

# Un algoritmo de muestreo exacto para BRDFs arbitrarias

R. Montes, C. Ureña, M. Lastra y R. García

E.T.S. Ingenierías Informática y de Telecomunicación  
Universidad de Granada. España  
{rosana,curena,mلاstral,ruben}@ugr.es

---

## Resumen

*Este trabajo presenta un algoritmo para el muestreo eficiente y exacto de BRDF genéricas, esto es, apto para cualquiera de los modelos de BRDFs analíticos o adquiridos de la literatura de informática gráfica. Nuestro objetivo principal consiste en proporcionar una función de probabilidad utilizable en algoritmos de iluminación global basados en métodos de Monte-Carlo, que permita un muestreo por importancias proporcional al producto de la función BRDF y un término coseno. Mediante la subdivisión adaptativa de la BRDF en el disco unidad, obtenemos una estructura jerárquica o quadtree que nos permite aplicar un muestreo por rechazo optimizado en los nodos. El número medio de intentos en el muestreo está acotado y es parámetro de la estructura utilizada. El método se aplica al muestreo de cualquier modelo de BRDF sin necesidad de guía por parte del usuario.*

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism. Color, shading, shadowing, and texture

---

## 1. Introducción

En las técnicas de iluminación intervienen dos elementos básicos: las fuentes productoras de luz y los objetos que se van a iluminar. 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 objetivo de los algoritmos basados en Monte-Carlo [CPC84, Kaj86, LW93, SBG95, JC95] 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* o BRDF. 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 tema de interés en los algoritmos basados en Monte-Carlo.

En el contexto de un estimador de Monte Carlo de la radiancia saliente, el punto de la superficie sobre el que se efec-

túa 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 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 principal inconveniente de Monte-Carlo es el error intrínseco en la solución. Debido a esto, las investigaciones se centran en desarrollar técnicas que reduzcan el error, minimizando el número de muestras necesarias. Para disminuir la varianza presentamos un método de muestreo para BRDF arbitrarias que consigue menos error, para un mismo número de muestras, que otros métodos de muestreo.

## 2. La ecuación de reflectancia

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. La función  $f_r$  representa a la BRDF. La integral de reflectancia (ecuación 1), 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.

$$L_r(\mathbf{u}) \stackrel{\text{def}}{=} \int_{\Omega} f_r(\mathbf{u}, \mathbf{v}) L_i(\mathbf{v}) (\mathbf{v} \cdot \mathbf{n}) d\sigma(\mathbf{v}) \quad (1)$$

### 2.1. Integración por Monte-Carlo

Considere una integral de la forma:

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

donde  $D$  es el dominio de integración y  $\mu$  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:

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

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 *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*:

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

### 2.2. Estimación numérica de $L_r$

La integral vista en la ecuación (1) 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  $\Omega$ , dado un vector  $\mathbf{u} \in \Omega$ . Es condición necesaria que  $P_{\mathbf{u}}(\Omega) = 1$ .

Los vectores de dirección normalizados  $\mathbf{v} = (x_v, y_v, z_v)$  en  $\Omega$  pueden expresarse equivalentemente en el dominio del disco unidad  $\mathcal{D}^2$ . Esta parametrización 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  $z_v$ , lo cual nos deja  $\mathbf{v}_{xy} \in \mathcal{D}^2$ .

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

$$h(\mathbf{v}) = h(x_v, y_v, z_v) \stackrel{\text{def}}{=} (x_v, y_v) = \mathbf{v}_{xy}$$

El dominio  $\mathcal{D}^2$  lleva implícito la utilización de la medida del ángulo sólido proyectado, siendo  $d\sigma_p(\mathbf{v}) \stackrel{\text{def}}{=} (\mathbf{v} \cdot \mathbf{n}) d\sigma(\mathbf{v})$ . 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. Esto significa que regiones diferenciales en  $\Omega$  pueden proyectarse en regiones diferenciales en  $\mathcal{D}^2$ , e igualmente a la inversa:

$$dx dy = dA(x_u, y_u) = d\sigma_p(h^{-1}(x_u, y_u)) = (\mathbf{u} \cdot \mathbf{n}) d\sigma(\mathbf{u})$$

