

# An Importance Sampling Method for arbitrary BRDFs

Rosana Montes, Carlos Ureña, Rubén García, and Miguel Lastra

Dpt. Lenguajes y Sistemas Informáticos,  
E.T.S.I. Informática y de Telecomunicación  
University of Granada, Spain  
{rosana,curena,ruben,mlastral}@ugr.es

**Abstract.** This paper introduces a new BRDF sampling method with reduced variance, which is based on a hierarchical adaptive PDF. This PDF also is based on rejection sampling with a bounded average number of trials, even in regions where the BRDF exhibits high variations. Our algorithm works in an appropriate way with both physical, analytical and measured reflectance models. Reflected directions are sampled by using importance sampling of the BRDF times the cosine term. This fact improves computation of reflected radiance when Monte-Carlo integration is used in Global Illumination.

## 1 Introduction

In Global Illumination software the Bidirectional Reflectance Distribution Function (BRDF) is used to describe how light is scattered at surfaces, and it determines the appearance of objects. Many reflection models have been proposed which account for real visual effects produced by object-to-object reflections, self-shadowing, retro-reflection, etc. Monte Carlo (MC) algorithms, which rely on BRDF sampling, include distributed ray tracing [6], path tracing [11], bidirectional path tracing [12], density estimation [23] and photon mapping [10].

A mayor challenge in incorporating complex BRDFs into a Monte-Carlo-based global illumination system is efficiency in sampling, however, complex reflectance models have no corresponding sampling strategies to use with. In [14] a Monte-Carlo importance sampling technique was presented for general analytic and measured BRDFs based on its factorization. We have used factorized approximations of those BRDFs in order to compare Lawrence's approach with ours.

This document presents a method to improve Monte-Carlo random walks by applying importance sampling of BRDFs to reduce the variance of the estimator. Reflected directions are generated with a probability density function that is exactly proportional to the BRDF times the cosine term. For generality, we have sampled many parametric BRDFs that are well-known in computer graphics: for plastics the Phong model and its variants [20, 5, 16, 13] and [22], for metals the He model [9], Strauss [24], Minnaert Lunar reflectance [18], for rough and

polished surfaces based on Torrance’s microfacet representation [4, 7, 21] and [19]. Anisotropy models [25, 2, 3] are also considered. We are able to sample reflectance data from measurements as well. We use a subset of the 100 materials of Mitsubishi Electric Research Laboratories (MERL) BRDF data base<sup>1</sup>. In fact our representation makes no assumptions on the BRDF model but the need for evaluating the function giving two directions.

The rest of this document is organized as follow: Section 2 gives an overview of current techniques for sampling the BRDF and explains how importance sampling works when Monte Carlo integration is used. Section 3 provides details of our algorithm which adaptively samples the BRDF. Results and time-error analysis are given in Section 4. Some discussion and ideas for future work conclude the paper.

## 2 Reflectance Equation and Monte-Carlo Estimation

One of the main interests in Global Illumination relies on the evaluation of the reflected radiance, by using the *reflectance equation*:

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

Here  $L_i$  stands for incoming radiance and  $L_r$  for reflected radiance. The above equation is usually solved in global illumination by using MC integration, because it is often impossible to obtain analytic expressions for  $L_r$  or  $L_i$ .  $\mathbf{w}_o = (u_x, u_y, u_z)$  and  $\mathbf{w}_i = (v_x, v_y, v_z)$  are two unit-vectors in  $\Omega$ , the hemisphere of unit radius with  $\mathbf{n} \stackrel{\text{def}}{=} (0, 0, 1)$ .

### 2.1 MC Numerical Estimation of $L_r$

Integration over the hemisphere  $\Omega$  can be done by using three related measures defined in that domain: (1) the solid angle measure (which we note as  $\sigma$ ), (2) the projected solid angle measure ( $\sigma_p$ ) and (3) an area measure  $A$ .

$$(\mathbf{w} \cdot \mathbf{n}) d\sigma(\mathbf{w}) = d\sigma_p(\mathbf{w}) = dA(h(\mathbf{w})) \quad (2)$$

Let  $D$  denote the unit radius disc in  $\mathbb{R}^2$ . By using equation (2), the reflectance equation (1) can be alternatively expressed as:

$$L_r(\mathbf{w}_o) = \int_D f_r(\mathbf{w}_o, \mathbf{w}_{xy}) L_i(\mathbf{w}_{xy}) dA(x, y) \quad (3)$$

where  $\mathbf{w}_{xy} \in D$  is the projection of  $\mathbf{w}_i$  onto  $D$ .

When numerical integration of an arbitrary integrable (w.r.t. a measure  $\mu$ ) function  $g \in S \rightarrow \mathbb{R}$  is done by using MC techniques, random samples in  $S$  must be generated from a random variable with probability measure  $P$  —which obeys

<sup>1</sup> MERL data base: <<http://www.merl.com/brdf/>> [last visit November 2007]

$P(S) = 1$  and it is absolutely continuous w.r.t  $\mu$ — . The function  $p \stackrel{\text{def}}{=} dP/d\mu$  is frequently called the probability density function (PDF) of those samples. From  $n$  such random samples (namely  $\{x_1, \dots, x_n\}$ ) we can build a new random variable (r.v.)  $X_n$  whose mean value is the integral  $I$  we want to compute. This is done by generating samples sets whose PDF is  $p$ , and evaluating  $X_n$  on them. The variance of  $X_n$  is a value which determines the efficiency of the method.

