

Fabrice Rousselle  
Petrik Clarberg  
Luc Leblanc  
Victor Ostromoukhov  
Pierre Poulin

## Efficient product sampling using hierarchical thresholding

---

Published online: 10 June 2008  
© Springer-Verlag 2008

---

F. Rousselle (✉) · L. Leblanc ·  
V. Ostromoukhov · P. Poulin  
University of Montreal, PO Box 6128,  
Station Centre-ville, Montréal QC, Canada,  
H3C 3J7

F. Rousselle  
Ecole Polytechnique Fédérale de Lausanne  
(EPFL), Lausanne, Switzerland  
fabrice.rousselle@epfl.ch

P. Clarberg  
Lund University, Box 117, 221 00 Lund,  
Sweden

**Abstract** We present an efficient method for importance sampling the product of multiple functions. Our algorithm computes a quick approximation of the product on-the-fly, based on hierarchical representations of the local maxima and averages of the individual terms. Samples are generated by exploiting the hierarchical properties of many low-discrepancy sequences, and thresholded against the estimated product. We evaluate direct illumination by sampling the triple product of environment map lighting, surface reflectance, and a visibility function estimated per pixel. Our results

show considerable noise reduction compared to existing state-of-the-art methods using only the product of lighting and BRDF.

**Keywords** Importance sampling · Rejection sampling · Multiple functions · Ray tracing · Visibility

---

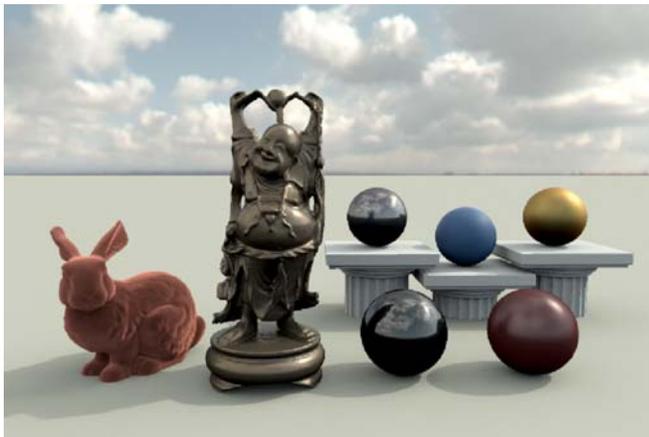
### 1 Introduction

Monte Carlo methods are widely used in photo-realistic rendering, but many samples are needed for noise-free results. Importance sampling is a popular way to improve the performance by concentrating the sampling efforts to important regions. Ideally, the sampling density should be proportional to the function itself, but this is hard to achieve in practice. In this paper, we focus on integrating the direct illumination under environment map lighting. The problem involves a product of the lighting, surface reflectance, and local visibility. This product has to be computed on-the-fly for each pixel, as precomputation is infeasible due to the large amounts of data.

We store hierarchical representations of the local *maxima* and *averages* of the involved functions. For any interval, the product of the functions' individual maxima is always a conservative estimate of the product's local maximum. This can be used for rejection sampling, but many

samples would be rejected in regions where the maximum is overly conservative. Instead, we compute an approximation of the product by hierarchically multiplying the local averages. The estimation is then refined in regions of potentially high contribution, indicated by the local maximum. Samples are generated by thresholding against this approximated product, exploiting the hierarchical properties of many low-discrepancy sequences.

Unlike many previous methods, we aim for quickly generating samples approximately following the product distribution. In terms of overall performance, this is better than going through great effort to create a small set of near optimal samples. This is particularly true today, when ray tracing has reached interactive speeds. The complexity of our product approximation grows only linearly with the number of terms. Hence, it is possible to use more than two functions at a small extra cost. As a proof of concept, we conservatively approximate the visibility per pixel, and directly sample the *triple* product of lighting, BRDF, and visibility.



**Fig. 1.** Our algorithm can sample the product of multiple functions exhibiting a very wide range of frequencies. This image results from sampling the *triple* product of environment map lighting, surface reflectance and estimated visibility

## 2 Previous work

In this paper, we concentrate on computing the direct illumination using Monte Carlo integration. For a general overview of Monte Carlo methods, we refer to [7, 11]. In this context, the rendering equation [12] involves an integral over the product of lighting, material reflectance, and visibility. These are all potentially high-frequency, which makes it expensive to compute their product. In addition, the exact visibility is unknown, and must be locally estimated.

Many techniques exist for importance sampling only one of the three functions. Numerical BRDF models can often be analytically inverted, e.g., [2, 25], and measured materials can be efficiently sampled [15, 16]. Another example is environment map sampling [1, 14, 19]. It is also possible to draw samples from a weighted combination of multiple functions [23]. However, none of these methods take the product of the functions into account.

Talbot et al. [22] suggest *importance resampling*, where an initial set of samples is first drawn from a simpler distribution. Then, by giving the samples appropriate weights and resampling the initial set, they obtain samples approximately following the product distribution. Burke et al. [3] generate a large set of samples according to one of the product's individual terms, and use either rejection sampling or resampling to pick out the most important ones based on the remaining terms.

