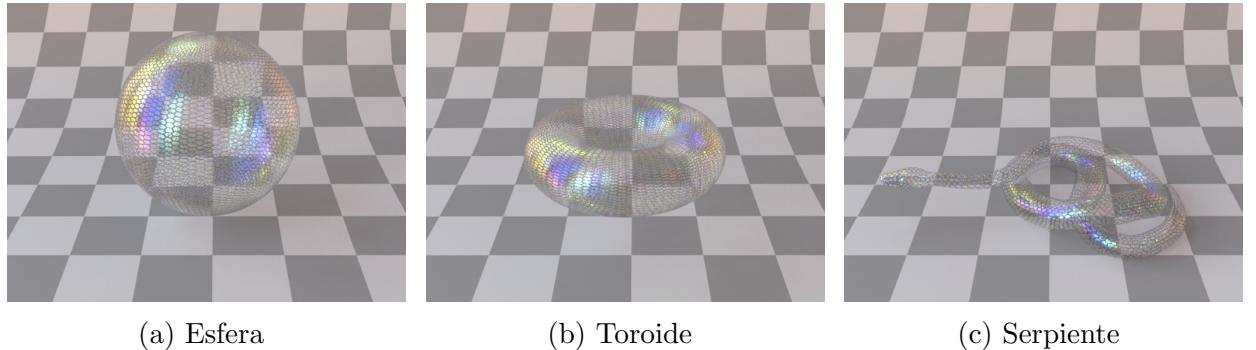


Figura 4.4: *Bump Mapping* aplicado sobre material con *Interferencia de Capa Fina*

Para finalizar el modelo queda modelar la apariencia de la capa de pigmentos oscura y unirla a los resultados de la imágen 4.4. Es para tal fin que se ha modelado la capa de absorción descrita en la Sección 3.5 con los parámetros comentados en la descripción del modelo de apariencia final en la Sección 3.2, cuyos resultados sobre los modelos geométricos en cuestión se ven en la imágen 4.5.

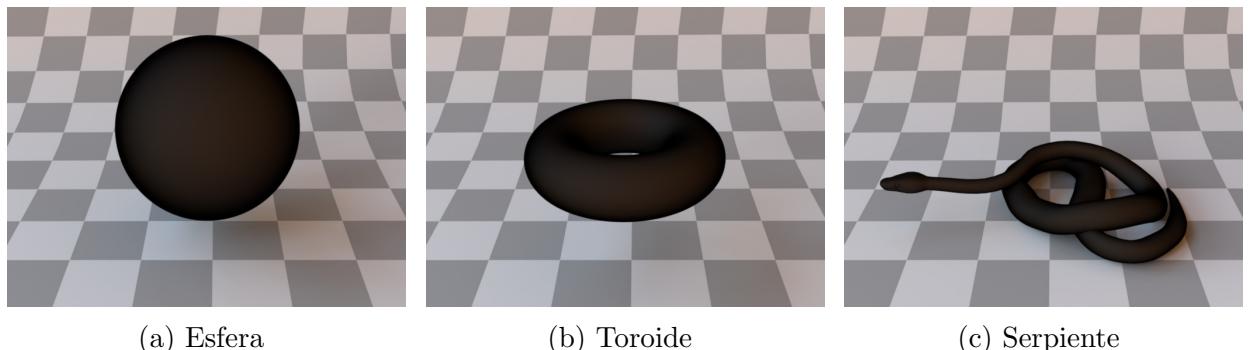


Figura 4.5: Material de absorción. Parámetros: coef-absorción = 0.7; anchura = 2

Por tanto si se juntan todos los materiales obtenidos anteriormente con esta capa de pigmentos interna usando *Position-Free Monte Carlo* para el modelado entre capas el resultado final de este modelo de apariencia de 3 capas se ve en las imágenes 4.6a para una esfera, 4.6b para un toroide y 4.7 para una serpiente.

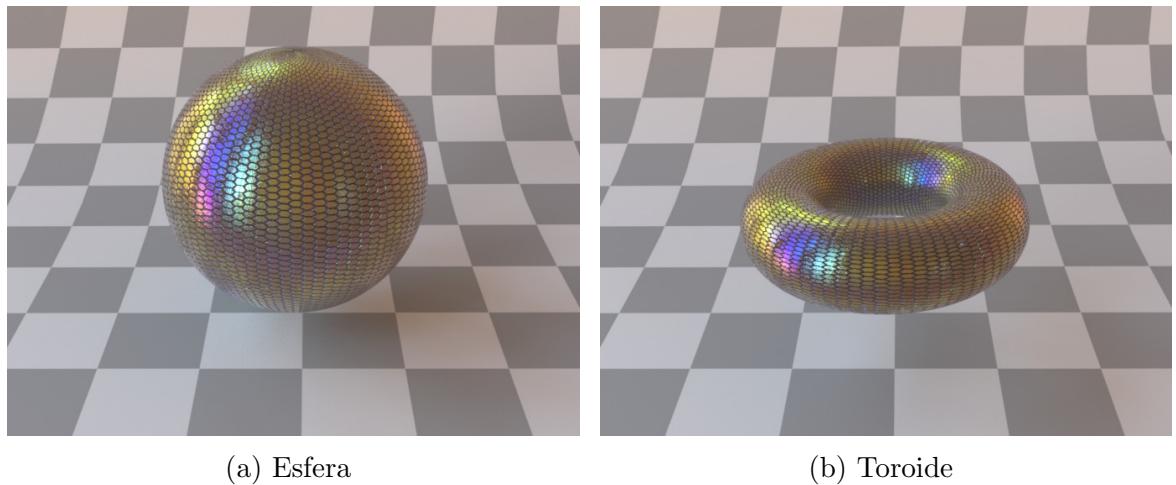


Figura 4.6: Modelo de apariencia final sobre una esfera y un toroide

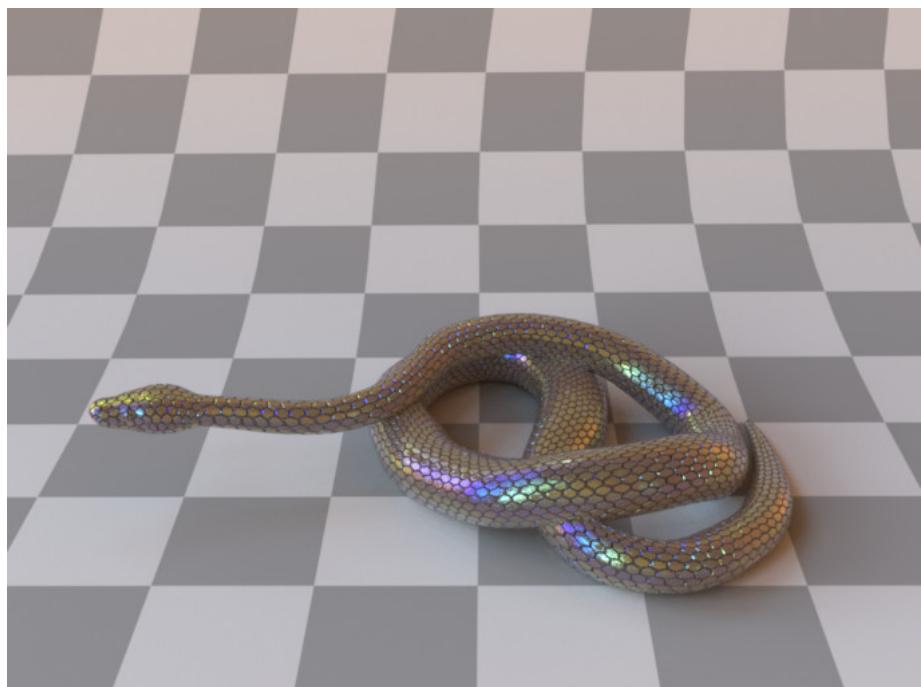


Figura 4.7: Modelo de apariencia final sobre una serpiente

Para concluir con los resultados se van a mostrar en la Figura 4.8 diversos renders con parámetros diferentes a los establecidos en el modelo final con respecto a *interferencia de capa fina* exceptuando el referente al índice de refracción externo que se mantiene como aire y el intermedio que se mantiene como la queratina que forma los iridóforos.