Designing efficient MC sampling methods usually means designing *good* PDFs by using all available information about  $g$ . The closer  $p$  is to  $g/I$  the less variance we obtain (ideally  $p = g/I$ ). Consider now integrals like equation (1) and assume we have no knowledge about irradiance or other terms of the integrand, but with a known BRDF. In these circumstances, the best option is to use a PDF which is as proportional as possible to the BRDF times the cosine term.

To compute an estimator of  $L_r(\mathbf{w}_o)$ , as defined in equation (1), for a given  $\mathbf{w}_o \in \Omega$ , we must use a set of samples  $(\mathbf{s}_1, \dots, \mathbf{s}_n)$ , which are  $n$  identically distributed random vectors defined in  $\Omega$ , with probability measure  $P_{\mathbf{w}_o}$  (the probability measure depends on  $\mathbf{w}_o$ ). With this sample set, the estimator of the outgoing radiance can be obtained as:

$$L_r(\mathbf{w}_o) \approx \frac{1}{n} \sum_{k=1}^n \frac{f_r(\mathbf{w}_o, \mathbf{s}_k) (\mathbf{s}_k \cdot \mathbf{n})}{q_{\mathbf{w}_o}(\mathbf{s}_k)} L_i(\mathbf{s}_k) \quad (4)$$

where  $q_{\mathbf{w}_o} = dP_{\mathbf{w}_o}/d\sigma$  is the PDF associated to  $P_{\mathbf{w}_o}$ .

An alternative expression can be given by using equation (3) instead of (1) and it is used in our algorithm. In this case, the set of samples  $((x_1, y_1), \dots, (x_n, y_n))$  contains random vectors in  $D$  instead of in  $\Omega$ , and the estimator becomes:

$$L_r(\mathbf{w}_o) \approx \frac{1}{n} \sum_{k=1}^n \frac{f_r(\mathbf{w}_o, \mathbf{s}_k)}{p_{\mathbf{w}_o}(x_k, y_k)} L_i(\mathbf{s}_k) \quad (5)$$

where  $\mathbf{s}_k$  is the projection of  $(x_k, y_k)$  onto  $\Omega$ .

In this case, the PDF  $p_{\mathbf{w}_o} = dP_{\mathbf{w}_o}/d\sigma_p = dP_{\mathbf{w}_o}/dA$  is defined w.r.t. area measure  $A$ , and its domain is  $D$ . Finally, from equations (4) and (5) we conclude that the PDF must be evaluated, and thus we should be capable to do this in a short time.

## 2.2 Sampling the BRDF

**Lobe distribution sampling** A well known class of BRDF models are based on cosine-lobes, which have an associated algorithm for sampling. Within this category are Phong, Blinn and their respective normalized versions delivered by Lewis, Lafortune and Ward. The single-lobe BRDF is defined as:

$$f_r(\mathbf{w}_o, \mathbf{w}_i) = C(n) (\mathbf{w}_i \cdot \mathbf{w}_{or})^n$$

where  $n \geq 0$  is a parameter, and  $C(n)$  is a normalization factor which normally depends on  $n$  and ensures these BRDFs obey conservation of energy.

For this BRDF, a related and normalized PDF can be defined as:

$$p_{\mathbf{w}_o}(\mathbf{w}_i) = \frac{1}{N_1(\mathbf{w}_o, n)} (\mathbf{w}_i \cdot \mathbf{w}_o)^n$$

where  $N_1$  ensures normalization and is defined as:

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

$N_1$  is called a *single axis moment around axis*  $\mathbf{a}$  and analytical expressions for it are known [1]. In order to obtain samples distributed according to this PDF, we obtain a random vector  $\mathbf{w}_i$  whose spherical coordinates are:

$$(\theta_{\mathbf{w}_i}, \phi_{\mathbf{w}_i}) = \left( \arccos \left( \xi_1^{\frac{1}{n+1}} \right), 2\pi \xi_2 \right)$$

where  $\xi_1$  and  $\xi_2$  are two independent uniformly distributed random variables with values in  $[0, 1)$ .

A variant of this PDF avoids evaluation of  $N_1$  by using samples on the whole sphere  $\mathcal{S}^2$ , instead of only the hemisphere  $\Omega$ . Taking into account the part of the lobe under the surface, it makes  $N_1(\mathbf{w}_o, n)$  independent of  $\mathbf{w}_o$  and equal to  $N_1(\mathbf{n}, n) = 2\pi/(n+1)$ . This PDF is defined in the sphere, however, when a sample is produced under the surface, the contribution of that sample to the integral is taken as zero. The algorithm is faster and still unbiased, but it has higher variance when  $\mathbf{w}_o$  approaches grazing angles.