Other work has focused on explicitly estimating and sampling the product. Clarberg et al. [4] precompute wavelet representations of the lighting and materials. These are multiplied on-the-fly, and uniform points are *warped* into the desired distribution. Cline et al. [5] avoid the precomputation by using a summed-area table for the light source, which is hierarchically divided into smaller

regions based on peaks in the BRDF. A major advantage is that spatially varying materials are supported.

We compute an approximation of the product by hierarchically multiplying precomputed representations of the *maxima* and *averages* of the individual terms. The approximation is adaptively refined, and samples are placed by hierarchical thresholding of low-discrepancy sequences, similar to [19]. Quasi-random numbers [17] are crucial for reducing the variance, and have a long history in computer graphics [13]. Our simple approach has a number of advantages. First, it gives a very fast algorithm. Second, our method allows inexpensive on-the-fly rotations of the involved functions (e.g., the BRDF), which means we avoid storing redundant pre-rotated data, and thus reduce memory requirements. Third, we can include additional terms in the product at a small cost. This opens up for novel sampling strategies.

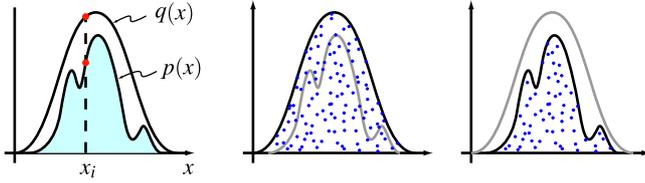
As a proof of concept, we include a third importance function, representing an estimation of the visibility. We use a conservative approximation of the geometry with inner spheres [24], and build a low resolution visibility map per pixel. By including a visibility term, we effectively avoid sampling in directions guaranteed to be occluded. Very few other techniques exist, which exploit visibility to reduce the variance. Ghosh et al. [9] propose a two-pass method. First, they apply bidirectional importance sampling [3] to compute an initial estimate and to identify partially occluded pixels. Then, the noise is reduced by redistributing the variance from nearby pixels using Metropolis sampling.

## 3 Approximate product importance sampling

Our sampling method is based on *hierarchical thresholding* of candidate points against an estimate of the function to be sampled. The idea is to gradually fill the sampling domain, and then perform a rejection test. Thresholding, or *rejection sampling* (see Fig. 2), is a classic Monte Carlo technique for sampling an arbitrary function. First, we extend the method to efficiently handle the product of multiple functions, by using a conservative estimate of the product's maximum. Then, we introduce a fast approximation of the product to avoid a large number of slow evaluations of the individual functions. We use 1D examples throughout for clarity, but our method generalizes to any dimension.

### 3.1 Hierarchical sample generation

We generate candidate samples using low-discrepancy sequences that can be hierarchically constructed. By this we mean any sequence where samples can be iteratively added, while being uniformly distributed. One example is the van der Corput (VDC) sequence [17, 21]. Using a VDC sequence in base  $b$ , a sample's position,  $X_i$ , is



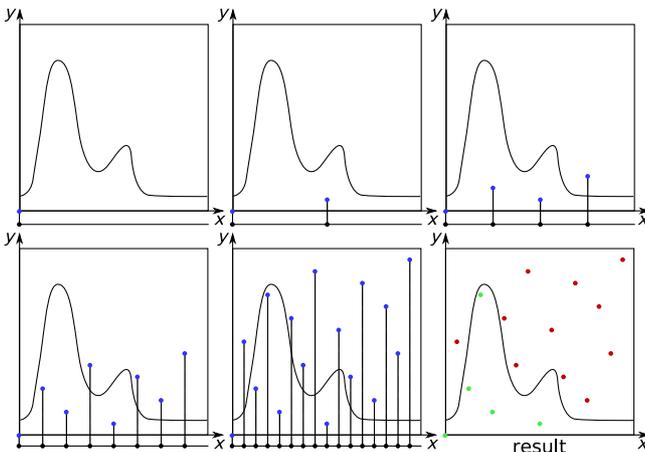
**Fig. 2.** In rejection sampling, samples are first drawn from a simpler *envelope* distribution,  $q(x)$ , and then randomly thresholded against the importance function  $p(x)$ . This can be illustrated as filling the space under  $q(x)$  with random points, and then rejecting all points above  $p(x)$ . The probability of accepting a sample,  $x_i$ , is equal to  $p(x_i)/q(x_i)$

defined as the *radical inverse* of its index  $i$ . Any positive integer  $i$  in base  $b$  can be expressed as a sequence of digits  $d_m \dots d_2 d_1$  uniquely defined by  $i = \sum_{j=1}^m d_j b^{j-1}$ . The radical inverse is obtained by reflecting these digits about the decimal point. For instance, if  $b = 2$ , the radical inverse of  $4 = (100)_2$  is  $(0.001)_2 = 0.125$ .