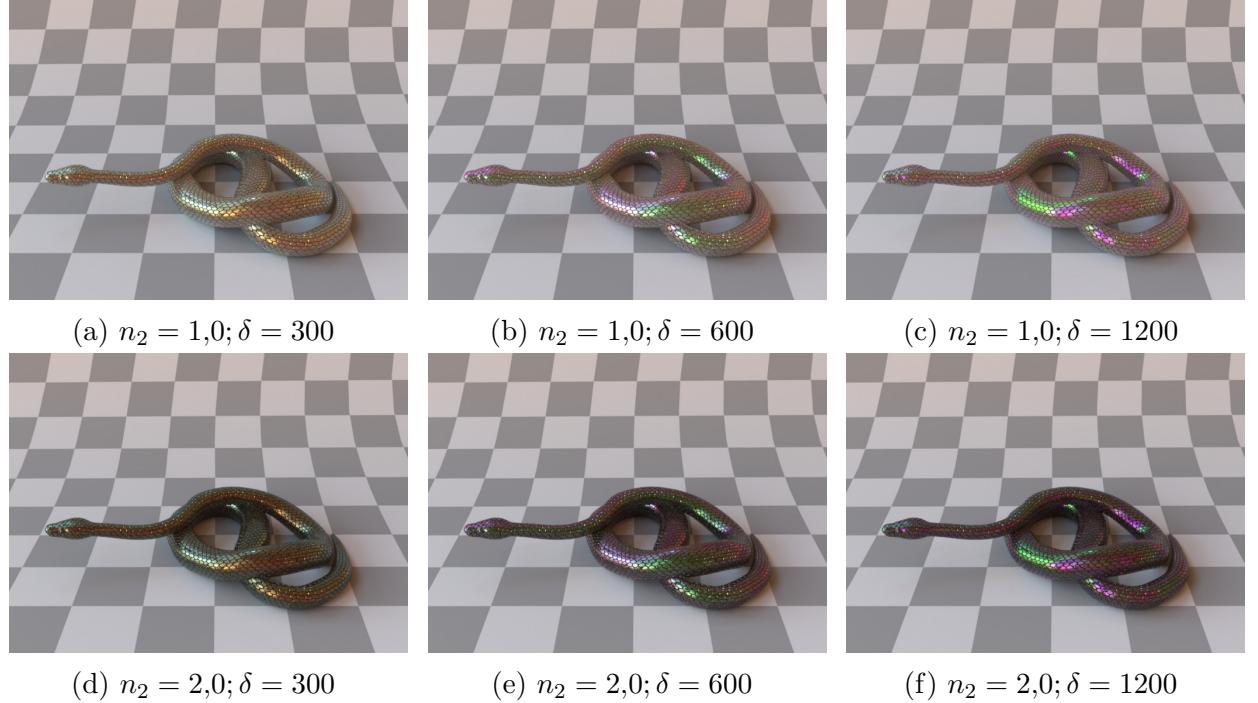


Figura 4.8: Figuras con diferentes parámetros a los del modelo final. En la primera fila se puede apreciar como cambia la apariencia cuando se incrementa la profundidad de la capa; mientras que en la segunda fila se cambia el índice de refracción de la capa intermedia

4.2. Validación

La validación de manera visual consta de la comparación del modelo de apariencia final obtenido con imágenes reales de la serpiente *Xenopeltis unicolor* y su objetivo es verificar si el modelo de apariencia realmente se asemeja a la serpiente a simular. Se han realizado diversas comparaciones con imágenes reales diferentes con el fin de conseguir la iluminación más parecida a la de la simulación.

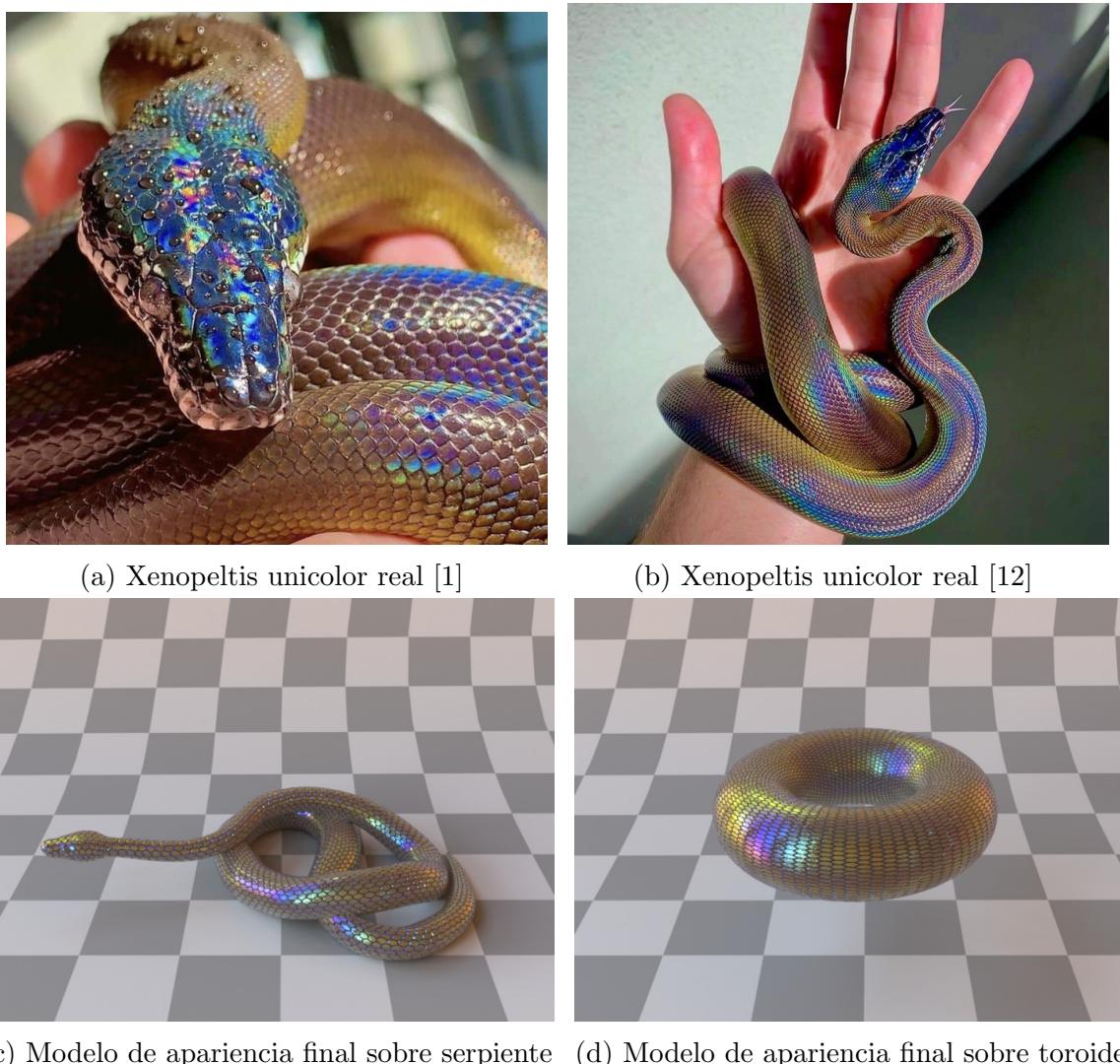


Figura 4.9: Comparación de una imagen real de la serpiente *Xenopeltis unicolor* con el modelo de apariencia final

Se puede ver que los resultados del modelo de apariencia consiguen simular de manera correcta una visión genérica de la serpiente. Con respecto a la coloración, predomina el color dorado característico seguido de partes con colores azulados, azul verdosos y morados. Con respecto al patrón de escamas también se consigue simular de manera precisa, pues se puede ver sobre todo en la Figura 4.9a como se sigue ese patrón

de escamas hexagonal comentado en numerosas ocasiones en anteriores secciones.

Es verdad que existen diferencias en la coloración entre la imagen real y los *renders*, esto es debido a que la iluminación es imposible de igualar en el proceso de renderizado y en imagen real, además de que los parámetros usados en el modelo final y en la realidad pueden variar debido a que para mayor simplicidad al renderizar se han asumido parámetros constantes mientras que en la realidad estos aunque de manera ínfima, varian.

Como conclusión final a la validación, se tiene que el modelo es satisfactorio, pues consigue modelar lo propuesto, tanto el patrón de escamas como la coloración general de la serpiente *Xenopeltis unicolor*.

Capítulo 5

Conclusiones y Trabajo Futuro

5.1. Conclusiones del Trabajo

En este trabajo se han realizado diferentes tareas para la obtención del modelado apariencia de la serpiente *Xenopeltis unicolor*, especie cuya apariencia se quería replicar. Para ello se han seguido las pautas denotadas en la sección 1.3 con el fin de cumplir todos y cada uno de los objetivos marcados en la sección 1.2.

En primer lugar se ha llevado a cabo un estudio de la coloración estructural en la naturaleza gracias a numerosos documentos. Esto ha permitido establecer unas bases sobre la coloración que han sido necesarias para apartados posteriores.

En segundo lugar se ha llevado a cabo un estudio de la estructura general de la familia de las serpientes para a continuación centrarse en lo que a la especie *Xenopeltis unicolor* se refiere, estableciendo así la manera de realizar un modelado de apariencia que se asemejase lo máximo posible a dicha estructura.