Podemos definir distintos estimadores de Monte-Carlo dependiendo de que la integración en el dominio  $\Omega$  se realice con la medida del ángulo sólido ( $\sigma$ ) o la versión proyectada de esta ( $\sigma_p$ ). En el segundo caso, la PDF:

$$q_{\mathbf{u}}(\mathbf{v}) \stackrel{\text{def}}{=} \frac{dP_{\mathbf{u}}(\mathbf{v})}{d\sigma_p(\mathbf{v})},$$

proporcionaría este estimador:

$$L_r(\mathbf{u}) \approx X_N(\mathbf{s}_1, \dots, \mathbf{s}_N) \stackrel{\text{def}}{=} \frac{1}{N} \sum_{i=1}^N \frac{f_r(\mathbf{u}, \mathbf{s}_i) L_i(\mathbf{s}_i)}{q_{\mathbf{u}}(\mathbf{s}_i)}. \quad (2)$$

Por lo general, la función de densidad basada en  $\sigma$  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.

### 2.3. Trabajos previos

En la actualidad, se utilizan técnicas que han conseguido aumentar la eficiencia de los estimadores de Monte Carlo [Vea97, BSW00, BGH05, 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 un algoritmo de muestreo que aumenta 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.

La forma más simple de generar una nueva dirección ocurre cuando se consideran únicamente eventos de reflexión ideales. Sin embargo el caso más general requiere de alguna estrategia para el muestreo eficiente de los picos presentes en una BRDF especular.

**Métodos basados en el lóbulo coseno.** Existen distintas 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 [Pho75], Blinn [Bli77] —y sus respectivas versiones normalizadas dadas por Lewis [Lew93], Ward [War94] y Lafortune [LW94]. Para estas BRDFs se obtiene de su normalización, la PDF de interés:

$$p_{\mathbf{u}}(\mathbf{v}) = \frac{1}{N_1(\mathbf{u}, n)} (\mathbf{v} \cdot \mathbf{u}_r)^n$$

La normalización se consigue mediante la función  $N_1$  de la cual existen expresiones analíticas [Arv95]

$$N_1(\mathbf{a}, n) \stackrel{\text{def}}{=} \int_{\Omega} (\mathbf{v} \cdot \mathbf{a})^n d\sigma(\mathbf{v})$$

Una variante de esta PDF evita evaluar  $N_1$  al tomar direcciones aleatorias en la esfera de direcciones  $\mathcal{S}^2$ , en lugar de la semiesfera positiva  $\Omega$ . Al considerar la parte del lóbulo que se encuentra bajo la superficie, se hace a  $N_1(\mathbf{u}, n)$  independiente de  $\mathbf{u}$  e igual a  $N_1(\mathbf{n}, n) = 2\pi/(n+1)$ . Por contra, este algoritmo origina valores altos de varianza cuando  $\mathbf{u}$  se aproxima a ángulos cercanos a 90 grados.

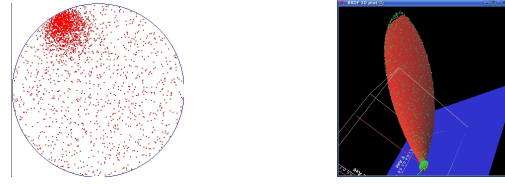
**La representación factorizada de la BRDF.** Lawrence y Rusinkiewicz [LRR04] presentan un método analítico de muestreo aplicable a un rango mayor de BRDFs. Su trabajo se basa en la descomposición en factores de la BRDF, que son almacenados tabularmente y de manera compacta. Tras un cambio a la parametrización basada en el *half-angle* [Rus98] la descomposición de la BRDF resulta en factores  $F_j$  y  $G_j$  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. Una de las funciones 2D,  $G$ , no puede ser evaluada directamente, y vuelve a descomponerse en funciones 1D  $u_{jk}$  y  $v_{jk}$  mucho más sencillas. De esta forma la BRDF se representa de forma aproximada como:

$$f_r(\mathbf{u}, \mathbf{v}) \cos(\mathbf{v}) \approx \sum_{j=1}^J F_j(\mathbf{u}) \sum_{k=1}^K u_{jk}(\theta_{\mathbf{w}}) v_{jk}(\phi_{\mathbf{w}}).$$

La generación de vectores aleatorios se obtiene a partir de los factores 1D precalculados y el recálculo de los factores 2D para una dirección dada  $\mathbf{u}$ . Los factores representan funciones de densidad de probabilidad (o PDF) que pueden usarse para calcular la función de distribución acumulada (o CDF) que se emplea en el muestreo por inversión.