This sequence can be constructed recursively, in which case each subdivision multiplies the number of samples by  $b$ . We set a maximum level of subdivision  $L$ , which will generate  $N = b^L$  samples, and define a sample's *threshold* value as  $Y_i = i/N$ . Now, for  $f(x)$ , such that  $0 \leq f(x) < 1$ , a sample is rejected if  $f(X_i) < Y_i$ . This process is illustrated in Fig. 3. Note that our definition of threshold values assures that, as samples are generated, their threshold values are strictly increasing. This is a key point that will be exploited in the following section.

### 3.2 Sampling using max trees

The efficiency of rejection sampling relies on how well the envelope function approximates the desired importance



**Fig. 3.** Sample generation gradually filling the space using a van der Corput sequence. Note that  $Y_j > Y_i$  for  $j > i$

function. A bad fit means more tested samples, many of which will be discarded. Consider the case of sampling a product with multiple terms:

$$f(x) = \prod_i f_i(x). \quad (1)$$

The construction of a good envelope function for Eq. 1 can be a difficult problem. For example, in the case of direct illumination, the individual terms represent lighting, BRDF, and visibility.

We propose to precompute a hierarchical representation of the *maximum* for each individual term, which we call the *max tree*. The max tree is created by recursively subdividing the domain, storing the maximum over each child in a tree structure. For discrete functions, e.g., environment maps, finding the maximum over a region is straightforward. For analytical functions, such as many BRDFs, it is in some cases possible to derive an expression for the maximum. However, to remain general, we rely on point sampling of the function, and use extensive oversampling to reduce the risk of missing peaks. This assumes the function is reasonably smooth, but it gives no guarantee of finding the true maximum.

By multiplying together the individual maxima, we get an upper bound for the product. More formally, for any region  $[a, b]$  in the function domain, the following holds true:

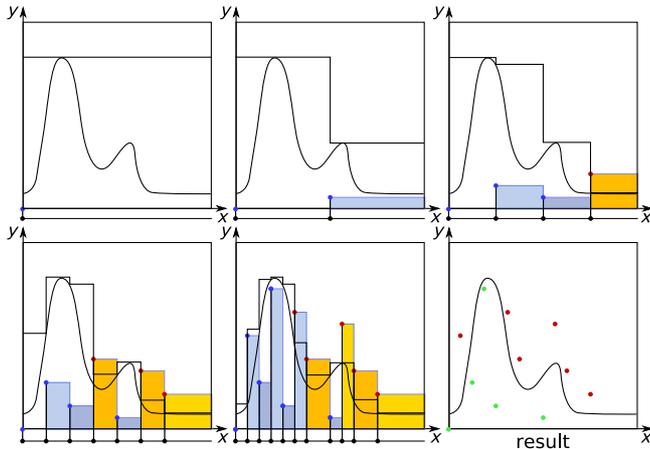
$$\max_{x \in [a, b]} f(x) \leq \prod_i \left( \max_{x \in [a, b]} f_i(x) \right). \quad (2)$$

Note that this upper bound gives a tighter fit at finer subdivisions. We generate samples by recursively subdividing the domain, while evaluating Eq. 2 at each level. Since samples are hierarchically generated and have ever increasing threshold values, we can safely stop the recursion as soon as a sample threshold value is larger than the local maximum, as illustrated in Fig. 4. This effectively limits the number of generated samples.

### 3.3 Product approximation

To speed up the sampling, we compute an *approximation* of the product, against which potential samples are thresholded. Hence, we avoid the expensive evaluation of the involved functions (e.g., environment map and BRDF) for each candidate point. On the negative side, we get samples only approximately following the target distribution, which increases the variance. However, in our application, faster sampling more than makes up for this, in terms of overall quality vs. time.

Previous work on importance sampling for direct illumination has used product approximations based on, e.g., wavelets [4] and summed-area tables [5]. We take a simpler approach, and multiply the local *averages* of the indi-



**Fig. 4.** Using the local maximum (solid black line), we can prune branches where all subsequent samples will be rejected (orange boxes), and thereby reduce the rejection rate

vidual terms. Looking at an interval  $[a, b]$ , we use:

$$\frac{1}{b-a} \int_a^b \prod_i f_i(x) dx \approx \prod_i \left( \frac{1}{b-a} \int_a^b f_i(x) dx \right), \quad (3)$$

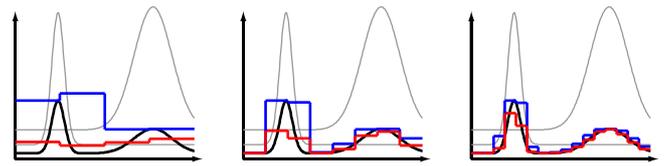
where the individual local averages,  $\int_a^b f_i(x) dx / (b-a)$ , are precomputed and stored in an *average tree*, similar to our *max tree*. We denote the piecewise constant importance function obtained this way by  $h(x)$ .

Equation 3 is clearly a crude approximation to the product, especially at coarse levels in the hierarchy. However, it converges towards the correct result at finer subdivisions. Note that, by construction, we adaptively refine the approximation where the function maximum is large, thereby minimizing the approximation error in regions having a significant contribution to the integral. This is a key point of our method. A challenging example is shown in Fig. 5.