Una vez la estructura de la serpiente cuya apariencia se iba a modelar se ha conocido se han llevado a cabo las implementaciones necesarias para una simulación de apariencia lo más realista y físicamente correcta posible. Con las implementaciones establecidas, se han acoplado estas al renderizador *Mitsuba* permitiendo así obtener imágenes de lo implementado.

Al tener la posibilidad de crear imágenes con el modelo de apariencia de la serpiente objeto de estudio, se ha procedido a realizar tal labor sobre diferentes modelos geométricos como son una esfera, un toroide y una serpiente.

Finalizando el trabajo realizado, se ha procedido a validar las imágenes obtenidas de manera visual gracias a la comparación de *renders* con imágenes reales de la serpiente *Xenopeltis unicolor*.

5.2. Trabajo Futuro

Con respecto al trabajo futuro, el mayor punto a explorar se encuentra en el *Renderizado Espectral*, este renderizado a diferencia del que se ha usado durante este proyecto (sistema RGB con los espectros de longitud de onda rojo, verde y azul) utiliza la mayor cantidad de longitudes de onda posibles con el fin de asemejarse a la realidad en mayor medida. Esto daría como resultado una iridiscencia con colores ligeramente diferentes a los que se han visto durante el transcurso de esta memoria y con una mayor precisión física [13].

Como otro punto a explotar se encuentra una variación espacial de anchuras de capa tanto a nivel de *Interferencia de capa fina* como de *Position-Free Monte Carlo* lo cual acercaría al modelo un paso más hacia el realismo físico. Sin embargo para que esta implementación tenga sentido se ha de estudiar más a fondo las propiedades de la serpiente a replicar y los cambios nanométricos de las anchuras de sus capas, así como los patrones (de haberlos) de dichos cambios. Este estudio complicaría de gran manera la resolución de este trabajo dada la gran labor de investigación que llevaría detrás.

Como último punto a añadir, se podrían renderizar imágenes con diferentes parámetros a los vistos en la sección referente a resultados 4, obteniendo patrones de colores muy diversos pero probablemente menos realistas que los ya obtenidos.

5.3. Conclusiones Personales

Este proyecto me ha servido como una forma de demostrar tanto a nivel de profesorado como a nivel individual los conocimientos adquiridos en la carrera y la capacidad de afrontar diferentes problemas informáticos pertenezcan a una rama u otra. Además me ha adentrado en el mundo de la investigación, pues la información para realizar este trabajo no venía ya dada sino que la he tenido que encontrar en una gran variedad de documentos (algunos relevantes, otros descartados), lo que ha sido un trabajo nunca antes realizado y que aportaba un punto extra de dificultad. Finalmente, tras este desafiante trabajo y pese a en ocasiones no ver la luz al final del tunel, he conseguido obtener los resultados que en un principio se establecieron por lo que me encuentro satisfecho con el punto y final a los cuatro años dedicados a esta carrera.

Capítulo 6

Bibliografía

- [1] 9GAG. <https://9gag.com/gag/awB01EB>, 2020.
- [2] Wenzel Jakob. <https://www.mitsuba-renderer.org/>, 2014.
- [3] Shuichi Kinoshita; Shinya Yoshioka. Structural colors in nature the role of regularity and irregularity in the structure. 2005.
- [4] Diego Gutierrez; Adolfo Muñoz; Julio Marco; Daniel Martin. Lighting. *Informática Gráfica*.
- [5] Melodi Yenmiş; Yusuf Bayrakçı; Dincer Ayaz. Hierarchical microstructure of the scales in grass snake (*natrix natrix*) and dice snake (*natrix tessellata*). 2022.
- [6] Stanislav N. Gorb Marie-Christin G. Klein. Epidermis architecture and material properties of the skin of four snake species. 2012.
- [7] Matt Pharr; Wenzel Jakob; Greg Humphreys. Physically based rendering. 2016.
- [8] Wenzel Jakob; Eugene d'Eon; Otto Jakob; Steve Marschner. A comprehensive framework for rendering layered materials. *ACM Transactions on Graphics*, 2014.
- [9] Yu Guo; Milos Hasan; Shuang Zhao. Position-free monte carlo simulation for arbitrary layered bsdfs. 2018.
- [10] Open Source. <https://www.gimp.org/>, 2022.
- [11] CgTrader. <https://www.cgtrader.com/free-3d-models?keywords=snake>, 2019.
- [12] Facebook. <https://m.facebook.com/ZENERGYUNDERSON/photos/>.

- [13] Laurent Belcour; Pascal Barla. A practical extension to microfacet theory for the modeling of varying iridescence. *ACM Transactions on Graphics, Association for Computing Machinery*, 2017.
- [14] D. S. Dhillon; J. Teyssier; M. Single; I. Gaponenko; M. C. Milinkovitch 3 and M. Zwicker. Interactive diffraction from biological nanostructures. 2014.
- [15] Golnaz Isapour; Marco Lattuada. Bioinspired stimuli-responsive color-changing systems. 2018.
- [16] Jessica Renee Baron. Procedurally generating biologically driven bird and non-avian dinosaur feathers. Tesis, Clemson University, 2018.
- [17] <https://es.wikipedia.org/wiki/>.

Listado de Figuras

1.1.	Serpiente de la especie <i>Xenopeltis unicolor</i> [1]	2
1.2.	Diagrama de Gantt del proyecto	4
1.3.	Horas dedicadas por tarea del proyecto	4
2.1.	Interferencia de capa fina, representación del transporte de luz por las capas y ejemplo de la iridiscencia producida	8
2.2.	Sección aumentada de la Figura 1.1	9
2.3.	Componentes para trazado de rayos; se pueden ver las diferentes geometrías, la fuente de luz, la cámara, el plano, el sistema de coordenadas y los rayos trazados	10
2.4.	Elementos de la <i>Ecuación de Render</i>	10
2.5.	Ejemplo de aplicación de <i>Bump mapping</i> mediante textura	13
2.6.	A la izquierda un ejemplo de un <i>Path</i> ; se puede ver como este sigue un solo camino rebotando en los materiales sin dividirse en todos los posibles rayos hasta una última intersección. A la derecha ejemplo de seguimiento de trazado de rayos, donde se ve como se exploran todas las posibilidades.	14
2.7.	Imagen ejemplo de la asunción de mismo punto de salida y entrada pues Δx se asume 0; también muestra el camino simulado - <i>path</i> - seguido por un rayo, siendo los puntos amarillos los de cálculo del color y las flechas el camino que sigue [9]	15
3.1.	Modelo de apariencia multicapa final desarrollado	19
3.2.	Diagrama de clases del modelo de apariencia implementado y su acoplamiento a <i>Mitsuba</i>	20
3.3.	Textura final	21
3.4.	Representación gráfica de un ejemplo de absorción	24
4.1.	Modelos geométricos usados para el modelado de apariencia, con un material difuso gris	29

4.2.	Textura aplicada a los modelos geométricos con <i>Bump Mapping</i>	30
4.3.	<i>Interferencia de Capa Fina</i> aplicada a los modelos 3D simulando una burbuja. Parámetros: $n_0 = 1,0; n_1 = 1,56; n_2 = 1,0; \delta = 900$	30
4.4.	<i>Bump Mapping</i> aplicado sobre material con <i>Interferencia de Capa Fina</i>	31
4.5.	Material de absorción. Parámetros: coef-absorción = 0.7; anchura = 2 .	31
4.6.	Modelo de apariencia final sobre una esfera y un toroide	32
4.7.	Modelo de apariencia final sobre una serpiente	32
4.8.	Figuras con diferentes parámetros a los del modelo final. En la primera fila se puede apreciar como cambia la apariencia cuando se incrementa la profundidad de la capa; mientras que en la segunda fila se cambia el índice de refracción de la capa intermedia	33
4.9.	Comparación de una imagen real de la serpiente <i>Xenopeltis unicolor</i> con el modelo de apariencia final	34
A.1.	Material difuso; comportamiento de la luz al intersectar con él y ejemplo	47
A.2.	Ilustración de reflexión perfecta	48
A.3.	Material especular; comportamiento de la luz al intersectar con él y ejemplo	48
A.4.	Refracción perfecta (no existe en la realidad)	49
A.5.	Material dieléctrico; comportamiento de la luz al intersectar con él y ejemplo	50
B.1.	Diagrama de las capas de la epidermis en serpientes [6]	52
B.2.	Sección aumentada de la Figura 1.1	53
C.1.	Ondas en contrafase creando una interferencia destructiva	55
C.2.	Ondas en fase creando una interferencia constructiva	56
C.3.	Representación gráfica de un número complejo	56
C.4.	Interferencia de capa fina caso base	58
C.5.	Interferencia de capa fina caso general	59
C.6.	Interferencia de capa fina con el resultado de sus ondas reflejadas y transmitidas	60
D.1.	Ejemplo de un <i>Path</i> ; se puede ver como este sigue un solo camino rebotando en los materiales sin dividirse en todos los posibles rayos hasta una última intersección	61
D.2.	Ejemplo de uso del método <i>Monte Carlo</i> para la resolución de una integral	63
D.3.	Ejemplo de la resolución de un <i>path</i> con <i>Path Tracing</i>	63
D.4.	Ejemplos de tipos de luz	65

- D.5. Imagen ejemplo de la asunción de mismo punto de salida y entrada Δx ; también muestra el camino simulado -*path*- seguido por un rayo, siendo las estrellas los puntos de cálculo del color y las flechas el camino que sigue [9] 66

Anexos

Anexos A

Materiales Básicos

Los materiales básicos son aquellos con *BSDF* sencilla de calcular y cuya combinación da lugar a la gran mayoría del resto de materiales, estando entre ellos el referente al objetivo de este estudio.