Cosine-lobe sampling is the most efficient sampling for Phong BRDF and its variations but this scheme is not suitable for non-lobe-based BRDFs.

**The Factorized BRDF Representation** Recent work about effective importance sampling strategies for arbitrary BRDFs is Lawrence’s factorization of the BRDF [14]. This function is decomposed as the product of two 1D functions, stored compactly in tabular form, and then it is used for sampling.

A first factorization, after a reparametrization based on the half angle, gives a decomposition into 2D factors of the initial data matrix  $Y$  containing  $N_{\mathbf{w}} \times N_{\mathbf{w}_o}$  values along the outgoing elevation angle and the outgoing azimuthal angle. After that,  $Y$  is approximated by the product of two matrices of lower dimension:  $G$  is  $N_{\mathbf{w}} \times J$  and  $F$  is an  $J \times N_{\mathbf{w}_o}$  matrix. Both matrices are always positive by using the non-negative matrix factorization (NMF) method.

A second factorization of the view independent  $G$  matrix leads to the product of two matrices of one dimension, very easy to sample by numerical inversion of the Cumulative Distribution Function after normalization.

$$f_r(\mathbf{w}_o, \mathbf{w}_i) \cos(\mathbf{w}_i) \approx \sum_j^J F_j(\mathbf{w}_o) \sum_k^K u_{jk}(\theta_{\mathbf{w}}) v_{jk}(\phi_{\mathbf{w}}).$$

Each  $L = J \times K$  factor is intuitively the approximation of a specific lobe of the original BRDF. When the factorization is used in generating random directions

two steps are necessary. First sampling according to  $F$  selects one of the  $L$  lobes that contributes more energy for the current view. The CDF for this step is recomputed when the outgoing direction changes. Next the hemisphere is sampled according to selected lobe  $l$  by sequential generation of elevation and azimuthal angles using pre-computed CDF for factors  $u_l$  and  $v_l$  respectively.

**The Cascade CDF Method** An improvement of the Factorized BRDF representation is the Cascade CDF method [15]. This is an adaptive technique orientated to the sampling of non uniform functions. The authors apply it to *environment maps* (EM) and acquired BRDFs. This technique is based on sampling by inversion of the CDF. Instead of uniformly distributing the samples, it uses a second and equivalent distribution which is compact. For this to be solved, they start with a N-dimensional PDF and divide it into the product of a 1D marginal distribution  $\tilde{p}$  and a set of 1D conditional distributions. Compression is carried out using the *Douglas-Peucker* greedy algorithm which approximates a curve (in this case the CDF) employing an optimal number of segments.

### 3 Our Algorithm

We consider the reflectance equation given in (3), and the estimator in (5). The proposed sampling scheme yields more samples in areas where the BRDF times the cosine term has higher values, thus achieving importance sampling. The usage of area measure  $A$  on  $D$  is better than  $\sigma$  on  $\Omega$  because this makes it unnecessary to include the cosine term in the formulation or the computation, making the first simpler and the second faster and more reliable. Also, the algorithm is independent of the BRDF and avoids user guidance.

Our method is based on rejection sampling [8]. This is a very simple and well known technique that yields a PDF proportional to any function  $g \in G \rightarrow \mathbb{R}$ . It only requires that  $g$  can be evaluated, and its maximum value  $m$  in  $G$  to be known. However, it runs a loop which in fact can be executed a unbounded number of times, thus it potentially yields large computing times even in the cases when  $g$  can be quickly evaluated.

The probability for a sample to be accepted is  $e/m$ , where  $e > 0$  is the average value of  $g$  in the domain  $G$ . The number of times the main loop is executed (until a valid sample is obtained) is a geometric distribution with success probability  $e/m$ , and thus the average number of trials is  $m/e$ , which can be quite large for  $e \ll m$ .

The core of our approach is an hierarchical quadtree structure which can be used to efficiently obtain samples with a PDF exactly proportional to the target function. The adaptive approach checks whether a region can be safely used for raw rejection sampling. This check consists on evaluating, for that region, the average number  $n_t$  of trials with rejection sampling in that region. This can be known provided we know both  $e$  and  $m$  for the region. If  $n_t$  is above a threshold number  $n_{max}$ , then the region is subdivided in four, and the criterion is applied to these four subregions. Otherwise, the region is not subdivided. If we apply this

recursive process starting from  $D$  (the unit radius disc centered at the origin), we obtain a quadtree which can be used to efficiently sample the BRDF. In the next section, further details are given about this process.