### 3.4 Avoiding bias

The deterministic nature of low-discrepancy sequences implies a fixed distribution of sample positions. To avoid bias, we use *scrambling* [8, 20] and randomly permute the assignment of sample positions when subdividing. The discretization of threshold values is another source of bias, which we address by adding a random offset in the range  $[0, 1/N]$  to each threshold value. This is a well-known technique [11], which ensures that, on average, the correct number of samples is selected.

The use of a low-discrepancy sequence based on the radical inverse yields samples at fixed positions, aligned on a grid defined by the level of subdivision. As the samples only cover a subset of the domain, the solution will



**Fig. 5.** Two non-overlapping peaks present a challenging case. Our product approximation (red) fails at coarse levels, as seen on the left. However, the large maximum (blue) around the left peak, will trigger further subdivision until the missing peaks are found, as shown on the right

be biased. To address this fact, all results in this paper were generated using the VDC-sequence with jittering on sample positions. However, it is possible (but more costly) to achieve better blue noise properties by taking the local neighborhood into account using structural indices, as in, Polyomino-based sampling [18].

### 3.5 Sample count

As mentioned in Sect. 3.1, the number of candidate samples is  $N = b^L$ , where  $b$  is the subdivision factor and  $L$  is the maximum level of subdivision. Since candidates are uniformly distributed, the average number of accepted samples  $\bar{n}$  is equal to  $N$  scaled by the integral of the importance function,  $H = \int h(x) dx$ , as follows:

$$\bar{n} = H \times b^L. \quad (4)$$

To obtain  $n$  samples on average, we scale  $h$  by a factor  $c$ , which is given by:

$$c = \frac{n}{H \times b^L}. \quad (5)$$

It is essential that  $c \cdot h(x)$  remains in  $[0, 1)$  for our rejection test to be valid. This is ensured by increasing  $L$  up to the point where  $c \cdot h(x) < 1$ . Note that  $c$  varies per pixel and cannot be precomputed.

As  $h(x)$  is defined through the sampling process, its integral  $H$  is initially unknown. In order to estimate the number of samples that will be generated, we approximate  $H$  by performing two subdivisions and then computing the average of  $h(x)$  at that level. If the final number of samples widely differs from  $n$ , we refine  $c$  and repeat the process. In our application, using a 20% tolerance, the average number of sampling iterations rarely exceeds 1.3.

### 3.6 Unbiased Monte Carlo integration

In importance sampling, the probability density function does, by definition, integrate to 1. We draw samples from the scaled product approximation,  $c \cdot h(x)$ , which must be divided by its integral,  $c \cdot H$ , in order to meet this criteria.

Since  $h(x)$  is piecewise constant, its integral can easily be computed by summing the contribution of all leaf nodes during the sampling process:

$$H = \frac{1}{b^L} \sum_{i=1}^m h_i \times b^{L-i}, \quad (6)$$

where  $h_i$  is the value of  $h(x)$  over the  $i$ th node, and  $l_i \in [0, L]$  is the node's level in the hierarchy. The size of the  $i$ th node with respect to the maximum level of subdivision,  $L$ , is equal to  $b^{L-l_i}$ . The resulting unbiased Monte Carlo estimator,  $\langle F \rangle$ , for the integral  $F = \int f(x) dx$  is:

$$\langle F \rangle = \frac{H}{n} \sum_{i=1}^n \frac{f(x_i)}{h(x_i)}, \quad (7)$$

where  $H$  is computed using Eq. 6. Note that the term  $H$  plays the same role as  $L_{ns}$  (i.e., “exitant radiance in the absence of shadows”) used in [3, 4].

#### 4 Application – direct illumination

The computation of direct illumination from distant HDR environment map lighting [6] is a problem that has generated considerable interest in recent years. The outgoing radiance is given by [12]:

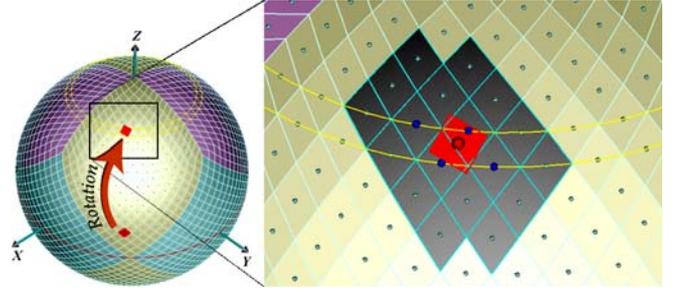
$$L_o(\mathbf{x} \rightarrow \omega_o) = \int_{\Omega} L(\mathbf{x} \leftarrow \omega_i) \rho(\omega_i \leftrightarrow \omega_o) V(\omega_i) d\omega, \quad (8)$$

where the lighting ( $L$ ), reflectance ( $\rho$ ), and visibility ( $V$ ), are integrated over the hemisphere. We define  $\rho$  as the BRDF weighted by the cosine of the incident angle, i.e.,  $\rho = f_r(\omega_i \leftrightarrow \omega_o)(\omega_i \cdot \mathbf{N})$ , as commonly done.