Como materiales básicos nos encontramos el material difuso, el material conductor (especular) y el material dieléctrico (reflexión y refracción), diferenciándose como se ha comentado previamente en el comportamiento de la luz cuando incide sobre dicho material.

Los materiales de origen difuso son aquellos que reflejan la luz a partes iguales por toda la hemisferia tal y resuelven el factor f_r con la ecuación A.1 siendo k_d el coeficiente difuso que indica la cantidad de luz reflejada por el material difuso y π la hemisferia sobre la que se trazan todos los rayos de salida.

$$f_r = \frac{k_d}{\pi} \quad (\text{A.1})$$

En la Figura A.1 se ilustra el comportamiento de un material difuso y un ejemplo del mismo.

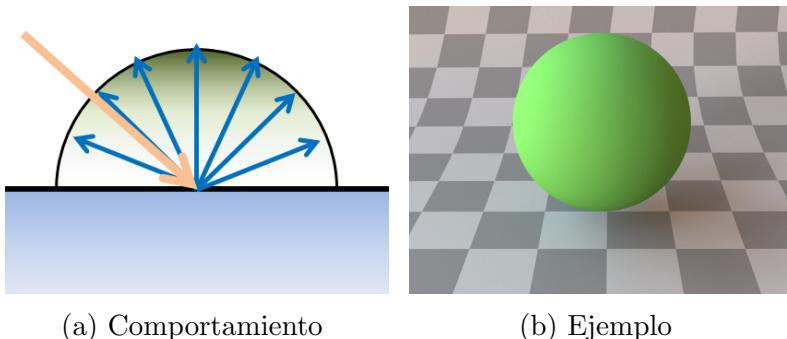


Figura A.1: Material difuso; comportamiento de la luz al intersectar con él y ejemplo

Los materiales conductores (especulares) son aquellos que reflejan la luz de manera perfecta, es decir, el ángulo del rayo de salida es el espejo (con respecto a la normal de la geometría) al ángulo del rayo de entrada, siguiendo la siguiente ecuación:

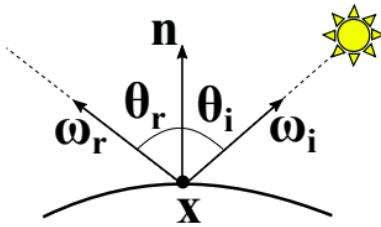


Figura A.2: Ilustración de reflexión perfecta

$$\theta_r = \theta_i \quad (\text{A.2})$$

Este tipo de materiales resuelven el factor f_r con la ecuación A.3 siendo δ una función con valor uno para un parámetro de entrada específico y 0 para todos los demás; y $wi \cdot n$ un factor encargado de anular el factor idéntico en la *Ecuación de Render* pues la dirección a la fuente de luz carece de importancia en estos materiales ya que el rayo saliente solo tiene una dirección posible.

$$f_r = \frac{\delta_r(w_i)}{w_i \cdot n} \quad (\text{A.3})$$

En la Figura A.3 se muestra a mano izquierda el comportamiento de la capa y a mano derecha un ejemplo de la misma.

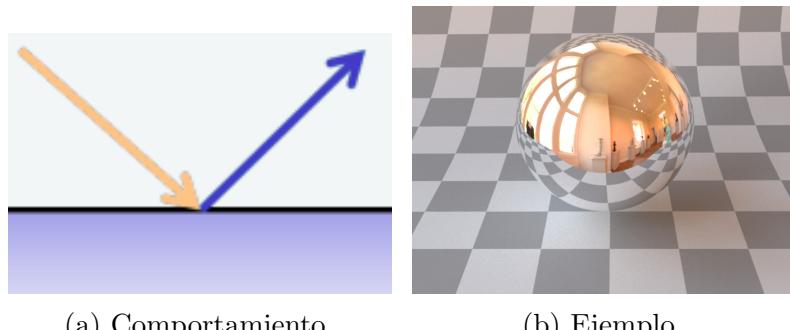


Figura A.3: Material especular; comportamiento de la luz al intersectar con él y ejemplo

Acabando con los 3 materiales básicos se encuentran aquellos denominados dieléctricos, esto es, que parte de la luz se ve reflejada y otra parte se ve refractada -transmitida- como sucede por ejemplo en las esferas de cristal.

Al propagarse la luz por dos medios separados por una capa de material dieléctrico hay que tener en cuenta el llamado índice de refracción, que se define como el cociente de la velocidad de la luz en el vacío y la velocidad de la luz en el medio sobre el que se quiera calcular ($n = \frac{c}{v}$) y cuyo uso es necesario para el cálculo del ángulo con el que el rayo se ve refractado.

Concretamente el uso de estos índices de refracción se encuentra en la *Ley de Snell*; ley que dados sendos índices de refracción pertenecientes a dos medios y el ángulo de

entrada, posibilita el cálculo del ángulo con el que el rayo se ve refractado. Siendo n el índice de refracción del medio indicado por su subíndice y θ los ángulos, se tiene que dada la imagen A.4 los cálculos de los ángulos vía *Ley de Snell* quedarían como en la ecuación A.4.

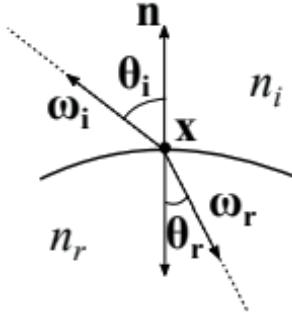


Figura A.4: Refracción perfecta (no existe en la realidad)

$$n_i \sin \theta_i = n_r \sin \theta_r \quad (\text{A.4})$$

En este tipo de material dieléctrico no basta con propagar los rayos de luz de la manera que indican las ecuaciones, sino que también hay que tener en cuenta la conservación de energía de forma más específica, es decir, la suma de energía del rayo reflejado y refractado ha de ser igual a la del rayo incidente y en ningún caso puede llegar a superarla. Es por ello que hay que usar las ecuaciones de *Fresnel*, las cuales relacionan las amplitudes de las ondas (cantidad de energía) reflejadas y refractadas en función de la amplitud de la onda (cantidad de energía) incidente. Antes de exponer dichas ecuaciones hay que saber qué es la luz s-polarizada y la p-polarizada:

- s-polarizada: la onda se propaga perpendicularmente, es decir, cambia de plano al perpendicular.
- p-polarizada: la onda se propaga paralelamente, es decir, se mantiene en el mismo plano.

Con esto en mente, y siguiendo la nomenclatura de la imagen A.4, a continuación se establecen las ecuaciones siendo r el coeficiente de amplitud reflejado y t el refractado -transmitido-, y con los subíndices indicando si es p-polarizado o s-polarizado:

$$r_s = \frac{n_i \cos \theta_i - n_r \cos \theta_r}{n_i \cos \theta_i + n_r \cos \theta_r}$$

$$t_s = \frac{2 n_i \cos \theta_i}{n_i \cos \theta_i + n_r \cos \theta_r}$$

$$r_p = \frac{n_r \cos \theta_i - n_i \cos \theta_r}{n_i \cos \theta_r + n_r \cos \theta_i}$$

$$t_p = \frac{2n_i \cos \theta_i}{n_i \cos \theta_r + n_r \cos \theta_i} \quad (\text{A.5})$$

Para la resolución del factor f_r en materiales dieléctricos se usa la ecuación A.6 donde el primer sumando representa la parte especular multiplicada por su coeficiente especular calculado con las ecuaciones de *Fresnel* (A.5) y el segundo la parte transmitida multiplicada por el coeficiente de transmisión calculado igualmente por las ecuaciones de *Fresnel* (A.5). Estos coeficientes se calculan con *Fresnel* (A.5) de manera que $k_s = \frac{r_s+r_t}{2}$ y $k_t = \frac{t_s+t_t}{2}$ pues se asume misma cantidad de energía p-polarizada y s-polarizada.