### 3.1 Building the Adaptive Structures

As the sampling process requires a PDF proportional to  $f_r(\mathbf{w}_o, \cdot)$  for arbitrary values of  $\mathbf{w}_o$  and for a finite collection of BRDFs in a scene, it is necessary to create a *quadtree* structure that subdivides the unit disc domain for each  $(f_r, \mathbf{w}_o)$  pair. In the case of  $\mathbf{w}_o$ , a finite set of vectors  $S = \{\mathbf{w}_1, \dots, \mathbf{w}_n\}$  can be used. When an arbitrary  $\mathbf{w}_o$  is given, it is necessary to select the nearest  $\mathbf{w}_j$  to  $\mathbf{w}_o$  and use the corresponding structure. The error induced by using  $\mathbf{w}_j$  instead of  $\mathbf{w}_o$  can be reduced by using a large  $n$  and uniformly distributing vectors  $\mathbf{w}_j$ . Note that, since we assume the BRDF to be isotropic, it is enough for  $S$  to include vectors in the plane XZ, thus a rotation must be applied to  $\mathbf{w}_o$  before finding the nearest  $\mathbf{w}_j$ . The inverse rotation must be applied to resulting samples.

For a given quadtree in this structure, each node  $i$  has an associated region  $R_i \subseteq D$ , which it is a square area defined by:

$$R_i = [u_i, u_i + s_i) \times [v_i, v_i + s_i)$$

where  $(u_i, v_i)$  is the lower left vertex of the region boundary and  $s_i$  is the edge length. The region associated to the root node is the full domain  $[-1, 1]^2$ .

The algorithm creates the root node and checks the criteria for subdivision. If the split is necessary, four new child nodes are created, each one with an associated region with a edge length size half of that of the parent. Then, this process is recursively applied to these new four nodes. The recursive algorithm ends in case no split is necessary or a predefined maximal depth is reached.

In order to check the subdivision criteria for node  $i$  these values must be computed:

$$\begin{aligned} M_i &= \max\{ f_r(\mathbf{w}_o, \mathbf{w}_{i xy}) \mid (x, y) \in R_i \} \\ I_i &= \int_{R_i} f_r(\mathbf{w}_o, \mathbf{w}_{i xy}) dA(x, y) \\ V_i &= s_i^2 M_i \end{aligned}$$

$M_i$  is the smaller upper bound for values of  $f_r$  in the  $i$ -th region,  $I_i$  is the integral of the BRDF in the region and  $V_i$  is the volume of the space where rejection sampling is done. Both  $M_i$  and  $I_i$  can be computed by evaluating  $f_r$  on a very dense grid of points in  $R_i$  creating the quadtree, or alternatively a bottom-up approach could be used which starts by obtaining these values at the maximum depth possible (with a high resolution grid) and then it stores them so the data can be used during tree construction. Therefore, the algorithm only requires to be able to evaluate the BRDF. In any case, it holds that the sum of the  $I_i$  values for the four children of a parent node must be equal to that value on the parent.

The subdivision criteria used must ensure that rejection sampling on leaf nodes can be done with an *a priori* bounded number of average trials  $n_{max}$ . This can be easily ensuring that:

$$n_{max} \frac{I_i}{V_i} \geq 1 \quad (6)$$

where the probability for accepting a sample is  $I_i/V_i$ .

When this inequality does not hold, the node must be split. In our implementation, we have used  $n_{max} = 2$ . The larger  $n_{max}$  the less memory that is needed (because the quadtree has smaller depth) and the less time is used for quadtree traversal, but more time is needed for rejection sampling on leaf nodes.

### 3.2 Obtaining Sample Directions

Generating a random direction involves selecting a leaf node and then doing rejection sampling on that node. If the  $i$ -th node is a leaf node, then the probability for selecting it must be proportional to  $I_i$  (more exactly it is  $I_i/I_0$ , if we assume the root node has index 0). A leaf node is selected following a path from the root to the leaf. On each step, starting from the root, the integrals  $I_i$  of the descendant nodes are used for randomly choosing one child to continue the path down.

To do this, we can store in each node  $i$  four values  $F_{i0}, \dots, F_{i3}$ , defined as:

$$F_{ik} = \frac{\sum_{j=0}^k I_{C_{ij}}}{\sum_{j=0}^3 I_{C_{ij}}}$$

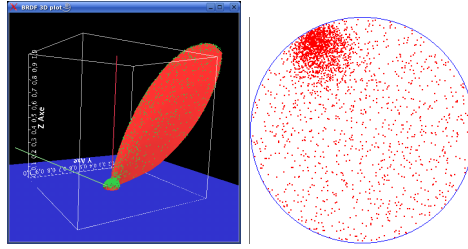
where  $C_{ij}$  is the index of  $i$ -th node  $j$ -th child node (note that  $F_{i3} = 1$ ). Leaf selection is then simply a loop:

```

algorithm LeafNodeSelection
   $i := 0$  (index of root node)
  while  $i$ -th node is not a leaf do begin
     $r :=$  uniform random value in  $[0, 1)$ 
     $j :=$  min. natural such that  $r < F_{ij}$ 
     $i := j$ 
  end
  return  $i$ 

```

Rejection sampling on the resulting  $i$ -th node is carried out. This consists in selecting a random vector  $(x, y, z) \subseteq \mathbb{R}^3$  with uniform distribution in the prism  $R_i \times [0, M_i]$ . The value  $z = \sqrt{x^2 + y^2}$  is then obtained and the condition  $f_r(\mathbf{w}_o, \mathbf{w}_{xy}) < z$  is checked. If it holds,  $\mathbf{w}_{xy}$  is returned as the resulting sample, otherwise a new sample must be generated and checked. A sample is valid with probability  $I_i/V_i$ , which is necessarily greater than  $1/n_{max}$ , because of inequality (6).