Our algorithm can be used to efficiently sample the product  $L \cdot \rho$ . We also show that an approximated visibility term,  $\tilde{V}$ , can be included. By sampling according to the triple product  $L \cdot \rho \cdot \tilde{V}$ , we further reduce noise in regions with large occlusion. Results with and without the visibility term are presented in Sect. 5.

##### 4.1 HEALPix mapping

All involved functions are defined over the (hemi)sphere, while our algorithm depends on hierarchical subdivision of the domain into quads. We use the HEALPix (hierarchical equal area isolatitude pixelization) mapping [10], and divide the sphere into 12 faces (see Fig. 6), as described by Gorski et al. [10]. Each face is a curvilinear quad, which can be recursively subdivided into  $2 \times 2$  smaller quads of equal area. We apply our sampling scheme on each face separately.



**Fig. 6.** After a rotation in the HEALPix mapping, the source quad usually overlaps a number of quads at the destination. The precomputed average at the *red dot* is linearly interpolated from the nearest neighbors, marked with *blue dots*. Similarly, we ensure that the local maximum is conservative by implicitly considering all quads marked *dark gray*

The HEALPix mapping has a number of desirable properties:

- (1) hierarchical representation,
- (2) area preservation, and
- (3) low distortion.

The preservation of area simplifies our implementation, as we do not have to compute form factors. Low distortion is important when rotating between different domains (see Sect. 4.3).

##### 4.2 BRDF representation

A general BRDF is a 4D function parameterized over incident and outgoing directions,  $(\theta_i, \phi_i)$  and  $(\theta_o, \phi_o)$  respectively. Isotropic BRDFs, currently implemented in our system, are reduced to 3D functions depending only on  $\theta_i$ ,  $\theta_o$ , and  $|\phi_i - \phi_o|$ . We store isotropic materials as 2D *slices*, i.e., one 2D reflectance map  $(\theta_i, |\phi_i - \phi_o|)$  for each  $\theta_o$ . Each slice is first encoded as a mipmap image, and then mapped to the HEALPix representation. Only the data for the upper hemisphere is stored.

All materials in the scene are resampled into this representation as a precomputation step. To avoid missing features, we use oversampling and assume the BRDF is reasonably smooth. This approach is taken by most algorithms using tabulated materials, and rarely presents a problem. The reflectance maps (as well as the environment map) are stored in RGB color, and the local maxima and averages are computed per channel. During sampling, we threshold against the luminance,  $Y$ , computed using the perceptual weighting:  $Y = 0.299R + 0.587G + 0.114B$ .

##### 4.3 Rotations

In our application, the lighting is given by a 2D environment map in world space, while material reflectance is defined in the local surface frame. Hence, a rotation be-

tween the two domains must be performed. This can be precomputed as in [4], but the increased memory requirements would limit us to low-resolution representations. Instead, our algorithm was designed to support fast *on-the-fly* rotation of the importance functions.

The estimations of a quad’s maximum and average (Eqs. 2 and 3) are both *local* operators, depending only on the values of the corresponding nodes in each term. Hence, we can simply rotate the coordinates used for locating a quad in the hierarchical representation. However, after rotation, a quad usually covers multiple quads at the destination. The low distortion in the HEALPix mapping helps reduce the overlap, but special care has to be taken to ensure that our estimation of the maximum remains conservative. We proceed as illustrated in Fig. 6.

#### 4.4 Visibility approximation

One of the advantages of our method is that we can inexpensively include additional terms in the product. A natural extension is to use an estimated visibility map to steer samples away from occluded directions.

We use a visibility estimation inspired by [24]. Each object in the scene is approximated by a set of inner spheres. These spheres are aggregated into a hierarchy, but only leaves act as occluders (to preserve inner conservativity). To create a visibility map, we traverse the sphere hierarchy and the HEALPix hierarchy in parallel. If the cone enclosing a quad of the HEALPix mapping is completely occluded by a leaf sphere, the quad is marked as occluded (zero visibility). Other quads are set to fully visible (one). Then, we propagate these values upwards and store their maxima and averages.

The cost of building the visibility approximation is independent of the number of samples. Hence, the amortized cost is smaller for high-quality rendering, where more samples are used. In our implementation, a visibility map is built per pixel, which is expensive with many occluders. Another approach is to reuse the visibility estimate over several pixels, or use adaptive updates. We have saved this for future work. Other occluder primitives can also be added. In addition to spheres, our implementation supports infinite planes.

The visibility term gives a large variance reduction where the information is accurate. In other parts, e.g., along shadow edges, the effect is smaller. To handle such regions, we have experimented with adaptive sampling. We increase the number of samples for each pixel where the number of occluded visibility rays is above a certain threshold, e.g., 50%, and repeat the sampling process.