$$f_r = k_s(w_i) \frac{\delta_r(w_i)}{w_i \cdot n} + k_t(w_i) \frac{\delta_t(w_i)}{w_i \cdot n} \quad (\text{A.6})$$

A continuación en la FIgura A.5 se ilustra el comportamiento de un material dieléctrico y a su derecha el ejemplo de uno con un índice de refracción interno igual al del diamante (2.419):

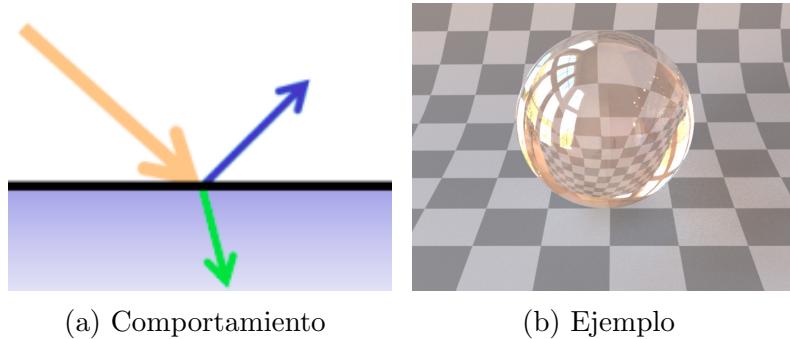


Figura A.5: Material dieléctrico; comportamiento de la luz al intersectar con él y ejemplo

Anexos B

Coloración y Características Estructurales en Serpientes

B.1. Características Generales

Las serpientes poseen diferentes características tanto a nivel macro-estructural como a nivel micro-estructural, cuyas propiedades están dotadas de un interés notable para este estudio [5, 6].

Hablando de la macro-estructura de las serpientes nos encontramos que estas, al contrario que una gran cantidad de seres vivos, poseen escamas con patrones estructurales de gran interés. En primer lugar hay que centrarse en la división más genérica por capas, la epidermis y la dermis. La primera de ellas es la capa exterior de la escama y por tanto la que impacta directamente con el medio, por lo que es más robusta; la segunda de ellas es por ende la interna y en la que se encuentran los vasos capilares y diversas glándulas. Dado que lo que interesa en este trabajo es el modelado de apariencia, es importante el estudio más a fondo de la capa exterior, es decir, la epidermis. Para un modelado de apariencia correcto es necesario el conocimiento de dos aspectos fundamentales sobre esta capa; el primero de ellos aborda el patrón que siguen las escamas, pues este difiere dependiendo de especies variando tanto la forma como la unión entre las mismas; el segundo se basa en la división en subcapas de la epidermis que nombradas de exterior a interior se obtiene: capa *Oberhäutchen*, capa *β-layer*, capa *mesos-layer*, capa *α-layer*, capa *lacunar tissue* y por ultimo capa *clear layer* tal y como se presenta en la Figura B.1. Como se ve, la capa *Oberhäutchen* es la exterior por lo que la micro-estructura se va a estudiar sobre ella.

Con esta división establecida es de fundamental interés llevar a cabo un estudio tanto a nivel macro-estructural como micro-estructural de la especie *Xenopeltis Unicolor* desde la cual se va a generar un modelo de apariencia, por lo que en la siguiente sub-sección se estudia senda estructura.

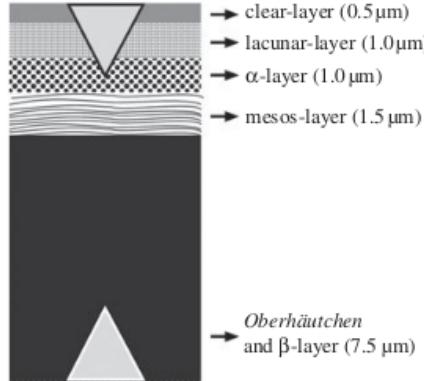


Figura B.1: Diagrama de las capas de la epidermis en serpientes [6]

B.2. *Xenopeltis Unicolor*

La serpiente *Xenopeltis Unicolor* es la especie sujeta a estudio y cuya apariencia se quiere modelar. Para dicho fin es importante conocer tanto su macro-estructura, para un correcto modelado de su patrón de escamas, como su micro-estructura para un modelado preciso de los colores iridiscentes obtenidos [14].

El estudio de la macro-estructura se centra en el patrón que siguen las escamas. Este estudio ha sido principalmente de origen visual, y las conclusiones obtenidas son que tal y como se puede ver en la Figura B.2 las escamas vistas en sí mismas poseen formas hexagonales con ligeras irregularidades (no son hexágonos regulares), mientras que vistas en conjunto se puede observar que estas forman un patrón simple; filas perfectamente encajadas de sendos hexágonos los cuales no llegan a tocarse entre ellos por mínimas separaciones.

Con la macro-estructura establecida ya se tiene el patrón de escamas a seguir durante la modelación, por lo que ahora es el turno del estudio de la micro-estructura, de gran importancia para la coloración de la serpiente.

La iridiscencia mostrada a la derecha de la Figura B.2 se obtiene gracias a la micro-estructura que posee la serpiente *Xenopeltis Unicolor*. Esta micro-estructura se basa en la existencia de iridóforos en la capa superior de las escamas del reptil (recordemos denominada *Oberhäutchen*); estos iridóforos constan de pequeñas láminas cristalinas que reflejan y refractan la luz, dando lugar a que las ondas de luz interfieran y por ende se modifiquen creando estos colores tan llamativos que se pueden llegar a ver. Si se tiene en cuenta lo mencionado en la sección de *Coloración estructural 2.1*, se ve que la descripción de los iridóforos encaja con la obtenida en la referente a *Interferencia de Capa Fina*.

Cabe destacar que debajo de estos denominados iridóforos se encuentra una capa de pigmentos oscura que otorga ese color ennegrecido a las escamas cuando la iridiscencia es menor o nula. Cuando la luz penetra por esta capa lo hace por un espacio reducido hasta incidir sobre el fondo de la misma, momento en el cual la luz rebota y comienza su camino a la superficie. Durante el camino por esta capa la luz sufre la denominada absorción, es decir, pierde energía. Se tiene pues aquí que esta capa de pigmentos oscura se puede dividir en dos, una superficial que modele la absorción y una segunda interior que modele el fondo de la misma y sea la encargada de aportar el color natural de la serpiente.



Figura B.2: Sección aumentada de la Figura 1.1

Anexos C

Interferencia de Capa Fina

Interferencia de Ondas

Para entender este fenómeno, es necesario saber lo que es la interferencia de ondas, por qué se produce y qué genera. La interferencia de ondas sucede cuando un par de ondas se encuentran por el espacio, esto hace que ambas se superpongan generando una onda resultante formada por la suma de ambas superpuestas, por lo que es necesario tener en cuenta tanto la amplitud de sendas ondas como la fase en la que se encuentran.

En primer lugar existe la interferencia de ondas destructiva, esta se da cuando ambas ondas tienen la misma amplitud pero están en contra fase tal y como se puede ver en la imagen C.1. El resultado es la anulación de la ondas, es decir, la amplitud de la onda resultante es cero. Vease que si no tuviesen la misma amplitud la onda no se destruiría hasta el punto de anularse.

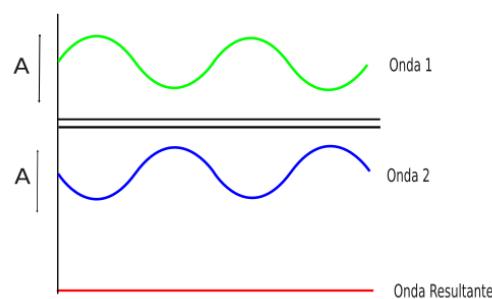


Figura C.1: Ondas en contrafase creando una interferencia destructiva

Por otro lado existen las ondas constructivas, dadas cuando ambas ondas superpuestas se encuentran en la misma fase tal y como se puede ver en la imagen C.2. Lo que sucede en este tipo de interferencia es que al estar en fase la suma de las ondas hace que la resultante obtenga un aumento de amplitud siendo este la suma de ambas.