**Muestreo compacto de BRDFs adquiridas.** El muestreo por inversión de la CDF es optimizado en [LRR05], y se aplica al muestreo de BRDFs adquiridas [MPBM03]. Los valores considerados en la CDF no se distribuyen uniformemente, sino de forma adaptativa. Se sigue la misma distribución original, haciendo uso de otra distribución equivalente que viene expresada de manera compacta. Primeramente se descompone la PDF  $N$ -dimensional en una distribución marginal 1D  $\tilde{p}$ , junto a una colección de distribuciones condicionales también en 1D.

$$\tilde{p}(x) = \int_{-\infty}^{\infty} p(x, y) dy$$



**Figura 1:** Izquierda: la distribución de direcciones dada por nuestro método cubre la parte difusa y especular de la función. Derecha: los puntos en 3D se adaptan a la gráfica de la BRDF.

$$p(y|x_i) = \frac{1}{x_i - x_{i-1}} \int_{x_{i-1}}^{x_i} \frac{p(x', y)}{\tilde{p}(x')} dx'$$

Seguidamente, se realiza el cálculo discretizado de las CDFs marginal y condicional. 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. El algoritmo de *Douglas-Peucker* [Ros97], dará las localizaciones de las muestras adaptativas que representan cada curva CDF.

### 3. Un algoritmo de muestreo exacto

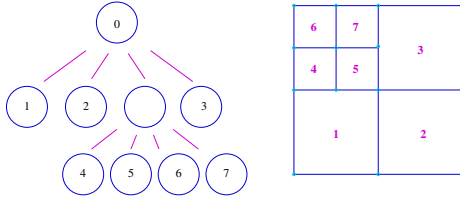
Considérese la ecuación de la radiancia (1) y el estimador de Monte Carlo (2). Proponemos un algoritmo de muestreo que proporciona direcciones aleatorias en zonas donde la BRDF por el término coseno presentan valores altos, consiguiendo de esta forma, aplicar muestreo por importancias o *importance sampling*. La distribución de vectores obtenida se ajusta a la BRDF y permite un muestreo exacto de las componentes difusa y especular que compongan la función, tal y como se ve en la figura 1.

La utilización de la medida de área  $A$  en  $\mathcal{D}^2$  es mejor que el uso del ángulo sólido  $\sigma$  en  $\Omega$  porque de esta forma se hace innecesario incluir el coseno de la formulación en los cálculos. Se consigue un estimador más simple y rápido. Además nuestro algoritmo es independiente de la BRDF en uso y no requiere la intervención del usuario para ajustar ningún parámetro dependiendo del modelo de reflectancia utilizado.

#### 3.1. Subdivisión adaptativa del dominio en base a la BRDF

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  $\mathcal{D}^2$ , lo que puede ser realizado mediante una estructura de datos tipo *quadtree*, que subdivide siempre en cuatro partes por igual. Puesto que tenemos una partición, se deben cumplir las siguientes propiedades:

$$R_i \cap R_j = \emptyset \quad \forall i \neq j, \\ \cup_{i=1}^n R_i = \mathcal{D}^2.$$



**Figura 2:** División espacial del dominio  $[-1, 1]^2$

Cada una de estas regiones corresponde a un conjunto de puntos en  $\mathcal{D}^2$ , que, al ser proyectados de nuevo en  $\Omega$ , 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  $[-1, 1]^2 - \mathcal{D}^2$ , garantizando la equivalencia entre  $\Omega$  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:

$$f_r(\mathbf{u}, \mathbf{v}) > 0 \implies P_{\mathbf{u}}(R) > 0 \wedge q_{\mathbf{u}}(\mathbf{v}) > 0.$$

Así pues  $P_{\mathbf{u}}(\mathcal{D}^2) = 1$ .

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 2. 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).

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 de la intersección del nodo con el disco unitario. Se representa 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.
- $M_i$ : 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:

$$I_i \stackrel{\text{def}}{=} \int_{x,y \in R_i} f_r(\mathbf{u}, \mathbf{v}_{xy}) dx dy$$

- 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:

$$V_i \stackrel{\text{def}}{=} \int_{x,y \in R_i} \bar{f}_r(\mathbf{u}, \mathbf{v}_{xy}) dx dy = size_i^2 \cdot M_i$$

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 re-

querimiento impuesto por nuestro algoritmo es ser capaces de evaluar la BRDF dados un par de direcciones.