#### 4.5 Early termination and biased evaluation

In our algorithm presented in Sect. 3, samples are always placed in leaf nodes at the maximum level of subdivision. This gives an importance function that is accurate, but at

a higher cost. Here, we present two optional extensions for increasing the performance.

First, we propose to terminate the recursion as early as possible. For this, we identify branches with *at most* one sample. Our sample threshold values are, by definition, strictly increasing. Thus, when we reach a point where the *next* threshold value is larger than the local maximum, only the current sample can possibly be accepted. Instead of traversing the hierarchy up to the leaf level, we place the sample in the current node and terminate. This gives a faster algorithm, but at the expense of a small increase in variance.

Second, we propose a biased version of our algorithm. Instead of point sampling the exact  $L$  and  $\rho$  in evaluating the rendering equation, we use the local averages as sample values for samples placed in large nodes. To avoid visual artifacts, e.g., with highly specular materials, we combine unbiased and biased evaluations. This is best explained with an example. Consider the case where the BRDF and the environment map are precomputed up to levels 5 and 8 respectively. If a sample is placed in a node at level 7, we compute the exact BRDF value, but use the precomputed average of the environment over the node. The sample is then assigned the product of these values. Note that this approach, although biased, is *consistent*, i.e., it converges towards the correct solution.

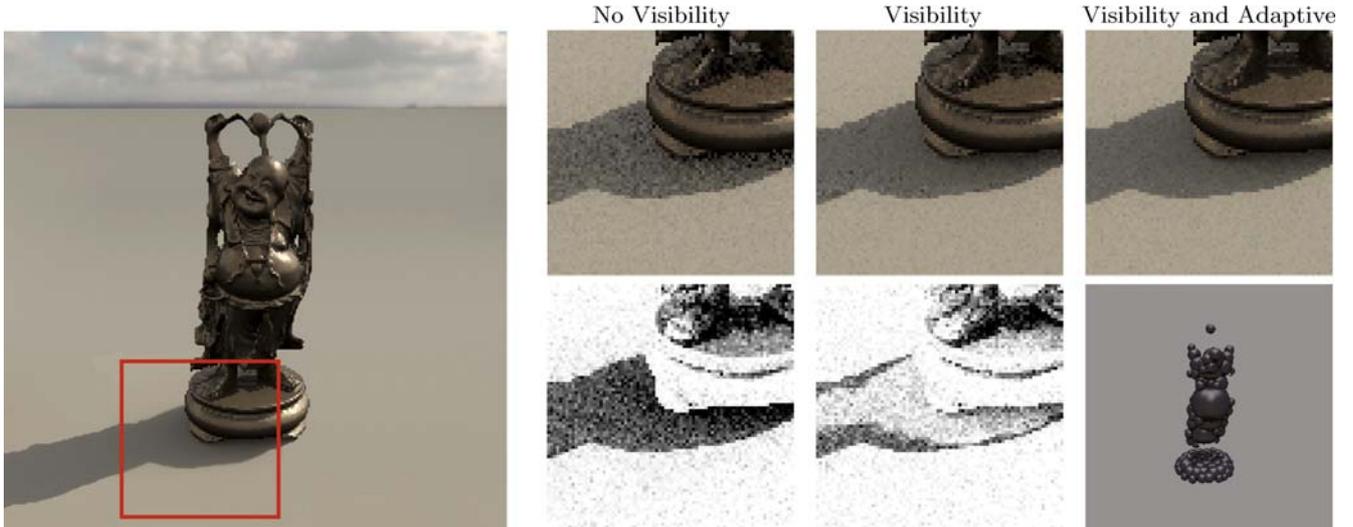
We have found the combination of these two extensions to be extremely useful. When a sample is placed at lower levels in the hierarchy (large nodes), the average tree approximately gives us the integral over the node, instead of a single point-sampled value. This significantly reduces the variance. In Sect. 5, we present results using both the unbiased and the biased versions.

## 5 Results

All results were obtained on a MacBook Pro with an Intel Core 2 Duo 2.4 GHz (using 1 core), and all functions were stored uncompressed in quadtrees of different depths. At depth  $n$ , each of the HEALPix mapping’s 12 faces contain  $2^n \times 2^n$  quads. In all images, the environment map was stored at depth 8, and occupies 24 MB. For a  $4k \times 4k$  angular map, the setup time was 1.46 s.

Isotropic BRDFs are represented with 50 slices computed for outgoing angles uniformly distributed in  $[0, \pi/2]$ . The only user set parameter is the BRDF resolution. In practice, we use depth 5 for most materials, while diffuse materials (e.g., the ground in Fig. 1) are stored at depth 4, and highly specular materials (e.g., the sphere in Fig. 8) are stored at depth 6. The precomputation times for measured materials [16] are given in Table 1.