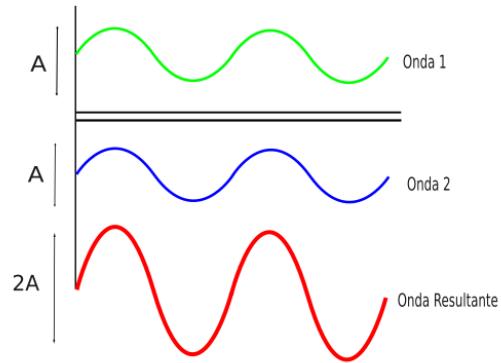


Figura C.2: Ondas en fase creando una interferencia constructiva

Para poder llevar a cabo los cálculos de interferencia de ondas es necesario formularlas. Una onda es matemáticamente representada por la ecuación C.1 donde A es la amplitud de onda, w es la frecuencia angular, t es el instante de tiempo y ϕ la fase de la onda.

$$y(t) = A e^{i(wt+\phi)} \quad (\text{C.1})$$

Esta representación de números complejos no es tan habitual pero tiene una explicación. Dada la imagen C.3, si se establece z como el número complejo $z = b + ia$ se puede calcular por trigonometría básica que $b = r \cos \theta$ y $a = r \sin \theta$. Lo que implica que si se sustituyen a y b en la ecuación del número complejo quedaría $r = r(\cos \theta + i \sin \theta)$. Por tanto, dada la fórmula de *Euler* (C.2) se tiene que $z = re^{i\theta}$, fórmula en la que se basa la representación de ondas vista en la ecuación C.1.

$$e^{i\theta} = \cos \theta + i \sin \theta \quad (\text{C.2})$$

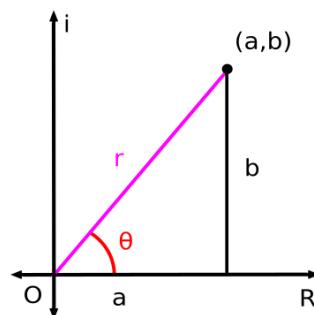


Figura C.3: Representación gráfica de un número complejo

Dada esta formulación, el cálculo de la onda resultante por la superposición de otras dos se puede calcular como la suma de dos ondas representadas con la fórmula C.1.

Con la interferencia de ondas establecida, se puede ver que los rebotes dentro de la capa que genera la *Interferencia de Capa Fina* van a hacer que la amplitud de onda cambie y por tanto la cantidad de energía que esta posee. Como se ha comentado anteriormente este cálculo de interferencia se lleva a cabo con ondas de misma longitud de onda, lo que quiere decir que se calcula para cada color con el que se quiera renderizar la imagen en cuestión, teniendo en este caso el sistema RGB se calcula entonces la cantidad de energía restante para cada color; va a ser esto lo que produzca esos patrones de colores.

Interferencia de Capa Fina

Una vez se ha explicado qué es la interferencia de ondas, se va a ver cómo se produce en *Interferencia de Capa Fina*. Vease que lo que se quiere obtener es la amplitud para cada canal del espectro RGB, pues esto dará la cantidad de energía restante en cada color.

Establezcamos que se tienen 3 materiales numerados; el material 0 es el material externo; el 1 corresponde al material intermedio; y por último el 2 corresponde al material interno. Con esta numeración, se va a establecer una nomenclatura necesaria:

- δ : representa la profundidad de la capa en cuestión.
- n : representa el índice de refracción de la capa en cuestión (cabe destacar que dos capas consecutivas han de diferir en este parámetro).
- θ : representa el ángulo de incidencia del rayo por capa.

Sabiendo que por la *Ley de Snell* (A.4) se pueden calcular los ángulos θ_k de incidencia en cada capa, estos no van a suponer un problema pues su cálculo va a ser constante. También se tiene que por las ecuaciones de *Fresnel* (A.5) se puede calcular la amplitud de onda resultante reflejada y refractada -transmitida- por lo que se puede calcular de manera constante la cantidad de energía en cada caso. Para relacionar las ecuaciones de *Fresnel* (A.5) con cada medio en cuestión, se va a establecer que r_{ij} es el coeficiente de amplitud reflejado del medio i al j con la polarización requerida y lo mismo pero referente a la refracción con t_{ij} .

Como caso base de interferencia de ondas se encuentra el que se muestra en la Figura C.4 en el que la onda incidente refracta del medio 0 al 1, refleja del medio 1 al

2 y seguidamente del 1 al 0 y concluye refractándose del medio 1 al 2. En resumen se puede ver 4 puntos de intersección con las capas, en dos de ellos se refracta (primer y último) y en los dos restantes se refleja (segundo y tercero). Por las ecuaciones de *Fresnel* en cada punto la energía se ve dividida entre la reflejada y la refractada por lo que el cálculo de la energía en una intersección dada es el producto de las ecuaciones de *Fresnel* siguiendo el mismo camino que la onda en cuestión, lo cual se resuelve en la siguiente ecuación:

$$t_{0|1} r_{1|2} r_{0|1} t_{1|2} = t_{0|1} t_{1|2} r_{1|0} r_{1|2}$$

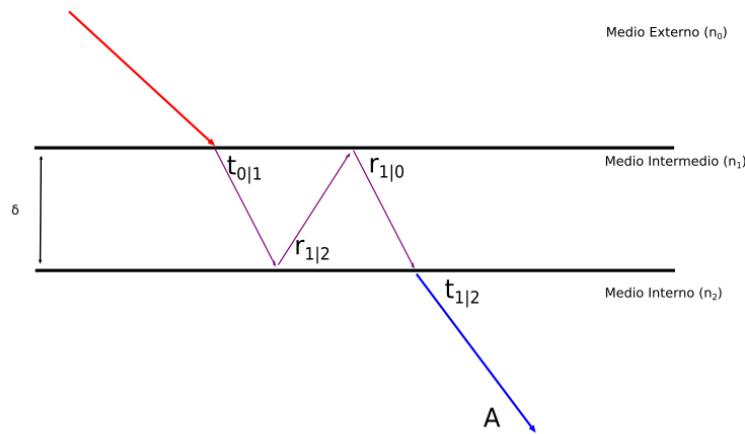


Figura C.4: Interferencia de capa fina caso base

Por tanto denotando A_k como la amplitud de la onda k transmitida al medio interno, se puede ver que el número que refracciones que ocurren durante su camino son siempre dos, la de entrada al medio intermedio y la de entrada al medio interno; mientras que el número de reflexiones varía dependiendo de la k de manera que en $k+1$ hay únicamente dos más que en k una del medio 1 al 2 y otra del medio 1 al 0 sucediéndose siempre el mismo patrón como se puede ver en la Figura C.5. Al sucederse dos reflexiones más y estas ser entre los mismos medios se tiene que $A_{k+1} = A_k * r_{1|0} * r_{1|2}$, desembocando esto en la siguiente ecuación para k rebotes:

$$A_k = t_{0|1} t_{1|2} r_{1|0}^k r_{1|2}^k$$

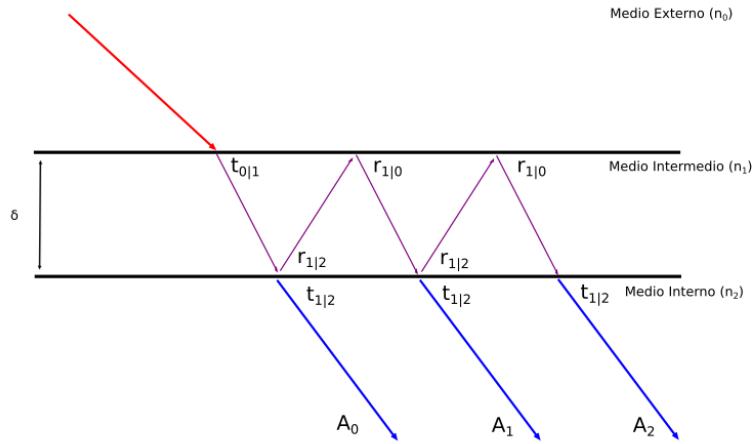


Figura C.5: Interferencia de capa fina caso general

Con esto se tiene el resultado de la amplitud refractada. Además, como se ha comentado en la sección referente a interferencia de ondas (ver en C) la fase en la que se encuentren ambas ondas al superponerse es importante. Si se tiene en cuenta que k es la k -ésima onda transmitida y se denota ϕ_k como su fase, la siguiente fórmula C.3 da el resultado de la fase en la que se encuentra dicha onda.

$$\phi_k = k \left[\frac{2\pi}{\lambda} (2n_1 \delta \cos \theta_1) + \Delta \right] \quad (\text{C.3})$$

En la fórmula C.3 cada símbolo es constante dependiendo de la capa en la que se sitúa y significa:

- λ : longitud de onda, constante pues previamente comentado que se realiza para cada color del espectro RGB de manera individual.
- Δ : cantidad de cambio de fase en reflexión. Teniendo en cuenta que el índice de refracción es una medida de la densidad de un medio, se puede calcular esta constante con ello.

$$\Delta_{i|j} = \begin{cases} 0 & \text{si } n_i > n_j, \\ \pi & \text{si } n_i < n_j, \end{cases}$$

Y por tanto al ser reflejada dos veces antes de ser transmitida se tiene la ecuación:

$$\Delta = \Delta_{1|0} + \Delta_{1|2}$$

- 1: el subíndice 1 de las capas implica que se tiene en cuenta la capa intermedia, desde la que pasan a la interior.