**Fig. 1.** Both images show a distribution of 2500 samples obtained with our disc method. The left one shows how the samples match the BRDF function (in red). The image on the right is the projection on disc of those directions.

With our method samples on the disc will follow a distribution where more samples are placed in parts of the domain where the function has higher values. In fact, it is exactly proportional to the BRDF.

### 3.3 Quadtree Traversing for Optimal Sampling

Some considerations should be taken in order to increase the time performance. For example rather than asking for a single sample  $\mathbf{s}_i$  we can implement a single recursive traversal algorithm which yields a set of  $n$  samples. Each node is visited once at most, instead of visiting it  $n$  times as it would be the case when using the basic approach we introduced.

First the algorithm starts by requesting  $n$  samples in the root node region and proceeding recursively. Whenever a node with index  $i$  is visited, the program must produce  $t$  random samples in  $R_i$ . If the  $i$ -th node is a leaf, those  $t$  samples are obtained by rejection sampling. When  $i$ -th node is an inner node, a partition of  $t$  is done, selecting four random integer values  $m_{i,0}, \dots, m_{i,3}$ , which hold  $m_{i,0} + m_{i,1} + m_{i,2} + m_{i,3} = t$  and in such a way that the average value of  $m_{i,j}$  is  $n I_{C(i,j)} / I_i$ . Then the algorithm is recursively called for each  $j$ -th child  $C(i, j)$  of  $i$ -th node (this is not done if  $m_{i,j} = 0$ ), and as a result we obtain four sets with  $t$  samples in total. These four sets can be joined in one, which is the resulting set of  $t$  samples. Each leaf node  $j$  contains  $n I_i / I_0$  samples on the average, as required by importance sampling.

### 3.4 Quadtree Set Construction Requirements

It was mentioned previously that our algorithm involves some computations in order to closely represent any BRDF function. Table 1 shows information related to the cost in seconds of the pre-computation for a given number of quadtree structures and varying incident angle directions. Once we have these structures on memory, they are used to estimate radiance. The values that are listed in the table correspond to the pre-computation of 90 quadtrees, which is high enough



**Table 1.** Quadtree creation times for each BRDF model for Adaptive method compare with factorization and pre-computation times of Factored PDF. Memory requirements for both methods are also given. Data is relative to the glossy scene.

| BRDF      | Adaptive |         | Factorized |      |
|-----------|----------|---------|------------|------|
|           | (sec)    | (KB)    | (sec)      | (KB) |
| Ashikhmin | 51.4     | 6.25    | 82.9       | 1031 |
| BeardMax. | 15.5     | 1713.25 | 17.3       | 6454 |
| Blinn     | 8.7      | 582.25  | 83.7       | 6481 |
| Coupled   | 22.6     | 6.25    | 22.2       | 1033 |
| He        | 102.7    | 2407.25 | 75.1       | 1034 |
| Lafortune | 6.6      | 1275.25 | 53.2       | 6445 |
| Lewis     | 6.9      | 1279.25 | 119.3      | 6445 |
| Minnaert  | 7.3      | 1461.25 | 4.9        | 1031 |
| Oren      | 10.5     | 6.25    | 8.5        | 1033 |
| Phong     | 6.9      | 1279.25 | 9.2        | 6445 |
| Poulin    | 35.5     | 297.25  | 5.4        | 1038 |
| SchlickD  | 19.1     | 342.25  | 77.9       | 1033 |
| SchlickS  | 13.2     | 780.25  | 26.5       | 1043 |
| Strauss   | 10.9     | 727.25  | 59.3       | 1052 |
| Torrance  | 8.3      | 631.25  | 122.0      | 1029 |
| Ward      | 20.7     | 483.25  | 51.3       | 1038 |

**Table 2.** Memory in KBytes for a single quadtree with varying parameters.

| Depth | $n_{max}$ |       |       |       |
|-------|-----------|-------|-------|-------|
|       | 1.3       | 2     | 2.5   | 3     |
| 4     | 12.47     | 6.93  | 4.67  | 3.76  |
| 5     | 29.55     | 14.54 | 10.46 | 8.4   |
| 6     | 68.29     | 32.34 | 23.53 | 18.61 |
| 7     | 150.52    | 67.55 | 46.49 | 32.85 |

to ensure a structure is available very close to any incident direction. Average value is 20.71 seconds compared with 51.27, the cost of factorized computation and pre-computation of CDFs for sampling by using Lawrence’s technique.

