

UNIVERSIDAD DE GRANADA
Programa Oficial de Postgrado en
Desarrollo de Sistemas Software
Departamento de Lenguajes y Sistemas Informáticos



UN ALGORITMO DE MUESTREO POR IMPORTANCIAS PARA
BRDFs ARBITRARIAS APLICADO A TÉCNICAS DE
ILUMINACIÓN GLOBAL

MEMORIA DE TESIS DOCTORAL
CON MENCIÓN EUROPEA
PRESENTADA POR

ROSANA MONTES SOLDADO

DIRECTOR DE TESIS
CARLOS UREÑA ALMAGRO

Editor: Editorial de la Universidad de Granada
Autor: Rosana Montes Soldado
D.L.: GR. 1217-2008
ISBN: 978-84-691-4088-8

A mis padres, Antonio y María

Agradecimientos

Tempus Fugit hace referencia al paso del tiempo, tan rápido, a veces. Pienso en los momentos y en todo el tiempo dedicado a todo el proceso de la tesis. También pienso en los bajones sufridos y las gratas finalizaciones de todas las tareas impuestas. Todo se ha ido superando y llega el final de esta tesis, que por fin se ve culminada con su lectura y publicación. También es el momento en el que dejo de pensar en los días dedicados y viene a mi memoria la gente que ha estado a mi lado. Gente que me inspira, gente que me anima, otros que me escuchan y los que están ahí, como mi familia, padres, sobrinos y hermanas, a pesar de no comprender ni una sola de las fórmulas aquí presentes.

He de agradecer la experta dirección de Carlos Ureña, que claramente ha tenido cierta paciencia conmigo. Dedico también unas líneas al buen hacer del grupo de gráficos y nuestros entrenidos *Cafés GIIG* con gente como Juan Carlos Torres, Germán Arroyo, Paco Velasco o Domingo Martín entre otros. Gracias a mis compañeros del subgrupo de *rendering*: Jorge Revelles, Miguel Lastra, y José Miguel Mantas. Poco a poco el *software* del repositorio ha ido enriqueciéndose por sus trabajos y sobre una base de utilidades construida por todos. Y no me olvido de Rubén García, que siempre me ayudó con las cosas más raras del C++, a recuperar Linux si se iba y mejorar el inglés de los artículos. Gracias igualmente a Alejandro León y Javier Melero por nuestras charlas y discusiones sobre gráficos y su implementación. Ellos incluían una perspectiva que complementaba mi visión y me permitía seguir adelante con nuevas ideas.

Esta tesis va dedicada especialmente a todos los miembros del *despacho* que me recibieron con los brazos abiertos a esta universidad. Con un ambiente tan bueno de trabajo, simpatía y con mucho cariño, cómo no se iba a trabajar bien. Chicos, por fin esto ha salido, la *festejos* promete fiesta para celebrarlo. Un besito.

Agradezco haber tenido la ocasión de compartir asignatura con gente tan maravillosa de la facultad de Filosofía y Letras: Mercedes Montoro, M^a José Martos y, en especial, Rodrigo López Carrillo. Rodrigo, debes ser la persona que más veces ha leído y releído esta tesis. Gracias por esas correcciones gramaticales y de semántica que han sabido dar luz a mi expresión escrita.

Dedico esta tesis a mi familia, a mi novio, Carlos Moral, y a todas mis amistades, que *alhamdulillah* —como diría Abdelaziz—, son muchas, son muy buenas y enriquecen mi vida en todos los sentidos.

Resumen

Hoy en día existen multitud de aplicaciones para el campo de la Iluminación Global: desde los modernos efectos especiales de las películas hasta el diseño de luminarias para interiores, pasando por los trepidantes videojuegos cargados de acción y realismo. La dispersión de la luz es un efecto importante responsable del aspecto de una superficie, característica distintiva de muchos materiales. En este sentido la función BRDF captura el comportamiento de la luz cuando alcanza un material y se refleja en varias direcciones a partir de éste.

El objetivo a cumplir en este trabajo consiste en obtener, mediante ordenadores, imágenes visualmente realistas, creando para ello programas de cálculo numérico que realizan una simulación de la propagación de la luz [Mil84, WRC88, GSCH93, GTGB84, CWH93]. El uso de estos programas permite obtener imágenes de ordenador con un aspecto más natural, similar a las fotografías. Mediante algoritmos de iluminación global como los sistemas de trazado de rayos [Whi80, CPC84, Kaj86, WH92], y aplicando métodos de Monte Carlo, es posible simular computacionalmente estas características. Dado que estos algoritmos de cálculo suelen presentar altos tiempos de ejecución, intentamos diseñar e implementar técnicas que los hagan más eficientes. Los algoritmos de Monte Carlo conllevan un error intrínseco en el estimador, que es evitable empleando un número elevado de muestras aleatorias. Esto, a su vez, implica un alto coste computacional que hace difícil el uso de técnicas de iluminación global de forma práctica.

En la actualidad, se utilizan las técnicas que han conseguido aumentar la eficiencia de los estimadores de Monte Carlo [VG94, Sbe96, Vea97, BSW00, ARBJ03, BGH05, JTE05, CJAMJ05, CETC06], que permiten obtener menos error para un número determinado de muestras, empleando a su vez menos tiempo. Motivados por este mismo interés hemos desarrollado el trabajo de esta tesis. De forma más específica, los resultados obtenidos pretenden aumentar la eficiencia del estimador proporcionando información exacta sobre una parte del estimando, que se traduce en un muestreo eficiente de la función de reflectancia de la superficie.

Para poder acometer este objetivo, nos preocupamos de conocer los modelos de reflexión más representativos, los más utilizados, así como los más complejos. Nos

dimos cuenta de la problemática existente: no todos los modelos de reflexión cuentan con un método de muestreo en base a la propia función en sí. Por otro lado, no existía en la fecha, un formato de escena apto para enriquecer la información geométrica de la escena con otra información más específica: color por vértice, reflectancia de una superficie, color de irradiancia, etc. Tras seleccionar un conjunto representativo de modelos de reflectancia de la superficie, desarrollamos herramientas para su definición y aplicación a escenas, como *software* de utilidad a un desarrollador gráfico, que disponía de un conjunto amplio de modelos para aplicar a sus escenas. Otras utilidades permiten la edición de los parámetros de la BRDF y su posterior visualización aplicado a un modelo. Llegados a este punto, disponíamos de suficientes facilidades para desarrollar un algoritmo de muestreo integrado en un sistema de simulación del transporte de la luz mediante un algoritmo de trazado de partículas.

De forma esquemática, las principales aportaciones de este trabajo son:

- 1. Estudio de los distintos modelos de reflexión.** Gracias al estudio realizado se ha conseguido agrupar y definir los parámetros de un número elevado de funciones BRDF. Al homogeneizar la notación el lector tiene mayor capacidad de comparar entre distintos modelos, al saber su significado exacto. En la memoria se muestran de forma cronológica, ayudando al lector a sumergirse en la capacidad de simulación de materiales de la informática gráfica para cada década.
- 2. Editor de escenas para sistemas de Iluminación Global.** Mediante la herramienta multiplataforma *Wannabe Amazing* [MUR⁺02] podemos importar escenas procedentes de diversos formatos gráficos. Se le añade información de importancia en el contexto de la síntesis realista de imágenes, como es la instancia de BRDF que se asocia a cada superficie. Todo esto y opcionalmente otra información de utilidad, se exporta en un formato gráfico de uso interno: *GRF*.
- 3. Editor de materiales.** Se han desarrollado varias herramientas para la comprensión y el uso adecuado de los parámetros de un conjunto considerable de modelos de BRDFs. De esta forma asistimos al diseñador gráfico, a la edición de parámetros en su mayoría poco intuitivos. Desde el punto de vista de función matemática se disponen de gráficas en coordenadas cartesianas y polares, y visualización en 3D actualizables a cambios en los parámetros. Desde el punto de vista del resultado visual se ha implementado en GPU modelos de BRDFs para la edición de esta en tiempo real.
- 4. Muestreo eficiente de modelos generales de BRDFs.** El núcleo de este trabajo reside en la creación de un algoritmo eficiente de muestreo que sea independiente del modelo concreto de BRDF que se desee utilizar. Trabajos previos existentes proporcionan técnicas individualizadas de muestreo aptas sólo para un subconjunto de BRDFs [War92, LW94, NN98, AS00a]. En ningún caso existía una técnica que permitiese de forma exacta el muestreo de cualquier función de reflectancia en un sistema basado en Monte Carlo. El resultado ha sido favorable y se ha

comparado numéricamente con otras técnicas de muestreo aproximado [LRR04]. Si bien el algoritmo de muestreo fue concebido para modelos analíticos y generales de BRDFs, ha sido a su vez probado con BRDFs adquiridas y disponibles como base de datos públicas [MPBM03].

5. Muestreo del producto de la BRDF y la iluminación. Se ha visto que nuestro algoritmo adaptativo de muestreo combina favorablemente con otras técnicas basadas en el muestreo por importancias. Nos referimos de forma concreta al muestreo conjunto de la radiancia incidente (procedente de un mapa de entorno (*environment map*) por la BRDF [ARBJ03, BGH05, JTE05, CJAMJ05, CETC06]. Este estimador de Monte Carlo usa más información sobre el integrando que el muestreo por sí sólo de la BRDF, proporcionando imágenes con un grado de realismo adecuado para su aplicación.

La presente memoria de tesis doctoral desarrolla con mucha más extensión los anteriores contenidos. El capítulo 1 explica los fenómenos físicos de reflexión de la luz sobre los que se basan los modelos simplificados que suelen emplearse en computación. Le sigue el capítulo 2 en donde se introduce la terminología utilizada en iluminación global, con el objetivo principal de entender el modelo simplificado del transporte de la luz que se utiliza en informática gráfica. El capítulo 3 realiza un estudio de los distintos modelos de reflexión existentes, considerando únicamente BRDFs empíricas. De cada uno de ellos se comenta lo más reseñable, manteniendo la perspectiva de su uso en un algoritmo de síntesis realista. Posteriormente, ya en el capítulo 4, se introducen los métodos de Monte-Carlo, con la finalidad de resolver la ecuación de transporte y se muestran las ideas básicas de algunos de los métodos de cálculo que lo aplican. En el capítulo 5 nos centramos en uno de los aspectos de los algoritmos de Monte-Carlo que mayor importancia tiene en la consecución de imágenes libres de ruido en tiempos razonables. Nos referimos al muestreo eficiente de la BRDF. Tras comentar las técnicas utilizadas, presentamos finalmente en el capítulo 6, un método de muestreo para BRDFs genéricas, cuya finalidad consiste en lograr que un sistema de síntesis de imágenes realistas trabaje con geometría compleja y, al mismo tiempo, utilice modelos genéricos de BRDFs, con los que pueda describir de forma precisa las características reflexivas de los materiales de una escena para su posterior visualización.

Confío en que el lector encuentre detalladas las partes de mayor interés, y disculpe las limitaciones en la expresión que haya podido cometer.



Rosana Montes

Índice

1. Introducción	1
1.1. La visualización realista	1
1.2. La naturaleza dual de la luz	2
1.2.1. Propagación de la luz	5
1.2.2. Comportamiento de la luz al incidir en una superficie	5
1.3. Caracterización de las superficies	6
2. Modelos de reflexión en <i>iluminación global</i>	7
2.1. Representación de la luz en <i>iluminación global</i>	7
2.1.1. Notación matemática	8
2.1.2. Densidad del flujo radiante	11
2.1.3. Radiancia	12
2.2. Dispersión de la luz en la superficie	13
2.2.1. Ley de reflexión	13
2.2.2. Ley de refracción	14
2.2.3. La función bidireccional de dispersión de la luz	16
2.3. Reflexión local de la luz	17
2.3.1. La función bidireccional de distribución de la reflectancia	17
2.3.1.1. Propiedades	18
2.3.1.2. Geometría de la BRDF	19
2.3.1.3. Clasificación de las BRDFs	20
2.3.2. Reflectancia de la superficie	22
2.3.3. Las ecuaciones de Fresnel	23
2.3.3.1. Caso A: materiales dieléctricos	25
2.3.3.2. Caso B: materiales conductores o metales	26
2.4. La ecuación de radiancia	28
3. Modelos de BRDFs	31
3.1. Reflexión especular ideal (1621)	32
3.2. Lambert (1760)	32
3.3. Minnaert (1941)	33
3.4. Torrance-Sparrow (1967)	34

3.5. Beard-Maxwell (1973)	36
3.6. Phong (1975)	39
3.7. Blinn (1977)	40
3.8. Cook-Torrance (1981)	41
3.9. Kajiya (1985)	44
3.10. Poulin-Fournier (1990)	44
3.11. Strauss (1990)	47
3.12. He-Torrance-Sillion-Greenberg (1991)	50
3.13. Ward (1992)	52
3.14. Westin (1992)	54
3.15. Schlick (1993)	55
3.16. Lewis (1993)	57
3.17. Hanrahan (1993)	59
3.18. Oren-Nayar (1994)	61
3.19. Neumann-Neumann (1996)	63
3.20. Lafortune (1997)	65
3.21. Shirley (1997)	66
3.22. Ashikhmin-Shirley (2000)	68
3.23. Granier-Hiedrich (2003)	70
4. Métodos de estimación de la radiancia	73
4.1. El cálculo de la radiancia reflejada	74
4.2. Monte-Carlo para la resolución de integrales	75
4.2.1. Teoría de las probabilidades	75
4.2.2. Caracterización estadística de una distribución aleatoria	77
4.2.3. Integración por Monte-Carlo	78
4.2.4. Estimadores de Monte-Carlo	79
4.2.5. El error en los métodos de Monte-Carlo	79
4.3. Muestreo según una función de densidad	80
4.3.1. Método de inversión	80
4.3.2. Método de muestreo por rechazo	80
4.3.3. Método de Metrópolis	82
4.4. Técnicas de reducción de la varianza	83
4.4.1. Ruleta rusa	83
4.4.2. Ruleta rusa optimizada	84
4.4.3. Muestreo por estratificación	85
4.4.4. Muestreo por importancias	86
4.4.5. Muestreo por importancias múltiple	87
4.4.6. Remuestreo por importancias - SIR Algorithm	88
4.4.7. Muestreo por importancias ponderado	89
4.4.8. Muestreo por importancias estructurado	89
4.4.9. Muestreo por importancias bidireccional	91
4.4.10. Remuestreo del muestreo por importancias	91
4.4.11. Muestreo de funciones mediante <i>Wavelets</i>	92

4.4.12. Muestreo por importancias en dos fases	94
4.5. Métodos de Monte-Carlo para la estimación de la radiancia	96
4.5.1. El algoritmo de <i>Ray-Tracing</i> distribuido	97
4.5.2. El algoritmo de <i>Path-Tracing</i>	97
4.5.3. El algoritmo de <i>Irradiance-Caching</i>	100
4.5.4. El algoritmo de <i>Photon-Mapping</i>	101
4.5.5. Metropolis Light Transport	104
4.5.6. Técnicas de reutilización de caminos	105
4.5.6.1. Múltiples caminos con <i>global lines</i>	105
4.5.7. Algoritmo de <i>Instant Radiosity</i>	106
5. Muestreo directo de la BRDF	107
5.1. Muestreo de los casos ideales	109
5.1.1. BRDF difusa perfecta	109
5.1.2. BRDF especular perfecta	110
5.2. Modelos basados en el lóbulo coseno	110
5.2.1. Muestreo basado en el ángulo de reflexión	110
5.2.2. Muestreo basado en el <i>half-angle</i>	112
5.2.2.1. Muestreo para las distribuciones de <i>microfacets</i>	113
5.2.2.2. Muestreo de la BRDF de Ward	113
5.3. Muestreo de la BRDF de Neumann	114
5.4. Muestreo de la BRDF de Ashikhmin	115
5.5. Muestreo general para BRDFs analíticas y adquiridas	116
5.6. Muestreo compacto de BRDFs adquiridas	119
6. Muestreo eficiente de BRDFs genéricas	123
6.1. Estimador de Monte-Carlo empleando BRDFs genéricas	124
6.1.1. Beneficios del Disco Adaptativo	126
6.2. Muestreo de una BRDF arbitraria	126
6.2.1. Subdivisión adaptativa del dominio	127
6.2.2. Muestreo por importancias en \mathcal{D}^2	131
6.2.3. Generación de direcciones	132
6.2.4. Comparación directa con otros métodos	133
6.3. Optimización del método	136
6.3.1. Precálculo de varios <i>quadtrees</i>	136
6.3.2. Generación múltiple de direcciones	137
6.3.3. Evaluación eficiente de la BRDF	138
6.3.4. Ajuste de la región para el muestreo por rechazo	139
6.4. Evaluación del método y resultados	142
6.4.1. Evaluación de la convergencia	142
6.4.2. Descripción del procedimiento de evaluación	142
6.4.3. Coste computacional del preprocesamiento asociado al método adaptativo	144
6.4.4. Requisitos de memoria asociados al método adaptativo	145

ÍNDICE

6.4.5.	Escena 1: <i>The Cornell Box</i>	147
6.4.6.	Escena 2: Esfera con alto brillo especular	148
6.4.7.	Escena 3: El dragón de Stanford	156
6.4.8.	Escena 4: Las tres jarras	157
6.5.	Muestreo del producto de la BRDF con la radiancia incidente	161
6.5.1.	Escena 6: Aplicación del disco adaptivo a los métodos de remuestreo para producto de funciones $f_r \cdot L_i$	161
6.6.	Muestreo de BRDFs procedentes de datos adquiridos	166
7.	Utilidades software desarrolladas	171
7.1.	<i>Wannabe Amazing</i>	171
7.2.	Visor de GRF	176
7.3.	Editor de BRDFs	177
7.4.	Muestreo por rechazo de la BRDF	180
7.5.	<i>Basic Path Tracer</i>	182
7.6.	<i>Mini Bpt</i>	186
7.7.	Generador gráfico de ficheros OP	187
7.8.	<i>GLSL BRDF Viewer</i> : evaluación de la BRDF en la GPU	189
7.9.	<i>Adaptive BRDF Subdivision</i> : Visor de la estructura jerárquica	192
7.10.	<i>BRDF 3D Plot</i> : Visualiza nubes de puntos	193
7.11.	<i>EM Viewer</i> : Muestreo del Environment Map	195
	Conclusiones y trabajos futuros	197
A.	El formato de escenas GRF	201
A.1.	Descripción general del formato	201
A.1.1.	Lista de BRDFs y sus parámetros expresada en BNF	205
A.1.2.	Lista de PDFs y sus parámetros en BNF	206
A.1.3.	Un ejemplo sencillo	206
A.1.4.	Un segundo ejemplo	208
B.	Escenas GRF renderizadas	209
	Referencias bibliográficas	223

Índice de abreviaturas, siglas y términos

- S Superficie estudiada.
- A Área total de la superficie S .
- dA Diferencial de área de la superficie.
- dA_p Diferencial de área proyectada con respecto a la dirección normal de la superficie.
- x Punto sobre la superficie, $x \in S$.
- \mathbf{w} Dirección de un rayo, expresado como vector en \mathbb{R}^3 en un sistema local a la normal.
- Ω Conjunto de direcciones de la semiesfera positiva, tomado en un punto de la superficie.
- \mathcal{U} Conjunto de direcciones de la semiesfera negativa, tomado en un punto de la superficie.
- \mathcal{S}^2 Conjunto de direcciones de la esfera centrada en un punto y con radio unidad.
- \mathcal{D}^2 Dominio del plano que divide a \mathcal{S}^2 en hemisferio positivo y negativo de direcciones.
- D Dominio genérico de integración.
- (θ, ϕ) Dirección de un rayo, expresado en coordenadas esféricas en un sistema local a la normal.
- $d\sigma$ Es la medida de ángulo sólido para un vector en la esfera.
- $d\sigma_p$ Es la medida de ángulo sólido proyectado en la esfera.
- P^* Función de densidad probabilística para la medida σ
- Q^* Función de densidad probabilística para la medida σ_p

ÍNDICE

- A_R Área de la zona de rechazo en el muestreo de una función.
- i Subíndice que indica incidencia.
- r Subíndice que indica reflexión.
- t Subíndice que indica transmisión.
- a Subíndice que indica absorción.
- d Subíndice que indica reflexión difusa.
- s Subíndice que indica reflexión especular.
- Φ Energía radiante total (w)
- $d\Phi$ Densidad del área del flujo radiante
- $E(x)$ Irradiancia o radiancia incidente.
- $M(x)$ Exitancia o radiancia saliente.
- $L(x, \mathbf{w})$ Función de radiancia ($w/sr\ m^2$).
- ρ Función de reflectividad semiesférica direccional o albedo de la superficie.
- ρ_d Función de reflectividad lambertiana.
- F Factor de reflexión de Fresnel.
- T Factor de transmitancia de un material.
- f_r Función bidireccional de distribución de la reflectancia – BRDF.
- f_s Función de distribución bidireccional de la dispersión de la luz – BSDF.
- f_{ss} Función bidireccional de distribución de la reflectancia en subcapas de la superficie – BSSRDF.
- f_t Función bidireccional de distribución de la transmisión – BTDF.
- λ Longitud de onda de la luz.
- ν Frecuencia de onda de la luz.
- η Índice de refracción del material.
- κ Coeficiente de extinción del material.
- σ_a Coeficiente de absorción del material.
- σ_s Coeficiente de reflexión especular del material.

-
- σ_t Coeficiente de transmisión del material.
 - k_a Coeficiente de reflectividad ambiental.
 - k_s Coeficiente de reflectividad especular.
 - k_d Coeficiente de reflectividad difusa.
 - \mathbf{n} Vector unitario normal a la superficie.
 - \mathbf{t} Vector unitario tangente a la superficie.
 - \mathbf{r} Vector unitario de reflexión perfecta.
 - \mathbf{h} Vector unitario *halfway*.
 - \mathbf{u} Vector unitario que indica dirección de salida.
 - \mathbf{v} Vector unitario que indica dirección de entrada.
 - \mathbf{u}_p Vector unitario proyectado de $\mathbf{u} \in \Omega$, perteneciente al dominio \mathcal{D}^2 .
 - \mathbf{s} Vector unitario generado de forma aleatoria siguiendo una distribución de probabilidades.
 - α Ángulo formado por los vectores \mathbf{u} y \mathbf{r} .
 - β Ángulo formado por los vectores \mathbf{u} y \mathbf{h} (igualmente \mathbf{v} y \mathbf{h}).
 - δ Ángulo formado por los vectores \mathbf{n} y \mathbf{h} .
 - n Exponente que depende del brillo de la superficie.
 - σ_m Desviación típica de una función de distribución.
 - P Familia de medidas de probabilidad.
 - $P_{\mathbf{u}}$ Medida de probabilidad definida en \mathcal{S}^2 .
 - $p_{\mathbf{u}}$ Función de densidad probabilística definida en $P_{\mathbf{u}}$ con la medida σ .
 - $q_{\mathbf{u}}$ Función de densidad probabilística definida en $P_{\mathbf{u}}$ con la medida σ_p .
 - $w(\mathbf{u})$ Función peso asociada a una dirección en un sistema de Monte-Carlo.
 - I Integral que se desea resolver mediante técnicas de Monte-Carlo.
 - N Número de muestras utilizadas en un estimador de Monte-Carlo.
 - F_N Estimador de Monte-Carlo de la integral I .
 - Q Estimado o valor exacto de la integral I .

ÍNDICE

v.a. Abreviatura de variable aleatoria

X Variable aleatoria definida sobre el espacio $(\mathcal{S}^2, \Omega, P)$.

Y Variable aleatoria definida sobre el espacio $(\mathcal{S}^2, \Omega, P)$.

F_X Función de distribución acumulada de X .

$E(F_N)$ Media del estimador de Monte-Carlo.

$V(F_N)$ Variancia del estimador de Monte-Carlo.

$\beta(F_N)$ Sesgo del estimador de Monte-Carlo.

$\epsilon(F_N)$ Eficiencia del estimador de Monte-Carlo.

U Distribución uniforme que genera valores aleatorios en el intervalo $[0, 1]$ con igual probabilidad.

ξ Variable aleatoria que sigue la distribución uniforme.

h Función de mapeo del dominio Ω a \mathcal{S}^2

h^{-1} Función de mapeo inverso del dominio \mathcal{S}^2 a Ω

$t(x, \mathbf{w})$ Función de *ray casting* que dado un rayo devuelve el punto de intersección más próximo.

Índice de figuras

2.1. Espacio de direcciones unidad \mathcal{S}^2	8
2.2. Sistema de coordenadas empleado.	9
2.3. Mapeo entre direcciones de \mathcal{D}^2 a Ω	10
2.4. Ejemplo de área proyectada	10
2.5. Eventos de reflexión y transmisión de la luz.	13
2.6. Ley de reflexión de un vector.	14
2.7. Casos de la refracción.	15
2.8. Ley de Snell.	16
2.9. Sistema de coordenadas local a la superficie.	19
2.10. Geometría de la reflexión de la luz.	20
2.11. Componentes de la reflexión.	22
2.12. Casos de la reflectancia	23
2.13. La luz como onda.	24
2.14. La luz como onda polarizada.	24
2.15. Geometría de las superficies para el cálculo de la radiancia.	29
3.1. Una instancia de la BRDF de Minnaert en la GPU	34
3.2. Modelo de cavidades o <i>microfacets</i>	35
3.3. Efectos de bloqueo y ocultación de la luz provocado por los <i>microfacets</i>	38
3.4. Una instancia de la BRDF de Beard-Maxwell en la GPU	38
3.5. Una instancia de la BRDF de Phong en la GPU	40
3.6. Una instancia de la BRDF de Blinn en la GPU	41
3.7. Una instancia de la BRDF de Cook-Torrance en la GPU	43
3.8. Modelo anisótropo de Poulin-Fournier	45
3.9. Una instancia de la BRDF de Poulin-Fournier en la GPU	47
3.10. Una instancia de la BRDF de Strauss en la GPU	49
3.11. Modelo óptico de He-Torrance.	50
3.12. Una instancia de la BRDF de He-Torrance en la GPU	52
3.13. Una instancia de la BRDF de Ward en la GPU	54
3.14. Una instancia de la BRDF de Schlick en la GPU	58
3.15. Una instancia de la BRDF de Lewis en la GPU	59
3.16. Modelo de sustratos de Hanrahan.	60

ÍNDICE DE FIGURAS

3.17. Una instancia de la BRDF de oren-nayar en la GPU	63
3.18. Dominio del plano empelado por Neumann-Neumann.	64
3.19. Una instancia de la BRDF de Lafortune en la GPU	66
3.20. Una instancia de la BRDF de coupled en la GPU	67
3.21. Geometría del modelo anisotrópico de Ashikhmin-Shirley.	68
3.22. Modelado de la superficie mediante capas.	69
3.23. Una instancia de la BRDF de ashikhmin en la GPU	70
4.1. Muestreo por inversión.	81
4.2. Muestreo por rechazo.	82
4.3. Estrategias de muestreo en el plano.	86
4.4. Caso de dos rebotes en <i>path-tracing</i>	98
5.1. Muestreo de la BRDF de Ashikhmin.	115
5.2. Casos de la implementación de Lawrence. FIT (4)	118
5.3. Casos de la implementación de CascadeCDF. FIT (1)	121
5.4. Casos de la implementación de CascadeCDF. FIT (2)	121
6.1. Muestreo por subdivisión uniforme del dominio.	126
6.2. Dominio de muestreo en el plano XY.	128
6.3. Dominio del <i>quadtree</i>	128
6.4. Selección de la región más importante.	131
6.5. Distribución de puntos dada por el método uniforme	134
6.6. Distribución de puntos dada por el método del lóbulo coseno	134
6.7. Distribución de puntos dada por el método factorizado	135
6.8. Distribución de puntos dada por el método adaptativo	135
6.9. Generación multiple de direcciones	137
6.10. Ajuste del area de rechazo	140
6.11. Codigos asociados a regiones	141
6.12. Gráfica de tiempo vs error para la Escena 1.	148
6.13. Escena 1 y 2 renderizadas por BPT.	150
6.14. Gráfica de tiempo vs error para la Escena 2 - <i>Sphere</i> 1.	154
6.15. Gráfica de tiempo vs error para la Escena 2 - <i>Sphere</i> 2.	155
6.16. Escena 3: Dragón de Stanford.	156
6.17. Escena 4: Tres jarras usando PDF: Uniforme y Lóbulo coseno.	157
6.18. Escena 4: Tres jarras. PDF: Factorización BRDF y Adaptativa.	158
6.19. Escena 5: Comparativa de nuestro método con el muestreo uniforme.	158
6.20. Gráfica de tiempo vs error para la Escena 4.	160
6.21. Gráfica de tiempo vs error para la Escena 6.	165
6.22. Muestreo de las BRDF adquiridas de MERL.	169
7.1. <i>Wannabe Amazing</i> herramienta para IG.	174
7.2. <i>Wannabe Amazing</i> conversor de formatos.	175
7.3. <i>Wannabe Amazing</i> editor de geometría.	176
7.4. Visor de GRF, versiones multiplataforma y bajo Linux.	177

7.5. Editor de BRDFs: gráficas cartesianas en 2D.	178
7.6. Editor de BRDFs: gráficas polares en 2D.	179
7.7. Editor de BRDFs: gráficas 3D.	180
7.8. Editor de BRDFs y visor de PDFs.	181
7.9. Muestreo por rechazo de la BRDF	181
7.10. MINIBPT combina el software GRF y BPT	186
7.11. Generador de ficheros OP (1).	187
7.12. Generador de ficheros OP (2)	188
7.13. Editor de BRDFs en la GPU (1)	190
7.14. Editor de BRDFs en la GPU (2)	190
7.15. A BRDF viewer on the GPU (3)	191
7.16. Visor de la estructura jerárquica (1)	192
7.17. Visor de la estructura jerárquica (2)	193
7.18. Visor 3D y 2D de ficheros FIT con el método adaptativo	194
7.19. Utilidad para el muestreo del EM	196
B.1. Imagen sintética Escena CORNELL BOX.	209
B.2. Imagen sintética Escena PLANTA.	210
B.3. Imagen sintética Escena LIVING ROOM 1.	210
B.4. Detalle Escena LIVING ROOM 1.	211
B.5. Imagen sintética Escena SIMPLESCENE.	211
B.6. Imagen sintética Escena BAGUESCENE.	212
B.7. Imagen sintética Escena BUZÓN.	212
B.8. Escena MULTISPHERE APLICANDO <i>resampling</i>	213
B.9. Escena MULTISPHERE CON BRDF ADQUIRIDAS.	213
B.10. Imagen sintética Escena HELICÓPTERO.	214
B.11. Imagen sintética Escena MOTORBIKE.	214
B.12. Imagen sintética Escena CHEVROLET.	215
B.13. Imagen sintética Escena PLANESCENE.	215
B.14. Imagen sintética Escena HAPPY BUDHA.	216
B.15. Imagen sintética Escena LIVING ROOM 2.	216
B.16. Imagen sintética Escena DRAGÓN.	217
B.17. Imagen sintética Escena ROADWAY.	218
B.18. Imagen sintética Escena WOODMAN.	218
B.19. Imagen sintética Escena PHLEGMATIC DRAGON.	219
B.20. Escena PHLEGMATIC DRAGON y PLANESCENE.	219
B.21. Imagen sintética Escena FERTILITY.	220
B.22. Imagen sintética Escena FERTILITY.	220
B.23. Imagen sintética Escena BUNNY.	221
B.24. Imagen sintética Escena PYRAMOB.	221
B.25. Imagen sintética Escena CUBE.	222

Índice de tablas y notación GRF

2.1. Índices de refracción de una variedad de objetos.	26
2.2. Propiedades de los materiales conductores.	27
3.1. Notación en GRF para PERFECTSPECULARBRDF	32
3.2. Notación en GRF para PERFECTDIFFUSEBRDF	33
3.3. Notación en GRF para MINNAERTBRDFs	33
3.4. Notación en GRF para BEARDMAXWELLBRDF	38
3.5. Notación en GRF para PHONGBRDF	40
3.6. Notación en GRF para BLINNBRDF	41
3.7. Notación en GRF para TORRANCEBRDF	44
3.8. Notación en GRF para CILINDRICO-PF-BRDF	46
3.9. Notación en GRF para STRAUSSBRDF	50
3.10. Notación en GRF para HTSG-BRDF	51
3.11. Notación en GRF para WARDBRDF	53
3.12. Notación en GRF para SCHLICKBRDF	57
3.13. Notación en GRF para LEWISBRDF	59
3.14. Notación en GRF para OREN-NAYAR-BRDF	63
3.15. Notación en GRF para LAFORTUNEBRDF	66
3.16. Notación en GRF para SHIRLEYBRDF	67
3.17. Notación en GRF para ASHIKHMINBRDF	70
5.1. Datos abreviados de los modelos de BRDFs comentados.	108
5.2. Notación en GRF para UNIFORMPDF	110
5.3. Notación en GRF para SPECULARPDF	110
5.4. Notación en GRF para HEMISPHEREPDF	111
5.5. Notación en GRF para SPHEREPDF	112
5.6. Notación en GRF para ANISOTROPICPDF	116
5.7. Notación en GRF para FACTOREDPDF	118
5.8. Notación en GRF para CASCADECDF	120
5.9. Memoria para las CDFs uniformes procedentes de BRDFs adquiridas .	122
5.10. Memoria para las CDFs compactas procedentes de BRDFs adquiridas .	122
6.1. Notación en GRF para ADAPTIVEPDF	136

ÍNDICE DE TABLAS Y NOTACIÓN GRF

6.2. BRDFs empleadas en las comparativas	143
6.3. PDFs empleadas en las comparativas	144
6.4. Coste en segundos de la fase de precálculo para nuestro método y el de Lawrence.	145
6.5. Coste en KB de la fase de precálculo para nuestro método al modificar el parámetro <code>nmax</code>	146
6.6. Coste en KB de la fase de precálculo para nuestro método y el de Lawrence.	147
6.7. Datos para Escena1: tiempo de muestreo	149
6.8. Datos para Escena1: error relativo medio	149
6.9. Datos para Escena2: Tiempos de muestreo para la PDF adaptativa. . .	151
6.10. Datos para Escena2: Error relativo para la PDF adaptativa.	151
6.11. Tiempos medios para Escena2 - <i>Sphere</i> 1.	153
6.12. Error relativo medio para Escena 2 - <i>Sphere</i> 1.	153
6.13. Tiempos medios para Escena2 - <i>Sphere</i> 2.	153
6.14. Error relativo medio para Escena 2 - <i>Sphere</i> 2.	155
6.15. Tiempos medios de precálculo para Escena 4.	159
6.16. Tiempos medios de muestreo para Escena 4.	159
6.17. Error relativo medio para Escena 4 con ajuste d	159
6.18. Parametros de las BRDFs.	163
6.19. Factorización de la BRDF y sus parámetros	163
6.20. Tiempos medios precálculo para Escena 6.	164
6.21. Tiempos medios para Escena 6.	164
6.22. Error relativo medio para la Escena 6	164
6.23. Datos Escena 7: tiempos de precálculo en escenas con BRDF adquiridas.	167
6.24. Datos Escena 7: tiempos de muestreo en escenas con BRDF adquiridas.	168
7.1. Comparación de los tamaños para originales y el formato de salida GRF.	173
7.2. Un fichero de opciones para BPT.	185

CAPÍTULO 1

Introducción

1.1. La visualización realista

Los detalles visuales del mundo real son ricos y variados, pero perdemos la costumbre de observar lo pequeño. La **visualización realista** es un campo de la Informática Gráfica que intenta cubrir la necesidad de realismo en este área. Para ello, se plantea el objetivo de generar imágenes realistas, que posean la mayoría de los efectos perceptibles que produce la luz cuando interactúa con los objetos físicos reales. Objetivo que constituye una gran empresa, ya que imitar la naturaleza nunca ha sido una tarea sencilla. Varios son los aspectos que se han de tratar:

- construcción de modelos geométricos,
- eliminación de partes ocultas,
- aplicación de texturas y transparencias,
- iluminación y técnicas de sombreado,
- implementación de técnicas por *hardware*: visualización en tiempo real.

Cada una de estas tareas tiene suficiente potencial como para marcar las líneas de trabajo de muchas personas que se han dedicado en profundidad a esta investigación. No es nuestra intención tratar de explicar aquí en qué consiste cada uno de ellas; sólo queremos que el lector observe que trabajar en visualización realista implica muchos temas —todos ellos relacionados— por lo que es mejor concentrarse en un aspecto concreto que intentar abarcarlos todos.

El desarrollo de los trabajos en síntesis de imágenes realistas exige la resolución de un número determinado de problemas diferentes. Por un lado, es necesario la

construcción del modelo geométrico optimizado en lo posible y, por otro, el cálculo de la iluminación, sombreado o *shading*, o término empleado para designar la intensidad y color aplicable a cada punto de la superficie. Centrándonos en éste último, vemos que puede enfocarse desde varias perspectivas según las suposiciones realizadas sobre la escena. Existen dos grandes familias de algoritmos de iluminación conocidos con las siglas MIL —*métodos de iluminación local*— y MIG —*métodos de iluminación global*—. La diferencia entre ambos radica en la consideración, o no, de las interreflexiones entre los objetos.

Dado un modelo de la realidad, es necesario elaborar uno numérico que sea tratado como una descripción objetiva de la escena. Este modelo abarcará principalmente el fenómeno del transporte de la luz y la interacción de ésta con los elementos que forman la escena. La representación se complementa con una descripción subjetiva que realiza el usuario cuando observa la escena, por lo que es importante conocer el funcionamiento del sistema visual humano que realiza dicha interpretación [Gla94, Bue01].

Al tratar principalmente las técnicas de iluminación y sombreado, cabe preguntarse por el aspecto de los objetos presentes en la escena. El aspecto de la superficie, es decir, la intensidad y color de luz con la que se observa un objeto dependerá de:

- las características de la superficie: color, grado de transparencia, reflectancia, etc. Se trata en definitiva de las propiedades del material.
- la posición del observador en la escena.
- las condiciones de iluminación, intensidad, geometría y posición de la fuentes de luz.
- las condiciones ambientales y atmosféricas (niebla, humo, etc.).

El núcleo de los cálculos por ordenador de sombreado se centra en la simulación de la compleja interacción de la luz con los objetos. Compleja en el sentido de que en la realidad se producen millones de interreflexiones, difíciles de abordar computacionalmente en su totalidad. Para simplificar este fenómeno, se suelen utilizar casos ideales más sencillos.

1.2. La naturaleza dual de la luz

La existencia de la luz es un hecho natural que siempre ha interesado al hombre, y como respuesta a este interés ha intentado descubrir su naturaleza. Para explicar la esencia de la luz se han ido proponiendo modelos, que han evolucionado en el curso de la historia. Inicialmente los científicos se debatían entre dos tendencias:

- los que, con Newton, consideraban que la luz estaba formada por corpúsculos;

- los que, siguiendo a Huygens, le asignaban a la luz una naturaleza ondulatoria.

Hasta muy recientemente —principios del siglo XX— no se ha encontrado una explicación satisfactoria acerca de la naturaleza del fenómeno luminoso, conocido como modelo **onda-corpúsculo**.

La primera teoría para explicar la naturaleza de la luz se debe a Newton, que en 1671 revitalizó las ideas sostenidas por Empédocles (siglo V a. de C.). Esta teoría —teoría de la emisión— está basada en la hipótesis de que la luz posee naturaleza corpuscular. En ella se supone que los cuerpos luminosos emiten pequeñas partículas, corpúsculos, que se propagan en línea recta si no están sometidos a ninguna fuerza. Esta teoría explica bien algunas propiedades de la luz, como la propagación rectilínea, la absorción, la reflexión e incluso la refracción.

Muy pocos años después, en 1678, Huygens emitió la hipótesis de que la luz posee carácter ondulatorio y demostró que las leyes de la reflexión y de la refracción se podían explicar con tal hipótesis. Una objeción notable a esta teoría consistió en que si la luz era efectivamente un movimiento ondulatorio debería *doblarse al llegar a las esquinas*, puesto que las ondas se doblan cuando encuentran obstáculos en su recorrido. Sabemos que este fenómeno ocurre realmente, que se llama *difracción*, y que si no se aprecia de esa forma tan notable es debido a la pequeña longitud de onda de la luz.

Posteriormente los trabajos de Thomas Young (1773-1839) pusieron de manifiesto fenómenos de interferencia que eran inexplicables según la teoría corpuscular, con lo que la teoría de Huygens cobró actualidad.

A principios del siglo XIX, Fresnel demostró la naturaleza ondulatoria de la luz a través de una serie de experimentos sobre interferencia, difracción y polarización. Como sólo se conocían vibraciones mecánicas, Fresnel admitió que las vibraciones luminosas correspondían a un fluido hipotético que llamó éter.

En 1865, y gracias a Maxwell, se descubrió cómo la perturbación de un campo electromagnético puede propagarse en el vacío y dio a tales ondas el nombre de *ondas electromagnéticas*, eliminando el hipotético éter. Además determinó a partir de medidas eléctricas y magnéticas la velocidad de la luz. Advierte en la luz las propiedades de frecuencia, fase, amplitud y longitud de onda. Esta última es la distancia entre dos picos sucesivos de una onda. Es bien conocido que la luz que percibe el ojo humano —la luz visible— comprende longitudes desde 400 a 700 nm. Longitudes de onda superiores a 700 nm no son visibles para el ojo y corresponden a la llamada luz infrarroja. Por debajo de 400 nm tenemos la radiación ultravioleta (100-400 nm). Después de esto resultaba evidente que la luz era una onda de naturaleza electromagnética a la que le correspondía una longitud de onda pequeña. Esta teoría se vió apoyada por las ondas ultracortas encontradas por Hertz.

Posteriormente llegó la hipótesis cuántica de Plank (1900), que intentaba interpretar teóricamente las curvas de radiación: los cuerpos calientes absorben y después emiten energía radiante en forma de ondas electromagnéticas de todas las frecuencias, que sólo corresponden al espectro visible si el cuerpo alcanza una cierta temperatura, como es el caso del filamento de una bombilla. La cantidad de radiación que emiten para cada frecuencia sólo depende de la temperatura. La relación intensidad-frecuencia para una determinada temperatura es mostrada en la gráfica o curva de radiación. Plank postuló que los intercambios de energía entre materia y radiación cumplen una ley limitada, según la cual, a cada proceso elemental de absorción o de emisión de una radiación de frecuencia ν debe corresponder un múltiplo entero de cuanto de energía.

$$E = h\nu$$

siendo h la constante de Planck, cuyo valor es $6,6 \cdot 10^{-34} \text{ J} \cdot \text{s}$ en el S.I.

Por primera vez se puso en duda la continuidad de los cambios de energía electromagnética. La energía de la radiación no está distribuida uniformemente en forma de onda electromagnética, sino concentrada en pequeñas partículas, llamadas *fotones*, a los que se les asigna una energía E , igual a la necesaria para la extracción de un electrón más la energía cinética del electrón expulsado. Esto es, los electrones al vibrar emiten energía, pero no de forma continua, sino por paquetes o *cuantos*.

A pesar de sus éxitos, la teoría cuántica no fue bien recibida por los científicos contemporáneos, ni siquiera por el mismo Plank. La razón estaba en el convencimiento de que los procesos de la naturaleza ocurren de forma continua. Sin embargo, según se demostró con el efecto fotoeléctrico, ciertas sustancias —metales principalmente— emiten fotones cuando sobre ellas incide la luz. En 1916 quedan establecidas las leyes experimentales —leyes del efecto fotoeléctrico— aunque ya habían sido previstas en 1905 por Albert Einstein. Las leyes se enumeran como: (1) Es instantáneo, aparece en el mismo momento que el metal es iluminado. (2) El número de electrones emitidos no depende de la intensidad de la luz incidente, sino de la frecuencia de radiación. (3) Por debajo de una frecuencia mínima, que depende de la naturaleza del metal, no se presenta el efecto fotoeléctrico.

Ante la imposibilidad de explicar este fenómeno dentro del marco de la teoría electromagnética de radiación, Einstein en 1905 notó que la luz se comporta de forma distinta cuando interacciona momentáneamente con la materia (cuantificada como fotones), compatible cuando se comporta como una onda en los procesos de propagación de la luz. El fotón se caracteriza por su energía —con la conocida expresión de $E = mc^2$ — y su cantidad de movimiento $p = mc$, siendo m la masa del fotón y c la velocidad de la luz.

Los datos anteriormente señalados manifiestan que la luz se comporta en algunos casos como si estuviera constituida por un flujo de partículas. Sin embargo, la teoría electromagnética, que sirve muy bien para explicar otros fenómenos, no puede admitir

la existencia de fotones. La luz presenta, según esto, un doble aspecto: ondulatorio y corpuscular. Y dependiendo de que se manifieste uno u otro, será aplicable la teoría electromagnética clásica o la de los fotones (corpuscular).

Pero ambas teorías son contradictorias. Al menos ésta fue la situación hasta que en 1924 Louis de Broglie postuló que a cada partícula corresponde una onda asociada de $\lambda = h/p$. A partir de aquí surgió una nueva óptica —la *óptica electrónica*—. De las ideas de De Broglie se ha desarrollado una nueva mecánica, llamada mecánica ondulatoria, que considera los problemas planteados en la mecánica clásica como problemas de propagación de ondas.

Actualmente se admite que la dualidad onda-corpúsculo, que se manifiesta en los electrones, es una propiedad general de la materia. No es posible conocer con exactitud cuál va a ser la historia exacta, o evolución de un fotón, únicamente podemos conocer la probabilidad de que ocurran secuencias concretas de eventos.

1.2.1. Propagación de la luz

Cuando un cuerpo no luminoso recibe luz se *ilumina*. Esto es lo que ocurre al abrir las ventanas o al encender una bombilla. Los cuerpos que se iluminan presentan distintos comportamientos, según sea su naturaleza; así podemos distinguir materiales transparentes, que dejan pasar la luz a través de ellos y permiten ver objetos situados al otro lado; translúcidos, cuando dejan pasar la luz pero no permiten ver otros objetos de detrás; y opacos si impiden el paso de la luz. Cuando la luz se encuentra en su camino un objeto opaco, forma detrás de él una zona de sombra y a veces otra de penumbra, lo que suele ocurrir si la fuente luminosa es grande y está próxima al objeto opaco.

La luz se propaga en todo cuerpo transparente, así como en el vacío, como es el caso de la luz solar que llega a la Tierra, que lo hace con gran rapidez, pero no de forma instantánea. En el vacío lo hace a una velocidad máxima aproximada de 300 000 *km/s*. Ocurre que en otros medios se introduce una variación en la velocidad del haz transmitido. La relación entre las dos velocidades, la del medio y la del vacío, es dada por el índice de refracción η .

1.2.2. Comportamiento de la luz al incidir en una superficie

Cuando la luz que se está propagando en un medio incide en otro medio, puede seguir distintos comportamientos:

- Parte de la luz es reemitida en la superficie de separación y continua dentro del mismo medio de incidencia. Se dice que la luz se refleja.

- Otra parte puede atravesar la superficie de separación, cambiando de dirección en el segundo medio. Entonces, se dice que se refracta.
- Parte de la luz se absorbe, transfiriendo energía al medio, normalmente en forma de calor.

La cantidad de luz reflejada o refractada es una función que depende de λ y del medio. Al interactuar con la superficie se modifica la dirección de propagación, pero no así su longitud de onda. También depende de otros factores como la geometría de la superficie y la luz y el ángulo de incidencia formado con respecto a la normal de la superficie. En *informática gráfica* estos efectos se suman y simulan en un modelo de iluminación (*Shading Model*). Un modelo de iluminación que incluya los distintos efectos que produce la radiación luminosa cuando interactúa con objetos reales considerará, entre otras cosas:

- la forma como la energía se refleja al alcanzar una superficie,
- la refracción, o cómo se dispersa la luz al producirse un cambio en el medio,
- la interacción del medio atmosférico (lluvia, polvo, humo, etc.) con la luz.

1.3. Caracterización de las superficies

La función más relevante que interviene en la reflexión local a un punto es la ***Función Bi-direccional de Distribución de la Reflectancia (BRDF)***, gracias a la cual se describe el comportamiento reflexivo de la superficie del objeto. Si bien debemos definir formalmente todos los términos relacionados con la caracterización de la energía radiante, baste por ahora ver la BRDF como una función de dos direcciones, una hacia la fuente de luz y otra hacia el visor, que determina el aspecto de un material percibido por un observador.

Un modelo de reflexión debe describir el color y la distribución espacial de la luz reflejada. La elección de un modelo de BRDF, así como sus parámetros, constituye una labor de relevancia dentro del proceso general de síntesis de imágenes realistas. Es por ello que debemos esclarecer todo lo relacionado con los modelos de reflexión de *informática gráfica*. Definición, clasificación, estrategias de trabajo, propiedades y utilización en un sistema real de iluminación serán los temas a tratar en sucesivos capítulos.

Esta tesis está dedicada a facilitar el uso de modelos de BRDF idóneos para la adecuada caracterización de una superficie, en un sistema de *visualización realista*, al permitir que se simule la trayectoria de la luz de forma predominante a cómo indique dicha función. Es nuestra intención para el siguiente capítulo realizar una introducción de la terminología utilizada en *iluminación global*, considerando un modelo simplificado del transporte de la luz —obtenido tras multitud de simplificaciones— útil en *informática gráfica*.

CAPÍTULO 2

Modelos de reflexión en *iluminación global*

2.1. Representación de la luz en *iluminación global*

En las técnicas de iluminación intervienen dos elementos básicos: las fuentes productoras de luz y los objetos que se van a iluminar. En la literatura sobre *iluminación global* la mayoría de los autores consideran que la luz o energía radiante está discretizada en pequeños paquetes, que se modelan mediante partículas o fotones con una propiedad y una determinada probabilidad de interacción con la materia. El flujo de energía que recorre un medio es simplemente el número de fotones existentes en una región del medio. Las partículas siguen caminos rectilíneos a lo largo de un medio homogéneo sin interferir entre sí. Sin embargo, sí interaccionan con los objetos que se encuentran a su paso —dentro de la escena que se desea iluminar— momento en el cual, se produce el efecto de dispersión de energía. Cuando la luz alcanza un objeto —concretamente en algún punto de su superficie— sucederá que alguna porción de la misma será reflejada, otra parte transmitida y el resto será absorbida por el objeto. La absorción es ignorada —únicamente contribuye a mantener la temperatura de la superficie estable con respecto al entorno— y sólo la luz reflejada y la transmitida pueden llegar a alcanzar al observador.

El flujo de energía, quedará descrito por una ecuación general de transporte, cuya resolución será la tarea principal de los algoritmos de síntesis realista. El transporte de partículas en un sistema 3D requiere la caracterización de diversos conceptos matemáticos aplicados a nuestro problema.

2.1.1. Notación matemática

Mediante un vector de tres valores reales notaremos puntos en la superficie en donde estudiamos la reflexión local $x \in S$ y direcciones con su respectiva magnitud. En este texto utilizaremos símbolos en negrita para denotar direcciones expresadas en un sistema cartesiano local al punto de la superficie que está siendo analizado. Si las direcciones están normalizadas, podemos considerar a éstas como puntos pertenecientes a una esfera de radio unidad. Esto es, si $\mathbf{u} \in \mathcal{S}^2$ (donde $\mathcal{S}^2 \in E^3$ es la esfera unidad con origen en E^3), entonces $\mathbf{u} = (x_{\mathbf{u}}, y_{\mathbf{u}}, z_{\mathbf{u}})$ y, además, $|\mathbf{u}| = 1$. En este sistema el eje Z local coincide con la normal al punto, y se tiene que $\mathbf{n} \stackrel{\text{def}}{=} (0, 0, 1)$ es el vector de longitud normal paralelo a dicho eje Z.

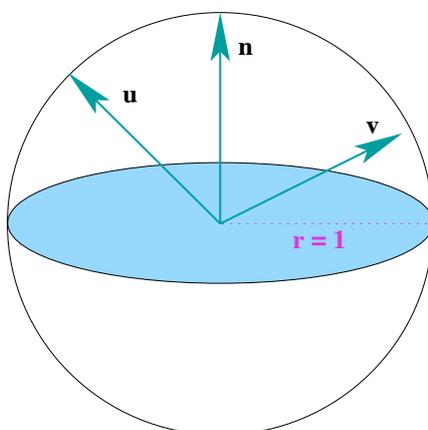


Figura 2.1: Espacio de direcciones unidad \mathcal{S}^2 . El plano XY divide a la esfera en dos hemisferios de direcciones, uno positivo Ω y otro negativo \mathcal{U} .

La semiesfera positiva de radio unidad $\Omega \subset \mathcal{S}^2$ (fig. 2.1), cuyo eje es el vector \mathbf{n} , se define como:

$$\Omega \stackrel{\text{def}}{=} \{ \mathbf{v} \in \mathcal{S}^2 \mid \mathbf{v} \cdot \mathbf{n} > 0 \}.$$

Su complemento se refiere a la semiesfera de direcciones negativas, esto es, $\Omega^- \subset \mathcal{S}^2$, y cumple:

$$\mathcal{U} \stackrel{\text{def}}{=} \{ \mathbf{v} \in \mathcal{S}^2 \mid \mathbf{v} \cdot \mathbf{n} \leq 0 \}.$$

Los vectores de dirección normalizados pueden expresarse igualmente en coordenadas polares mediante un par de ángulos. Para un vector \mathbf{u} , el ángulo polar $\theta_{\mathbf{u}}$ mide el ángulo con el eje Z y el azimutal $\phi_{\mathbf{u}}$, considera la proyección en el plano XY con respecto al eje X. De tal manera puede verse en la figura 2.2. Así un vector se escribe $\mathbf{u} = (\theta_{\mathbf{u}}, \phi_{\mathbf{u}})$. Las coordenadas cartesianas $(x_{\mathbf{u}}, y_{\mathbf{u}}, z_{\mathbf{u}})$ de \mathbf{u} pueden obtenerse a partir de $(\theta_{\mathbf{u}}, \phi_{\mathbf{u}})$ usando la siguiente relación:

2.1. Representación de la luz en iluminación global

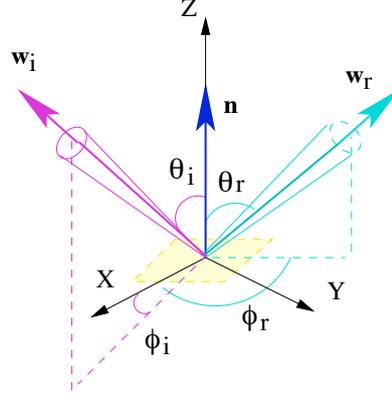


Figura 2.2: Ángulos formados por los vectores incidente y reflejado, medidos en un sistema de referencia con respecto a la normal del punto.

$$\begin{aligned}x_{\mathbf{u}} &= \cos(\phi_{\mathbf{u}}) * \sin(\theta_{\mathbf{u}}), \\y_{\mathbf{u}} &= \sin(\phi_{\mathbf{u}}) * \sin(\theta_{\mathbf{u}}), \\z_{\mathbf{u}} &= \cos(\theta_{\mathbf{u}}).\end{aligned}$$

Junto a la polar y la cartesiana existe una tercera parametrización para las direcciones $\mathbf{v} \in \mathcal{S}^2$. Consiste en representar la dirección como un punto 2D, correspondiente a la proyección del vector —que se realiza en un sistema de referencia local a un punto—, simplemente descartando la coordenada $z_{\mathbf{v}}$, lo cual nos deja $\mathbf{v}_{xy} \in \mathcal{D}^2$. El conjunto de todas las proyecciones conforman el dominio \mathcal{D}^2 :

$$\mathcal{D}^2 \stackrel{\text{def}}{=} \{ (x, y) \in E^2 \text{ t.q. } x^2 + y^2 < 1 \}.$$

Una función de mapeo — $h : \Omega \rightarrow \mathcal{D}^2$ — fácilmente invertible se utiliza para realizar paso de un dominio a otro:

$$h(\mathbf{v}) = h(x_{\mathbf{v}}, y_{\mathbf{v}}, z_{\mathbf{v}}) \stackrel{\text{def}}{=} (x_{\mathbf{v}}, y_{\mathbf{v}}) = \mathbf{v}_{xy}.$$

La conversión inversa usando h^{-1} de un punto incluido en el círculo unidad embebido en un rectángulo $[-1, 1) \times [-1, 1)$ a un vector en la semiesfera, puede verse en la figura 2.3.

$$\mathbf{v} = h^{-1}(\mathbf{v}_{xy}) = h^{-1}(x_{\mathbf{v}}, y_{\mathbf{v}}) = (x_{\mathbf{v}}, y_{\mathbf{v}}, z_{\mathbf{v}}) \mid z_{\mathbf{v}} = \sqrt{1 - x_{\mathbf{v}}^2 - y_{\mathbf{v}}^2}.$$

Nos interesa conocer las áreas proyectadas para describir qué porción de una superficie es visible desde una dirección. Cuando una superficie es vista desde un

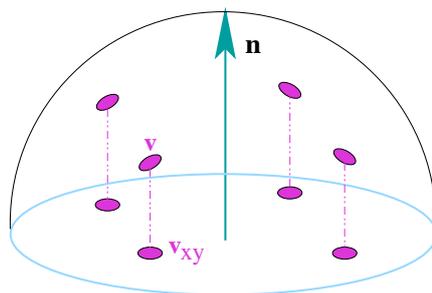


Figura 2.3: Proyección inversa: del punto en la superficie base al vector de dirección en Ω .

determinado ángulo, su área es disminuida por el factor coseno del ángulo que forman la dirección de vista y la normal de la superficie¹. Así en la figura 2.4 el área de la superficie A_1 que recibe luz es A , y para la segunda superficie, $A_2 = A / \cos(\theta)$.

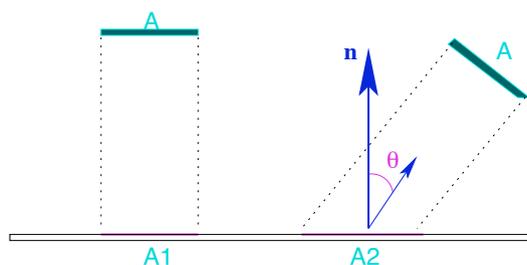


Figura 2.4: Área proyectada aplicada a la Ley del coseno.

El concepto de ángulo en 2D —longitud de la circunferencia con respecto al radio— al extenderse a 3D se denomina *ángulo sólido*². Representa la generalización del vector de dirección para incluir un rango de direcciones y equivale —en la esfera \mathcal{S}^2 — al diferencial de superficie que representan dichas direcciones. Aplicada a una dirección cualquiera, $\mathbf{v} \in \Omega$ —medida notada con la letra σ y con unidad esteroradián— se calcula como:

$$d\sigma(\mathbf{v}) \stackrel{\text{def}}{=} \frac{dA(\theta_{\mathbf{v}}, \phi_{\mathbf{v}})}{r^2} = \frac{(r d\theta_{\mathbf{v}})(r \sin(\theta_{\mathbf{v}}) d(\phi_{\mathbf{v}}))}{r^2} = \sin(\theta_{\mathbf{v}}) d(\theta_{\mathbf{v}}) d(\phi_{\mathbf{v}}). \quad (2.1)$$

y resulta independiente del radio de la esfera considerada, que es r .

¹ La ley del coseno del ángulo afirma que la iluminación es proporcional al coseno del ángulo de incidencia.

² Matemáticamente, el *ángulo sólido* es parte del espacio comprendida dentro de la cavidad de cierta superficie cónica (con generatriz cerrada).

2.1. Representación de la luz en iluminación global

El producto escalar de dos vectores —término coseno— podrá expresarse utilizando notación de vectores, al igual que la de ángulos:

$$(\mathbf{v} \cdot \mathbf{n}) = \cos(\mathbf{v}) = \cos(\theta_{\mathbf{v}}).$$

Una medida de interés en Ω es el ángulo sólido proyectado σ_p , definido para cualquier vector $\mathbf{v} \in \Omega$:

$$d\sigma_p(\mathbf{v}) \stackrel{\text{def}}{=} (\mathbf{v} \cdot \mathbf{n}) d\sigma(\mathbf{v}). \quad (2.2)$$

Para todo par de puntos $(x, y) \in \mathcal{D}^2$ se utiliza la medida de área $dA(x, y)$, que puede relacionarse fácilmente con las otras unidades de medida de la esfera. Significa que regiones diferenciales en Ω pueden proyectarse en regiones diferenciales en \mathcal{D}^2 , e igualmente con la inversa:

$$dA(\mathbf{u}_{xy}) = dA(x_{\mathbf{u}}, y_{\mathbf{u}}) = dx dy = d\sigma_p(h^{-1}(x_{\mathbf{u}}, y_{\mathbf{u}})),$$

$$d\sigma_p(\mathbf{u}) = dA(h(\mathbf{u})) = dA(\mathbf{u}_{xy}). \quad (2.3)$$

Una vez introducida la notación general continuaremos con las magnitudes y unidades de medida fundamentales empleados para valorar y comparar las cualidades y los efectos de la radiación luminosa.

2.1.2. Densidad del flujo radiante

Se conoce como flujo radiante Φ al número de partículas que fluyen a través de un medio, transportando una cantidad de energía y con una determinada probabilidad al interaccionar con la superficie. En la descripción matemática del comportamiento de la luz se utiliza la medida de densidad de flujo radiante, que considera la potencia recibida, emitida, transmitida o reflejada por la unidad de área de una superficie, referida a radiación de una longitud de onda fija. Las unidades con las que se expresa son $W \cdot m^{-2}$ (vatios por metro cuadrado).

Supongamos que la potencia reflejada que abandona la superficie lo hace desde el mismo punto donde se recibió la potencia que causó la reflexión. Entonces la potencia saliente en una región Φ resulta de la propia emisión del objeto Φ_e y de la porción de energía que es reflejada Φ_r , a partir de un flujo incidente Φ_i . Esto es:

$$\Phi = \Phi_e + \Phi_r.$$

Nótese que podríamos referirnos al flujo radiante transmitido como Φ_t . A su vez, para diferenciar entre el sentido o dirección utilizaremos el término E (*irradiancia*)

CAPÍTULO 2. Modelos de reflexión en iluminación global

para referirnos a la densidad de energía que llega a un punto de la superficie, y M (*exitancia*) al hablar de densidad de energía que abandona la misma:

$$M(x) = \frac{d\Phi(x)}{dA(x)} \quad E(x) = \frac{d\Phi_i(x)}{dA(x)}.$$

La densidad de energía que abandona la superficie $M(x)$ es la conjunción la producida en el evento de reflexión $M_r(x)$ y de la que la superficie es capaz de generar como emisión $M_e(x)$:

$$M(x) = M_e(x) + M_r(x) = \frac{d\Phi_e(x)}{dA(x)} + \frac{d\Phi_r(x)}{dA(x)}.$$

Tanto la irradiancia como la exitancia son funciones que miden la energía en todas las direcciones posibles, de ahí que puedan expresarse como una integral en la semiesfera, como se verá en el próximo apartado.

2.1.3. Radiancia

Podemos definir la radiancia L como la densidad de flujo radiante que abandona por un cono de direcciones un punto y que se propaga en una dirección concreta, por ángulo sólido y por área proyectada en la dirección perpendicular a la normal. La unidad con la que se expresa es $W \cdot m^{-2} \cdot sr^{-1}$:

$$dL_r(x, \mathbf{u}) = \frac{d^2\Phi(x, \mathbf{u})}{dA_p(x) d\sigma(\mathbf{u})} = \frac{d^2\Phi(x, \mathbf{u})}{\cos(\mathbf{u}) dA(x) d\sigma(\mathbf{u})} = \frac{d^2\Phi(x, \mathbf{u})}{dA(x) d\sigma_p(\mathbf{u})}.$$

Corresponde a la forma diferencial de la radiancia, que puede relacionarse fácilmente con la expresión anteriormente dada para la irradiancia:

$$dL_i(x, \mathbf{v}) = \frac{d(d\Phi_i(x, \mathbf{v}))}{dA(x) \cos(\mathbf{v}) d\sigma(\mathbf{v})} = \frac{dE(x, \mathbf{v})}{d\sigma_p(\mathbf{v})},$$

por tanto,

$$dE(x, \mathbf{v}) = dL_i(x, \mathbf{v}) \cos(\mathbf{v}) d\sigma(\mathbf{v}). \quad (2.4)$$

La luz abandona la superficie a través de un punto, y puede ser diferente dependiendo de la dirección de entrada con respecto a la de salida. El ángulo sólido es una fracción de la semiesfera de direcciones. Integrando la radiancia sobre los ángulos sólidos, podemos calcular toda la energía que incide o deja la superficie por unidad de tiempo y área. Con esta idea, las medidas anteriores pueden expresarse en forma de integral:

$$M_r(x) = \int_{\Omega} dL_r(x, \mathbf{u}) d\sigma_p(\mathbf{u}), \quad (2.5)$$

$$E(x) = \int_{\Omega} dL_i(x, \mathbf{v}) d\sigma_p(\mathbf{v}). \quad (2.6)$$

2.2. Dispersión de la luz en la superficie

Cuando un haz de luz que se propaga por un medio ingresa en otro distinto; una parte del haz se refleja mientras que la otra sufre una refracción, consistente en el cambio de dirección del haz (véase la figura 2.5). El proceso de transmisión, reflexión y refracción son manifestaciones macroscópicas de la dispersión de la luz que ocurren a niveles microscópicos.

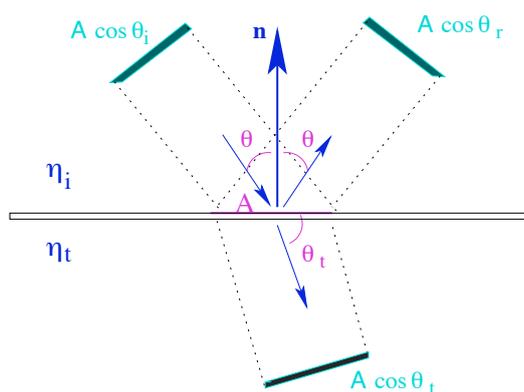


Figura 2.5: Reflexión y transmisión de un haz incidente.

Siempre que un rayo alcanza la interfaz entre la superficie lisa y un medio diferente, con índices refractivos diferentes, se debe calcular la energía que es transferida en esa interfaz, para el área cubierta por el haz de luz incidente. La energía total que fluye hacia el área A por unidad de tiempo debe conservarse y, por tanto, debe ser igual a la que sale:

$$\Phi_i A \cos(\theta_i) = \Phi_r A \cos(\theta_r) + \Phi_t A \cos(\theta_t).$$

2.2.1. Ley de reflexión

La interfaz entre dos medios transparentes presenta una discontinuidad que hace que parte de la luz que llega a esta se disperse hacia atrás en el mismo medio. Si la transición entre los dos medios es gradual se produce poca reflexión. En este fenómeno físico intervienen las moléculas interiores a la superficie, en un grosor $\lambda/2$, que nos muestra una dependencia de la longitud de onda del flujo incidente. Un flujo radiante incidente de mayor λ conseguirá que dispersen moléculas de capas más profundas para la región de contacto. Sin embargo, es sabido que la intensidad de la luz dispersada es proporcional a $1/\lambda$, por lo que se produce finalmente una situación de balance.

Debido a la reflexión podemos ver los objetos que no emiten luz, siempre que estén iluminados. Las superficies que reflejan la luz lo hacen de distinta manera, según estén o no pulimentadas. En las primeras, la luz reflejada sale en una única dirección;

CAPÍTULO 2. Modelos de reflexión en iluminación global

por esta razón se llama reflexión especular ideal, o reflexión dirigida. En las segundas, la luz reflejada sale en todas las direcciones, y por eso se denomina reflexión difusa.

Una superficie pulimentada y opaca se llama espejo. Su vector normal es la línea perpendicular en dicho punto a la superficie del espejo si éste es plano, o a la tangente si es curvo. Las leyes de reflexión [FvDaSFH90], considerando que uno de los medios es el vacío, se enuncian como:

1. El rayo incidente, el rayo reflejado y la normal al espejo están en el mismo plano.
2. Los ángulos de incidencia y reflexión son iguales.

El vector reflejado se calcula dada una dirección incidente (ver figura 2.6), como sigue:

$$\begin{aligned}\theta_r &= \theta_u, \\ \phi_r &= \phi_u \pm \pi, \\ \mathbf{r} &= \mathbf{u} - 2(\mathbf{u} \cdot \mathbf{n})\mathbf{n}.\end{aligned}$$

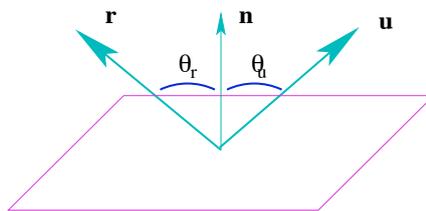


Figura 2.6: Reflexión de un vector con respecto a la normal.

Una superficie perfectamente lisa se representaría mediante el valor de su reflectancia, que en este caso coincidiría con el factor de Fresnel (ver 2.3.3); si la superficie es muy rugosa la luz deja de reflejarse preferentemente en una única dirección.

2.2.2. Ley de refracción

Cuando la luz incide en una superficie que separa dos medios transparentes de distinta densidad, una parte se refleja y otra se refracta en el segundo medio, variando su dirección de propagación. Este último fenómeno se conoce como refracción de la luz.

En 1621, Willebrord Snell, observó que en función de la velocidad de la luz, al penetrar esta en otro medio, sufría una desviación que guardaba relación con el grado de incidencia, y también con el grado de refracción del medio. Pero sería Huygens, que consideraba la luz como fenómeno ondulatorio, quien elaboró índices de refracción, estableciendo que el índice de refracción de un material está determinado por la

2.2. Dispersión de la luz en la superficie

velocidad con que lo traspasa la luz. Dentro del nuevo medio, los rayos se trasladan rectilíneamente hasta encontrar la superficie límite opuesta, actuando nuevamente como rayos incidentes.

Por ejemplo (figura 2.7), los rayos de luz que pasan del aire al agua, disminuyen su velocidad de propagación — $v_1 > v_2$ —, haciéndose 1'3 veces menor. Al incidir en el agua, cambian su dirección, aproximándose a la normal de la superficie que separa ambos medios —aire y agua—. De la misma manera, los rayos procedentes de un foco de luz que está dentro del agua de una piscina se refractan al salir al aire, pero en este caso, como su velocidad aumenta a pasar del uno al otro medio — $v_1 < v_2$ —, al cambiar de dirección se aleja de la normal a la superficie de separación. En general, cuando la velocidad de propagación del segundo medio es mayor que en el primero, el ángulo de refracción es mayor que el de incidencia y viceversa.