Como las ondas transmitidas se encuentran en un medio diferente al de incidencia hay que tener en cuenta el ratio de indices de refracción debido a la diferencia de los

mismos y el ratio de cosenos debido al cambio de ángulo de la luz refractada para la conservación de energía tal y como se muestra en la siguiente ecuación:

$$\frac{n_2 \cos \theta_2}{n_0 \cos \theta_0} \quad (\text{C.4})$$

Una vez se tiene el cálculo de la amplitud, de la fase y del ratio de conservación de energía, el resultado final es la ecuación siguiente en la que el primer factor corresponde al ratio de conservación y el segundo calcula la interferencia de ondas para k rebotes siguiendo la formulación vista de las ondas:

$$I_T = \frac{n_2 \cos \theta_2}{n_0 \cos \theta_0} \left| \sum_{k=0}^{\infty} A_k e^{i\phi^k} \right|^2$$

Que simplificada se queda en:

$$I_T = \left(\frac{n_2 \cos \theta_2}{n_0 \cos \theta_0} \right) \frac{|t_{0|1} t_{1|2}|^2}{|r_{1|0} r_{1|2}|^2 - 2r_{1|0} r_{1|2} \cos \phi + 1} \quad (\text{C.5})$$

Y como hay que tener en cuenta la conservación de energía se tiene:

$$I_T + I_R = 1 \quad (\text{C.6})$$

Una vez resuelto esto se ha resuelto el problema tratado, concretamente se han obtenido los coeficientes de refracción -transmisión- I_T y reflexión I_R representados en la Figura C.6, es decir, la cantidad de luz reflejada y refractada -transmitida-, más concretamente la amplitud de la onda reflejada o refractada -transmitida- con respecto a la incidente.

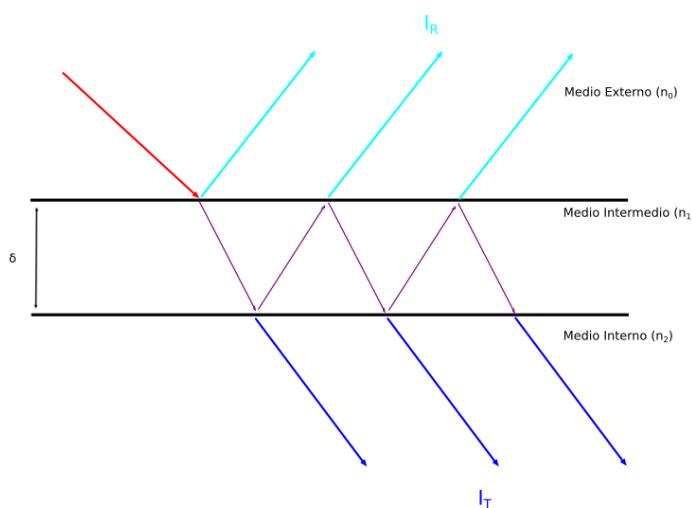


Figura C.6: Interferencia de capa fina con el resultado de sus ondas reflejadas y transmitidas

Anexos D

Path Tracing y Position-Free Monte Carlo

D.1. Path Tracing

Este método de simulación del transporte de luz es el usado durante el trabajo, lo cual quiere decir que todas las imágenes obtenidas lo han sido gracias al uso de *Path Tracing*, por lo que es importante tener unos conocimientos básicos sobre ello [7].

Definición de Path

Path Tracing es una técnica basada en trazado de rayos, anteriormente comentado en la sección 2.3. La gran característica de este tipo de simulación es que se simula un solo camino del rayo denominado *path*, obteniendo en cada intersección una dirección aleatoria de salida que corresponda con el material en cuestión hasta que se encuentre una fuente de luz o se absorba por completo. En la Figura D.3 se ve el ejemplo de un *path*.

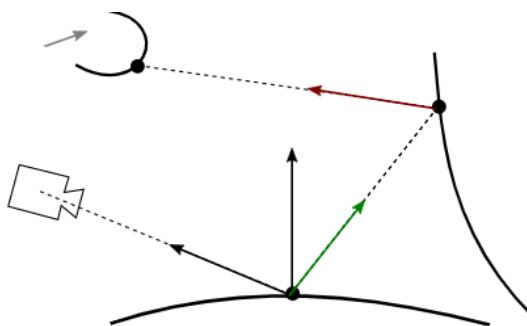


Figura D.1: Ejemplo de un *Path*; se puede ver como este sigue un solo camino rebotando en los materiales sin dividirse en todos los posibles rayos hasta una última intersección

D.1.1. Path Integral

Para la resolución de todo el conjunto de *paths* que se trazan en una imagen en el algoritmo *Path Tracing*, se va a denotar un *path* con sus k puntos de intersección de la manera $\bar{x} = (x_0, \dots, x_k)$, siendo x_n cada punto de intersección, por lo que en resumen viene definido por un conjunto de segmentos. Con ello, la ecuación de la resolución de la radiancia en *Path Tracing* queda de la manera que se ve en la ecuación D.1, donde I es la radiancia resultante, Ω es el conjunto de *paths*, f es la contribución del *path* en cuestión y μ es la medida del producto de las áreas de intersección de cada una de siendo *path*.

$$I = \int_{\Omega} f(\bar{x}) d\mu(\bar{x}) \quad (\text{D.1})$$

La contribución de cada *path* denotada con f en la ecuación anterior (D.1) viene dada por la resolución de la *Ecuación de Render* (2.1) para cada *path*. Por tanto, la resolución del factor f para un único *path* viene dada por la expresión D.2 donde $L_e(x_k)$ es la luz emitida por el último objeto intersectado que será siempre una fuente de luz o la absorción completa y por tanto cero; y el productorio indica la resolución recursiva del *path* siendo que el factor L_i de la *Ecuación de Render* (2.1) viene dado por la contribución de la siguiente intersección del *path*, por lo que se van acumulando multiplicaciones hasta llegar al objeto emisor de luz o a la absorción.

$$f(\bar{x}) = L_e(x_k) * \left(\prod_{j=1}^{k-1} f_r(x, w_{ij}, w_{obj}) * |n_j \cdot w_{ij}| \right) \quad (\text{D.2})$$

D.1.2. Monte Carlo

La integral denominada *Path Integral* (D.1) introducida anteriormente es una integral cuyo rango está en Ω , es decir, todos los posibles *paths* en la imagen. La resolución de tal integral es de una magnitud cuantiosa por lo que su resolución ha de basarse en una técnica de resolución de integrales numéricas de manera aproximada.

Monte Carlo es una técnica aproximación de expresiones matemáticas complejas como por ejemplo la integral nombrada. Este método se basa en la obtención de puntos a evaluar aleatorios denominados muestras para la resolución de dichas integrales con el fin de obtener de esa manera la resolución aproximada. Vease que la resolución de la integral de la Figura D.2a aplicando el método de *Monte Carlo* se vería resuelta de la manera vista en la Figura D.2b. Por tanto, la resolución de una integral por el método de *Monte Carlo* viene dada por la Fórmula D.3 donde N el número de muestras utilizadas y $prob(x_i)$ es la función de distribución de probabilidad usada para la obtención de muestras también denominada *pdf* (*probability density function*).

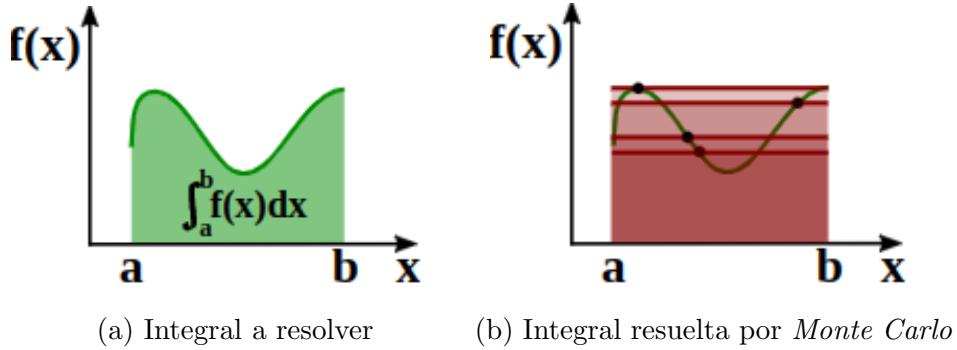


Figura D.2: Ejemplo de uso del método *Monte Carlo* para la resolución de una integral

$$\int_V f(x) \approx I = \frac{1}{N} * \sum_{i=1}^N \frac{f(x_i)}{prob(x_i)} \quad (\text{D.3})$$

Dada la *Ley de los Grandes Números* [17] que afirma que dado un número infinito de muestras, un ensayo aproximado obtiene el valor correcto, se tiene que el método de *Monte Carlo* converge dado el límite D.4.

$$\lim_{x \rightarrow -\infty} I = \int_V f(x) \quad (\text{D.4})$$

D.1.3. Path Tracing

Para la resolución de *Path Tracing* basta con resolver la *Path Integral* aplicando *Monte Carlo* de manera que el conjunto de muestras N es el número de *paths* muestreados y la función de distribución de probabilidad *pdf* es la usada para obtener cada *path*.