Another issue concerning the requirements of our method is memory consumption. Let us consider firstly our basic algorithm with no optimizations. A single quadtree represents the unit disc domain as node regions given an incident direction. Its depth, and so its memory, depends on the  $n_{max}$  parameter (see equation 6). Table 2 shows the cost of a single quadtree at incidence direction when the maximum depth is fixed and the value of  $n_{max}$  varies. We found the optimal value of  $n_{max} = 2$  and we used it in the quadtrees calculated and stored for the Dragon scene. Table 1 shows the cost in KB of these 90 quadtrees for each BRDF. The average value of our method is 0.81 MB compared with 2.67 MB of the factorized BRDF.

## 4 Results

In this section we provide results for our adaptive sampling method for various reflectance models, and we compare the computing time and average relative error we obtain for several images under different sampling strategies (PDFs): (1) uniform sampling technique, (2) cosine lobe sampling on  $\mathcal{S}^2$  and  $\Omega$ , (3) Lawrence’s factorization [14] and (4) the proposed adaptive method. All the images were rendered using a naive path tracing algorithm in a Linux machine with an AMD64 processor and 2GB of RAM.

The maximum quality (1000<sup>2</sup> samples) has been used to produce a reference image. We assign to each image a relative error value, computed with respect to this reference image. We average the relative error for all pixels with non-null radiance in the reference image and report it as a percentage.

### 4.1 Sampling Analytical BRDF Models

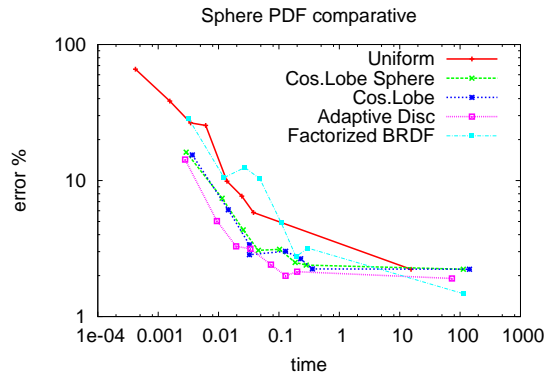
**Glossy Sphere** Considering a sphere object lit by a single area light, we focused our measures on the portion of the image containing the highlight on the sphere, because in that portion is where efficiency of different sampling approaches differs the most.

Each PDF model, with exception of uniform sampling and our method, is assigned a set of manually adjusted parameters in order to match the target BRDF. For example, a cosine-lobe based PDF uses an exponent parameter  $n$ . This value could be taken from the corresponding exponent in the BRDF in use, however, there is no information to set the PDFs exponent if we sample a BRDF model which does not depend on that parameter, thus a constant must be used. To make comparisons fairer we have manually found the exponent that yields the best match between the lobe-based PDF and each BRDF function. Even for Phong-based BRDFs, the best  $n$  for the PDF can be different to the BRDFs exponent. This is because both the PDF and the BRDF include the term  $(\mathbf{w}_o \cdot \mathbf{w}_i)^n$ , however the BRDF also includes the cosine term  $(\mathbf{w}_i \cdot \mathbf{n})$  whereas the PDF does not.

For the Factored PDF we have found the best factorization. It is necessary to find seven values for each BRDF. Parameters are:  $N_{\theta_{\mathbf{w}_o}} \times N_{\phi_{\mathbf{w}_o}}$  and  $N_{\theta_{\mathbf{p}}} \times N_{\phi_{\mathbf{p}}}$  for matrix size,  $J \times K$  for the numbers of lobes that approximates the BRDF and whether or not to use the half-angle reparametrization. Best values are found comparing the average original matrix value with the average from the product of factors.

To compare the various PDF functions we plot the sampling time obtained vs. non-null pixel averaged relative error. By considering this, we can select the best method as the one that gives less error for a given time. The results are plotted in Figure 2. Numerical data is given in Table 3.

As the graph shows, the plot of our method is in most cases below the others. This means that, with same time our sampling performed best and also with same error our method needs less time. The adaptive method can not only be used with any isotropic BRDF but it also does not need manual selection of



**Fig. 2.** PDF comparative for Sphere scene. Manual selection of the cosine lobe exponent is needed, as well as the best factorization has to be found.

**Table 3.** Relative error and average sampling time in seconds for each PDF and test scene when  $50^2$  samples are taken compared to the  $1000^2$  sample reference image.

|                 | error | time    |
|-----------------|-------|---------|
| Uniform         | 5.82% | 0.03792 |
| C.Lobe $S^2$    | 2.39% | 0.28328 |
| C.Lobe $\Omega$ | 2.24% | 0.35734 |
| Adaptive        | 2.13% | 0.20032 |
| Factored        | 3.19% | 0.2986  |

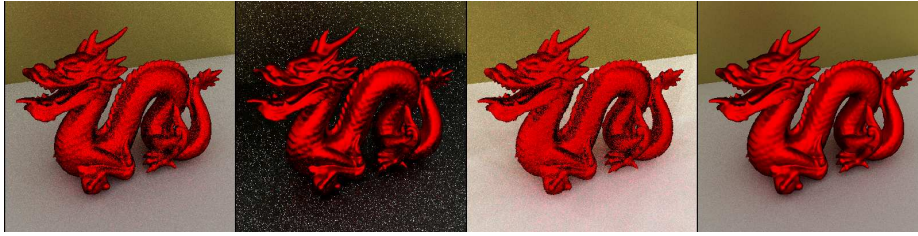
parameters, and it requires no knowledge of the BRDF. It just requires the ability to evaluate the BRDF.