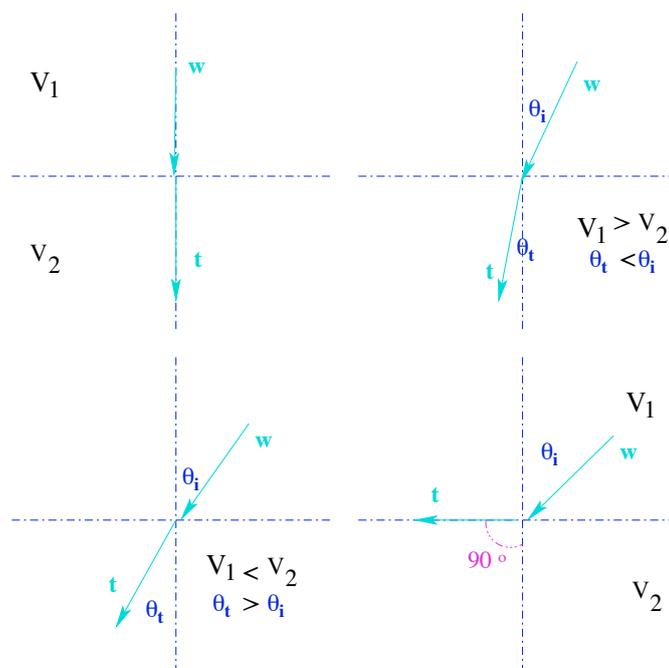


Figura 2.7: Distintos casos en los que se produce refracción.

El ángulo que forma el rayo incidente con la normal a la superficie se denomina ángulo de incidencia, y el que forma el rayo refractado con dicha normal se denomina ángulo de refracción. Para un determinado valor del ángulo de incidencia, el rayo de luz se refracta con un ángulo de refracción de 90° . El ángulo de incidencia al que le corresponde uno de refracción de 90° se llama ángulo límite, y en el medio agua-aire es de 48° . Si el rayo incide con un ángulo mayor que el límite, ya no se puede refractar y toda la luz que llega a la superficie de separación se refleja. Se dice, entonces, que se

produce una reflexión interna total.

Encontrar la relación entre los ángulos de incidencia y refracción es un proceso experimental cuyos datos se conocen desde Ptolomeo, aunque fue descubierto más tarde por Snell, en 1621, y por Descartes, en 1638. En la figura 2.8 puede verse la diferencia entre el ángulo de incidencia y el de refracción del haz —antes y después de ingresar al nuevo material—. La ratio constante x_i/x_t proporciona el valor del índice de refracción del material: η .

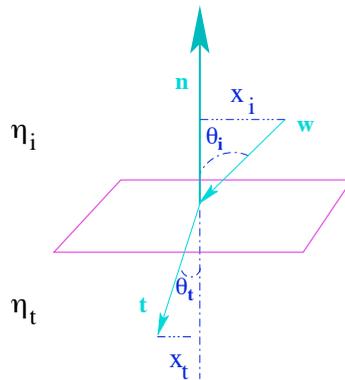


Figura 2.8: Geometría para la transmisión según la Ley de Snell.

Las leyes que rigen el fenómeno de refracción son:

1. El rayo incidente, la normal a la superficie de separación entre el aire y el vidrio, y el rayo refractado, se encuentran en el mismo plano.
2. Para dos medios físicos, el cociente entre el seno del ángulo de incidencia y el seno del ángulo de refracción es constante.

Este último enunciado corresponde a la *ley de Snell*, que se formula como:

$$\begin{aligned}\eta_i \sin \theta_i &= \eta_t \sin \theta_t, \\ \phi_t &= \phi_i \pm \pi.\end{aligned}$$

2.2.3. La función bidireccional de dispersión de la luz

El comportamiento de la luz al incidir sobre los materiales va más allá del fenómeno de reflexión local a un punto. Los objetos no son puramente reflexivos ni transmisivos, y se produce transferencia de energía procedentes de la dispersión de la luz en otras superficies en otros puntos. En este caso, aparece en la formulación la BSDF (f_s) o *Bidirectional Scattering Distribution Function*, como la función matemática que

describe el comportamiento del material frente a la dispersión de la luz. Es la unión de cuatro funciones en una: dos BRDFs para considerar la reflexión en ambos lados de la superficie, y dos BTDFs para la transmisión de la luz en cada dirección.

La *Bidirectional Transmission Distribution Function* (BTDF) es una función real no acotada de valores positivos que tiene como unidad sr^{-1} (*estereoradián a la inversa*). Se define de forma similar con respecto a la BSDF:

$$f_t : \Omega_i \times \Omega_t \longrightarrow \mathbb{R},$$

siendo la semiesfera de transmisión el complemento de la semiesfera incidente, $\Omega_t = \mathcal{U}$. Si bien son más numerosos los autores que únicamente consideran la BRDFs y la reflexión local de la luz, también existen modelos de BTDFs [Sta01].

La función BRDF se define de forma completa en la sección 2.3.1. Por ahora basta por saber que la BRDF es una restricción del dominio de la BSDF a un dominio menor:

$$f_r : \Omega_i \times \Omega_r \longrightarrow \mathbb{R},$$

con $\Omega_i = \Omega_r$ ya que se refieren a un mismo conjunto de direcciones.

2.3. Reflexión local de la luz

2.3.1. La función bidireccional de distribución de la reflectancia

La BRDF, o *Bidirectional Reflectance Distribution Function*, es una representación parametrizada de la reflectancia de una superficie. Constituye un modelo de la interacción de la luz con la superficie, relacionando la radiancia que llega desde una dirección \mathbf{v} , con la que sale desde el punto, en la dirección \mathbf{u} . Visto de otra forma, es la constante de la relación:

$$dL_r(x, \mathbf{u}) \propto dE_i(x, \mathbf{v}).$$

Se denomina *bidireccional* por ser una función de direcciones incidentes y reflejadas. En su forma completa depende de seis parámetros; sin embargo su expresión suele verse simplificada por las siguientes suposiciones:

- Puesto que la distribución de reflectancia es invariante a lo largo de la superficie, y se asume que la reflexión es local a un punto, el término x queda implícito en la formulación.
- La luz se refleja inmediatamente, por lo que no se considera la variable tiempo. No se simula la *fosforescencia*.

CAPÍTULO 2. Modelos de reflexión en iluminación global

- La luz que deja la superficie tiene la misma longitud de onda —y por tanto la misma frecuencia ν — que la que incide; se omite el fenómeno de *fluorescencia*.
- Además de los parámetros anteriores, la BRDF es, junto a la radiancia, dependiente de la longitud de onda λ . Si repetimos los cálculos de la reflectancia para varias longitudes de onda, conseguiremos la reconstrucción del color. Esta dependencia no se suele expresar en la formulación y se considera implícita.

La expresión de la BRDF, notada por $f_r(\mathbf{u}, \mathbf{v})$, nos da la ratio de radiancia observada por un visor en la dirección \mathbf{u} , proporcional a la irradiancia procedente de un ángulo sólido infinitesimal alrededor de \mathbf{v} :

$$f_r(\mathbf{u}, \mathbf{v}) = \frac{dL_r(\mathbf{u})}{dE_i(\mathbf{v})} = \frac{dL_r(\mathbf{u})}{L_i(\mathbf{v}) d\sigma_p(\mathbf{v})}. \quad (2.7)$$

También podemos expresarla utilizando la notación de ángulos:

$$f_r(\theta_r, \phi_r, \theta_i, \phi_i) = \frac{dL_r(\theta_r, \phi_r)}{L_i(\theta_i, \phi_i) \cos(\theta_i) d\sigma(\mathbf{w}_i)}.$$

Se trata de una función real no acotada de valores positivos, que tiene como unidad sr^{-1} (*estereoradián* a la inversa):

$$f_r(\mathbf{u}, \mathbf{v}) \in [0, \infty).$$

2.3.1.1. Propiedades

La BRDF posee una serie de propiedades importantes —basadas en leyes físicas— que enumeramos a continuación:

1. Simetría.

Cumple la *ley de reciprocidad de Helmholtz* [CP85], que dice que si un fotón tiene cierta probabilidad de moverse a lo largo de un camino, conservará la misma probabilidad aunque las direcciones se reviertan. Indica, por tanto, que la BRDF es invariante frente al intercambio de direcciones:

$$f_r(\mathbf{u}, \mathbf{v}) = f_r(\mathbf{v}, \mathbf{u}) \quad \forall \mathbf{u}, \mathbf{v} \in \Omega.$$

2. Conservación de la energía.

La cantidad de luz reflejada no puede superar a la incidente [Hel25], de otra forma se rompería el equilibrio de la ecuación que modela el flujo de energía. Para la conservación de ésta, la siguiente condición debe satisfacerse siempre:

$$\int_{\Omega} f_r(\mathbf{u}, \mathbf{v}) d\sigma_p(\mathbf{v}) < 1 \quad \forall \mathbf{u} \in \Omega.$$

Este hecho está controlado por la reflectividad direccional-hemiesférica de la BRDF.

2.3.1.2. Geometría de la BRDF

Algunos autores describen en sus trabajos la BRDF basándose en la figura 2.2 para definir el proceso en el cual, al incidir la luz procedente de la dirección $(\theta_{\mathbf{w}_i}, \phi_{\mathbf{w}_i})$ con un ángulo sólido $d\sigma(\mathbf{w}_i)$ en un punto de la superficie, la misma se verá reflejada hacia fuera en la dirección de reflexión $(\theta_{\mathbf{w}_r}, \phi_{\mathbf{w}_r})$ con un ángulo sólido $d\sigma(\mathbf{w}_r)$.

En otros trabajos [PF90, KM99] se ha utilizado el vector binormal \mathbf{b} y tangente \mathbf{t} para formar un sistema de coordenadas local —como el de la figura 2.2— apropiado para cálculos de iluminación. Dado un vector arbitrario \mathbf{w} , se deben cumplir las siguientes propiedades geométricas:

$$\begin{aligned} \mathbf{b} &\stackrel{\text{def}}{=} (\mathbf{w} \times \mathbf{t}), \\ \cos(\theta_{\mathbf{w}}) &\stackrel{\text{def}}{=} (\mathbf{n} \cdot \mathbf{w}), \\ \tan(\phi_{\mathbf{w}}) &\stackrel{\text{def}}{=} \frac{(\mathbf{b} \cdot \mathbf{w})}{(\mathbf{t} \cdot \mathbf{w})}. \end{aligned}$$

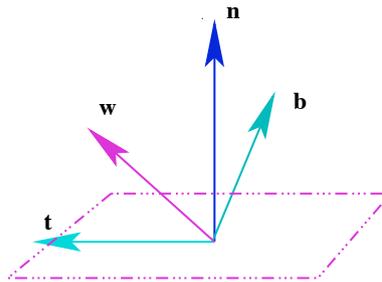


Figura 2.9: El sistema de referencia utilizado corresponde a un sistema local, dado un vector.

Puesto que la notación en los trabajos estudiados no es homogénea, en esta tesis se propone considerar una única descripción de la geometría de la BRDF. El sistema de coordenadas utilizado se forma alineando el eje Z con la normal perpendicular a la superficie estudiada. En la geometría intervienen —de forma elemental— el vector incidente \mathbf{u} , reflejado perfecto del incidente \mathbf{r} , la normal \mathbf{n} , y el vector de salida \mathbf{v} . Este sistema de referencia, llevado a nuestra particular notación, es el mostrado en la figura 2.10.

En general utilizaremos \mathbf{u} y \mathbf{v} para representar un par de vectores cualquiera (desligados del significado incidente o reflejado), ya que se entiende que la energía radiante tiene la misma probabilidad de viajar en un sentido que en otro, y que las BRDFs son simétricas. De esta forma notaremos $\mathbf{r}_{\mathbf{u}}$ al reflejado perfecto del vector \mathbf{u} y $\mathbf{r}_{\mathbf{v}}$ al reflejado perfecto del vector \mathbf{v} .

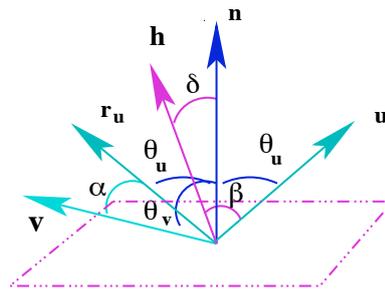


Figura 2.10: Vectores que intervienen en la reflexión de la luz local a un punto.

$$\forall \mathbf{u}, \mathbf{v}, \mathbf{n} \in \Omega \quad \left\{ \begin{array}{l} \alpha \stackrel{\text{def}}{=} \cos^{-1}(\mathbf{r}_u \cdot \mathbf{v}), \\ \beta \stackrel{\text{def}}{=} \cos^{-1}(\mathbf{u} \cdot \mathbf{h}), \\ \delta \stackrel{\text{def}}{=} \cos^{-1}(\mathbf{h} \cdot \mathbf{n}), \\ \mathbf{h} \stackrel{\text{def}}{=} \frac{\mathbf{u} + \mathbf{v}}{\|\mathbf{u} + \mathbf{v}\|}. \end{array} \right.$$

El vector \mathbf{h} o *halfway* fue introducido por Blinn [Bli77] y posteriormente utilizado en trabajos como los de Cook-Torrance [CT82]. Representa la dirección de máximo brillo, y permite agilizar los cálculos del modelo de reflexión que lo emplee. Más tarde, se propuso como sistema de referencia para representar una BRDF, por Rusinkiewicz [Rus98] dando los ángulos polar y azimutal con respecto al vector *highlight*, en lugar de hacerlo con respecto a la normal. La mejora se aprecia si utilizamos funciones base para representar la BRDF, ya que reduce el número de coeficientes que son necesarios.

2.3.1.3. Clasificación de las BRDFs

En general es posible clasificar las BRDFs atendiendo a tres criterios. En el primero de ellos [War92], se distinguen los siguientes tipos de modelos de reflexión:

- **Aproximados:** Se persigue una fórmula simple para la BRDF, ajustable mediante parámetros designados para simular un determinado tipo de reflexión. Se trata de modelos que son muy rápidos computacionalmente, pero no atienden a las propiedades físicas de la dispersión de la luz. Los casos más destacados por su utilización en *informática gráfica* son los de Phong [Pho75] y Blinn [BN76].
- **Teóricos:** Se basan en leyes físicas para dar una mejor aproximación a la distribución real. Sin embargo, implican un mayor costo computacional que hace que no sean utilizados en sistemas de iluminación comerciales. Entran dentro de esta categoría los modelos de Cook-Torrance [CT82], Kajiya [Kaj85] y He-Torrance-Sillion-Greenberg [HTSG91]. El problema de estos modelos es que sólo pueden

representar un conjunto limitado de materiales, dependiendo de los detalles que se asumen sobre la geometría.

- **Experimentales:** En ocasiones poder simular las características físicas no compensa, y se hace necesario ajustes, mediante medidas de la reflectancia, ya que algunos parámetros físicos son difíciles de medir. Si la teoría no se adapta a los datos, se propone un nuevo enfoque que ajusta lo mejor posible los valores obtenidos experimentalmente. Este es el caso de modelos que proponen Ward [War92] y posteriormente Schlick [Sch93].

El segundo criterio considera dos tipos de BRDFs, basados en la distinción de las características de los materiales, ya comentado en páginas precedentes.

- Las **BRDFs isótropas** son aquellas que representan a algunos materiales que no dependen de la orientación de la superficie y su reflexión no varía cuando ésta es rotada con respecto al sistema de referencia local. Este fenómeno ocurre generalmente en superficies rugosas. Para todo par de ángulos incidentes y reflejados, se comprueba que la BRDF es invariante:

$$\forall \alpha \in [0, 2\pi], \alpha > 0 \implies f_r(\theta_r, \phi_r, \theta_i, \phi_i) = f_r(\theta_r, \phi_r + \alpha, \theta_i, \phi_i + \alpha),$$

y, por tanto, depende sólo de tres parámetros:

$$f_r(\theta_r, \phi_r, \theta_i, \phi_i) = f_r(\theta_i, \theta_r, \phi_r - \phi_i).$$

- Una **BRDF anisótropa** refleja la luz de forma diferente para varios ángulos de rotación α , en concreto, si fijamos \mathbf{u} y \mathbf{v} y rotamos la superficie con respecto al vector normal \mathbf{n} , el porcentaje de luz reflejada puede cambiar. Por ello, necesita considerar las muestras de varias direcciones incidentes —en concreto los ángulos azimut ϕ_i — para la obtención de la BRDF.

Igualmente podemos encontrar otra clasificación basada en la forma de representación de la BRDF. Nos encontramos con dos estrategias de trabajo:

- **Técnicas de medición directa.** La BRDF puede ser medida directamente usando un goniorelectómetro [War92, GA97, GA99], que mecánicamente varía la dirección de una fuente de luz pequeña, y un sensor espectral que recoge un conjunto de medidas para la BRDF. También se han realizado medidas utilizando dispositivos más asequibles, como las cámaras digitales [DGNK99, MPBM03].
- **Métodos empíricos.** Existe una gran variedad de modelos de reflexión empíricos, algunos de los cuales nos resultan muy familiares, como son los modelos introducidos por Gouraud [Gou71] y Phong [Pho75, Bli77]. Desde entonces, otros muchos modelos han sido desarrollados.

En el primer caso, los datos procedentes de mediciones externas son precisos y nos permiten simular con elevada calidad el aspecto de un material. Sin embargo ello nos limita, ya que existen miles de materiales distintos y cada nuevo elemento que deseemos simular debe ser capturado. Es preferible trabajar con un modelo aproximado de reflexión, lo que nos permitirá aplicar el sistema de síntesis en la simulación de escenarios reales e imaginarios (interiores de inmuebles, pabellones deportivos, señalizaciones de carreteras, edificios en construcción, etc.). De esta forma, con tan solo modificar unos pocos parámetros, podemos reproducir cualquier BRDF.

2.3.2. Reflectancia de la superficie

La reflectancia ρ es un valor que mide la razón de energía reflejada con respecto a la energía incidente en un punto. Muestra lo que le ocurre a la luz cuando es reflejada desde una superficie; toma valores acotados en $[0, 1)$ y carece de unidades.

La BRDF es la función que proporciona más información, ya que expresa la dependencia de la dirección de la reflectancia y no sólo una ratio entre cero y uno. La reflectancia sobre un punto x de la superficie puede ser modelada calculando la integral sobre la semiesfera de la función BRDF, relativa a la medida del ángulo sólido proyectado [CWH93], lo que vendría a estar representado por la siguiente expresión:

$$\rho(\mathbf{u}) \stackrel{\text{def}}{=} \int_{\Omega} f_r(\mathbf{u}, \mathbf{v}) \cos(\mathbf{v}) d\sigma(\mathbf{v}).$$

En el estudio de la dispersión por reflexión se distinguen tres componentes denominados: dispersión especular perfecta, difusa perfecta y dispersión direccional difusa.

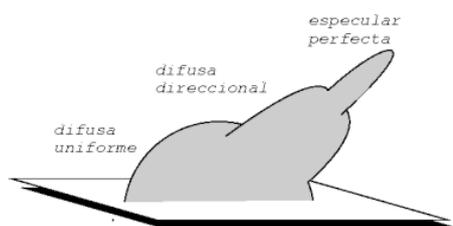


Figura 2.11: El fenómeno de reflexión de la luz visto desde la perspectiva de sus tres componentes: difusa uniforme, difusa direccional y especular.

La reflexión especular perfecta es un comportamiento ideal, apreciable a pequeña escala en las superficies especulares tales como un espejo. Su reflectividad se nota con el símbolo ρ_s . En determinadas condiciones —ángulos de incidencia próximos a 90° —, las superficies se comportan como un espejo. La cantidad de energía reflejada con

respecto a la incidente, depende por tanto del ángulo de incidencia \mathbf{u} , $\rho_s(\mathbf{u}) \in [0, 1)$ y está controlado por el *término de Fresnel* que se explica en el siguiente apartado. Normalmente es menor, cuanto menor es $(\mathbf{v} \cdot \mathbf{n}) = \cos(\mathbf{v})$.

La reflexión difusa perfecta, también conocida como *Lambertiana*, es independiente de la dirección de reflexión, ya que distribuye la proporción de luz reflejada igualmente en todas las direcciones. No existen materiales en el mundo real que sean puramente difusos. Para este caso idealizado de superficie la reflectividad se nota como $\rho_d \in [0, 1)$.

Finalmente, la componente direccional difusa es aquella que incluye los modos de dispersión que no son ideales —ni difusa, ni especular—. Es la forma de transferencia más compleja. En este caso, es posible considerar varios casos o tipos de reflectancia [Gla94]. Sea la *reflectancia hemisférica-direccional*, aquella que considera la fracción de energía reflejada en la dirección \mathbf{u} , respecto a la incidente en todas las direcciones:

$$\rho_{hd}(x, \mathbf{u}) = \frac{L_r(x, \mathbf{u})}{E(x)} = \int_{\Omega} f_r(x, \mathbf{u}, \mathbf{v}) d\sigma_p(\mathbf{u}). \quad (2.8)$$

Sea la *reflectancia direccional-hemisférica*, aquella que calcula la fracción de energía que es reflejada en todas direcciones, para un valor de irradiancia en un punto y una dirección:

$$\rho_{dh}(x, \mathbf{v}) = \frac{dE_r(x, \mathbf{v})}{dE_i(x)} = \int_{\Omega} f_r(x, \mathbf{u}, \mathbf{v}) d\sigma_p(\mathbf{v}). \quad (2.9)$$

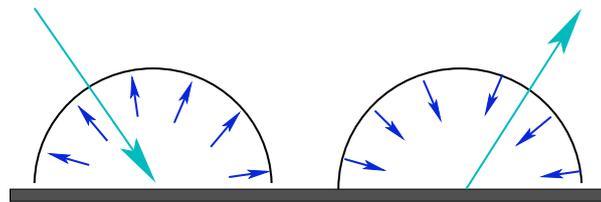


Figura 2.12: Consideraciones para la reflectancia direccional-hemisférica y hemisférica-direccional

Ambos valores son equivalentes y $\rho_{hd} = \rho_{dh}$, por lo que utilizaremos como notación única el símbolo ρ para referirnos a esta función que tanta importancia tiene en los algoritmos de *iluminación global*.

2.3.3. Las ecuaciones de Fresnel

En el evento de reflexión o transmisión no es suficiente con calcular la nueva dirección de la luz, sino que además es necesario conocer la fracción de luz incidente que

CAPÍTULO 2. Modelos de reflexión en iluminación global

es reflejada o transmitida. Este valor se conoce como *reflectividad* o *transmisividad*; físicamente estos términos no son constantes en la superficie y, además, dependen de la dirección de observación. La solución de las ecuaciones de Maxwell para superficies pulidas se conoce como *ecuaciones o términos de Fresnel*.

En un movimiento ondulatorio transversal (ver figura 2.13) los puntos vibran en sentido perpendicular a la dirección de propagación. Direcciones perpendiculares a las de propagación son todas las que se encuentran en un plano perpendicular a dicha dirección. En el caso que todos los puntos del medio, al ser alcanzados por la onda, vibren en una sola dirección del plano P, se dice que la onda está polarizada. Este tipo de polarización se llama lineal porque la proyección de todas las vibraciones de los puntos sobre el plano P es un segmento de recta, el segmento AB (de la figura 2.14). El plano en el que vibran todos los puntos de una onda polarizada se llama plano de polarización.

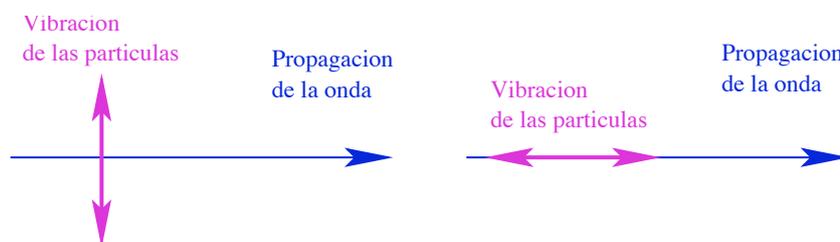


Figura 2.13: Onda transversal y onda longitudinal.

Un caso frecuente es que la vibración de los puntos no tenga lugar en un solo plano. Un movimiento ondulatorio de este tipo no dará una sola dirección de vibración sobre un plano perpendicular a la dirección de propagación. Las ondas de este tipo son ondas no polarizadas.

Las ondas no polarizadas, cuando atraviesan una rendija resultan polarizadas, pues sólo atravesará la abertura la componente de onda que lleve su misma dirección. El fenómeno de la polarización sólo es posible en movimientos ondulatorios transversales. De tal forma que puede servir de criterio, para distinguir entre ondas transversales y longitudinales.

Se distinguen dos tipos de ecuaciones: uno para materiales dieléctricos —aquellos que no conducen la corriente, como el cristal— y otro para eléctricos o conductores —metales—. Para cada uno de estos casos, las ecuaciones tienen dos formas, dependiendo de la polarización de la luz incidente —perpendicular \perp y paralelo \parallel — en lo que se conocen como coeficientes de amplitud de Fresnel: r para los asociados al fenómeno de reflexión y t para los involucrados en la transmisión.

Con los datos anteriores, y para una dirección incidente particular, se calcula la *reflectividad*, *término* o *factor de Fresnel* —notada como $F(\theta)$ o simplemente

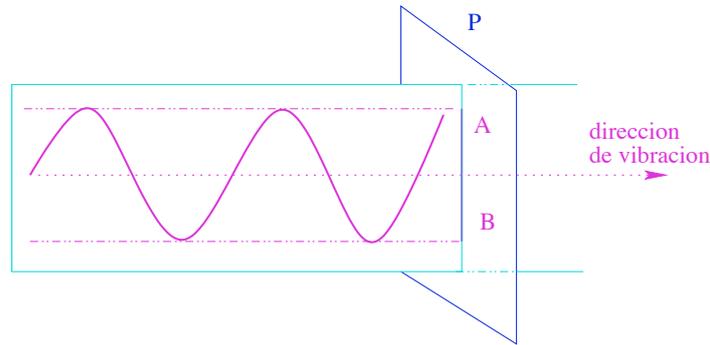


Figura 2.14: Onda polarizada rectilíneamente.

F — y la transmisividad, T . Debido al principio de conservación de la energía, la cantidad de energía que transmite un dieléctrico es $T = 1 - F$, también conocida como *transmitancia*. Así se tiene que el flujo se puede considerar dividido en dos componentes $\Phi = F\Phi + T\Phi$. Al calcular la radiancia reflejada o la transmitida, deberemos considerar, a su vez, ambos términos:

$$L_r = F L_i,$$

y, por considerar los ángulos sólidos de los flujos incidentes y transmitidos:

$$L_t = T \left(\frac{\eta_t}{\eta_i} \right)^2 L_i.$$

2.3.3.1. Caso A: materiales dieléctricos

Para el cálculo de la reflectancia de un dieléctrico es necesario conocer los índices de refracción de los dos medios —algunos materiales usuales pueden verse en la tabla 2.1— y considerar los coeficientes perpendicular y paralelo de la formulación:

$$r_{\parallel} = \frac{\eta_t \cos(\theta_i) - \eta_i \cos(\theta_t)}{\eta_t \cos \theta_i + \eta_i \cos \theta_t},$$

$$r_{\perp} = \frac{\eta_i \cos(\theta_i) - \eta_t \cos(\theta_t)}{\eta_i \cos(\theta_i) + \eta_t \cos(\theta_t)},$$

donde r_{\parallel} es la reflectancia para la luz incidente polarizada paralela y r_{\perp} es la reflectancia para la luz perpendicularmente polarizada. El medio incidente muestra un índice de refracción η_i y η_t y corresponde al medio en donde se produce la transmisión. Los coeficientes para la transmisión serían:

Medio	Índice de refracción η
Vacío	1.0
Aire (a nivel del mar)	1.00029
Hielo	1.31
Agua (20° C)	1.333
Cuarzo	1.46
Cristal	1.5 - 1.6
Diamante	2.42

Tabla 2.1: Índices de refracción de una variedad de objetos. Conocida la ratio de velocidad de la luz en el vacío con respecto a la velocidad de la luz en el medio.

$$t_{\parallel} = \frac{2\eta_i \cos(\theta_i)}{\eta_i \cos(\theta_t) + \eta_t \cos(\theta_i)},$$

$$t_{\perp} = \frac{2\eta_i \cos(\theta_i)}{\eta_i \cos(\theta_i) + \eta_t \cos(\theta_t)}.$$

La reflectancia es una suma ponderada de las reflectancias para el caso del haz incidente perpendicular y paralelo, asegurándose que se cumpla $s + p = 1$:

$$F = sF_{\parallel} + pF_{\perp},$$

siendo $F_{\parallel} = r_{\parallel}^2$ y $F_{\perp} = r_{\perp}^2$.

Para el caso de un sistema que tome luz despolarizada se escoge $s = p = 1/2$, lo cual toma una orientación aleatoria del haz de luz con igual probabilidad. En este caso, el término de Fresnel se calcula como:

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

De forma similar se obtiene la transmitancia a partir de:

$$T_{\parallel} = \left(\frac{\eta_t \cos(\theta_t)}{\eta_i \cos(\theta_i)} \right) t_{\parallel}^2,$$

$$T_{\perp} = \left(\frac{\eta_t \cos(\theta_t)}{\eta_i \cos(\theta_i)} \right) t_{\perp}^2.$$

Material	Índice de refracción η	Coefficiente de absorción κ
Oro	0.37	2.82
Plata	0.177	3.638
Cobre	0.617	2.63
Acero	2.485	3.433

Tabla 2.2: Valores medios representativos de algunos materiales conductores.

2.3.3.2. Caso B: materiales conductores o metales

Al contrario que los dieléctricos los conductores no transmiten luz, siendo parte de ésta absorbida y transformada en calor. La energía reflejada depende de η y del coeficiente de absorción κ . Algunos valores para estos datos son mostrados en la tabla 2.2. Ambas magnitudes dependen de la longitud de onda λ , por lo que se dan los valores medios.

La expresión de los coeficientes de Fresnel para conductores es como sigue:

$$r_{\parallel}^2 = \frac{(\eta^2 + \kappa^2) \cos^2 \theta_i - 2\eta \cos \theta_i + 1}{(\eta^2 + \kappa^2) \cos^2 \theta_i + 2\eta \cos \theta_i + 1},$$

$$r_{\perp}^2 = \frac{(\eta^2 + \kappa^2) - 2\eta \cos \theta_i + \cos^2 \theta_i}{(\eta^2 + \kappa^2) + 2\eta \cos \theta_i + \cos^2 \theta_i}.$$

El término F debe ser obtenido del espectro de reflectancia del material pulido, posteriormente, al multiplicarlo por $1/\pi$, conseguimos el espectro de un material más rugoso. Con su valor obtenemos la reflectancia de una superficie completamente especular, en términos del índice de refracción η , y el coeficiente de extinción κ para un ángulo de iluminación.

Para muchos conductores los valores de η y κ no son conocidos. En el artículo de Cook y Torrance [CT82] se presenta una aproximación que supone que la reflectancia del objeto ha sido medida en la dirección incidente; esto mismo, permitiría despejar de la ecuación los términos η y κ .

Cuando estos valores son desconocidos se puede ajustar F en base a la reflectancia para los ángulos 0 y $\frac{\pi}{2}$ —ya que $F(\frac{\pi}{2})$ vale 1 , independientemente de $\eta(\lambda)$ — y suponiendo $\kappa = 0$. Este cálculo alternativo presupone que la luz está despolarizada. Sea $c = \cos(\theta)$, se tiene entonces que la expresión para F dada por Cook-Torrance, es:

$$F(\theta) = \frac{1}{2} \frac{(g - c)^2}{(g + c)^2} \left\{ 1 + \frac{[c(g + c) - 1]^2}{[c(g - c) + 1]^2} \right\},$$

con $g^2 = \eta^2 + c^2 - 1$. Tomando el ángulo de incidencia, la anterior expresión se reduce a:

$$F(0) = \left\{ \frac{\eta - 1}{\eta + 1} \right\}^2,$$

y despejando η :

$$\eta = \frac{1 + \sqrt{F(0)}}{1 - \sqrt{F(0)}}.$$

Podemos realizar el mismo proceso para aproximar el coeficiente de absorción, asumiendo que $\eta = 1$, y dar una formulación de Fresnel alternativa. Luego se toma su valor para el ángulo de incidencia:

$$F(0) = \frac{\kappa^2}{\kappa^2 + 4},$$

y despejando:

$$\kappa = 2 \sqrt{\frac{F(0)}{1 - F(0)}}.$$

2.4. La ecuación de radiancia

En esta sección introducimos la ecuación de radiancia, que describe la transferencia de luz en entornos generales y es considerada como el cálculo más importante en la *síntesis realista de imágenes*. La interacción de la luz con la superficie es un evento local al punto en el que incide el rayo; Kajiyá [Kaj86] introdujo en la literatura de *iluminación global* una expresión matemática global de la radiancia como una ecuación integral de Fredholm de segundo grado, que representa la ecuación del transporte de la luz (abreviado como LTE).

La ecuación del transporte de la luz describe la distribución en equilibrio de la función de radiancia en una escena. Los algoritmos de *iluminación global* calculan el valor total de la radiancia que sale de un punto, considerando la dispersión de la luz producida por la radiancia incidente y la propia emisión del objeto. Es más, si no se considera la participación del medio, la radiancia no varía a lo largo de una línea recta, por lo que —siguiendo la geometría del gráfico 2.15— se tiene que:

$$L_i(x, \mathbf{v}) = L_r(y, \mathbf{u}'),$$

siendo $y = t(x, \mathbf{v})$, si t es la función que nos calcula la primera intersección del rayo —con origen en x y dirección \mathbf{v} — con alguna superficie. Esta función es denominada *ray casting function* y consideramos que está definida para entornos abiertos y

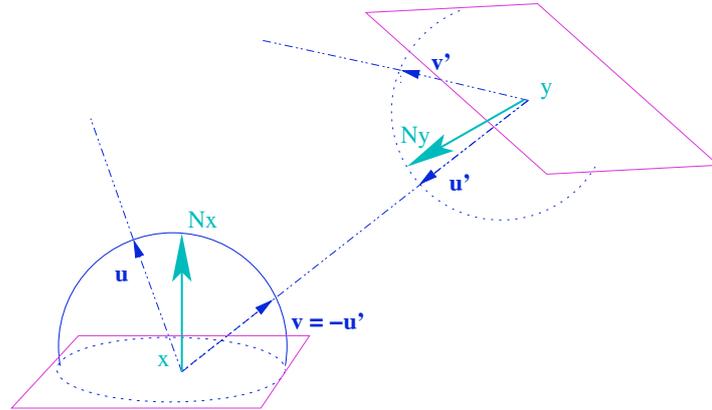


Figura 2.15: Geometría de la radiancia que deja un punto \$y\$ y llega a otro punto \$x\$.

cerrados, aunque no es siempre el caso.

La radiancia total que abandona un elemento de superficie, $L(x, \mathbf{u})$, se descompone en dos: la radiancia emitida y la reflejada. La primera, L_e , es dada en función de la dirección de salida; es un dato conocido, ya que forma parte de la descripción de la escena. La segunda, L_r , no es tan sencilla, ya que depende a su vez de toda la radiancia que incide en la superficie L_i —integrado sobre el hemisferio de direcciones entrantes—, y que proviene de la reflejada de otra superficie, por lo que interviene en su propia definición. Utilizando esta definición en la radiancia reflejada, y eliminando la x de la notación por comodidad tenemos:

$$L_r(\mathbf{u}) \stackrel{\text{def}}{=} \int_{\Omega} dL_r(\mathbf{u}) d\sigma(\mathbf{v}) \quad (2.10)$$

$$= \int_{\Omega} f_r(\mathbf{u}, \mathbf{v}) L_i(\mathbf{v}) (\mathbf{v} \cdot \mathbf{n}) d\sigma(\mathbf{v}). \quad (2.11)$$

Evaluar la radiancia reflejada de forma analítica es imposible —salvo escenarios muy simples—, ya que la visibilidad de un objeto, o el cálculo de la radiancia incidente, deja de ser trivial en escenas complejas donde no hay facilidades para predecir el integrando. En la literatura de *iluminación global* es usual la notación de ángulos:

$$L_r(\theta_r, \phi_r) = \int_0^{2\pi} \int_0^{\frac{\pi}{2}} f_r(\theta_i, \phi_i, \theta_r, \phi_r) L_i(\theta_i, \phi_i) \cos(\theta_i) \sin(\theta_i) d\theta_i d\phi_i$$

Finalmente podemos dar la ecuación completa de la radiancia, de gran interés en los algoritmos de síntesis de imágenes realistas, haciendo uso de la medida de ángulo sólido proyectado (aquella de la ecuación 2.2):

$$L(\mathbf{u}) = L_e(\mathbf{u}) + \int_{\Omega} f_r(\mathbf{u}, \mathbf{v}) L_i(\mathbf{v}) d\sigma_p(\mathbf{v}). \quad (2.12)$$

CAPÍTULO 3

Modelos de BRDFs

El papel de los modelos de iluminación es determinar cuánta luz es reflejada hacia el observador procedente de un punto visible, así como describir el color y la distribución espacial de la misma. Vista la evolución de los modelos de reflexión, no siempre se ha considerado la realidad y las propiedades físicas del modelo, sino que se ha intentado encontrar un resultado válido para el sistema visual humano bajo un entorno simple. En un principio estos trabajos sólo consideran el comportamiento isótropo de superficies con un tipo de rugosidad aleatoria.

En *informática gráfica* se da importancia tanto a la calidad de los resultados, como a la rapidez en la obtención de los mismos. Sin embargo, además de estas características también se intenta asegurar otras propiedades en la formulación en los algoritmos de sombreado, como, por ejemplo, la reciprocidad de la BRDF y la conservación de la energía.

Ciertamente, al comenzar el repaso histórico de los modelos de iluminación y sombreado, se encuentran numerosas referencias al que resulta ser el modelo más conocido por excelencia: Phong. Los trabajos de Beckmann & Spizzichino [BS63] y Torrance & Sparrow [TS67] consiguen incorporar un gran número de propiedades físicas; entre ellas podríamos citar la polarización de la luz, cuyo estudio se centró en la reflexión especular, observando que el máximo brillo no tiene por qué aparecer a lo largo del vector \mathbf{h} (*highlight*). Este efecto es modelado por la *ecuación de Fresnel*, que causa incrementos en la reflectividad a medida que el ángulo de incidencia se aproxima a $\pi/2$. Posteriormente otros trabajos incluirían, a su vez, el factor de Fresnel.

Muchos de estos modelos, al merecer una atención especial, se comentarán más adelante en los sucesivos apartados. La mayoría de ellos han sido estudiados para su incorporación a un sistema de síntesis global basado en métodos de Monte-Carlo. Relacionado con su implementación, nos referimos a ellos en su notación GRF

(*G*Ranada *File*), formato de escena propio que puede conocerse en mayor detalle consultando el apéndice A.

3.1. Reflexión especular ideal (1621)

Cuando un rayo alcanza la superficie la transferencia producida en ese momento, se modela como la emisión de un nuevo rayo, tal y como se hace en los algoritmos basados en trazado de rayos [Whi80, Arv86]. En el caso de una superficie idealizada de forma especular, la luz sólo se refleja en una dirección, cumpliendo la *Ley de reflexión* vista en el apartado 2.2.1. La BRDF que lo modela es una distribución *delta de Dirac*, δ , que toma siempre el valor cero menos en una dirección \mathbf{r} . Se define a continuación:

$$\delta_{\mathbf{u}}(\mathbf{v}) \stackrel{\text{def}}{=} \begin{cases} 1 & \text{si } \mathbf{v} = \mathbf{r}_{\mathbf{u}}, \\ 0 & \text{en otro caso.} \end{cases}$$

Simplifica en gran medida el cálculo de la energía reflejada: $L_{r,s}(x, \mathbf{u}) = L_i(x, \mathbf{r}_{\mathbf{u}})$. En este caso además, se mantiene el ángulo sólido del haz luminoso reflejado, esto es, $d\sigma(\mathbf{r}_{\mathbf{u}}) = d\sigma(\mathbf{u})$.

Computacionalmente la BRDF resultante es muy sencilla, y se basa en la reflectividad del caso especular ideal:

$$f_{r,s}(\mathbf{u}, \mathbf{v}) = \rho_s(\mathbf{v}) \delta_{\mathbf{u}}(\mathbf{v}), \quad (3.1)$$

La expresión de esta BRDF en formato GRF considera ρ_s (notado ks) parametrizable:

```
PERFECTSPECULARBRDF ::=  
  brdf specular ks =valor .
```

Tabla 3.1: Notación en GRF para PERFECTSPECULAR-BRDF

3.2. Lambert (1760)

La reflexión difusa ideal —conocida por muchos como *lambertiana*— es independiente de la dirección de salida, en el que influye tan sólo la dirección incidente. Intervienen en este evento varios fenómenos de dispersión, producidos tanto en la superficie como en las capas internas del objeto.

Una superficie difusa tiene una BRDF igual para todas las direcciones entrantes y salientes; viene dada por la reflectividad difusa ρ_d , un valor que depende del punto y que caracteriza a la superficie. El factor π en el denominador es el resultado de la

integración del término $\cos(\theta_{\mathbf{v}})$ en todas las direcciones:

$$f_{r,d}(\mathbf{u}, \mathbf{v}) = \frac{\rho_d}{\pi} \quad (3.2)$$

donde $f_{r,d}$ representa el caso de BRDF constante para superficies difusas ideales. Permite simplificar notablemente el cálculo de la radiancia reflejada.

La expresión de esta BRDF en formato BNF (con ρ_d notado kd) sería:

PERFECTDIFFUSEBRDF ::=
brdf diffuse kd =*valor* .

Tabla 3.2: Notación en GRF para PERFECTDIFFUSEBRDF

3.3. Minnaert (1941)

Minnaert [Min41] fue uno de los primeros en describir los modelos de iluminación. Su propósito es la simulación de la reflectancia de la superficie lunar, que aplicado a otros materiales produce en ellos un efecto de oscurecimiento. Considera sólo materiales isótropos y emplea únicamente dos parámetros:

- ρ_d : reflectividad difusa (lambertiana)
- k : exponente de oscurecimiento

La expresión de esta BRDF es la siguiente:

$$f_r(\mathbf{u}, \mathbf{v}) = \frac{\rho_d}{\pi} (\cos(\mathbf{u}) \cos(\mathbf{v}))^{k-1}. \quad (3.3)$$

Cuando $k = 1$, esta BRDF coincide con BRDF de Lambert.

Si utilizamos la notación BNF para el formato de descripción de escenas GRF de nuestro sistema de síntesis, el modelo queda descrito como:

MINNAERTBRDFs ::=
brdf minnaert k =*valor* kd =*valor*

Tabla 3.3: Notación en GRF para MINNAERTBRDFs

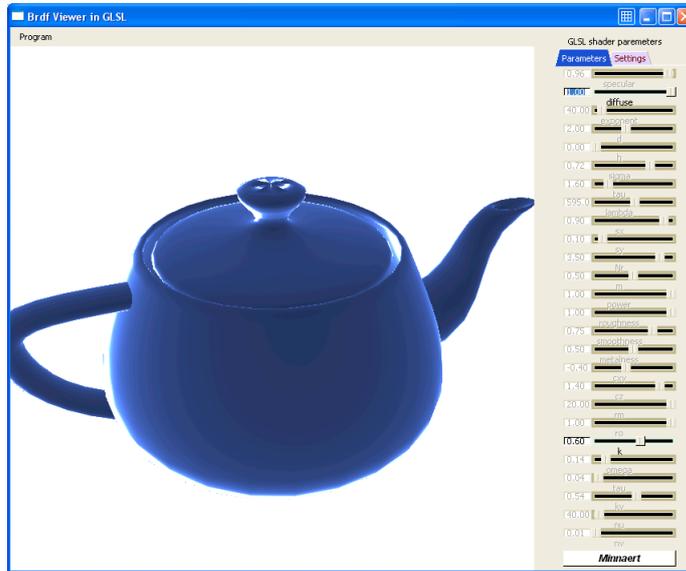


Figura 3.1: La BRDF *Minnaert* se visualiza según sus parámetros, desde nuestro editor GLSL de BRDFs.

3.4. Torrance-Sparrow (1967)

Modelo físico que se centra en la simulación de la interacción de la luz con la superficie del objeto. Considera que esa superficie está formada por una serie de cavidades cóncavas de igual lado, orientadas de forma aleatoria, denominadas *microfacets*, y son las encargadas de dar aspecto rugoso a la superficie [TS66, TS67].

$$f_r(\mathbf{u}, \mathbf{v}) = \frac{k_d}{\pi} + \frac{k_s}{4\pi (\mathbf{n} \cdot \mathbf{v})} D(\mathbf{h}) F(\theta_{\mathbf{u}}) G(\mathbf{u}, \mathbf{v}). \quad (3.4)$$

La anterior es la expresión de la BRDF, en la que intervienen:

- $D : \Omega \rightarrow \mathbb{R}$ con $D(\mathbf{h}) \in [0, 1]$. Fracción de *microfacets* que se encuentran orientados en la dirección del vector \mathbf{h} .
- $F(\theta_{\mathbf{u}}) \in [0, 1]$. Factor de Fresnel, que nos da la fracción de luz incidente que es reflejada en cada *microfacet*. Aunque no se indique expresamente, esta función depende de propiedades del material como η y κ .
- $G(\mathbf{u}, \mathbf{v}) \in [0, 1]$. Coeficiente de atenuación geométrica, entre dos vectores cualesquiera, o razón de luz que no es ocultada por la superficie.

Cada término es dependiente de la longitud de onda λ , aunque en ocasiones se omita en la notación por comodidad. En consecuencia, habrá que calcular cada término para cada longitud de onda del espectro visible (al menos tres, en el modelo RGB).

La orientación de los *microfacets* sigue una función de distribución D (similar a la figura 3.2), y, aunque se han propuesto diferentes distribuciones de probabilidad, siempre se construyen basándose en el parámetro m que caracteriza en que grado están alineadas las normales de la distribución, al vector \mathbf{h} y que caracteriza la rugosidad de la superficie:

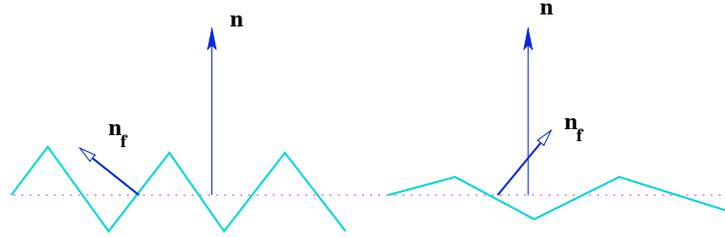


Figura 3.2: Un modelo de cavidades queda descrito por una distribución de las normales del *microfacet* con respecto a la normal de la superficie. Cuanto mayor sea la variación, mayor será la rugosidad.

En el caso de usar una distribución gaussiana, la fracción de *microfacets* orientados hacia \mathbf{h} (según el ángulo δ de la figura 2.10) dada por Beckmann [BS63] es la siguiente:

$$D(\mathbf{h}) = \frac{1}{m^2 \cos(\delta)^4} \exp\left(\frac{\cos(\delta)^2 - 1}{m^2 \cos(\delta)^2}\right)$$

Este modelo incorpora efectos de polarización de la luz mediante el uso de matrices de Jones para contar las diferencias de fase y de estado de polarización. Con esto, propone una formulación diferente a las ecuaciones clásicas de Fresnel, basadas en la teoría de Maxwell (consúltese la sección 2.3.3). Se distinguen dos tipos en función de la polarización del flujo incidente:

- Polarización perpendicular

$$r_{\perp}(\theta) = \frac{a^2 + b^2 - 2a \cos(\theta) + \cos^2(\theta)}{a^2 + b^2 + 2a \cos(\theta) + \cos^2(\theta)}$$

donde:

$$2a^2 = \sqrt{(\eta^2 - \kappa^2 - \sin^2(\theta))^2 + 4\eta^2\kappa^4} + (\eta^2 - \kappa^2 - \sin^2(\theta))$$

$$2b^2 = \sqrt{(\eta^2 - \kappa^2 - \sin^2(\theta))^2 + 4\eta^2\kappa^4} - (\eta^2 - \kappa^2 - \sin^2(\theta))$$

con η el índice simple de refracción y κ el coeficiente de extinción. Ambos son dependientes de la frecuencia ν , ya que estas fórmulas parten de las ecuaciones de Maxwell.

- Polarización paralela

$$r_{\parallel}(\theta) = r_{\perp}(\theta) \frac{a^2 + b^2 - 2a \sin(\theta) \tan(\theta) + \sin^2(\theta) \tan^2(\theta)}{a^2 + b^2 + 2a \sin(\theta) \tan(\theta) + \sin^2(\theta) \tan^2(\theta)}$$

Los dos coeficientes se utilizan para calcular la reflectancia producidas del fenómeno de la reflexión. El fractor de Fresnel, F , se calcula mediante una combinación lineal de ambos controlada por los dos valores reales s y p , tal y como se vio en la sección 2.3.3 sobre las ecuaciones de Fresnel.

Otros efectos reproducidos por este modelo son el bloqueo de la luz, originados por los altibajos formados en los *microfacets*; son conocidos como *shadowing* y *masking*, y se modelan con el factor geométrico G de la siguiente forma:

$$G(\mathbf{u}, \mathbf{v}) = \min \left\{ 1, \frac{2(\mathbf{n} \cdot \mathbf{h})(\mathbf{n} \cdot \mathbf{u})}{(\mathbf{u} \cdot \mathbf{h})}, \frac{2(\mathbf{n} \cdot \mathbf{h})(\mathbf{n} \cdot \mathbf{v})}{(\mathbf{u} \cdot \mathbf{h})} \right\}.$$

En general podemos decir que, si bien este modelo considera muchas de las características que determinan el comportamiento de la superficie ante la luz, queda limitado al no poder modelar diferentes tipos de superficies. No obstante, es el modelo físico más completo para materiales isótropos.

3.5. Beard-Maxwell (1973)

Modelo destinado a la descripción de materiales pintados (preferentemente isótropos). Con un enfoque mixto Maxwell *et al.* [MBWL73] propusieron un modelo empírico basado en propiedades físicas, dependiente de un número reducido de parámetros.

La BRDF se representa mediante dos componentes, difuso y especular, que se denominan volumétrico y superficial, respectivamente. La superficie se modela con varios *microfacets* orientados de varias formas en un espacio tridimensional, siguiendo una función de distribución estadística D . La reflexión se produce principalmente por dos eventos: reflexión especular local a la primera capa (cuya normal es \mathbf{n}) y otra que aproxima la aportación de las reflexiones producidas en los substratos debido al hecho de que la luz profundiza en el material:

$$f_{r\lambda}(\mathbf{u}, \mathbf{v}) = f_{r,sup\lambda}(\mathbf{u}, \mathbf{v}) + f_{r,vol\lambda}(\mathbf{u}, \mathbf{v}).$$

La reflexión $f_{r,sup}$ se rige principalmente por el término de Fresnel para dieléctricos (tomando $\kappa = 0$ y $\eta = 1,65$), interviniendo, además, el ángulo β o bisectriz entre la dirección entrante y la saliente:

$$f_{r,sup}(\mathbf{u}, \mathbf{v}) = -\frac{F(\beta)}{F(0)} \frac{f_r(\mathbf{h}) \cos^2(\mathbf{h})}{\cos(\mathbf{u}) \cos(\mathbf{v})} SO(\tau, \Omega),$$

siendo

$$f_r(\mathbf{h}) = \frac{F(0) D(\mathbf{h})}{4 \cos(\mathbf{u}) \cos(\mathbf{v})}.$$

El trabajo de Beard y Maxwell [MBWL73] está muy vinculado al modelo de Torrance-Sparrow [TS66, TS67], que toman del mismo la idea de modelar la superficie utilizando *microfacets*; sin embargo, derivan una función empírica propia para modelar los efectos de enmascaramiento e interbloqueo (mostrados en la figura 3.3). El término SO , que se muestra a continuación, se refiere a ambos, llamados *shadowing* y *obscuration*:

$$SO(\tau, \Omega) = \frac{1 + \frac{\theta_{\mathbf{h}}}{\Omega} e^{-2\beta/\tau}}{1 + \frac{\theta_{\mathbf{h}}}{\Omega}} \left(\frac{1}{1 + \frac{\phi_{\mathbf{h}} \theta_{\mathbf{u}}}{\Omega}} \right).$$

Los parámetros que intervienen τ , Ω se toman a partir de mediciones, pero nada se indica con respecto a $\phi_{\mathbf{h}}$. Trabajos más recientes de Westlund y Meyer [WM02], proporcionan una base de datos de dominio público¹ con más de 400 materiales (agrupados en 12 categorías), que emplean un modelo de Beard-Maxwell modificado denominado *NEF-BM*. Una de las simplificaciones que realiza este modelo reside en la eliminación del parámetro $\phi_{\mathbf{h}}$ y las funciones f y g ; consiguen la anisotropía promediando los valores isótropos calculados.

En la reflexión volumétrica no se considera explícitamente la absorción provocada por el medio, ni tan siquiera la posible transmisión de energía a través de él. Así mismo, se considera que la luz no cambia de polarización, esto es, está depolarizada.

$$f_{r,vol}(\mathbf{u}, \mathbf{v}) = \frac{2 \rho_v f(\beta) g(\theta_{\mathbf{h}})}{\cos(\mathbf{u}) \cos(\mathbf{v})}.$$

El valor ρ_v es un valor constante que se obtiene midiendo la reflectancia del material considerando $\theta_{\mathbf{u}} = \theta_{\mathbf{v}} = 0$ y $f(\beta) = g(\theta_{\mathbf{h}}) = 1$. La expresión detallada de f y g puede encontrarse en Maxwell *et al.* [MBWL73].

¹ NEF User Guide, 9.1 <<http://math.nist.gov/~FHunt/appearance/nedfs.html>> Consultado en Octubre de 2005

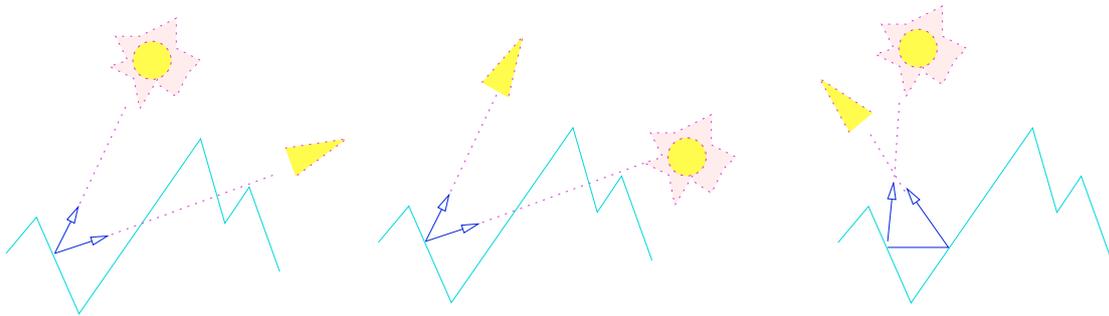


Figura 3.3: De izquierda a derecha: *masking*, la intensidad de un *microfacet* no es visible al observador por el bloqueo producido por otro *microfacet*; *shadowing*, cuando la luz no llega a un *microfacet* por oclusión; *interreflexión* cuando la luz se refleja más de una vez en el *microfacet* antes de alcanzar al observador.

```

BEARDMAXWELLBRDF ::=
brdf beardmaxwell omega =valor    tau =valor
nreal =valor    nimag =valor    Rs =valor
Rd =valor    Rv =valor
    
```

Tabla 3.4: Notación en GRF para BEARDMAXWELLBRDF

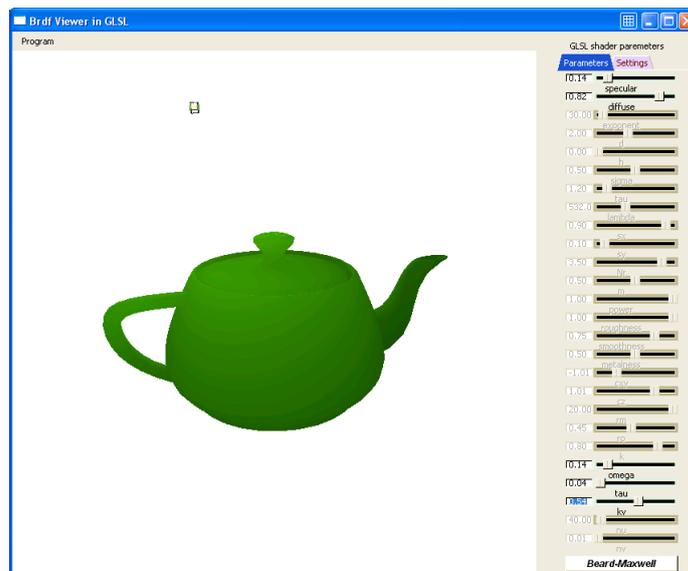


Figura 3.4: La BRDF de *Beard-Maxwell* visualizada en nuestro editor GLSL de BRDFs.

3.6. Phong (1975)

El primer modelo de reflexión para superficies no-lambertianas usado en *informática gráfica* fue propuesto por Phong [Pho75]. Entra dentro de la categoría de modelos empíricos y como tales, rápidos y sencillos computacionalmente pero faltos de significado físico; tan sólo dependen del ángulo α , calculado tras determinar el vector reflejado \mathbf{r} con respecto a la dirección incidente \mathbf{v} :

$$\cos(\alpha) = (\mathbf{u} \cdot \mathbf{r}_{\mathbf{v}}).$$

Lo que hace a este modelo tan rápido es que ignora los factores G y F del de Torrance & Sparrow. El término D queda reducido a la siguiente expresión:

$$D(\mathbf{u}, \mathbf{v}) = \cos^n(\alpha),$$

con el parámetro $n \in [0, \infty[$ para caracterizar el brillo especular del material.

En la bibliografía de *informática gráfica* cabe citar numerosos trabajos que han intentado reducir, con mayor o menor éxito, el costo asociado con la exponenciación. Se ha almacenado de forma tabular la igualdad $\cos^n(\alpha) = e^{n \log \cos(\alpha)}$, y también se ha buscado su aproximación. Schlick [Sch94a] propuso la siguiente aproximación:

$$\cos^n(\alpha) \approx \frac{\cos(\alpha)}{n - n \cos(\alpha) + \cos(\alpha)}.$$

Otros parámetros de este modelo son las constantes k_s y k_d , ambas en el rango $[0, 1]$, que permiten la combinación lineal las componentes de la reflexión: difusa y especular.

$$f_r(\mathbf{u}, \mathbf{v}) = k_d f_{r,d}(\mathbf{u}, \mathbf{v}) + k_s f_{r,s}(\mathbf{u}, \mathbf{v}).$$

Mostramos, finalmente, la expresión para la BRDF dada por Phong:

$$f_r(\mathbf{u}, \mathbf{v}) = k_d (\mathbf{v} \cdot \mathbf{n}) + k_s (\mathbf{u} \cdot \mathbf{r})^n, \quad (3.5)$$

En este modelo se echa en falta la propiedad de conservación de la energía, si no se impone ninguna restricción a los parámetros k_d y k_s . Cuando $k_d + k_s \leq 1$ se cumple la conservación de la energía, pero el albedo es muy inferior a 1 y los resultados aparecen oscurecidos. Los parámetros utilizados no se corresponden con propiedades de los materiales.

La definición de esta BRDF para un sistema de síntesis de imágenes se daría como:

PHONGBRDF ::= *brdf phong n =valor ks =valor kd =valor*

Tabla 3.5: Notación en GRF para PHONGBRDF

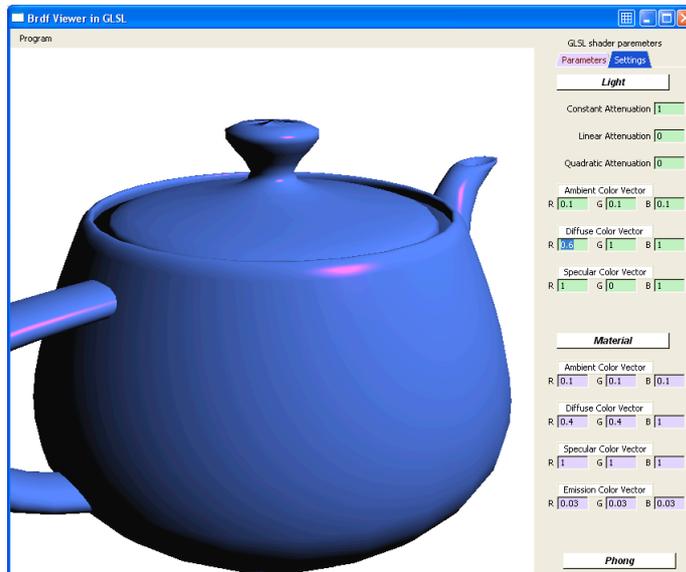


Figura 3.5: La BRDF de *Phong* se visualiza desde nuestro editor GLSL de BRDFs.

3.7. Blinn (1977)

Según Blinn [Bli77] las superficies tienden a reflejar más luz en la dirección en donde el ángulo incidente y el reflejado se igualan con respecto a la normal. Basándose en la idealización de *microfacets*, e ignorando los términos F y G como Phong, extiende el modelo de este último y emplea en lugar del vector reflejado (véase ilustración 2.10), el vector \mathbf{h} . Con ello, se aproxima la distribución D que orienta los *microfactes* mediante una la caída exponencial que es computacionalmente más eficiente:

$$D(\mathbf{h}) = (\mathbf{n} \cdot \mathbf{h}) = \cos(\delta).$$

De esta manera, la BRDF de Blinn tiene la siguiente expresión:

$$f_r(\mathbf{u}, \mathbf{v}) = k_d(\mathbf{u} \cdot \mathbf{n}) + k_s(\mathbf{n} \cdot \mathbf{h})^n \quad \text{con } k_d + k_s = 1. \quad (3.6)$$

La definición de esta BRDF es muy similar a la de Phong:

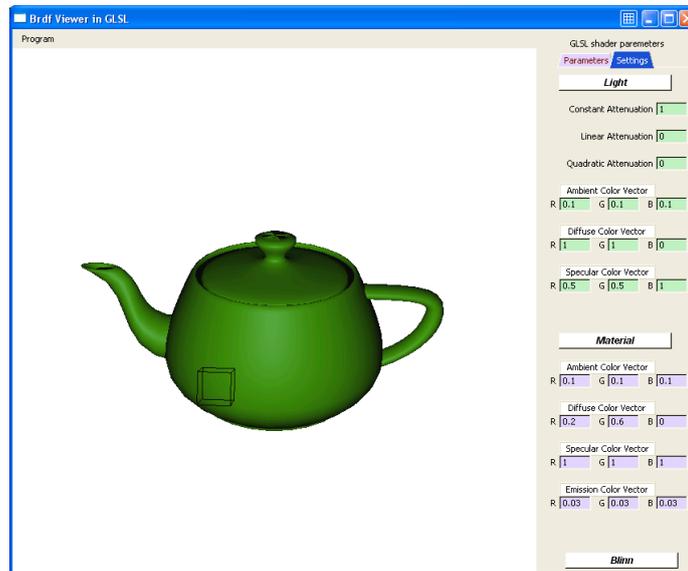


Figura 3.6: La BRDF de *Blinn* es editada con una aplicación GLSL para las BRDFs.

BLINNBRDF ::=
brdf blinn n =*valor* ks =*valor* kd =*valor*

Tabla 3.6: Notación en GRF para BLINNBRDF

3.8. Cook-Torrance (1981)

Los objetos reales no son puramente especulares (espejos), ni puramente difusos (superficies *lambertianas*), por lo que una combinación lineal de ambos componentes, como sugiere Phong [Pho75], no es suficiente. Blinn [Bli77] también usa esa idea, salvo que considera los brillos que ocurren en los ángulos cercanos al horizonte (90° con respecto a la normal de la superficie). Cook y Torrance usaron los trabajos de Beckmann y Torrance-Sparrow [BS63, TS67] con la siguiente salvedad: sólo contribuyen a la reflexión aquellos *microfacets* orientados en la dirección \mathbf{h} . Además se propone una mejora a modelos anteriores, como el de Blinn, incorporando la distinción entre superficies metálicas y no metálicas, y considerando las variaciones que produce λ en la reflectividad. Cook y Torrance se basan en óptica geométrica y consideran que la reflexión contiene tres componentes:

- Ambiente, procedente de la distribución uniforme de la luz incidente, suponiendo la existencia de fuentes de área extendida.
- Difuso, cuando consideramos que las fuentes de luz de la escena son incidentes en todas las direcciones.

- Especular, cuando se considera la contribución de las fuentes de luz, tan sólo para una dirección particular.

Las dos primeras componentes tomarán el mismo color que el del material. A diferencia de éstas, la componente especular utilizará la *ecuación de Fresnel* para obtener la variación angular de la intensidad y no así el color.

En primer lugar se calcula el vector normalizado \mathbf{h} y se interpreta como la normal a una superficie imaginaria que refleja especularmente la luz.

Este modelo depende implícitamente de varias variables, principalmente:

- Los coeficientes k_s y k_d , se obtienen a partir de las propiedades del material. Controlan la contribución especular y difusa del objeto.
- La reflectancia que depende, además, de las dos variables anteriores, de la geometría y rugosidad de la superficie, así como del ángulo de incidencia de la luz. Se obtiene de la aportación de dos componentes, una parte difusa y otra especular, a modo de combinación lineal:

$$\rho = \rho_a + k_s \rho_s + k_d \rho_d, \quad \text{donde } k_s + k_d < 1.$$

Al mismo tiempo se considera la intensidad de luz ambiental reflejada, siendo el término ρ_a la reflectancia ambiental, calculada a partir de ρ_s y ρ_d . Sobre ρ_a y ρ_d , se supone, que no dependen de la posición del observador.

La componente especular de la reflectancia emplea en sus cálculos los términos F , D y G . Recordemos que el primero de ellos se conoce como *factor o término de Fresnel* (ver sección 2.3.3) y representa el modo en el que se produce la reflexión de la luz en una superficie poco rugosa. D es la distribución que siguen la inclinación o pendiente de las normales de cada *microfacet*. Hasta la fecha se usaba típicamente la distribución gaussiana. Cook-Torrance [CT82] propone una distribución alternativa la dada por Beckmann [BS63]. Ambas están controladas por un parámetro del modelo utilizado para caracterizar la rugosidad del material m . De esta forma, si el valor de m es pequeño, la distribución es altamente direccional, siendo la dirección resultante la especular perfecta. Para valores grandes de m la distribución se extiende más y se hace uniforme:

$$D(\mathbf{h}) = \cos(\beta) e^{-\left(\frac{\alpha}{m}\right)^2} \quad \text{Cook - Torrance,}$$

$$D(\mathbf{h}) = \frac{1}{m^2 \cos^4(\alpha)} e^{-\left[\frac{\tan(\alpha)}{m}\right]^2} \quad \text{Beckman.}$$

3.8. Cook-Torrance (1981)

Por último, el *factor de atenuación geométrico*, G , considera los aspectos relacionados con los bloqueos provocados por los mismos *microfacets* entre sí (*shadowing* y *masking*). Su definición es igual a la vista en la sección 3.4 correspondiente a la BRDF de Torrance-Sparrow. El término G esta pensado para compensar el bloqueo de la luz y las oclusiones que se producen en la luz al ser reflejados de forma local por una concavidad. El inconveniente de los modelos que sólo consideran una reflexión primaria en la superficie, es que asumen que toda la luz se absorbe y no consideran lo que pasa con la que ha sido bloqueada o reflejada más veces. Finalmente, nos queda la expresión de la BRDF como:

$$f_{r,s}(\mathbf{u}, \mathbf{v}) = \frac{F(\beta) D(\mathbf{h}) G(\mathbf{u}, \mathbf{v})}{\pi (\mathbf{n} \cdot \mathbf{u})(\mathbf{n} \cdot \mathbf{v})}. \quad (3.7)$$

Una vez calculada la componente especular, se ha observado que el objeto conserva el color y se combina con el color de la iluminación incidente, con la salvedad de los plásticos y los materiales pintados, que suelen mostrar un aspecto diferenciado en sus componentes difuso y especular.

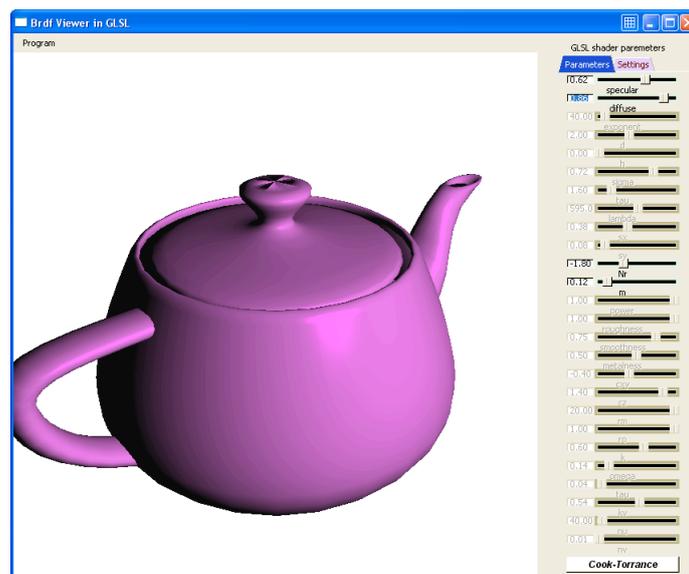


Figura 3.7: La BRDF *Cook-Torrance* se visualiza según sus parámetros, desde nuestro editor GLSL de BRDFs.

Este modelo considera las características físicas de las superficies, relacionando el brillo con la intensidad y tamaño de cada fuente de luz, y con la capacidad de reflexión de dichas superficies. Pero, como en el modelo de Phong, los parámetros no son independientes y es difícil derivar su valor, ya que éste como aquéllos, no son muy intuitivos al diseñador gráfico. Utilizando el formato GRF se define mediante los siguiente parámetros:

TORRANCEBRDF ::=
brdf *torrance* *rough* =*valor* *nreal* =*valor*
nimag =*valor* *ks* =*valor* *kd* =*valor*

Tabla 3.7: Notación en GRF para TORRANCEBRDF

3.9. Kajiya (1985)

Otro modelo, el de Kajiya [Kaj85], implementa un método anisótropo de fuerza bruta calculando de forma analítica la intensidad de luz reflejada en una superficie. El modelo está basado en la aproximación de Kirchoff y en un método de base estacionaria para el cálculo por aproximación de la integral que expresa la ecuación de radiancia (ver Eq. 2.12). Realiza una simplificación y reemplaza toda superficie rugosa por su plano tangente normal más próximo.