Con estos valores, podemos conocer la probabilidad de seleccionar la región proyectada  $R_i$ , ya que es proporcional a la integral de la BRDF en la región,  $P_{\mathbf{u}}(R_i) \propto I_i$ :

$$P_{\mathbf{u}}(R_i) \stackrel{\text{def}}{=} \frac{1}{\int_{\mathcal{D}^2} f_r(\mathbf{u}, \mathbf{v}_{xy}) dx dy} \cdot I_i$$

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. El valor máximo permitido para el número medio de intentos al realizar *rejection sampling* en una región  $R_i$  es  $n_{max}$ . Este parámetro interviene en el de subdivisión, junto con la probabilidad de aceptación en el muestreo por rechazo. Al cumplirse la siguiente desigualdad se debe subdividir:

$$n_{max} \cdot \frac{I_i}{V_i} \leq 1. \quad (3)$$

### 3.2. 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:

$$\text{si } F_{i,k-1} < r \leq F_{i,k} \text{ se selecciona el nodo } k$$

Finalmente, con este paso nos quedamos con una región, 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.

### 3.3. 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*. 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 es similar al necesario para muestrear en un disco inscrito en un cuadrado unidad, tan sólo es necesario cambiar los límites por los pertenecientes a la región previamente seleccionada tal y como indica el algoritmo 1. Una vez seleccionado un punto, se realiza la transformación inversa pasando de una dirección en  $\mathcal{D}^2$  a una en  $\mathcal{S}^2$ .

---

**Algoritmo 1** pseudocódigo RECHAZOREGIÓN
 

---

**RECHAZOREGIÓN**( )  $\rightarrow [0,1]^2$

*do begin*

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

Generar  $x = u_i + \text{size} * \xi_1$

Generar  $y = v_i + \text{size} * \xi_2$

Generar  $z$  uniforme en  $[0, M_i)$

Calcular  $f_r(\mathbf{u}, \mathbf{s}_{xy})$

*while*  $f_r \leq z$

**return** (x,y)

---

Almacenamos junto a cada dirección generada, el valor de la función peso asociado a cada muestra  $\mathbf{s}$ :

$$w_{\mathbf{u}}(\mathbf{s}) = \begin{cases} (\mathbf{s} \cdot \mathbf{n}) / p_{\mathbf{u}}(\mathbf{s}) & \text{para la PDF tipo } p_{\mathbf{u}}, \\ 1 / q_{\mathbf{u}}(\mathbf{s}) & \text{para una PDF tipo } q_{\mathbf{u}}. \end{cases}$$

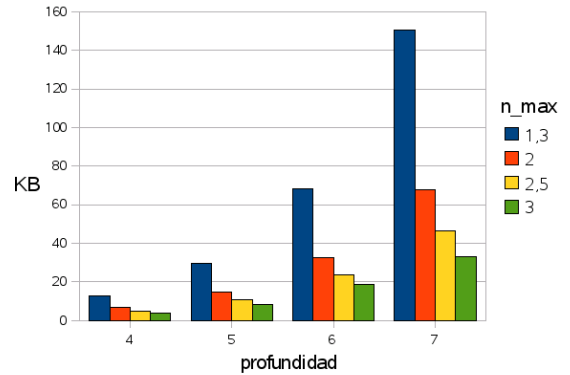
Con esto, nuestro método de muestreo de la BRDF se integra fácilmente en el estimador  $X_N$ .

### 3.4. 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  $\theta$  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. Para BRDFs anisótropas el precálculo se debe realizar en base a valores discretos de los ángulos  $\theta$  y  $\phi$ .

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

### Consumo medio de memoria



**Figura 3:** Necesidades de memoria en KB de almacenamiento para un *quadtree*, considerando variaciones en el parámetro real  $n_{max}$  y en la profundidad máxima

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.

### 3.5. 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 varianza. Sin embargo otros aspectos igualmente importantes deben ser considerados: la velocidad de muestreo, la optimización de recursos como la memoria y por supuesto la facilidad de implementación para aplicarlo en otros entornos.

El parámetro que afecta a la profundidad de una estructura jerárquica, y por consiguiente, al número de nodos que se almacenan, corresponde al parámetro  $n_{max}$  utilizado en el criterio de subdivisión del árbol (ver  $n_{max}$  en la desigualdad de la ecuación 3). Si variamos el valor promedio de intentos en el método de muestreo por rechazo tomando valores desde 1,3 hasta 3 obtenemos los datos que se muestran en la figura 3. Lo hacemos calculando en promedio para cada BRDF las necesidades de memoria de un *quadtree* cuando fijamos una profundidad máxima. En las pruebas realizadas, hemos encontrado que los valores más eficientes se obtenían para  $n_{max} = 2$  y  $depth = 5$ . De esta forma se establecen como predeterminados en el algoritmo. El usuario se encuentra con un método de pocos parámetros y de significado simple.