**A Scene with many BRDFs** In this point we treat on the Dragon model from the Stanford University.<sup>2</sup> The reflectance function used in this scene corresponds to Oren’s [19] with a rough value of 0.83, and a Strauss instance [24] mostly smooth for floor and wall respectively. The dragon itself has a Lafortune BRDF [13] with exponent  $n = 20$ . With this mixture of BRDFs, visually we can compare our sampling method with uniform sampling, cosine lobe in  $\Omega$ , and the Factored representation of Lawrence, with manually adjusted parametrization to fit the shape of each BRDF instance. You can see in Figure 3 that with only 100 samples, our algorithm gives results with less noise than the others, and without the need to manually set the parameters for each BRDF.

## 4.2 Adaptive Sampling of Measured Data

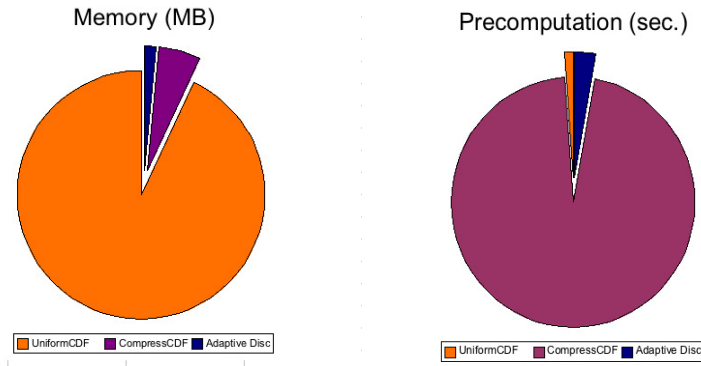
Sampling by inversion of the CDF is a common technique for sampling tabular data such as the BRDF from Merl database. The Uniform Cascade CDF [15] is

<sup>2</sup> The 3D Scanning Repository at <http://graphics.stanford.edu/data/3Dscanrep/>



**Fig. 3.** From left to right, images corresponding to uniform PDF, adjusted cosine-lobe strategy in  $\Omega$ , the Factored representation of the BRDF and finally our algorithm sampling. Adaptive Disc shows less noise using the same number of samples than the others. The resolution is 400 x 400 pixels. Following the same order, sampling time is: 9.232, 114.735, 90.028 and 133.172 seconds respectively.

an approximate method which uses a resolution of  $32 \times 16 \times 256 \times 32$ . This implies significantly more storage requirements than the BRDF itself (as shown in Figure 4) and becomes prohibitive in scenes with many BRDF instances. The compression of the CDF tables by Lawrence [15] lets us use this method without so much memory penalization. The CDF curve is approximated and the resulting CDF table is much smaller, allowing fast sampling by the binary search procedure. As could be seen in Figure 4 the compression step is time consuming.

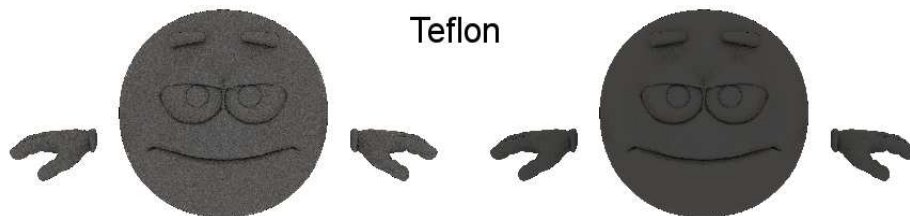


**Fig. 4.** When sampling measured data our algorithm imposes no penalty with the precomputation.

The Adaptive Disc sampling using the precomputation of 60 *quadtrees* do not need as much time as the compressed Cascade CDF nor do we require so much memory (an average of 445.33 KB) as the the uniform and even, the compressed CDF tables. On the other hand we use more sampling time to deliver  $5^2$  samples that best contribute to the estimator (see Figure 5). Numerical data relative to the sampling is given in Table 4.

**Table 4.** Sampling times in seconds for a subset of the Merl database using many PDFs.

| MERL BRDF            | Uniform | UnifCDF | CompCDF | Adap.Disc |
|----------------------|---------|---------|---------|-----------|
| 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      |
| AVERAGE              | 0.35    | 2.41    | 1.02    | 15.24     |

**Fig. 5.** Image of  $450 \times 220$  resolution with 25 samples. Left image use Compress Cascade CDF and right image use our Adaptive Disc PDF.

## 5 Conclusion

We have presented a sampling method based on an adaptive and few parameter algorithm which implements a PDF exactly proportional to an arbitrary BRDF. Reflected directions were sampled using importance sampling of the BRDF times the cosine term which is preferable to only sampling the BRDF. The method can be used for numerical Monte-Carlo based integration in global illumination or in other contexts. Its efficiency is similar or even better than standard sampling methods with manually selected optimal parameter values.