Calcula la reflectancia y la almacena en una tabla para obtener posteriormente valores por interpolación lineal. Sin embargo, es necesario volver a calcular dicha tabla para cada superficie nueva alcanzada. Además, explora una técnica numérica al usar las propiedades de la luz no polarizada y el término de Fresnel.

Como inconvenientes a este modelo se encuentra el que no considera reflexiones entre elementos de una escena, ni tampoco las sombras arrojadas, no garantizando, por otra parte, el principio de conservación de la energía.

3.10. Poulin-Fournier (1990)

En su trabajo Poulin y Fournier [PF90] presentan un modelo para la reflexión y la refracción de la luz en materiales anisotrópicos, que son simulados mediante un conjunto de cilindros paralelos fuertemente empaquetados, similares a los usados por Miller [Mil84] y cuyas características ópticas son ajustables mediante dos parámetros que describen los cilindros: la altura y la distancia entre ellos. De esta forma se consigue variar el grado de anisotropía.

La superficie anisótropa es modelada mediante una distribución de cilindros que se agregan o sustraen a la superficie base. Teniendo en cuenta que el radio del cilindro es muy pequeño en comparación con la longitud de éste, la reflexión se puede aproximar por dicho evento, pero producido en el corte longitudinal del cilindro. Se utiliza el vector tangente, para especificar la orientación de los cilindros; en concreto, se emplea el plano NB (normal-binormal), donde $\mathbf{b} = \mathbf{t} \times \mathbf{n}$. Los ángulos incidente $\theta_{\mathbf{v}}$ y reflejado $\theta_{\mathbf{u}}$ se toman proyectados en el plano NB, esto es: θ_v y θ_u . Por tanto, la información necesaria en cada punto de la superficie es el vector normal \mathbf{n} y la tangente \mathbf{t} .

La anisotropía es creada variando la distribución de las normales a lo largo del cilindro. El grado de anisotropía se controla con dos parámetros: distancia entre cilindros $d \in [0, \infty]$ y la altura de la superficie base $h \in [0, 1]$. Ambos parámetros carecen de unidades y pueden verse de manera gráfica en la figura 3.8. Si tomamos $d = 0$, este modelo se comporta como Torrance-Sparrow (comentado en la sección 3.4). El grado máximo de anisotropía —la variabilidad máxima entre las normales de los cilindros es de π — ocurre en la distancia $d = 2$. Justo al superar ese valor, aparece entre los cilindros una superficie base —cuya normal es constante— para participar como una componente más en la reflexión. Cuanto mayor sea la altura de la superficie base —controlado con el parámetro h — menor será la variación de las normales, lo que se suaviza la anisotropía.

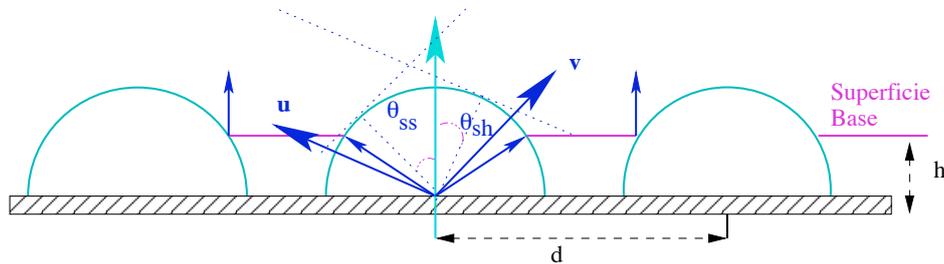


Figura 3.8: La anisotropía es controlable mediante dos parámetros.

La expresión principal de este modelo es la siguiente:

$$I_r = \frac{(I_a \times l_{vi}) + (I_f \times f_{vi})}{l_v + f_v}. \quad (3.8)$$

La cantidad de luz reflejada se calcula determinando la porción visible e iluminada del cilindro en el que incide la luz —notado I_a —, más la reflejada por la parte visible de la superficie base I_f , cuando ésta interviene. Poulin y Fournier recomiendan calcular I_f utilizando un modelo de iluminación sencillo y con \mathbf{u} y \mathbf{v} (visor y fuente de luz) constantes. Esta suposición es válida ya que el cilindro es muy pequeño en comparación con la posición de éstos. Para calcular I_a es necesario conocer las longitudes de la proyección del cilindro sobre la base para la parte visible del cilindro l_v , la parte visible e iluminada del cilindro l_{vi} , así como la parte visible e iluminada de la superficie base f_{vi} . La fórmula de la ecuación 3.8 tiene tres partes: la suma de dos productos —que reflejan las contribuciones a la intensidad difusa reflejada y la especular reflejada—, dividido por una suma.

Para considerar los efectos de ocultación de la luz —*shadowing* y *masking*— es necesario conocer los ángulos en donde el cilindro comienza a ocultarse a sí mismo desde la dirección de observación, θ_{hs} , o por otros cilindros θ_h . De manera similar calcularemos los ángulos en los que el cilindro bloquea la luz hacia el visor, por él (θ_{ss}), o por los que están a su lado θ_s .

La parte visible del cilindro viene dada por:

$$l_v = \frac{\sin(\theta_{\mathbf{h}} - \theta_{\mathbf{u}}) + \sin(\theta_{sh} - \theta_{\mathbf{u}})}{\cos(\theta_{\mathbf{u}})}.$$

La parte visible e iluminada del cilindro es:

$$l_{vi} = \begin{cases} \frac{\sin(\theta_{\mathbf{v}} - \theta_{\mathbf{u}}) + \sin(\theta_{\mathbf{u}} + \theta_{\mathbf{u}})}{\cos(\theta_{\mathbf{u}})} & \text{si } \mathbf{u}, \mathbf{v} \text{ en la misma cara que } \mathbf{n}, \\ \frac{\sin(\theta_{\mathbf{u}} - \theta_{\mathbf{u}}) + \sin(\theta_{\mathbf{v}} + \theta_{\mathbf{u}})}{\cos(\theta_{\mathbf{u}})} & \text{en otro caso.} \end{cases}$$

Sea f la longitud total del suelo, la porción visible o f_v , es la porción restante a la longitud oculta: $f - f_h$. De forma parecida la porción iluminada es la que no está en sombra: $f - f_s$. Ambos valores se calculan como:

$$\begin{aligned} f &= d - 2\sqrt{1 - h^2}, \\ f_h &= \min\left(\frac{1}{\cos(\theta_{\mathbf{u}})} - \sqrt{1 - h^2} - \left(\frac{h \sin(\theta_{\mathbf{u}})}{\cos(\theta_{\mathbf{u}})}\right), f\right), \\ f_s &= \min\left(\frac{1}{\cos(\theta_{\mathbf{v}})} - \sqrt{1 - h^2} - \left(\frac{h \sin(\theta_{\mathbf{v}})}{\cos(\theta_{\mathbf{v}})}\right), f\right). \end{aligned}$$

Utilizando la notación del formato GRF este modelo queda descrito como:

CILINDRICO-PF-BRDF ::=
`brdf` poulin `d =valor` `h =valor`
`n =valor` `ks =valor` `kd =valor`

Tabla 3.8: Notación en GRF para CILINDRICO-PF-BRDF

La ventaja de este modelo es que el conjunto de normales ya conocidas se pueden emplear en el cálculo de las direcciones para los rayos reflejados y refractados, ya que es posible emplear un modelo complementario de cilindros negativos, para simular, por ejemplo, el interior de un océano. Los diversos efectos de bloqueo de la luz (figura 3.3) se pueden tratar, considerando el precálculo de los ángulos para los que se produce.

Sin embargo, como inconvenientes de este modelo podemos decir que no se garantiza la conservación de la energía; la bondad de los resultados dependen de la medida en que se sepa dar correctamente los parámetros que caracterizan la superficie. A su vez, cada cilindro es individual, lo que impide que se consigan efectos como sombras arrojadas o reflexiones múltiples.

comportamiento; los mismos son los siguientes:

- Color cromático C o color de la superficie. Debe distinguirse del color especular C_s , o color del brillo, y que es calculado.
- Grado de suavidad: $s = 0$, cuando es difuso perfecto, y $s = 1$, para especulares perfectos. Este valor afecta a la ratio difuso / especular y modifica el tamaño del brillo.
- Grado de metalidad: $m = 1$, en superficies metálicas, y $m = 0$, para eliminar esta característica. Modifica el color especular y la apariencia difusa del objeto.
- Transparencia: $t = 0$, en superficies opacas, $t = 1$ en superficies totalmente transparentes. La transmitancia es la transparencia medida para la dirección incidente en la normal. Indica la cantidad de luz que pasa a través del material.
- Índice de refracción η : determina la dirección que tomará la luz al cambiar de medio.

El modelo de reflexión de Strauss emplea coeficientes de reflectividad difusa —medida de cómo de rugosa y opaca es una superficie— y especular, que hacen que varíen de forma lineal las componentes difusas y especulares de la BRDF:

$$f_r(\mathbf{u}, \mathbf{v}) = k_d f_{r,d}(\mathbf{u}, \mathbf{v}) + k_s f_{r,s}(\mathbf{u}, \mathbf{v}),$$

cada componente se expresa de la siguiente forma:

$$f_{r,d}(\mathbf{u}, \mathbf{v}) = (\mathbf{u} \cdot \mathbf{n}) d r_d C,$$

$$f_{r,s}(\mathbf{u}, \mathbf{v}) = r_s C_s.$$

A continuación detallamos cada uno de estos dos factores y otras funciones relacionadas:

$$\begin{aligned} r_d &= (1 - s^3)(1 - t) & d &= (1 - ms), \\ r_n &= (1 - t) - r_d & e &= \frac{3}{1-s}, \\ r_j &= \min[1, r_n + (r_n + k_j)j] & k_j &= 0,1, \\ r_s &= [-(\mathbf{h} \cdot \mathbf{v})]^e r_j & j &= F(\alpha) G(\alpha) G(\delta). \end{aligned}$$

El término r_j es un factor que intenta considerar los brillos especulares. Se basa en el factor de Fresnel y del término geométrico utilizado para los efectos de bloqueo de la luz. Cada uno es aproximado mediante una curva que emplea respectivamente las siguientes constantes: $k_f = 1,12$ y $k_g = 1,01$. La expresión se basa en el parámetro $x \in [0, 1]$ correspondiente al coseno del ángulo utilizado:

$$F(\theta) = \frac{\frac{1}{(x-k_f)^2} - \frac{1}{k_f^2}}{\frac{1}{(1-k_f)^2} - \frac{1}{k_f^2}},$$

$$G(\theta) = \frac{\frac{1}{(1-k_g)^2} - \frac{1}{(x-k_g)^2}}{\frac{1}{(1-k_g)^2} - \frac{1}{k_g^2}}.$$

Los valores de F y G pueden precalcularse para posteriormente ser consultados por tabla. De esta forma, operacionalmente Strauss es sólo un poco más costoso que Phong. Con respecto a la bondad del modelo, hay que decir que Cook-Torrance es mucho más preciso, sin embargo el atractivo de esta BRDF es su simplicidad.

En los no-metales, el color de la reflexión es el mismo que el de la fuente de luz. Sin embargo, en los metales deberemos distinguir el ángulo de incidencia, e interpolar entre el color de la superficie y el color especular blanco —notado como C_1 — que se obtiene, por medición, para un ángulo de $\pi/2$. Esto es:

$$C_s = \begin{cases} C_1 & \text{si no metal,} \\ C_1 + m(1 - F(\alpha))(C - C_1) & \text{si metal.} \end{cases}$$

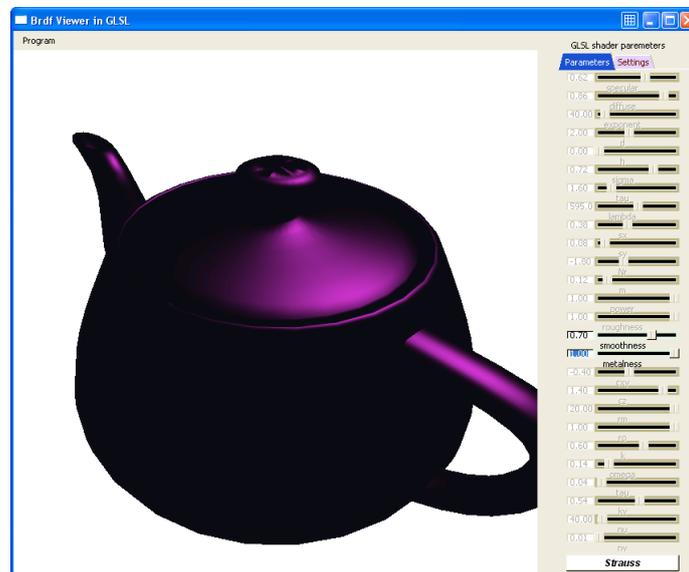


Figura 3.10: La BRDF de *Strauss* es visualizada desde nuestro editor GLSL de BRDFs.

Nuestro sistema de síntesis de imágenes considera únicamente superficies opacas, por lo que nuestra definición para la BRDF es como sigue:

STRAUSSBRDF ::=
brdf **strauss** *s* =valor *m* =valor
 ks =valor *kd* =valor

Tabla 3.9: Notación en GRF para STRAUSSBRDF

3.12. He-Torrance-Sillion-Greenberg (1991)

Otro de los modelos físicos es el de He *et al.* [HTSG91] que considera, además, otras propiedades tales como la polarización, difracción e interferencia. La BRDF es una función de varios parámetros, que en general pertenecen a dos grupos: parámetros geométricos y parámetros físicos. Los geométricos quedan ilustrados en la figura 2.2 y los físicos en la figura 3.11. En la primera se ven los ángulos polar y azimut para la dirección incidente (θ_i, ϕ_i) y la reflejada (θ_r, ϕ_r) , respectivamente, así como el ángulo sólido del haz incidente $d\omega_i$. Todos estos ángulos se miden con respecto a la normal de la superficie, como ya se ha dicho anteriormente. En el caso de superficies isótropas, la BRDF es rotacionalmente simétrica con respecto a la normal, y eso se traduce en una independencia al ángulo ϕ_i .

En cuanto a los parámetros físicos, la BRDF depende también de la longitud de onda λ de la luz incidente, así como de otros parámetros que caracterizan la superficie. Entre éstos está la rugosidad, dada por σ y τ , que especifican la altura y el espaciado de pequeños picos en la superficie, y su índice de refracción η , que también es función de λ .

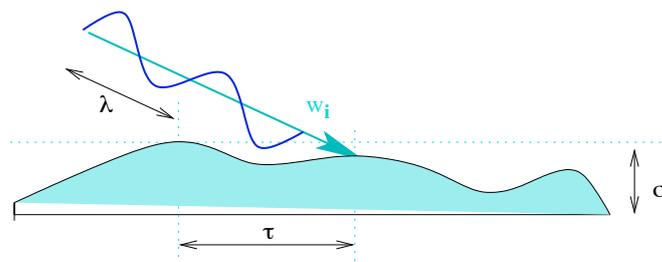


Figura 3.11: Aspectos ópticos de la incidencia de la luz en una superficie.

La característica principal de este modelo isótropo es la distinción de tres tipos de reflexión, ya que hasta ese entonces los modelos como el de Phong o Blinn, sólo consideraban dos, el difuso y el especular. La BRDF se divide en:

$$f_r = f_r^{sp} + f_r^{ud} + f_r^{dd},$$

donde:

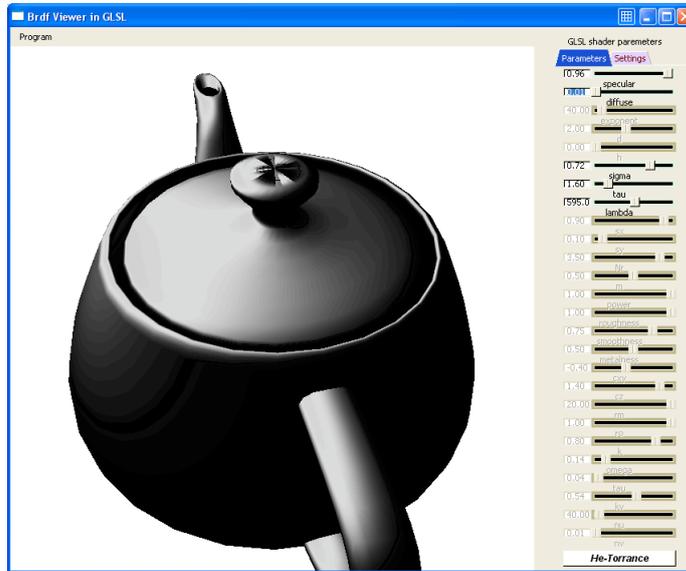


Figura 3.12: La BRDF de *He-Torrance* se edita con un *software* GLSL que implementa las BRDFs.

3.13. Ward (1992)

En su trabajo Ward [War92] presenta un nuevo dispositivo para medir la distribución espacial de reflectancia de las superficies a estudiar, acompañado de un modelo matemático anisótropo de la reflectancia.

Este modelo de reflexión se basa en un lóbulo gaussiano y puede ser clasificado como experimental en el sentido de que no hace una justificación teórica, ni aproximada. La intención de este trabajo radica en conseguir un modelo simple en cuanto a los parámetros utilizados, de forma que se ajusten lo mejor posible a los resultados experimentales medidos, minimizando el error.

Aunque existen modelos isótropos de reflexión, son muy pocos los publicados por esas fechas que puedan considerar el caso más general de la anisotropía. Entre éstos están los ya comentados de Kajiya y Poulin-Fournier (secciones 3.9 y 3.10 respectivamente). Ward comienza con una expresión matemática isotrópica para su modelo:

$$f_{r,iso}(\mathbf{u}, \mathbf{v}) = \frac{k_d}{\pi} + \frac{k_s}{\sqrt{\cos(\mathbf{u}) \cos(\mathbf{v})}} \frac{\exp\left[-\tan^2\left(\frac{\delta}{\alpha_m}\right)\right]}{4\pi\alpha_m^2}, \quad (3.9)$$

donde δ es el ángulo formado por la dirección normal y el vector normalizado *halfway* $\mathbf{h} = (\theta_{\mathbf{h}}, \phi_{\mathbf{h}})$, y α_m es la desviación típica que caracteriza el lóbulo gaussiano.

Se incluye un factor de normalización de la BRDF, $\frac{1}{4\pi\alpha_m^2}$, ya que cuando el parámetro α_m toma valores por encima de 0.2 —momento en el cual la superficie deja de ser prácticamente difusa— los que se devuelven son excesivamente altos y producen efectos no deseados en las imágenes resultantes.

El caso anisótropo se desarrolla empleando una distribución gaussiana elíptica, definida con dos direcciones, \mathbf{x} e \mathbf{y} . Tenemos, por tanto, tres parámetros para controlar el lóbulo gaussiano que representa la componente *glossy*: k_s para la magnitud, α_x y α_y para proporcionar el ancho en las direcciones de los ejes anisotrópicos. El valor k_d corresponde a la magnitud de la componente difusa. Empleando nuestra notación en GRF para esta BRDF, los símbolos varían ligeramente:

WARDBRDF ::=
brdf ward sx =valor sy =valor ks =valor kd =valor

Tabla 3.11: Notación en GRF para WARDBRDF

Si al definir una instancia usamos sx=sy, se tiene el modelo isotrópico. Se pueden ver las distinciones de ambos modelos de Ward en la figura 3.13. Finalmente damos la expresión anisotrópica de Ward:

$$f_{r,ani}(\mathbf{u}, \mathbf{v}) = \frac{k_d}{\pi} + \frac{k_s}{4\pi \alpha_x \alpha_y \sqrt{\cos(\mathbf{u}) \cos(\mathbf{v})}} e^{\left[-\tan^2(\theta_{\mathbf{h}}) \left(\frac{\cos^2(\phi_{\mathbf{h}})}{\alpha_x^2} + \frac{\sin^2(\phi_{\mathbf{h}})}{\alpha_y^2} \right) \right]}.$$

A pesar de ser ésta la expresión completa, Ward propone una versión aproximada orientada a su implementación:

$$f_{r,ani}(\mathbf{u}, \mathbf{v}) = \frac{k_d}{\pi} + k_s \frac{1}{4\pi \alpha_x \alpha_y \sqrt{\cos(\mathbf{u}) \cos(\mathbf{v})}} e^{\left[-\frac{\left[\frac{(\mathbf{h} \cdot \mathbf{x})}{\alpha_x} \right]^2 + \left[\frac{(\mathbf{h} \cdot \mathbf{y})}{\alpha_y} \right]^2}{(\mathbf{h} \cdot \mathbf{n})^2} \right]}, \quad (3.10)$$

donde:

$$(\mathbf{h} \cdot \mathbf{x}) = \frac{\sin(\theta_{\mathbf{u}}) \cos(\phi_{\mathbf{u}}) + \sin(\theta_{\mathbf{v}}) \cos(\phi_{\mathbf{v}})}{|\mathbf{h}|},$$

$$(\mathbf{h} \cdot \mathbf{y}) = \frac{\sin(\theta_{\mathbf{u}}) \sin(\phi_{\mathbf{u}}) + \sin(\theta_{\mathbf{v}}) \sin(\phi_{\mathbf{v}})}{|\mathbf{h}|},$$

$$(\mathbf{h} \cdot \mathbf{n}) = \frac{\cos(\theta_{\mathbf{u}}) + \cos(\theta_{\mathbf{v}})}{|\mathbf{h}|},$$

$$|\mathbf{h}| = \left[2 + 2 \sin(\theta_{\mathbf{u}}) \sin(\theta_{\mathbf{v}}) (\cos(\phi_{\mathbf{u}}) \cos(\phi_{\mathbf{v}}) + \sin(\phi_{\mathbf{u}}) \sin(\phi_{\mathbf{v}})) + 2 \cos(\theta_{\mathbf{u}}) \cos(\theta_{\mathbf{v}}) \right]^{\frac{1}{2}}.$$

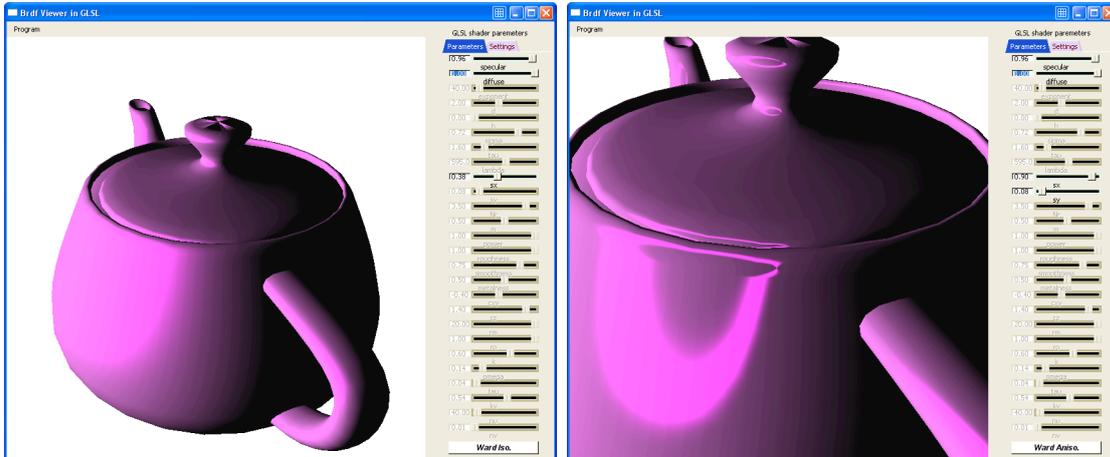


Figura 3.13: La BRDF de *Ward* se visualiza según sus parámetros. A la izquierda, una instancia del modelo isotrópico. A la derecha la BRDF es anisotrópica.

La principal aportación de este modelo está relacionada con la distribución empleada y por el factor de normalización, que asegura que sea fácilmente integrable sobre la semiesfera.

3.14. Westin (1992)

Tras los desarrollos de modelos físicos de reflexión de la luz [CT82, HTSG91, Kaj85, Pho75], Westin *et al.* [WAT92] se centraron en encontrar un modelo de reflexión rápido y con resultados aceptables. Al igual que el modelo de He *et al.* [HTSG91], la BRDF es dividida en tres componentes: la especular, difusa direccional y difusa uniforme:

$$f_r = f_r^{sp} + f_r^{ud} + f_r^{dd}$$

La componente especular es la responsable de generar un comportamiento reflexivo similar al producido en un espejo, esto es, un brillo intenso en una dirección preferente. En este modelo se expresa con el siguiente producto:

$$f_r^{sp}(\mathbf{u}, \mathbf{v}) = |F(\theta_{\mathbf{u}})|^2 e^{-g(\sigma, \lambda)} S(\tau),$$

donde F es el término de *Fresnel* para la reflectividad, que depende del ángulo incidente y del índice de refracción η del material. La rugosidad de la superficie se caracteriza con el término g , y S es una función geométrica de sombreado. Este último valor es similar a un test de oclusión, su valor depende de τ y su pseudocódigo se encuentra en [HTSG91]. La función g se define como sigue:

$$g(\sigma, \lambda) = \left(\frac{2\pi\sigma}{\lambda} (\cos(\mathbf{r}_{\mathbf{u}}) + \cos(\mathbf{v})) \right)^2.$$

Es posible simplificar y tomar $\cos(\mathbf{v}) = \cos(\mathbf{r}_{\mathbf{u}})$, con lo que la expresión de g queda reducida a:

$$g(\sigma, \lambda) = \left(\frac{4\pi\sigma \cos(\mathbf{v})}{\lambda} \right)^2$$

Una superficie parecerá más lisa cuanto más se acerque la dirección incidente al horizonte, ya que conforme se aproxima θ_i a 90° , el valor $\cos\theta_i$ tiende a 0 y, consecuentemente, el valor g es menor.

La componente difusa direccional representa los casos de reflexión más allá de los ideales, en los que existe una dirección dominante. Por ello su carácter es no uniforme, siendo la responsable de producir brillos más pronunciados. Físicamente corresponde a los eventos de difracción y dispersión. La expresión de esta BRDF es:

$$f_r^{dd}(\mathbf{u}, \mathbf{v}) = \frac{|F(\theta_{\mathbf{u}})|^2 G(\mathbf{u}, \mathbf{v}) S(\tau) D(\mathbf{u}, \mathbf{v})}{\pi \cos(\theta_{\mathbf{u}})} \quad (3.11)$$

El término anterior representa la extensión homogénea que se produce sobre la semi-esfera situada sobre la superficie que se comporta de forma difusa. Interviene el término geométrico G de Torrance-Sparrow [CT82], que varía con el ángulo de incidencia θ_i . La función de distribución D se representa con una sumatoria infinita:

$$D(\mathbf{u}, \mathbf{v}) = \frac{\pi^2 \tau^2}{4\lambda^2} \sum_{m=1}^{\infty} \frac{g(\sigma, \lambda)^m e^{-g(\sigma, \lambda)}}{m! m} e^{-\frac{h(\tau, \lambda, \mathbf{u}, \mathbf{v})}{4m}}$$

donde h es una función que depende del parámetro de la superficie τ , de la longitud de onda λ y de los ángulos incidentes y reflejados:

$$h(\tau, \lambda, \mathbf{u}, \mathbf{v}) = \left(\frac{2\pi\tau}{\lambda} \right)^2 ((\sin(\theta_{\mathbf{v}}) \cos(\phi_{\mathbf{u}}) - \sin(\theta_{\mathbf{v}}))^2 + (\sin(\theta_{\mathbf{u}}) \sin(\phi_{\mathbf{u}}))^2)$$

3.15. Schlick (1993)

El modelo propuesto por Schlick [Sch93, Sch94b] intenta fundamentalmente dar una BRDF *plausible* (conservación de la energía y reciprocidad), al mismo tiempo que incorpora la teoría de *microfacets* y el uso de las ecuaciones de Fresnel.

En otros modelos de reflexión ya se vio que separar la BRDF en partes difusa y especular, usando una combinación lineal, no es del todo correcto. Esto es así porque

son componentes que no proceden de un mismo fenómeno físico. Así, en un material como el plástico, la reflexión en la superficie es especular y en las capas interiores —las que se denominan *subfaces*— la reflexión es difusa. El plástico es un ejemplo de lo que se conoce como materiales heterogéneos. Según Shirley [SHSL97] las proporciones difusas y especulares no son constantes sino función del ángulo incidente. En el caso de materiales homogéneos —como los metales— no es necesario distinguir entre componente difusa y especular.

Schlick propone dos BRDFs, una *single*, para las superficies homogéneas y otra *double*, para las heterogéneas, considerando, a su vez comportamientos isótropos y anisótropos.

Una superficie sencilla —*single*— se define como una tripleta (C_λ, r, p) . Una superficie compuesta —*double*— se define como dos tripletas *single*: (C_λ, r, p) y (C'_λ, r', p') . Para ambos casos se tiene:

- C_λ es el factor de reflexión para λ , que puede tomar valores entre 0 y 1.
- $r \in [0, 1]$ define la rugosidad. Cuando $r = 0$ estamos en el caso de materiales especulares perfectos; si $r = 1$ tenemos los difusos perfectos.
- $p \in [0, 1]$ caracteriza la isotropía del material. Cuando $p = 0$ la superficie es anisótropa perfecta y si $p = 1$ es isótropa perfecta.

En la formulación de este modelo de reflexión intervienen diversas funciones, que tomaran los parámetros de una tripleta u otra, dependiendo del tipo de material. Por ejemplo $S(u)$ se calcula usando C_λ y $S'(u)$ hace lo mismo con C'_λ . Las funciones necesarias corresponden a:

- comportamiento espectral, o aproximación computacionalmente eficiente al término de Fresnel:

$$S(u) = C_\lambda + (1 - C_\lambda)(1 - u),$$

- comportamiento direccional, que considera fenómenos de emisión y reemisión:

$$D(t, v, v', w) = \frac{G(v)G(v')}{4\pi v v'} Z(t) A(w) + \frac{(1 - G(v)G(v'))}{4\pi v v'},$$

- factor geométrico: $G(v)$

$$G(v) = \frac{v}{\sqrt{r - rv + v}},$$

- dependencia en el parámetro zenith: $Z(t)$

$$Z(t) = \frac{r}{\sqrt{1 + rt^2 - t^2}}$$

- dependencia en el parámetro azimuth:

$$A(w) = \sqrt{\frac{p}{p^2 - p^2 w^2 + w^2}}$$

- reflectancia espectral, cuando la superficie es sencilla:

$$R(t, u, v, v', w) = S(u) D(t, v, v', w),$$

de la misma forma para superficies compuestas:

$$R(t, u, v, v', w) = S(u) D(t, v, v', w) + [1 - S'(u)] S'(u) D'(t, v, v', w).$$

Finalmente la BRDF sustituye los parámetros (t, u, v, v', w) de la reflectancia espectral con los siguientes ángulos:

$$f_r(\mathbf{u}, \mathbf{v}) = R(\mathbf{n} \cdot \mathbf{h}, (\mathbf{u} \cdot \mathbf{h}), (\mathbf{v} \cdot \mathbf{n}), (\mathbf{u} \cdot \mathbf{n}), \phi_{\mathbf{h}}). \quad (3.12)$$

En notación GRF este modelo tiene cuatro parámetros obligatorios y tres opcionales que deben ser proporcionados siempre que el cuarto parámetro tome un valor verdadero.

SCHLICKBRDF ::=
brdf schlick SC =*valor* Sr =*valor*
Sp =*valor* Ly =*valor-lógico*
[DC =*valor* Dr =*valor* Dp =*valor*]

Tabla 3.12: Notación en GRF para SCHLICKBRDF

3.16. Lewis (1993)

En un artículo, Lewis [Lew93] hace un estudio de los modelos de iluminación de la época, como Phong, Blinn y Torrance-Sparrow, dando una formulación alternativa, que permite la plausibilidad de éstos.

Los modelos basados en un lóbulo coseno, como los de Phong y Blinn, se diferencian tan sólo en la función de distribución de los *microfacets* D . Esta distribución debe ser normalizada para asegurar que el modelo es físicamente plausible y, por tanto, válido para su utilización en algoritmos de iluminación. Por normalizar la distribución entendemos que si sumamos todas las áreas proyectadas de los *microfacets* en un área

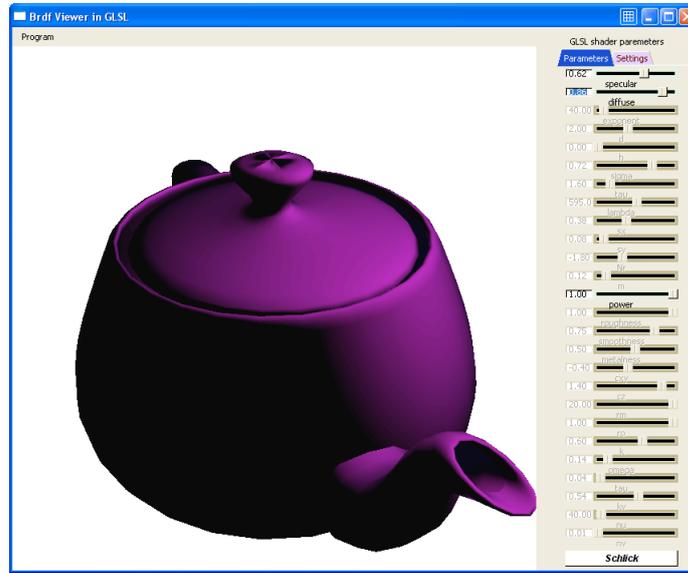


Figura 3.14: La BRDF de *Schlick* se visualiza desde nuestro editor GLSL de BRDFs.

dA , esta suma debe ser igual a dA . Si asumimos la conservación de la energía, se tiene que:

$$\int_S D(\mathbf{h}) \cos(\theta_{\mathbf{h}}) dA(\mathbf{h}) = \int_{\Omega} D(\mathbf{h}) \cos(\theta_{\mathbf{h}}) d\sigma(\mathbf{h}) = 1$$

Desarrollando esta condición tenemos, para el caso de Phong, con $D(\mathbf{u}, \mathbf{v}) = (\mathbf{u} \cdot \mathbf{r}_{\mathbf{v}})^n$:

$$\begin{aligned} \int_{\Omega} c (\mathbf{u} \cdot \mathbf{r}_{\mathbf{v}})^n \cos(\theta_{\mathbf{u}}) d\sigma(\mathbf{u}) &= \int_0^{2\pi} \int_0^{\pi/2} c (\mathbf{u} \cdot \mathbf{r}_{\mathbf{v}})^{n+1} \sin(\theta_{\mathbf{u}}) d(\theta_{\mathbf{u}}) d(\phi_{\mathbf{u}}), \\ &= 2 c \pi \int_0^1 u^{n+1} du, \\ &= 2 c \pi \left. \frac{u^{n+2}}{n+2} \right]_0^1 \\ &= \frac{2 c \pi}{n+2} = 1. \end{aligned}$$

De aquí despejamos c como factor de normalización de Phong:

$$c = \frac{n+2}{2\pi}. \quad (3.13)$$

De igual manera el uso del factor de normalización en el modelo de Blinn, visto en la sección 3.7, nos asegura la conservación de la energía. Se expresa ahora de la

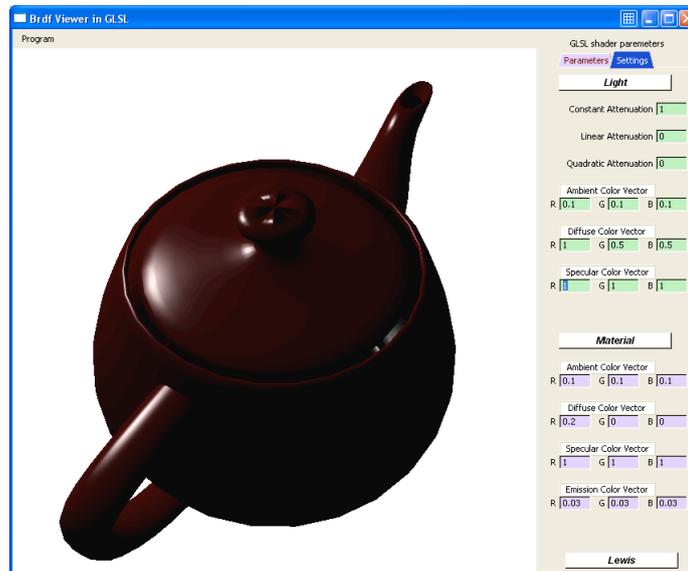


Figura 3.15: La BRDF de *Lewis* desde nuestro editor GLSL de BRDFs.

siguiente manera:

$$f_r(\mathbf{u}, \mathbf{v}) = k_d \cos(\mathbf{u}) + k_s \frac{n+2}{2\pi} \cos(\delta)^n. \quad (3.14)$$

Cualitativamente podemos ver como resulta la normalización en la figura 3.15. La definición de esta BRDF es muy similar a la de Phong y Blinn, y nos permite hacer un uso normalizado de ellos en nuestro sistema de síntesis con la siguiente expresión:

LEWISBRDF ::= *brdf* lewis n =valor ks =valor kd =valor

Tabla 3.13: Notación en GRF para LEWISBRDF

3.17. Hanrahan (1993)

En el trabajo de Hanrahan y Krueger [HK93] se presenta un modelo de propagación de la luz para superficies formadas por capas, en términos de la teoría de transporte lineal unidimensional. Los autores utilizan principalmente el modelo de reflexión físico para la simulación de las capas subcutáneas de la piel. Hanrahan y Krueger distinguen dos capas básicas epidérmicas: una que contiene una serie de reflectores y otra con propiedades de absorción. Ambas hacen uso de la dependencia en la longitud de onda de la reflexión en los pigmentos que las forman.

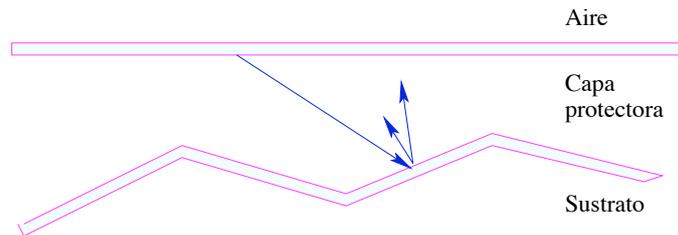


Figura 3.16: Modelo de sustratos para las superficies heterogéneas.

Como en la mayoría de los materiales modelados hasta la fecha, en la reflexión de la luz interviene un término especular y otro difuso. El primero suele modelarse considerando que la superficie está compuesta de pequeños reflectores perfectos o *microfacets*, y el segundo considera todas las reflexiones producidas en el interior y en la capa más externa de la superficie.

Se ha utilizado, por lo general, la *ley del coseno* de Lambert para calcular el término difuso; esta ley considera que la radiancia saliente es isotrópica y proporcional a la irradiancia en la superficie, que depende del coseno del ángulo de incidencia. Sin embargo, la reflexión difusa puede ser explicada por la propagación de la luz a lo largo de las capas internas del material. Cuando la luz entra en el material, la misma puede ser absorbida o reflejada en una dirección aleatoria un número elevado de veces y, eventualmente, podrá abandonar el material.

En este modelo las contribuciones de la reflexión en la superficie y en el resto de las capas internas son muy sensibles al término de Fresnel. Además, la reflectancia considerada en las subcapas no es necesariamente isotrópica, pero varía con la dirección y predominan las reflexiones hacia adelante frente a aquéllas que retroceden en la dirección de reflexión.

El valor de la radiancia que se refleja desde la superficie proviene de dos componentes: la reflexión producida en la parte externa de la superficie y la generada por todo el volumen de la superficie.

$$L_r(\theta_r, \phi_r) = L_{r,s}(\theta_r, \phi_r) + L_{r,v}(\theta_r, \phi_r).$$

Este modelo también calcula la transmisión de luz entre las diferentes capas. La radiancia transmitida tiene dos componentes: la primera se denomina *intensidad reducida* y es la cantidad de luz incidente que se transmite a través de la capa y que sólo puede llegar a producir absorción. La segunda componente es la reflexión producida en el volumen de la capa.

$$L_t(\theta_t, \phi_t) = L_{r,i}(\theta_t, \phi_t) + L_{t,v}(\theta_t, \phi_t).$$

En estos dos fenómenos físicos separados —reflexión y transmisión de la luz— intervienen la BRDF y la BTDF. Estas funciones de distribución poseen dos componentes:

$$f_r = f_{r,s} + f_{r,v},$$

$$f_t = f_{r,i} + f_{t,v}.$$

Si se asumen superficies planas, interviene entonces el clásico factor de Fresnel F , que depende de los índices de refracción η_i y η_t , así como del coseno de los ángulos θ_i y θ_t . El factor de Fresnel interviene en la BRDF en la medida en que cuanto menor sea su valor, más luz entrará a la superficie, por lo que mayor será la reflexión producida por las subcapas.

Más detalles, sobre la aproximación de primer orden de las fórmulas derivadas con la teoría de transporte lineal unidimensional, pueden encontrarse en el propio artículo de Hanrahan y Krueger [HK93].

3.18. Oren-Nayar (1994)

El modelo de reflexión propuesto por Michael Oren y Shree Nayar en 1994 supuso una mejora al clásico modelo lambertiano para materiales difusos. Utiliza la misma conceptualización de superficie rugosa de Cook-Torrance, basado en cavidades simétricas en forma de V que fue primeramente ideado por Torrance & Sparrow (véase apartado 3.4), con la salvedad de que, los *microfacets* son superficies difusas perfectas. Es importante señalar que, además de considerar efectos conocidos como *masking* y *shadowing*, las interreflexiones son modeladas mediante la simulación de una segunda reflexión en cada *microfacet*. Como se vio en el apartado anterior, la reflexión de la luz producida en una superficie puede ser clasificada en dos categorías: *reflectancia en la superficie*, que tiene lugar en la interfaz entre dos medios con índices de refracción diferentes y *reflectancia interior*, que se produce en las subcapas internas del objeto y que suele, en la mayoría de los casos, ser considerada como lambertiana.

El modelo descrito por Oren-Nayar [ON94, ON95] puede ser utilizado para modelar materiales que no son del todo difusos, sino que presentan brillos cuando la dirección de iluminación se aproxima a la de visión. Este es el caso de algunos tejidos, la arena y algunos plásticos.

Inicialmente se tiene concavidades caracterizadas por su orientación, la normal \mathbf{a} . Seguidamente se considera que el área de una concavidad $d(\mathbf{a})$ es muy superior a la

longitud de onda del haz incidente λ , y éste a su vez es menor que el área total de la superficie dA , esto es:

$$\lambda^2 \ll d(\mathbf{a}) \ll dA,$$

obteniéndose posteriormente un valor representativo de la fracción de área que es ocupada por las concavidades orientadas hacia \mathbf{a} , notado como $P(\mathbf{a})$ y cuya expresión es:

$$P(\mathbf{a}) = N(\mathbf{a}) \cos(\mathbf{a}) d(\mathbf{a}),$$

donde $\cos(\mathbf{a}) d(\mathbf{a})$ es el área proyectada en la superficie utilizada (recuérdese la figura 2.4) para calcular la contribución de cada *microfacet* a la radiancia de la superficie. La función de distribución que modela la superficie es una gaussiana dependiente de la desviación típica σ_m de los ángulos que orientan las concavidades.

Dada una dirección incidente, la radiancia total de la superficie para una dirección de salida se obtiene integrando sobre todas las concavidades existentes sobre la misma, considerando únicamente dos eventos de reflexión. Es por ello que la BRDF se basa en el cálculo de la radiancia proyectada L_p :

$$f_r(\mathbf{u}, \mathbf{v}) = L_p(\mathbf{u}, \mathbf{v}, \mathbf{a}) P(\mathbf{a})$$

siendo $L_p(\mathbf{u}, \mathbf{v}, \mathbf{a}) = L_p^1(\mathbf{u}, \mathbf{v}, \mathbf{a}) + L_p^2(x, \mathbf{u}, \mathbf{v}, \mathbf{a})$. El primero de ellos, la radiancia reflejada proyectada, es originada al considerar un evento de reflexión en la superficie. Se calcula como:

$$L_p^1(\mathbf{u}, \mathbf{v}, \mathbf{a}) = \frac{\rho}{\pi} E_0 \frac{(\mathbf{u} \cdot \mathbf{a})}{(\mathbf{a} \cdot \mathbf{n})} \frac{(\mathbf{v} \cdot \mathbf{a})}{(\mathbf{a} \cdot \mathbf{n})} G(\mathbf{u}, \mathbf{v}, \mathbf{a}),$$

donde el término E_0 denota la irradiancia procedente de la dirección normal. El factor de atenuación geométrica se obtiene a partir de la siguiente expresión:

$$G(\mathbf{u}, \mathbf{v}, \mathbf{a}) = \min \left[1, \max \left(0, \frac{2 (\mathbf{u} \cdot \mathbf{n}) (\mathbf{a} \cdot \mathbf{n})}{(\mathbf{u} \cdot \mathbf{a})}, \frac{2 (\mathbf{v} \cdot \mathbf{n}) (\mathbf{a} \cdot \mathbf{n})}{(\mathbf{v} \cdot \mathbf{a})} \right) \right].$$

La radiancia reflejada proyectada, originada por dos eventos de reflexión interna, considera el ancho de la concavidad. Al suponer que la longitud l es mucho mayor que su ancho w , se puede considerar unidimensional. Los límites de la integral consideran los efectos de *masking* y *shadowing*: m_v es el ancho de la porción visible de la concavidad y m_s es el ancho de la porción iluminada de la concavidad adjacente. Finalmente queda la siguiente expresión:

$$L_p^2(\mathbf{u}, \mathbf{v}, \mathbf{a}) = \frac{\rho}{\pi} \int_{t=\frac{m_v}{w}}^1 \int_{r=\frac{m_s}{w}}^1 K(t, r, \mathbf{a}) d(r) d(t),$$

siendo aquí K el *kernel* de simetría transaccional para las dos distancias más cortas, x e y , resultantes de la intersección de dos concavidades.

$$K(x, y, \mathbf{a}) = \frac{\pi \sin^2(2\theta_{\mathbf{a}})}{2} \frac{xy}{(x^2 + 2xy \cos(2\theta_{\mathbf{a}}) + y^2)^{\frac{3}{2}}}$$

El modelo propuesto considera las reflexiones producidas en la superficie, que tratan como un caso isotrópico, así como las reflexiones producidas en el interior del cuerpo sólido, considerando a éste como anisotrópico. Sin embargo, tan sólo se consideran dos reflexiones internas, descartando el resto. Esta simplificación no se ajusta con lo sucedido en la realidad dentro del objeto.

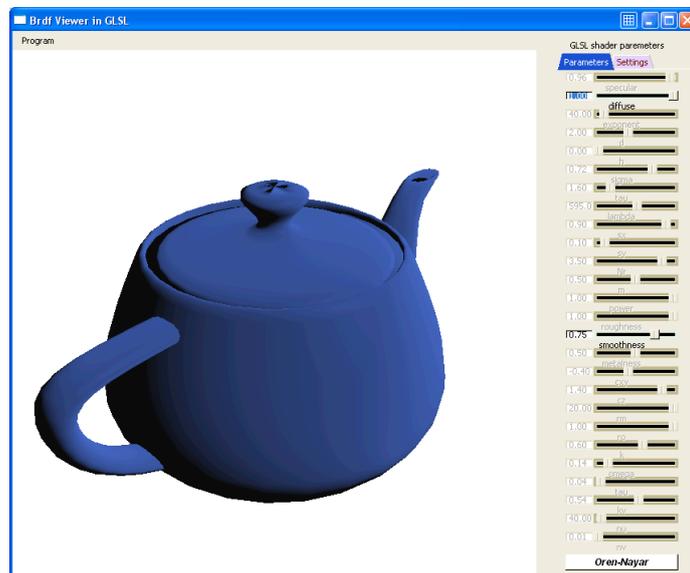


Figura 3.17: La BRDF de *oren-nayar* es visualizada en tiempo real usando una instancia GLSL de la BRDF.

Este modelo se define dando dos parámetros rugosidad $s \in [0, 1]$ —correspondiente a σ — y reflectividad $r \in [0, 1]$ —el equivalente a ρ en la formulación— y se describen como:

OREN-NAYAR-BRDF ::= *brdf* oren s =*valor* r =*valor*

Tabla 3.14: Notación en GRF para OREN-NAYAR-BRDF

3.19. Neumann-Neumann (1996)

Neumann & Neumann [NN96] presentan un modelo de BRDF plausible matemáticamente, fácil de aplicar y, que incluye los casos ideales. La definición resultante es

simple y, adecuada al muestreo en algoritmos de Monte-Carlo. Visualmente es bueno para simular materiales muy diversos, como metales, plásticos, cerámica, objetos retroreflectivos, así como materiales anisotrópicos.

En primer lugar define un modelo básico sobre el que se basan modelos más complejos, como los necesarios para los materiales anteriormente citados. Se trabaja en el plano de proyección de la semiesfera de direcciones, utilizando la medida σ_p , por lo que esta BRDF ya lleva implícito un término coseno. Es necesario disponer de una función distancia entre dos vectores del plano de proyección. Nótese que el subíndice p en los vectores indica proyección.

$$m(\mathbf{u}, \mathbf{v}) = |\mathbf{v}_p - \mathbf{r}_{u,p}| = |\mathbf{h} - (\mathbf{n} \cdot \mathbf{h})\mathbf{n}| \text{ con } \mathbf{h} = \mathbf{u} + \mathbf{v}.$$

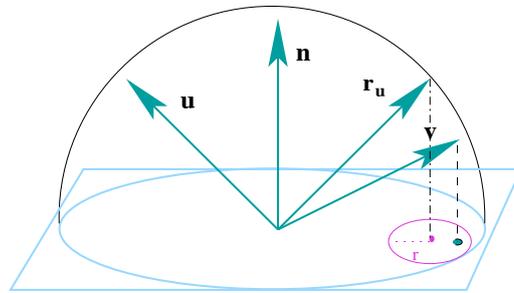


Figura 3.18: La BRDF de Neumann *et al.* se representa mediante discos de radio parametrizable en el dominio \mathcal{D}^2 .

Para la proyección del vector reflejado se toma un disco C cuyo radio de acción es r , de forma que valor de la BDDF es constante si se cumple:

$$f_r(\mathbf{u}, \mathbf{v}) = \begin{cases} \pi/r & \text{si } \mathbf{v}_p \in C(\mathbf{r}_{u,p}, r), \\ 0 & \text{en caso contrario.} \end{cases}$$

El parámetro $0 < r \leq 2$ distingue el caso especular perfecto ($r = 0$) del difuso perfecto ($r = 2$). Así mismo, permite un sencillo cálculo de la *reflectancia direccional-hemisférica*, ya que:

$$\rho(\mathbf{u}) = r \cdot \text{Area}(C(\mathbf{r}_{u,p}, r)).$$

Con esta base, se puede definir otras BRDFs. Podemos dar un modelo retroreflectivo, utilizando en la función distancia el vector $\mathbf{g} = \mathbf{u} - \mathbf{v}$ en lugar del vector \mathbf{h} , consiguiendo de esta forma, reflejar la luz preferentemente en la dirección de entrada. Un modelo anisotrópico tomará elipses E —con ejes a y b — en lugar de círculos, y utilizará la norma elíptica para calcular la distancia entre dos vectores proyectados:

$$f_{r,aniso}(\mathbf{u}, \mathbf{v}) = \begin{cases} b & \text{si } \mathbf{v}_p \in E(\mathbf{r}_{u,p}, a, b), \\ 0 & \text{en caso contrario.} \end{cases}$$

3.20. Lafortune (1997)

Una forma común en la obtención de imágenes realistas, ha sido la utilización de datos procedentes de mediciones, como lo hace Ward. Sin embargo origina un problema de almacenamiento debido a la gran cantidad de datos que son necesarios. Este modelo surge, por tanto, ante la necesidad de ajustar datos medidos utilizando uno simple y definible con pocos parámetros. Lo cierto es que en el caso isotrópico se consiguen excelentes resultados [LFTG97]. Se trata sencillamente de una extensión de la BRDF de Phong, que cumple que es físicamente plausible: recíproca y a su vez conservadora de la energía.

Tomemos $R(\mathbf{u}, \mathbf{n})$ como el operador de reflexión de un vector con respecto a la normal, que resulta de la operación $\mathbf{r} = \mathbf{u} - 2(\mathbf{u} \cdot \mathbf{n}) \mathbf{n}$. Aplicándolo tenemos la BRDF modificada de Phong, que es recíproca:

$$f_r(\mathbf{u}, \mathbf{v}) = (\mathbf{u} \cdot R(\mathbf{v}, \mathbf{n}))^n = (\mathbf{v} \cdot R(\mathbf{u}, \mathbf{n}))^n. \quad (3.15)$$

En un sistema de coordenadas canónico el operador de reflexión es equivalente a $R(\mathbf{u}, \mathbf{n}) = (-x_{\mathbf{u}}, -y_{\mathbf{u}}, z_{\mathbf{u}})$. Este caso correspondiente a Phong puede generalizarse usando una matriz M para modificar el vector original \mathbf{u} .

$$f_r \propto (\mathbf{v} \cdot M\mathbf{u})^n$$

Si usamos los valores de la diagonal de M como un vector de orientación \mathbf{o} , el factor de escalado $(-1, -1, 1)$ nos deja el caso de la BRDF de Phong normalizada. Para simular otros comportamientos, como el retroreflexivo, con tan sólo habría que tomar la orientación $(1, 1, 1)$.

La BRDF de Lafortune es la suma de N lóbulos coseno, cada uno con una orientación distinta, más un término difuso:

$$f_r(\mathbf{u}, \mathbf{v}) = \frac{\rho_d}{\pi} + \sum_{i=1}^N (\mathbf{v} \cdot (x_{\mathbf{u}}x_{\mathbf{o}_i}, y_{\mathbf{u}}y_{\mathbf{o}_i}, z_{\mathbf{u}}z_{\mathbf{o}_i}))^{n_i}. \quad (3.16)$$

Si queremos expresar este modelo para un único lóbulo utilizando la notación GRF, usaremos los parámetros $k_p = \rho_d$ y $(C_x, C_y, C_z) = \mathbf{o}$:

LAFORTUNEBRDF ::=
brdf lafortune n =*valor* kp =*valor*
 Cx =*valor* Cy =*valor* Cz =*valor*

Tabla 3.15: Notación en GRF para LAFORTUNEBRDF

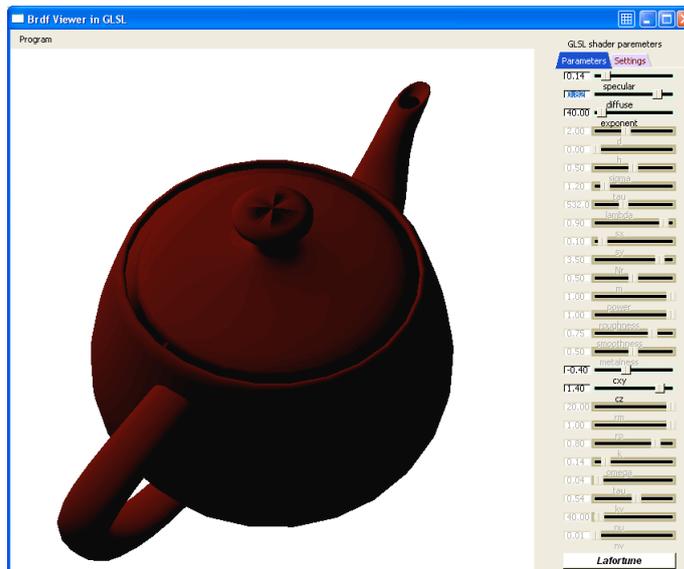


Figura 3.19: La BRDF *Lafortune* desde nuestro editor GLSL de BRDFs.

3.21. Shirley (1997)

En Shirley *et al.* [SHSL97] se presenta un modelo especializado en la simulación de superficies mates. Consideran que el resto de modelos de reflexión de la época no dan buenos resultados por intentar simular un amplio abanico de materiales. Los autores utilizan un coeficiente especular derivado de la formulación de Fresnel, que depende, al igual que el coeficiente mate, del ángulo de incidencia. Además de ser un modelo físico es recíproco y conserva la energía. Se le denomina *coupled model* por ser el primero en combinar de forma balanceada los eventos difusos y especulares.

Se tiene que la reflexión especular es causada por una supercie lisa de un material dieléctrico en contacto con el aire; la luz que no es reflejada especularmente se transmite dando lugar a la componente mate. Como en He *et al.* [HTSG91] se espera que la componente mate sea casi constante y con comportamiento isotrópico. La BRDF combina la parte especular y mate de la siguiente manera:

$$f_r(\mathbf{u}, \mathbf{v}) = F(\theta_{\mathbf{u}}) + k R_m f(\theta_{\mathbf{u}}) f(\theta_{\mathbf{v}}),$$

donde k es la constante necesaria para la restricción de la conservación de la enería,

3.22. Ashikhmin-Shirley (2000)

Ashikhmin y Shirley [AS00a] proponen un modelo anisotrópico de *microfacets* que busca destacar con respecto a otras BRDFs por la sencillez de sus parámetros, la eficiencia de cálculo y el hecho de disponer de un método de muestreo apto para algoritmos basados en Monte-Carlo.

Consiguen su objetivo mediante el uso de dos parámetros que controlan la función de distribución de los vectores \mathbf{h} de los *microfacets*: e_x proporciona un exponente que orienta \mathbf{h} a lo largo del eje X de una elipse, e_y hace lo mismo para la orientación en el eje Y (puede verse más claro en la figura 3.21). El exponente e se calcula encontrando el radio de la elipse dado el ángulo ϕ de orientación. La expresión de la distribución empleada es la siguiente:

$$D(\mathbf{h}) = \sqrt{(e_x + 1)(e_y + 1)} (\mathbf{h} \cdot \mathbf{n})^{e_x \cos^2(\phi_{\mathbf{h}}) + e_y \sin^2(\phi_{\mathbf{h}})}.$$

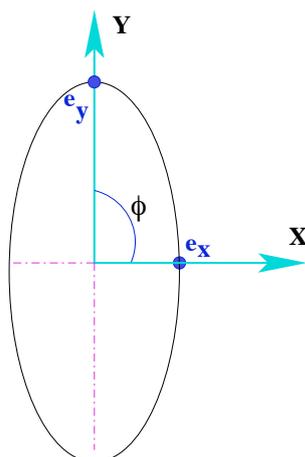


Figura 3.21: Se debe entender los exponentes e_x y e_y como la longitud de los ejes que forman una elipse que presenta la orientación deseada.

Mediante relaciones trigonométricas entre los ángulos, podemos realizar la siguiente simplificación en el exponente que aparece en $D(\mathbf{h})$ siendo $\mathbf{h} = (x_{\mathbf{h}}, y_{\mathbf{h}}, z_{\mathbf{h}})$.

$$\begin{aligned} \cos^2(\phi) &= \frac{x_{\mathbf{h}}^2}{1 - z_{\mathbf{h}}^2}, \\ \sin^2(\phi) &= \frac{y_{\mathbf{h}}^2}{1 - z_{\mathbf{h}}^2}, \end{aligned}$$

Una publicación posterior de Ashikhmin [AS00b] se centra en la descripción de la BRDF en sí. Se basa en un modelo difuso recubierto de una capa especular, que

permite reproducir los efectos visuales de las superficies pulidas como una mesa de madera barnizada. En éstas superficies, la visión desde la dirección normal de la superficie muestra con todo detalle las características de la madera (comportamiento difuso), mientras que si observamos el mismo punto desde el horizonte, comprobaremos que la superficie es altamente especular y ya no podríamos distinguir los detalles anteriores. Este comportamiento queda descrito con las ecuaciones de Fresnel.

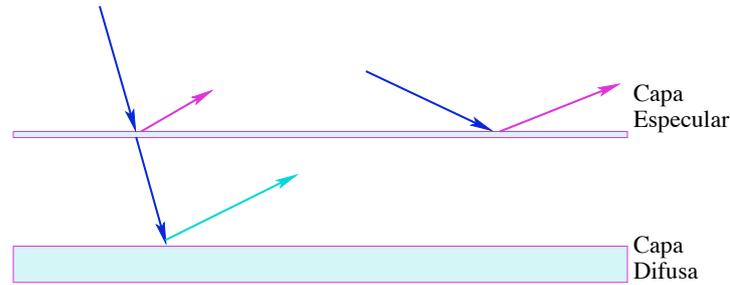


Figura 3.22: En este trabajo la superficie se modela suponiendo dos capas superpuestas, una especular y otra más interna, difusa.

Cuando la luz incide cerca de la dirección normal, la mayor parte de la misma se transmite hasta la capa difusa (dominando dicho término) y, si el ángulo incidente es próximo a 90° , predomina la reflexión *glossy* o especular, como puede verse en la figura 3.22. Este modelo se basa en la suma ponderada del término difuso y el *glossy*. Intervienen la función de distribución D y el valor de la reflectancia de Fresnel F , en su versión aproximada:

$$f_{r,s}(\mathbf{u}, \mathbf{v}) = \frac{D(\mathbf{h}) F(\theta_v)}{8 \pi (\mathbf{h} \cdot \mathbf{u}) \max(\cos(\mathbf{u}), \cos(\mathbf{v}))}. \quad (3.17)$$

Lo fundamental de este modelo es la consecución de un término difuso que obedece la ley de reciprocidad y el principio de conservación de la energía:

$$f_{r,d}(\mathbf{u}, \mathbf{v}) = \frac{28\rho_d}{23\pi} (1 - \rho_s) \left(1 - \left(1 - \frac{\cos(\mathbf{u})}{2} \right)^5 \right) \left(1 - \left(1 - \frac{\cos(\mathbf{v})}{2} \right)^5 \right). \quad (3.18)$$

Finalmente, la función BRDF final es la suma de las componentes difusa y especular, anteriormente vistas. Esto es: $f_r = f_{r,s} + f_{r,d}$. Utilizando la notación del formato GRF este modelo queda descrito como:

ASHIKHMINBRDF ::=
brdf ashikhmin ex =*valor* ey =*valor*
Rs =*valor* Rd =*valor*

Tabla 3.17: Notación en GRF para ASHIKHMINSBRDF

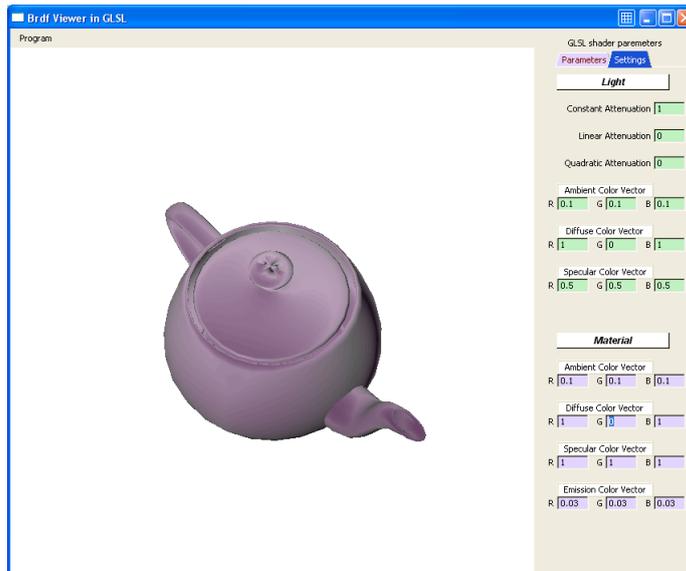


Figura 3.23: La BRDF *ashikhmin* se visualiza según sus parámetros, desde nuestro editor GLSL de BRDFs .

3.23. Granier-Hiedrich (2003)

El trabajo de Granier y Hiedrich [GH03] proporciona un modelo de reflexión de la luz especialmente orientado a aquellos materiales, como los metales viejos, cuyo aspecto les viene dado por una pequeña capa transparente exterior a la superficie.

Busca como uno de sus objetivos conseguir una expresión para la BRDF que sea sencilla de evaluar y presente un número reducido de parámetros, idea similar a la de otros trabajos como los de Ward, Schlick y Lafortune; sin embargo ninguno de estos incluyen implícitamente el valor λ . En esta línea existen otras aportaciones como la dada por Stam [Sta99] en la que se consigue el efecto de difracción que se observa en superficies como las de un CD-ROM. Hirayama [HKYM01] simula el arco iris con un modelo multicapa de películas finas, que maneja superficies lisas y rugosas al mismo tiempo.

Granier y Hiedrich consiguen dos efectos basados en la longitud de onda:

1. Cambio de fase entre la luz, que es directamente reflejada, y la parte de ésta que

atraviesa la capa. Interviene el parámetro que caracteriza el grosor de la superficie de revestimiento. La expresión a considerar es:

$$P(\mathbf{u}, \mathbf{v}, \lambda) = \pi \frac{\eta_2(\lambda) d}{\lambda} (-\mathbf{n} \cdot \mathbf{h}).$$

2. El índice de reflexión, que depende de la longitud de onda, $\eta(\lambda)$ y que puede crear dispersión en el color (efecto producido en un prisma). Se representa por un modelo de subcapas no paralelas, en donde la normal de la capa superior y la correspondiente a la inferior no están alineadas.

Aunque el modelo es dependiente de la longitud de onda, se realiza una primera aproximación para el modelo de color RGB (usando R = 645 nm, G = 525 nm, y B = 445 nm). Así mismo, se supone que la capa más externa η_1 corresponde al medio vacío (consultar tabla 2.1), y que $\eta_1 \leq \eta_2$, por lo que siempre se produce transmisión.

La expresión de esta BRDF incluye componentes para la reflexión R y para la transmisión T correspondientes a las ecuaciones de Fresnel para dieléctricos, vistas en la sección 2.3.3:

$$f_r(\mathbf{u}, \mathbf{v}, \lambda) = R(\mathbf{u}, \mathbf{v}, \lambda) + T(\mathbf{u}, \mathbf{v}, \lambda) + 2 \cos(2P(\mathbf{u}, \mathbf{v}, \lambda)) \sqrt{R(\mathbf{u}, \mathbf{v}, \lambda) T(\mathbf{u}, \mathbf{v}, \lambda)}. \quad (3.19)$$

Se modela la reflexión mediante un lóbulo coseno ponderado por la cantidad de energía que es reflejada directamente; igualmente podría implementarse usando un modelo difuso. El valor de reflectividad —término de Fresnel— se calcula utilizando la aproximación de Schlick:

$$R(\mathbf{u}, \mathbf{v}, \lambda) = F(-\mathbf{u}, \lambda) (\mathbf{v} \cdot \mathbf{n})^n.$$

En este trabajo de Granier y Hiedrich [GH03] se combina la reflexión difusa y la especular para crear un modelo completo de superficie estratificada y utiliza aceleración por *hardware* mediante texturas. En el artículo en cuestión se exponen imágenes que presentan desviación entre las normales de cada capa (que dejan de ser paralelas), cambios en el grosor de la misma y ambos efectos simultáneos.

CAPÍTULO 4

Métodos de estimación de la radiancia

El núcleo principal de la *síntesis realista de imágenes* consiste en determinar el aspecto visual de una escena, dada la descripción de ésta, considerando los aspectos físicos del transporte de la luz en un medio vacío. La luz procedente de iluminarias interacciona con los objetos presentes, dispersándose y siguiendo los procesos de transmisión, absorción y reflexión. El resultado de la simulación queda descrito mediante funciones de radiancia o flujos radiantes, que son transportados de una superficie a otra de la escena hasta que se alcanza la posición del observador, resumiendo los resultados en forma de imagen estática. Existen distintas formas de resolver este cálculo, algunas de las cuales las exponemos a continuación.

En primer lugar, podemos hablar de los algoritmos de radiosidad, que son aquellos que resuelven un modelo difuso de la escena al utilizar métodos de discretización o de **elementos finitos**, en los cuales se usa un operador de transporte discreto para resolver el problema de la *iluminación global*; fueron introducidos por Goral *et al.* [GTGB84] en 1984, siendo durante mucho tiempo la línea seguida por la mayoría de investigadores. Por otro lado, Kajiya en 1986 [Kaj86] proporcionó una solución más general utilizando un **algoritmo estocástico de trazado de rayos**.

Posteriormente, todas las contribuciones, partiendo de una u otra línea, han ido evolucionando combinándose a su vez entre sí. En general, los algoritmos de *iluminación global* pueden clasificarse en atendiendo a dos criterios: el *qué* se intenta resolver y el *cómo* se soluciona.

En función de *qué* se intenta resolver, hay dos tipos de algoritmos:

Algoritmos basados en objetos. Intentan resolver el problema de forma indepen-

diente a los parámetros de visualización (punto de vista, ángulo de visión, etc.). Es el caso de los métodos de radiosidad, algoritmos jerárquicos y *wavelets* [GTGB84, CWH93, GSCH93, NNB97]. En un primer paso se calcula la solución considerando interacciones difusas entre superficies, que han sido subdivididas en parches donde la radiosidad es constante; luego se sintetiza la imagen.

Algoritmos basados en imagen. Toman los parámetros de visualización como su núcleo central y calculan soluciones específicas para una imagen. Es el caso de los métodos basados en trazado de rayos [Whi80, Mil84, WRC88, UPTdS92, MWLT00]. En este caso la solución es parcial, ya que nada se calcula de las partes no visibles de la escena.

Igualmente, en función de *cómo* se intenta resolver hay dos tipos de algoritmos:

Algoritmos determinísticos. Son aquellos basados en técnicas numéricas clásicas para resolver integrales de varias dimensiones o grandes conjuntos de ecuaciones lineales [GSCH93].

Algoritmos basados en Monte-Carlo. Utilizan técnicas estocásticas para resolver integrales de alta dimensionalidad como las del problema de *iluminación global* [SAWG91, Shi96]. Básicamente consisten en promediar los resultados obtenidos de un conjunto elevado de muestras aleatorias. Cuantas más muestras se empleen, mayor será la precisión de la solución final y menor será el ruido intrínseco —el error— que se aprecia en las imágenes. Además del número de muestras, existen varias optimizaciones para acelerar la convergencia a la solución y disminuir el error. Debido a su simplicidad los métodos de Monte-Carlo son ampliamente utilizados [UT93, UTR⁺97, Ure98, JC95] y se combinan a la perfección con los otros algoritmos comentados.

4.1. El cálculo de la radiancia reflejada

La radiancia —como hemos visto en la sección 2.1.3— es la medida de energía calculada para la *síntesis realista de imágenes*. Un sistema de síntesis de imágenes basado en rayos tratará de calcular el valor de radiancia que transporta el rayo —con origen en las fuentes de luz o en la cámara— por toda la escena, y que sufre modificaciones al interactuar con los objetos de la misma. Finalmente el color en el píxel es la radiancia media que transportan los rayos que pasan a través de él.