### 3.6. Generación múltiple de direcciones

Una mejora clara a nuestro procedimiento consiste en obtener  $N$  valores aleatorios en una única llamada, en lugar de recorrer el *quadtree*  $N$  veces para generar  $N$  direcciones



aleatorias. 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.

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  $n_1 + n_2 + \dots + n_M = N$ .

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:

$$n_1 + n_2 + \dots + n_M \leq N,$$

$$\delta = N - \sum_{i=1}^M \lfloor P_{\mathbf{u}}(l_i) * N \rfloor.$$

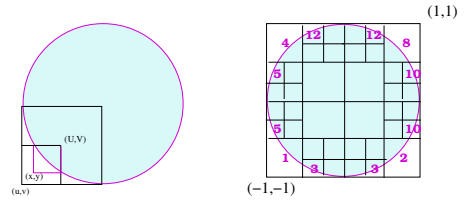
De lo que se deduce que  $\delta$  es el número entero de muestras —el resto— que es necesario volver a repartir. Considerando de nuevo las probabilidades de cada nodo hoja y mediante un proceso iterativo el resto se hace cero. Finalmente cada nodo  $l_i$  conoce el número de muestras  $n_i$  que debe generar. El reparto de un número  $N$  de forma proporcional no requiere que las probabilidades  $p_i$  estén normalizadas, siempre que no sean todas valores nulos.

### 3.7. 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 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*.

Asociado a cada *quadtree* 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:

$$\text{Imagen}(\mathbf{u}, i_x, j_y) \simeq f_r(\mathbf{u}, \mathbf{v}_{xy}).$$



**Figura 4:** Izquierda: Es necesario reducir el área de rechazo en determinadas regiones del disco. Derecha: Se han localizado y codificado las regiones que deben ajustar el área de muestreo.

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.

### 3.8. Ajuste de la región para el muestreo por rechazo

Por la relación existente entre un cuadrado y un círculo inscrito en él, las regiones que cuentan con la frontera del círculo, sobre todo las esquinas del plano presentan un área de rechazo muy alta, ya que la porción del disco es pequeña. Esta situación es la observada en la figura 4 izquierda, y nos planteamos mejorar el área de la región.

En lugar de usar 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. Primero calculamos la distancia al centro del círculo para los puntos  $u, v, U$  y  $V$ . Acto seguido evaluamos si esa distancia es mayor o menor a 1 y por tanto podemos saber qué puntos están fuera del círculo. De estos tests conseguimos un código booleano o equivalentemente decimal (como se ve en la figura 4 derecha) mediante el cual podemos detectar en qué caso o situación estamos y ajustar la región en consecuencia.

Por ejemplo en el caso de la figura 4 izquierda, se debe intersectar el círculo con dos aristas para encontrar el punto  $(x, y)$ . Tras lo cual, actualizamos el área de la región a  $(x, y) \times (U, V)$  y disminuimos de esta forma el área de rechazo.

### 3.9. Coste del preprocesamiento

Un número determinado de estructuras *quadtree*, correspondientes a una serie de valores discretos del ángulo con la normal de 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

BRDF	Adaptativa Disco		Factorización
	60QTs	90QTs	Mejor.Param.
Ashikhmin	57.61	83.46	3.12
Beard-Maxwell	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 1:** Tiempos (en segundos) de inicialización del algoritmo en donde se realizan los precálculos que optimizan su uso. Los datos son referentes a la escena de la esfera.

resolución del ángulo incidente de unidad expresada en grados. En algunos casos hemos tomado 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 utilizado 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 métodos bajo diferentes situaciones —las explicadas en las subsiguientes secciones de resultados— obteniendo los datos mostrados en la tabla 1. 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.

## 4. Resultados

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. 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*.

### 4.1. Muestreo general de BRDFs

Para apreciar mejor el ajuste realizado por el método de muestreo es necesario encontrar una situación en la que la función de reflectancia presente un alto pico o variabilidad que sea difícil de muestrear. Podemos hallar esta situación en una esfera iluminada por una fuente de luz. Si la misma es altamente especular, como 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.