We also tested our adaptive sampling method with tabulated BRDF representations [17] since they can be evaluated. We further plan to develop, as future work, a method to acquire BRDF data from an inexpensive 3D scanner, as a way to deal with real world materials and anisotropic measures. A more optimized method will use an area-preserving spherical mapping.

**Acknowledgements** This work has been supported by a grant coded as TIN2004-07672-C03-02 of the Spanish Ministry of Education and Science.

## References

- [1] Arvo, J.: Applications of Irradiance Tensors to the Simulation of Non-Lambertian Phenomena. ACM Press SIGGRAPH'95 Proceedings, 335–342 (1995)

- [2] Ashikhmin, M., Shirley, P.: An anisotropic phong BRDF model. *Journal. Graph. Tools*, vol. 5, num. 2, 25–32 (2000)
- [3] Ashikhmin, M., Shirley, P.: A microfacet-based brdf generator. *ACM Press SIGGRAPH'00 Proceedings* (2000)
- [4] Maxwell, J. R., Beard, J., Weiner, S., Ladd, D.: Bidirectional reflectance model validation and utilization. Technical report AFAL-TR-73-303. ERIM (1973)
- [5] Blinn, J.F.: Models of Light Reflection for Computer Synthesized Pictures. *ACM Press SIGGRAPH'77 Proceedings*, 192–198 (1977)
- [6] Cook, R.L., Porter, T., Carpenter, L.: Distributed ray tracing. *ACM Press SIGGRAPH'84 Proceedings*, 137–145, (1984)
- [7] Cook, R.L., Torrance, K.E.: A Reflectance Model for Computer Graphics. *ACM Press SIGGRAPH'81 Proceedings*, 7–24 (1982)
- [8] Gentle, J.E.: Random number generation and Monte Carlo methods. (2nd ed.) Springer (2003)
- [9] He, X.D., Torrance, K.E., Sillion, F.X., Greenberg, D.P.: A Comprehensive Physical Model for Light Reflection. *ACM SIGGRAPH'91 Proceedings*, 175–186 (1991)
- [10] Jensen, H.W., Christensen, N.: Photon maps in bidirectional monte carlo ray tracing for complex objects. *Computer & Graphics* vol. 19, num. 2, 215–224 (1995)
- [11] Kajiya, J.T.: The rendering equation. *ACM Press SIGGRAPH '86 Proceedings*, 143–150 (1986)
- [12] Lafortune, E.P., Willems, Y.D.: Bi-directional Path Tracing. *Proceedings of Computational Graphics and Visualization Techniques*, 145–153. Alvor, Portugal (1993)
- [13] Lafortune, E.P., Willems Y.D.: Using the Modified Phong Reflectance Model for Physically Based Rendering. Technical Report CW197. Dpt. Computer Science, K.U.Leuven (1994)
- [14] Lawrence, J., Rusinkiewicz, S., Ramamoorthi, R.: Efficient BRDF Important Sampling Using a Factored Representation. *ACM Transaction of Graphics*, vol. 23, num. 3, 496–505 (2004)
- [15] Lawrence, J., Rusinkiewicz, S., Ramamoorthi, R.: Adaptative Numerical Cumulative Distribution Functions for Efficient Importance Sampling. *Eurographics Symposium on Rendering* (2005)
- [16] Lewis, R.R.: Making Shaders More Physically Plausible. *Eurographics Workshop on Rendering*, 47–62 (1993)
- [17] Matusik, W., Pfister, H., Brand, M., McMillan, L.: A data-driven reflectance model. *ACM Trans. Graph.* vol. 22, num. 3, 759–769 (2003)
- [18] Minnaert, M.: The reciprocity principle in Lunar Photometry. *Astrophysical Journal* vol. 93, 403–410 (1941)
- [19] Oren, M., Nayar, S.K.: Generalization of Lambert's Reflectance Model. *ACM Press SIGGRAPH'94 Proceedings*, 239–246 (1994)
- [20] Phong, B.: Illumination for computer generated pictures. *ACM Siggraph'75 Conference Proceedings*, vol. 18, num. 6, 311–317 (1975)
- [21] Poulin, P., Fournier, A.: A Model for Anisotropic Reflection. *ACM Press SIGGRAPH'90 Proceedings*, vol. 24, num. 4, 273–282 (1990)
- [22] Schlick, C.: A Customizable Reflectance Model for Everyday Rendering. *Eurographics Workshop on Rendering*, 73–84 (1993)
- [23] Shirley, P., Bretton, W., Greenberg, D.: Global Illumination via Density-Estimation Radiosity. *Eurographics Workshop on Rendering* (1995)
- [24] Strauss, P.S.: A Realistic Lighting Model for Computer Animators. *IEEE Comput. Graph. Appl.*, vol. 10, num. 6, 56–64 (1990)
- [25] Ward, G.J.: Measuring and modelling anisotropic reflection. *ACM Siggraph'92 Conference Proceedings*, vol. 26, num. 4, 265–272 (1992)