Las ecuaciones integrales que describen el transporte de la luz no suelen tener soluciones analíticas, por lo que se emplean métodos numéricos, como la regla del trapecio, la cuadratura gaussiana, etc. que funcionan bien en integrales de bajas dimensiones —la complejidad es $O(N^{-d/2})$ con d la dimensión y N el número de evaluaciones del integrando—. Los métodos de Monte-Carlo presentan una mejor ratio de convergencia y, además, ésta es independiente del número de dimensiones del dominio de integración; utiliza números aleatorios para evaluar la integral, dando

4.2. Monte-Carlo para la resolución de integrales

el resultado deseado *en media*, lo que significa que si se emplean otras secuencias de números aleatorios el resultado será distinto, pero en media es el valor de la integral.

Fijémonos en el caso de la integral 2.10 para el cálculo de la radiancia reflejada. Los métodos básicos de Monte-Carlo hacen posible estimar la radiancia reflejada simplemente escogiendo un conjunto de direcciones \mathbf{v} en la semiesfera, para posteriormente calcular radiancia en esas direcciones. El estimador es de la forma:

$$F_N = \frac{1}{N} \sum^N \frac{f_r(\mathbf{u}, \mathbf{v}_i) L_i(\mathbf{v}_i) \cos(\mathbf{v}_i) d\sigma(\mathbf{v}_i)}{p(\mathbf{v}_i)}. \quad (4.1)$$

El principal inconveniente de Monte-Carlo es que para N muestras en el estimador, el algoritmo converge al resultado en $O(N^{-1/2})$, lo que significa que para disminuir el error a la mitad son necesarias $4N$ muestras. En términos de trazado de rayos, cada muestra implica trazar más de un rayo (rayos primarios, secundarios y rayos de sombra) y supone un alto costo computacional. En la imagen resultante el error de Monte-Carlo aparece visible como ruido —píxeles aleatoriamente más brillantes o más oscuros—. Debido a esto, las investigaciones se centran en desarrollar técnicas que reduzcan el error, minimizando el número de muestras necesarias.

En las secciones siguientes comentamos brevemente en qué consisten los métodos de Monte-Carlo empleados en este trabajo, introduciendo, a su vez, las medidas probabilísticas necesarias.

4.2. Monte-Carlo para la resolución de integrales

4.2.1. Teoría de las probabilidades

La teoría de las probabilidades estudia los métodos de análisis que son comunes en el tratamiento de los fenómenos aleatorios, cualquiera que sea el área en que éstos se presenten. Se representan los resultados de un experimento aleatorio —a los que se le asocia una familia de sucesos— por un par (\mathcal{S}^2, Ω) , donde \mathcal{S}^2 representa el suceso seguro y Ω es una clase de subconjuntos de \mathcal{S}^2 .

El espacio muestral \mathcal{S}^2 , con la clase Ω , nos da una estructura en la que es posible establecer una probabilidad, definida como una aplicación de la siguiente forma:

$$\mathbf{s} \in \Omega \longrightarrow P(\mathbf{s}) \in \mathbb{R}.$$

Entendemos por *variable aleatoria* X toda función definida sobre el espacio \mathcal{S}^2 , en el campo de los números reales. Se trata de un valor escogido mediante algún proceso aleatorio. Aplicar una función f a una variable aleatoria (en adelante v.a.) nos da otra

CAPÍTULO 4. Métodos de estimación de la radiancia

v.a. $Y = f(X)$. Revisaremos a continuación los siguientes conceptos:

- Medida de probabilidad. Consideraremos en este trabajo familias de medidas de probabilidad, notadas por P , de forma que para cualquier vector $\mathbf{u} \in \Omega$ tenemos $P_{\mathbf{u}}$ en \mathcal{S}^2 que cumple la siguiente propiedad:

$$P_{\mathbf{u}}(\Omega) = 1.$$

Se asume que la semiesfera negativa presenta probabilidad 0 y, en consecuencia, el resultado seguro es Ω .

- Función de distribución acumulada (CDF) F_X de una variable aleatoria:

$$F_X(x) = Pr[X \leq x].$$

De la definición se deduce que $F_X(+\infty) = 1$ y $F_X(-\infty) = 0$. Además:

$$Prob[a < X \leq b] = F_X(b) - F_X(a).$$

Esta función clasifica las variables aleatorias en *discretas*, cuando $F_X(x)$ es escalonada —constante salvo un conjunto de puntos donde tiene saltos o discontinuidades—, o *continuas*, cuando la función de distribución toma un conjunto continuo de valores. Tomemos como ejemplo el hecho de que en iluminación se puede muestrear aleatoriamente una fuente de luz, seleccionando puntos aleatorios x_i en su área. Si se toma como probabilidad la potencia de una fuente lumínica, con respecto a la potencia de todas ellas, entonces:

$$Prob(x \in L_i) = \frac{\Phi_i}{\sum_{j=0}^M \Phi_j},$$

donde L_i es al área de la fuente, y Φ_i la energía total emitida por la fuente. Se debe cumplir que $\sum_{i=0}^M Prob(x \in L_i) = 1$.

- Función de densidad de probabilidad (PDF) o función de densidad $p(x)$. Bajo un conjunto adecuado de condiciones —que no entraremos a considerar aquí— la función de distribución de una variable aleatoria continua es derivable, siendo su derivada:

$$p(x) = \frac{dF_X(x)}{dx},$$
$$F_X(x) = \int_{-\infty}^x p(t) d(t).$$

Describe la probabilidad relativa de una variable aleatoria para tomar un valor particular. Es similar a ver que esta función representa el límite de la probabilidad media en un intervalo cuya longitud es cada vez menor:

$$Prob(X \in [a, b]) = \int_a^b p(x) d(x).$$

Una función de densidad debe ser siempre positiva e integrable en la unidad en su dominio.

En principio el dominio de una v.a. puede ser discreto o continuo, siendo este último el más frecuente. Existe una v.a. continua que es particularmente común; se denomina v.a. *canónica* y toma valores con igual probabilidad en el dominio $[0, 1]$. Es de importancia, puesto que es computacionalmente muy fácil generar una variable con esta distribución y porque se usa como punto de partida para muestrear con otras distribuciones. Cuando se genera un valor aleatorio x , siguiendo una distribución concreta, se expresa como $x \sim p$. Utilizaremos ξ para denotar un valor aleatorio uniformemente distribuido, generado como $\xi \sim U$.

4.2.2. Caracterización estadística de una distribución aleatoria

El *valor esperado* $E[f]$ del estimador de la función f se define como el valor medio de la función sobre una distribución de valores p en el dominio D :

$$E[f] = \int_D f(x) p(x) d(x).$$

La *varianza* del estimador es la desviación esperada de la función con respecto a su media. Su expresión es la siguiente:

$$V[f] = E[(f - E[f])^2].$$

La media y la varianza presentan las siguientes propiedades:

$$\begin{aligned} E[af] &= a E[f], \\ E\left[\sum f\right] &= \sum E[f], \\ V[af] &= a^2 V[f] \\ \sum V[f] &= V\left[\sum f\right] \text{ si v.a. independientes.} \end{aligned}$$

Haciendo uso de las propiedades anteriores es posible dar una expresión más eficiente en cálculo de la varianza:

$$V[f] = E[f^2] - E[f]^2.$$

La varianza, o momento central de segundo orden, proporciona una forma precisa de cuantificar el error en los métodos de Monte-Carlo. A la raíz cuadrada de este momento se le conoce con el nombre de *desviación típica* (σ). En la literatura de Monte-Carlo es común encontrar la varianza notada como σ^2 .

4.2.3. Integración por Monte-Carlo

Considerese una integral de la forma:

$$I = \int_D f(x) d\mu(x),$$

donde D es el dominio de integración y μ una unidad de medida definida en el mismo. Las integrales de este tipo se aproximan usando métodos numéricos de cuadratura, consistentes en una suma de la forma:

$$\hat{I} = \sum_{i=1}^N w_i f(x_i).$$

Las muestras x_i siguen una distribución aleatoria uniforme conocida. Sin embargo, este tipo de muestreo puede perder detalles de la función que aproxima si ésta no es uniforme. Es necesario muestrear de forma inteligente y generar valores aleatorios (o *sampling*) que cubran el espacio que define el integrando, lo que se consigue aportando una la *función de densidad de probabilidad*.

Así, al muestrear una serie de N variables aleatorias iguales e independientes, X_1, \dots, X_N y siguiendo una función de densidad p concreta, obtenemos el siguiente *estimador*:

$$F_N \equiv F_N(X_1, \dots, X_N) = \frac{1}{N} \sum_{i=1}^N \frac{f(X_i)}{p(X_i)}.$$

Cada valor particular de F_N se denomina *estimado*. Es fácil probar que el estimador F_N da en media un resultado correcto:

$$\begin{aligned} E[F_N] &\stackrel{\text{def}}{=} E \left[\frac{1}{N} \sum_{i=1}^N \frac{f(X_i)}{p(X_i)} \right] \\ &= \frac{1}{N} \sum_{i=1}^N \int_D \frac{f(x)}{p(x)} p(x) d\mu(x) \\ &= \int_D f(x) d\mu(x) \\ &= I. \end{aligned}$$

Para simplificar la notación llamamos $Y_i = f(X_i)/p(X_i)$, $Y = Y_1$ y calculamos su varianza como:

$$V[Y] \stackrel{\text{def}}{=} E[Y^2] - E[Y]^2 = \int_D \frac{f(x)^2}{p(x)} d\mu(x) - I^2,$$

$$V[F_N] = V \left[\frac{1}{N} \sum_{i=1}^N Y_i \right] = \frac{1}{N^2} V \left[\sum_{i=1}^N Y_i \right] = \frac{1}{N^2} \sum_{i=0}^N V[Y_i] = \frac{1}{N} V[Y].$$

Con un desarrollo similar, el estimado de la desviación típica es:

$$\sigma[F_N] = \frac{1}{\sqrt{N}} \sigma[Y].$$

Por lo demás, la integración por Monte-Carlo es bastante simple —muestrear y evaluar— y es sabido que posee una convergencia a la solución $O(N^{-1/2})$, luego cuantas más muestras se utilicen, más lenta será la obtención de la solución, pero estaremos más cerca de obtener su valor real. Este razonamiento es conocido como *ley de los números largos* y puede expresarse de la siguiente manera:

$$\text{Prob} \left\{ \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N Y_i = E[Y] \right\} = 1.$$

Así mismo, el error en la aproximación dependerá de si el estimador es sesgado o no. Siguiendo la notación de Veach [Vea97], sea Q el *estimado*, la cantidad $F_N - Q$ es el error y su valor estimado $\beta[F_N]$, el *sesgo*.

$$\beta[F_N] = E[F_N - Q].$$

4.2.4. Estimadores de Monte-Carlo

Un estimador es *no sesgado* cuando independientemente del número de muestras se cumple que $E[F_N] = Q$, y por tanto el error es, $\beta[F_N] = 0$. De la misma forma se dice que un estimador es *consistente* si el error tiende a 0 con probabilidad 1 cuando aumenta el número de muestras.

Por lo general para medir la bondad del estimador, su *eficiencia* ϵ , se utiliza la varianza V y el tiempo necesario para evaluarlo T :

$$\epsilon[F_N] = \frac{1}{V[F_N]T[F_N]}. \quad (4.2)$$

El caso ideal es aquel que hace nula la varianza y existen numerosas técnicas que se han desarrollado con el objetivo principal de disminuir este valor. Una de ellas se conoce como *importance sampling* y se comenta en el apartado 4.4.4.

4.2.5. El error en los métodos de Monte-Carlo

Los métodos de Monte-Carlo son métodos de muestreo, por lo que tienen asociados errores de muestreo. El hecho de que el estimado no sea igual a su valor esperado

—suponiendo que el estimador es no sesgado— no es un “error” o “fallo”, es sólo un resultado de la varianza de los datos aleatorios con los que se trabaja. La variabilidad de los resultados aleatorios resulta en el *error experimental* [Gen03] como en cualquier otro tipo de experimento científico en las que se reconoce el componente aleatorio.

En cualquier problema de estimación estadística es necesario proporcionar una cota probabilística del error producido. Para ello se debe dar el estimado de la varianza, así como el valor la desviación típica.

4.3. Muestreo según una función de densidad

En este apartado vamos a ver cómo podemos conseguir generar una muestra aleatoria siguiendo una función de densidad particular $p(x)$, partiendo de valores aleatorios distribuidos de forma uniforme.

4.3.1. Método de inversión

Este método consigue mapear una variable aleatoria uniforme en la distribución deseada. Lawrence [LRR05] aplica este tipo de muestreo a los datos de reflectancia tabulados para una BRDF adquirida, consiguiendo un alto ratio de compresión en el tamaño final de los datos necesarios en el muestreo. Primeramente es necesario disponer de la función de distribución o CDF, que en el caso continuo es la integral de la densidad. Tomamos ξ y lo usamos como altura para seleccionar un valor de la función de distribución acumulada F_X (figura 4.1 izquierda). El valor deseado se obtiene evaluando la inversa de la CDF para el valor de ξ . Para ello se siguen los siguientes pasos:

<pre>INVERSION (p: [0,1] \rightarrow \mathbb{R} con $\int_0^1 p = 1$) \rightarrow \mathbb{R} F_X = función de distribución acumulada a partir de p F_X^{-1} = calcular la inversa de la CDF Generar $\xi \sim$ uniforme [0,1] Calcular $x = F_X^{-1}(\xi)$ <i>return</i> x</pre>

Algorithm 1: pseudocódigo INVERSION

4.3.2. Método de muestreo por rechazo

Cuando no podemos obtener —o es muy costoso— la CDF invertida, integrando analíticamente, podemos hacer uso de la técnica conocida como *rejection sampling* [Gen03]. Nuestra función objetivo $f(x)$ corresponde a la densidad con la que

4.3. Muestreo según una función de densidad

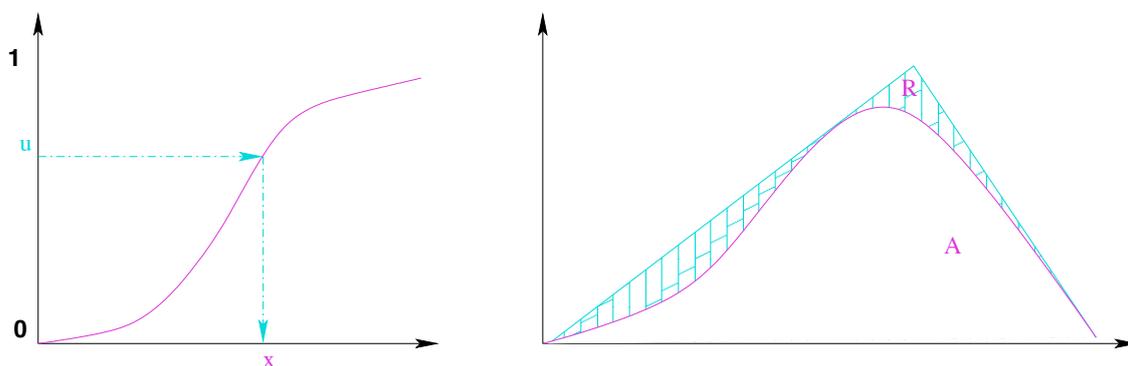


Figura 4.1: Izquierda: valor aleatorio x generado con el método de inversión. Derecha: Áreas de rechazo/aceptación de la muestra en muestreo por rechazo.

deseamos generar x . Esta función cubre un área A de aceptación y conocemos de ella una cota superior $f(x) \leq cg(x)$. La función $g(x)$ es conocida como *hat function* y es fácil de muestrear. El área entre g y f corresponde a la zona de rechazo R como puede verse en la figura 4.1 (derecha).

Este método es bastante simple y funciona sea cual sea la dimensionalidad. Se genera un valor ξ uniforme en $[0,1]$ y un valor $x \sim g(x)$ también en $[0,1]$. Si el punto $(x, \xi cg(x))$ cae por debajo de $f(x)$ la muestra se acepta; de lo contrario se rechaza y el proceso se vuelve a repetir (ver algoritmo 2). La eficiencia de este método va a depender de lo ajustada que esté la función cota.

```

RECHAZO ( )  $\rightarrow \mathbb{R}$ 
  while  $\infty$  do begin
    Generar  $x \sim g$ 
    Generar  $\xi \sim$  uniforme  $[0,1]$ 
    if  $\xi \leq f(x)/(cg(x))$  then
      return  $x$ 
  end

```

Algorithm 2: pseudocódigo RECHAZO

Pongamos el caso de la integración de una función definida en el disco de radio unidad \mathcal{D}^2 circunscrito en un cuadrado con dominio $[0, 1] \times [0, 1]$ y que se muestra a continuación en la figura 4.2. El porcentaje esperado de rechazos viene dado por el área del círculo con respecto a la del cuadrado, esto es $(1 - \frac{\pi}{4}) = 21,5\%$. Esta técnica es bastante simple como puede apreciarse en los pseudocódigos 2 y 3 que acompañan a este texto.

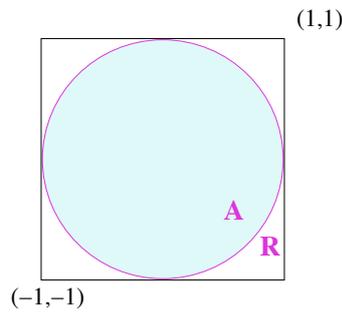


Figura 4.2: Dominio de muestreo \mathcal{D}^2 correspondiente a un disco de radio unidad circunscrito en un cuadrado $[0, 1] \times [0, 1]$.

```
RECHAZODISCO() →  $[0,1]^2$ 
```

```
do begin
```

```
  Generar  $\xi_1$  y  $\xi_2$  ambos uniformes en  $[0,1]$ 
```

```
  Generar  $sx = 1 - 2 * \xi_1$ 
```

```
  Generar  $sy = 1 - 2 * \xi_2$ 
```

```
while ( $sx * sx + sy * sy > 1$ )
```

```
return (sx, sy)
```

Algorithm 3: pseudocódigo RECHAZODISCO

4.3.3. Método de Metrópolis

Una tercera técnica de muestreo general, también conocida como *Markov chain Monte Carlo* [Vea97], presenta como principal ventaja que la función de densidad no debe estar normalizada. Además la PDF puede ser multidimensional. Sin embargo, ocurre que las muestras generadas no son independientes y están altamente correlacionadas ¹.

En 1953, Metropolis y Rosenbluth [MRR⁺53] introdujeron un algoritmo para manejar problemas de muestreo en física computacional relacionados con las propiedades de materiales líquidos, que ha sido utilizado con posterioridad en varias áreas de la física y la química.

Se tiene un espacio de estados Ω , una función de densidad objetivo $p : \Omega \rightarrow \mathbb{R}^+$ y un estado inicial $\mathbf{w}_0 \in \Omega$. El objetivo consiste en generar una secuencia $\mathbf{w}_0, \mathbf{w}_1, \dots$ o *random walk* de forma que \mathbf{w}_i está distribuido eventualmente proporcional a p , independientemente del estado inicial. Cada muestra \mathbf{w}_i es obtenida haciendo un cambio aleatorio al estado \mathbf{w}_{i-1} . Dicho cambio se acepta o rechaza según la probabilidad $a(x, y)$. El esquema de este procedimiento de aceptación-rechazo aparece en el algoritmo 4 siendo K , el núcleo de transición de un estado a otro y que cumple:

¹ La correlación es el grado de asociación entre las variables que nos indicará en qué medida la expresión encontrada explica una variable en función de la otra.

$$\int_{\Omega} K(\mathbf{x} \rightarrow \mathbf{y}) d\mu(\mathbf{y}) = 1 \text{ para todo } \mathbf{x} \in \Omega.$$

METROPOLIS () \rightarrow

Inicializar \mathbf{w}_0 e $i=1$

do begin

 Generar el valor tentativo \mathbf{w}^* según la probabilidad $K(\mathbf{w}_{i-1}, \mathbf{w}^*)$

 Generar ξ uniforme en $[0,1]$

if $u \leq a(\mathbf{w}_{i-1}, \mathbf{w}^*)$ **then**

$\mathbf{w}_i = \mathbf{w}^*$

else

$\mathbf{w}_i = \mathbf{w}_{i-1}$

$i++$

end

Algorithm 4: pseudocódigo METROPOLIS

La probabilidad de aceptar o rechazar un valor propuesto se evalúa como:

$$a(\mathbf{x}, \mathbf{y}) = \min \left[\frac{p(\mathbf{y}) K(\mathbf{y}, \mathbf{x})}{p(\mathbf{x}) K(\mathbf{x}, \mathbf{y})}, 1 \right]$$

4.4. Técnicas de reducción de la varianza

La eficiencia de un estimador de Monte-Carlo se ve afectado por el valor de la varianza. En la literatura se encuentran métodos que relacionan con más probabilidad aquellos caminos aleatorios que contribuyen más al estimador. Comentamos a continuación las técnicas más significativas.

4.4.1. Ruleta rusa

Utilizando la *russian roulette* estamos aumentando la posibilidad de que cada muestra tenga una contribución significativa al resultado, y ahorramos de esta manera el costo de muestrear y calcular para hacer contribuciones pequeñas.

Fijémonos en el caso del estimador de la integral de la radiancia reflejada de la ecuación 4.1. Para cada término de la suma se lanza un rayo secundario con el fin de testear la visibilidad de la fuente de luz desde el punto p . Podemos ver que si el valor de f_r es 0 o muy bajo, o bien, el vector de dirección muestreo \mathbf{v} cae cerca del horizonte, entonces la muestra no contribuye mucho a la integral. En este último caso no es necesario lanzar ningún rayo secundario. Todo esto es posible sin influir en el

valor medio del estimador, pues no introduce sesgo.

Para aplicar esta técnica necesitamos una probabilidad de terminación q entre 0 y 1, que determine cuando el integrando no es evaluado para una muestra concreta, y en su lugar se utiliza como contribución un valor constante c —puede ser cero—. Con probabilidad $1-q$ el integrando es evaluado, pero su valor se ponderará por el factor $\frac{1}{1-q}$ que efectivamente considera las muestras que han sido ignoradas. El nuevo estimador es:

$$F'_N = \begin{cases} \frac{F_N - qc}{1-q} & \text{si } \xi > q, \\ c & \text{en otro caso.} \end{cases}$$

El valor esperado de este estimador es igual al del original:

$$E[F'_N] = (1 - q) \left(\frac{E[F_N] - qc}{1 - q} \right) + qc = E[F_N].$$

Esta técnica no llega a reducir la varianza —incluso puede llegar a aumentarla—; sin embargo, mejora la eficiencia al reducir el esfuerzo de cálculo.

4.4.2. Ruleta rusa optimizada

En su tesis Eric Veach [Vea97] presenta una optimización al método de la ruleta rusa, y lo aplica el algoritmo de *path-tracing* bidireccional. Con él, consigue que se reduzca el esfuerzo en la evaluación de muestras que contribuyen poco al estimador F_N , al tiempo que maximiza la eficiencia de éste (definida en la ecuación 4.2). Tras estudiar el efecto que tiene la función de probabilidad q sobre la varianza y el costo de F_N proporciona una heurística a seguir basada en un análisis matemático del valor óptimo de q .

Se tiene un estimador F_N , que se muestrea en varias ocasiones. Se decide si se evalúa cada muestra o se para, dependiendo de un valor umbral δ . Esta técnica permite considerar la varianza media σ_0^2 y el número de rebotes de cada rayo n_0 . El umbral para parar se calcula y actualiza con cada muestra de la siguiente forma:

$$\delta = \sqrt{\frac{\sigma_0^2}{n_0}}$$

Y aplicando este umbral a todas las contribuciones individuales t obtenemos la siguiente probabilidad modificada:

$$q = \min(1, t/\delta)$$

4.4.3. Muestreo por estratificación

Existe una forma de muestreo que coloca adecuadamente las muestras para que haya menos posibilidad de perder detalles importantes de la función considerada. Con esta técnica subdividimos el dominio de integración D en M regiones no solapadas $\Lambda_1, \dots, \Lambda_M$ llamadas *estratos*, de forma que cubren el dominio original:

$$\cup_{i=1}^M \Lambda_i = D.$$

Se tomarán n_i muestras en cada Λ_i siguiendo la función de densidad p_i , con $\int_{\Lambda_i} p_i = 1$. En el caso particular del muestreo del área de un pixel, éste se puede subdividir utilizando una cuadrícula de $k \times k$ y luego tomar una muestra en cada celda. El resultado obtenido de esta forma, sería mejor que tomar k^2 muestras aleatorias, ya que se disminuye la varianza [Vea97].

Dentro de una region o *stratum* tenemos el siguiente estimador:

$$F_i = \frac{1}{n_i} \sum_{j=1}^{n_i} \frac{f(X_{ij})}{p_i(X_{ij})},$$

donde X_{ij} es la j -ésima muestra obtenida de la función de densidad p_i . El estimador final correspondiente es:

$$F_N = \sum_{i=1}^n v_i F_i,$$

siendo v_i la fracción de volumen del estrato i ($v_i \in [0, 1]$). En promedio se tiene:

$$\mu_i = E[f(X_{ij})] = \frac{1}{v_i} \int_{\Lambda_i} f(x) d(x),$$

y calculamos la varianza en este estrato como:

$$\sigma_i^2 = \frac{1}{v_i} \int_{\Lambda_i} (f(x) - \mu_i)^2 d(x).$$

Si en lugar de usar una muestra empleamos n_i la varianza se convierte en σ_i^2/n_i . Por ello la varianza del estimador final tiene la siguiente expresión:

$$V[F_N] = V\left[\sum v_i F_i\right] = \sum v_i^2 V[F_i] = \frac{v_i^2 \sigma_i^2}{n_i}.$$

En el caso de que n_i fuese proporcional a v_i podríamos considerar la varianza como:

$$V[F_N] = \frac{1}{N} \sum v_i \sigma_i^2.$$

CAPÍTULO 4. Métodos de estimación de la radiancia

Con esto estamos diciendo que escoger una muestra no estratificada es equivalente a seleccionar un estrato aleatorio, muestrear según una función de densidad definida por su volumen y seleccionar un valor aleatorio dentro del estrato.

En el caso de múltiples dimensiones (con d el número de dimensiones) es posible estratificar una de las dimensiones (la que presente mayor correlación) y asociar aleatoriamente muestras del resto. Otra solución consiste en utilizar la técnica conocida como *N-Rooks sampling* [Shi96] o *Latin hypercube sampling* para generar un número de muestras independientemente de las dimensiones. Para un total de N muestras, este método subdivide el dominio $[0, 1]^d$ en N subintervalos para cada dimensión y se asegura de que una muestra caerá en cada subintervalo al aplicar s permutaciones π_1, \dots, π_s de $1, \dots, N$. Finalmente, la posición de cada muestra es:

$$X_i^j = \frac{\pi_j(i) - \xi_{i,j}}{N}.$$

En cualquier caso, ninguna de las estrategias utilizadas (véase figura 4.3) será peor —es decir, con mayor varianza— que un muestreo aleatorio.

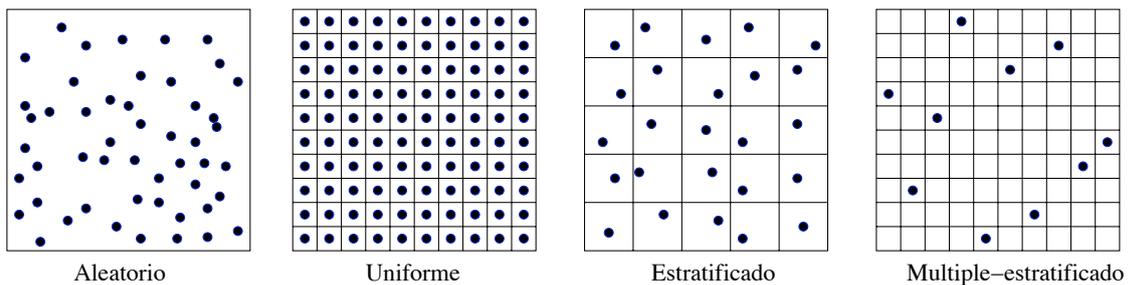


Figura 4.3: Distintas estrategias de muestreo para el dominio $[0, 1] \times [0, 1]$.

4.4.4. Muestreo por importancias

La técnica de muestreo por importancias (IS o *Importance Sampling*) utiliza una función de densidad p proporcional al integrando f . En el caso ideal $p(x) = c f(x)$, la constante de proporcionalidad c es:

$$c = \frac{1}{\int_D f(x) d\mu(x)}.$$

Este caso es ideal, puesto que consigue hacer nula la varianza del estimador F , ya que para todos los puntos X generados en el muestreo se tiene:

$$F = \frac{f(X)}{p(X)} = \frac{1}{c} = \int_D f(x) d\mu(x).$$

Sin embargo, el cálculo de c implica conocer el valor de la integral que de por sí es desconocida. La reducción de la varianza se consigue aproximando f , mediante una función g similar. Es de vital importancia encontrar la densidad que mejor ajuste a f , ya que de lo contrario podríamos aumentar la varianza del estimador.

Cuando el integrando es producto de más de una función —es del tipo $f(x) = f_1(x) f_2(x)$ —, encontrar una PDF que sea similar al producto puede ser muy difícil o quizás imposible. Sin embargo, se conseguiría mejorar la varianza con sólo tener la densidad cercada a uno de sus factores ($g = f_1$, $g = f_2$), intentando que la PDF $p \propto g$ sea además fácil de muestrear con alguna de las técnicas ya comentadas en el apartado 4.3.

Volvamos al caso del estimador de la radiancia saliente (Eq. 4.1). Si generamos aleatoriamente una dirección de salida perpendicular a la normal no nos compensa evaluar la BRDF o estimar la radiancia incidente, puesto que será ponderado por el coseno y es 0. Podemos muestrear en \mathcal{S}^2 o en Ω disminuyendo la posibilidad de escoger direcciones en el ecuador de éstos, o mejor intentar aproximar algunos de los factores del integrando, como la BRDF o la radiancia incidente.

En Informática Gráfica se utiliza IS como técnica para muestrear funciones que no son uniformes; es el caso de la BRDF ($p \propto f_r$), que suele presentar picos en la dirección especular, lo que implica valores muy altos en pequeñas partes del dominio y muy bajos en el resto. En el capítulo 6 (más concretamente en la página 127 apartado 6.2.1) se explica una técnica para subdividir de forma adaptativa el dominio del integrando, aislar estas irregularidades y muestrear mejor estas características.

4.4.5. Muestreo por importancias múltiple

Considérese el siguiente tipo de integral:

$$I = \int_D f(x) g(x) d(x).$$

Sería difícil encontrar una función de densidad proporcional a $f(x) g(x)$, pero sabemos cómo muestrear $f(x)$ y $g(x)$. Es posible combinar estrategias de muestreo cuando nos resultan más fáciles de realizar por separado que de forma conjunta, puesto que la varianza es aditiva; muestrear primero con una, luego con otra y promediar, podría dar muy malos resultados.

El MIS o *multiple importance sampling* [Vea97] proporciona un método para ponderar las muestras de cada técnica, eliminando las variaciones grandes que se puedan producir entre el integrando y la densidad de muestreo, resultantes en divisiones con valores muy pequeños o muy altos. Se toman n_f muestras $x_1 \dots x_{n_f}$ siguiendo p_f y n_g

CAPÍTULO 4. Métodos de estimación de la radiancia

muestras $y_1 \dots y_{n_g}$ según p_g , para obtener el siguiente estimador:

$$F_{MIS} = \frac{1}{n_f + n_g} \left(\sum_i^{n_f} \frac{f(x_i)g(x_i)w_f(x_i)}{p_f(x_i)} + \sum_j^{n_g} \frac{f(x_j)g(x_j)w_g(x_j)}{p_g(x_j)} \right),$$

donde w_f y w_g son funciones peso escogidas de forma que el valor esperado del estimador sea la integral. Hay varias heurísticas para estas funciones; una buena elección es la de balanceado:

$$w_s(x) = \frac{n_s p_s(x)}{\sum_i n_i p_i(x)}.$$

Otra función de peso, conocida como *power heuristic*, es la que sigue:

$$w_s(x) = \frac{(n_s p_s(x))^\beta}{\sum_i (n_i p_i(x))^\beta} \text{ con } \beta = 2.$$

Pensemos ahora en el caso cuando p_f es pequeña y, puesto que se ajusta bien a f , f también será pequeña. Puede aparecer un problema cuando g es grande y evaluamos $\frac{f(x)g(x)}{p_f(x)}$. Sin embargo, podemos ver por la contribución de un estimado x :

$$\frac{f(x) g(x) w_f(x)}{p_f(x)} = \frac{f(x) g(x) n_f p_f(x)}{p_f(x) (n_f p_f(x) + n_g p_g(x))} = \frac{f(x) g(x) n_f}{n_f p_f(x) + n_g p_g(x)},$$

que el denominador no sería demasiado pequeño gracias al término $n_g p_g(x)$, eliminándose este problema.

4.4.6. Remuestreo por importancias - SIR Algorithm

En el libro de Tannen [Tan96] se explica el metodo no iterativo SIR *Sampling Importance Resampling* desarrollado por Rubin en 1987. Esta técnica posibilita suplir datos incompletos de una población con el objetivo de poder usar métodos estándar para conjuntos completos de datos. En este contexto la fracción de datos no disponibles N es pequeño.

Para conseguir datos que se distribuyan aproximadamente según una función objetivo p se requiere una segunda distribución q mucho más fácil de muestrear y evaluar. Sobre esta última se realizará un muestreo por reemplazo en el cual un elemento puede ser seleccionado más de una vez en la muestra: se extrae un elemento de la población se observa y se devuelve a la población, por lo que de esta forma se pueden hacer infinitas extracciones de la población aun siendo esta finita.

1. Se generan M valores $x_i \sim q$.
2. Se calculan los pesos w_j .
3. De los anteriores, se generan por reemplazo $N < M$ valores y_1, \dots, y_m . La probabilidad de que un x_i sobreviva es proporcional a su peso.

Así pues, si se elige como peso $w(x_j) = p(x_j)/q(x_j)$, el remuestreo tomará valores propuestos por q y los filtrará para que se distribuyan aproximadamente respecto a p . Conforme $M/N \rightarrow \infty$ los puntos y_j se distribuyen conforme p .

Se considera una buena elección una proporción $M = 20 \cdot N$.

4.4.7. Muestreo por importancias ponderado

La técnica conocida como *Weighted Importance Sampling* (WIS) modifica el estimador por importancias de Monte-Carlo en otro equivalente con menos variancia. Para ello, se emplea una segunda distribución de densidad.

$$I_{WIS} = \frac{\frac{1}{N} \sum_{i=1}^N \frac{f(x_i)}{p(x_i)}}{\frac{1}{N} \sum_{i=1}^N \frac{g(x_i)}{p(x_i)}}$$

Sea g la distribución objetivo, las muestras x_i serán generadas con una distribución p . Al tiempo se calculan unos pesos que actuarán como factor de corrección que compensan el hecho de no usar g . Esta función tiene como característica que se ajusta mejor al integrando f que la distribución p . Por contra debe estar normalizada y por lo general es un requerimiento difícil de conseguir eficientemente. Puesto que g no se utiliza directamente para el muestreo, disponemos de una mayor libertad para elegir una distribución p eficiente. Este estimador es consistente, aunque adolece de sesgo.

En Iluminación Global, Bekaert [BSW00] utilizó esta técnica en un sistema de radiosidad basado en Monte-Carlo.

4.4.8. Muestreo por importancias estructurado

Agarwal [ARBJ03] introducen una técnica para el muestreo eficiente de *environment maps* mediante una estratificación jerárquica y automática de la imagen, que incluye además, la función de visibilidad V . Ellos estiman la siguiente integral de irradiancia:

$$E(x) = \int_{\Omega} L_i(\mathbf{v}) V(x, \mathbf{v}) \cos(\mathbf{v}) d\sigma(\mathbf{v})$$

Si la iluminación es conocida (el mapa de imagen) es interesante aplicar un preproceso que no sea demasiado costoso al EM y posteriormente aplicar muestreo por importancias sólo en la función de visibilidad. Para mejorar lo segundo también se dispone de una métrica propia de muestreo que guía la fase de estratificación

CAPÍTULO 4. Métodos de estimación de la radiancia

del mapa, asignando un número de muestras distinto en cada estrato. Cada estrato, preintegra la iluminación y permite aproximar el EM como un conjunto discreto de fuentes de luz direccionales.

Sabiendo que la función de visibilidad se comporta con una cierta coherencia —la varianza se espera alta cuando σ es grande y pequeña cuando el ángulo sólido así lo sea— se define la siguiente función de correlación:

$$\alpha(\theta) = \text{Prob}(V(x, \mathbf{w}_1) = V(x, \mathbf{w}_2) \mid \|\mathbf{w}_1 - \mathbf{w}_2\| = \theta)$$

La correlación α es la probabilidad, basada en el ángulo θ , de que la función de visibilidad sea la misma en dos vectores separados por un ángulo θ . Los puntos cercanos resultan con valores similares de probabilidad. Al aumentar la distancia, los valores decaen exponencialmente. Así pues, la media que se emplea es $\hat{\alpha}(\theta) = e^{-\theta/T}$, siendo T el ángulo de correlación.

Basándonos en el anterior análisis de la visibilidad se obtiene la métrica Γ para la distribución de puntos en el dominio del integrando. Depende de un ángulo sólido $\Delta\sigma$ y de L , la iluminación neta de todo el EM.

$$\Gamma(L, \Delta\sigma) = L^a \Delta\sigma^b$$

Los parámetros a y b distribuyen los puntos según una medida de área ($a = 1$, $b = 0$), o bien, el ángulo sólido (con $a = 0$, $b = 1$). Se recomiendan los valores $a = 1$ y $b = 1/4$ [ARBJ03], de esta forma se consigue reducir el número de muestras en fuentes de luz pequeñas.

Esta medida se utiliza en la estratificación en dos fases del *environment map*. En la primera, se detectan regiones como niveles dentro de una jerarquía. Aquellos puntos que comparten valores similares de intensidad se agrupan. Entonces a cada región j le serán asignados $N_j = N \cdot \frac{\Gamma_j}{\Gamma_{4\pi}}$ muestras, donde N es el número total de puntos y $\Gamma_{4\pi}$ es la medida asociada al área completa del EM.

En una segunda fase, las regiones se compactan en estratos minimizando las distancias entre los centros y un punto arbitrario interno a ella. Esto se consigue utilizando un método estándar de clustering. Combinar una serie de píxeles en un estrato aproxima una única fuente de luz direccional localizada en el centro del estrato. Introduce sesgo por ignorar la variación de la BRDF y la orientación de la superficie pero, por contra, permite usar EMs sin necesidad de modificar el sistema de síntesis que ya trabaje con fuentes de luces puntuales.

El muestreo por importancias es implementado considerando el *environment map* como una función unidimensional, y los valores aleatorios como puntos en \mathcal{S}^2 que se distribuyen con una probabilidad proporcional a la irradiancia.

4.4.9. Muestreo por importancias bidireccional

Esta técnica presentada por Burke [BGH05] realiza un muestreo aproximado de la BRDF por la radiancia incidente, no incluyendo el factor coseno. En concreto lo aplican al caso de la integral de iluminación directa, por lo que la distribución finalmente usada reduce el número de test de visibilidad necesarios para el cálculo de la iluminación.

El muestreo de la radiancia incidente —representada mediante un *environment map*— se puede expresar usando la integral del valor de intensidad del pixel, y calculando la CDF. Posteriormente se usa la CDF invertida para obtener una muestra. El muestreo de la BRDF se puede hacer mediante tablas [MPBM03] y un árbol de probabilidad [MH97] o bien de forma analítica si la función dispone de un método propio de muestreo.

El estimador del *Bidirectional Importance Sampling* muestrea ambas funciones de forma conjunta, es decir, mezcla ambos estimadores. Esto puede considerarse una diferencia con respecto a MIS, que considera en su lugar el producto de ambos estimadores.

$$L(\mathbf{u}) = \frac{1}{N} \sum_j^N \frac{f_r(\mathbf{u}, \mathbf{v}_j) \cos(\mathbf{v}_j) L_i(\mathbf{v}_j) V(\mathbf{v}_j)}{p(\mathbf{v}_j)} \simeq \frac{L_{ns}}{N} \sum_j^N V(\mathbf{v}_j)$$

Siendo L_{ns} la radiancia totalmente visible (*no-shadow*), o lo que es lo mismo, la ecuación de iluminación directa omitiendo el test de visibilidad:

$$L_{ns} = \int_{\Omega} f_r(\mathbf{u}, \mathbf{v}) \cos(\mathbf{v}) L_i(\mathbf{v}) d\sigma(\mathbf{v})$$

La probabilidad de cada muestra es el producto normalizado de ambas funciones:

$$p(\mathbf{v}) = \frac{f_r(\mathbf{u}, \mathbf{v}) \cos(\mathbf{v}) L_i(\mathbf{v}) d\sigma(\mathbf{v})}{\int_{\Omega} f_r(\mathbf{u}, \mathbf{v}) \cos(\mathbf{v}) L_i(\mathbf{v}) d\sigma(\mathbf{v})}$$

Tras el muestreo por importancias se aplica muestreo por rechazo. Con ello se consigue descartar aquellos valores aleatorios para los cuales el producto de las dos funciones no sea lo suficientemente alto para motivar el muestreo. Los autores [BGH05] informan que en algunos casos esta técnica es demasiado costosa, dándose un 90% de rechazos. Para acotar el tiempo de ejecución se recomienda emplear SIR (visto en la sección 4.4.6) en lugar de *rejection sampling*.

4.4.10. Remuestreo del muestreo por importancias

La técnica presentada por Talbot *et al.* [JTE05] (denominada en adelante RIS - *Resampling Importance Sampling*) combina el *Importance Resampling* (de la

sección 4.4.6) usado en estadística, con el muestreo por importancias. Se utiliza para generar valores aleatorios proporcionales a una función de densidad g difícil de muestrear directamente, cuando además ésta no está normalizada. Para ello se genera un valor $Y \sim g$ a partir de M muestras originadas con una distribución sencilla p . Cada uno de los valores $x_i \sim p$ lleva asociado un peso $w_i = g(x_i)/p(x_i)$. Posteriormente las muestras serán combinadas entre X e Y de forma similar a WIS, salvo por los pesos. Es más sencillo pensar en RIS como una interpolación de densidades que depende del valor de M tomado. Es característica de RIS el hecho de que la distribución p es peor aproximación al estimando que g , así como que g no está normalizada.

$$\frac{Y \sim p}{M = 1} \qquad \frac{Y \sim g}{M \rightarrow \infty}$$

Los valores Y serán proporcionales a los pesos w_1, \dots, w_M que se utilizan, a su vez, como factores de corrección por el hecho de que g no está normalizada.

El estimador RIS debe, para obtener N muestras y_i , generar $N \cdot M$ valores $x_{ij} \sim p$. En el caso de $M = 1$ estamos en el caso del estimador de *Importance Sampling*.

$$\hat{I}_{RIS} = \frac{1}{N} \sum_i^N \left(\frac{f(y_i)}{g(y_i)} \frac{1}{M} \sum_j^M \frac{g(x_{ij})}{p(x_{ij})} \right) \quad \text{con } N, M > 0$$

Es importante para el buen funcionamiento de este método el hecho de que g y p sean fáciles de evaluar. Los autores proporcionan además una heurística para la selección óptima de los valores de N y M .

4.4.11. Muestreo de funciones mediante *Wavelets*

Wavelet Importance Sample o WaIS [CJAMJ05] es una técnica de muestreo basada en IS que genera valores proporcionales al producto de dos funciones que son representadas mediante *wavelets*. Basándose en trabajos anteriores que representan el producto de funciones 2D, usando como base funciones *Haar*, se presenta la generalización del producto para funciones N-dimensionales. Así, el producto G de dos funciones de N dimensiones F y E —expresadas como *wavelets*— es el producto de los coeficientes de ambas.

$$G = F \cdot E \Leftrightarrow \sum G_i \psi_i = \left(\sum E_j \psi_j \right) \left(\sum F_k \psi_k \right)$$

Siendo G_i, E_j , y F_k las funciones *wavelets* con base *Haar* y ψ los coeficientes de cada una de las anteriores. El producto generalizado se obtiene como:

$$G_i = \sum_j \sum_k C_{ijk} E_j F_k$$

donde

$$C_{ijk} = \underbrace{\int \cdots \int}_n \psi_i(x) \psi_j(x) \psi_k(\bar{x}) dx$$

En este caso los coeficientes toman valores en distintas dimensiones: $x = (x_1, \dots, x_n)$ y $\bar{x} = (x_1, \dots, x_m)$, debiéndose cumplir $m \leq n$.

La segunda aportación de este trabajo es un método de muestreo que toma una distribución inicial de puntos —uniformemente distribuidos, o tomados de una secuencia de quasi-MonteCarlo de baja discrepancia— que serán transformados o *warped* siguiendo la función deseada, esto es, el producto de la función BRDF y la radiancia incidente expresada mediante un *environment map*.

Para ello, se toman probabilidades en cada uno de los niveles de detalle que define el *wavelet* y que representa la función usando las distintas jerarquías de resolución. Utilizan estos niveles para realizar un *warping* jerárquico. En cada nodo, se calcula la probabilidad de los *wavelets* hijos utilizando los coeficientes de escalado $H_{i,0}^l$ de cada región:

$$P_i^l(x) = \frac{H_{i,0}^l}{\sum_t H_{t,0}^l}$$

Es trivial reconstruir los coeficientes de escalado H_t^l a partir de los coeficientes que se emplea en la función base *Haar* y que vienen definidos como:

$$\phi_t^l(x) = 2^{l/2} \phi(2^l x - t) \text{ donde } \phi(x) := \begin{cases} 1 & \text{si } 0 \leq x < 1, \\ 0 & \text{en otro caso.} \end{cases}$$

En el algoritmo de muestreo se comienza desde el nivel más grueso —de menos detalle— y se sigue al más fino de forma recursiva. En cada nivel se considera cada dimensión de forma independiente. Para la primera dimensión tomada, se divide los puntos de entrada en dos mitades, lo que se repetirá de forma independiente para cada dimensión, creando una estructura *KD-tree* de puntos.

En cada nodo del *KD-tree* se almacena la probabilidad de la fracción de volumen que cada mitad de puntos almacena en la fase de división, con respecto al plano de división utilizado. Por ejemplo, si subdividimos según el eje X, nos quedan los conjuntos inferior x^- y superior x^+ con probabilidad $P_{i,x^+}^l = 1 - P_{i,x^-}^l$. Posteriormente los puntos de ambos conjuntos son escalados, para que sus valores se mantengan dentro del intervalo unidad. Finalmente, el número de puntos localizados en cada región hija es proporcional a la probabilidad de dicha región.

Para muestrear según el producto de la BRDF y el *environment map* es necesario representar ambas —y cada una de las funciones BRDFs existentes en la escena— usando

wavelets, lo cual puede ser muy costoso en escenas con varias BRDFs definidas. Una vez calculadas estas representaciones, se genera la estructura jerárquica que distribuye un conjunto de puntos según el producto $G = \sum G_i \psi_i$ y asigna a cada muestra \mathbf{s} una probabilidad expresada en términos de los coeficientes de escalado, y una constante derivada de la descomposición *Haar normalizada*.

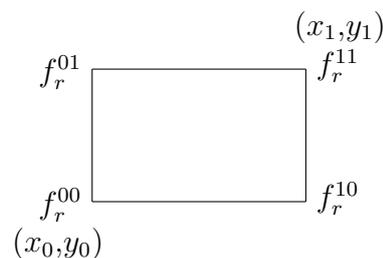
$$p(\mathbf{s}) = 2^{n_l/2} \frac{G_{t,0}^l}{G_{0,0}^0}$$

4.4.12. Muestreo por importancias en dos fases

Cline et al. [CETC06] presentan un método de dos pasadas utilizado para el muestreo aproximado del producto de la BRDF y la radiancia incidente, cuando ésta viene expresada mediante un EM. Este muestreo es aplicado al cálculo de la ecuación de *iluminación directa*.

Este método de dos pasadas puede verse como una variante del muestreo mediante *wavelets* (WaIS), con la salvedad de que el acoplamiento jerárquico de los puntos es guiado por un particionamiento del *environment map*, similar a los árboles de probabilidad de Mc. Cool [MH97].

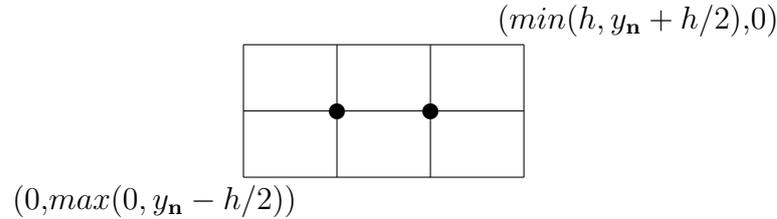
En la primera pasada se construye una estructura jerárquica para cada rayo primario presente en la escena —en *iluminación directa* no es necesario evaluar sucesivos rayos— siendo una aproximación del producto de ambas funciones. El *environment map* se particiona usando un árbol BSP. La función L_i se toma como una tabla de luminancias sumadas —y, si es necesario, compensadas por un factor coseno para evitar el efecto en los polos— y la BRDF se evalúa y anota en las esquinas de cada región, como se muestra en el siguiente dibujo:



El producto de ambas funciones es el valor producto de la BRDF almacenada en las esquinas y la tabla de áreas sumadas, junto con la interpolación bilineal que se efectúa en los nodos hoja del árbol BSP. Aunque la interpolación ofrece una aproximación, los valores en las esquinas sí son exactos.

Otra aportación de este trabajo [CETC06] viene dada por la subdivisión del EM de forma se incluye el término coseno en f_r en una de las regiones, por el hecho de

orientar el EM con el hemisferio positivo formado por la normal. En la construcción del BSP, el nodo raíz se divide en 6 regiones delimitadas por los puntos (x_n, y_n) y $((x_n + w/2) \% w, y_n)$. El primero de ellos corresponde a la representación de la normal \mathbf{n} en el EM y el segundo hace referencia a las dimensiones $w \times h$ de la imagen.



Cada vez que se subdivide una región, otras regiones vecinas a éste, pueden subdividirse en cascada si cumplen un determinado criterio. Con lo que se consigue compensar las BRDF especulares cuyos lóbulos no se alinean con ningún eje de subdivisión. Al llegar a los nodos hoja se busca una subdivisión más para representar los picos de la BRDF, que suelen encontrarse en la dirección del vector de reflexión. Para modelos más generales, encontrar los picos requiere realizar un muestreo de la BRDF para seleccionar un total de $N/4$ puntos, y añadir éstos al BSP.

El particionamiento del *environment map* se complementa con otro criterio de subdivisión basado en el potencial, que permite dividir regiones que no se ajustan bien al producto porque sostienen valores altos sólo en una de las funciones.

Con esta estructura finalizada se procede a la transformación (*warping*) de cada punto al producto $f_r \cdot L_i$. Esto se realiza de forma individual y no en conjunto como WaIS. Cada muestra inicial $(x_s, y_s) \sim [0, 1]^2$ se adapta dependiendo del eje de subdivisión de cada nodo que visita. Cada nodo es seleccionado basándose en la probabilidad que relaciona el peso de una región hija con la de su ancestro. La probabilidad inicial de la muestra se actualiza apropiadamente usando la fracción de área que sostiene el nodo hijo con respecto a la región que está siendo visitada.

$$p(\mathbf{s}) = \frac{\text{area}(EM)}{\text{area}(\text{root})} \cdot \dots \cdot \frac{\text{area}(\text{child}_{j,i})}{\text{area}(\text{region}_i)}$$

Al llegar a un nodo hoja, se realiza un segundo acoplamiento de la muestra. Es distinto al anterior puesto que considera regiones virtuales del EM para poder considerar las regiones hijas que no existen. El resultado final es un punto \mathbf{s} y su probabilidad $p(\mathbf{s})$, que serán utilizados en el estimador de Monte-Carlo.

Este método de compara favorablemente con versiones de MIS (*Multiple Importance Sampling*), RIS (*Resampling Importance Sampling*) y con una versión simulada de WaIS.

4.5. Métodos de Monte-Carlo para la estimación de la radiancia

El objetivo principal de los métodos de Monte-Carlo es la utilización de técnicas estocásticas, basadas en la probabilidad de la ocurrencia de ciertos eventos físicos —emisión, reflexión o absorción de la luz— relacionados con la simulación del transporte de fotones en una escena. A este proceso se le denomina *trazado de partículas*. Esta idea se ha utilizado en el modelado de medios participativos y no participativos. Conceptualmente tiene la misma dificultad tratar con superficies ideales que generales, luz polarizada o depolarizada, geometría simple o escenas complejas, lo que supone una gran ventaja para los métodos estocásticos de síntesis de imágenes.

Las diferentes soluciones desarrolladas pertenecen a dos grupos principalmente. Aquellos que trazan partículas comenzando desde el observador y se dirigen hacia las fuentes de luz, recogen la emisión de los puntos visitados por el camino aleatorio en un proceso denominado *gathering*. Otros, originan los caminos o *paths* en las fuentes de luz y siguen su trayectoria, también conocido como *shooting*.

```

GATHERINGMC() → imagen
  for p := 1 to nPixels do begin
    Color = 0
    for i := 1 to N do begin
      Generar Rayo desde el observador pasando por el pixel p
      ColorMuestra = Trazar(Rayo)
      Color += ColorMuestra ·  $\frac{1}{N}$ 
    end
    EstablecerColorPixel(p, Color)
  end
end

```

Algorithm 5: pseudocódigo GATHERINGMC

```

SHOOTINGMC() → imagen
  for i := 1 to N do begin
    Luz = elegir fuente de luz con probabilidad  $p_e$ 
    Rayo = Muestreo(Luz)
    Potencia = RadianciaEmitida(Luz) ·  $\frac{\cos(\theta)}{p_e}$  ·  $\frac{1}{N}$ 
    Disparar(Rayo, Potencia)
  end
end

```

Algorithm 6: pseudocódigo SHOOTINGMC

La particularidad de cada algoritmo reside no sólo en la categoría a la que pertenece, sino en cómo genera aleatoriamente nuevos rayos —equivalente al problema del

4.5. Métodos de Monte-Carlo para la estimación de la radiancia

muestreo eficiente de direcciones en la semiesfera— y en la manera de implementar la función [Trazar](#) o [Disparar](#) respectivamente. Sobre este tema iremos desarrollando los sucesivos apartados.

4.5.1. El algoritmo de *Ray-Tracing* distribuido

El algoritmo de *ray-tracing* sugerido por Cook [CPC84] ofrece una solución al problema de *iluminación global*, modelando varios tipos de caminos. Es un método de trazado de rayos basado en *gathering* que no simula reflexiones o refracciones ideales, sino que genera nuevos rayos aleatoriamente conforme a la BRDF que caracterize la superficie. La media de las contribuciones de los rayos resultantes del muestreo corresponden a la estimación de las interreflexiones producidas entre objetos. El algoritmo es recursivo, como puede verse por el pseudocódigo 7.

```
TRAZARRTDISTRIBUIDO (Rayo) → color
  (objeto,x) = PrimeraInterseccion(rayo, escena)
  if no interseccion return ColorFondo
  u = -rayo.Direccion()
  Color = RadianciaEmitida(x,u) + IluminacionDirecta(x,u)
  for i := 1 to N do begin
    p = MuestreoBRDF(s,u)
    if p > 0 begin
      RayoGenerado = nuevo rayo desde x en dirección s
      Color += TrazarRTDistribuido(RayoGenerado) ·  $\frac{\text{Peso}(\mathbf{s},\mathbf{u})}{p}$  ·  $\frac{1}{N}$ 
    end
  end
  return Color
```

Algorithm 7: pseudocódigo TRAZARRTDISTRIBUIDO

4.5.2. El algoritmo de *Path-Tracing*

La ecuación del transporte de la luz (abreviada como LTE) es demasiado compleja y, salvo mediante restricciones difusas sobre el material, no puede ser calculada analíticamente.

Es muy habitual encontrarse la ecuación expresada en forma de integral de área, sin embargo, es más flexible la versión dada por Kajiya [Kaj86] integrando caminos de rayos de longitudes infinitas, o *paths*. Se obtiene a partir de la ecuación 2.12 (presente en la página 29) asumiendo que no hay medio participativo, luego la radiancia se conserva en las líneas rectas y por tanto $L_i(x, \mathbf{w})$ es igual a $L(y, -\mathbf{w})$. Si consideramos pocos términos —como se ve en la figura 4.4— la radiancia tiene la siguiente expresión:

$$\begin{aligned}
 L(r_0) &= L_e(r_0) + \int_A L_i(r_1) f_r(r_2, r_1) G(r_2, r_1) dA(r_2) \\
 &+ \int_A \int_A L_e(r_2) f_r(r_3, r_2) G(r_3, r_2) f_r(r_2, r_1) G(r_2, r_1) dA(r_3) dA(r_2) \\
 &+ \dots
 \end{aligned}$$

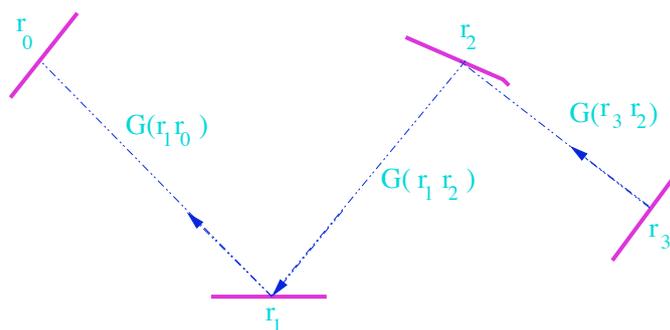


Figura 4.4: Cálculo de la radiancia que deja el vértice r_1 en dirección al punto r_0 después de dos rebotes.

La suma infinita de caminos de longitud creciente puede escribirse de forma compacta. Sea r_i el vértice de la cadena de rebotes —camino o *path walk*— que está sobre una fuente de luz y r_0 la posición del observador, la radiancia producida por la dispersión de la luz sobre el camino i -ésimo se expresa como:

$$\ell(r_i) = \underbrace{\int_A \dots \int_A}_{\dots i-1 \dots} L_i(r_{i-1}) \left(\prod_{j=1}^{i-1} f_r(r_{j+1}, r_j) G(r_{j+1}, r_j) \right) dA(r_2) \dots dA(r_i),$$

$$L(r_0) = \sum_i^\infty \ell(r_i).$$

La función $G(r_i, r_j)$ simplifica la anterior expresión al considerar de forma conjunta la visibilidad entre dos vértices y los términos coseno, dividido entre la distancia que separan ambos puntos.

El algoritmo de *path-tracing* calcula la integral de caminos basándose en estimadores de Monte-Carlo, siendo el primer algoritmo para LTE no sesgado de propósito general. Consiste en generar caminos de forma incremental —comenzando desde el observador y terminando en las fuentes de luz— que rebotan por toda la escena, con el objetivo de calcular la radiancia reflejada de r_1 a r_0 . En la versión bidireccional del algoritmo

4.5. Métodos de Monte-Carlo para la estimación de la radiancia

—desarrollada simultánea pero independientemente por Lafortune-Willems [LW93] y Veach-Guibas [VG94]— los caminos se generan desde la posición del observador y las luces al mismo tiempo, para combinarse luego en uno sólo utilizando un rayo de visibilidad. El *bidirectional path-tracing* evita situaciones del algoritmo básico en donde los rayos encuentran mucha dificultad para alcanzar la fuente de luz, ya sea porque éstas son muy pequeñas, o bien, por oclusiones provocadas por otros objetos.

Caminos con muchos vértices dispersan menos la luz que otros más cortos debido al principio de conservación de la energía de la BRDF. Se utiliza el método de *ruleta rusa* para cortar la cadena, sin introducir sesgo en el estimador. Utilizando q como la probabilidad de parada —supongamos que se dan cuatro rebotes— seguimos teniendo el factor del sumatorio infinito:

$$\ell(r_1) + \ell(r_2) + \ell(r_3) + \frac{1}{1-q} \sum_{i=4}^{\infty} \ell(r_i).$$

Sin embargo, si paramos aleatoriamente según q , entonces:

$$\frac{1}{1-q_1} \left(\ell(r_1) + \frac{1}{1-q_2} \left(\ell(r_2) + \frac{1}{1-q_3} (\ell(r_3) + \dots \right. \right.$$

Se define una probabilidad discreta sobre los objetos de la escena asociada al área de éstos. De esta forma, la probabilidad de muestrear un vértice de la superficie del objeto i se calcula como $p_A(r_i) = \frac{1}{\sum A_j}$, siendo $\sum A_j$ el área total de todas las superficies.

El camino se construye de forma incremental comenzando desde la cámara. En cada vértice la BRDF es muestreada —por lo general según el ángulo sólido— generando una nueva dirección \mathbf{w} con probabilidad $p_{\mathbf{w}}$. El siguiente vértice r_{i+1} se encuentra trazando un rayo desde r_i , siguiendo la dirección \mathbf{w} hasta encontrar el objeto intersecado más próximo. El vértice $i+1$ se muestrea sobre la fuente de luz y no puede hacerse con la misma densidad, ya que se incrementaría la varianza por todos los caminos en donde no hubiesen emisores.

El estimador de Monte-Carlo para un camino es:

$$\frac{L_e(r_{i+1})}{p_A(r_i)} \left(\prod_{j=1}^{i-1} \frac{f_r(r_{j+1}, r_j) |\cos(\theta_{\mathbf{w}_j})|}{p_{\mathbf{w}}(r_j)} \right),$$

cuya implementación puede verse como una extensión del algoritmo de *ray tracing* realizando los siguientes pasos:

```
TRAZARPT (Rayo) → color
  (objeto, x) = PrimeraInterseccion(rayo, escena)
  if no interseccion return ColorFondo
  u = -rayo.Direccion()
  Color = RadianciaEmitida(x, u) + IluminacionDirecta(x, u)
  p = MuestreoBRDF(s, u)
  if p = 0 return Color
  RayoGenerado = nuevo rayo desde x en direccion s
  Color += TrazarPT(RayoGenerado) · Peso(s, u) ·  $\frac{1}{p}$ 
  return Color
```

Algorithm 8: pseudocódigo TRAZARPT

4.5.3. El algoritmo de *Irradiance-Caching*

El algoritmo de *path tracing* puede tomar mucho tiempo y muestras hasta alcanzar la imagen solución sin ningún tipo de ruido. Es posible acelerar los tiempos usando de nuevo de caminos ya calculados. Los vértices $1, \dots, (i - 1)$ son utilizados como punto de partida de un nuevo camino, simplemente trazando un nuevo rayo. Trabajar de esta forma introduce correlación entre los términos del estimador, lo que reduce la calidad del resultado. Las técnicas sesgadas que se han desarrollado para resolver la ecuación del transporte de la luz (LTE) con mayor éxito son *irradiance caching* (IC en adelante) y *photon mapping*.

El algoritmo de IC [WH92] se basa en la idea de que la iluminación indirecta cambia gradualmente en la escena más despacio de lo que lo hace la iluminación directa (pensemos por ejemplo en el borde de una sombra oscura). Si calculamos la solución indirecta de forma precisa en un conjunto reducido de puntos en la escena, podemos interpolar el valor de la iluminación indirecta en la posición evaluada, utilizando muestras de puntos cercanos. Así se espera que el error introducido por no calcular la iluminación indirecta que no sea muy alto y que el ahorro de cálculo sea apreciable.

La iluminación indirecta no se calcula en una fase de preproceso, sino bajo demanda, y es almacenada en una estructura de datos espacial o caché de irradiancias. En el cálculo de la radiancia saliente se busca en primer lugar dentro de la caché las muestras que se consideren válidas por ser lo suficientemente cercanas para interpolar. Es importante ser eficiente con respecto a la representación y almacenamiento de la irradiancia, en la determinación de cuándo hay que calcular un nuevo valor y con qué frecuencia se debe interpolar.

Para materiales difusos la irradiancia por sí sola es suficiente información para calcular la reflexión de la luz en una superficie, por lo que es una suposición muy habitual en este tipo de algoritmos. Se almacena como un único valor en cada punto:

$$E(x) = \int_{\Omega} L_i(x, \mathbf{w}_i) |\cos(\mathbf{w}_i)| d\sigma(\mathbf{w}_i),$$

$$\begin{aligned} L_r(x, \mathbf{w}_r) &= \int_{\Omega} f_r(\mathbf{w}_i, \mathbf{w}_r) L_i(x, \mathbf{w}_i) |\cos(\mathbf{w}_i)| d\sigma(\mathbf{w}_i) \\ &= c \int_{\Omega} L_i(x, \mathbf{w}_i) |\cos(\mathbf{w}_i)| d\sigma(\mathbf{w}_i) \\ &= c E(x). \end{aligned}$$

Las posiciones en donde se calculan nuevos estimados para la irradiancia corresponden a zonas en donde la iluminación cambia más rápidamente, por ejemplo, en las esquinas. El estimador que utilice muestreo por importancias se basará en el factor coseno, que es mucho más sencillo de muestrear que la radiancia incidente. La función de densidad correspondiente, $p(\mathbf{w})$, es $\cos(\theta_{\mathbf{w}})/\pi$.

$$E(x, \mathbf{n}) = \frac{1}{N} \sum_j \frac{L_i(x, \mathbf{w}_j) |\cos(\theta_{\mathbf{w}_j})|}{p(\mathbf{w}_j)} = \frac{\pi}{N} \sum_j L_i(x, \mathbf{w}_j).$$

Determinar la radiancia se convierte en un problema de estimación de la radiancia incidente para cada rayo generado según el lóbulo coseno, lo que se puede hacer utilizando el algoritmo de *path-tracing*. Una vez calculados, la cuestión se centra en cómo reutilizarlos. En general, cuantos más objetos hay cerca del punto en consideración, mayor es el potencial para que se produzca un cambio rápido en la irradiancia. El error cometido por el algoritmo está relacionado con la distancia entre el punto evaluado y el utilizado en el precálculo. Por ello cada muestra de irradiancia en la caché almacena una distancia máxima, o área de influencia en el estimado.

Al introducir sesgo las imágenes finales se obtienen con menor ruido sin emplear mucho más tiempo, sin embargo, la varianza no se reduce de forma conocida al utilizar más muestras. La imagen resulta sin errores visibles pero existen otros substanciales en la solución —*artifacts*— que no se eliminan necesariamente aumentando el muestreo.

4.5.4. El algoritmo de *Photon-Mapping*

En la década de los 90 se inició el desarrollo del algoritmo de *Photon Mapping*, también conocido como *fotosimulación*, de la mano de Jensen [JC95, JMLH01, Jen01, WCS01]; permite simular de forma eficiente efectos de iluminación global. El enfoque de trabajo está basado en desacoplar la información de iluminación con respecto de la geometría. Todo lo relacionado con la luz se almacena en una estructura de datos independiente, denominada propiamente *photon map* (PM). Se construye emitiendo fotones —o paquetes de energía—

desde las fuentes de luz hacia la escena. Todos los datos relacionados con los impactos de los fotones con las superficies de los objetos es utilizada para sintetizar la escena.

A diferencia de *irradiance-caching*, PM calcula reflexiones difusas, difusas-direccionales o *glossy* y especulares (utilizando un paso de trazado de rayos). Se basa en el trazado de partículas, esto es, en la simulación del recorrido de la luz mediante un camino, en donde los vértices almacenan la cantidad de luz incidente que llegan a éstos. Las partículas, o fotones, viajan desde las fuentes de luz por toda la escena, depositando energía en aquellas superficies que son alcanzadas. El primero en la literatura en introducir esta idea fue Arvo [Arv86] quien utilizó mapas de texturas para almacenar la luz. Con un número elevado de estas muestras, acumuladas en un paso de preproceso, obtenemos una representación de la distribución de la luz por toda la escena. En el momento de sintetizar la imagen se utiliza esta representación para estimar la iluminación en un punto, interpolando las contribuciones de las partículas más próximas al mismo.

Las partículas se representan mediante una posición en la superficie, la dirección de entrada y un peso, y expresan la iluminación incidente que llega al punto desde esa trayectoria con un peso asociado. Se utiliza una estructura de tipo *KD-tree* —denominada *photon map*— para almacenar todos los fotones, siendo independiente de la representación geométrica utilizada. Jensen no sólo fue el autor del algoritmo de PM [JC95], sino que, además, fue el primero en almacenar las contribuciones de la luz en una estructura 3D en lugar de emplear una textura.

Se considera una ventaja el hecho de que, gracias a desacoplar la información de la iluminación de la geometría, es posible no sólo una representación simplificada, sino que además es independiente de la complejidad del modelo. Por otro lado se puede utilizar geometría procedural —algo que no es posible en métodos como el de radiosidad—, así como distintas BRDFs. En este punto es importante señalar la necesidad de un método eficiente de muestreo de la BRDF para la generación de caminos aleatorios que disminuya el error del estimador de Monte-Carlo.

Al igual que en el mundo real cada fuente de luz emite un gran número de fotones. Es necesario, igualmente, muestrear eficientemente las luminarias, ya que la potencia de cada una de ellas se divide entre todos los fotones emitidos. Esto significa que cada partícula transporta una fracción de la potencia de la luz emisora y, por tanto, la potencia de los fotones es proporcional solamente al número de fotones emitidos y no al número de fotones almacenados.

Los fotones se consideran de tres tipos distintos, y son almacenados en un *photon map* propio: (1) directos, correspondientes al primer impacto tras la fuente en un camino, (2) cáusticos, cuando llegan a una superficie no especular o transparente después de haber interactuado con uno o varios objetos especulares, y (3) indirectos, que son el resto de reflexiones. Se puede incrementar el número de partículas utilizadas

4.5. Métodos de Monte-Carlo para la estimación de la radiancia

en cada *photon map*, para reproducir con mayor precisión un tipo determinado de reflexión, lo cual hace que este algoritmo sea muy flexible. Como es natural, cuantos más fotones se empleen, mayor es el tiempo —y el consumo de memoria— necesario hasta alcanzar la solución. La persona que utiliza el algoritmo puede, a su vez, indicar cuántos fotones se emplearán en la interpolación. Cuantos más se utilicen, mayor será el suavizado producido en la imagen, pudiendo llegar a emborronar demasiado la misma.