The estimated visibility function is computed at depth 4 and occupies 8 KB. Figure 7 illustrates a practical situation where a Buddha is approximated by 110 inner spheres. It should be noted that the cost of estimating vis-



**Fig. 7.** Images illustrating the impact of adding visibility information and adaptive sampling, using 16 samples per pixel and biased rendering. The two *gray scale images* (bottom row) show the percentage of occluded rays, where *black* is 100% and *white* 0%. The *bottom right image* shows the sphere approximation used for the Buddha. The *leftmost image* is a reference

**Table 1.** Precomputation times for measured materials

Depth $n$	Memory	Precomputation
4	2.55 MB	0.36 s
5	9.78 MB	1.35 s
6	38.45 MB	5.31 s

ibility is independent of the number of samples, and the cost of including a third term in the product is marginal. For this scene, adding visibility increases the rendering time by about 14 seconds. The rendering times for the left image in Fig. 7, at resolution  $256 \times 256$ , are given in Table 2.

Figure 8 shows a comparison against several recent techniques. Results with sampling of only the BRDF or the environment map are also included. The material is a normalized Phong with diffuse and specular lobes. The shininess coefficient is 5000 for the sphere and 10 for the plane. These values were chosen to illustrate a full range of frequencies. Strong light blocked by occlusion results in a high noise level with algorithms sampling only the product of lighting and BRDF. This effect is diminished by taking visibility into account.

This scene presents an extremely easy case for our visibility estimation, only increasing the rendering time by 11% (with 64 samples). This is not representative for the general case, but it shows, as a proof of concept, that including a visibility approximation can dramatically lower the noise. More sophisticated algorithms for estimating the visibility term have been left for future work. The

**Table 2.** Rendering times for the left image in Fig. 7 at resolution  $256 \times 256$

Number of samples	No visibility	Non-adaptive visibility	Adaptive visibility
16	4.1 s	17.8 s	18.3 s
128	14.7 s	28.5 s	33.0 s
512	49.8 s	64.0 s	81.0 s

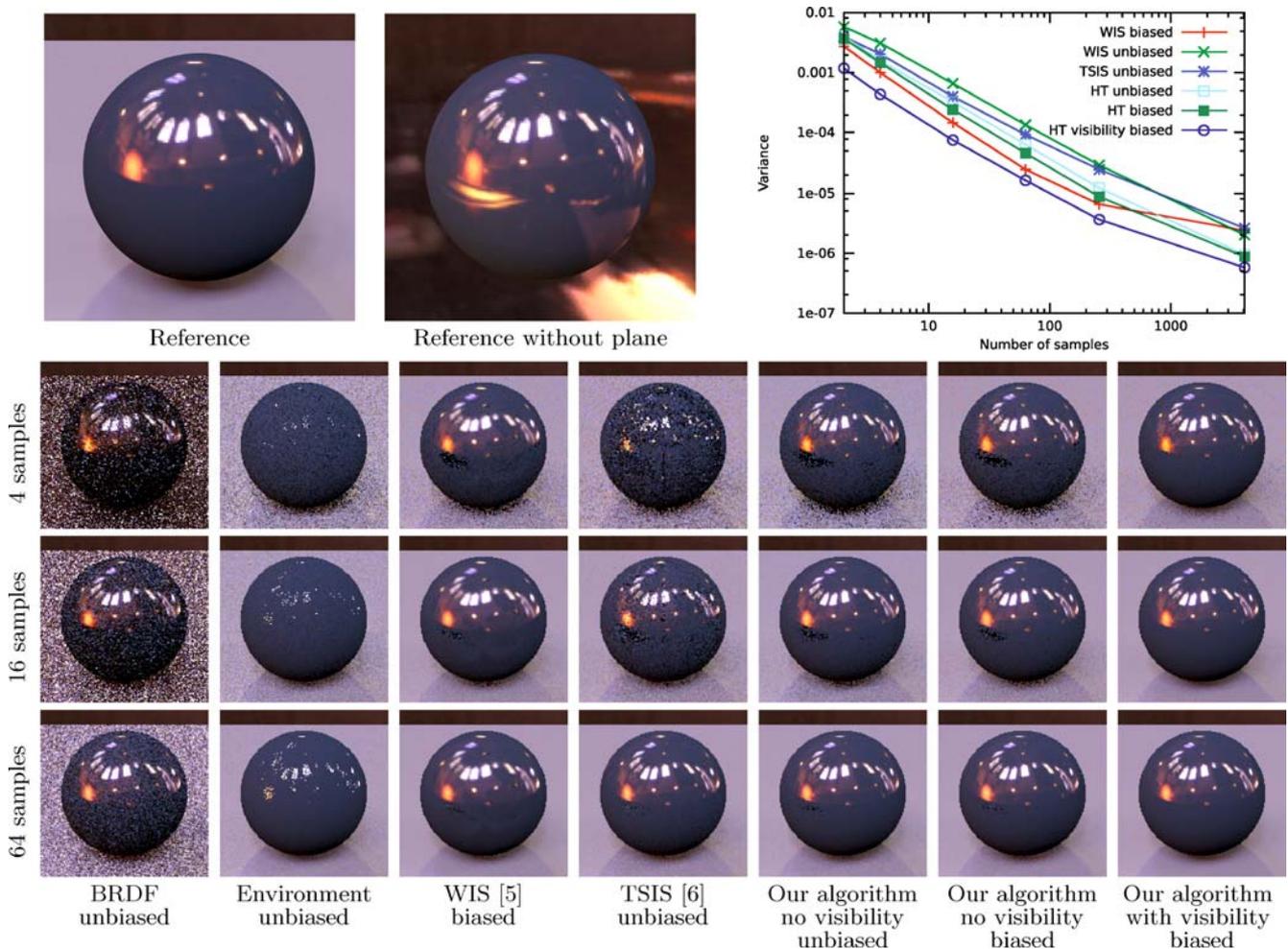
**Table 3.** Rendering times for images of Fig. 8