Para la evaluación de nuestro método adaptativo de muestreo para 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 1. Igualmente otros parámetros que varían en las pruebas son el número de muestras a generar, y el procedimiento de muestreo. Cada BRDF ha sido muestreada con una de estas PDFs: (1) muestreo aleatorio uniforme, (2) lóbulo coseno en  $S^2$  (3) la anterior en  $\Omega$ , (4) la representación factorizada de Lawrence [LRR04] y (5) nuestro método adaptativo. Así, pues, para el experimento —la comparativa dada en la figura 5— se generaron  $16 \times 5 \times 9 = 720$  imágenes.

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 y a la factorización de la BRDF. Se ha observado que el valor del 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 polares. 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 lineales, tal como Levenberg-Marquardt [PFTV92]. 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.

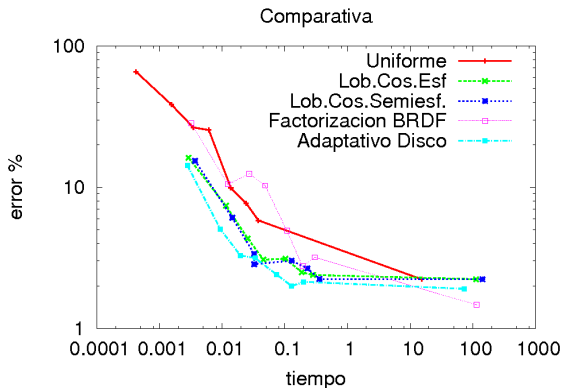
Por otro lado, la factorización de BRDFs analíticas necesita encontrar la mejor factorización para cada instancia de BRDF, ya que con una factorización constante no aproximáramos adecuadamente la instancia de la BRDF. Para

este trabajo, hemos encontrado la mejor factorización de la siguiente forma: comparamos el valor medio de la matriz inicial 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.

Bajo estas condiciones —con la mejor parametrización para los métodos con los que nos comparamos— hemos realizado el experimento que se resume en la tabla 2 y en la gráfica 5. En ambas se relacionan los resultados de los distintos métodos de muestreo en cuanto a la relación error vs. tiempo. Cuanto mayor sea el tiempo utilizado, la imagen resultante converge a la imagen de mínimo error. La gráfica que resulte más a la izquierda y por debajo que las otras, disminuye el error del estimador en menos tiempo.

	error	tiempo
Uniforme	5.82 %	0.03792
Lóbulo C. en $S^2$	2.39 %	0.28328
Lóbulo C. en $\Omega$	2.24 %	0.35734
Factorización BRDF	3.19 %	0.2986
Adaptativa Disco	2.13 %	0.20032

**Tabla 2:** Error medio relativo y tiempo medio de muestreo en segundos utilizado por cada PDF. Los datos son relativos a  $50^2$  muestras cuando se comparan con una imagen de referencia que utiliza  $1000^2$  muestras.



**Figura 5:** Comparativa de PDFs para el muestreo de la BRDF en la escena de la esfera.

#### 4.2. Muestreo de varias BRDFs en la misma escena

A continuación cambiamos a otra escena, consistente en tres jarras, para las cuales se emplean las siguientes BRDFs —en notación GRF [MUR\*02]— considerando las geometrías de las jarras de izquierda a derecha:



**Figura 6:** Izquierda: el muestreo uniforme empleando 1,970 segundos tiene un 36,81 % de error. Derecha: muestreo basado en el lóbulo coseno en  $S^2$ . El tiempo de muestreo es 15,200 segundos y el error relativo 20,86 %.



**Figura 7:** Izquierda: la factorización de la BRDF empleando 26,337 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.

- el modelo de Beard-Maxwell [MBWL73] con  $\Omega=1$   $\tau=0$  el índice de refracción del oro y  $rs=0.7$   $rd=0.5$   $rv=0.2$ . Se utilizará 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 de factores  $J \times K$  utilizados han sido  $2 \times 3$ , junto con 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 toma los siguientes valores:  $16 \times 16$   $32 \times 16$   $J = 2$   $K = 1$ , 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:  $16 \times 16$   $100 \times 32$   $J = 2$   $K = 2$  con reparametrización en  $\mathbf{h}$ .

Con esta escena se hace hincapié en el hecho de que los métodos existentes deben ajustarse a cada una de las BRDFs de la escena. Nuestro método, por el contrario, es general y no necesita de la intervención del usuario. Una vez observado el resultado de las figuras 6 y 7, podemos concluir que nuestro método obtiene resultados satisfactorios con situaciones variables en complejidad y calidad.