La síntesis de la imagen utilizando *photon-mapping* se hace empleando dos pasadas. En la primera se emiten fotones desde las fuentes de luz y se almacenan en el mapa en la segunda etapa. El PM se emplea en el cálculo de la radiancia saliente en el punto x . Una función delta, selecciona las partículas cercanas a x :

$$\begin{aligned} L_r(x, \mathbf{w}_r) &= \int_{\Omega} f_r(x, \mathbf{w}_i, \mathbf{w}_r) L_i(x, \mathbf{w}_i) |\cos(\mathbf{w}_i)| d\sigma(\mathbf{w}_i) \\ &= \int_A \int_{S^2} \delta(x, y) f_r(y, \mathbf{w}_i, \mathbf{w}_r) L_i(y, \mathbf{w}_i) |\cos(\mathbf{w}_i)| d\sigma(\mathbf{w}_i) dA(y). \end{aligned}$$

Se introduce sesgo en el estimador al considerar que la información sobre la luminosidad de los puntos cercanos se puede utilizar en la estimación de la iluminación en un punto y , por tanto, ésta es obtenida por interpolación. El error introducido dependerá de si la iluminación cambia muy rápidamente en puntos cercanos, lo que normalmente no es muy frecuente. En las cáusticas existe mayor variación y por eso se introducen normalmente más fotones de este tipo.

La implementación de la primera pasada puede verse en el pseudocódigo [9].

```
SHOOTINGPM(Rayo, Potencia) →  
  (objeto, x) = PrimeraInterseccion(rayo, escena)  
  if no interseccion then return  
  if x es visible desde el pixel p then begin  
    Color[p] += Potencia · Peso(rayo.Direccion(), observador) · g(x)  
  end  
  if  $k_r > 0$  then shootingPM(rayo reflejado,  $k_r \cdot$  Potencia)  
  if  $k_t > 0$  then shootingPM(rayo refractado,  $k_t \cdot$  Potencia)
```

Algorithm 9: pseudocódigo SHOOTINGPM

Posteriormente, se realiza una segunda pasada en la que se trazan rayos que recogen la radiancia distribuida por la escena en la primera pasada. Este proceso es conocido como *gathering* y utiliza muestreo por importancias basado en la BRDF para generar nuevas direcciones. En este segundo paso, el *photon map* es una estructura de datos

estática utilizada para calcular estadísticas de iluminación en el modelo.

Habiendo encontrado el primer punto de intersección del rayo j -ésimo para realizar *gather* desde x en la dirección \mathbf{w}_j , necesitamos calcular la radiancia que abandona dicho punto en la dirección $-\mathbf{w}_j$, lo que nos da la radiancia incidente en x . Aplicamos el siguiente estimador:

$$L_i(x, \mathbf{w}_j) = L(t(x, \mathbf{w}_j), -\mathbf{w}_j) \approx \frac{1}{N} \sum_j \frac{f_r(x, \mathbf{w}_r, \mathbf{w}_j) L_{ii}(x, \mathbf{w}_j) |\cos(\theta_{\mathbf{w}_j})|}{p(\mathbf{w}_j)},$$

donde L_{ii} es la radiancia incidente debida al fenómeno de reflexión indirecto.

Una vez que los fotones cercanos han sido encontrados necesitan ser interpolados teniendo en cuenta la densidad de puntos y los pesos de éstos. Se realiza mediante la técnica estadística conocida como *density estimation*, que reconstruye la radiancia a partir de los puntos de impacto almacenados en el *photon-map*, asumiendo que están distribuidos según una función de nuestro interés. Shirley [SBG95] fue el primero en introducir técnicas para DE, y posteriormente, surgieron optimizaciones [Las00, Las04] que consiguen eliminar muchos de los errores intrínsecos del método (*artifacts*), como las zonas oscuras de las esquinas.

4.5.5. Metropolis Light Transport

Veach y Guibas [VG94] proponen un innovador método para LTE —*Light Transport Equation*— aplicando el *metropolis sampling*. Se generan muestras que siguen, como función de distribución, una función arbitraria —no normalizada— notada como f . Este algoritmo consigue resolver escenarios difíciles debido a la mayor componente de iluminación indirecta, por ejemplo, una habitación cuya única luz es exterior y llega a través de una puerta entreabierta.

El algoritmo de Metrópolis genera una secuencia de caminos z , partiendo de un *path* inicial, que es obtenido mediante *path-tracing* bidireccional —aunque cualquier algoritmo de paseos aleatorios es válido—. Un nuevo camino z_t es generado a partir de otro existente z_i , aplicando una pequeña perturbación. Se utiliza la función de densidad tentativa, $T(z_i \rightarrow z_t)$, que evalúa qué probabilidad hay de estando en el estado z_t se evolucione al estado z_i . Las perturbaciones pueden consistir en la eliminación de un vértice de la cadena, la adición de uno nuevo, o simplemente un cambio de dirección.

Para que un camino esté en estado de equilibrio se debe cumplir la ecuación conocida como *detailed balance*, que es:

$$f(z_i) T(z_i \rightarrow z_t) a(z_i \rightarrow z_t) = f(z_t) T(z_t \rightarrow z_i) a(z_t \rightarrow z_i),$$

4.5. Métodos de Monte-Carlo para la estimación de la radiancia

donde la probabilidad de aceptación a indica si la mutación es aceptada o rechazada. Se define como:

$$a(z_i \rightarrow z_t) = \min \left\{ \frac{f(z_t) \cdot T(z_t \rightarrow z_i)}{f(z_i) \cdot T(z_i \rightarrow z_t)}, 1 \right\}.$$

La función de transición mide la contribución de un nuevo camino y, si ésta es muy baja, se genera entonces un camino totalmente nuevo en lugar de realizar alguna mutación. A grandes rasgos, los pasos a realizar se reflejan en el algoritmo [10].

```
ALGORITMOMLT ( )  $\rightarrow$ 
  Generar un path inicial  $z_i$  con PathTracing bidireccional
  for  $i = 1$  to  $N$  do begin
    Encontrar  $z_t$  mutando  $z_i$  con  $T(z_i \rightarrow z_t)$ 
    if  $f(z_t) = 0$  then
      Generar un path  $z_{i+1}$  totalmente nuevo
    else
       $a(z_i \rightarrow z_t) = (f(z_t) \cdot T(z_t \rightarrow z_i)) / (f(z_i) \cdot T(z_i \rightarrow z_t))$ 
      Generar  $\xi$  uniformemente distribuido en  $[0, 1]$ 
      if  $\xi < a(z_i \rightarrow z_t)$  then
         $z_{i+1} = z_t$ 
      else
         $z_{i+1} = z_i$ 
    end
    ColorPath = contribucion del path  $z_{i+1}$  al pixel  $p$ 
    Color[ $p$ ] += ColorPath /  $f(z_{i+1}) \cdot \frac{1}{N}$ 
  end
```

Algorithm 10: pseudocódigo ALGORITMOMLT

4.5.6. Técnicas de reutilización de caminos

4.5.6.1. Múltiples caminos con *global lines*

Los algoritmos de trazado de partículas calculan la función de radiancia de forma local al punto encontrado, por lo que no pueden explorar correctamente la coherencia de la misma. Si en su lugar calculamos la transferencia de la luz de forma global nos encontramos con un nuevo tipo de algoritmos, que representan la unión de técnicas de iteración con los paseos aleatorios.

Sbert [Sbe96] propone la idea de *global lines* —líneas que se trazan y se intersecan a nivel de escena— y lo aplica a un algoritmo de radiación basado en Monte Carlo. En este sentido, el núcleo del algoritmo de Monte Carlo radica en la evaluación de la

integral de factores de forma. Para este propósito, se necesita una densidad apropiada de líneas globales. Inicialmente se calcula la transferencia de potencia entre todos aquellos parches que son intersecados en la dirección de una línea particular. Esta línea inicia aquellos caminos que comenzarían en los parches intersecados y continúa con los anteriores que llevaban energía a los mencionados parches. De esta forma se avanza varios caminos aleatorios en un mismo paso.

4.5.7. Algoritmo de *Instant Radiosity*

Keller [Kel97] presentó a la comunidad de *informática gráfica* un algoritmo híbrido denominado *Instant Radiosity*, relacionado con los métodos de dos pasadas y con el algoritmo bidireccional de *Path-Tracing*. La idea principal consiste en reemplazar la iluminación difusa indirecta por una componente difusa directa, procedente de un conjunto de fuentes puntuales. Este algoritmo subdivide los *paths* que han sido generados bajo un proceso de *shooting* en caminos dependientes de la dirección incidente —generados en un primer paso— que representan la potencia en el vértice final de la cadena, y se consideran como *luces virtuales*.

En un segundo paso, el algoritmo refleja hacia el observador la potencia de estas luces. Es una técnica parecida a *Photon-Mapping* salvo que no se utiliza *ray-tracing* para realizar el paso de *gather*, sino que aplica un trazado de los fotones que se consideran emisivos, similar al *Path-Tracing*. Además, las nuevas direcciones no se generan de forma aleatoria, sino mediante secuencias de Halton. Esta técnica es conocida como *Quasi Monte-Carlo*.

La principal ventaja de este algoritmo es la correlación positiva que se produce en las muestras por el hecho de utilizar los mismos *light-paths* para todos los pixels. Como resultado, las imágenes aparecen suavizadas, disminuyendo el ruido visual. La limitación más importante de este método es el problema que ocurre cuando una *luz virtual* está muy cerca de las zonas que ilumina, pudiendo diverger considerablemente la radiancia calculada.

CAPÍTULO 5

Muestreo directo de la BRDF

El objetivo de los algoritmos basados en Monte-Carlo consiste en la integración de la ecuación del transporte de la luz, que es simulada, mediante la evaluación de la radiancia en caminos que se construyen aleatoriamente, trazando recursivamente rayos en la escena. En el cálculo de la iluminación indirecta las direcciones generadas deben seguir la distribución de la energía saliente, determinada por la *función bidireccional de distribución de la reflectancia*. Por lo general, se utilizan técnicas de muestreo por separado para la reflexión difusa y la especular y no todos los métodos propuestos para el muestreo de la BRDF cuentan con estrategias para ello. Por ello, el poder acoplar de forma eficiente modelos complejos de BRDFs se ha convertido en un desafío propuesto en los algoritmos basados en Monte-Carlo. Igualmente, esta dificultad ha sido la razón de que no se haya favorecido el uso de modelos realísticos.

En el capítulo 3 se realizó una revisión de modelos de BRDFs. Los presentamos de nuevo a modo de resumen en la tabla 5.1, indicando en ellos características como anisotropía, plausibilidad y, lo que más nos concierne en estos momentos, si son adecuados o no para el uso de técnicas de muestreo.

De lo visto en el capítulo 3 y de observar la tabla 5.1, podemos concluir que, no existe un modelo de BRDF totalmente completo —por lo que interesa la combinación de varios modelos— que cumpla la siguiente descripción de características deseables para una BRDF:

1. Físicamente plausible: que respete las propiedades físicas básicas, como la no negatividad, reciprocidad y que cumpla el principio de conservación de la energía.
2. Que describa el comportamiento de una multitud de materiales —incluyendo aquellos con características de anisotropía— con suficiente precisión visual.

		Físico	Plausible	Fresnel	Anisótropo	Muestreo	Material
Modelos de BRDFs	Minnaert	▼	...	▼	▼	▼	superf. lunar
	Torrance-Sparrow	★	▼	★	★	▼	rugosas
	Beard-Maxwell	▼	▼	★	▼	▼	pintados
	Phong-Blinn	▼	▼	▼	▼	★	mates
	Whitted	★	...	★	▼	▼	especulares
	Cook-Torrance	★	▼	★	▼	▼	metales, plásticos
	Kajiya	★	▼	★	★	▼	metales, plásticos
	Poulin-Fournier	★	▼	...	★	▼	difusos
	Strauss	▼	...	★	▼	▼	metales, plásticos
	He et al.	★	★	★	▼	▼	metales
	Ward	▼	▼	▼	★	★	metales
	Westin	★	...	★	★	▼	metales
	Schlick	▼	★	★	★	▼	heterogéneos
	Lewis	▼	★	▼	▼	★	mates
	Oren-Nayar	★	...	★	▼	★	difusos con brillo
	Neumann	▼	★	▼	★	★	metales, plásticos
	Lafortune	▼	★	▼	★	★	mates
	Coupled	★	★	★	▼	★	mates
	Ashikhmin	★	★	★	★	★	pulidas
	Granier-Hiedrich	★	...	★	▼	▼	metales viejos

Tabla 5.1: Datos abreviados de los modelos de BRDFs comentados. El símbolo (★) indica que el modelo posee esa característica; con (▼) indicamos que carece de ella. En los casos no aplicables utilizamos (...).

3. Que describa nuevos materiales sin necesidad de basarse en mediciones.
4. Que su expresión analítica se ajuste bien a datos medidos.
5. Que sea fácilmente calculable.
6. Que sea fácilmente descrita. Pocos parámetros, intuitivos y simples de dar.
7. Que presente un comportamiento especular regido por el *factor de Fresnel*.
8. Que presente un comportamiento difuso no lambertiano.
9. Que sea muestreable fácilmente en algoritmos de Monte-Carlo.

En los sucesivos apartados nos centraremos en aquellas BRDFs que disponen de técnicas de muestreo (condición 9), para ver cómo generar una dirección en la semiesfera siguiendo dicha distribución. Relacionado con su implementación, nos referimos a ellos en su notación GRF (*GRanada File*), formato de escena propio que puede conocerse en mayor detalle consultando el apéndice A.

5.1. Muestreo de los casos ideales

La forma más simple de generar una nueva dirección ocurre cuando se consideran únicamente eventos de reflexión ideales. Al tomar la BRDF difusa perfecta, o especular perfecta, estamos proporcionando una expresión analítica muy rápida de evaluar, que además reduce la función a un espacio menor 2D.

5.1.1. BRDF difusa perfecta

Todas las direcciones tienen la misma probabilidad y están distribuidas según la distribución coseno. Se genera una dirección utilizando el algoritmo de [RejectionSampleDisk](#) —comentado en método de muestreo por rechazo (sección 4.3.2)— que es posteriormente re proyectada en Ω con la función h^{-1} (definida en la página 9 de la presente memoria). El valor de la PDF es:

$$p_{\mathbf{u}}(\mathbf{v}) = \frac{1}{\pi} (\mathbf{n} \cdot \mathbf{v}).$$

Hay que hacer notar que $p_{\mathbf{u}}(\mathbf{v}) = p_{\mathbf{v}}(\mathbf{u})$. Este tipo de muestreo es idóneo para modelos como los de Lambert y Oren-Nayar. La dirección \mathbf{v} también puede ser generada mediante ángulos:

$$(\theta_{\mathbf{v}}, \phi_{\mathbf{v}}) = \left(\arccos \sqrt{\xi_1}, 2\pi \xi_2 \right).$$

donde ξ_1 y ξ_2 son dos variables aleatorias (o números reales aleatorios) uniformemente distribuidos en el intervalo $[0, 1)$ (en adelante utilizaremos este símbolo con este significado).

UNIFORMPDF ::=
pdf diffuse

Tabla 5.2: Notación en GRF para UNIFORMPDF

5.1.2. BRDF especular perfecta

La generación de una dirección aleatoria para esta BRDF sigue siempre el mismo cálculo: dada una dirección \mathbf{u} en un sistema de coordenadas local devolver siempre —es decir, con probabilidad 1— la dirección reflejada simétrica a \mathbf{u} , $\mathbf{r}_\mathbf{u}$ (ver figura 2.6, pág. 14).

La PDF vale infinito para la dirección reflejada, y cero en el resto. Esto implica que en este caso, la PDF deba interpretarse como una distribución, en lugar de como una función. Por tanto la Pdf es $p_\mathbf{u}(\mathbf{v}) = \delta_\mathbf{u}(\mathbf{v})$, donde $\delta_\mathbf{u}$ es la distribución *Delta de Dirac* centrada en $\mathbf{r}_\mathbf{u}$, y definida en Ω .

SPECULARPDF ::=
pdf specular

Tabla 5.3: Notación en GRF para SPECULARPDF

5.2. Modelos basados en el lóbulo coseno

En este apartado se mencionan las estrategias para el muestreo de las BRDFs basadas en la expresión $\cos^n(\alpha)$, más conocido por el nombre de lóbulo coseno ya que su gráfica adopta esa forma. Entran dentro de esta categoría los modelos de Phong, Blinn —y sus respectivas versiones normalizadas dadas por Lewis—, Ward y Lafortune. Se muestrea de manera diferenciada dependiendo del ángulo utilizado en el coseno.

5.2.1. Muestreo basado en el ángulo de reflexión

Lafortune [LW94] ofrece una versión modificada de la BRDF de Phong, que cumple el criterio de plausibilidad. Como señala Arvo [Arv95] la integral que representa la

reflectancia (el albedo) es un momento de doble eje —denominado N — y para el cual este autor proporcionó un método para calcularlo.

$$\rho(\mathbf{u}) = \int_{\Omega} \cos^n(\mathbf{v} \cdot \mathbf{r}_{\mathbf{u}}) \cos(\mathbf{v}) d\sigma(\mathbf{v}) = N(\mathbf{r}_{\mathbf{u}}, n).$$

Cuando ocurre que $\mathbf{v} = \mathbf{r}_{\mathbf{u}}$ se obtiene el valor máximo del albedo, N , y nos permite conocer el factor de normalización, que es:

$$N(\mathbf{n}, n) = \frac{2\pi}{n+1}.$$

Para generar direcciones aleatorias según la BRDF de Phong modificada se ha de diferenciar entre la componentes difusa —que se muestrea con la distribución coseno— y la componente especular. Se utiliza una variable aleatoria para decidir entre ambos fenómenos. La PDF especular cumple que es integrable, y se obtiene de la normalización de la función que nos es de interés —la BRDF—.

$$p_{\mathbf{u}}(\mathbf{v}) = \frac{f_r(\mathbf{u}, \mathbf{v})}{\rho(\mathbf{u})} = \frac{n+1}{2\pi} \cos^n(\mathbf{r}_{\mathbf{v}} \cdot \mathbf{u}).$$

La forma de muestrear según esta distribución consiste en: dada una dirección \mathbf{u} , se generan aleatoriamente dos ángulos dentro del cono de direcciones alrededor de \mathbf{v} . Los ángulos se calculan de la siguiente manera:

$$\alpha = \arccos\left(\xi_1^{\frac{1}{n+1}}\right),$$

$$\phi = 2\pi \xi_2,$$

$$\mathbf{r}_{\mathbf{v}} = (\sin(\alpha) \cos(\phi), \sin(\alpha) \sin(\phi), \cos(\alpha)).$$

así tenemos la dirección reflejada $\mathbf{r}_{\mathbf{v}}$ de la que podemos obtener de forma muy simple por rotación \mathbf{v} , el vector de salida.

HEMISPHEREPDF ::=
pdf hemisphere exp =valor

Tabla 5.4: Notación en GRF para HEMISPHEREPDF

Existe otro método estándar para muestrear la BRDF de Phong. Consiste en usar una PDF en el dominio \mathcal{S}^2 que rota el lóbulo para que el eje principal coincida con el vector normal. De esta forma se obtiene una dirección aleatoria $\mathbf{v} \sim p_{\mathbf{n}}$, si bien inicialmente se puede generar un vector $\mathbf{s} \in \Omega$ como:

$$z_s = \sqrt[n+1]{\xi_1},$$

$$\mathbf{s} = (\sqrt{1 - z_s^2}, 0, z_s),$$

y posteriormente rotarlo alrededor de \mathbf{n} según un ángulo aleatorio, $\alpha = 2\pi\xi_2$, en $[0, 2\pi)$. El vector final \mathbf{v} se obtiene con la rotación inversa de \mathbf{s} .

`SPHEREPDF ::=`
`pdf sphere exp =valor`

Tabla 5.5: Notación en GRF para SPHEREPDF

5.2.2. Muestreo basado en el *half-angle*

Dada una dirección fija de entrada \mathbf{u} a un punto de la superficie la dirección de muestreo se generará con respecto al vector *halfway*. A partir de ahí, y con un vector $\mathbf{h} \sim p_{\mathbf{h}}$ aleatorio podríamos obtener la dirección de salida deseada $\mathbf{v} \sim p_{\mathbf{u}}$. Considerado un sistema de referencia alineado con respecto a \mathbf{h} , la dirección de salida corresponde a la reflexión perfecta de \mathbf{u} , esto es:

$$\mathbf{v} = -\mathbf{u} + 2(\mathbf{u} \cdot \mathbf{h})\mathbf{h}.$$

El procedimiento para muestrear una dirección \mathbf{h} toma dos valores aleatorios uniformemente distribuidos y aplica una transformación relacionada con la función de densidad deseada. Esta transformación —así como la siguiente de \mathbf{h} a \mathbf{v} — debe hacerse de forma controlada para asegurar que conseguimos muestrear con la función de densidad adecuada.

En general, si muestreamos aleatoriamente según un espacio origen, $s_i \in S$, según la PDF p_s y queremos transformarlo en una nueva variable aleatoria para el espacio destino, $t_i \in T$, deseamos que la transformación preserve la medida utilizada, por lo que la PDF p_t se calcula como:

$$p_t = p_s \left\| \frac{\partial s}{\partial t} \right\|.$$

La anterior ecuación se obtiene despejando la expresión que iguala áreas de una subregión de $A_B \subseteq S$ que mapea con otra región $B \subseteq T$.

$$\int_{A_B} p_s(s_1, s_2) ds_1 ds_2 = \int_B p_t(t_1, t_2) dt_1 dt_2.$$

Para la primera transformación consideramos el espacio uniforme $(s_1, s_2) \equiv (\xi_1, \xi_2)$ y $(t_1, t_2) \equiv (\theta_{\mathbf{h}}, \phi_{\mathbf{h}})$. En consecuencia con la primera, para la segunda $(s_1, s_2) \equiv (\theta_{\mathbf{h}}, \phi_{\mathbf{h}})$ y $(t_1, t_2) \equiv (\theta_{\mathbf{v}}, \phi_{\mathbf{v}})$. Obtenemos la proporción adecuada entre medidas, como:

$$p_{\mathbf{u}}(\mathbf{v}) = p_{\mathbf{h}}(\mathbf{v}) \left\| \frac{d\sigma(\mathbf{h})}{d\sigma(\mathbf{u})} \right\|. \quad (5.1)$$

$$\begin{aligned} \frac{d\sigma(\mathbf{h})}{d\sigma(\mathbf{u})} &= \frac{\sin(\theta_{\mathbf{h}}) d\theta_{\mathbf{h}} d\phi_{\mathbf{h}}}{\sin(\theta_{\mathbf{v}}) d\theta_{\mathbf{v}} d\phi_{\mathbf{v}}} = \frac{\sin(\theta_{\mathbf{h}}) d\theta_{\mathbf{h}} d\phi_{\mathbf{h}}}{\sin(2\theta_{\mathbf{h}}) 2d\theta_{\mathbf{h}} d\phi_{\mathbf{h}}} \\ &= \frac{\sin(\theta_{\mathbf{h}})}{4 \cos(\theta_{\mathbf{h}}) \sin(\theta_{\mathbf{h}})} = \frac{1}{4 \cos(\theta_{\mathbf{h}})} \\ &= \frac{1}{4(\mathbf{v} \cdot \mathbf{h})} = \frac{1}{4(\mathbf{u} \cdot \mathbf{h})} \end{aligned}$$

5.2.2.1. Muestreo para las distribuciones de *microfacets*

Para muestrear la componente especular de Blinn (la difusa se realiza según su caso ideal, visto en la sección 5.1, en la página 109), correspondiente a la expresión 3.6, debemos obtener el vector \mathbf{h} como:

$$\begin{aligned} \theta_{\mathbf{h}} &= \sqrt[n+1]{\xi_1}, \\ \phi_{\mathbf{h}} &= 2\pi\xi_2. \end{aligned}$$

Con el vector \mathbf{h} calculamos el vector de salida aleatorio $\mathbf{v} = -\mathbf{u} + 2(\mathbf{u} \cdot \mathbf{h})\mathbf{h}$, cada uno de los cuales cuenta con la siguiente probabilidad:

$$p(\mathbf{v}) = \frac{n+1}{2\pi} \frac{\cos^n(\theta_{\mathbf{h}})}{4(\mathbf{v} \cdot \mathbf{h})}.$$

5.2.2.2. Muestreo de la BRDF de Ward

Ward [War92] presentó un modelo isótropo y anisótropo de BRDF usando un lóbulo de Gauss (o lóbulo basado en la función de Gauss) e indicó, a su vez, cómo generar direcciones que siguieran el factor exponencial de la BRDF. Su expresión puede consultarse a su vez en el *Global Illumination Compendium* [Dut].

De la expresión isótropa de Ward (ecuación 3.9), se despeja y calcula la dirección \mathbf{h} como:

$$\begin{aligned} \theta_{\mathbf{h}} &= \arctan\left(\sigma_m \sqrt{-\log(\xi_1)}\right), \\ \phi_{\mathbf{h}} &= 2\pi\xi_2. \end{aligned}$$

En la expresión anisotrópica de Ward correspondiente a la ecuación 3.10, la dirección \mathbf{h} se calcula como:

$$\theta_{\mathbf{h}} = \arctan \left(\sqrt{\frac{-\log(\xi_2)}{\cos^2(\phi_{\mathbf{h}})/\alpha_x^2 + \sin^2(\phi_{\mathbf{h}})}} \right),$$

$$\phi_{\mathbf{h}} = \arctan \left(\frac{\alpha_x}{\alpha_y} \tan(2\pi\xi_1) \right).$$

La PDF asociada ajusta la expresión del lóbulo gaussiano, como puede apreciarse por la siguiente expresión:

$$p(\mathbf{v}) = \frac{1}{4\pi \alpha_x \alpha_y (\mathbf{h} \cdot \mathbf{v}) \cos^3(\theta_{\mathbf{h}})} e^{-\tan^2(\theta_{\mathbf{h}}) \left(\frac{\cos^2(\phi_{\mathbf{h}})}{\alpha_x^2} + \frac{\sin^2(\phi_{\mathbf{h}})}{\alpha_y^2} \right)}.$$

Ward usa como pesos asociados a las direcciones el parámetro de su modelo k_s . Esta suposición es correcta en la mayoría de los casos, pero no se cumple para direcciones que forman 90° con la normal (o *grazing angles*), produciéndose valores finales muy bajos que se observan como un oscurecimiento de la imagen. En un *technical report* de B.Walter [Wal05] se discuten detalles de implementación para la BRDF de Ward relacionados con los pesos adecuados para el muestreo en un sistema de Monte-Carlo y, —considerando la proporción obtenida en la ecuación 5.1— se deriva la siguiente expresión:

$$w(\mathbf{v}) = \frac{f_r(\mathbf{u}, \mathbf{v}) \cos(\mathbf{v})}{p_{\mathbf{v}}(\mathbf{v})} = k_s (\mathbf{h} \cdot \mathbf{u}) (\mathbf{h} \cdot \mathbf{n})^3 \sqrt{\frac{(\mathbf{v} \cdot \mathbf{n})}{(\mathbf{u} \cdot \mathbf{h})}}.$$

5.3. Muestreo de la BRDF de Neumann

La BRDF de Neumann, vista en la sección 3.19 (en la página 63), se basaba en el plano de proyección de las direcciones de la semiesfera, por lo que se consigue muestrear por el producto de la BRDF y el término coseno al mismo tiempo [NN98, NNSK99].

Se generan N puntos s_{xy} del dominio \mathcal{D}^2 y se proyectan de nuevo en la semiesfera según la siguiente función:

$$\mathbf{s} = s_{xy} + \sqrt{1 - s_{xy}^2}.$$

La forma más sencilla de realizarlo consiste en distribuir uniformemente puntos en $[-1, 1]^2$ y luego utilizar la técnica de *rejection sampling*. También se pueden utilizar en lugar de números aleatorios, series de quasi-MC. En ambos casos, coincide con una distribución coseno, tan utilizada para materiales difusos.

5.4. Muestreo de la BRDF de Ashikhmin

Partiendo de una dirección \mathbf{u} deseamos obtener una dirección aleatoria \mathbf{v} . Ashikhmin proporciona un método para su modelo de BRDF [AS00a]; comienza generando un vector \mathbf{h} siguiendo la siguiente distribución de probabilidad:

$$q(\mathbf{h}) = \frac{\sqrt{(e_x + 1)(e_y + 1)}}{2\pi} (\mathbf{n} \cdot \mathbf{h})^{e_x \cos^2(\phi_{\mathbf{h}}) + e_y \sin^2(\phi_{\mathbf{h}})}.$$

Con \mathbf{u} y \mathbf{h} se obtiene el vector deseado:

$$\mathbf{v} = -\mathbf{u} + 2(\mathbf{u} \cdot \mathbf{h}) \mathbf{h},$$

sin embargo, la densidad del vector resultante no es $q(\mathbf{v})$, $p_{\mathbf{u}}(\mathbf{u}) \neq q(\mathbf{v})$, debido a la diferencia de medidas entre ambos espacios. Como ya sabemos por la ecuación 5.1, se corrige de la siguiente manera:

$$p(\mathbf{v}) = \frac{q(\mathbf{h})}{4(\mathbf{u} \cdot \mathbf{h})}.$$

Para generar el vector \mathbf{h} de una forma eficiente se muestrea siempre generando los ángulos dentro del primer cuadrante de la esfera (ver figura 5.1):

$$\phi_{\mathbf{h}} = \arctan \left(\sqrt{\frac{e_x + 1}{e_y + 1}} \tan \left(\frac{\pi \xi_1}{2} \right) \right),$$

$$\theta_{\mathbf{h}} = \arccos \left(\xi_2^{(e_x \cos^2(\phi_{\mathbf{h}}) + e_y \sin^2(\phi_{\mathbf{h}}) + 1)^{-1}} \right).$$

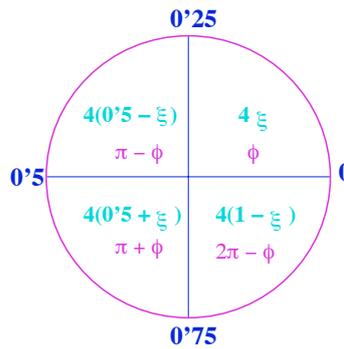


Figura 5.1: Dependiendo del cuadrante elegido al azar, los valores aleatorios se mapean de nuevo.

Después se mapea de nuevo al cuadrante seleccionado mediante un valor aleatorio ξ_1 . Este mismo número se volverá a utilizar tras una adecuada modificación —únicamente hay cuatro posibles— para que se preserve la estratificación. Por ejemplo, para

$\xi_1 \in [0,25, 0,5)$ se obtiene ϕ usando $1 - 4(0,5 - \xi_1)$ y después se gira sobre el eje $\pi/2$.

A continuación se refleja el vector \mathbf{h} generado y se calcula la probabilidad asociada, considerando las características de la distribución anisótropa, como ya se ha comentado. Los parámetros e_x y e_y se notan en GRF como *nu*, *nv*, esto es:

ANISOTROPICPDF ::=
pdf anisotropic nu =*valor* nv =*valor*

Tabla 5.6: Notación en GRF para ANISOTROPICPDF

5.5. Muestreo general para BRDFs analíticas y adquiridas

Existen pocos algoritmos analíticos de muestreo y, como hemos visto, la mayoría están relacionados con los lóbulos coseno. Se han presentado trabajos que intentan ser aplicables a un rango de BRDFs —el mayor posible— que, debido a su complejidad, no cuentan con una forma de muestreo óptima para su uso en algoritmos de Monte-Carlo.

Lawrence y Rusinkiewicz [LRR04] lo hacen con un método basado en la descomposición en factores de la BRDF, que son almacenados tabularmente y de manera compacta. La factorización de una BRDF es algo que ya se había hecho por otros autores [KM99, MAA01], pero Lawrence va más allá al indicar como utilizarlo como función de densidad en la generación de direcciones aleatorias. Matusik [MPBM03] proporciona estas muestras de forma tabulada, para las funciones de reflectancia que él mismo adquiere (Merl DB). Sin embargo, esta representación no es compacta —suele ser necesaria una resolución de 90 x 90 x 180 x 180 para BRDFs isotropas— y requiere tanto espacio o más, como la misma BRDF.

El método propuesto por Lawrence y Rusinkiewicz comienza con un cambio a la parametrización basada en el *half-angle* [Rus98] y una descomposición en factores llevados a un dominio 2D más simple. Los factores deben ser funciones positivas para que se puedan tomar como PDFs y al menos una de ellas debe cumplir que sea evaluable directamente. La BRDF se utiliza en su forma analítica para su evaluación y para el muestreo por importancias se usa una aproximación a la misma, basada en dos series de funciones, que permiten expresar la BRDF como una suma de productos de funciones que dependen de uno solo de los parámetros \mathbf{u} y \mathbf{v} .

$$f_r(\mathbf{u}, \mathbf{v}) \cos(\mathbf{v}) \approx \sum_j^J F_j(\mathbf{u}) G_j(\mathbf{w}),$$

5.5. Muestreo general para BRDFs analíticas y adquiridas

siendo \mathbf{w} el vector de dirección resultante del muestreo, con respecto a la parametrización seleccionada por el usuario. Es posible generar direcciones que representen la dirección incidente, \mathbf{v} , o bien parametrizar con respecto a \mathbf{h} y de esa forma alinear características importantes de la BRDF como es el brillo especular. Los autores han constatado que para BRDF difusas, como Poulin-Fournier, es mejor basar \mathbf{w} en la parametrización en \mathbf{v} .

El procedimiento de factorizar la función de reflectancia comienza con la construcción de la matriz Y que almacena f_r multiplicado por el término coseno. Utilizando una resolución $N_{\mathbf{u}} = N_{\theta} + N_{\phi}$, entonces, para cada $N_{\mathbf{u}}$ se generan $N_{\mathbf{w}}$ muestras, que se almacenan en la matriz $Y = N_{\mathbf{w}} \times N_{\mathbf{u}}$. Esta matriz se descompone en otras dos matrices de menor dimensionalidad y que representan a las funciones F y G :

$$\begin{bmatrix} Y \end{bmatrix} = \begin{bmatrix} G \end{bmatrix} \begin{bmatrix} F \end{bmatrix} \quad \text{con } G : N_{\mathbf{w}} \times J \quad \text{y} \quad F : J \times N_{\mathbf{u}}.$$

La función G , puesto que no puede ser evaluada directamente, vuelve a descomponerse en funciones 1D mucho más sencillas y de esta forma:

$$f_r(\mathbf{u}, \mathbf{v}) \cos(\mathbf{v}) \approx \sum_{j=1}^J F_j(\mathbf{u}) \sum_{k=1}^K u_{jk}(\theta_{\mathbf{w}}) v_{jk}(\phi_{\mathbf{w}}).$$

Por ejemplo, esta representación es válida para el modelo anisótropo de Ashikhmin si utilizamos $J=1$ y $K=2$. Para Blinn tan sólo sería necesario $J=K=1$. En general no son necesarios emplear muchos términos o factores para representar BRDFs adquiridas y analíticas, pero hay que ajustar de forma manual los parámetros $J, K, N_{\mathbf{u}}, N_{\mathbf{w}}$ así como el tipo de parametrización.

Para su correcta utilización los factores son normalizados, tras hallar una cota superior:

$$f_r(\mathbf{u}, \mathbf{v}) \cos(\mathbf{v}) \approx \sum_{l=1}^L F'_l(\mathbf{u}) u'_l(\theta_{\mathbf{w}}) v'_l(\phi_{\mathbf{w}}) = \sum_{l=1}^L T_l(\mathbf{u}, \theta_{\mathbf{w}}, \phi_{\mathbf{w}}),$$

con $L = JK$.

Finalmente las muestras se obtienen al usar la técnica de inversión de la CDF (vista en la sección 4.3.1) empleando los factores 1D tabulados. Puesto que se tiene que recalculer la CDF, cada vez que cambia la dirección \mathbf{u} se deben habilitar mecanismos de reutilización para las muestras de un mismo pixel, o bien precalcular dichas matrices.

La función de densidad de la dirección de salida es:

tesis, *BRDF 3D Plot* (consúltese apartado 7.10),

Comparativas medidas en tiempo y error de este método frente al algoritmo que nosotros presentamos, se considerarán en el siguiente capítulo de la presente memoria.

5.6. Muestreo compacto de BRDFs adquiridas

Con un trabajo posterior, los mismos autores de la factorización de la BRDF ofrecen una mejora aplicable a este método [LRR05], que se aplica exclusivamente al muestreo de BRDFs adquiridas. La mejora al método consiste en hacerlo adaptativo, de forma que en el muestreo por inversión de la CDF los datos considerados no se distribuyan uniformemente, y se siga la misma distribución haciendo uso de otra distribución equivalente que viene expresada de manera compacta. Este objetivo se consigue mediante la aplicación de una serie de pasos. Primeramente se descompone la PDF N-dimensional en una distribución marginal 1D \tilde{p} , junto a una colección de distribuciones condicionales a esta también 1D.

$$\begin{aligned}\tilde{p}(x) &= \int_{-\infty}^{\infty} p(x, y) dy \\ p(y|x_i) &= \frac{1}{x_i - x_{i-1}} \int_{x_{i-1}}^{x_i} \frac{p(x', y)}{\tilde{p}(x')} dx'\end{aligned}$$

Seguidamente, la representación numérica empleada consiste en el cálculo discretizado de las CDFs (*Cumulative Distribution Function*) marginal y condicional, para ser utilizadas en conjunción con el muestreo por inversión. Un punto interesante del método es que las CDFs discretas son invertidas en tiempo de ejecución, mediante la realización de una búsqueda binaria que selecciona el intervalo $[P_i, P_{i+1}]$ que contiene el valor aleatorio ξ .

Otro punto a favor es que estas tablas CDFs, expresadas con N valores uniformes, pueden comprimir su tamaño (de 65 MB a 2.3 MB en algunos casos) si se encuentran los $M < N$ valores adaptativos que representan la misma distribución. Esto conseguiría facilitar su manejo en un sistema de síntesis de imágenes, teniendo en cuenta que se creará una tabla por cada BRDF presente en la escena a renderizar. El proceso de compresión comienza con la aproximación a la tabla CDF mediante una curva; el algoritmo de tipo *greedy* de *Douglas-Peucker* [Ros97] dará las localizaciones de las muestras adaptativas que representan dicha curva. La misma aproximación resultante es la CDF marginal a utilizar en el método de inversión, siendo necesario recalcular las CDFs condicionales.

En general se puede decir que la factorización de la BRDF en el producto de factores positivos, mejora anteriores métodos de factorización al proporcionar un método específico de muestreo. Si bien los autores únicamente notifican datos referentes

CAPÍTULO 5. Muestreo directo de la BRDF

al muestreo de unas pocas BRDFs analíticas (y otras tantas procedentes de datos adquiridos), nosotros hemos incorporado esta técnica a *BPT*, el sistema de trazados de rayos utilizado en todas las imágenes sintéticas presentes en el apéndice B. A continuación se muestra la sintaxis para definir esta PDF, que representa la CDF uniforme con $N = N_\theta + N_\phi$ valores. Un parámetro de tipo lógico indica si el método debe comprimir la tabla CDF o no. La implementación de esta PDF, su definición y uso se caracterizan por su notación GRF (más detalles sobre este formato de escena puede ser consultados en el apéndice A).

```
CASCADECDF ::=  
pdf cascadeCDF nTh =valor nPh =valor  
compress =valor-lógico
```

Tabla 5.8: Notación en GRF para CASCADECDF

Esta técnica es aplicable, a su vez, al muestreo de los mapas de iluminación distantes, o *Environment Maps* (EM). En tal caso sería necesario añadir algunos valores de muestreo no-uniformes, para garantizar que la distribución marginal está bien representada y que se aproxima adecuadamente la variación de la función en la dimensión ortogonal a la dada. Los nuevos valores se obtienen, en las regiones en las cuales la evaluación de la función gradiente muestra valores altos. En nuestro caso, no hemos calculado el gradiente ya que únicamente hemos aplicado este método al muestreo de BRDFs adquiridas.

De igual forma que en el artículo original [LRR05], nosotros hacemos uso de las mediciones de reflectancias de mas de 100 materiales de Mitsubishi Electric Research Laboratories (MERL)¹. Restringiéndonos a un subconjunto de estas, evaluamos los aspectos concernientes al consumo de memoria de este método y al tiempo de precálculo. Por un lado la tabla 5.9 presenta los datos de memoria cuando se emplean para el muestreo de la BRDF un conjunto de tablas CDFs evaluadas de forma uniforme—cuando las muestras se distribuyen como en la figura 5.3—, e independientemente de la BRDF. Y por otro, la tabla 5.10 presenta información sobre los tiempos de precálculo cuando se utiliza compresión de la tabla CDF (incluye el tiempo de cálculo de la correspondiente CDF uniforme, la aproximación de la PDF marginal por una curva y el recálculo de la PDF condicional), la memoria empleada por esta técnica (tras liberar las estructuras de datos iniciales, correspondientes a las CDFs uniformes de alta resolución tomadas para su compresión) y, por último, el porcentaje de reducción con respecto a la CDF uniforme con resolución $16 \times 32 \times 256 \times 32$.

¹ Datos de BRDFs adquiridas de libre disposición <<http://www.merl.com/brdf/>> [consultado en noviembre de 2007]

5.6. Muestreo compacto de BRDFs adquiridas

El siguiente capítulo, una vez presentada nuestra propuesta para el muestreo de BRDFs, dará una comparativa entre este método y el nuestro, aplicado a BRDFs adquiridas.

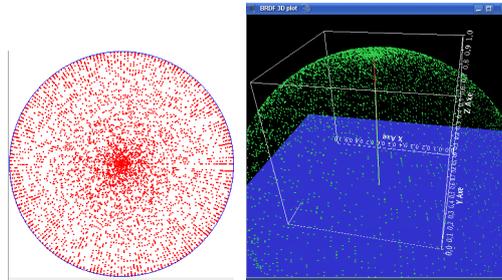


Figura 5.3: Gráfica en 2D y 3D de la *Cumulative Distribution Function* dada por la herramienta *BRDF 3D Plot*. Se aprecia cómo se obtienen valores uniformemente distribuidos. La discretización de vectores es alta y se cubre bien la semiesfera de direcciones.

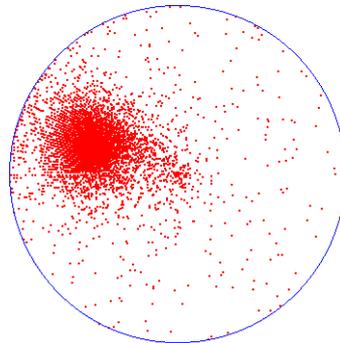


Figura 5.4: Gráfica en 2D de la BRDF adquirida *nickel.binary* BRDF. Con el método de inversión de las CDFs calculadas y almacenadas, las direcciones pueden ser generadas conforme a los datos tabulares de la reflectancia.

nTh	nPh	Mem (KB)	Mem (MB)
16	64	8320	8.13
100	32	82500	80.57
64	100	323200	315.63

Tabla 5.9: Consumo de memoria del método CascadeCDFs empleando muestras distribuidas uniformemente

Merl BRDFs	Compr.(sec)	CDF (KB)	Compr.(KB)	Reducción
alum-bronze.binary	572.13	33792	998.15	97.05 %
alumina-oxide.binary	577.4	33792	174.75	99.48 %
aluminium.binary	568.31	33792	1156.5	96.58 %
aventurnine.binary	562.68	33792	331.65	99.02 %
beige-fabric.binary	559.53	33792	187.66	99.44 %
black-fabric.binary	579.51	33792	265.13	99.22 %
black-obsidian.binary	580.54	33792	802.2	97.63 %
black-oxidized-steel.binary	575.38	33792	761.45	97.75 %
black-phenolic.binary	577.38	33792	901.96	97.33 %
black-soft-plastic.binary	562.24	33792	799.8	97.63 %
blue-acrylic.binary	576.79	33792	382.53	98.87 %
blue-fabric.binary	578.26	33792	351.58	98.96 %
blue-metallic-paint2.binary	572.29	33792	1038.0	96.93 %
blue-metallic-paint.binary	575.35	33792	1110.05	96.72 %
nickel.binary	567.49	33792	1023.62	96.97 %
red-plastic.binary	586.5	33792	275.83	99.18 %
teflon.binary	561.78	33792	184.51	99.45 %
violet-acrylic.binary	578.69	33792	826.69	97.55 %
white-marble.binary	565.71	33792	247.79	99.27 %
yellow-paint.binary	566.06	33792	143.47	99.58 %

Tabla 5.10: Consumo de memoria del método CascadeCDFs empleando muestras distribuidas de forma adaptativa lo que compacta considerablemente el conjunto de CDFs almacenadas.

CAPÍTULO 6

Muestreo eficiente de BRDFs genéricas

En anteriores capítulos hemos hablado de la importancia de conseguir modelos BRDFs simples de evaluar, algunos de los cuales cuentan con un algoritmo de generación de direcciones de salida, proceso básico para la mayoría de métodos de estimación de la radiancia en el cálculo de *iluminación global* de una escena.

En el contexto de un estimador de Monte Carlo de la radiancia saliente, el punto de la superficie sobre el que se efectúa la evaluación es conocido, así como su vector normal y la BRDF que modela las propiedades ópticas de la superficie. El objetivo del estimador consiste primeramente en:

- proporcionar un vector aleatorio de salida conocida la dirección incidente, usando una función de densidad de probabilidad proporcional a la BRDF. En el mejor de los casos, para que el muestreo sea más eficiente, la función de densidad debe ser proporcional a la BRDF por el término coseno [LW94].
- estimar la radiancia saliente en la dirección generada.

El actual capítulo presenta un algoritmo para el muestreo eficiente y exacto de BRDF genéricas, esto es, apto para cualquiera de los modelos de BRDFs mencionados con anterioridad. El objetivo principal consiste en proporcionar una función de probabilidad adaptativa, que permita un muestreo proporcional al producto de la función BRDF y un término coseno.

6.1. Estimador de Monte-Carlo empleando BRDFs genéricas

La ecuación de reflectancia, puede ser resuelta haciendo uso de métodos de integración por Monte-Carlo, ya que no siempre es posible obtener expresiones analíticas para la radiancia saliente o entrante.

La integral vista en la ecuación 2.10 (localizada en la página 29) es aproximada mediante el valor medio de la función al ser evaluada usando un número N de valores aleatorios —vectores de dirección— $(\mathbf{s}_1 \dots \mathbf{s}_N) \in \Omega$ independientes e idénticamente distribuidos en el dominio, que siguen una distribución de probabilidad $P_{\mathbf{u}}$ definida en Ω , dado un vector $\mathbf{u} \in \Omega$. Es condición necesaria que $P_{\mathbf{u}}(\Omega) = 1$. La expresión de dicho estimador fue dada en la ecuación 4.1 (pág. 75).

Podemos definir distintos estimadores de Monte-Carlo dependiendo de que la integración en el dominio Ω se realice con la medida del ángulo sólido (σ) o la versión proyectada de esta (σ_p). En cada caso asignamos al estimador una PDF distinta. En el primer caso:

$$p_{\mathbf{u}}(\mathbf{v}) \stackrel{\text{def}}{=} \frac{dP_{\mathbf{u}}(\mathbf{v})}{d\sigma(\mathbf{v})},$$

daría el siguiente estimador:

$$L_r(\mathbf{u}) \approx X_N(\mathbf{s}_1, \dots, \mathbf{s}_N) \stackrel{\text{def}}{=} \frac{1}{N} \sum_{i=1}^N \frac{f_r(\mathbf{u}, \mathbf{s}_i) L_i(\mathbf{s}_i) \cos(\mathbf{s}_i)}{p_{\mathbf{u}}(\mathbf{s}_i)}. \quad (6.1)$$

En el segundo caso:

$$q_{\mathbf{u}}(\mathbf{v}) \stackrel{\text{def}}{=} \frac{dP_{\mathbf{u}}(\mathbf{v})}{d\sigma_p(\mathbf{v})},$$

proporcionaría este otro:

$$L_r(\mathbf{u}) \approx Y_N(\mathbf{s}_1, \dots, \mathbf{s}_N) \stackrel{\text{def}}{=} \frac{1}{N} \sum_{i=1}^N \frac{f_r(\mathbf{u}, \mathbf{s}_i) L_i(\mathbf{s}_i)}{q_{\mathbf{u}}(\mathbf{s}_i)}. \quad (6.2)$$

Por lo general, la función de densidad $p_{\mathbf{u}}$ en σ es la utilizada en la literatura de *iluminación global*. Sin embargo, la PDF que proponemos para el muestreo de direcciones de salida será del tipo $q_{\mathbf{u}}$, proporcional a la BRDF por el término coseno que aparece en la ecuación de radiancia. Esto permite simplificar el cálculo del estimador al ahorrar la evaluación del coseno, así como una multiplicación. Así mismo, en determinados casos, se evitan posibles errores asociados a divisiones por valores muy cercanos a cero.

Nuestro algoritmo se ha implementado y ha evaluado en un sistema de síntesis de imágenes basado en *path-tracing*, aunque el método de muestreo bien puede ser

6.1. Estimador de Monte-Carlo empleando BRDFs genéricas

empleado con cualquier otro sistema de cálculo de la radiancia basado en Monte-Carlo. El muestreo de la BRDF se realiza generando vectores \mathbf{s}_i proporcionales a la propia BRDF multiplicada por un factor coseno, gracias a la utilización del dominio \mathcal{D}^2 , tal y como se irá comentando en lo sucesivo.

Usando la función de mapeo h , que se vió en la sección 2.1.1 (página 8), podemos realizar un cambio en el dominio de integración, y aproximar la ecuación de reflectancia en \mathcal{D}^2 . El estimador de nuestro interés es el dado en la ecuación 6.2. El uso de σ_p es equivalente a la medida de área A de la ecuación 2.3, por lo que la expresión para la radiancia saliente en una dirección es:

$$L_r(\mathbf{u}) = \int_{\mathcal{D}^2} f_r(\mathbf{u}, \mathbf{v}_{xy}) L_i(\mathbf{v}_{xy}) dA(x, y) \quad (6.3)$$

$$= \int_{[-1,1)} \int_{[-1,1)} f_r(\mathbf{u}, \mathbf{v}_{xy}) L_i(\mathbf{v}_{xy}) dx dy, \quad (6.4)$$

siendo $\mathbf{v}_{xy} = h^{-1}(x, y)$.

Aunque la definición más usual de la BRDF sea dando dos vectores en la semiesfera, conviene introducir otras notaciones por simplicidad. Utilizaremos indistintamente:

$$f_r(\mathbf{u}, \mathbf{v}_{xy}) = f_r(\mathbf{u}, h^{-1}(\mathbf{v}_{xy})) = f_r(\mathbf{u}, \mathbf{v}).$$

Usar \mathbf{v}_{xy} en lugar de simplemente \mathbf{v} , pretende hacer explícito que es un vector que está asociado a un punto (x, y) en el disco. Lo mismo se puede decir para otras funciones definidas en Ω que acepten vectores en \mathcal{D}^2 , como $L_i(\mathbf{v}_{xy})$.

Finalmente, el estimador que implementamos es:

$$X_n(\mathbf{s}_1, \dots, \mathbf{s}_n) \stackrel{\text{def}}{=} \frac{1}{n} \sum_{i=1}^n \frac{f_r(\mathbf{u}, \mathbf{s}_i) L_i(\mathbf{s}_i)}{q(\mathbf{s}_i)}. \quad (6.5)$$

ESTIMARRADIANCIA1 ($\mathbf{u} \in \Omega, N \in \mathbb{R}^+, f_r, L_i, P_{\mathbf{u}}$) $\rightarrow \mathbb{R}$

Sea $q_{\mathbf{u}} = dP_{\mathbf{u}}/d\sigma_p$

Sea $s := 0$

for $i := 1$ **to** N **do begin**

 Generar un punto aleatorio $\mathbf{s}_{xy} \in \mathcal{D}^2$ distribuido conforme a $P_{\mathbf{u}}$

$s := s + (f_r(\mathbf{u}, \mathbf{s}_{xy}) L_i(\mathbf{s}_{xy})) / q_{\mathbf{u}}(\mathbf{s}_{xy})$

end

return s/N

Algorithm 11: pseudocódigo ESTIMARRADIANCIA1

6.1.1. Beneficios del Disco Adaptativo

Concluimos esta sección sumando ideas sobre nuestro algoritmo de selección de direcciones para BRDF arbitrarias que desarrollamos en este capítulo. Es un método optimizado de muestreo exacto de la BRDF basado en la técnica de *rejection sampling*, para su uso en un sistema de síntesis de imágenes realistas por Monte-Carlo, según el estimador presentado en la ecuación (6.5) y el algoritmo [11]. Genera direcciones aleatorias $\mathbf{s}_{xy}^i \in \mathcal{D}^2$, distribuidos según una PDF $q_{\mathbf{u}}$. El uso de la medida de área A en \mathcal{D}^2 es mejor que σ en Ω , puesto que en el primero el término coseno queda implícito, y el segundo requiere dos operaciones de más.

6.2. Muestreo de una BRDF arbitraria

Cuando no se dispone de algoritmos analíticos para el muestreo de la BRDF se utiliza la técnica de *rejection sampling* —véase la sección 4.3.2— debido a su gran simplicidad: muestrear y rechazar hasta que se cumpla una determinada condición. Sin embargo, aplicada a la función BRDF, la probabilidad de que se produzca el rechazo es alta en muchos casos. Esto se debe a la forma habitual de dicha función: suave y poco variante en la mayor parte del dominio, salvo en la zona próxima a la dirección especular, en donde existe una pequeña región que puede presentar altos valores o picos. Únicamente la mencionada región requiere un muestreo más denso que el resto, por lo que es apropiado una técnica no uniforme, para no desperdiciar tiempo de cálculo en muestras situadas en áreas donde la función es poco importante. Con un muestreo uniforme, podemos perder detalles de la función en zonas donde ésta cambia rápidamente, pero no lo detectamos por un número insuficiente de valores aleatorios generados.

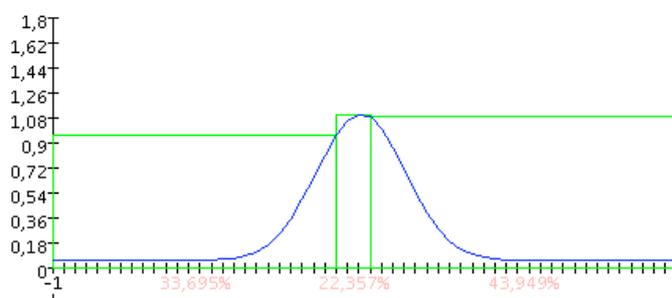


Figura 6.1: División sencilla —casi uniforme— de la BRDF. El área de rechazo (por encima de la función) es muy superior a la de aceptación (por debajo de ésta).

En la figura 6.1 se divide el espacio en tres regiones que se muestrearían de forma uniforme. Esto no es suficiente, ya que cada región no se adapta bien a la forma de la función. Se puede ver que, el área que queda por encima de la función es muy superior al área inferior (la de aceptación). Por consiguiente el porcentaje de muestras rechazadas —1 menos el área de la función— es muy alto. El muestreo puede ser más eficiente —en menos tiempo de cálculo— añadiendo la subdivisión adaptativa a la BRDF, de forma que en aquellas zonas donde la función es más variable se utilice mayor resolución que cuando es más uniforme.

No es posible conocer de antemano el número de divisiones necesarias para aproximar adecuadamente la función. A más intervalos menos tiempo de cálculo usando la técnica de muestreo por rechazo, pero también tenemos un mayor coste computacional en otros puntos del algoritmo. En cada caso se debe evaluar el beneficio que conlleva muestrear a mayor resolución.

En este trabajo no se trata de encontrar el número óptimo de intervalos, sino más bien, que el dominio de cada subintervalo se aproxime adecuadamente a la función que va a muestrear. Nosotros proponemos un método basado en *rejection sampling*, pero con un número acotado de intentos, que es realizado en subintervalos más óptimos. Realizamos la subdivisión en el dominio \mathcal{D}^2 , para ello hacemos uso de una estructura jerárquica cuyo nodo raíz representa la región $[-1, 1]^2$, que irá dividiéndose de forma adaptativa a la función. De esta forma la estructura en árbol, o *quadtree*, aproxima de forma exacta la BRDF. En el siguiente apartado se comenta en mayor detalle.

6.2.1. Subdivisión adaptativa del dominio

Deseamos particionar el dominio original $[-1, 1]^2$ en una serie de regiones $R_1, R_2, \dots, R_n \subseteq \mathcal{D}^2$ disjuntas que cubren por completo el disco \mathcal{D}^2 . Esto puede ser realizado mediante una estructura de datos tipo *quadtree*, que subdivide siempre en cuatro partes por igual. Puesto que estamos en un particionamiento, se deben cumplir las siguientes propiedades:

$$\begin{aligned} R_i \cap R_j &= \emptyset \quad \forall i \neq j, \\ \bigcup_{i=1}^n R_i &= \mathcal{D}^2. \end{aligned}$$

Cada una de estas regiones corresponde a un conjunto de puntos en \mathcal{D}^2 , que, al ser proyectados de nuevo en Ω , siguen cumpliendo la propiedad de cubrir por completo el dominio de la función BRDF. Por definición de la BRDF, ésta vale 0 al ser evaluada en los puntos comprendidos entre $[-1, 1]^2 - \mathcal{D}^2$ (figura 6.2), garantizando la equivalencia entre Ω y \mathcal{D}^2 . Además, debemos asegurar que $q_{\mathbf{u}}$ es positiva antes de emplearla como divisor del estimador X_N . Por tanto, podemos asegurar que la medida de probabilidad para una región R_i es positiva, siempre y cuando la BRDF también lo sea:

$$f_r(\mathbf{u}, \mathbf{v}) > 0 \quad \implies \quad P_{\mathbf{u}}(R) > 0 \quad \wedge \quad q_{\mathbf{u}}(\mathbf{v}) > 0.$$

Así pues $P_{\mathbf{u}}(\mathcal{D}^2) = 1$.

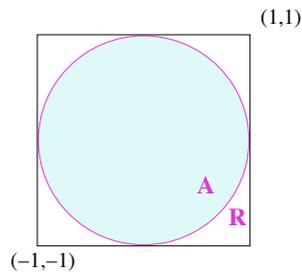


Figura 6.2: La BRDF y la PDF se considera 0 en los puntos fuera del disco de radio unidad.

La construcción del árbol es recursiva, comenzando a dividir el dominio inicial $[-1, 1]^2$ —equivalente al nodo raíz— en cuatro partes iguales en cada paso, como se puede ver en la figura 6.3. Al subdividir se reparte la región actual del nodo entre sus cuatro descendientes, de forma que la unión de los cuatro sucesores pueda proporcionarnos la información del nodo ancestro (se han etiquetado los nodos terminales).

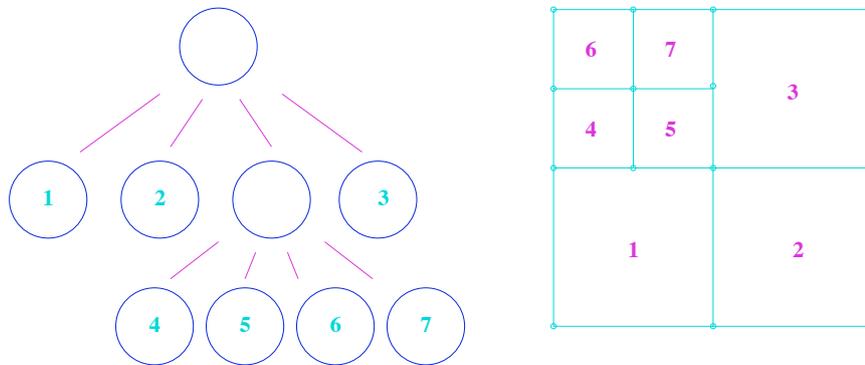
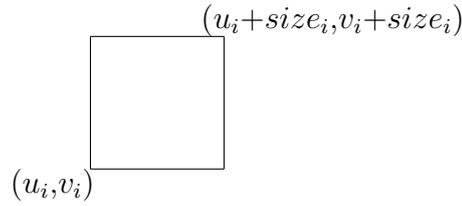


Figura 6.3: División espacial del dominio $[-1, 1]^2$

La información específica a una región, y que es almacenada en cada nodo del *quadtree*, es el conjunto de valores que se indican a continuación:

- Dominio R_i : área representada por $[u_i, u_i + size_i) \times [v_i, v_i + size_i)$, donde (u_i, v_i) es la posición del nodo en la región total y $size_i$ es el lado de la región cuadrada. Gráficamente:



- M_i : máximo de los valores de f_r en la región i -ésima:

$$M_i \stackrel{\text{def}}{=} \max_{(x,y) \in R_i} \{ f_r(\mathbf{u}, \mathbf{v}_{xy}) \}$$

- I_i es la integral de la BRDF en la región i -ésima:

$$I_i \stackrel{\text{def}}{=} \int \int_{x,y \in R_i} f_r(\mathbf{u}, \mathbf{v}_{xy}) dx dy$$

- El volumen total de la región en donde se realiza muestreo por rechazo. Se toma \bar{f}_r como una cota superior constante a trozos de la función BRDF:

$$V_i \stackrel{\text{def}}{=} \int \int_{x,y \in R_i} \bar{f}_r(\mathbf{u}, \mathbf{v}_{xy}) dx dy = \text{size}_i^2 * M_i$$

- La probabilidad de seleccionar la región proyectada R_i que es proporcional a la integral de la BRDF en la región, $P_{\mathbf{u}}(R_i) \propto I_i$:

$$P_{\mathbf{u}}(R_i) \stackrel{\text{def}}{=} \frac{1}{\int_{\mathcal{D}^2} f_r(\mathbf{u}, \mathbf{v}_{xy}) dx dy} \cdot I_i$$

Tanto M_i como V_i pueden ser calculados evaluando la función BRDF en un grid de puntos denso, al tiempo que construimos la jerarquía de nodos. En cualquier caso el único requerimiento impuesto por nuestro algoritmo es ser capaces de evaluar la BRDF dado un par de direcciones.

Para encontrar un conjunto de regiones que se adapten lo mejor posible a la función, necesitamos un criterio de subdivisión que nos diga si se detecta variabilidad y debemos subdividir para que cada subregión busque una mejor aproximación. Esto además, repercute en el muestreo final ya que la probabilidad de aceptación al realizar *rejection sampling* estaría acotada. Haciendo uso de \bar{f}_r , la cota superior constante a trozos de la función BRDF, tendríamos que:

$$\forall R_i \subseteq D \wedge \mathbf{u} \in \Omega \wedge \mathbf{v}_{xy} \in R_i \begin{cases} \bar{f}_r(\mathbf{u}, \mathbf{v}_{xy}) = C, \\ f_r(\mathbf{u}, \mathbf{v}_{xy}) \leq \bar{f}_r(\mathbf{u}, \mathbf{v}_{xy}). \end{cases}$$

El valor máximo permitido para el número medio de intentos al realizar *rejection sampling* en una región R_i , es n_{max} . Este parámetro nos da el criterio necesario para

subdividir cada nodo. Al cumplirse la siguiente desigualdad se debe subdividir:

$$\frac{1}{n_{max}} \leq \frac{I_i}{V_i} \leq 1. \quad (6.6)$$

En nuestro caso hemos elegido un valor pequeño para el número medio máximo de intentos. En concreto $n_{max} = 2$. Así está expresado en el algoritmo [12] que lo implementa.

```

PARTICIONAR( ) →
  Sea  $dx, dy$  diferenciales de área de  $R_i$ 
  Sea  $m = 0, v_1 = 0$ 
  for  $x := u$  to  $u + s$  do begin
    for  $y := v$  to  $v + s$  do begin
       $v_1 += f_r(x, y) * dx * dy$ 
       $m = \max(f_r)$ 
    end
  end
   $v_2 = s * s * m$ 
   $p = v_1 / v_2$ 
  if  $p \leq 0,5$  then begin
    for  $j := 0$  to 4 do begin
       $R_i.Hijo(j) = \text{new SampleQuadtreeNode}$ 
       $R_i.Hijo(j).Particionar()$ 
    end
  end

```

Algorithm 12: pseudocódigo PARTICIONAR

Las clases `SampleQuadtree` y `SampleQuadtreeNode` implementan el árbol que representa de forma adaptativa la función BRDF en el dominio \mathcal{D}^2 mediante un conjunto de regiones que cubren todo el área. Utilizan la clase `DirCoder` para pasar las direcciones del dominio \mathcal{S}^2 a \mathcal{D}^2 y viceversa. El árbol se construye dada una instancia genérica de la función BRDF, la normal al punto considerado, y la dirección incidente $\mathbf{u} \in \Omega$ sobre éste.

En el peor de los casos, si la probabilidad nunca cumple el criterio deseado, el árbol crecería en profundidad sin control posible. Lo cierto es que la profundidad puede controlarse con el valor n_{max} , ya que números altos implican una estructura más pequeña y menos gasto de memoria. Sin embargo ésto aumentaría la probabilidad de rechazo de una muestra.

6.2.2. Muestreo por importancias en \mathcal{D}^2

Una vez que hemos construido el *quadtree* adaptativo, lo usaremos como función de densidad para generar una dirección aleatoria \mathbf{s} en \mathcal{D}^2 proporcional a $f_r(\mathbf{u}, \mathbf{s}) \cos(\mathbf{s})$. Primeramente se verá cómo hacer un recorrido por el árbol hasta alcanzar el nodo hoja —correspondiente a una región— que se empleará en el algoritmo de muestreo.

Sea $C(i, j)$ la función que devuelve un entero correspondiente al índice del nodo hijo j -ésimo, dado un nodo i -ésimo. La función de distribución acumulada $F(R_i)$, para el nodo que representa la región R_i , se utiliza en el descenso del árbol. Representa la probabilidad condicional de descender por uno de los nodos hijos, supuesto que hemos alcanzado el padre, hasta llegar a un nodo hoja.

$$F_{i,k} = \frac{\sum_{j=0}^k I_{C(i,j)}}{\sum_{j=0}^3 I_{C(i,j)}}.$$

Partiendo del objeto raíz y tomando r , un valor aleatorio uniformemente distribuido, cada nodo decide por donde continuar seleccionado el nodo hijo que cumpla la siguiente condición:

$$\text{si } F_{i,k-1} < r \leq F_{i,k} \text{ se selecciona el nodo } k,$$

tal y como se muestra en el algoritmo [13].

Finalmente, con este paso nos quedamos con una región (figura 6.4), la cual se ha seleccionado con una probabilidad proporcional a su valor I_i y que se corresponde a la zona de la BRDF que necesita ser muestreada con mayor densidad.

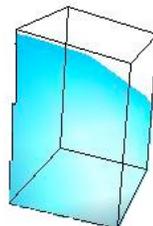


Figura 6.4: Vista en 3D de la región seleccionada mostrando el volumen encerrado.

```

SELECCIONARREGIONALEATORIA ( ) → SampleQuadtreeNode
  if ( $R_i$  es hoja) then
    return  $R_i$ 
  else
    Generar  $r \sim$  uniforme  $[0,1]$ 
    Sea  $n$  un SampleQuadtreeNode nulo
    Sea  $S = \sum R_i.Hijo(j).v_1$  para  $j = 1 \dots 4$ 
    Sea  $F = 0$ 
    for  $j := 1$  to 4 do begin
      SampleQuadtreeNode  $m = Hijo(j)$ 
      if ( $F < r$ ) Y ( $r \leq F + m.v_1/S$ ) then
         $n = m$ 
      else
         $F += m.v_1/S$ 
      end
    end
    return  $n.SeleccionarRegionAleatoria()$ 
  end

```

Algorithm 13: pseudocódigo SELECCIONARREGIONALEATORIA

6.2.3. Generación de direcciones

Hemos visto que realizando un recorrido por el *quadtree* se selecciona de forma aleatoria una región R_i . El siguiente paso consiste en muestrear dentro de ese dominio para obtener un vector de salida cumpliendo $\mathbf{s} \sim q_{\mathbf{u}}$.

Para obtener este valor aleatorio usamos *rejection sampling*. Recordemos que mediante esta técnica, para muestrear siguiendo $p(x)$, se utiliza otra función de densidad más conveniente $q(x)$. Para esta segunda función, debemos conocer la constante M , que cumpla lo siguiente:

$$p(x) \leq M q(x).$$

En nuestro algoritmo $p(x)$ es la BRDF y $M q(x)$ es la función cota \bar{f}_r . El procedimiento a seguir —método **Muestrear**—, es similar al necesario para muestrear en un disco inscrito en un cuadrado unidad, que ya ha sido comentado en el algoritmo [2] (página 81). Tan sólo es necesario cambiar los límites del cuadrado unidad por los pertenecientes a la región importante previamente seleccionada. El último paso es simple, pues consiste en realizar la transformación inversa pasando de una dirección en \mathcal{D}^2 a una en \mathcal{S}^2 .

Hacemos uso de la clase **Sample** para almacenar, junto a cada dirección generada con este método, el valor de la función peso asociado a cada muestra \mathbf{s} :

$$w_{\mathbf{u}}(\mathbf{s}) = \begin{cases} (\mathbf{s} \cdot \mathbf{n})/p_{\mathbf{u}}(\mathbf{s}) & \text{para la PDF tipo } p_{\mathbf{u}}, \\ 1/q_{\mathbf{u}}(\mathbf{s}) & \text{para una PDF tipo } q_{\mathbf{u}}. \end{cases}$$

tras consultar al nodo del *quadtree* por el valor de la PDF para $\mathbf{s} \in R_i$.

```

OBTENERNMUESTRAS( $\mathbf{u} \in \Omega, \mathbf{n} \in \Omega, f_r, \mathbb{N}^+ N$ )  $\rightarrow$  Sample[]
Sea qt = new SampleQuadtree( $\mathbf{u}, \mathbf{n}, f_r$ )
Sea sample un vector de  $N$  instancias de Sample
for  $i = 1$  to  $N$  do begin
    SampleQuadreeNode region = qt.SeleccionarRegionAleatoria()
    Sea  $\mathbf{s} =$  region.Muestrear()
    Sea  $q =$  region.ObtenerValorPDF()
    sample[i] = new Sample( $\mathbf{s}, 1.0/q$ )
end
return sample
    
```

Algorithm 14: pseudocódigo OBTENERNMUESTRAS

Con esto, nuestro método de muestreo de la BRDF se integra en el estimador X_N tal y como se presenta en el algoritmo [15]. El estimador le ira solicitando valores aleatorios, así como sus pesos asociados (ver algoritmo [14]).

```

ESTIMARRADIANCIA2( $\mathbf{u} \in \Omega, \mathbf{n} \in \Omega, N \in \mathbb{R}^+, f_r, L_i, P_{\mathbf{u}}$ )  $\rightarrow$   $\mathbb{R}$ 
Sea  $w_{\mathbf{u}}$  la funcion peso asociada a  $P_{\mathbf{u}}$  en  $\mathcal{D}^2$ 
Sea  $\mathbf{s}[N] =$  ObtenerNMuestras( $\mathbf{u}, \mathbf{n}, f_r, N$ )
Sea  $s = 0$ 
for  $i := 1$  to  $N$  do begin
     $s += f_r(\mathbf{u}, \mathbf{s}[i]) \cdot L_i(\mathbf{s}[i]) \cdot w_{\mathbf{u}}(\mathbf{s}[i])$ 
end
return  $s/N$ 
    
```

Algorithm 15: pseudocódigo ESTIMARRADIANCIA2

6.2.4. Comparación directa con otros métodos

Es interesante obtener una descripción local del comportamiento de nuestro método de selección de direcciones, al igual que para otros métodos. Tomemos la herramienta *BRDF 3D Plot* (para más detalles consultese el apartado 7.10) mediante la cual podemos visualizar el resultado de solicitar N direcciones aleatorias dado un punto de la superficie, su normal, la posición del observador y la BRDF que se muestrea. Los vectores resultantes se dibujan como puntos de colores en 2D y en 3D, de forma que,

CAPÍTULO 6. Muestreo eficiente de BRDFs genéricas

podemos observar si la distribución que siguen esos vectores se asemeja a la forma de la función BRDF.