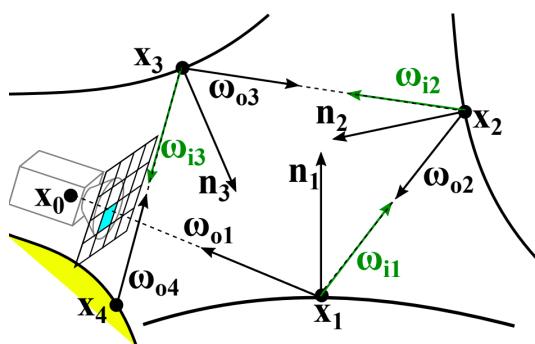


Figura D.3: Ejemplo de la resolución de un *path* con *Path Tracing*

Vease que para la Figura D.3 que ejemplifica el recorrido de un *path*, dada la *Path Integral* y la aproximación por *Monte Carlo* la resolución de la raciancia L_o vendría dada por la ecuación D.5 donde $\text{prob}(w_{i1})$ es la anteriormente mencionada *pdf* usada por *Monte Carlo*. La ausencia de integral se debe a que se está siguiendo un solo camino.

$$L_o(x_1, w_{o1}) = \frac{L_i(x_1, w_{i1}) f_r(x, w_{i1}, w_{o1}) |n_1 \cdot w_{i1}|}{\text{prob}(w_{i1})} \quad (\text{D.5})$$

Siendo que el factor $L_i(x_1, w_{i1})$ es la cantidad de luz que recibe el punto x_1 desde la dirección w_{i1} esta se va a resolver aplicando la *Ecuación de Render* recursivamente de manera que se simula que el observador ahora se encuentra en el punto x_1 , quedando pues la resolución de $L_i(x_1, w_{i1})$ como en la ecuación D.6.

$$L_i(x_1, w_{o2}) = \frac{L_i(x_2, w_{i2}) f_r(x, w_{i2}, w_{o2}) |n_2 \cdot w_{i2}|}{\text{prob}(w_{i2})} \quad (\text{D.6})$$

Esta recursividad seguiría hasta encontrarse una fuente de luz o producirse absorción, donde el factor L_i quedaría resuelto por la ecuación D.7.

$$L_i(x_4, w_{o4}) = L_e(x_4, w_{o4}) \quad (\text{D.7})$$

D.1.4. Luz Directa e Indirecta

Para el correcto cálculo de la radiancia en un punto hay que tener en cuenta las dos divisiones de tipos de luz, la luz indirecta y la luz directa.

La luz indirecta es aquella que viene dada por la resultante de calcular la radiancia del siguiente punto de intersección del *path*, es decir, es la que viene calculada por la *Path Integral* (D.1) pues se calcula de manera recursiva hasta haber alcanzado la fuente de luz. Para que tal cálculo sea posible es necesario llevar a cabo un **muestreo** de la *BSDF* es decir, un cálculo del rayo de salida con respecto al de entrada teniendo en cuenta el tipo de material sobre el que se ha intersectado. Es gracias a este cálculo del rayo de salida que se va a poder continuar el *path* y por ende seguir intersectando hasta el momento de intersectar con una fuente de luz y que se pueda resolver la *Path Integral*, por lo que el muestreo de la *BSDF* es esencial. En la Figura D.4a se puede ver un ejemplo de luz indirecta.

Por otro lado, la luz directa es aquella que solo tarda un rebote en llegar al observador, por lo que la unión entre el punto de intersección y la fuente de luz ha de carecer de objetos entre la misma. Tal iluminación puede darse en dos ocasiones, aleatoriamente o de manera intencionada, siendo esta última la denominada **evaluación** de una *BSDF*. Esto quiere decir que en ocasiones, cuando la fuente de luz es muy difícil de encontrar de manera aleatoria es necesario encontrarla de manera intencionada o la imagen no se verá iluminada, por lo que se lleva a cabo la evaluación de la *BSDF* que se basa en que en cada intersección, posteriormente

a haberse muestreado la *BSDF* en tal punto, se le añade a la radiancia indirecta la incidente de luz directa correspondiente a la distancia entre fuente de luz e intersección por la *Ley de los Cuadrados Inversos* la cual divide la radiancia de la luz directa entre la distancia al cuadrado. En la Figura D.4b se puede ver un ejemplo de luz directa.

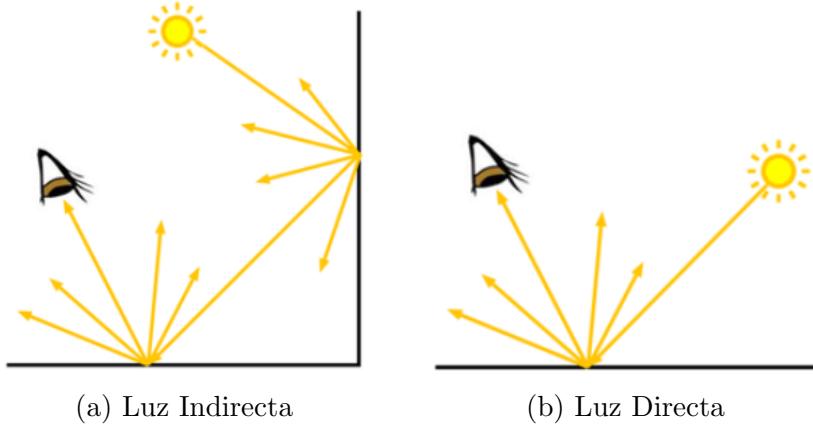


Figura D.4: Ejemplos de tipos de luz

D.2. Position-Free Monte Carlo

En la Sección 2.4.1 se hace referencia a los materiales multicapa y a que existen diferentes formas de simular el transporte de la luz por ellos. *Position-Free Monte Carlo* [9] es un método de simulación del transporte de la luz en materiales multicapa, modelando desde las intersecciones con las capas hasta el camino que sigue la luz entre ellas.

Concretamente este método trata de evaluar una *BSDF* (obtener el factor f_r de la *Ecuación de Render*, vease 2.1) aplicando la técnica *Path Tracing* (D.1) dentro del propio modelo multicapa para simular el paso de la luz por las capas intersectando tanto con las propias capas como con el medio entre las mismas. La diferencia con *Path Tracing* tradicional es que la *Path Integral* usada es *position-free* es decir, se asume que aunque los puntos se encuentren en lugares diferentes con respecto al eje x la variación en tal eje va a ser tan pequeña que se asume inexistente, en conclusión, no importa la posición en el eje x pues se asume en todas las intersecciones del *path* la misma. Esto hace que la definición de *path* cambie y esta venga dada en vez de por puntos de intersección por vértices formados por pares de profundidad y dirección. La profundidad denotada con $z \in [0, 1]$ indica la profundidad de la capa en la que se encuentra la intersección mientras que la dirección denotada con d denota la dirección

de salida de la luz entre vértices. Esto implica que la definición de un *path* viene dada por $\bar{x} = d_0, z_1, d_1, \dots, z_k, d_k$. En la Figura D.5 se encuentra el ejemplo de un *path* que evalua una *BSDF* mediante *Position-Free Monte Carlo*.

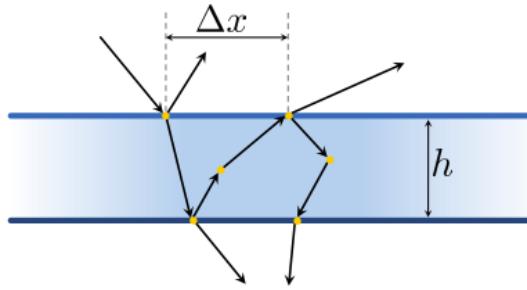


Figura D.5: Imagen ejemplo de la asunción de mismo punto de salida y entrada Δx ; también muestra el camino simulado -*path*- seguido por un rayo, siendo las estrellas los puntos de cálculo del color y las flechas el camino que sigue [9]

Finalmente se tiene que la evaluación queda resuelta pues con la ecuación D.8 donde (w_i, w_o) indica que se está evaluando la *BSDF* con tales rayos de incidencia y de salida.

$$f_r(w_i, w_o) = \int_{\Omega(w_i, w_o)} f(\bar{x}) d\mu(\bar{x}) \quad (\text{D.8})$$