#### 4.3. Muestreo de BRDFs adquiridas

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.

Matusik [MPBM03] proporciona una base de datos pública con datos de reflectancia de 100 materiales isotrópicos reales. 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 que ocupa unos 33 MB. Nosotros hemos usado un subconjunto de estos ficheros binarios para probar nuestro método de muestreo adaptativo para BRDF analíticas [MULG08], dado el hecho de que dados dos vectores, uno incidente y otro reflejado, podemos evaluar la BRDF adquirida.

Nuestro método se compara frente al muestreo compacto de la CDF [LRR05] o *Compress Cascade CDF*. Esta técnica de muestreo consigue reducir considerablemente el tamaño de las tablas CDF. El resultado es muy similar a la versión no compacta, que adolece de problemas de almacenamiento, ya que puede llegar a requerir tanto o más espacio que la propia BRDF adquirida. Los datos del proceso de compresión revelan que en media se requieren 2090,57 segundos para reducir los 33MB a algo más 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  $32 \times 16 \times 1024 \times 512$ .

Nuestro método exacto de muestreo con 60 estructuras *quadtree* requiere algo más de un minuto de tiempo 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 8. Con  $5^2$  muestras contribuimos más al estimador que con las mismas muestras las dadas por otras técnicas de muestreo y todo ello con un consumo de memoria mínimo.

## 5. Conclusiones

Hemos presentado un método de selección de direcciones en base a la BRDF por el término coseno que cumple que es válido para un conjunto amplio de BRDFs independientemente de sus características y cómo esten representadas: de forma analítica o procedente de mediciones. Se ha comparado favorablemente con otras técnicas de muestreo estándar y generales, ya que para un número determinado de muestras

MERL BRDF	Uniforme	UnifCDF	CompCDF	Adap. Disco
ALUM-BRONZE	0.35	2.20	1.15	16.03
ALUMINA-OXIDE	0.35	2.31	0.91	16.29
BEIGE-FABRIC	0.34	4.01	0.94	4.04
BLUE-METALLIC-PAINT2	0.37	1.97	1.11	50.55
BLUE-METALLIC-PAINT	0.35	2.20	1.11	4.60
NICKEL	0.37	1.87	1.13	12.31
RED-PLASTIC	0.37	2.59	0.98	4.31
TEFLON	0.34	2.47	0.93	4.25
VIOLET-ACRYLIC	0.34	2.19	1.13	38.86
WHITE-MARBLE	0.34	2.31	0.97	12.54
YELLOW-PAINT	0.34	2.37	0.91	3.86
PROMEDIO	0.35	2.41	1.02	15.24

**Tabla 3:** Datos referentes al tiempo de muestreo de distintos métodos aplicados a BRDFs adquiridas.



**Figura 8:** Para cada imagen de  $450 \times 220$  se han tomado 25 muestras. Columna izquierda con *Cascade CDF* comprimida y columna derecha con *PDF Adaptativa en disco*.

disminuye el error visible. Como resultado se pueden sintetizar escenas que hagan uso de varias BRDFs complejas sin necesidad de guía por parte del usuario.

## 6. Agradecimientos

Este trabajo ha sido financiado por el proyecto de investigación TIN2004-07672-C03-02 (Comisión Interministerial de Ciencia y Tecnología).

## Referencias

- [Arv95] ARVO J.: 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* (1995), ACM Press, pp. 335–342.
- [BGH05] BURKE D., GHOSH A., HEIDRICH W.: Bidirectional importance sampling for direct illumination. In