En las figuras 6.5, 6.6, 6.7 y 6.8 tenemos resultados para una misma situación: punto, normal, vector incidente y función de reflectancia dada mediante una instancia de la BRDF de Ward (con los parámetros de la tabla 6.18, pág. 163). Tan solo variamos los métodos de selección de dirección que empleamos para solicitar 2500 valores aleatorios.

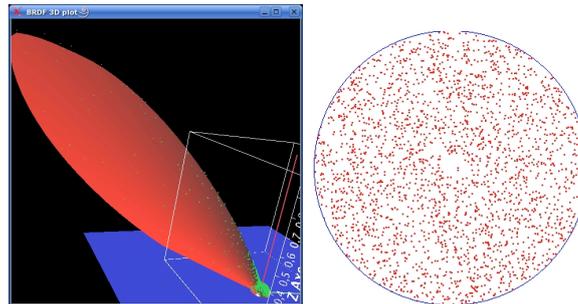


Figura 6.5: Distribución de puntos dada por el método uniforme. La componente especular no recibe suficientes muestras.

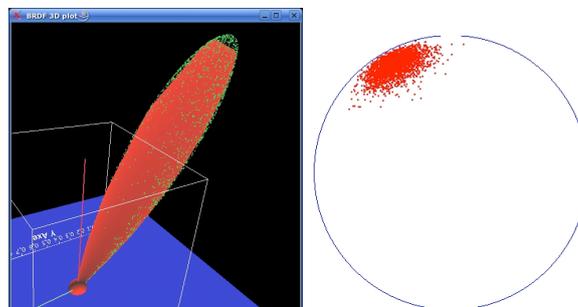


Figura 6.6: Distribución de puntos dada por el método del lóbulo coseno. No se muestrea la componente difusa.

Como puede verse, los métodos uniforme y los basados en el lóbulo coseno se centran únicamente en una componente concreta de la función. No obstante nuestro método adaptativo consigue combinar ambas, muestreando la componente uniforme y suministrando más muestras de forma localizada para muestrear la parte especular de la función BRDF. El muestreo basado en la factorización de la BRDF produce demasiados vectores de dirección en la semiesfera negativa, por lo que muchos vectores aleatorios son descartados y no son empleados en el estimador.

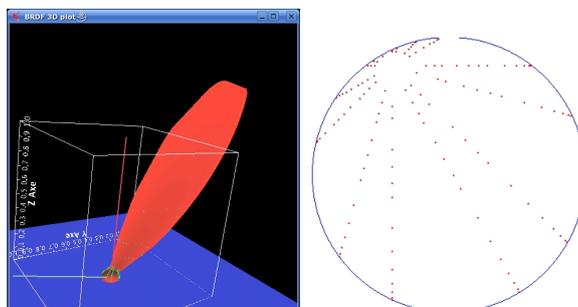


Figura 6.7: Distribución de puntos no rechazados dada por el método factorizado (a baja resolución).

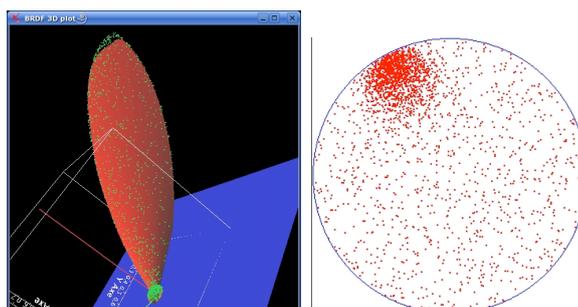


Figura 6.8: La distribución de direcciones dada por nuestro método cubre la parte difusa y especular de la función.

6.3. Optimización del método

6.3.1. Precálculo de varios *quadtrees*

Por motivos de eficiencia es interesante considerar el precálculo de cada *quadtree* para cada instancia de BRDF en uso, y con respecto a una discretización de la dirección incidente. En nuestro caso, tomamos BRDFs isótropas y de esta forma es posible generar valores discretos sólo en θ considerando el plano incidente con $\phi_{\mathbf{u}} = 0$. Sin embargo, es necesario aplicar una transformación al vector 3D resultante del muestreo que lo deposite en el cuadrante apropiado con respecto a la dirección incidente real. Esta operación puede verse en el pseudocódigo número [16]. Para BRDFs anisótropas el precálculo se debe realizar en base a valores discretos de los ángulos θ y ϕ .

Al igual que hicimos con la expresión en GRF de los modelos de BRDF y otras PDFs que se reconocen en nuestro sistema de síntesis, damos la expresión GRF de nuestro método. Los parámetros de BPT `BPT_QT_Precomputed` y `BPT_QT_Anisotropic`, definen la discretización según los ángulos θ y ϕ respectivamente.

`ADAPTIVEPDF ::=`
`pdf adaptive nQT =valor nAnisoQT =valor`

Tabla 6.1: Notación en GRF para ADAPTIVEPDF

<p><code>MAPEOALCUADRANTE</code> ($\mathbf{u} \in \Omega, \mathbf{s} \in \Omega$) \rightarrow <code>LocalS</code> $\in \Omega$</p> <p><i>if</i> ($\phi_{\mathbf{u}}$ en primer cuadrante) <i>then</i> <i>return</i> \mathbf{s}</p> <p><i>if</i> ($\phi_{\mathbf{u}}$ en el segundo cuadrante) <i>then</i> <i>return</i> $(-x_{\mathbf{s}}, y_{\mathbf{s}}, z_{\mathbf{s}})$</p> <p><i>if</i> ($\phi_{\mathbf{u}}$ en el tercer cuadrante) <i>then</i> <i>return</i> $(-x_{\mathbf{s}}, -y_{\mathbf{s}}, z_{\mathbf{s}})$</p> <p><i>if</i> ($\phi_{\mathbf{u}}$ en el cuarto cuadrante) <i>then</i> <i>return</i> $(x_{\mathbf{s}}, -y_{\mathbf{s}}, z_{\mathbf{s}})$</p>

Algorithm 16: pseudocódigo MAPEOALCUADRANTE

El error cometido por este precálculo se corrige tomando un número alto de valores discretos uniformes. En nuestras pruebas hemos usado 90 y 60 valores de $\theta_{\mathbf{u}}$. Si bien valores por debajo de éstos disminuirían el tiempo inicial de construcción de estas estructuras no podríamos garantizar localizar una dirección discreta próxima al \mathbf{u} dado.

6.3.2. Generación múltiple de direcciones

En el uso de un estimador de Monte Carlo, la evaluación de una función de interés es repetida un número determinado de veces, como hemos visto en el algoritmo [15] (página 133). Es necesario, por tanto, disponer de un método de generación de direcciones rápido.

Una mejora clara a nuestro procedimiento consiste en obtener N valores aleatorios en una única llamada. Este objetivo puede conseguirse mediante la paralelización del descenso de la estructura de datos –en nuestro caso un árbol *quadtree*– hasta el nodo hoja que realiza el muestreo.

Tal y como se muestra en el algoritmo [14], es necesario recorrer el *quadtree* N veces para generar N direcciones aleatorias. La parte de proceso que podemos agilizar consiste en realizar un único recorrido del árbol que localice N puntos de muestra. Tras lo cual cada región generará un número determinado de direcciones y de esa forma el procedimiento devolvería las N muestras aleatorias.

Sea la secuencia de M elementos l_1, l_2, \dots, l_M , el conjunto de nodos hoja del árbol *quadtree* correspondientes a las regiones $R_1, R_2, \dots, R_n \subseteq \mathcal{D}^2$. Sea N el número de muestras a generar, debemos calcular los M valores de n_i que cumplan:

$$n_1 + n_2 + \dots + n_M = N.$$

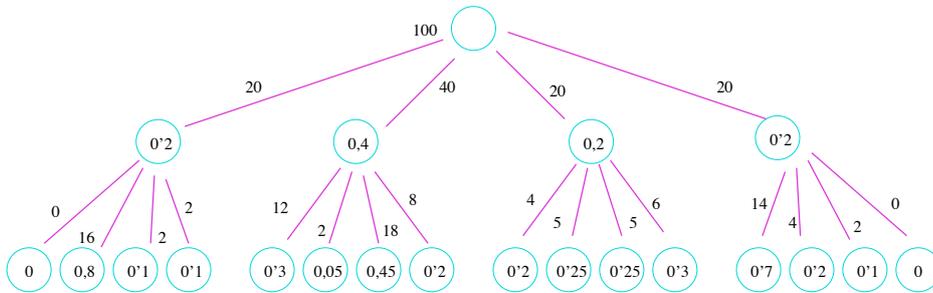


Figura 6.9: Caso sencillo en el que se muestra el proceso iterativo del reparto de 100 muestras. Los nodos contienen la probabilidad y las aristas muestran el número de muestras asignado.

Un ejemplo sencillo de esta asignación puede verse en la figura 6.9. Otra propiedad clara es que los valores de n_i deben ser enteros. Puesto que nuestro método no es un muestreo rigurosamente estratificado sino adaptativo, el valor medio de n_i debe ser, en media, proporcional a la probabilidad de cada nodo. Así pues $n_i \stackrel{\text{def}}{=} P_{\mathbf{u}}(l_i) * N$. Sin embargo estos valores son reales y el mero hecho de realizar una conversión a entero no resuelve el problema. De forma general, sucederá lo siguiente:

$$n_1 + n_2 + \dots + n_M \leq N,$$

$$\delta = N - \sum_{i=1}^M \lfloor P_u(l_i) * N \rfloor.$$

De lo que se deduce que δ es el número entero de muestras —el resto— que es necesario volver a repartir. Considerando de nuevo las probabilidades de cada nodo hoja y mediante un proceso iterativo el resto se hace cero. Finalmente cada nodo l_i conoce el número de muestras n_i que debe generar.

El reparto de un número N de forma proporcional no requiere que las probabilidades p_i estén normalizadas, siempre que no sean todos valores nulos. El procedimiento se muestra en los algoritmos [17] y [18] que se repiten recursivamente desde el nodo raíz hasta los nodos hoja.

```

REPARTEN(N) → N+ n[4]
  Sean las matrices p y q ∈ ℝ[4]
  Inicializar δ a N
  for i=0 to 3 do begin
    p[i] = Pu(Hijo[i]) * N
    n[i] = ⌊ p[i] ⌋
    δ = δ - n[i] ;
  end
  if ( δ > 0 ) then
    for i=0 to 3 do begin
      q[i] = p[i] - n[i] ;
    end
    do
      n[SeleccionaI(q)]++ ;
      δ = δ - 1
    while ( δ ≠ 0 ) ;
  end
  return n

```

Algorithm 17: pseudocódigo REPARTEN

6.3.3. Evaluación eficiente de la BRDF

Cada vez que hacemos uso del método adaptativo, por ejemplo al solicitar el peso de una muestra, hacemos uso de la evaluación de la BRDF para dos direcciones en la semiesfera. Este tipo de llamadas son aún más numerosas en la construcción de una estructura *quadtree*, ya que debemos calcular la integral de la BRDF para saber

```

SELECCIONAI( $\mathbb{R} \mathbf{q}[4]$ )  $\rightarrow \mathbb{N}^+$ 
 $\mathbb{R} \xi \sim U(0,1)$ 
 $\mathbb{R} r = \xi \cdot (\mathbf{q}[0] + \mathbf{q}[1] + \mathbf{q}[2] + \mathbf{q}[3])$ 
 $\mathbb{R} s = 0.0$ 
for  $i=0$  to 2 do begin
     $s = s + \mathbf{q}[i]$  ;
    if ( $r \leq s$ ) then
        return  $i$  ;
end
return 3;

```

Algorithm 18: pseudocódigo SELECCIONAI

si dividir un nodo. Puesto que es posible precalcular y almacenar estas estructuras, también podemos predecir y calcular las evaluaciones que se realizan de la función de reflectancia para un caso genérico. Proponemos una estructura de datos que permita construir el árbol con un planteamiento similar a las técnicas *bottom-up*.

Considerando que en cada región se debe aproximar mediante Monte-Carlo la siguiente integral:

$$\int_{x,y \in R_i} f_r(\mathbf{u}, \mathbf{v}_{xy}) d(x) d(y),$$

es importante el ahorro que se produciría al consultar el valor de la BRDF de una tabla precalculada para valores de x e y contenidos en el plano $[-1, 1]^2$.

Asociado a cada *quadtrees* se ha implementado, pues, un *mapa de imagen* de la BRDF que es evaluada en todo el dominio $[-1, 1]^2$. La BRDF evalúa a cero para direcciones fuera de \mathcal{D}^2 . Solicitar el valor $f_r(\mathbf{u}, \mathbf{v}_{xy})$, conlleva calcular las coordenadas del punto relativas a una imagen de proporciones cuadradas, que son equivalentes a la posición del vector:

$$\text{Imagen}(\mathbf{u}, i_x, j_y) \simeq f_r(\mathbf{u}, \mathbf{v}_{xy}).$$

La correspondencia entre vector proyectado y punto de imagen es trivial. Lo más importante para la consecución fructífera de esta aproximación, es que la resolución tomada en la imagen nunca sea inferior a la densidad de muestreo que un nodo hoja pueda requerir para el cálculo de la integral de la BRDF en dicha región.

6.3.4. Ajuste de la región para el muestreo por rechazo

Para reducir los tiempos de muestreo en nuestro método, nos hemos propuesto optimizar todos los aspectos básicos del algoritmo. Nuestro método recae en la

CAPÍTULO 6. Muestreo eficiente de BRDFs genéricas

técnica conocida como muestreo por rechazo (sección 4.3.2), que se caracteriza por iterar un bucle hasta que el valor generado se acepta. La forma óptima de esta técnica minimiza el número de iteraciones, o lo que es lo mismo, el área de rechazo.

Por la relación existente entre un cuadrado y un círculo inscrito en él, las zonas de las esquinas aumentan la región de rechazo. Si bien las cuatro esquinas son las zonas más claras para optimizar, no son los únicos casos hallados, y dependerá en gran medida del nivel de subdivisión máximo. En lugar de tomar todas las regiones por igual y muestrear usando el área por defecto de la región $(u, v) \times (u + size, v + size)$ (en adelante notaremos (U, V) al punto $(u + size, v + size)$), será necesario ajustar los límites a otros más óptimos. Esta situación es la observada en la figura 6.10, en la cual tras encontrar el punto (x, y) se aprecia que la región $(x, y) \times (U, V)$ disminuye el área de rechazo.

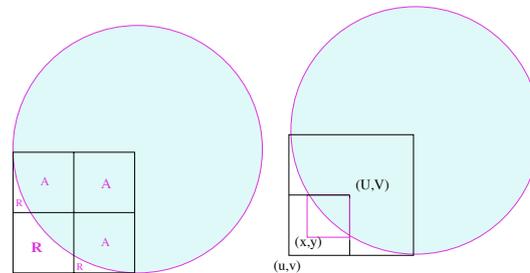


Figura 6.10: Es necesario reducir el área de rechazo en determinadas regiones del disco.

El procedimiento para decidir si una región debe recalcular sus límites comienza calculando la distancia de cada uno de sus vértices al origen del disco.

- $A^2 = u^2 + v^2$, y de aquí $b_0 = \begin{cases} 1 & \text{si } A \leq 1, \\ 0 & \text{en otro caso.} \end{cases}$
- $B^2 = U^2 + v^2$, y de aquí $b_1 = \begin{cases} 1 & \text{si } B \leq 1, \\ 0 & \text{en otro caso.} \end{cases}$
- $C^2 = u^2 + V^2$, y de aquí $b_2 = \begin{cases} 1 & \text{si } C \leq 1, \\ 0 & \text{en otro caso.} \end{cases}$
- $D^2 = U^2 + V^2$, y de aquí $b_3 = \begin{cases} 1 & \text{si } D \leq 1, \\ 0 & \text{en otro caso.} \end{cases}$

Con los cuatro valores lógicos $b_0b_1b_2b_3$ obtenemos un código binario que identifica el tipo de región. En la figura 6.11 se muestra un caso sencillo (la profundidad máxima del árbol es 3) con los correspondientes códigos de interés en decimal.

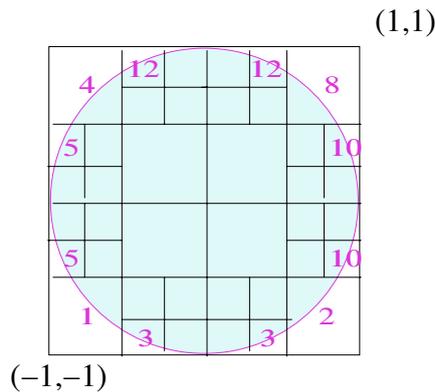


Figura 6.11: Se han localizado y codificado las regiones que deben ajustar el área de muestreo.

Existen, por tanto, ocho variantes que optimizan el dominio $(u, v) \times (U, V)$. Cada uno de ellos modifica el área de la región considerando los siguientes valores:

$$\begin{aligned} x &= -\sqrt{1-v^2} & y &= -\sqrt{1-u^2} \\ X &= -\sqrt{1-V^2} & Y &= -\sqrt{1-U^2} \end{aligned}$$

- 0001 (1) : toma la región $(x, y) \times (U, V)$
- 0010 (2) : toma la región $(u, y) \times (X, V)$
- 0011 (3) : toma la región $(u, y) \times (U, V)$
- 0100 (4) : toma la región $(x, v) \times (U, Y)$
- 0101 (5) : toma la región $(x, v) \times (U, V)$
- 1000 (8) : toma la región $(u, v) \times (x, y)$
- 1010 (10): toma la región $(u, v) \times (x, Y)$
- 1100 (12): toma la región $(u, v) \times (U, Y)$

Con todas estas mejoras, los tiempos de muestreo para el nuestro método adaptativo han disminuido en un alto valor con respecto a los primeros resultados. El porcentaje de reducción se ha sido algo mayor del 90% ya que en la Escena 1 se ha bajado de 229,58 segundos a 8.2 y en la Escena 2 los tiempos han pasado de 525,73 en promedio a 40,36 segundos.

6.4. Evaluación del método y resultados

6.4.1. Evaluación de la convergencia

Puesto que el método aquí presentado es en concreto un método para obtener puntos de muestra según una función de densidad de probabilidad, debemos asegurar que el algoritmo converge a la función deseada. Básicamente necesitamos generar un número alto de muestras aleatorias con nuestro método y encontrar una proporción numérica que equivalga a la función objetivo: en este caso la integral de la BRDF evaluada en un punto de la superficie.

Sea m el número total de direcciones que se generan, a continuación contamos el número n_i de muestras pertenecientes a la i -ésima región cubierta por la estructura *quadtree*. Una vez encontrada la integral de la BRDF en cada región I_i , debemos comprobar la siguiente proporción:

$$I_i \propto \frac{n_i}{m}.$$

Existe un valor constante fijo a que cumple $\frac{n_i/m}{I_i} = a$. Y extendiendo esta sentencia al conjunto total de N regiones obtenemos el valor final de la integral:

$$I = \sum_{i=1}^N I_i = \frac{n_1/m}{a} + \dots + \frac{n_N/m}{a} = \frac{1}{a}.$$

Así como m se va haciendo mayor, el método prueba la convergencia al valor constante a .

$$\lim_{m \rightarrow \infty} \frac{n_i/m}{I_i} = a.$$

Usando la anterior propiedad, podemos verificar que la implementación es válida

6.4.2. Descripción del procedimiento de evaluación

Llegados a este punto procede hacer una evaluación práctica del método. Se ha implementado un sistema de síntesis realista, basado en el algoritmo básico de *path tracing* bajo la plataforma Linux. Da como salida una imagen que puede ser almacenada en varios formatos gráficos conocidos, como son: JPG, TGA, PMG, EXR (de *Industrial Light & Magic*) y RAW. El formato RAW es utilizado para las comparaciones numéricas entre imágenes ya que no tiene pérdidas y no sufre modificaciones no lineales por *tone mapping*. El equipo usado en las pruebas es un AMD Opheron con 2GB de RAM.

Para la evaluación del método adaptativo de muestreo de BRDF genéricas, se procederá a generar imágenes para cada modelo de BRDF implementado. En total se han usado 16 modelos distintos. Éstos pueden ser consultados en la tabla 6.2.

6.4. Evaluación del método y resultados

Ashikhmin	Beard-Maxwell	Blinn	Coupled
He-Torrance	Lafortune	Lewis	Minnaert
Oren-Nayar	Phong	Poulin-Fournier	Schlick (single)
Schlick (double)	Strauss	Cook-Torrance	Ward

Tabla 6.2: Conjunto de BRDFs que es empleado en las pruebas para la evaluación del algoritmo

Igualmente otros parámetros que varían en las pruebas son el número de muestras generado, y el procedimiento de muestreo. Ambos listados en la tabla 6.3. Así, pues, para cada experimento —escena a evaluar— se toman en unos casos $16 \times 4 \times 9 = 576$ imágenes y en otros $16 \times 5 \times 9 = 720$ imágenes.

Adicionalmente, para la generación de una imagen se tomaran las siguientes consideraciones:

- La calidad con la que se genera cada imagen viene determinada por el número de muestras empleados en el estimador de Monte-Carlo. En nuestro caso este número es expresado como la raíz cuadrada del número total de valores aleatorios que son requeridos.
- El tamaño de imagen para las comparativas entre el método presentado y otras funciones de densidad, es 256×256 píxeles. Para otro tipo de imágenes que ilustren los beneficios del algoritmo se toman valores de resolución mayores.
- Para evitar *aliasing* se puede indicar como parámetro la resolución a la que se muestrea el área del píxel. Para las comparativas se ha tomado 1×1 muestras; para el resto de imágenes el valor más frecuente corresponde a 2×2 .
- El muestreo dentro del píxel se puede realizar siguiendo una distribución radial basada en la circunferencia, o bien, aleatoriamente según una distribución uniforme.
- Se toman varias medidas de tiempo: la de inicialización del método adaptativo, la de muestreo, y la de tiempo total de generación o *rendering*. Este último incluye el tiempo de muestreo y contabiliza el esfuerzo producido por el cálculo de la intersección rayo-escena. En general sólo necesitamos considerar los tiempos de generación de direcciones para la comparativa, ya que el sistema de *rendering* es idéntico para todos los casos y únicamente cambiará el método de selección de direcciones —además de la BRDF que esté siendo aplicada—.
- En el caso de las PDFs basadas en el lóbulo coseno (dos son aquí las usadas) es necesario indicar como parámetro el valor del exponente del lóbulo. Si no se indica, se tomará un valor constante, igual a 30.

CAPÍTULO 6. Muestreo eficiente de BRDFs genéricas

Uniforme	Lóbulo en \mathcal{S}^2	1^2	5^2	10^2	15^2
Lóbulo en Ω	Adaptativa en Disc	20^2	30^2	40^2	50^2
Factorización de la BRDF		500^2	1000^2		

Tabla 6.3: El método propuesto se compara con otras funciones PDF para cada instancia de BRDF a considerar y con un número variable de muestras.

La imagen resultante con el valor más alto de muestras (500^2 o 1000^2 dependiendo de la escena) es utilizada para generar la imagen de referencia. Promediando la imagen más próxima a la solución real, para cada BRDF y para cada PDF, obtenemos una única solución resultante de la convergencia de los métodos de muestreo utilizados. Este procedimiento se hace manualmente una vez obtenidas todas las imágenes del experimento. A partir de ahí, podemos comparar cada una de las imágenes individuales —con distinto número de muestras, obtenidas con una PDF particular, y para una BRDF dada— con la imagen de referencia. El error relativo medio obtenido al comparar cada uno de los píxeles de radiancia no nula, con el correspondiente en la referencia, es un dato que utilizaremos para medir cuantitativamente la bondad de un método de selección aleatoria de direcciones.

6.4.3. Coste computacional del preprocesamiento asociado al método adaptativo

Como ya hemos mencionado en la sección 6.3.1, nuestro algoritmo realiza varios cálculos en orden de ajustar lo más próximo posible cualquier función BRDF. Esta parte se realiza al comienzo y antes del uso corriente del método como parte del estimador de Monte-Carlo de la ecuación de la radiancia.

Un número determinado de estructuras *quadtree*, correspondientes a una serie de valores discretos del ángulo entre la normal y la dirección incidente, son creadas una única vez por cada instancia distinta de BRDF que contenga la escena. Cuanto mayor sea el número de discretizaciones tomadas sobre el ángulo incidente, mejor será la aproximación. Así pues, se han tomado para la comparación del método con otras PDFs un total de 90 *quadtrees*, equivalente a una resolución del ángulo incidente de unidad expresada en grados. En algunos casos hemos usado un número de 60, pero para que el beneficio —referente a la medida de error— del método persista, no es conveniente tomar cantidades menores de las citadas. No se han empleado cantidades mayores, por considerar suficiente la reconstrucción de la PDF dado un ángulo. De esta forma se mantiene un punto de equilibrio entre calidad y coste computacional.

De forma similar a nuestro método, la factorización de la BRDF dada por Lawrence hace uso de un precálculo. Hemos evaluado el tiempo requerido por ambos

6.4. Evaluación del método y resultados

BRDF	Adaptativa Disco		Factorización
	60QTs	90QTs	Param. manual
Ashikhmin	57,61	83,46	3,12
BeardMaxwell	49,08	74,75	102,9
Blinn	32,12	45,08	565,7
Coupled	51,25	76,17	3,21
He	300,8	452,7	8,00
Lafortune	32,33	40,34	312,5
Lewis	32,49	48,86	312,4
Minnaert	35,33	54,28	3,17
Oren-Nayar	28,99	40,99	3,08
Phong	29,99	45,69	312,3
Poulin-Fournier	112,7	160,9	8,09
Schlick (D)	55,17	82,49	3,24
Schlick (S)	41,65	63,03	8,30
Strauss	38,82	56,86	15,36
Cook-Torrance	38,84	58,43	2,97
Ward	33,57	48,99	26,62

Tabla 6.4: Tiempos (en segundos) de inicialización del algoritmo en donde se realizan los precálculos que optimizan su uso. Los datos son referentes a Escena2.

métodos bajo diferentes situaciones —las explicadas en las subsiguientes secciones de resultados— obteniendo los datos de la tabla 6.4. En ella se muestra información referente al coste en segundos de nuestro precálculo para tamaños de 60 y 90 *quadrees*, así como distintas factorizaciones cuyos parámetros se han ajustado manualmente. La diferencia computacional se aprecia no sólo entre los métodos aplicados, sino que es más aparente entre modelos de BRDFs, constatando la diferencia entre los modelos de reflexión experimentales frente a los teóricos.

En promedio obtenemos los valores de 62,58 y 94,69 segundos en el precálculo de 60 y 90 *quadrees* respectivamente, frente a los 105,69 segundos resultante de las diferentes factorizaciones para cada BRDF. Para ambos métodos resultará costoso un número elevado de BRDFs en la escena, sin embargo, en nuestro caso el tiempo de precálculo total será algo mayor de 1 segundo por BRDF y *quadtree*, por lo que podremos ajustar el número de estructuras jerárquicas a un valor final razonable de tiempo.

6.4.4. Requisitos de memoria asociados al método adaptativo

Para evaluar la eficiencia de un método de muestreo aplicado a algoritmos basados en Monte Carlo, utilizamos como medidor fundamental la variancia. Sin embargo otros aspectos igualmente importantes deben ser considerados, entre los cuales podemos citar los siguientes: la velocidad de muestreo, la optimización de recursos como la

CAPÍTULO 6. Muestreo eficiente de BRDFs genéricas

BRDF	nmax - Adaptativo en disco			
	1,3	2,0	2,5	3,0
Ashikhmin	1188,3	0,103	0,103	0,103
Beardmax	1412,8	28,710	12,360	230,332
Blinn	522,410	6,914	0,103	0,103
Coupled	322,969	0,103	0,103	0,103
He	1346,3	28,711	10,184	0,103
Lafortune	2344,6	30,891	11,273	111,539
Lewis	4701,9	40,699	12,363	304,441
Minnaert	974,695	28,711	9,094	8,549
Oren	0,103	0,103	0,103	0,103
Phong	2455,7	21,082	6,914	115,898
Poulin	308,801	6,914	0,103	0,103
Schlick (D)	522,410	9,094	0,103	0,103
Schlick (S)	522,410	9,094	0,103	0,103
Strauss	380,730	13,453	6,914	15,633
Torrance	728,391	0,103	0,103	0,103
Ward	340,406	8,004	4,734	13,453
Promedio	1129,5	14,54	4,67	50,05

Tabla 6.5: Necesidades de memoria en KB de almacenamiento en nuestro método de un *quadtree*, considerando variaciones en el parámetro real `nmax`.

memoria y por supuesto la facilidad de implementación para aplicarlo en otros entornos.

Habiendo descrito en anteriores secciones el funcionamiento de los procesos fundamentales del método adaptativo y el coste del precálculo, pasamos a evaluar los requerimientos de memoria. Primeramente consideramos los parámetros que afectan a la profundidad de una estructura jerárquica, y por consiguiente, al número de nodos que se almacenan. Esto se realiza modificando el parámetro `nmax` utilizado en el criterio de subdivisión del árbol (ver n_{max} en la desigualdad de la ecuación 6.6, pág. 130), y cuyo resultado se muestra en la tabla 6.5. Se observa que un valor de 1,3 —correspondiente al número en promedio de reintentos al realizar muestreo por rechazo— es muy restrictivo e impone subdividir cada nodo hasta alcanzar un alto grado de profundidad. Por el contrario, un valor de 3 es muy poco restrictivo y hace que el nodo raíz casi nunca se subdivide. Es por tanto que hemos encontrado el valor óptimo de `nmax=2` y lo empleamos de aquí en adelante.

Al igual que ya hicimos en el apartado anterior, vamos a comparar nuestro método cuantitativamente con el método de factorización de la BRDF de Lawrence, ya que al igual que nosotros, se precalcula una estructura de datos para su posterior uso. Los datos relativos a la comparativa se muestran en la tabla 6.6. Se han tomado como parámetros de la PDF Adaptativa un conjunto de 60 direcciones incidentes distintas para la creación de estructuras *quadtree* y el valor `nmax=2`. El método de Lawrence toma para la factorización de cada BRDF la mejor parametrización encontrada, tal y

6.4. Evaluación del método y resultados

BRDF	Adaptativa	Mejor Fact.
Ashikhmin	6,25	1031
BeardMax.	1713,25	6454
Blinn	582,25	6481
Coupled	6,25	1033
He	2407,25	1034
Lafortune	1275,25	6445
Lewis	1279,25	6445
Minnaert	1461,25	1031
Oren	6,25	1033
Phong	1279,25	6445
Poulin	297,25	1038
Schlick (D)	342,25	1033
Schlick (S)	780,25	1043
Strauss	727,25	1052
Torrance	631,25	1029
Ward	483,25	1038

Tabla 6.6: Necesidades de memoria en KB de los métodos de selección de direcciones Adaptativa en Disco y Factorización de la BRDF. Ambos métodos son genéricos, en cuanto a la BRDF, y precálculan datos en un paso previo a su uso.

como se lista en la Tabla 6.19 (en la página 163).

Se observa en la mayoría de los casos que el uso de memoria del método que nosotros presentamos en este trabajo, se encuentra muy por debajo de las necesidades del método factorizado. En los siguientes apartados se comprobará, además, que con los mismos parámetros —y en ambos casos— el error o varianza cometido por nuestro método es inferior a la PDF de Lawrence.

6.4.5. Escena 1: *The Cornell Box*

La caja de Cornell o *Cornell Box*¹ es una escena muy característica en la evaluación de los métodos de iluminación global. Por ello se ha considerado como primer experimento práctico en la evaluación del método adaptativo de muestreo. Corresponde a una habitación con tres paredes de colores, techo, suelo y una luz de área extendida, que es descrita usando notación GRF. En la figura 6.13 el gráfico de la derecha presenta la situación inicial con la escena GRF.

Los datos relativos a tiempos medios de muestreo así como error relativo con respecto a la imagen de referencia, pueden ser consultados en las tablas 6.7 y 6.8. De

¹ <<http://www.graphics.cornell.edu/online/box/>> [consultado el 18-02-2007].

forma alternativa a la información numérica, se muestra en el gráfico 6.12 la comparativa de los distintos métodos de muestreo evaluados, incluyendo el propuesto en esta tesis.

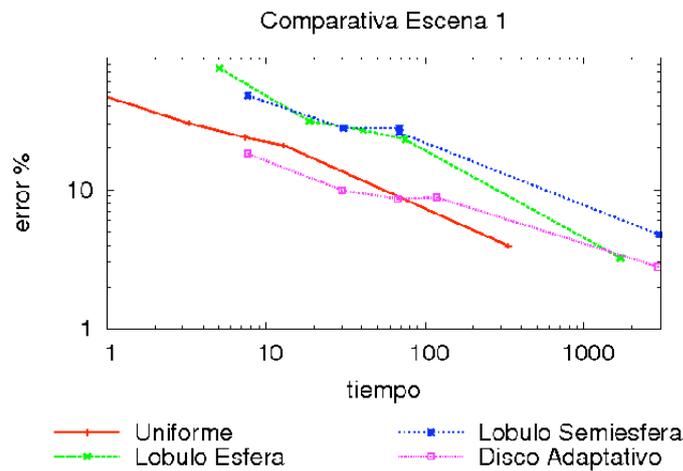


Figura 6.12: Gráfica en la que se relacionan los resultados de los distintos métodos de muestreo probados, en cuanto a relación error tiempo. Cuanto mayor sea el tiempo utilizado, la imagen resultante converge a la imagen de mínimo error.

El resultado está directamente relacionado con la característica de la escena en sí. Predominan las interreflexiones difusas y no se producen situaciones de variabilidad en la radiancia incidente. Este aspecto difuso, hace que el método de muestreo uniforme obtenga resultados satisfactorios (el error es algo mayor que los demás) en poco tiempo. Considerando la máxima reducción de la varianza (ruido visible en la imagen) el método adaptativo es el que proporciona el error porcentual promedio más bajo.

6.4.6. Escena 2: Esfera con alto brillo especular

La anterior Escena 1 presenta un alto grado de uniformidad en la distribución de la iluminación incidente, por lo que un método adaptativo no presenta demasiado beneficio. Para apreciar mejor el ajuste realizado por el método de muestreo es necesario encontrar una situación en la que la función presente un alto pico o variabilidad que sea difícil de muestrear.

Podemos hallar esta situación en la escena izquierda de la figura 6.13. Se trata de una esfera iluminada por una fuente de luz. Si la misma es altamente especular, como

6.4. Evaluación del método y resultados

MUESTRAS	Uniform	Lóbulo \mathcal{S}^2	Lóbulo Ω	Adapt.Disc
1^2	0,0793	0,4136	0,3216	0,3396
5^2	0,9022	5,0623	7,6766	7,6304
10^2	3,2513	18,6036	30,5650	29,9153
15^2	7,2801	40,7002	68,6088	66,7410
20^2	12,9957	74,3947	68,6088	118,2702
30^2	28,8140	163,221	273,053	263,706
40^2	50,8346	286,605	486,620	483,715
50^2	94,1930	510,074	761,316	878,420
100^2	333,768	1690,65	2947,66	2882,860

Tabla 6.7: Tiempo de muestreo, en segundos, obtenidos para Escena 1 Cornell Box. A medida que la calidad aumenta (número de muestras utilizadas) el tiempo necesario es mayor.

MUESTRAS	Uniform	Lóbulo \mathcal{S}^2	Lóbulo Ω	Adapt.Disc
1^2	109,97	88,42	94,72	80,36
5^2	48,30	74,92	47,50	18,18
10^2	29,99	31,15	27,75	9,89
15^2	23,82	26,85	27,86	8,58
20^2	20,77	23,01	25,74	8,78
30^2	23,84	19,35	18,52	8,65
40^2	16,80	16,94	17,37	9,55
50^2	8,33	9,03	17,28	9,98
100^2	3,93	3,21	4,75	2,79

Tabla 6.8: Error porcentual promediado obtenido para Escena 1 Cornell Box. Es obtenido al comparar las imágenes generadas para cada BRDF y para cada valor de muestras, con la imagen de referencia. Aumentando la calidad se disminuye el error. En este caso, el error mínimo es obtenido con nuestro método de muestreo.

consecuencia el observador verá un brillo en la zona en la que la dirección desde un punto al observador es —con respecto a la dirección de dicho punto hacia la fuente de luz— próxima a la reflexión especular perfecta.

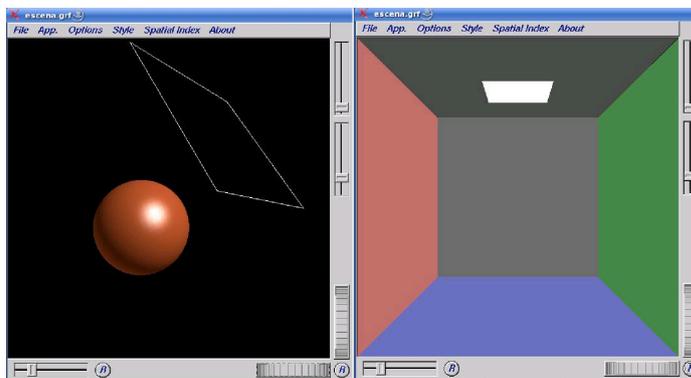


Figura 6.13: Escenas 1 y 2: Las escenas GRF serán sintetizadas mediante un algoritmo básico de *path-tracing*

Mostramos a continuación una serie de tablas con los datos recogidos de la generación de 9 imágenes por cada método de muestreo y por cada BRDF. Las tablas 6.9 y 6.10 recogen los datos particulares de nuestro método de muestreo, para los tiempos de generación y el error medio relativo medido con respecto a la imagen de referencia calculada con los datos de cada BRDF. Otras dos tablas, la 6.11 y 6.12, muestran datos resumidos —promediados para todas las BRDFs— de los cuatro métodos que nos sirven de comparativa.

En esta escena y con los datos obtenidos evaluamos el comportamiento (tiempo vs. error) de distintos métodos de selección de direcciones. No presuponemos ningún tipo de información relativa a la radiancia incidente, y por ello, muestreamos la semiesfera de direcciones siguiendo la función de reflectancia. Todo esto en el contexto del cálculo de la radiancia de salida empleando un algoritmo de iluminación global basado en técnicas de Monte Carlo. Para un sistema de síntesis como un *path-tracer* esto se traduce en que el usuario describe la geometría de la escena (objetos y fuentes de luz) y sus características (color y reflectancia). El usuario no es requerido salvo para indicar determinadas características de la imagen de salida (resolución y calidad). Sin embargo, y a raíz de los experimentos realizados, para un muestreo eficiente de la BRDF y con las actuales técnicas, es necesaria la intervención del usuario. Nos referimos al muestreo basado en el lóbulo coseno (tanto en \mathcal{S}^2 como en Ω) y a otras técnicas más generales como la factorización de la BRDF dada por Lawrence (ver sección 5.5 en la página 116).

Se ha observado que el exponente de una BRDF basada en el lóbulo coseno, aplicado a una PDF de este tipo, no produce el mismo resultado al dibujarlo en coordenadas

6.4. Evaluación del método y resultados

BRDF	1 ²	5 ²	10 ²	15 ²	20 ²	30 ²	40 ²	50 ²	1000 ²
ASHIKHMIN	0,00020	0,00241	0,0091	0,0201	0,0352	0,0787	0,1390	0,2198	80,82
BEARDMAX	0,00029	0,00389	0,0121	0,0227	0,0385	0,0812	0,1402	0,2236	76,75
BLINN	0,00028	0,00272	0,0088	0,0184	0,0316	0,0692	0,1200	0,1862	69,26
COUPLED	0,00019	0,00226	0,0083	0,0187	0,0329	0,0739	0,1308	0,2029	78,83
HE	0,00030	0,00271	0,0090	0,0189	0,0328	0,0709	0,1259	0,1917	70,01
LAFORTUNE	0,00031	0,00296	0,0095	0,0197	0,0339	0,0736	0,1293	0,2015	70,10
LEWIS	0,00031	0,00288	0,0094	0,0187	0,0338	0,0718	0,1255	0,1942	72,70
MINNAERT	0,00024	0,00308	0,0102	0,0209	0,0332	0,0693	0,1183	0,1884	66,01
OREN	0,00016	0,00225	0,0086	0,0189	0,0321	0,0727	0,1292	0,2050	77,36
PHONG	0,00036	0,00279	0,0093	0,0187	0,0328	0,0730	0,1232	0,1939	73,30
POULIN	0,00024	0,00236	0,0082	0,0175	0,0317	0,0702	0,1234	0,1920	69,84
SCHLICK(D)	0,00024	0,00264	0,0087	0,0189	0,0312	0,0715	0,1257	0,1951	69,68
SCHLICK(S)	0,00021	0,00255	0,0088	0,0188	0,0309	0,0697	0,1262	0,1953	69,82
STRAUSS	0,00033	0,00330	0,0110	0,0231	0,0409	0,0878	0,1512	0,2331	83,33
TORRANCE	0,00024	0,00289	0,0091	0,0189	0,0324	0,0722	0,1224	0,1886	68,83
WARD	0,00028	0,00290	0,0093	0,0192	0,0336	0,0718	0,1250	0,1930	69,48
PROMEDIO	0,00026	0,00279	0,0093	0,0195	0,0336	0,0736	0,1285	0,2003	73,32

Tabla 6.9: Tiempos de muestreo para Escena2 empleando la PDF adaptativa. Los valores promediados para esta PDF se comparan satisfactoriamente con los obtenidos para otros métodos (ver tablas 6.11 y 6.12).

BRDF	1 ²	5 ²	10 ²	15 ²	20 ²	30 ²	40 ²	50 ²	1000 ²
ASHIKHMIN	116,8	78,35	78,77	79,50	78,80	78,18	77,95	77,98	77,63
BEARDMAX	287,2	36,27	20,62	14,30	13,67	13,90	14,26	14,62	15,90
BLINN	132,5	23,66	21,02	21,02	20,97	20,30	20,83	20,45	20,44
COUPLED	183,9	50,67	31,14	26,78	25,65	22,51	21,57	21,67	21,10
HE	106,1	12,12	9,97	9,70	9,67	9,80	9,98	9,93	9,93
LAFORTUNE	21,0	14,20	13,71	13,50	13,46	13,53	13,56	13,55	13,51
LEWIS	34,0	25,82	26,58	26,15	26,18	25,88	26,00	25,98	25,98
MINNAERT	262,7	46,11	17,40	12,09	13,53	11,88	12,70	11,88	12,44
OREN	179,7	58,12	29,09	24,72	26,36	21,20	19,64	19,71	18,91
PHONG	34,0	25,82	26,58	26,15	26,18	25,88	26,00	25,98	25,98
POULIN	169,7	50,20	41,83	39,65	41,01	41,10	40,99	41,71	41,54
SCHLICK (D)	186,5	41,63	20,10	16,11	16,63	17,04	15,85	16,83	16,37
SCHLICK (S)	191,3	36,49	19,77	16,16	14,73	15,13	15,35	15,92	14,94
STRAUSS	105,8	10,12	7,86	6,15	5,92	5,99	5,85	5,88	5,72
TORRANCE	161,1	17,85	8,09	7,11	7,34	6,93	6,91	6,73	7,07
WARD	85,4	31,78	32,77	32,33	32,27	32,08	32,09	32,09	32,13
PROMEDIO	141,1	34,95	25,33	23,21	23,27	22,58	22,47	22,56	22,47

Tabla 6.10: Error relativo para Escena2 y nuestro método adaptativo de selección de direcciones. Los valores están expresados en porcentaje, aunque se omite deliberadamente dicho símbolo.

polares [MULG08]. De lo cual se deduce que al usuario le corresponde encontrar el exponente que mejor se ajusta a la gráfica de la BRDF. Otra alternativa consiste en el uso de métodos numéricos de ajuste no lineal, tal como Levenberg-Marquardt [PFTV92]. Sin embargo, esta técnica iterativa suele caer en un mínimo local, por lo que el éxito del proceso recae en los valores iniciales dados por el usuario, no siendo del todo automático. Se suma el hecho de ser métodos costosos en tiempo de cálculo. Nosotros por nuestra parte hemos ajustado el exponente que mejor se aproxima a la BRDF de forma manual intentando aproximar lo más posible ambas gráficas. Para ello hemos utilizado herramientas propias incluidas en *Wannabe Amazing*.

Por otro lado, la factorización de BRDFs analíticas ofrecida por Lawrence *et al.* [LRR04] tiene, como ya hemos visto, siete parámetros que la caracteriza. Es necesario encontrar la mejor factorización para cada instancia de BRDF, ya que con una misma factorización los resultados obtenidos no son del todo buenos, como puede verse en la tabla de error 6.12. Hemos encontrado la mejor factorización de la siguiente forma: comparamos el valor medio de la matriz inicial Y que contiene el producto de la BRDF por el término coseno, con el valor medio de la suma de factores multiplicativos resultantes de una factorización arbitraria. Repetimos el procedimiento varias veces, modificando los parámetros de forma sistemática y obtenemos la factorización con el error mínimo como resultado.

Llamamos *Sphere 1* a la situación en la que el exponente del lóbulo coseno, así como la factorizaciones son constantes. Este caso no requiere la intervención del usuario, pero no define bien algunos de los métodos de muestreo como los dos basados en el lóbulo coseno y la factorización de la BRDF dada por Lawrence. *Sphere 2* denota el cambio en estos tres métodos de selección de direcciones, mediante el cual se localiza la mejor parametrización y en tal caso, están en la mejor situación para ser comparados con otros métodos. Serán los métodos uniforme y adaptativo los que se mantengan de una escena a otra. El lector debe notar sin embargo, que los errores relativos para los anteriores algoritmos son distintos y sí varían entre escenas, ya que ha cambiado la imagen de referencia.

De la comparación de las figuras 6.14 y 6.15 se observa una aproximación de las gráficas para los métodos basados en el lóbulo coseno y en la factorización de la BRDF. Esto significa que mejoran sus resultados en la escena *Sphere 2* bajo una correcta parametrización. Sin embargo, en ambos casos la gráfica que se sitúa por debajo de las otras, es decir, aquella que obtiene menor error empleando el mismo o una cantidad inferior de tiempo, corresponde a nuestro método adaptativo. Datos numéricos más completos (sobre error y tiempo al comparar con el método uniforme y otros con ajuste de parámetros) son dados en las tablas 6.13 y 6.14.

Resaltar que nuestro método, aunque tiene parámetros que lo controlan (número medio de intentos al realizar muestreo por rechazo en el nodo, discretización del precálculo de la BRDF, número de quadtrees precalculados por BRDF, etc.), no

6.4. Evaluación del método y resultados

PROMEDIO	1 ²	5 ²	10 ²	15 ²	20 ²	30 ²	40 ²	50 ²	1000 ²
UNIFORME	0,00006	0,00042	0,0016	0,0034	0,0061	0,0138	0,0243	0,037	15,39
LÓBULO \mathcal{S}^2	0,00018	0,00298	0,0118	0,0266	0,0471	0,1062	0,1884	0,294	117,74
LÓBULO Ω	0,00025	0,00350	0,0137	0,0306	0,0544	0,1223	0,2188	0,341	135,99
ADAPTATIVA	0,00026	0,00279	0,0094	0,0196	0,0336	0,0736	0,1285	0,200	73,32
FACTORED	0,00023	0,00346	0,0135	0,0300	0,0527	0,1183	0,2096	0,326	128,36

Tabla 6.11: Tiempos medios para Escena2. No se conoce en todas las BRDFs el parámetros exponente. En este caso, las PDFs basadas en el lóbulo coseno usan un valor constante. Así mismo, la factorización de Lawrence se toma igual para todas las BRDFs.

PROMEDIO	1 ²	5 ²	10 ²	15 ²	20 ²	30 ²	40 ²	50 ²	1000 ²
UNIFORME	225,0	85,38	59,31	49,27	44,29	31,06	29,77	26,72	22,48
LÓBULO \mathcal{S}^2	75,8	59,86	32,65	29,31	27,65	26,63	26,24	25,24	25,07
LÓBULO Ω	75,5	58,00	32,50	34,24	27,86	26,21	26,18	25,37	25,08
ADAPTATIVA	141,1	34,95	25,33	23,21	23,27	22,58	22,47	22,56	22,47
FACTORED	973,2	88,15	68,45	72,56	68,04	65,10	64,00	63,99	62,75

Tabla 6.12: Error relativo medio para Escena2 con exponente constante, en el caso de las PDF basadas en el lóbulo coseno, y factorización constante. Por espacio, se omite el símbolo de porcentaje.

PROMEDIO	1 ²	5 ²	10 ²	15 ²	20 ²	30 ²	40 ²	50 ²	1000 ²
UNIFORME	0,00006	0,00042	0,0016	0,0034	0,0061	0,0138	0,0243	0,0379	15,39
LÓBULO \mathcal{S}^2	0,00018	0,00291	0,0114	0,0256	0,0456	0,1023	0,1853	0,2833	112,96
LÓBULO Ω	0,00025	0,00368	0,0145	0,0324	0,0577	0,1288	0,2292	0,3573	142,96
ADAPTATIVA	0,00026	0,00279	0,0094	0,0196	0,0336	0,0736	0,1285	0,2003	73,32
FACTORED	0,00028	0,00321	0,0123	0,0273	0,0483	0,1083	0,1921	0,2986	115,39

Tabla 6.13: Tiempos medios para Escena 2 con ajuste de exponentes en el caso de las PDFs basadas en el lóbulo coseno. Así mismo, la factorización de Lawrence, emplea la mejor parametrización para cada BRDF.

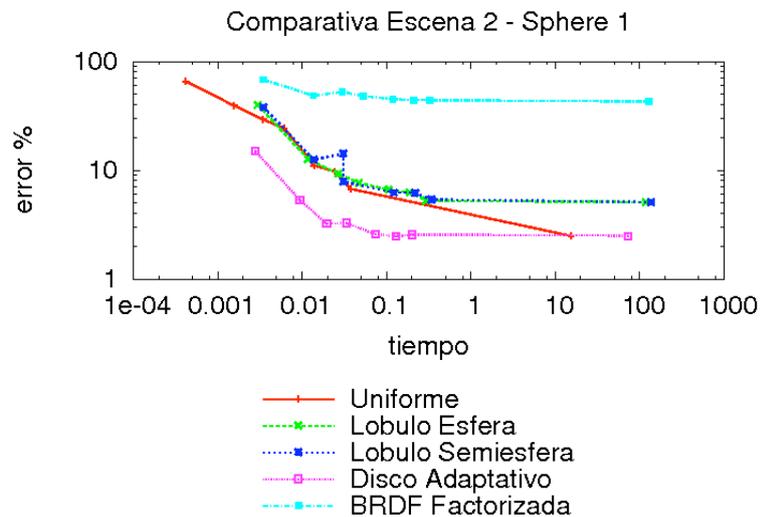


Figura 6.14: Gráfico que compara métodos de selección aleatoria de direcciones. Muestra en el eje de abcisas valores para tiempos y en el eje de ordenadas el error relativo.

requiere la intervención del usuario. Nos referimos a que el método es independiente de la BRDF a la que se aplique y el usuario no es requerido en ningún momento para especificar características del modelo de reflexión empleado. En general, lo único que nuestro método requiere para muestrear de forma exacta una función, es tener la habilidad de evaluarla.

6.4. Evaluación del método y resultados

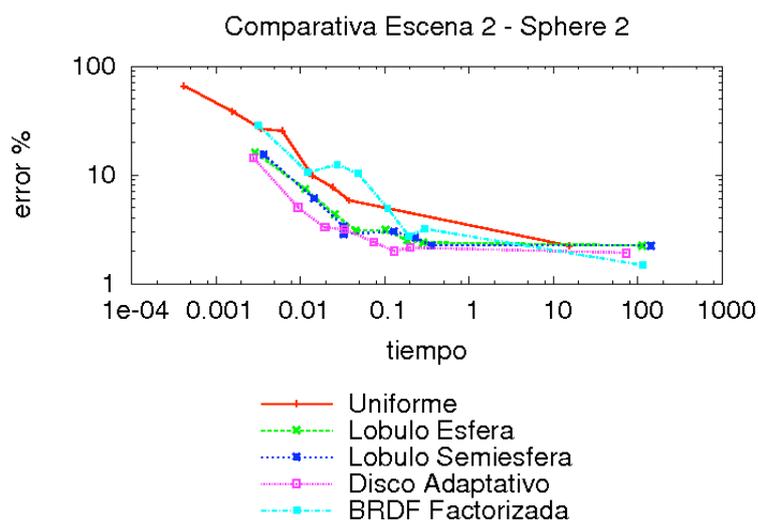


Figura 6.15: Gráfico que compara tiempos y error relativo para distintos métodos. Se considera la mejor parametrización para aquellos métodos que así lo requieran.

PROMEDIO	1 ²	5 ²	10 ²	15 ²	20 ²	30 ²	40 ²	50 ²	1000 ²
UNIFORME	270,7	101,85	74,36	62,54	61,40	45,89	43,68	41,82	38,23
LÓBULO \mathcal{S}^2	123,1	52,17	43,41	40,34	39,07	39,12	38,50	38,39	38,23
LÓBULO Ω	117,2	51,42	42,09	39,37	38,85	39,02	38,66	38,24	38,23
ADAPTATIVA	161,9	50,27	41,04	39,28	39,15	38,41	37,99	38,14	37,91
FACTORED	178,8	64,41	46,51	48,47	46,29	40,92	38,75	39,19	37,47

Tabla 6.14: Error relativo medio (en tanto por ciento) para Escena2 con ajuste de exponentes en el caso de las PDF basadas en el lóbulo coseno. Se omite el símbolo %.

6.4.7. Escena 3: El dragón de Stanford

En la presente escena consideramos el muestreo simultáneo de modelos de reflexión de distinta índole: experimentales, físicos y analíticos ajustables a datos medidos.

Primeramente tomamos la escena del Dragón procedente del Repositorio 3D de la Universidad de Stanford.² Hemos usado las siguientes funciones de reflectancia: para el suelo una instancia de Oren-Nayar [ON94] con un valor de rugosidad alto correspondiente a 0.83; para las paredes se ha empleado Strauss [Str90] con parámetros suaves, casi difusos. La geometría del dragón se caracteriza por la BRDF de Lafortune [LW94] utilizando como exponente $n = 46$ y como factorización $16 \times 32 \times 100 \times 32$ y 1×2 (que equivale a 2 MB de datos en RAM). Igualmente los parámetros para las BRDFs de Oren-Nayar y Strauss fueron ajustados. Con esta mezcla de modelos de reflectancia, comparamos nuestro método con los comentados en la tabla 6.3. Nuestro método se aplica igualmente al muestreo de las tres BRDFs sin necesidad de ajustar ningún parámetro.

Con pocas muestras —tan sólo 100— nuestro algoritmo da un resultado con menos ruido en la imagen, como puede verse en la figura 6.16. Significa que para un mismo número de muestras, nuestro método obtiene menos error ya que se ajusta mejor al integrando (la BRDF).

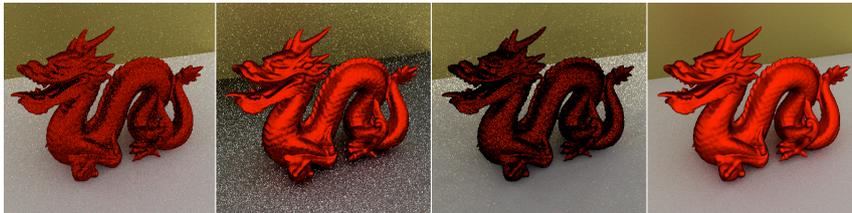


Figura 6.16: Escena 3: De izquierda a derecha las imágenes proceden de los métodos uniforme, lóbulo coseno, la factorización de Lawrence, y finalmente el método propuesto en este trabajo. Para una resolución de 400x400 píxeles, los tiempos obtenidos (siguiendo el mismo orden) son: 9,749, 29,06, 24,539 y 42,632 segundos respectivamente. La imagen con menos ruido corresponde a la dada por nuestro trabajo.

² <<http://graphics.stanford.edu/data/3Dscanrep/>> [consultado el 26-09-2007]

6.4.8. Escena 4: Las tres jarras

A continuación cambiamos a otra escena, consistente en tres jarras, para las cuales se emplean las siguientes BRDFs —en notación GRF— considerando las geometrías de izquierda a derecha:

- El modelo de BeardMaxwell [MBWL73] considerando $\Omega=1$, $\tau=0$, el índice de refracción del oro y $rs = 0,7$ $rd = 0,5$ $rv = 0,2$. Se utiliza un exponente para la PDF lóbulo coseno de 81. La factorización de Lawrence ha sido $N_{\theta_u} = 16 \times N_{\phi_u} = 16$ y $N_{\theta_p} = 32 \times N_{\phi_p} = 16$ para la matriz inicial. El número considerado de factores $J \times K$ ha sido 2×3 , y se ha usado la reparametrización en base al vector \mathbf{h} .
- El modelo de He-Torrance-Sillion-Greenberg [HTSG91] con los siguientes parámetros: $\sigma=0.5$ $\tau=3$ $\lambda=800$. El exponente para la PDF lóbulo coseno ha sido $n = 3$. La factorización empleada usa los siguientes valores: 16×16 32×16 $J = 2$ $K = 1$, no reparametrizando en este caso.
- El clásico modelo de Blinn [Bli77] con exponente $n = 100$ y $ks = 1$ $kd = 0,67$. La PDF lóbulo coseno que mejor lo muestrea utiliza como exponente $n=81$ (nótese que es diferente al parámetro n de Blinn). La factorización se ha realizado con los siguientes parámetros: 16×16 100×32 $J = 2$ $K = 2$ con reparametrización en \mathbf{h} .



Figura 6.17: Escena 4: imágenes con 400 muestras frente a una imagen de referencia de 10000 muestras. Izquierda: el muestreo uniforme empleando 1,9708 segundos tiene un 36,81% de error. Derecha: muestreo basado en el lóbulo coseno en \mathcal{S}^2 . El tiempo de muestreo es 26,3372 segundos y el error relativo 14,87%.

Una vez observado el resultado de las figuras 6.17 y 6.18, podemos concluir que nuestro método obtiene resultados satisfactorios en todas escenas probadas, con situaciones variables en complejidad y calidad. En cuanto al tiempo de muestreo debemos hacer una puntualización. Si bien en la anterior escena y las demás vistas, el método uniforme es siempre el más rápido computacionalmente, debemos tener muy presente que no es el mejor en situaciones complejas. Escenas con un alto grado de especularidad no serían adecuadamente muestreadas.



Figura 6.18: Escena 4: imágenes con 400 muestras frente a una imagen de referencia de 10000 muestras. Izquierda: la factorización de la BRDF empleando 26,3372 segundos tiene un 14,87 % de error. Derecha: muestreo adaptativo en el disco unidad. El tiempo de muestreo es 112,189 segundos y el error relativo de 9,37 % es el más bajo.

Para mostrar con mayor claridad la anterior sentencia consideremos una escena con la BRDF de Lafortune [LW94] instanciada varias veces con exponentes aumentando desde 10 hasta 200. La muestreamos con la PDF uniforme y la PDF adaptativa en el disco, tomando 100 muestras. Nuestro método empleó 43,45 segundos, frente a los 6,11 del método uniforme. Sin embargo, es más importante considerar el error que el tiempo, ya que el método uniforme utilizando 2500 muestras no llega a eliminar el ruido en la imagen, como puede verse en la figura 6.19. Por lo tanto comparando tiempos (43,45 segundos frente a los 150,54 de la segunda imagen con PDF uniforme), así como el error, nuestro método resulta favorable en ambos casos.

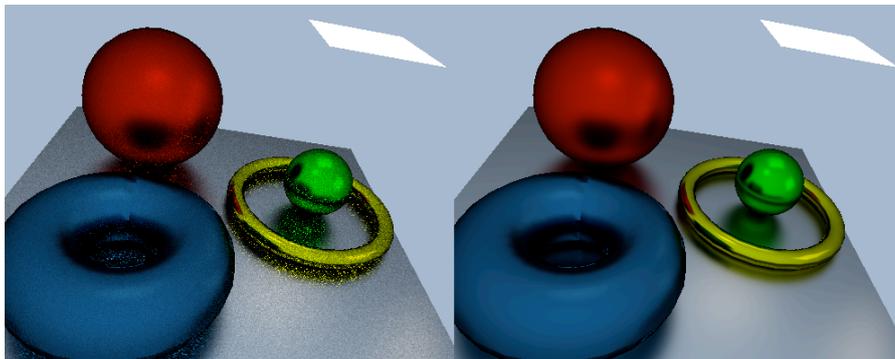


Figura 6.19: La imagen de la izquierda es obtenida usando muestreo uniforme y 2500 muestras. La imagen de la derecha es obtenida con nuestro método usando menos muestras, tan solo 100. A pesar de emplear más muestras, el método uniforme no consigue eliminar el ruido presente en la imagen.

A continuación pasamos a realizar un estudio del comportamiento sobre esta escena, de varios métodos de muestreo y en base a un número mayor de modelos de

6.4. Evaluación del método y resultados

PROMEDIO	5 ²	10 ²	15 ²	20 ²	30 ²	40 ²	50 ²
FACTORIZACION	126,19	128,782	125,661	127,300	129,731	127,282	126,94
ADAPTATIVO	36,64	36,289	36,389	36,387	36,448	36,371	36,54

Tabla 6.15: Tiempos medios de precálculo para Escena 4.

PROMEDIO	5 ²	10 ²	15 ²	20 ²	30 ²	40 ²	50 ²
FACTORIZACION	1,57	6,064	13,108	23,326	52,330	92,628	145,14
ADAPTATIVO	5,00	19,453	42,775	81,290	77,883	311,985	473,75

Tabla 6.16: Tiempos medios de muestreo para Escena 4. Las imágenes empleadas tienen una resolución de 500 × 220 píxeles.

reflectancia (los citados en la tabla 6.18). En este caso la factorización de la BRDF se realiza con una resolución fija $N_u \times N_p$ de $32 \times 16 \times 64 \times 32$ y se mantiene el número de factores de la tabla 6.19. Este cambio implica que ahora requiere más memoria, en concreto 4096 KBytes para cada BRDF. Nuestro método, sin embargo, requiere para el muestreo de dichas funciones 564,2 KBytes en media. Otro dato importante de esta comparativa es el tiempo de precálculo. En media, para la construcción de 60 estructuras *quadtree* se emplearon 125,17 segundos, mientras que la Factorización de Lawrence fue de 36,44 segundos. Datos mas completos del resultado final de la actuación de ambos métodos se encuentran en las tablas 6.16 y 6.17.

Las tablas de datos pueden interpretarse de forma gráfica como en la figura 6.20. De forma similar a otras escenas, la factorización obtiene vectores aleatorios de forma muy rápida, en media sólo 704,42 segundos. Mientras, nuestro método requiere en promedio 2273,36 segundos. Esto no debería cerrar la comparativa, ya que debemos evaluar el error obtenido con la aproximación dada por la factorización, y el hecho de que nuestro método proporciona una solución exacta. Finalmente la balanza se inclina del otro lado: la factorización obtiene un error del 35,59 % y el muestreo adaptativo en el disco lo reduce al 8,62 %. Fácilmente concluimos que puesto que nuestro método disminuye el error, podemos emplear menos muestras para dar un resultado similar

PROMEDIO	5 ²	10 ²	15 ²	20 ²	30 ²	40 ²	50 ²
FACTORIZACION	38,12	39,06	38,63	35,04	35,69	35,08	38,80
ADAPTATIVO	10,65	8,79	8,43	8,33	8,23	8,20	8,19

Tabla 6.17: Error relativo medio en tanto por ciento obtenido en Escena 4. Se promedia para un total de 16 BRDFs y se compara nuestro método adaptativo de selección de direcciones con la factorización de la BRDF.

CAPÍTULO 6. Muestreo eficiente de BRDFs genéricas

a la factorización. Menos muestras implica a su vez, un menor tiempo de muestreo, de forma que nuestra técnica no penalice al algoritmo de síntesis de imágenes que lo aplique.

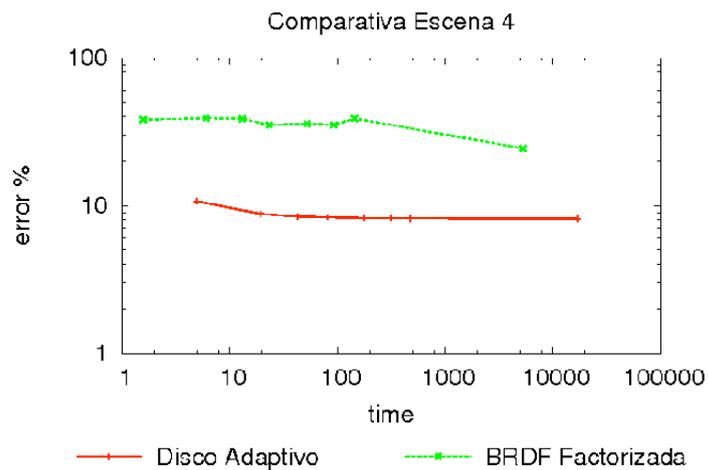


Figura 6.20: Escena 4: Gráfico que muestra en el eje de abscisas valores para tiempos en comparación con valores de error relativo (eje de ordenadas). La factorización usa la misma resolución para todas las BRDFs.

6.5. Muestreo del producto de la BRDF con la radiancia incidente

Para realizar de forma más efectiva la estimación de la *ecuación de radiancia* (ecuación 2.12) mediante métodos de Monte Carlo, es conveniente disponer de una función de densidad que mimetice lo más posible el integrando. Es por ello que en los últimos años han surgido técnicas basadas en el *muestreo por importancias* que se centran en el producto de la función de reflectancia (la BRDF), y la radiancia incidente. Por lo general la radiancia incidente se expresa como luces distantes tomadas de un *Environment Map* (EM), que capta mediante fotografía la iluminación real de un escenario interior o exterior. Utilizamos en este trabajo imágenes equirectangulares³ como *Environment Maps*.

Dentro de estos trabajos están los ya citados, esto es, muestreo por importancias ponderado (WIS), visto en la sección 4.4.7, muestreo por importancias estructurado (SIS), tratado en la sección 4.4.8, muestreo por importancias bidireccional (BIS), en la sección 4.4.9, remuestreo del muestreo por importancias (RIS), en la sección 4.4.10 y, finalmente, el muestreo de funciones mediante *Wavelets* (WaIS), comentado en la sección 4.4.11. Cada uno de estos métodos aplica una técnica particular para el muestreo conjunto de la función producto, sin embargo, en métodos como BIS y RIS se hacen uso de técnicas separadas para el muestreo de la BRDF, por un lado, (*Probability Trees* de McCool [MH97]), y el EM, por otro para posteriormente combinarlos.

Es interesante evaluar la mejora en el comportamiento de estos métodos cuando se utiliza nuestro método adaptativo para generación de direcciones en base a la BRDF (multiplicado por un factor coseno) para la obtención de un estimador de Monte Carlo proporcional al producto $f_r \cdot L_i$. Como punto de partida combinamos nuestra técnica con RIS (*Resampling Importance Sampling*), que comentamos en el siguiente apartado. El uso del estimador RIS combinado con nuestro método de selección se muestra en el algoritmo [19].

6.5.1. Escena 6: Aplicación del disco adaptivo a los métodos de remuestreo para producto de funciones $f_r \cdot L_i$

Consideremos la Escena 4 de las tres jarras iluminadas por un *environment map* de fondo; con una ligera modificación consideramos únicamente una jarra y una resolución de imagen de 300 x 300 píxeles. Para el conjunto de BRDFs de prueba empleamos los métodos de muestreo uniforme, la factorización de la BRDF y nuestro método de muestreo. Los información concreta de cada instancia de BRDF utilizada,

³ Bajo permiso del autor: Sébastien Pérez-Duarte. Disponible *on-line* en la siguiente dirección: <<http://flickr.com/photos/sbprzd/>> [consultado el 7-9-2007]

```

ESTIMARRADIANCIARIS ( $\mathbf{u} \in \Omega, N \in \mathbb{R}^+, M \in \mathbb{R}^+, f_r, L_i, P_{\mathbf{u}}$ )  $\rightarrow \mathbb{R}$ 
Sea  $p_{\mathbf{u}} = dP_{\mathbf{u}}/d\sigma_p$ 
Sea  $q \propto f_r \cdot L_i$ 
Sea  $\mathbb{R} s = 0$ 
for (j = 0 to N) begin
  Sea  $\mathbb{R} W_j = 0$ 
  Generar  $x \sim p_{\mathbf{u}}$  con ObtenerNMuestras( $\mathbf{u}, \mathbf{n}, f_r, M$ )
  for (i = 0 to M) begin
    Calcular los pesos  $w_i = \frac{q(x_i)}{p(x_i)}$ 
    Generar  $y_j \sim w_i$  por reemplazo
     $W_j += w_i$ 
  end
   $W_j /= M \cdot q(y_j)$ 
   $s += (f_r(\mathbf{u}, y_j) \cdot L_i(y_j) \cdot W_j)$ 
end
return  $s/N$ 

```

Algorithm 19: pseudocódigo ESTIMARRADIANCIARIS

así como los parámetros de la mejor factorización de Lawrence, pueden encontrarse detalladamente en las tablas 6.18 y 6.19 de forma respectiva.

Ejecutamos el experimento para 16 BRDFs \times 7 valores distintos de muestras \times 6 métodos de selección de direcciones que nos dan un total de 672 imágenes. Con estos ficheros obtenemos los datos estadísticos que nos permiten relacionar los métodos uniforme, factorización de la BRDF y nuestro propio algoritmo en relación a considerar, en el estimador, el uso o no del remuestreo para la radiancia incidente.

Numéricamente, los datos relativos a tiempos medios de muestreo así como el error relativo con respecto a la imagen de referencia, pueden ser consultados en las tablas 6.21 y 6.22. De forma alternativa a la información numérica, se muestra en el gráfico 6.21 la comparativa de los distintos métodos de muestreo evaluados, incluyendo el propuesto en esta tesis.

Concluimos esta sección con los comentarios referentes a la figura 6.21 y las repercusiones de este trabajo. Se puede observar que la gráfica que se dispone por debajo de las otras dos es la relativa a nuestro método de selección estocástica de puntos en el dominio del disco unidad. Numéricamente se constata a su vez cómo la combinación de este algoritmo con los métodos de remuestreo por importancias, RIS, mejoran considerablemente el error cometido en la estimación. Disminuimos de un error ya bajo de por sí, del 4,74% a un 2,32% por haber considerado la información que es posible obtener del muestreo del *environment map*. Se observa, que en la comparación directa con el método general de muestreo de BRDFs

6.5. Muestreo del producto de la BRDF con la radiancia incidente

BRDFs	Parámetros
ASHIKHMIN	$nu = 25$ $nv = 25$ $ks = 0,39$ $kd = 0,6$
BEARD-MAXWELL	$\Omega = 1$ $\tau = 0$ $n_{real} = 0,37$ $n_{imag} = 2,82$ $rs = 0,4$ $rd = 0,75$ $rv = 0,5$
BLINN	$n = 100$ $ks = 1$ $kd = 0,67$
COUPLED	$R_m = 1$ $R_o = 0,05$
HE	$\sigma = 0,5$ $\tau = 3$ $\lambda = 800$
LAFORTUNE	$cx = -1$ $cy = -1$ $cz = 1$ $n = 100$ $kp = 0,96$
LEWIS	$n = 100$ $ks = 1$ $kd = 0$
MINNAERT	$k = 0,8$ $kd = 1$
OREN	$s = 0$ $r = 1$
PHONG	$n = 100$ $ks = 0,4$ $kd = 0,6$
POULIN	$d = 2$ $h = 0,01$ $n = 100$ $ks = 0,8$ $kd = 0,2$
SCHLICK (D)	$sc = 0,7$ $sr = 0,31$ $sp = 1$ $ly = \text{true}$ $dc = 1$ $dr = 0$ $dp = 1$
SCHLICK (S)	$sc = 0,7$ $sr = 0,31$ $sp = 1$ $ly = \text{false}$
STRAUSS	$s = 0,76$ $m = 0,91$ $ks = 0,64$ $kd = 0,5$
TORRANCE	$m = 0,35$ $n_{real} = 0,617$ $n_{imag} = 2,63$ $ks = 0,7$ $kd = 0,3$
WARD	$sx = 0,2$ $sy = 0,2$ $ks = 0,5$ $kd = 0,5$

Tabla 6.18: Listado de los modelos de reflexión considerados así como los parámetros que se han utilizado en las escenas de ejemplo.

BRDF	$N_{\theta_u} \times N_{\phi_u}$	$N_{\theta_p} \times N_{\phi_p}$	$J \times K$	Reparam.
ASHIKHMIN	16×16	32×16	1×2	false
BEARD-MAXWELL	16×16	100×32	2×3	false
BLINN	16×16	100×32	3×3	true
COUPLED	16×16	32×16	1×3	false
HE	16×16	32×16	2×1	false
LAFORTUNE	16×16	100×32	2×2	true
LEWIS	16×16	100×32	2×2	true
MINNAERT	16×16	32×16	1×2	false
OREN	16×16	32×16	1×3	false
PHONG	16×16	100×32	2×2	true
POULIN	16×16	32×16	2×2	false
SCHLICK (D)	16×16	32×16	1×3	true
SCHLICK (S)	16×16	32×16	2×3	false
STRAUSS	16×16	32×16	3×3	false
TORRANCE	16×16	32×16	1×1	false
WARD	16×16	32×16	2×2	true

Tabla 6.19: La factorización de Lawrence de la BRDF requiere encontrar manualmente siete parámetros de forma que la diferencia entre el valor real y su aproximación sea mínima. El siguiente listado muestra la mejor parametrización para cada BRDF.

CAPÍTULO 6. Muestreo eficiente de BRDFs genéricas

PROMEDIO	5 ²	10 ²	15 ²	20 ²	30 ²	40 ²	50 ²
FACTORIZACION	4,019	3,753	4,533	3,693	3,019	6,406	6,653
FACTORIZACION RIS	2,837	2,373	2,691	2,583	2,277	2,333	2,270
ADAPTATIVO	44,500	44,499	44,695	44,429	44,552	44,268	44,821
ADAPTATIVO RIS	46,298	46,370	46,369	46,425	46,294	46,395	46,346

Tabla 6.20: Escena 6: Coste en segundos del precálculo para la factorización de Lafortune y nuestro muestreo adaptativo en condiciones de IS y RIS.

PROMEDIO	5 ²	10 ²	15 ²	20 ²	30 ²	40 ²	50 ²
UNI. RIS	8,187	29,822	65,593	121,697	273,865	486,684	733,209
FACTORIZACION	0,873	3,029	6,568	11,432	25,195	44,742	69,992
FACT. RIS	1,615	6,620	14,597	27,800	60,622	105,511	408,870
ADAPTATIVO	1,615	6,620	14,597	27,800	60,622	105,511	408,870
ADAP. RIS	4,363	17,126	38,099	67,194	152,204	271,681	424,314

Tabla 6.21: Tiempos medios para Escena 6 con y sin utilización de la técnica de remuestreo para la distribución de valores proporcionales al producto de la BRDF y la radiancia incidente.

PROMEDIO	5 ²	10 ²	15 ²	20 ²	30 ²	40 ²	50 ²
UNI. RIS	19,08	13,63	11,73	10,75	8,43	9,22	8,80
FACTORIZACION	19,22	17,72	16,75	16,47	16,77	15,54	15,98
FACT. RIS	22,14	12,26	9,62	12,69	9,37	8,29	8,45
ADAPTATIVO	6,70	5,18	4,86	4,75	4,67	4,65	4,72
ADAP. RIS	11,56	7,36	5,78	5,04	4,54	4,09	3,90

Tabla 6.22: Error relativo medio (en tanto por ciento) para Escena 6 con y sin la utilización de del remuestreo de valores para la generación de aleatorios proporcionales al producto $f_r \cdot L_i$.

6.5. Muestreo del producto de la BRDF con la radiancia incidente

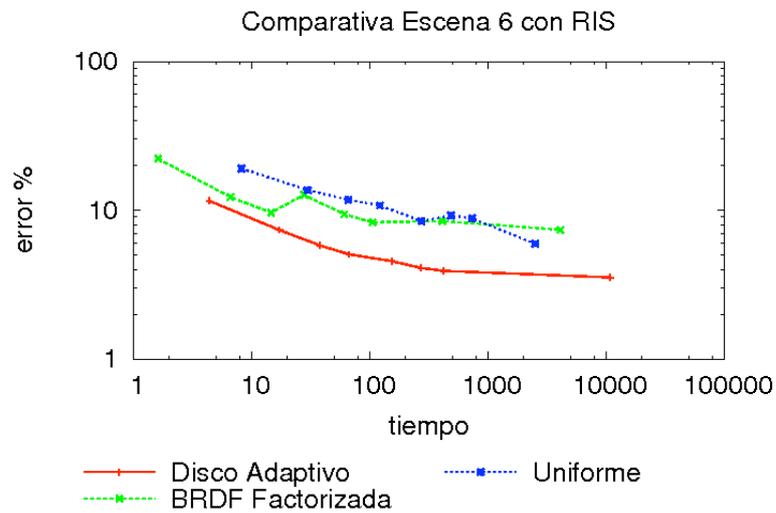


Figura 6.21: Escena 6: Gráfico que muestra en el eje de abscisas valores para tiempos y en comparación con valores de error relativo (eje de ordenadas).

analíticas y adquiridas [LRR04], nuestro método obtiene valores muy inferiores de error. También se ha mostrado que los tiempos de muestreo a pesar de ser más altos que los del método de generación uniforme, justifican la calidad de imagen obtenida.

6.6. Muestreo de BRDFs procedentes de datos adquiridos

En la actualidad no existe ningún modelo analítico que caracterice la reflectancia de una superficie para todos los posibles tipos de materiales de la realidad. Simplemente se agrupan y se intenta sintetizar alguna de sus características: materiales brillantes, rugosos, plásticos, etc. para los cuales sí disponemos de un modelo de BRDF que lo caracteriza.

Si deseamos obtener un alto grado de realismo, necesitamos observar y capturar el aspecto de lo que vemos de alguna manera. Recientemente se está prestando mucha atención a la adquisición y representación del aspecto de objetos reales, además de la geometría, en campos como la informática gráfica y la visión por ordenador. Los datos relativos a la BRDF pueden ser adquiridos principalmente de dos formas:

- Utilizando un *goniorelectómetro*, consistente en una fuente de luz y un sensor fotométrico, podemos obtener con alta precisión el valor de la BRDF en par de direcciones. El sensor y la luz pueden ser posicionados en distintos ángulos, de forma que se cubre totalmente el hemisferio superior de direcciones, para una muestra plana de un material homogéneo. Repitiendo el proceso múltiples veces obtenemos un conjunto denso de valores que son almacenados de manera tabular. Se puede obtener más información si se consideran medidas para distintos valores de longitudes de onda del espectro visible, obteniendo, por tanto, la correcta reproducción del color. El inconveniente principal de este método de adquisición es el tiempo. Se ha de reposicionar cada par de dirección incidente y saliente cada vez, por lo que sólo se obtiene una muestra del material cada vez. Si quisiéramos capturar materiales heterogéneos necesitaríamos, además, tomar muestras en cada punto de la superficie del material. Los resultados adquiridos de esta forma se encuentran por lo general disponibles como bases de datos públicas [DGNK99, MPBM03].
- Las técnicas de adquisición basadas en imagen [War92, MWLT00, Len03] tienen como objetivo agilizar el proceso de adquisición. Se basan en obtener varias muestras de la BRDF en una sola toma, utilizando una cámara digital en lugar de un único sensor fijo. Cada pixel de la imagen resultante contiene una muestra distinta de la BRDF. De esta forma también se puede adquirir a partir de geometría curva.

Dentro del primer grupo, Matusik [MPBM03] proporciona una base de datos pública con datos de reflectancia de 100 materiales reales distintos. Éstos fueron tomados haciendo uso de un dispositivo robotizado para poder aumentar la densidad de las tomas. El resultado es una BRDF tabular almacenada en un fichero binario.

6.6. Muestreo de BRDFs procedentes de datos adquiridos

MERL BRDF	Tiempo de precálculo			Memoria (KB)		
	UNIFCDF	CompCDF	A.Disc	UnifCDF	CompCDF	A.Disc
ALUM-BRONZE.BINARY	28,80	2507,26	70,30	33792	1984,56	523,28
ALUMINA-OXIDE.BINARY	29,55	2571,24	66,88	33792	223,00	241,01
BEIGE-FABRIC.BINARY	27,57	2547,19	65,70	33792	374,16	272,62
BLUE-METALLIC-PAINT2.BINARY	30,24	2482,86	69,97	33792	2063,55	721,91
BLUE-METALLIC-PAINT.BINARY	30,00	119,78	71,22	33792	1685,64	715,91
NICKEL.BINARY	30,13	2576,81	71,36	33792	2057,51	846,15
RED-PLASTIC.BINARY	30,22	2550,61	69,63	33792	557,46	626,82
TEFLON.BINARY	29,77	113,56	64,64	33792	299,56	140,48
VIOLET-ACRYLIC.BINARY	29,69	2471,41	67,29	33792	1647,40	409,40
WHITE-MARBLE.BINARY	30,23	2531,04	67,09	33792	488,80	254,64
YELLOW-PAINT.BINARY	29,30	2524,48	64,54	33792	285,18	146,47
PROMEDIO	29,59	2090,57	68,06	33792	1060,62	445,33

Tabla 6.23: Datos del precálculo de distintos métodos de muestreo aplicados a BRDFs adquiridas.

Nosotros hemos usado un subconjunto de estos ficheros binarios para probar nuestro método de muestreo adaptativo para BRDF analíticas, dado el hecho de que dados dos vectores, uno incidente y otro reflejado, podemos evaluar la BRDF adquirida. Recordemos que éste es el único requisito exigido a nuestro método de muestreo adaptativo en el disco unidad. Los resultados gráficos se muestran en la figura 6.6 y los resultados numéricos en las tablas 6.23 y 6.24, que corresponden a imágenes de resolución 450 x 200 píxeles y son evaluadas con 5^2 muestras. Se han tenido en cuenta las siguientes consideraciones:

Muestro con la PDF Uniforme. La hemos utilizado para poder demostrar que el muestreo aleatorio uniforme no es la mejor opción para el muestreo de la BRDF. En la tabla 6.24 se observa como ofrece los tiempos más rápidos, pero el error más alto (primera columna de la figura 6.6).

Muestreo con la CDF sin comprimir (*Uniform Cascade CDF*). Éste es el método clásico para el muestreo de datos tabulados (recuérdese la sección 4.3.1). Se observa que requiere tanto o más espacio que la propia BRDF en sí (véase tabla 6.24) por lo que es un método prohibitivo para renderizar una escena con varias BRDFs. Por lo demás, el tiempo de precálculo y de muestreo son bajos, pero se observa ruido en la imagen resultante (segunda columna de la figura 6.6). Se ha utilizado una resolución de $32 \times 16 \times 256 \times 32$.

Muestreo compacto de la CDF (*Compress Cascade CDF*). Esta técnica de muestreo de Lawrence et al. [LRR05] consigue reducir considerablemente el tamaño de las tablas CDF. Ello se realiza mediante una aproximación de la curva CDF con el método de Douglas-Peucker [Ros97]. Esta técnica se comentó en detalle en la sección 5.6. El resultado es muy similar a la versión no compacta, tal y como puede verse en la tercera columna de la figura 6.6. Los datos del proceso de compresión revelan que en media se requieren 2090,57 segundos para reducir los 33MB a algo más

CAPÍTULO 6. Muestreo eficiente de BRDFs genéricas

MERL BRDF	Uniforme	UnifCDF	CompCDF	A.Disc
ALUM-BRONZE.BINARY	0,35	2,20	1,15	16,03
ALUMINA-OXIDE.BINARY	0,35	2,31	0,91	16,29
BEIGE-FABRIC.BINARY	0,34	4,01	0,94	4,04
BLUE-METALLIC-PAINT2.BINARY	0,37	1,97	1,11	50,55
BLUE-METALLIC-PAINT.BINARY	0,35	2,20	1,11	4,60
NICKEL.BINARY	0,37	1,87	1,13	12,31
RED-PLASTIC.BINARY	0,37	2,59	0,98	4,31
TEFLON.BINARY	0,34	2,47	0,93	4,25
VIOLET-ACRYLIC.BINARY	0,34	2,19	1,13	38,86
WHITE-MARBLE.BINARY	0,34	2,31	0,97	12,54
YELLOW-PAINT.BINARY	0,34	2,37	0,91	3,86
PROMEDIO	0,35	2,41	1,02	15,24

Tabla 6.24: Datos referentes al tiempo de muestreo de distintos métodos aplicados a BRDFs adquiridas.

de 1MB. Esta reducción del espacio de almacenamiento se traduce en unos tiempos mucho más bajos de muestreo, ya que es más rápido realizar la búsqueda binaria en tablas más pequeñas. Se ha utilizado una resolución inicial de $32 \times 16 \times 1024 \times 512$.

Muestreo adaptativo de la BRDF basado en el disco unidad. Aquí comparamos nuestra técnica exacta de muestreo con las anteriormente mencionadas. El precálculo de 60 estructuras *quadtree* requiere algo más de un minuto y tan solo 445,33 KB (en media) de espacio en memoria. Vemos que este precálculo no penaliza tanto los tiempos finales —como el caso del método CascadeCDF compacto— ni los requerimientos finales de memoria del sistema de síntesis de imágenes. Si bien los tiempos de muestreo son más altos, debemos ver que el error resultante es muy bajo para este número de muestras, tal y como se aprecia en la última columna de la figura 6.6. Nuestras 5^2 muestras contribuyen más al estimador que las dadas por otras técnicas de muestreo y todo ello con un consumo de memoria mínimo.

Con esto podemos concluir la actual sección, y el presente capítulo de esta memoria, indicando que el nuestro método de muestreo general para BRDFs incluye no sólo aquellas analíticas, sino que, además, también es válido para BRDFs adquiridas y almacenadas de forma tabular.

6.6. Muestreo de BRDFs procedentes de datos adquiridos

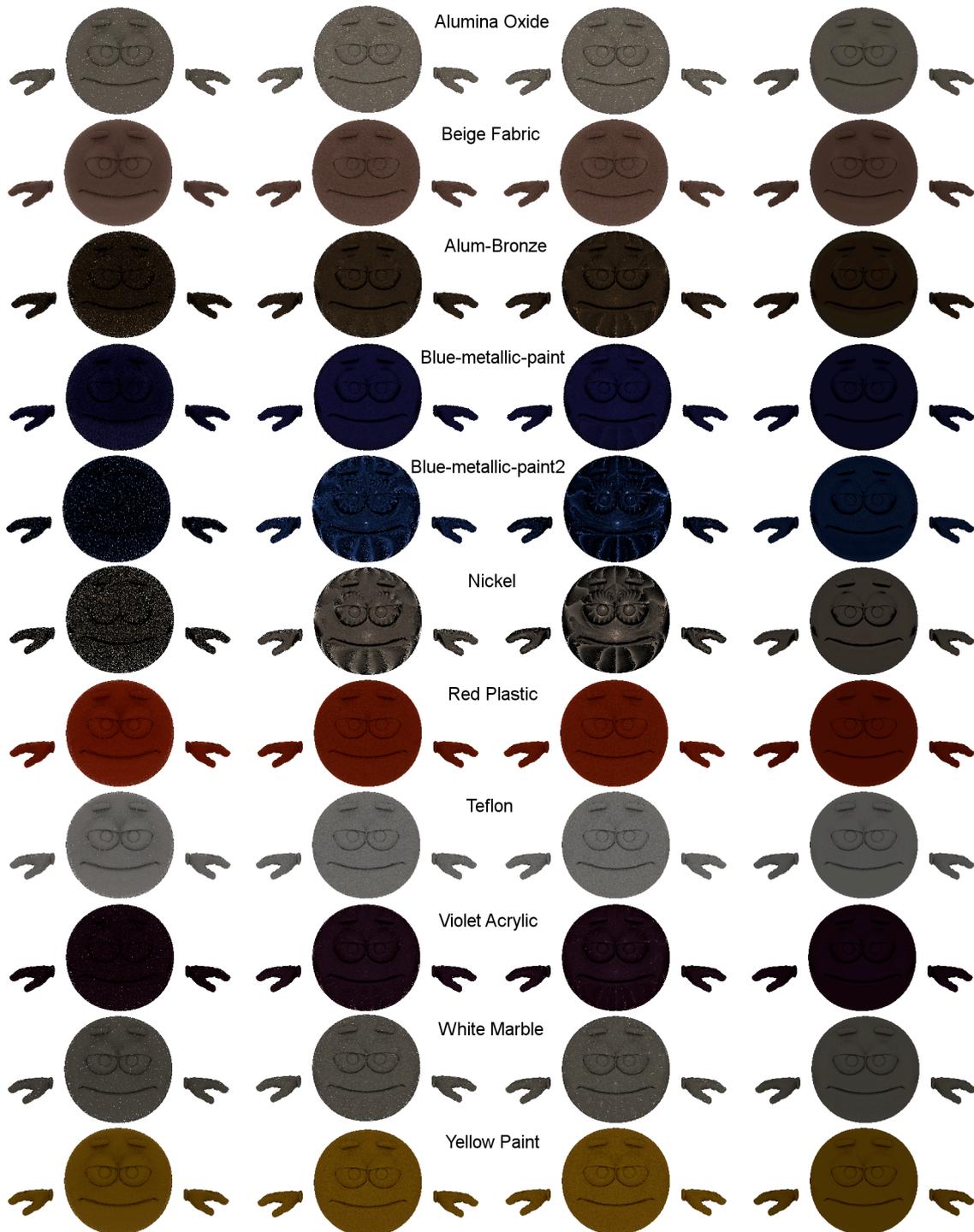


Figura 6.22: Para cada imagen de 450×220 se han tomado 25 muestras. Se han usado cuatro métodos de muestreo para cada BRDF. De izquierda a derecha: PDF uniforme, CascadeCDF, Cascade CDF comprimida y PDF Adaptativa en disco.

CAPÍTULO 7

Utilidades software desarrolladas

Hoy en día el *software de síntesis de imágenes realistas* ha entrado de lleno en el ámbito comercial, y no es raro encontrar anuncios, aplicaciones inmobiliarias y películas que aplican técnicas para generar imágenes de alto realismo. Sin embargo todo ello supone un alto coste computacional, pero son muchas las aplicaciones y muchos los progresos que se están realizando en este campo. El *grupo de investigación de informática gráfica*, del Departamento de Lenguajes y Sistemas Informáticos de la Universidad de Granada, lleva una línea de investigación en *visualización realista*, que comenzó con la implementación de un sistema clásico de trazado de rayos, denominado **GIRT** (*Global Illumination Ray Tracer*) [UPTdS92, UTR⁺97]. Posteriormente, el sistema contó con varias ampliaciones y mejoras dentro del ámbito de *photon-mapping* y estimación de densidades [Las00, LURM02a, LURM02b, GUL⁺04, Las04, GUL⁺05, GUR⁺06]. Sin embargo, en el contexto de un proyecto más amplio de creación de sistemas usables de síntesis de imágenes realistas se vio la necesidad de facilitar el diseño de modelos geométricos, así como la aceptación de otros generados por programas comerciales de amplio uso entre los profesionales del sector. Dada la no disponibilidad de aplicaciones de este tipo, se consideró la necesidad de su diseño e implementación. Varias de las utilidades disponibles serán comentadas en sucesivos apartados.

7.1. *Wannabe Amazing*

Wannabe Amazing es un programa que permite la conversión de datos de formatos 3D comerciales —aquellos considerados estándares del modelado— y el diseño de modelos geométricos, basados en una representación de mallas. Sin embargo, esto puede parecer común en otras herramientas *software*. Lo que lo hace valioso como herramienta de *iluminación global* es que ofrece la posibilidad de enriquecer la información contenida en el modelo, añadiendo todos los datos necesarios en el proceso

CAPÍTULO 7. Utilidades software desarrolladas

de simulación de la iluminación. Los formatos soportados son: (1) Kinetix 3D Studio, (2) Autocad Drawing eXanche Format, (3) Wavefront Technologies, (4) MultiGen OpenFlight y (5) Virtual Reality Modelling Language.

Tras un estudio de las capacidades de cada formato ¹ no se puede decir que sean lo mismo, pero sí que todas se fundamentan en la representación de la geometría mediante mallas. Partiendo de dicha información, se extiende la lista de vértices y la lista de caras con los siguientes datos adicionales:

- normales, si no aparecen en el fichero importado, se calculan y se añaden,
- las coordenadas de textura, que si no están presentes en los ficheros originales serán calculadas,
- el color y la irradiancia en cada vértice,
- la reflectancia, esto es la BRDF que se aplica a cada malla,
- la emisividad de la superficie al definir la malla del tipo *Mesh Light*,
- y otras propiedades del material, como la imagen de textura.
- De forma global a la escena aparecería la definición de luces externas y las definiciones de reflectancia, colores y texturas.

El formato definido, **GRF** *Granada File*, permite, a su vez, la definición y uso posterior de cada una de los atributos mencionados —color, reflectancia, etc.—, mediante el nombrado de éstos, evitando de esta forma tener que repetir datos ya dados. Finalmente, se puede añadir también que es posible realizar anexos de otros ficheros de escenas con formatos GRF. El resultado, como es lógico, es un incremento en el tamaño del fichero de texto GRF de salida. En la tabla siguiente se recoge dicha proporción, que sólo en algunos casos como DXF se convierte en disminución del tamaño. En realidad el incremento en tamaño es causado por el hecho de tratarse de un formato tipo texto, que utiliza nombres descriptivos y separadores, para hacerlo legible por el usuario y totalmente editable. Por esta razón, la diferencia de tamaño se dispara entre formatos binarios (3ds, por ejemplo) y el nuestro, y es más cercano a otros formatos de texto como VRML (archivos wrl).

¹ The programmer's File Format Collections. <<http://www.wotsit.org>>. [Consultado en enero de 2002]

Escena	Tamaño	Obj.	Mat.	Vert.	Caras	GRF
anhk.3ds	7.8 KB	1	3	50	96	85 KB
alambrada.3ds	72 KB	1	3	2402	1266	245 KB
cage.3ds	23.2 KB	6	1	593	954	71 KB
cart_wheel.3ds	32.6 KB	14	3	720	1400	91 KB
greek_pelekys.3ds	403 KB	4	4	9736	18714	1.276 KB
syringe.3ds	169 KB	12	8	4993	5636	521 KB
tetera.3ds	68 KB	1	1	530	1024	68 KB
doric_column.obj	137 KB	1	1	2673	5326	315 KB
exclamation.obj	14 KB	1	1	280	306	31 KB
galleon.obj	289 KB	133	1	3459	2384	2.758 KB
knoxs.obj	22 KB	1	1	552	420	165 KB
tri.dxf	3 KB	1	1	12	16	2 KB
int.ft	137 KB	116	26	1216	1044	203 KB
spinnercar.ft	399 KB	358	64	4506	4070	686 KB
veritech.ft	237 KB	86	7	1422	1669	374 KB
cage.wrl	46 KB	6	1	593	954	80 KB
alien.wrl	218 KB	54	10	1632	2796	241 KB
mono.wrl	152 KB	48	9	1904	3404	296 KB
pompano.wrl	158 KB	19	7	1207	2316	191 KB
exclamation.wrl	20 KB	2	2	280	423	43 KB
coche.wrl	10 KB	61	10	816	1002	73 KB

Tabla 7.1: Comparación de los tamaños para originales y el formato de salida GRF.

Aspectos generales de Wannabe Amazing

La herramienta desarrollada para la generación rápida de escenas adecuadas para el cálculo de iluminación está escrita en **Java**² y su biblioteca gráfica **Java3D**.³

La API de Java 3D es un conjunto de clases para crear aplicaciones y *applets* con elementos 3D. Ofrece a los desarrolladores la posibilidad de manipular escenas complejas en tres dimensiones. La principal ventaja que presenta este API 3D frente a otros entornos de programación 3D es que permite crear aplicaciones gráficas 3D independientes del tipo de sistema y presentables en la Web. Las aplicaciones están constituidas por elementos gráficos individuales como objetos separados que se ven conectados dentro de una estructura en árbol llamada *grafo de escena*, que contiene una descripción completa de la escena, o, como suele denominarse, *universo virtual*.

Las características básicas conseguidas por esta herramienta se pueden enumerar como sigue:

1. **Portabilidad**: La principal ventaja que presenta el uso de Java 3D y Java frente a otros entornos de programación 3D es que la aplicación generada es independiente del tipo de sistema, pudiéndose ejecutar en cualquier plataforma. Otro característica esencial es que puede ser ejecutado en la Web como applet.

² Web oficial de Sun Microsystems. <<http://java.sun.com/products/>>. [Consultado en diciembre de 2005]

³ Java3D Community Site. <<http://www.j3d.org>>. [Consultado en enero de 2006]

2. Trabajo con una librería de alto nivel: El mundo está lleno de objetos con un estado y un comportamiento. Es más lógico trabajar abstrayendo los detalles propios de cada sistema. OpenGL es un estándar de hecho en la comunidad gráfica, pero no podemos olvidar los inconvenientes de su modelo de programación y las alternativas existentes actualmente.
3. Interfaz gráfica amigable: Las funciones de la aplicación se encuentran tanto en los menús como en una barra de botones. El usuario interactúa con el modelo empleando tanto el teclado como el ratón. Se ha incorporado una ayuda en línea en formato hipertexto para facilitar la navegación a través de los contenidos.
4. Cargador de contenidos extensible: Actualmente está implementada la conversión de cinco formatos comerciales, pero, gracias a que *Wannabe Amazing* es una aplicación diseñada y desarrollada siguiendo la metodología orientada a objetos, está preparada para ampliar la gama de formatos sin necesidad de rediseñar las clases.
5. Específica: No es fácil encontrar en el mercado aplicaciones que se ajusten a la medida de lo que se necesita. Sin embargo, la misma se desarrolla para cubrir unas necesidades concretas que ninguna otra herramienta posee en su totalidad.

Esta herramienta se encuentra disponible a través de la web y con funcionamiento probado en las plataformas Windows, Linux y Mac OS X Tiger. De hecho las clases base de manejo de universos virtuales de Java3D desarrolladas para *Wannabe Amazing* se han empleado con éxito en otros trabajos de relevancia [MM04, AMS06].

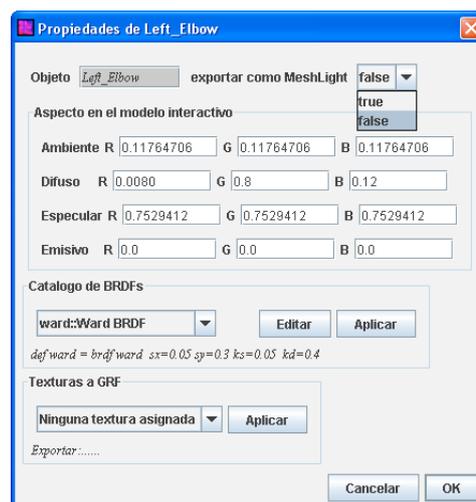


Figura 7.1: La ventana de propiedades de *Wannabe Amazing* permite asignar distintos modelos de BRDFs a mallas del modelo.

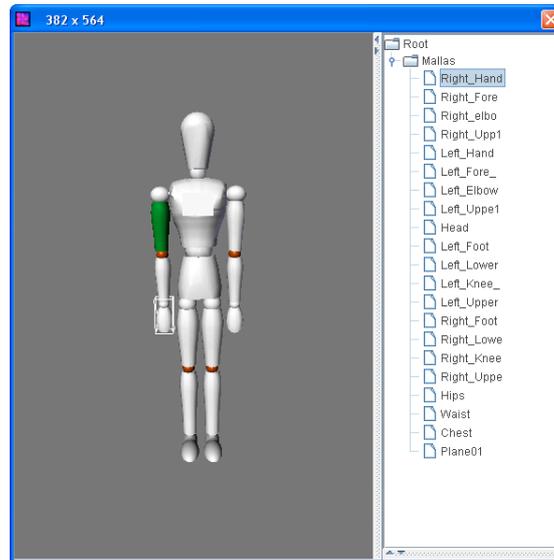


Figura 7.2: *Wannabe Amazing* como herramienta que permite editar de geometría para añadirle información para iluminación global.

Las siguientes imágenes 7.2 y 7.3, muestran cómo la herramienta puede aplicar distintas propiedades a distintas partes o submallas del modelo. Las propiedades susceptibles de ser modificadas son: el nombre de la malla, una marca indicando si la malla es emisiva y el color de visualización en Java3D (emisivo, difuso, especular y ambiente). Lo más interesante se encuentra en la asignación de instancias de BRDFs y de texturas, que previamente han sido definidas y se encuentran en el *catálogo* para su uso. El diálogo de propiedades se muestra en la figura 7.1.

En resumen, *Wannabe Amazing* consigue unificar todas las particularidades de cada formato en uno solo. La visualización tridimensional es llevada a cabo empleando la biblioteca Java3D, y, gracias a la incorporación de objetos tipo *scene*, *appearance*, *material*, es posible aplicar propiedades a un grupo de elementos independientemente del formato de fichero de procedencia.

Para consultar más información, disponer de la última versión o descargar ficheros de escena, se presenta a su disposición la siguiente dirección web: <http://lsi.ugr.es/~rosana>. La aplicación se distribuye como archivo JAR ejecutable directamente en la siguiente URL: <http://lsi.ugr.es/~rosana/software/wannabe/amazing.jar>

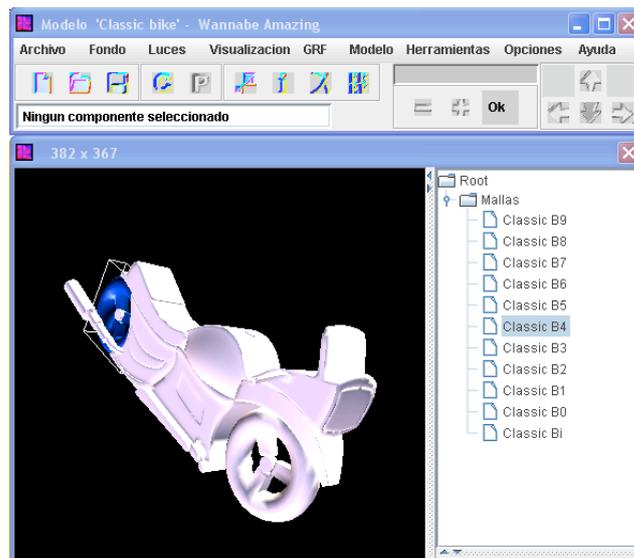


Figura 7.3: Las escenas importadas en *Wannabe Amazing* son exportadas al formato GRF.

7.2. Visor de GRF

El formato de escena GRF presentado en el apéndice A cuenta con herramientas para la facilidad de su uso. Esta claro que al ser un formato de tipo texto, la modificación puede realizarse con cualquier editor. En el caso de escenas sencillas, podemos incluso escribirlas a mano, y para el caso de geometrías más extensas la escena se genera con *Wannabe Amazing*.

La captura 7.4 (izquierda) muestra un ejemplo del visor GRF desarrollado con Java3D y que es parte de *Wannabe Amazing*, aunque se ejecute como aplicación independiente ⁴.

De forma mucho más completa se dispone de un visor GRF e IGRF —IGRF es la extensión aplicada al fichero de escena GRF cuando cuenta con información de irradiancias— desarrollado bajo Linux, que incluye modos de visualización de irradiancias así como de las estructuras de subdivisión espaciales que pueden ser utilizadas en la escena. Una captura de la interfaz basada en la librería FLTK ⁵ se muestra en la imagen derecha de la figura 7.4.

⁴ *Wannabe GRF viewer* URL: <<http://lsi.ugr.es/~rosana/software/wannabe/grf.jar>>

⁵ Web oficial *Fast Light Toolkit*: <<http://www.fltk.org/>> [consultado en octubre de 2007]

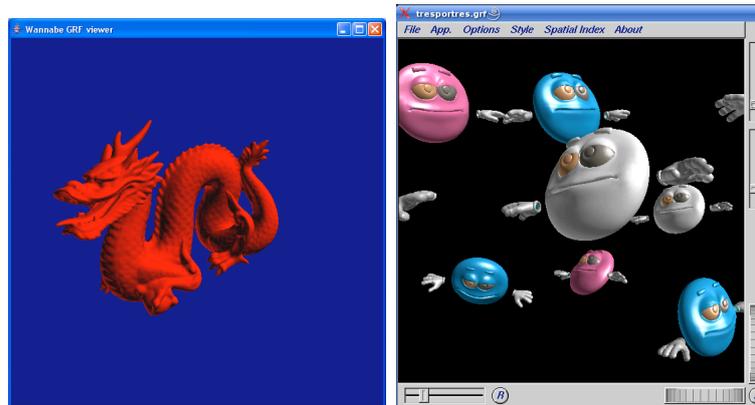


Figura 7.4: Izquierda: sencillo visor de fichero GRFs multiplataforma que emplea la librería gráfica Java3D. Derecha: contamos igualmente con un visor de GRF en C++ para Linux.

7.3. Editor de BRDFs

Una herramienta de gran utilidad de *Wannabe Amazing* es la que permite ver representada, de diversas formas, la función BRDF; al mismo tiempo se pueden modificar los parámetros de ésta. También disponible como aplicación independiente.⁶

Con un total de 16 modelos distintos —lo que denominamos **catálogo de BRDFs**— la aplicación permite seleccionar uno de ellos, que aparecerá inicializado con los siguientes valores por defecto para cada caso:

- diffuse $kd = 0,54$
- specular $ks = 0,54$
- phong $n = 50$ $ks = 0,4$ $kd = 0,6$
- blinn $n = 17,5$ $ks = 0,17$ $kd = 0,35$
- torrance $rough = 0,13$ $nreal = 13,22$ $nimag = -0,78$ $ks = 0,19$ $kd = 0,98$
- poulin $d = 7,5$ $h = 0,5$ $n = 10,0$ $ks = 0,4$ $kd = 0,19$
- he $sigma = 0,4$ $tau = 2,4$ $lambda = 577$
- ward $sx = 0,3674$ $sy = 0,7765$ $ks = 0,05$ $kd = 0,4$
- lewis $n = 17,5$ $ks = 0,34$ $kd = 0,72$

⁶ *Editor de BRDFs* URL:
<<http://lsi.ugr.es/~rosana/software/wannabe/realistic.jar>>

CAPÍTULO 7. Utilidades software desarrolladas

- schlick $sc = 0,24$ $sr = 0,09$ $sp = 0,34$ $ly=T-F$ [$dc = 0,07$ $dr = 0,67$ $dp = 0,61$]
- oren $s = 0,57$ $r = 0,83$
- strauss $s = 0,81$ $m = 0,26$ $ks = 0,5$ $kd = 0,5$
- lafortune $cx = -1,1$ $cy = -0,9$ $cz = 1,2$ $n = 20,0$ $kp = 0,74$
- coupled $Rm = 1,0$ $Ro = 0,05$
- minnaert $k = 0,5$ $kd = 0,7$
- beardmaxwell $omega = 0,4$ $tau = 0,15$ $nreal = 3,5$ $nimag = -0,2$ $rs = 0,8$ $rd = 0,1$ $rv = 0,1$
- ashikhmin $nu = 10$ $nv = 10$ $Rs = 0,5$ $Rd = 0,5$

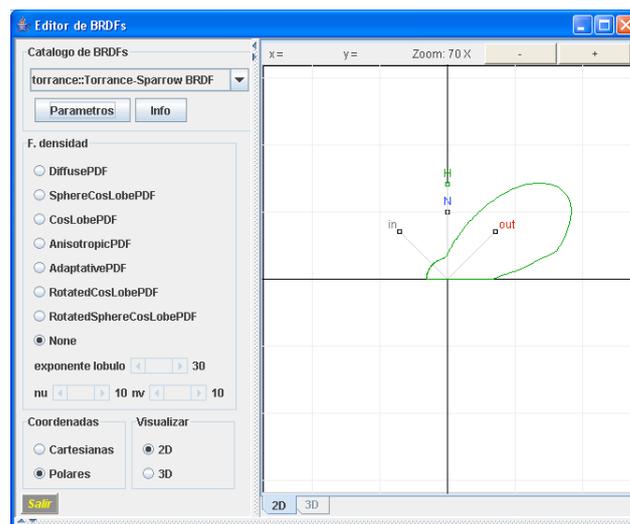


Figura 7.5: La herramienta permite obtener una gráfica de la función BRDF en coordenadas cartesianas 2D.

Una vez seleccionado el modelo, podemos modificar sus parámetros (botón PARÁMETROS de la figura 7.5), o bien obtener una pequeña descripción del significado de estos y el rango de valores permitidos (botón INFO en la figura 7.5). La visualización de la función se puede realizar de tres formas distintas:

- **Gráfica 2D** empleando coordenadas polares, tal y como se muestra en la imagen 7.5. Además de esta funcionalidad se permite realizar zoom sobre la gráfica, así como desplazamientos laterales de la imagen resultante (combinación CTRL + MOUSE DRAG).

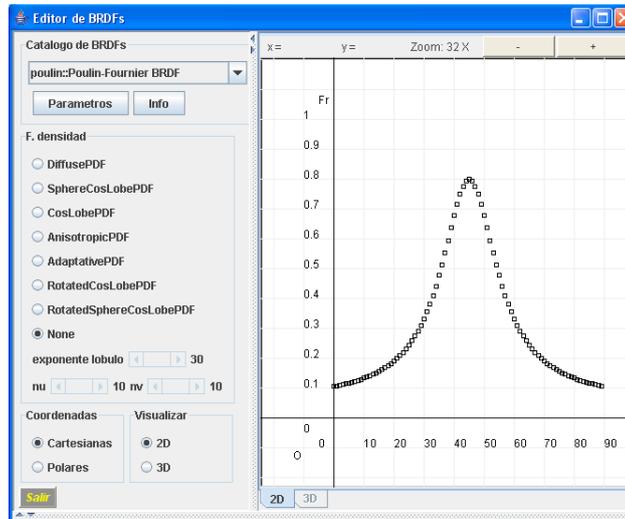


Figura 7.6: Incluye la posibilidad de realizar gráficas en coordenadas polares, representando la BRDF en el plano de incidencia.

- **Gráfica 2D** empleando coordenadas cartesianas, mostrada en la figura 7.6.
- **Gráfica 3D** empleando un sistema de coordenadas local, centrado en el vector normal $(0,0,1)$. Un ejemplo es mostrado en la imagen 7.7.

Sólo en la vista 2D en coordenadas polares se permite la modificación del vector incidente \mathbf{in} . Los cambios en dicha dirección repercuten en todas las gráficas. Los parámetros de la función pueden modificarse en cualquiera de los tres modos, visualizándose los cambios de forma inmediata.

Junto a la función BRDF podemos mostrar la función PDF. El usuario puede elegir de entre las distintas funciones PDF propuestas:

- Distribución uniforme o difusa. Sin parámetros.
- Lóbulo coseno en Ω . Activa la modificación del parámetro exponente que define a la función.
- Lóbulo coseno en \mathcal{S}^2 . Activa la modificación del parámetro exponente que define a la función. Tanto éste como el anterior elemento son variantes de un mismo tipo de muestreo. Ambos fueron vistos en la sección 5.2.1 (pág. 110).
- Función de densidad anisótropa basada en la BRDF de Ashikhmin. Vista en la sección 5.4. Activa los parámetros nu y nv encargados de modificar el grado de anisotropía.

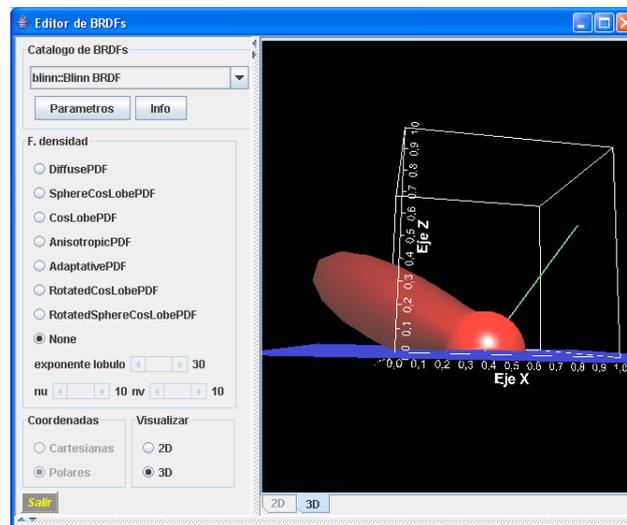


Figura 7.7: Es posible obtener una gráfica de la función BRDF en el espacio tridimensional.

- Función de densidad adaptativa, correspondiente a la aportación de esta tesis y que es objeto del capítulo 6 (más concretamente en la sección 6.2, página 126).

7.4. Muestreo por rechazo de la BRDF

Mediante la técnica de muestreo por rechazo podemos generar de forma sencilla valores aleatorios según una función de distribución deseada. Tan sólo es necesario ser capaces de evaluar la función y conocer el valor máximo de ésta en el dominio de integración.

De forma sencilla podemos visualizar los resultados de este método aplicado a funciones de distribución conocidas, así como los modelos de BRDFs implementados que son los que siguen:

- Estadísticas: Normal, Gamma, Chi-Cuadrado, Student, Beta, LogNormal, así como una función ejemplo.
- BRDFs: Diffuse, He, Lafortune, Lewis, Blinn, Oren-Nayar, Poulin-Fournier, Schlick, Strauss, Specular, Torrance-Sparrow, Ward, Coupled, Minnaert, Beard-Maxwell y Ashikhmin-Shirley.

La herramienta desarrollada en Java, e integrada dentro de *Wannabe Amazing*, implementa un método de muestreo adaptativo mediante el cálculo de subdominios que mejor se ajusten a la función objetivo. A través de la interfaz gráfica de usuario podremos modificar el número de muestras tomadas, el número de subintervalos a considerar, así como los parámetros que definen la función BRDF.

7.4. Muestreo por rechazo de la BRDF

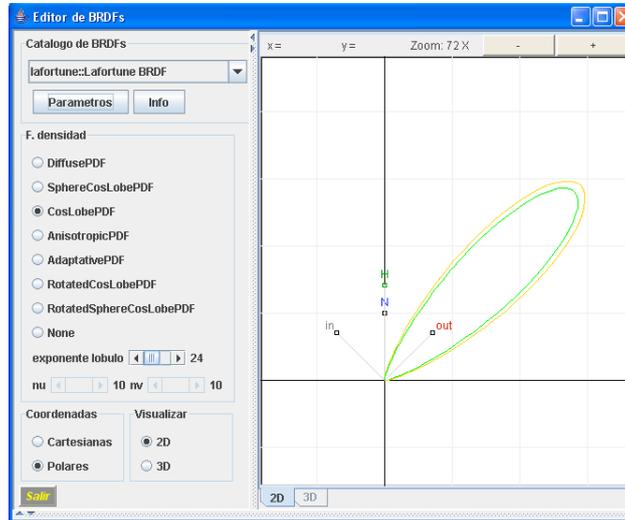


Figura 7.8: Es posible ver al mismo tiempo que la BRDF la gráfica de de la distribución coseno y editar el valor exponencial para aproximar la forma de la BRDF.

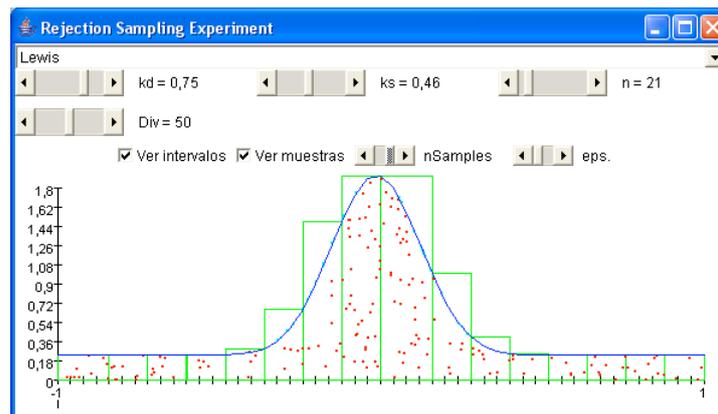


Figura 7.9: Muestreo por rechazo aplicado a distintas distribuciones, visto de forma gráfica y modificable por parámetros.

7.5. *Basic Path Tracer*

BPT es el acrónimo para BASIC PATH TRACER, la aplicación en C++ que implementa el algoritmo de *path-tracing* comentado en la sección 4.5.2. Todas las imágenes mostradas a modo de galería en el apéndice B han sido generadas con BPT.

Este programa toma como entrada un fichero de parámetros OP —fichero de opciones u *options file* que será tratado en la sección 7.7— y genera un fichero de salida correspondiente a la escena sintetizada con el algoritmo de *iluminación global* de trazado de caminos. Todos los parámetros cuentan con un valor por defecto y pueden ser omitidos, salvo el que indica el nombre de la escena GRF. Por tanto el fichero de parámetros es obligatorio.

Detallamos a continuación el significado de cada parámetro. Para un ejemplo de uso obsérvese la tabla 7.2.

- BPT_SCENE: Nombre del fichero grf que contiene la escena que se desea sintetizar.
- BPT_IMAGE_RESOLUTION: Valores enteros que indican la anchura y la altura de la imagen resultante.
- BPT_OUTPUT_IMAGE: Nombre del archivo con la imagen de salida (al que se le antepone el número de muestras tomadas). Las extensión determina el formato de salida. Valores posibles: TGA, JPG, EXR, PNG, RAW.
- BPT_VIEWER: Posición del observador en coordenadas de mundo.
- BPT_VLOOK: Parámetro de cámara que indica el punto al que se mira.
- BPT_PIXEL_RADIAL: Indica si se realiza muestreo uniforme (radial) del píxel.
- BPT_PIXEL_RANDOM: Indica si se realiza un muestreo aleatorio del píxel.
- BPT_BACKGROUND: Valores RGB entre 0 y 1 correspondientes al color de fondo.
- BPT_EM_BACKGROUND: Indica el nombre de la imagen que se utilizará como *environment map* en el sistema de rendering.
- BPT_PIXEL_SAMPLERES: Resolución de muestreo del píxel para evitar el problema de *aliasing*.
- BPT_RE_NSECRAYSROOT: Raíz del número de rayos secundarios generados por cada rayo primario durante el *path tracing*.
- BPT_DEPTH: Profundidad máxima del árbol *quadtrees* para el método adaptativo.

- BPT_PDF_NAME: Nombre de la PDF que realiza el muestreo de la BRDF. Valores posibles: DIFFUSE, SPECULAR, HEMISPHERE, SPHERE, ANISOTROPIC, DISK, FACTORED, CASCADECDF.
- BPT_PDF_LOBEEXPONENT: Parámetro exponente para las PDFs basadas en el muestreo del lóbulo coseno en Ω y \mathcal{S}^2 .
- BPT_PDF_ANISONu: Parámetro que controla la anisotropía de la PDF tipo ANISOTROPIC.
- BPT_PDF_ANISONv: Parámetro que controla la anisotropía de la PDF tipo ANISOTROPIC.
- BPT_PDF_MEMSTAT: Parámetro tipo lógico que indica si se ha de exportar al fichero *mem.op* el consumo de memoria en KB de la PDF en uso.
- BPT_QT_PRECOMPUTED: Parámetro para la PDF DISK. Número de *quadtrees* precalculados basándose en el ángulo θ .
- BPT_QT_ANISOTROPIC: Parámetro para la PDF adaptativa DISK. Indica si se discretiza la dirección incidente basándose en el ángulo incidente ϕ . Por defecto el valor es 1 (equivalente a un precálculo isótropo).
- BPT_QT_AVGNMAX: Parámetro para la PDF DISK. Valor máximo permitido para el número medio de intentos al aplicar *rejection sampling* en los nodos hoja (definido en la sección 6.2.1, pág. 127).
- BPT_QT_IMGRES: Parámetro para la PDF DISK. Resolución del precálculo de evaluaciones de la BRDF, que serán almacenadas como una imagen. Por defecto su valor es de 800, indicando una resolución de 800x800.
- BPT_QT_RELEASE: Parámetro de tipo lógico para la PDF DISK que indica si se ha de liberar o no, el recurso de memoria consumido por la imagen precalculada de evaluaciones de la BRDF. Por defecto su valor es *false*.
- BPT_PDF_FACTORS_RESU: Parámetro de tipo Vector2D de enteros utilizado en la factorización de la BRDF. Indica con qué resolución se efectúa la discretización en base al vector \mathbf{u} . Por ejemplo: 16 16
- BPT_PDF_FACTORS_RESP: Parámetro de tipo Vector2D de enteros utilizado en la factorización de la BRDF. Indica con qué resolución se efectúa la discretización en base al vector \mathbf{p} . Por ejemplo: 32 16
- BPT_PDF_FACTORS_JK: Parámetro de tipo Vector2D de enteros utilizado en la factorización de la BRDF. Indica el número de factores JxK empleados en la aproximación. Por ejemplo: 2 1

- `BPT_PDF_FACTORS_HALF`: Parámetro de tipo lógico de `FACTOREDPDF`. Cuando está a *true* indica que se reparametriza la BRDF en base al vector **h**.
- `BPT_PDF_FACTORS_FILE`: Indica el nombre del fichero que contiene la factorización de la BRDF precalculada, así como los valores necesarios para su muestreo. Por ejemplo, al indicar *factored_plastic* el programa busca y lee dos ficheros de texto *factored_plastic.txt* y *factored_plastic_cdf.txt*.
- `BPT_PDF_CDF_RES`: Parámetro de tipo Vector2D de enteros utilizado en el método `CASCADECDF` para el muestreo de BRDF adquiridas. Se trata de la resolución utilizada en la discretización del vector **h** ya que por defecto se reparametriza la BRDF en base al vector *half angle*. El valor por defecto es 256 x 32.
- `BPT_PDF_CDF_COMPRESS`: Parámetro de tipo lógico de `CASCADECDF`. Si se establece a *true* entonces la tabla CDF utilizada de forma uniforme, se comprime tras el cálculo de su representación adaptativa. Por defecto está a *false*.
- `BPT_RAYCASTING`: Parámetro de tipo lógico que activa/desactiva el modo *ray casting*. Cuando está activado, se calcula la intersección rayo-escena y se asigna como color en el punto, las componentes del vector normal de la superficie intersectada.
- `BPT_DISPLAYWINDOW`: Muestra el resultado de la imagen conforme se va procesando.
- `BPT_CHRONO_SAMPLING`: Indica si exportamos al fichero *samplingTime* el tiempo de muestreo.
- `BPT_CHRONO_RENDERING`: Indica si exportamos al fichero *renderingTime* el tiempo de síntesis.
- `BPT_CHRONO_PREBUILD`: Indica si exportamos al fichero *prebuildTime* el tiempo de precálculo del método de muestreo en uso.

BPT_SCENE	escena.grf
BPT_IMAGE_RESOLUTION	500 400
BPT_OUTPUT_IMAGE	buzon.exr
BPT_VIEWER	0.5 0.5 1
BPT_VLOOK	0.5 0.1 0
BPT_PIXEL_RADIAL	false
BPT_PIXEL_RANDOM	true
BPT_BACKGROUND	1 1 1
BPT_PIXEL_SAMPLERES	1 2
BPT_RE_NSECRAYSROOT	100
BPT_DEPTH	5
BPT_PDF_NAME	Disk
BPT_PDF_LOBEEXPONENT	30
BPT_PDF_ANISONU	25
BPT_PDF_ANISONV	25
BPT_PDF_MEMSTAT	true
BPT_QT_PRECOMPUTED	90
BPT_QT_ANISOTROPIC	1
BPT_QT_IMGRES	800
BPT_QT_AVGNMAX	1.7
BPT_RAYCASTING	false
BPT_DISPLAYWINDOW	false
BPT_CHRONO_SAMPLING	true
BPT_CHRONO_RENDERING	false
BPT_CHRONO_PREBUILD	false

Tabla 7.2: Ejemplo de un fichero OP destinado a guiar la ejecución de BPT.

7.6. *Mini Bpt*

La aplicación de la cual vamos a tratar en este apartado no es más que la ejecución modificada de BPT. MINIBPT es la combinación del visor de GRF y BPT, que no requiere de ningún fichero de parámetros (anteriormente mencionados como ficheros OP) para su ejecución. De hecho está diseñado para poder modificar dichos parámetros y ver de forma inmediata su resultado. Las funcionalidades de esta aplicación desarrollada en C++ (y probada en Linux) se enumeran a continuación:

1. Permite la carga de escenas .GRF iluminadas opcionalmente por un EM o *environment map*.
2. Permite la modificación mediante *sliders* o barras de desplazamiento de los vectores observador y punto de visión, correspondientes a los parámetros BPT_VIEWER y BPT_VLOOK. Para hacer efectivos los cambios se debe presionar el botón RUN, tras lo cual se actualiza el modelo en vista *ray-casting*.
3. Permite la selección de un modelo de BRDF y sus parámetros y visualizar la imagen resultante de aplicar el algoritmo de *path-tracing* al modelo de una esfera sobre un fondo reticulado.
4. Permite el ajuste por parte del usuario de los parámetros que determinan la calidad de la imagen sintética resultante, esto es, el número de muestras a utilizar por el estimador.
5. Una vez obtenida la imagen ya sea en vista *ray-casting* o renderizada por el *path-tracer* es posible almacenar la imagen en fichero usando alguna de las formatos permitidos: TGA, JPG, EXR, PNG, RAW

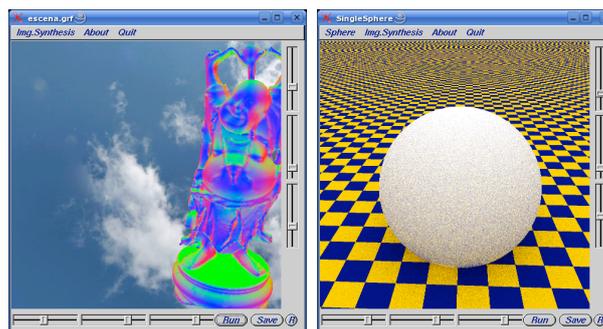


Figura 7.10: MINIBPT combina el software GRF y BPT para encontrar los mejores los parámetros de visualización y síntesis.

7.7. Generador gráfico de ficheros OP

Utilidad gráfica que permite la generación de ficheros de parámetros (anteriormente mencionados como ficheros OP por su extensión) para las distintas herramientas ubicadas en ZEUS. Está implementado en Java y se incluye dentro de las clases de *Wannabe Amazing* aunque también se puede descargar y ejecutar de forma independiente.⁷ Dos capturas de la interfaz Java se muestran en las figuras 7.11 y 7.12.

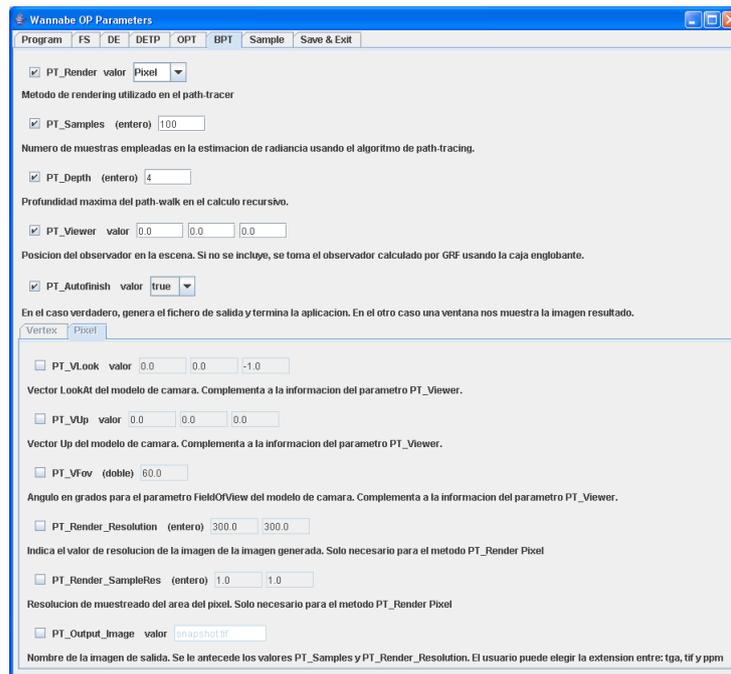


Figura 7.11: Herramienta gráfica para la edición controlada de parámetros de las distintas aplicaciones desarrolladas.

Tras seleccionar la aplicación resultante un listado de opciones pueden ser marcadas para incluir dicha opción en la salida. A su vez puede editar los valores que usa la aplicación por defecto para cada parámetro. Lo más ventajoso y útil a la hora de generar un fichero OP es que la aplicación chequea de forma automática las dependencias entre variables, que aparecerán habilitadas o deshabilitadas de forma oportuna.

⁷ *Wannabe OP creator* URL: <<http://lsi.ugr.es/~rosana/software/wannabe/opcreator.jar>>

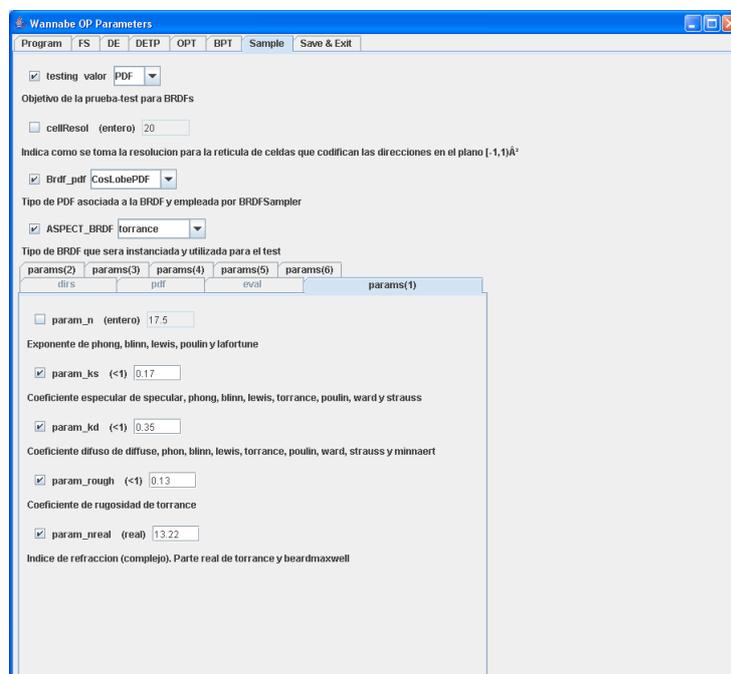


Figura 7.12: La edición de determinados parámetros pueden depender de la activación de otros.

7.8. *GLSL BRDF Viewer*: evaluación de la BRDF en la GPU

Hasta ahora la forma de visualizar la BRDF era como una función que se dibuja atendiendo a varios criterios. Para comprobar el aspecto que presenta un objeto cuando se le aplica una determinada BRDF (modelo y parámetros) es necesario calcular el valor de radiancia que se aplica a cada píxel, esto es, sintetizar la escena mediante un método de iluminación global.

Un método más rápido para comprobar la apariencia de un objeto dado una BRDF es disponer de la evaluación de estas funciones en soporte *hardware*. Por ello se ha desarrollado una herramienta que implementa los modelos con los que hemos trabajado en esta tesis, en la GPU.

Para este objetivo, hemos hecho uso del lenguaje de *shaders* GLSL⁸ que el estándar OpenGL 2.0 presentó el 10 de agosto de 2004. GLSL *OpenGL Shading Language* permite visualizar el aspecto de una superficie o un objeto, mediante un algoritmo, o *shader*, que se ejecuta en la tarjeta gráfica de forma mucho más rápida que en la CPU. Lo usamos para apreciar el aspecto del material que un objeto debe tener, aunque también podríamos utilizarlo para lograr otros efectos.

La aplicación desarrollada en C++ bajo Windows permite modificar los parámetros de las BRDFs (véase el panel derecho de la figura 7.13), que se activan o desactivan dependiendo del modelo seleccionado. También es posible modificar el color y las propiedades de la luz y el material, tal y como se presenta en la figura 7.14. Por defecto se utiliza como geometría la clásica tetera de GLUT aunque también es posible cargar modelos OBJ con la condición de que incluyan coordenadas de textura. Éstas son necesarias para el cálculo de los vectores tangente y binormal. La interfaz permite rotar el modelo, realizar *zoom* sobre éste, cambiar la posición de la fuente de luz puntual, así como aplicar iluminación precalculada como un *environment map*.

En el caso de que la tarjeta gráfica no sea capaz de soportar determinados operadores del lenguaje (necesarios en la implementación de la BRDF) en lugar de informar de error, se carga la BRDF de Blinn, modelo usado por defecto en las aplicaciones OpenGL.

⁸ *GLSL Language Specification*, v. 1.20 [consultado el 15-09-2007]:
<<http://www.opengl.org/registry/doc/GLSLangSpec.Full.1.20.8.pdf>>

7.8. GLSL BRDF Viewer: evaluación de la BRDF en la GPU

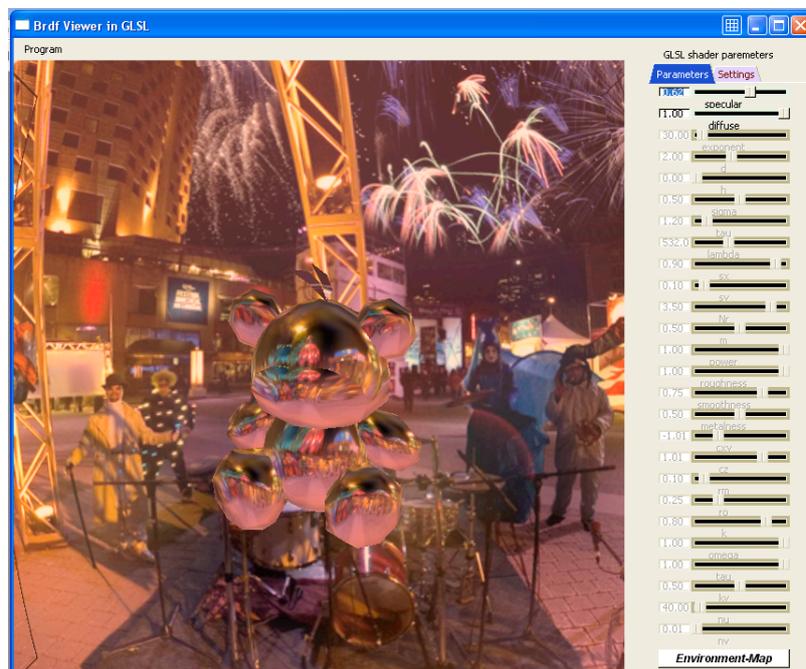


Figura 7.15: La interfaz permite cargar ficheros OBJ, como el TeddyBeard, así como aplicar iluminación precalculada, como el EM de Montreal.

7.9. Adaptive BRDF Subdivision: Visor de la estructura jerárquica

En esta sección tratamos una aplicación incluida en *Wannabe Amazing* basada en el editor de BRDFs ya comentado en la sección 7.3, que ha sido ampliada para soportar una interfaz más adaptada. Su distribución como aplicación independiente está disponible en la siguiente URL: <http://lsi.ugr.es/~rosana/software/wannabe/adaptiveDiskViewer.jar>.

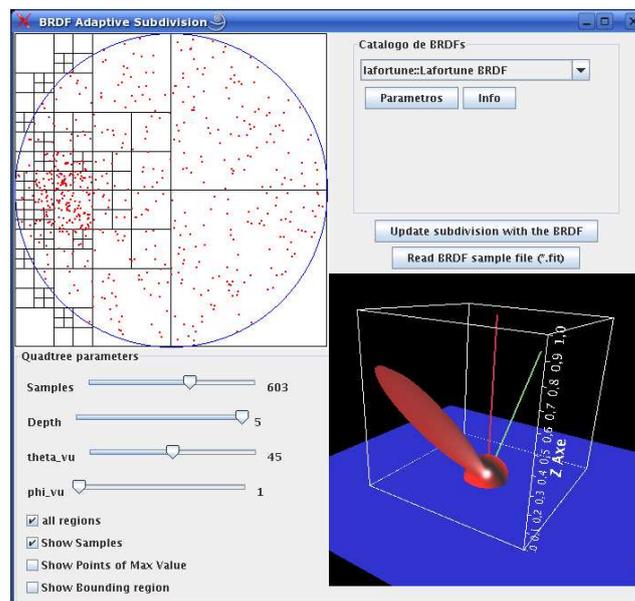


Figura 7.16: Modificando los selectores de la zona denominada *Quadtree parameters* podemos observar de forma interactiva, las distintas subdivisiones que se crean en el disco unidad.

Partimos del visor tridimensional del catálogo de funciones BRDFs, esto es, el editor de BRDFs en modo gráfica 3D, y le incluimos una gráfica del dominio disco unidad \mathcal{D}^2 . En este caso, además, nos informa de la subdivisión adaptativa que realiza el *quadtree* y nos permite modificar sus parámetros de forma interactiva.

Haciendo uso de la interfaz (mostrada en las figuras 7.16 y 7.17) podremos modificar los parámetros que definen la BRDF, el número de muestras, observador y demás parámetros de la subdivisión adaptativa. Además podemos usar esta herramienta como un visor estático de direcciones previamente generadas y almacenadas en ficheros .FIT. El resultado es tanto para el disco unidad como para la visualización 3D de la BRDF.

7.10. BRDF 3D Plot: Visualiza nubes de puntos

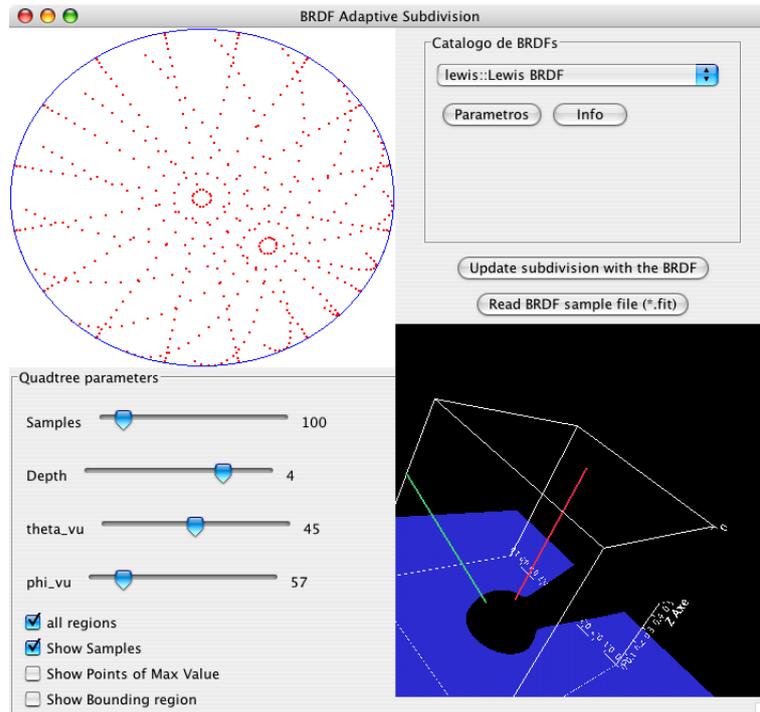


Figura 7.17: Podemos almacenar y visualizar cualquier nube de puntos. En este caso mostramos la diferencia entre usar la reparametrización basada en el vector *halfway* o no, para cualquier factorización de Lawrence.