- Eurographics Symposium on Rendering* (2005), pp. 147–156.
- [Bli77] BLINN J. F.: Models of light reflection for computer synthesized pictures. In *SIGGRAPH '77: Proceedings of the 4th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1977), ACM Press, pp. 192–198.
- [BSW00] BEKAERT P., SBERT M., WILLEMS Y. D.: Weighted importance sampling techniques for monte carlo radiosity. In *Rendering Techniques 2000 (Proceedings of the Eleventh Eurographics Workshop on Rendering)* (New York, NY, 2000), Peroche B., Rushmeier H., (Eds.), Springer Wien, pp. 35–46.
- [CETC06] CLINE D., EGBERT P. K., TALBOT J. F., CARDON D. L.: Two stage importance sampling for direct lighting. In *In Rendering Techniques 2006 (Eurographics Symposium on Rendering)* (2006), pp. 103–113.
- [CJAMJ05] CLARBERG P., JAROSZ W., AKENINE-MÖLLER T., JENSEN H. W.: Wavelet importance sampling: efficiently evaluating products of complex functions. *ACM Trans. Graph.* 24, 3 (2005), 1166–1175.
- [CPC84] COOK R. L., PORTER T., CARPENTER L.: Distributed ray tracing. In *SIGGRAPH '84: Proceedings of the 11th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1984), ACM Press, pp. 137–145.
- [HTSG91] HE X., TORRANCE K., SILLION F., GREENBERG D.: A comprehensive physical model for light reflection. In *SIGGRAPH '91: Proceedings of the 18th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1991), no. 4, ACM Press, pp. 175–186.
- [JC95] JENSEN H. W., CHRISTENSEN N.: Photon maps in bidirectional monte carlo ray tracing for complex objects. In *Computer & Graphics* (1995), no. 2, pp. 215–224.
- [Kaj86] KAJIYA J. T.: The rendering equation. In *SIGGRAPH '86: Proceedings of the 13th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1986), ACM Press, pp. 143–150.
- [Lew93] LEWIS R. R.: Making shaders more physically plausible. In *Fourth Eurographics Workshop on Rendering, Eurographics'93* (Vancouver, BC, Canada, Canada, 1993), University of British Columbia, pp. 47–62.
- [LRR04] LAWRENCE J., RUSINKIEWICZ S., RAMAMOORTHI R.: Efficient importance sampling using a factored representation. In *ACM Transaction of Graphics. Siggraph 2004* (2004), no. 3, pp. 496–505.
- [LRR05] LAWRENCE J., RUSINKIEWICZ S., RAMAMOORTHI R.: Adaptative numerical cumulative distribution functions for efficient importance sampling. In *Eurographics Symposium on Rendering* (2005).
- [LW93] LAFORTUNE E. P., WILLEMS Y. D.: Bidirectional path tracing. In *Proceedings of Computer Graphics, Alvor, Portugal* (1993), pp. 145–153.
- [LW94] LAFORTUNE E. P., WILLEMS Y. D.: *Using the Modified Phong Reflectance Model for Physically Based Rendering*. Tech. Rep. Report CW197, Department of Computer Science, K.U.Leuven, Leuven, Belgium, 1994.
- [MBWL73] MAXWELL J. R., BEARD J., WEINER S., LADD D.: *Bidirectional reflectance model validation and utilization*. Tech. rep., Technical Report AFAL-TR-73-303, Environmental Research Institute of Michigan (ERIM), 1973.
- [MPBM03] MATUSIK W., PFISTER H., BRAND M., McMILLAN L.: A data-driven reflectance model. *ACM Trans. Graph.* 22, 3 (2003), 759–769.
- [MULG08] MONTES R., UREÑA C., LASTRA M., GARCÍA R.: Generic brdf sampling: A sampling method for global illumination. In *GRAPP 2008 Proceedings of the Third International Conference on Computer Graphics Theory and Applications* (Funchal, Madeira, Portugal, 2008), pp. 191–198.
- [MUR\*02] MONTES R., UREÑA C., REVELLES J., LASTRA M., GARCÍA R.: Wannabe amazing, una herramienta de trabajo para iluminación global. In *Jornadas Regionales de Informática Gráfica* (Jaén, España, 2002), pp. 147–158.
- [PFTV92] PRESS W., FLANNERY P., TEUKOLSKY S., VETTERLING W.: *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 1992.
- [Pho75] PHONG B.-T.: Illumination for computer generated pictures. In *Computers Graphics, ACM Siggraph'75 Conference Proceedings* (New York, NY, USA, 1975), no. 6, ACM Press, pp. 311–317.
- [Ros97] ROSIN P. L.: Techniques for assessing polygonal approximations of curves. In *IEEE Transactions on Pattern Analysis and Machine Intelligence* (1997), no. 6, pp. 659–666.
- [Rus98] RUSINKIEWICZ S.: A new change of variables for efficient brdf representation. In *Ninth Eurographics Workshop on Rendering* (1998), pp. 11–23.
- [SBG95] SHIRLEY P., BRETTON W., GREENBERG D.: Global illumination via density-estimation radiosity. In *Eurographics Rendering Workshop* (1995).
- [Vea97] VEACH E.: *Robust Monte Carlo methods for light transport simulation*. PhD thesis, Stanford University, 1997.
- [War94] WARD G. J.: The radiance lighting simulation and rendering system. In *SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1994), ACM Press, pp. 459–472.