7.10. *BRDF 3D Plot: Visualiza nubes de puntos*

Asistiéndonos de un fichero de texto, podemos visualizar valores discretos de la BRDF para una dirección concreta y una dirección incidente fija. De esta forma, podemos almacenar el conjunto de direcciones salientes que genera un método concreto de muestreo y posteriormente ver cómo se ajustan esas direcciones a la gráfica de la instancia de BRDF utilizada en el muestreo. El formato de fichero es muy sencillo y contiene la siguiente información:

1. Definición de la BRDF utilizando su notación GRF (detallado el apartado A.1.1).
2. Descripción de la dirección incidente. Tras el token *viewer* se especifica los valores en coordenadas polares de los ángulos $\theta_{\mathbf{u}}$ y $\phi_{\mathbf{u}}$.
3. Entre los tokens *begin* y *end* se listan los valores para cada vector de dirección. Esto es: $(x_{\mathbf{v}}, y_{\mathbf{v}}, z_{\mathbf{v}})$ y $f_r(\mathbf{u}, \mathbf{v})$.

A continuación se muestra un ejemplo de un fichero .fit con 7 vectores de dirección:

```
ashikhmin nu = 25 nv = 25 ks = 0.39 kd = 0.6
viewer (1.33763, 344.888)
begin
  (0.344659,-0.331753,0.878151) 1
  (0.048125,0.00957266,0.998795) 0.85
  (0.327274,-0.329171,0.885742) 1
  (0.0407984,0.0272606,0.998795) 0.85
  (0.312922,-0.319833,0.894308) 1
  (0.0272606,0.0407984,0.998795) 0.85
  (0.30379,-0.30516,0.902546) 1
end
```

El fichero se guarda con la extensión FIT y puede ser leído desde *3D BRDF Plot*. El resultado será la visualización en 3D de una nube de puntos verde junto a la instancia de la BRDF en rojo. A su vez obtendremos la proyección de estos vectores en el disco unidad, como puntos rojos. Puede verse el ejemplo de la figura 7.18. Esta herramienta nos ha sido de gran utilidad en la detección de incorrecciones en la implementación de los distintos métodos de selección de direcciones empleados en esta tesis.

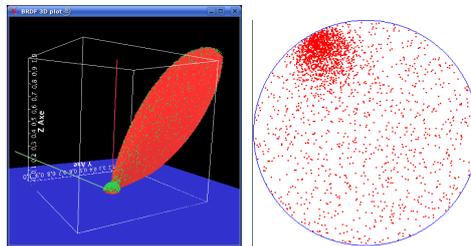


Figura 7.18: Nube de 2500 puntos en 3D y 2D para la BRDF de Ward y la definición del observador en un fichero FIT. Las direcciones fueron muestreadas con nuestro método adaptativo.

7.11. *EM Viewer*: Muestreo del Environment Map

Tal y como se ha desarrollado en la sección 6.5 es de nuestro interés evaluar la mejora en comportamiento de la técnica RIS (*Resampling Importance Sampling* visto con detalle en el apartado 4.4.10). Hemos utilizado nuestro método adaptativo para generación de direcciones en base a la BRDF (multiplicado por un factor coseno) para la obtención de un estimador de Monte Carlo proporcional al producto $f_r \cdot L_i$ siguiendo los pasos del algoritmo [19].

Como punto de partida desarrollamos una utilidad en Java que nos permitiese evaluar de forma visual la distribución de vectores de dirección en la esfera de direcciones en condiciones parametrizables:

1. Cambio en el observador mediante dos barras de desplazamiento que controlan los valores $\theta_{\mathbf{u}}$ y $\phi_{\mathbf{u}}$.
2. La imagen equirectangular utilizada como EM, procedente de una lista desplegable. Una vez cargado el *environment map*, se calcula automáticamente su imagen de luminancias, empleada como CDF o función de distribución acumulada. Un *checkbox* nos permitirá alternar entre ambas imágenes.
3. Dos barras de desplazamiento nos permitirán introducir dos valores N número de muestras resultantes de la función de densidad destino y M número de muestras propuestas por la función PDF origen. Estos dos valores están íntimamente ligados, cumpliendo la restricción $M \geq N$.

Una vez establecidos los parámetros anteriores podemos ejecutar el algoritmo de muestreo con RIS presionando el botón *Run Experiment*. Como resultado veremos direcciones distribuidas sobre el *environment map*. De forma selectiva podemos decidir ver las direcciones obtenidas únicamente por una de las dos distribuciones o por ambas. Una captura de esta interfaz es mostrada en la figura 7.19. Gracias a esta utilidad, la implementación en el sistema de síntesis realistas BPT fue directa y sin demasiados errores de desarrollo.

Esta aplicación, como la mayoría de las anteriores, se distribuye mediante fichero JAR ejecutable. Se puede descargar directamente de la siguiente dirección:

<<http://lsi.ugr.es/~rosana/software/wannabe/emViewer.jar>>

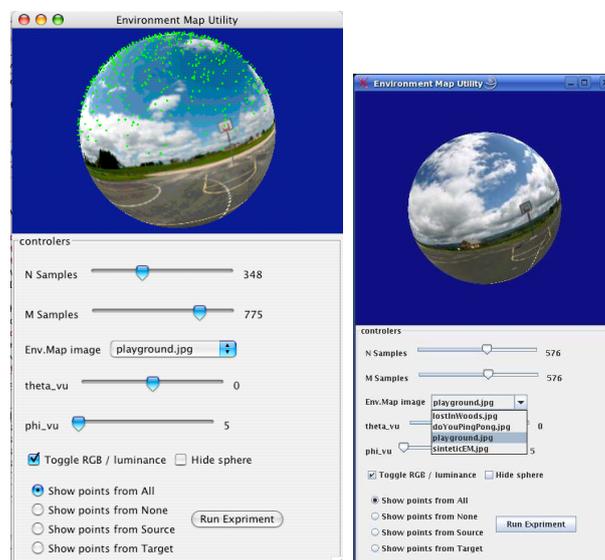


Figura 7.19: *Resampling* aplicado al muestreo del producto de la BRDF y un *Environment Map*.

Conclusiones y trabajos futuros

Llegados a este punto y después de muchas páginas, secciones, definiciones, fórmulas y tablas me acuerdo de cierta leyenda cristiana según la cual había una vez un ermitaño que corría por el desierto gritando:

- ¡Tengo una respuesta, tengo una respuesta! ¿Quién tiene una pregunta?

En esta memoria de tesis doctoral se han expuesto ciertos conceptos relacionados con la *informática gráfica* y más en concreto con las técnicas de *iluminación global* y *visualización realista* para ser capaces de entender el contexto del problema, nuestra motivación y el resultado obtenido: **un método de muestreo eficiente de BRDFs** en el contexto de algoritmos de Monte Carlo para iluminación global.

Recordemos que en este documento se ha presentado **un estudio de los modelos de reflexión** existentes en la literatura de *informática gráfica*, y que queda cubierto por el capítulo 3. El lector tiene a su disposición en la sección de bibliografía las referencias originales. Sin embargo en esta memoria doctoral hemos logrado unificar la notación empleada en todos los trabajos, de forma que es más fácil compararlos y ver semejanzas y diferencias existentes. No debemos olvidar que este estudio ha sido completado por una serie de herramientas software que permiten un estudio comprensivo de dichas funciones BRDFs al poder ver el comportamiento de la función en distintas situaciones. Contamos con un editor de BRDFs que permite visualizar esta función en coordenadas polares y cartesianas, ante variaciones en sus parámetros. A su vez, la BRDF puede ser editada en 3D y ver su resultado sobre una geometría, gracias a nuestro editor en la GPU.

Este estudio se ha completado en el capítulo 5 con el **estudio e implementación de distintas técnicas de muestreo** existentes para determinadas funciones BRDFs. Una cosa nos ha quedado clara, y es que si bien no existe ningún modelo analítico de reflexión que caracterice adecuadamente todos los materiales existentes en el mundo real, tampoco existe un método de muestreo que cumpla todos los requerimientos que podemos exigirle a este tipo de procedimientos: que sea rápido, que disminuya el error del estimador de Monte Carlo, que sea genérico, que no consuma mucha memoria y que sea fácilmente parametrizable.

Ciertamente el **método de muestreo exacto para BRDFs genéricas** que hemos presentado en el capítulo 6 se ajusta bastante bien a las características ideales anteriormente comentadas. Se basa en el uso del disco unidad como dominio de integración de la BRDF, y se adapta a las características de ésta mediante el uso de una estructura jerárquica o *quadtree*. Hemos encontrado algunas formas de optimizar el algoritmo, tanto en su construcción como en su uso. También lo hemos comparado favorablemente con otros métodos analíticos de muestreo y aquellos basados en el uso de tablas de valores de reflectancia reales, procedentes de mediciones de la reflectancia de un material. Los resultados han sido satisfactorios en cuanto a disminución del error, aplicabilidad a cualquier material y escena, y la sencillez de su parametrización. Quizá se pueda pensar que sus altos tiempos de muestreo no justifiquen su uso. Yo creo que depende en gran medida de la BRDF que se muestrea en dicho momento. Un modelo empírico, simple y que se ajusta a pocos materiales, puede encontrar otras técnicas más rápidas de muestreo. Una BRDF compleja, que se basa en características físicas, o bien una BRDF adquirida con millones de datos, pueden encontrar aquí un método de muestreo ideal. No olvidemos que en cualquier momento pueden aparecer nuevos modelos de reflectancia. El muestreo de dichas nuevas funciones queda garantizado siempre y cuando podamos evaluarlo para un par de direcciones en la semiesfera.

- *¡Tengo un método, tengo un método! ¿Quién tiene una aplicación?*

El método de Monte Carlo es, en general, un procedimiento matemático que nos permite simular cualquier sistema físico, matemático, químico, biológico o de cualquier otra rama de la ciencia que tenga unas leyes capaces de poder ser traducidas a un lenguaje matemático. Realmente es una herramienta de investigación que, mediante el muestreo artificial, nos permite operar numéricamente sistemas complejos. Por mencionar algunas aplicaciones interesantes del método de Monte Carlo tenemos:

- Simulación de los diferentes procesos de fabricación de dispositivos desde la oblea de silicio, hasta la caracterización y comprobación de las curvas I-V de un circuito integrado.
- Diseño de reactores nucleares.
- Aplicaciones en el campo de la radioterapia, mediante la simulación de cascadas electrón-fotón.
- Estudio de la densidad y flujo del tráfico.
- Econometría, pronóstico del índice de la bolsa, o bien, análisis de riesgos en proyectos de inversión, etc.
- Criptografía y otros programas de ordenador como el campo de la *informática gráfica* o la simulación acústica de edificios.

Para la consecución del trabajo presentado en esta memoria de tesis doctoral, he ido implementando las herramientas software que he necesitado en cada momento. Me he visto desde el punto de vista del usuario de aplicaciones gráficas y me he encontrado con distintas necesidades como el ser capaz de cargar modelos de otras escenas, editar los parámetros de una BRDF y asignarlos a una malla concreta de la geometría. Estas operaciones pueden ser más comunes de lo que se piensa. Es por ello que la aplicación multiplataforma basada en la librería gráfica Java3D **Wannabe Amazing** puede ser de utilidad a otras personas. Es gratificante saber que tanto estudio ha tenido su parte práctica en determinadas aplicaciones software. Estas han sido comentadas en el capítulo 7.1. Por supuesto que muchas de estas herramientas pueden ser consideradas más específicas de este trabajo, como el visor de la estructura adaptativa, del muestreo empleando *resampling*, o bien los visores del formato gráfico GRF.

Trabajos futuros

El principal problema de las técnicas basadas en muestreo por rechazo es la falta de habilidad para dar distribuciones aleatorias con buenas propiedades. Prácticamente ninguna implementación que se base en el método de Monte Carlo hará uso de un generador uniforme de aleatorios. Lo habitual es basarse en algún tipo de estratificación, o bien en secuencias quasi-aleatorias (QMC). El método que aquí presentamos no es del todo compatible con las anteriores técnicas y de ahí que nos planteemos dicha posibilidad en el futuro.

Actualmente, nuestro método adaptativo de muestreo no realiza un mapeo de direcciones del disco unidad a la semiesfera preservando el área. El hecho de usar direcciones proyectadas en el disco unidad, nos permite considerar de forma implícita el término coseno. Sin embargo ese valor no es considerado cuando un punto es reprojectado en la esfera de direcciones. Es deseable que un método de mapeo preserve el área, de esa forma la integral en una región en el disco debería dar el mismo valor que al integrar el ángulo sólido de los vectores de dirección equivalentes en la esfera. Este hecho es usado en [WNLH06] donde se propone una función de mapeo directamente en la semiesfera. Estudiaremos este punto, ya que las regiones cuya distancia al centro del disco es próximo al radio de este, el coseno toma valores cercanos a cero. Este valor no es deseable cuando nos encontramos con BRDF especulares, que para determinados ángulos de entrada contaría con valores importantes que no quedarían adecuadamente representados.

Para terminar y concluir este trabajo, al que le he dedicado mucho tiempo, mucho esfuerzo y un puñado de años, quería decir que es, a mi punto de vista, una forma de empezar mi *investigación real* de la mano de mi propia experiencia e intuición, aunque siempre con el apoyo de otros compañeros. Este no es un final, es un principio.

Rosana Montes

APÉNDICE A

El formato de escenas GRF

A.1. Descripción general del formato

Mediante la herramienta multiplataforma *Wannabe Amazing* [MUR⁺02] podemos importar escenas procedentes de diversos formatos gráficos. Se le añade información de importancia en el contexto de la síntesis realista de imágenes, como es la instancia de BRDF que se asocia a cada superficie. Todo esto y opcionalmente otra información de utilidad, se exporta en un formato gráfico de uso interno.

Este formato propio denominado GRF (*Granada File*) permitirá describir con mayor riqueza sintáctica los aspectos de iluminación de un modelo de mallas. Tanto la descripción del formato GRF como ejemplos de ficheros generados pueden encontrarse en la siguiente dirección web: <<http://lsi.ugr.es/~rosana>>. Mostramos a continuación el formato empleando la notación BNF para representarlo; a su vez se irá comentando poco a poco los aspectos más relevantes del formato GRF.

```
escena ::= lista_bloques | nada.  
lista_bloques ::= bloque | bloque lista_bloques  
| nada.  
bloque ::= bloque_malla | bloque_fuente_luz  
| bloque_include | bloque_transformacion  
| bloque_definicion | bloque_descripcion  
| nada.
```

De forma general una escena va a estar dividida en secciones. Cada sección aporta

una semántica extra a la dada por el conjunto de mallas que describen la geometría de los objetos de la escena. Las secciones permiten:

1. La denominación de los elementos. En lugar de repetir una y otra vez la descripción de una misma propiedad, es posible asignar un identificador a la declaración de un elemento, para posteriormente nombrarlo cuando sea necesario.
2. Instanciación. Denominamos instanciación, o uso, a la reutilización de una definición anterior. Es una forma de ahorrar espacio en el fichero y de hacerlo más legible.
3. Elementos reutilizables. Con ellos se permite la inclusión de toda información sobre mallas, declaraciones y uso que estén descritas en otro fichero GRF, dentro del actual, que se realiza indicando el nombre del mismo, siendo el resultado igual que si se hubiese realizado un típico Copiar / Pegar bloque. La diferencia estriba en que aparece más estructurada la información, sobre todo para el caso de escenas complejas.
4. Definiciones. Sección que nos permite asociar el significado de un bloque a un identificador como comodidad para su posterior uso. Se admite más de una definición elemental, aunque su ámbito es local al bloque en el que se realizó la declaración.
5. Asignar texturas a las caras. Es posible asignar más de una textura a la malla mediante el uso de identificadores y de la cláusula *use*. A su vez, los vértices contendrán la información correspondiente a las coordenadas de textura para su correcta visualización.
6. Realizar transformaciones a la geometría. Se trata de una operación común consistente en la rotación entorno a un eje, y que puede verse realizada mediante una cláusula *rotation* seguida de un vector que indique el ángulo de rotación que se debe aplicar. Otras transformaciones permitidas son la escala y la traslación.

```
bloque_fuente_luz      ::= light begin region [definicion_color]
                        end .
region                 ::= vector3D arista arista.
arista                 ::= vector3D.
bloque_include         ::= include nombreFichero
bloque_transformacion ::=
                        transform transformacion
                        begin lista_bloques end
transformacion         ::= transformacion transformacion_elemental
                        | transformacion_elemental
transform_elemental    ::= scale vector3D | rotation vector3D
                        | translation vector3D
```

<i>bloque_definicion</i>	::= def <i>definiciones</i> begin <i>lista_bloques</i> end
<i>definiciones</i>	::= <i>definiciones</i> <i>definicion definiciones_elementales</i>
<i>definiciones_elementales</i>	::= <i>identificador = cuerpo_definicion</i>
<i>cuerpo_definicion</i>	::= <i>definicion_textura</i> <i>definicion_color</i> <i>definicion_brdf</i>
<i>definicion_textura</i>	::= texture <i>nombreFichero</i>
<i>definicion_color</i>	::= rgb <i>colorValue</i>
<i>definicion_brdf</i>	::= brdf <i>nombreFichero</i> brdf <i>instancia_BRDF</i> .
<i>definicion_pdf</i>	::= pdf <i>instancia_PDF</i> .
<i>uso</i>	::= <i>uso_textura</i> <i>uso_color</i> <i>uso_brdf</i> .
<i>uso_textura</i>	::= texture <i>identificador</i> .
<i>uso_color</i>	::= color <i>identificador</i> .
<i>use_brdf</i>	::= brdf <i>identificador</i> .
<i>use_pdf</i>	::= pdf <i>identificador</i> .

De esta serie de elementos, los susceptibles para que se les asigne un identificador son: texturas, color y BRDFs. No existe una sintaxis común como el caso de los nodos *def* del formato *vrml*, optándose por una palabra clave que informe del tipo de descripción asociada al identificador. Más parecidos son los casos de instanciación, que comienzan con la palabra clave *use* y les sigue la palabra empleada en la definición y el identificador. Tanto en las declaraciones como los usos de ésta no existe ningún tipo de restricción en cuanto a orden de aparición y posición dentro del fichero de escena. El ámbito de la cláusula *use* se mantiene desde el momento de su aparición hasta el final de la malla, o bien hasta la aparición de una cláusula similar.

La definición de la BRDF viene dada generalmente por la declaración explícita del nombre de la función y sus parámetros. Para mayor facilidad en su consulta, hemos dejado la descripción de esa sentencia en la sección A.1.1 de este apéndice. Una función BRDF será muestreada usando la función PDF definida para la escena. Sin embargo, cuando en una escena se utilizan distintas BRDFs puede ser más conveniente distinguir distintas funciones de muestreo. Por ello el formato GRF permite la definición de la PDF, que vendrá seguida por una declaración explícita del nombre de la función y de sus parámetros, al igual que se hace con la BRDF. Consultese la sintaxis BNF para la función PDF en la sección A.1.2.

Nuestro formato expresa cada malla como una lista de vértices y una enumeración de caras. Adicionalmente se pueden incluir normales a los vértices, color e irradiancia en cada vértice, así como reflectancia y otras propiedades del material (directamente o bien usando el nombrado de elementos y la posibilidad de realizar *includes* de otro fichero). Dentro de cada malla se listan los vértices que van a aparecer junto

Apendice A

a la información asociada a cada uno. Posteriormente se describen las caras como una lista de vértices dados mediante un índice o posición ocupada dentro de la lista de coordenadas de esa malla. De esta forma si cada vértice aparece en más de una cara, que es lo usual, no es necesario dar de nuevo toda esa información. Se permiten dos sintaxis distintas de expresar las caras, semejantes a la empleada por otros formatos.

<i>bloque_malla</i>	::= begin mesh listaVertices listaCaras end mesh .
<i>listaVertices</i>	::= begin vertexs vertices end vertexs .
<i>vertices</i>	::= <i>vertice</i> <i>vertice vertices</i> <i>nada</i> .
<i>vertice</i>	::=[normal vector3D] [listaIrradiancias] [uv vector2D] vector3D
<i>listaIrradiancias</i>	::= <i>irradiancia</i> begin list (rayo irradiancia)* end list .
<i>rayo</i>	::= ray vector3D .
<i>irradiancia</i>	::= irad vector3D
<i>vector3D</i>	::=(valor, valor, valor) .
<i>vector2D</i>	::=(valor, valor) .
<i>listaCaras</i>	::= begin faces caras end faces .
<i>caras</i>	::=[definicion_color] [definicion_brdf] [definicion_textura] cara cara caras <i>nada</i> .
<i>cara</i>	::= <i>nvertices posicion*</i> (<i>posicion, posicion*</i>) .
<i>nvertices</i>	::= <i>literal_entero</i> .
<i>posicion</i>	::= <i>literal_entero</i> .

Cuando se trabaja con lista de irradiancias, es necesario proporcionar la definición del observador. De esta forma el visor GRF que analiza estos datos, puede emplear en tiempo real las irradiancias precalculadas y almacenadas con este formato para la visualización. Los ficheros con esta información utilizan la extensión **igrf**.

<i>bloque_descripcion</i>	::= params begin lista_parametros end
<i>lista_parametros</i>	::= dependent observador
<i>observador</i>	::= view vector3D .

Otros elementos que aparecen en la descripción BNF son:

<i>nombreEscena</i>	<code>::=nombreFichero .</code>
<i>nombreFichero</i>	<code>::=cadena .</code>
<i>colorValue</i>	<code>::=(valor, valor, valor) .</code>
<i>cadena</i>	<code>::=" conjunto de caracteres Unicode " .</code>
<i>valor</i>	<code>::=literal_real literal_entero .</code>
<i>literal_real</i>	<code>::=(decimales "." [decimales] [exponente] [sufijo_tipo]) ("." decimales [exponente] [sufijo_tipo]) (decimales [exponente] [sufijo_tipo]) .</code>
<i>decimales</i>	<code>::= 0..9 .</code>
<i>exponente</i>	<code>::=" e " ["+" "-"] decimales .</code>
<i>sufijo_tipo</i>	<code>::=" f " " d " .</code>
<i>literal_entero</i>	<code>::=0..9 .</code>
<i>identificador</i>	<code>::=a..z, A..Z, ,0..9, \$, _ .</code>
<i>comentario</i>	<code>::=// cadena .</code>
<i>nada</i>	<code>::= ;</code>

A.1.1. Lista de BRDFs y sus parámetros expresada en BNF

<i>definicion_brdf</i>	<code>::=brdf nombreFichero brdf instancia_BRDF .</code>
<i>instancia_BRDF</i>	<code>::=PerfectDiffuse PerfectSpecular Phong Blinn Torrance-Sparrow Cilindrico-PF HTSG Ward Lewis Schlick Oren-Nayar Strauss Lafortune Shirley Minnaert BeardMaxwell combinacion_BRDFs MerlDB_BRDF .</code>
<i>PerfectDiffuse</i>	<code>::=diffuse kd=valor .</code>
<i>PerfectSpecular</i>	<code>::=specular ks=valor .</code>
<i>Phong</i>	<code>::=phong n=valor ks=valor kd=valor .</code>
<i>Blinn</i>	<code>::=blinn n=valor ks=valor kd=valor .</code>
<i>Torrance-Sparrow</i>	<code>::=torrance rough=valor nreal=valor nimag=valor ks=valor kd=valor .</code>
<i>Cilindrico-PF</i>	<code>::=poulin d=valor h=valor n=valor ks=valor kd=valor .</code>
<i>HTSG</i>	<code>::=he sigma=valor tau=valor lambda=valor .</code>
<i>Ward</i>	<code>::=ward sx=valor sy=valor ks=valor kd=valor .</code>

Apendice A

<i>Lewis</i>	::= lewis <i>n</i> =valor <i>ks</i> =valor <i>kd</i> =valor .
<i>Schlick</i>	::= schlick <i>SC</i> =valor <i>Sr</i> =valor <i>Sp</i> =valor <i>Ly</i> =valor_logico [<i>DC</i> =valor <i>Dr</i> =valor <i>Dp</i> =valor] .
<i>Oren-Nayar</i>	::= oren <i>s</i> =valor <i>r</i> =valor .
<i>Strauss</i>	::= strauss <i>s</i> =valor <i>m</i> =valor <i>ks</i> =valor <i>kd</i> =valor .
<i>Lafortune</i>	::= lafortune <i>Cx</i> =valor <i>Cy</i> =valor <i>Cz</i> =valor <i>n</i> =valor <i>kp</i> =valor.
<i>Shirley</i>	::= coupled <i>Rm</i> =valor
<i>Minnaert</i>	::= minnaert <i>k</i> =valor <i>kd</i> =valor
<i>BeardMaxwell</i>	::= beardmaxwell <i>omega</i> =valor <i>tau</i> =valor <i>nreal</i> =valor <i>nimag</i> =valor <i>Rs</i> =valor <i>Rd</i> =valor <i>Rv</i> =valor
<i>combinacion_BRDFs</i>	::= combine <i>base_BRDF</i> , <i>base_BRDF</i> *
<i>MerlDB_BRDFs</i>	::= measured <i>nombreFichero</i>
<i>base_BRDF</i>	::=valor <i>identificador</i> <i>nada</i> .

A.1.2. Lista de PDFs y sus parámetros en BNF

<i>definicion_pdf</i>	::= pdf <i>instancia_PDF</i> .
<i>instancia_PDF</i>	::= <i>Uniform_PDF</i> <i>Specular_PDF</i> <i>Sphere_PDF</i> <i>Hemisphere_PDF</i> <i>Adaptive_PDF</i> <i>Anisotropic_PDF</i> <i>Rejection_PDF</i> <i>Factored_PDF</i> <i>Cascade_CDF</i>
<i>Uniform_PDF</i>	::= diffuse
<i>Specular_PDF</i>	::= specular
<i>Sphere_PDF</i>	::= sphere <i>exp</i> =valor.
<i>Hemisphere_PDF</i>	::= hemisphere <i>exp</i> =valor
<i>Adaptive_PDF</i>	::= adaptive <i>nQT</i> =valor <i>nAnisoQT</i> =valor
<i>Anisotropic_PDF</i>	::= anisotropic <i>nu</i> =valor <i>nv</i> =valor
<i>Rejection_PDF</i>	::= reject
<i>Factored_PDF</i>	::= factored <i>uTh</i> =valor <i>uPh</i> =valor <i>pTh</i> =valor <i>pPh</i> =valor <i>J</i> =valor <i>K</i> =valor <i>half</i> =valor_logico.
<i>Cascade_CDF</i>	::= cascadeCDF <i>nTh</i> =valor <i>nPh</i> =valor <i>compress</i> =valor_logico.

A.1.3. Un ejemplo sencillo

Mostramos a continuación el clásico ejemplo de un cubo tridimensional, expresado en el formato GRF.

```
// Wannabe Amazing sep 01, 2001, 12:58
// Bounding Box :
//   Lower=(-2.797451055, -3.276146411, -2.716787639)
//   Upper=(2.5823486120, 2.1036532564, 2.6630120283)

def  cromo = brdf ward sx=0.05 sy=0.3 ks=0.05 kd=0.4 begin
begin mesh
  brdf cromo
  begin vertexs
    normal(0.0, 0.0, 1.0)
    (2.0,-1.0, 0.5)
    normal(0.0, 0.0, 1.0)
    (2.0, 1.0, 0.5)
    normal(0.0, 0.0, 1.0)
    (-2.0, 1.0, 0.5)
    normal(0.0, 0.0, 1.0)
    (-2.0,-1.0, 0.5)
    normal(0.0, 0.0,-1.0)
    (-2.0,-1.0,-0.5)
    normal(0.0, 0.0,-1.0)
    (-2.0, 1.0,-0.5)
    normal(0.0, 0.0,-1.0)
    (2.0, 1.0,-0.5)
    normal(0.0, 0.0,-1.0)
    (2.0,-1.0,-0.5)
  end vertexs
  begin faces
    rgb(0.3,0.8,1.0)
    (0,1,2,3)
    (4,5,6,7)
```

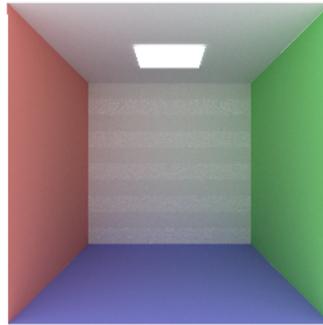
```
(7,6,1,0)
(3,2,5,4)
(1,6,5,2)
(3,4,7,0)
end faces
end mesh
end
```

A.1.4. Un segundo ejemplo

```
//ply2grf dragon200K.ply 20/07/2006
include "light.grf"
transform
  rotation (1.570796,0,0)
  translation(0,0.5,0)
begin
  include "light.grf"
end
transform
  rotation (1.570796,0,0)
  translation(0,0.75,-0.75)
begin
  include "wall.grf"
end
def
  dragon = brdf ward sx=0.3674 sy=0.7765 ks=0.05 kd=0.4
  suelo = brdf lafortune cx=-1 cy=-1 cz=1 n=20.0 kp=1
begin
  include "suelo.grf"
  transform
    scale (3, 3, 3)
    translation (0,-0.15, 0)
  begin
    include "dragon200K.grf"
  end
end
end
```

APÉNDICE B

Escenas GRF renderizadas



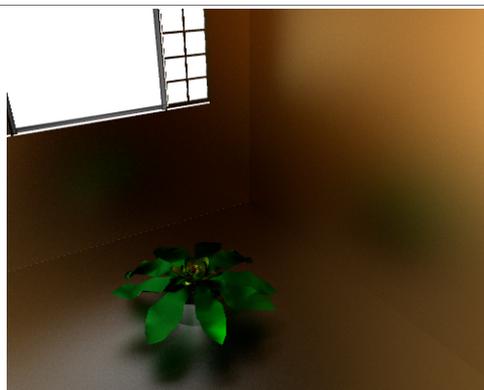
ESCENA	VÉRTICES	CARAS	BRDFS	MALLAS
Cornell Box	605	1000	1	5
MUESTRAS	900	T.MUESTREO	397.402	secs.

Figura B.1: Aplicación del método *Adaptive Disc* a la generación de la escena de CORNELL.



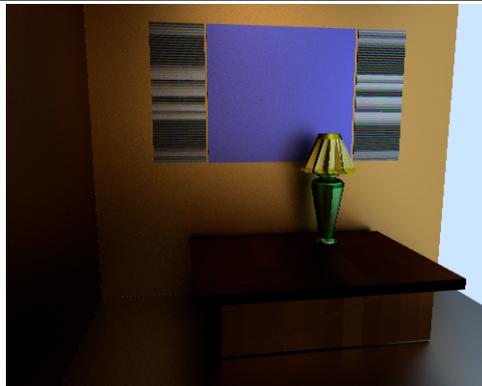
ESCENA	VÉRTICES	CARAS	BRDFS	MALLAS
Planta	1184	1372	1	49
MUESTRAS	400	T.MUESTREO	38.91	secs.

Figura B.2: Aplicación del método *Adaptive Disc* a la generación de la escena de PLANTA.



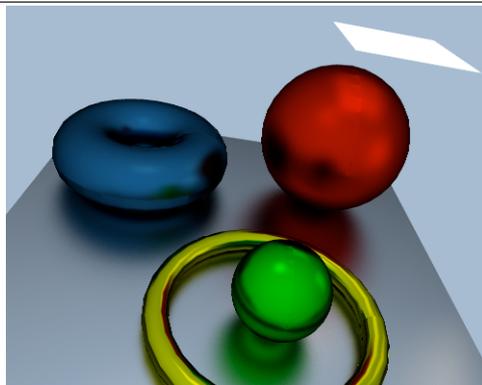
ESCENA	VÉRTICES	CARAS	BRDFS	MALLAS
Living room 1	3311	4608	7	68
MUESTRAS	400	T.MUESTREO	447.839	secs.

Figura B.3: Aplicación del método *Adaptive Disc* a la generación de la escena de LIVING ROOM 1.



ESCENA	VÉRTICES	CARAS	BRDFS	MALLAS
Living room 1	3311	4608	7	68
MUESTRAS	400	T.MUESTREO	466.254	secs.

Figura B.4: Detalle lámpara, en la escena de LIVING ROOM 1.



ESCENA	VÉRTICES	CARAS	BRDFS	MALLAS
SimpleScene	5086	7776	4	5
MUESTRAS	400	T.MUESTREO	144.95	secs.

Figura B.5: Aplicación del método *Adaptive Disc* a la generación de la escena de SIMPLESCENE.



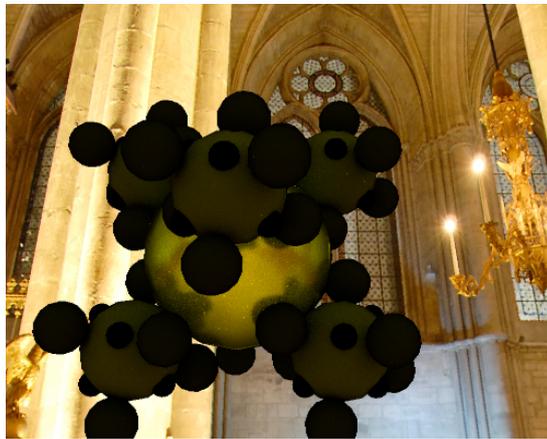
ESCENA	VÉRTICES	CARAS	BRDFS	MALLAS
BagueScene	6179	8744	4	9
MUESTRAS	400	T.MUESTREO	212.87	secs.

Figura B.6: Aplicación del método *Adaptive Disc* a la generación de la escena de BAGUESCENE.



ESCENA	VÉRTICES	CARAS	BRDFS	MALLAS
Buzon	9362	18398	1	1
MUESTRAS	400	T.MUESTREO	121.715	secs.

Figura B.7: Aplicación del método *Adaptive Disc* a la generación de la escena de BUZÓN.



ESCENA	VÉRTICES	CARAS	BRDFS	MALLAS
Multisphere	12534	18696	4	84
MUESTRAS	10000	T.MUESTREO	34178.9	secs.

Figura B.8: Aplicación combinada de *resampling* y *Adaptive Disc* a la escena MULTISPHERE.



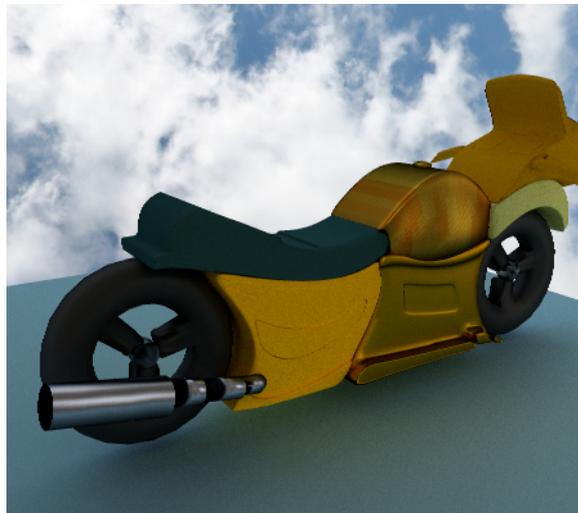
ESCENA	VÉRTICES	CARAS	BRDFS	MALLAS
Multisphere	12534	18696	5	84
MUESTRAS	200	T.MUESTREO	553.81	secs.

Figura B.9: MULTISPHERE con BRDF adquiridas: aluminio, violet-acrylic, blue-metallic-paint2 y red-plastic .



ESCENA	VÉRTICES	CARAS	BRDFS	MALLAS
Helicóptero	15199	24060	1	45
MUESTRAS	2500	T.MUESTREO	1179.44	secs.

Figura B.10: Aplicación del método *Adaptive Disc* a la generación de la escena de HELICÓPTERO.



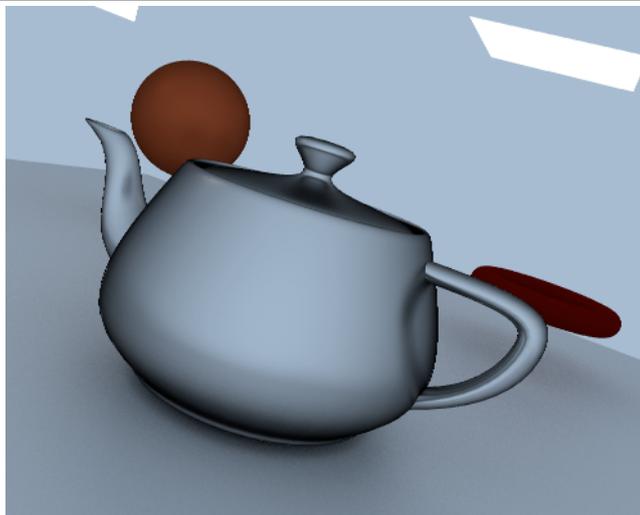
ESCENA	VÉRTICES	CARAS	BRDFS	MALLAS
Motorbike	20001	29948	7	12
MUESTRAS	400	T.MUESTREO	196.55	secs.

Figura B.11: Aplicación del método *Adaptive Disc* a la generación de la escena de MOTORBIKE.



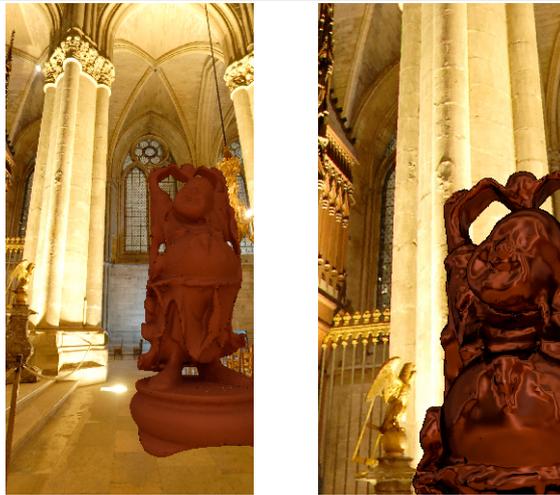
ESCENA	VÉRTICES	CARAS	BRDFS	MALLAS
Chevrolet	20512	33112	5	18
MUESTRAS	2500	T.MUESTREO	1197.48	secs.

Figura B.12: Aplicación del método *Adaptive Disc* a la generación de la escena de CHEVROLET.



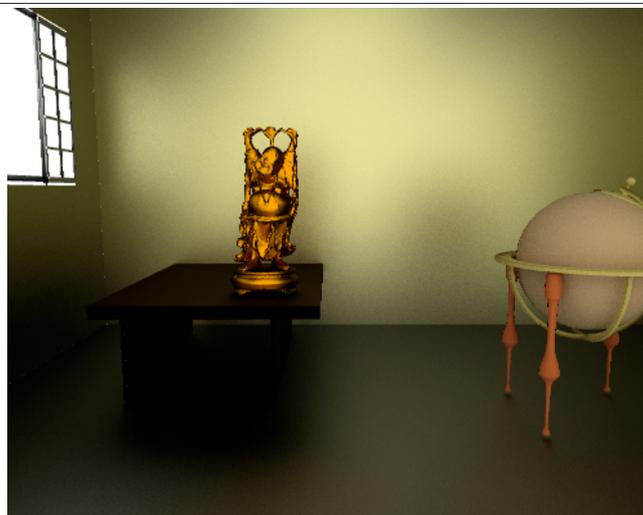
ESCENA	VÉRTICES	CARAS	BRDFS	MALLAS
PlaneScene	21917	4318	4	4
MUESTRAS	400	T.MUESTREO	198.59	secs.

Figura B.13: Aplicación del método *Adaptive Disc* a la generación de la escena de PLANESCENE.



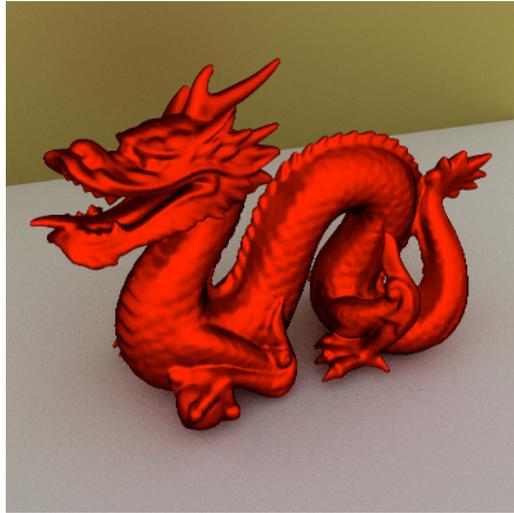
ESCENA	VÉRTICES	CARAS	BRDFS	MALLAS
Happy Budha	32328	67240	1	1
MUESTRAS	2500	T.MUESTREO	3114.98	secs.

Figura B.14: Escena HAPPY BUDHA para la BRDF de Poulin-Fournier y *resampling*.



ESCENA	VÉRTICES	CARAS	BRDFS	MALLAS
Living room 2	38233	78536	4	24
MUESTRAS	400	T.MUESTREO	332.19	secs.

Figura B.15: Aplicación del método *Adaptive Disc* a la generación de la escena de LIVING ROOM 2.



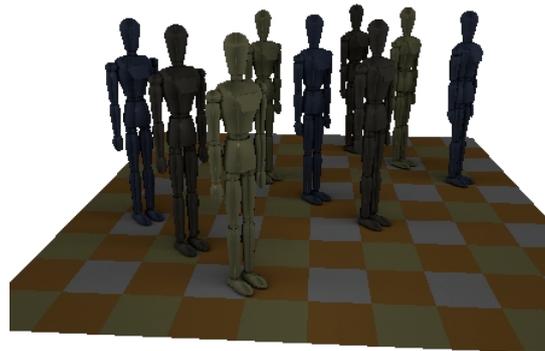
ESCENA	VÉRTICES	CARAS	BRDFS	MALLAS
Dragon	100258	202524	3	3
MUESTRAS	400	T.MUESTREO	174.975	secs.

Figura B.16: Aplicación del método *Adaptive Disc* a la generación de la escena de DRAGÓN.



ESCENA	VÉRTICES	CARAS	BRDFS	MALLAS
Roadway	63576	97903	11	221
MUESTRAS	2500	T.MUESTREO	249.494	secs.

Figura B.17: Aplicación del método *Adaptive Disc* a la escena de ROADWAY.



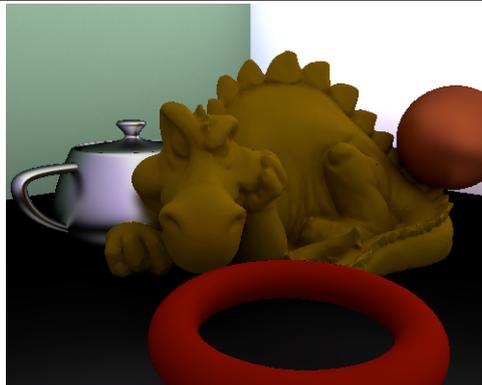
ESCENA	VÉRTICES	CARAS	BRDFS	MALLAS
Woodman	86521	145595	9	190
MUESTRAS	2500	T.MUESTREO	294.308	secs.

Figura B.18: Aplicación del método *Adaptive Disc* a la generación de la escena de WOODMAN.



ESCENA	VÉRTICES	CARAS	BRDFS	MALLAS
Phlegmatic dragon	2889278	480052	5	12
MUESTRAS	10000	T.MUESTREO	2646.73	secs.

Figura B.19: Aplicación del método *Adaptive Disc* a la escena PHLEGMATIC DRAGON.



ESCENA	VÉRTICES	CARAS	BRDFS	MALLAS
Phlegmatic dragon	311191	523234	8	15
MUESTRAS	10000	T.MUESTREO	8344.4	secs.

Figura B.20: PHLEGMATIC DRAGON con otros elementos de la escena PLANESCENE.



ESCENA	VÉRTICES	CARAS	BRDFS	MALLAS
Fertility	112681	212834	1	4
MUESTRAS	200	T.MUESTREO	22.298	secs.

Figura B.21: FERTILITY en EM de Notre-dame sin aplicar *re-sampling*.



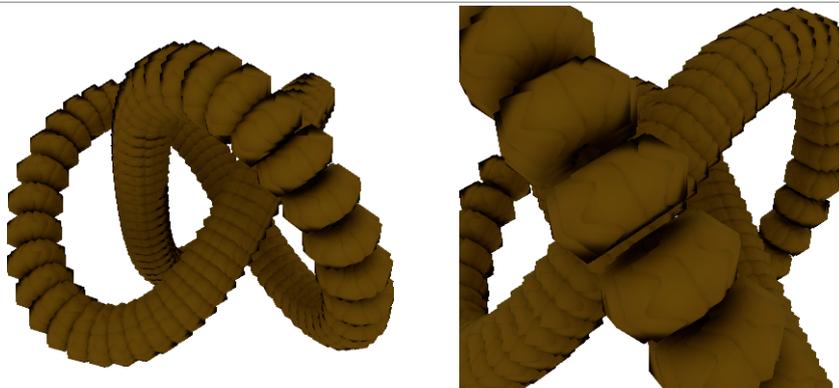
ESCENA	VÉRTICES	CARAS	BRDFS	MALLAS
Fertility	112681	212834	1	4
MUESTRAS	200	T.MUESTREO	224.101	secs.

Figura B.22: Aplicación combinada de *resampling* y *Adaptive Disc* a la escena FERTILITY.



ESCENA	VÉRTICES	CARAS	BRDFS	MALLAS
Bunny	35947	69451	3	3
MUESTRAS	400	T.MUESTREO	444.296	secs.

Figura B.23: Aplicación del método *Adaptive Disc* a la escena BUNNY.



ESCENA	VÉRTICES	CARAS	BRDFS	MALLAS
Pyramob	18876	31460	1	1
MUESTRAS	5000	T.MUESTREO	1311.3 / 1901.55	secs.

Figura B.24: BRDF adquirida *yellow-paint* muestreada con *Adaptive Disc* para la escena PYRAMOB.

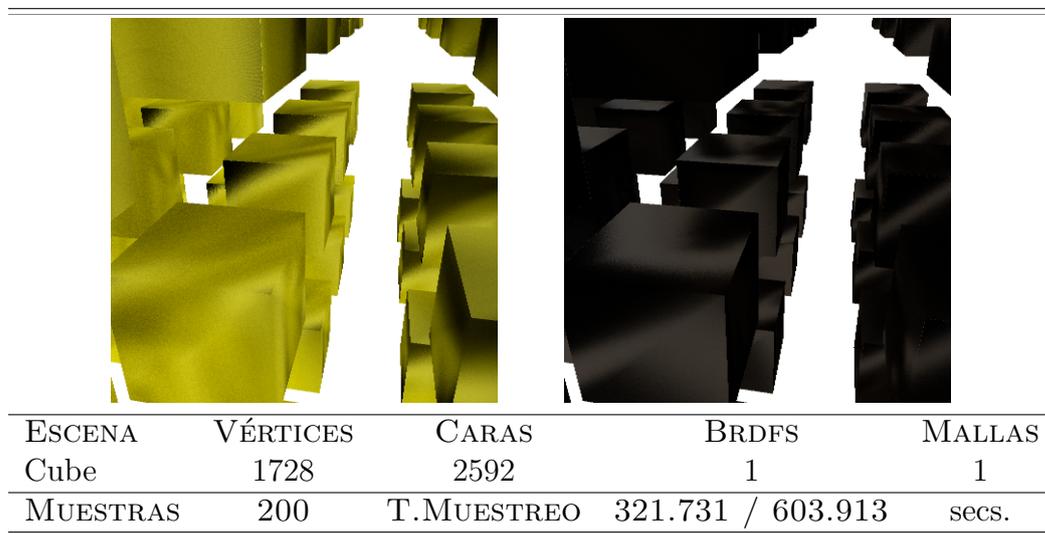


Figura B.25: A la izquierda escena usando la BRDF analítica de Phong. A la derecha la BRDF adquirida *nickel* muestreada con *Adaptive Disc* para la escena CUBE.

Referencias bibliográficas

- [AMS06] M. Mar Abad, Rosana Montes, and Paola Sebastiani.
Building chromosome-wide ld maps.
Bioinformatics Oxford Journals, (Online ISSN 1460-2059 - Print ISSN 1367-4803), 2006.
- [ARBJ03] Sameer Agarwal, Ravi Ramamoorthi, Serge Belongie, and Henrik W. Jensen.
Structured importance sampling of environment maps.
ACM Trans. Graph., 22(3):605–612, Julio 2003.
- [Arv86] James Arvo.
Backward ray tracing.
In *SIGGRAPH '86: Developments in Ray Tracing seminar notes*. ACM Press, 1986.
- [Arv95] James Arvo.
Applications of irradiance tensors to the simulation of non-lambertian phenomena.
In *SIGGRAPH '95: Proceedings of the 22nd annual Conference on Computer Graphics and Interactive Techniques*, pages 335–342. ACM Press, 1995.
- [AS00a] Michael Ashikhmin and Peter Shirley.
An anisotropic phong brdf model.
Journal on Graphics Tools, 5(2):25–32, 2000.
- [AS00b] Michael Ashikhmin and Peter Shirley.
A microfacet-based brdf generator.
In *SIGGRAPH 2000 Proceedings*, 2000.
- [BGH05] David Burke, Abhijeet Ghosh, and Wolfgang Heidrich.
Bidirectional importance sampling for direct illumination.
In *Eurographics Symposium on Rendering*, pages 147–156, Junio 2005.
- [Bli77] James F. Blinn.
Models of light reflection for computer synthesized pictures.
In *SIGGRAPH '77: Proceedings of the 4th annual conference on Computer graphics and interactive techniques*, pages 192–198, New York, NY, USA, 1977. ACM Press.

REFERENCIAS BIBLIOGRÁFICAS

- [BN76] James F. Blinn and Martin E. Newell.
Texture and reflection in computer generated images.
Commun. ACM, 19(10):542–547, 1976.
- [BS63] Peter Beckmann and Andre Spizzichino.
The scattering of electromagnetic waves from rough surfaces.
1963.
- [BSW00] Philippe Bekaert, Mateu Sbert, and Yves D. Willems.
Weighted importance sampling techniques for monte carlo radiosity.
In B. Peroche and H. Rushmeier, editors, *Rendering Techniques 2000 (Proceedings of the Eleventh Eurographics Workshop on Rendering)*, pages 35–46, New York, NY, 2000. Springer Wien.
- [Bue01] Juan M. Bueno.
Reversibilidad óptica en polarización: aplicación al ojo humano.
In *Revista Digital Puntex*, number 152 in 1, Enero 2001.
- [CETC06] David Cline, Parris K. Egbert, Justin F. Talbot, and David L. Cardon.
Two stage importance sampling for direct lighting.
In *In Rendering Techniques 2006 (Eurographics Symposium on Rendering)*, pages 103–113, Junio 2006.
- [CJAMJ05] Petrik Clarberg, Wojciech Jarosz, Tomas Akenine-Möller, and Henrik Wann Jensen.
Wavelet importance sampling: efficiently evaluating products of complex functions.
ACM Trans. Graph., 24(3):1166–1175, 2005.
- [CP85] F. Clarke and D. Parry.
Helmholtz reciprocity: Its validity and application to reflectometry.
Lighting Research and Technology, 1(17):1–11, 1985.
- [CPC84] Robert L. Cook, Thomas Porter, and Loren Carpenter.
Distributed ray tracing.
In *SIGGRAPH '84: Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, pages 137–145, New York, NY, USA, 1984. ACM Press.
- [CT82] R.L. Cook and K.E. Torrance.
A reflectance model for computer graphics.
1(1):7–24, 1982.
- [CWH93] Michael F. Cohen, John Wallace, and Pat Hanrahan.
Radiosity and Realistic Image Synthesis.
Academic Press Professional. San Diego, CA, 1993.
- [DGNK99] Kristin J. Dana, Bram Van Ginnekin, Shree K. Nayar, and Jan J. Koenderink.
Reflectances and texture of real-world surfaces.
In *ACM Transaction on Graphics*, volume 18, pages 1–34, 1999.
- [Dut] Philip Dutre.

- Global illumination compendium.
 Disponible en: <<http://www.cs.kuleuven.ac.be/~phil/GI>>
 [consultado el 6 de marzo de 2007].
- [FvDaSFH90] J. Foley, A. van Dam and S. Feiner, and J. Hughes.
Computer graphics: principles and practice (2nd ed.).
 Addison-Wesley. Massachusetts, 1990.
- [GA97] T.A. Germer and C.C. Asmail.
 A goniometric optical scatter instrument for bidirectional reflectance
 distribution function measurements with out-of-plane and polarime-
 try capabilities.
 In *Scattering and Surface Roughness. Proc. SPIE 3141*, pages 220–231.
 Z.-H. Gu and A.A. Maradudin, Editors, 1997.
- [GA99] T.A. Germer and C.C. Asmail.
 Goniometric optical scatter instrument for out-of-plane ellipsometry
 measurements.
Review of Scientific Instruments, 70:3688–3695, Septiembre 1999.
- [Gen03] James E. Gentle.
Random number generation and Monte Carlo methods. (2nd ed.).
 Springer, 2003.
- [GH03] Xavier Granier and Wolfgang Heidrich.
 A simple layered rgb brdf model.
Graph. Models, 65(4):171–184, 2003.
- [Gla94] A.S. Glassner.
Principles of Digital Image Synthesis.
 Morgan Kaufmann Publishers Inc. San Francisco, CA, 1994.
- [Gou71] H. Gouraud.
 Continuous shading for curved surfaces.
 In *IEEE Transactions on Computers*, number 6, pages 623–628, Junio
 1971.
- [GSCH93] Steven J. Gortler, Peter Schröder, Michael F. Cohen, and Pat Hanrahan.
 Wavelet radiosity.
Computer Graphics, 27(Annual Conference Series):221–230, 1993.
- [GTGB84] C. Goral, K. Torrance, D. Greenberg, and B. Battaile.
 Modelling the interaction of light between diffuse surfaces.
 In *ACM Computer Graphics*, pages 213–222, Julio 1984.
- [GUL⁺04] Rubén García, Carlos Ureña, Miguel Lastra, Rosana Montes, and Jorge
 Revelles.
 Interactive global illumination for quasi-static scenes.
 In *CGI'2004 Short Communications Proceedings*, Creta (Greece), 2004.
 Computer Graphics International.
- [GUL⁺05] Rubén García, Carlos Ureña, Miguel Lastra, Rosana Montes, and Jorge
 Revelles.

REFERENCIAS BIBLIOGRÁFICAS

- Optimizaciones en estimación de densidades para iluminación global.
In *Proceedings of the first Spanish Conference on Informatics (CEDI 2005), Congreso Español de Informática Gráfica*, pages 33–42. CEDI'05, Thomson, Septiembre 2005.
- [GUR⁺06] Rubén García, Carlos Ureña, Jorge Revelles, Miguel Lastra, and Rosana Montes.
Density estimation optimizations for global illumination.
In *WSCG'2006 Short Communications Proceedings*, 2006.
- [Hel25] H.V. Helmholtz.
Treatise on physiological optics.
1(17), 1925.
- [HHP⁺92] Xiao D. He, Patrick O. Heynen, Richard L. Phillips, Kenneth E. Torrance, David H. Salesin, and Donald P. Greenberg.
A fast and accurate light reflection model.
SIGGRAPH Computer Graphics, 26(2):253–254, 1992.
- [HK93] P. Hanrahan and W. Krueger.
Reflection from layered surfaces due to subsurface scattering.
In *SIGGRAPH '93: Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, number 3, pages 165–174, New York, NY, USA, 1993. ACM Press.
- [HKYM01] H. Hirayama, K. Kaneda, H. Yamashita, and Y. Monden.
An accurate illumination model for objects coated with multilayer films.
Computers and Graphics, 25:391–/400(10), Junio 2001.
- [HTSG91] X.D. He, K.E. Torrance, F.X. Sillion, and D.P. Greenberg.
A comprehensive physical model for light reflection.
In *SIGGRAPH '91: Proceedings of the 18th annual conference on Computer graphics and interactive techniques*, number 4, pages 175–186, New York, NY, USA, Julio 1991. ACM Press.
- [JC95] Henrik W. Jensen and N. Christensen.
Photon maps in bidirectional monte carlo ray tracing for complex objects.
In *Computer & Graphics*, number 2, pages 215–224, 1995.
- [Jen01] Henrik W. Jensen.
Realistic Image Synthesis Using Photon Mapping.
AK Peters, 2001.
- [JMLH01] Henrik W. Jensen, Stephen R. Marschner, Marc Levoy, and Pat Hanrahan.
A practical model for subsurface light transport.
In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 511–518, New York, NY, USA, 2001. ACM Press.
- [JTE05] David Cline Justin Talbot and Parris K. Egbert.

- Importance resampling for global illumination.
 In *Rendering Techniques 2005 (Eurographics Symposium on Rendering)*,
 pages 139–146, Junio 2005.
- [Kaj85] James T. Kajiya.
 Anisotropic reflectance models.
 In *Computers Graphics, ACM Siggraph'85 Conference Proceedings*,
 number 4, pages 15–21, Julio 1985.
- [Kaj86] James T. Kajiya.
 The rendering equation.
 In *SIGGRAPH '86: Proceedings of the 13th annual conference on Com-
 puter graphics and interactive techniques*, pages 143–150, New York,
 NY, USA, 1986. ACM Press.
- [Kel97] Alexander Keller.
 Instant radiosity.
 In *SIGGRAPH '97: Proceedings of the 24th annual conference on Com-
 puter graphics and interactive techniques*, pages 49–56, New York,
 NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.
- [KM99] Jan Kautz and Michael McCool.
 Hardware rendering with bidirectional reflectances.
 Technical report, CS-99-02. University of Waterloo, 1999.
- [Las00] Miguel Lastra.
 Síntesis eficiente de imágenes fotorealistas por simulación del modelo de
 partículas de la luz.
 Technical report, Dpto. Lenguajes y Sistemas Informáticos. Universidad
 de Granada, 2000.
- [Las04] Miguel Lastra.
Stochastic Rendering Techniques for Complex Environments.
 PhD thesis, Universidad de Granada, 2004.
- [Len03] Hendrik P. A. Lensch.
Efficient Image-Based Appearance Acquisition of Real-World Objects.
 PhD thesis, Max-Planck-Institut für Informatik. Saarbrücken. Germany,
 Diciembre 2003.
- [Lew93] Robert R. Lewis.
 Making shaders more physically plausible.
 In *Fourth Eurographics Workshop on Rendering, Eurographics'93*, pages
 47–62, Vancouver, BC, Canada, Canada, Junio 1993. University of
 British Columbia.
- [LFTG97] Eric P. Lafortune, Sing-Choong Foo, Kenneth E. Torrance, and Do-
 nald P. Greenberg.
 Non-linear approximation to reflectance functions.
 In *SIGGRAPH '97: Proceedings of the 24th annual conference on Com-
 puter graphics and interactive techniques*, pages 117–126, New York,
 NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.

REFERENCIAS BIBLIOGRÁFICAS

- [LRR04] Jason Lawrence, Szymon Rusinkiewicz, and Ravi Ramamoorthi. Efficient importance sampling using a factored representation. In *ACM Transaction of Graphics. Siggraph 2004*, number 3, pages 496–505, 2004.
- [LRR05] Jason Lawrence, Szymon Rusinkiewicz, and Ravi Ramamoorthi. Adaptive numerical cumulative distribution functions for efficient importance sampling. In *Eurographics Symposium on Rendering*, 2005.
- [LURM02a] Miguel Lastra, Carlos Ureña, Jorge Revelles, and Rosana Montes. Density estimation on the tangent plane for radiosity. In *1er Simposium Iberoamericano de Informatica Grafica (SIACG'2002)*, Guimaraes (Portugal), Julio 2002.
- [LURM02b] Miguel Lastra, Carlos Ureña, Jorge Revelles, and Rosana Montes. A particle-path based method for monte carlo density estimation. In *Poster en 13er Eurographics Workshop on Rendering*, Pisa (Italia), Junio 2002.
- [LW93] Eric P. Lafortune and Yves D. Willems. Bi-directional path tracing. In *Proceedings of Computer & Graphics, Alvor, Portugal*, pages 145–153, Diciembre 1993.
- [LW94] Eric P. Lafortune and Yves D. Willems. Using the modified phong reflectance model for physically based rendering. Technical Report Report CW197, Department of Computer Science, K.U.Leuven, Leuven, Belgium, Noviembre 1994. Disponible en: <<http://www.cs.kuleuven.ac.be/cwis/research/graphics/ERICL/Phong.html>>[consultado el 6 de marzo de 2007].
- [MAA01] Michael D. McCool, Jason Ang, and Anis Ahmad. Homomorphic factorization of BRDFs for high-performance rendering. In Eugene Fiume, editor, *SIGGRAPH 2001, Computer Graphics Proceedings*, pages 185–194. ACM Press / ACM SIGGRAPH, 2001.
- [MBWL73] J. R. Maxwell, J. Beard, S. Weiner, and D. Ladd. Bidirectional reflectance model validation and utilization. Technical report, AFAL-TR-73-303, Environmental Research Institute of Michigan (ERIM), Octubre 1973.
- [MH97] Michael D. McCool and Peter K. Harwood. Probability trees. In Wayne A. Davis, Marilyn Mantei, and R. Victor Klassen, editors, *Graphics Interface '97*, pages 37–46. Canadian Human-Computer Communications Society, 1997.
- [Mil84] Hoffman Miller. Illumination and reflection maps: Simulated objects in simulated and real environments.

- In *Course Notes from Advances Computer Graphics Animation SIG-GRAPH 84*. ACM Press, Julio 1984.
- [Min41] M. Minnaert.
The reciprocity principle in lunar photometry.
Astrophysical Journal, pages 403–410, 1941.
- [MM04] Rosana Montes and Francisco J. Melero.
A multimedia 3d game for museums.
In Nicolucci, editor, *Beyond the artifact: Computer Applications in Archaeology*, Oxford, Abril 2004. Archaeopress.
- [MPBM03] Wojciech Matusik, Hanspeter Pfister, Matt Brand, and Leonard McMillan.
A data-driven reflectance model.
ACM Trans. Graph., 22(3):759–769, 2003.
- [MRR+53] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller.
Equation of state calculations by fast computing machines.
Journal of Chemical Physics, (1087), 1953.
- [MULG08] Rosana Montes, Carlos Ureña, Miguel Lastra, and Rubén García.
Generic brdf sampling: A sampling method for global illumination.
In *GRAPP 2008 Proceedings of the Third International Conference on Computer Graphics Theory and Applications*, pages 191–198, Funchal, Madeira, Portugal, Enero 2008.
- [MUR+02] Rosana Montes, Carlos Ureña, Jorge Revelles, Miguel Lastra, and Rubén García.
Wannabe amazing, una herramienta de trabajo para iluminación global.
In *Jornadas Regionales de Informática Gráfica*, pages 147–158, Jaén, España, Febrero 2002.
- [MWLT00] Stephen R. Marschner, Sephen H. Westin, Eric P. Lafortune, and Kenneth E. Torrance.
Image-based bidirectional reflectance distribution function measurement.
Applied Optics, Optical Society of America, (16):2592–2600, Junio 2000.
- [NN96] Laszlo Neumann and Attila Neumann.
A new class of brdf models with fast importance sampling.
Technical Report TR-186-2-96-24, Diciembre 1996.
- [NN98] Lazlo Neumann and Attila Neumann.
New simple reflectance models for metals and other specular materials.
Technical report, Institute of Computer Graphics, Vienna University of Technology, 1998.
- [NNB97] Lazlo Neumann, Attila Neumann, and Philippe Beakaert.
Radiosity with well distributes ray sets.

REFERENCIAS BIBLIOGRÁFICAS

- In *8th Eurographics Workshop on Rendering, Eurographics'97*, number 3, 1997.
- [NNSK99] Lazlo Neumann, Attila Neumann, and Lazlo Szirmay-Kalos.
Compact metallic reflectance models.
In P. Brunet and R. Scopigno, editors, *Computer Graphics Forum (Eurographics '99)*, volume 18(3), pages 161–172. The Eurographics Association and Blackwell Publishers, 1999.
- [ON94] M. Oren and S.K. Nayar.
Generalization of lambert's reflectance model.
In *SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 239–246, New York, NY, USA, 1994. ACM Press.
- [ON95] M. Oren and S.K. Nayar.
Visual appearance of matte surfaces.
Science, 267(5201):1153–1156, Febrero 1995.
- [PF90] Pierre Poulin and Alain Fournier.
A model for anisotropic reflection.
In *SIGGRAPH '90: Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, number 4, pages 273–282, New York, NY, USA, Agosto 1990. ACM Press.
- [PFTV92] W.H. Press, P.B. Flannery, S.A. Teukolsky, and W.T. Vetterling.
Numerical Recipes in C: The Art of Scientific Computing.
Cambridge University Press, 1992.
- [Pho75] Bui-Tuong Phong.
Illumination for computer generated pictures.
In *Computers Graphics, ACM Siggraph'75 Conference Proceedings*, number 6, pages 311–317, New York, NY, USA, Junio 1975. ACM Press.
- [Ros97] Paul L. Rosin.
Techniques for assessing polygonal approximations of curves.
In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, number 6, pages 659–666, 1997.
- [Rus98] S.M. Rusinkiewicz.
A new change of variables for efficient brdf representation.
In *Ninth Eurographics Workshop on Rendering*, pages 11–23, Junio 1998.
- [SAWG91] François Sillion, James Arvo, Stephen Westin, and Donald P. Greenberg.
A global illumination solution for general reflectance distributions.
In *SIGGRAPH'91 conference proceedings*, pages 187–196. ACM Press, 1991.
- [Sbe96] Mateu Sbert.

- The use of global random directions to compute radiosity — Global Monte Carlo techniques.*
 PhD thesis, Universitat Politecnica de Catalunya, Barcelona, España, Noviembre 1996.
- [SBG95] Peter Shirley, Wade Bretton, and Donald Greenberg.
 Global illumination via density-estimation radiosity.
 In *Eurographics Rendering Workshop*, 1995.
- [Sch93] Christophe Schlick.
 A customizable reflectance model for everyday rendering.
 In *Fourth Eurographics Workshop on Rendering, Eurographics'93*, pages 73–84, Junio 1993.
- [Sch94a] Christophe Schlick.
 A fast alternative to phong's specular model.
 In *Graphics Gems*, pages 363–366. Eds Paul Heckbert, Academic Press, 1994.
- [Sch94b] Christophe Schlick.
 An inexpensive brdf model for physically-based rendering.
 In *Proc. Eurographics'94, Computer Graphics Forum*, number 3, pages 233–246, 1994.
- [Shi96] Peter Shirley.
 Monte carlo methods for rendering.
 In *ACM SIGGRAPH '96 Course Notes - Global Illumination in Architecture and Entertainment*, pages 1–26. 1996.
- [SHSL97] Peter Shirley, Helen Hu, Brian Smits, and Eric Lafortune.
 A practitioners' assessment of light reflection models.
 In *Pacific Graphics 97 conference proceedings*, pages 40–49, 1997.
- [Sta99] Jos Stam.
 Diffraction shaders.
 In *Proc. Siggraph'99*, 1999.
- [Sta01] Jos Stam.
 An illumination model for a skin layer bounded by rough surfaces.
 In *Proceedings of the 12th Eurographics Workshop on Rendering Techniques*, pages 39–52, London, UK, 2001. Springer-Verlag.
- [Str90] Paul S. Strauss.
 A realistic lighting model for computer animators.
IEEE Computer Graphics Applications, 10(6):56–64, 1990.
- [Tan96] Martin A. Tanner.
Tools for Statistical Inference: Methods for the Exploration of Posterior Distributions and Likelihood Functions.
 Springer Series in Statistics, 1996.
- [TS66] K.E. Torrance and E.M. Sparrow.

REFERENCIAS BIBLIOGRÁFICAS

- Off-specular peaks in the directional distribution of reflected thermal radiation.
In *Journal of Heat Transfer*, pages 223–230, Mayo 1966.
- [TS67] K.E. Torrance and E.M. Sparrow.
Theory for off-specular reflection.
In *Journal of the Optical Society of America*, number 9, Septiembre 1967.
- [UPTdS92] Carlos Ureña, Jose Parets, Juan C. Torres, and Vicente del Sol.
An object-oriented approach to ray tracing image synthesis implementations.
Computers & Graphics, (4):363–368, 1992.
- [Ure98] Carlos Ureña.
Metodos de Monte-Carlo Eficientes para Iluminacion Global.
PhD thesis, Universidad de Granada, 1998.
- [UT93] Carlos Ureña and Juan C. Torres.
A bidirectional monte carlo method for image rendering.
Technical report, LSI-1993-4. Universidad de Granada, 1993.
- [UTR⁺97] Carlos Ureña, Juan C. Torres, Jorge Revelles, Pedro Cano, Vicente del Sol, and Marcelino Cabrera.
Girt: Un sistema orientado a objetos para simulacion precisa de la iluminacion.
Technical report, LSI-1997-1. Universidad de Granada, 1997.
- [Vea97] Eric Veach.
Robust Monte Carlo methods for light transport simulation.
PhD thesis, Stanford University, 1997.
- [VG94] Eric Veach and L. Guibas.
Bidirectional estimators for light transport.
In *Proceedings of the Fifth Eurographics Workshop on Rendering*, pages 147–162, Junio 1994.
- [Wal05] Bruce Walter.
Notes on the ward brdf.
Technical report, Cornell Program of Computer Graphics, Abril 2005.
- [War92] Gregory J. Ward.
Measuring and modelling anisotropic reflection.
In *Computers Graphics, ACM Siggraph'92 Conference Proceedings*, number 4, pages 265–272, Julio 1992.
- [WAT92] Stephen H. Westin, James R. Arvo, and Kenneth E. Torrance.
Predicting reflectance functions from complex surfaces.
In *SIGGRAPH '92: Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, number 2, pages 255–264, New York, NY, USA, Julio 1992. ACM Press.
- [WCS01] Henrik W.Jensen, Per H. Christensen, and Frank Suykens.

- A practical guide to global illumination using photon mapping, siggraph course notes, 2001.
- [WH92] Gregory J. Ward and Paul Heckbert.
Irradiance Gradients.
In *Third Eurographics Workshop on Rendering*, pages 85–98, Bristol, UK, 1992.
- [Whi80] Turner Whitted.
An improved illumination model for shaded display.
In *Computers Graphics, ACM Siggraph'80 Conference Proceedings*, number 6, pages 343–349, New York, NY, USA, Junio 1980. ACM Press.
- [WM02] Harold B. Westlund and Gary W. Meyer.
A brdf database employing the beard-maxwell reflection model.
In *Graphics Interface*, 2002.
- [WNLH06] Rui Wang, Ren Ng, David Luebke, and Greg Humphreys.
Efficient wavelet rotation for environment map rendering.
In *Eurographics Symposium on Rendering*, 2006.
- [WRC88] G. Ward, F. Rubinstein, and R. Clear.
A ray tracing solution for diffuse interreflection.
In *Computers Graphics, ACM Siggraph'88 Conference Proceedings*, number 4, Agosto 1988.