#Samples	4	16	64	256
HT unbiased	5.1 s	7.3 s	10.0 s	20.4 s
HT biased	2.3 s	3.2 s	4.8 s	9.9 s
HT+Vis biased	2.8 s	3.7 s	5.3 s	10.3 s
WIS [4]	6.4 s	6.7 s	7.8 s	11.3 s
Two-stage [5]	0.8 s	1.8 s	5.8 s	19.3 s

images also show that our biased extension gives strong noise reduction, while being consistently closer to the reference. The rendering times are given in Table 3.<sup>1</sup>

Figure 1 illustrates the robustness of our method and its applicability to a wide range of materials, ranging from diffuse to highly specular, in a lighting environment with an extreme dynamic range.

<sup>1</sup> Note that the timings are not directly comparable as the algorithms were implemented in different ray tracers. However, all systems show a performance similar to `pbrt` [21].



**Fig. 8.** Comparison with other recent methods. For WIS, we used wavelets of  $128^2$  resolution, 2%–5% sparsity and Poisson points as input. For TSIS, we did not use adaptive sampling. BRDF sampling was done analytically, and environment map sampling with the unbiased version of our algorithm. For our algorithm, the BRDFs for the plane and the sphere were precomputed to HEALPix levels 5 and 6, respectively, and we did not use adaptive sampling. Ground truth is shown at the *top*. Visibility sampling effectively removes noise due to occlusion. The variance was measured after tone-mapping

## 6 Discussion and future work

Our sampling scheme is based on hierarchical thresholding against an approximated importance function. The approximation is computed from hierarchical representations of the local maxima and averages, and enables several important features. First, we can sample products of multiple functions, including rotations between different domains. Second, many useful optimizations are possible, e.g., early termination and biased integration, which improve speed and reduce noise. Although our method requires the involved functions to be smooth and bounded, we have found it to be very robust.

For estimating the direct illumination, our results compare favorably to existing state-of-the-art methods. As a proof of concept, we include a visibility term estimated per

pixel, and show that it is possible to significantly reduce the noise due to occlusion. Exploiting visibility information to speed up the computation of direct illumination is an interesting direction of research. We would also like to remove some of the limitations of our algorithm, most importantly the precomputation step. One possibility could be to compute the necessary data on-the-fly directly from analytical or factorized BRDFs. This would allow spatially varying materials, which is important in a number of applications.

**Acknowledgement** We thank Nicolas Tittley for the inner spheres, as well as Tomas Akenine-Möller and Henrik Wann Jensen for their insightful comments. This work was supported by the Natural Sciences and Engineering Research Council of Canada, and the Swedish Foundation for Strategic Research. Lastly, we thank Paul Debevec who provided us with the environment maps.

## References

1. Agarwal, S., Ramamoorthi, R., Belongie, S., Jensen, H.W.: Structured importance sampling of environment maps. *ACM Trans. Graph.* **22**(3), 605–612 (2003)
2. Ashikhmin, M., Shirley, P.: An anisotropic Phong BRDF model. *J. Graph. Tools* **5**(2), 25–32 (2000)
3. Burke, D., Ghosh, A., Heidrich, W.: Bidirectional importance sampling for direct illumination. In: *Rendering Techniques 2005: Eurographics Symposium on Rendering*, pp. 147–156. Eurographics Association, Aire-la-Ville, Switzerland (2005)
4. Clarberg, P., Jarosz, W., Akenine-Möller, T., Jensen, H.W.: Wavelet importance sampling: efficiently evaluating products of complex functions. *ACM Trans. Graph.* **24**(3), 1166–1175 (2005)
5. Cline, D., Egbert, P.K., Talbot, J.F., Cardon, D.L.: Two stage importance sampling for direct lighting. In: *Rendering Techniques 2006: Eurographics Symposium on Rendering*, pp. 103–113. Eurographics Association, Aire-la-Ville, Switzerland (2006)
6. Debevec, P.: Rendering synthetic objects into real scenes: bridging traditional and image-based graphics with global illumination and high dynamic range photography. In: *Proc. ACM SIGGRAPH 98*, pp. 189–198. ACM, New York (1998)
7. Dutré, P., Bala, K., Bekaert, P.: *Advanced Global Illumination*, 2nd edn. A.K. Peters Ltd, Natick, MA (2006)
8. Friedel, I., Keller, A.: Fast generation of randomized low-discrepancy point sets. In: *Monte Carlo and Quasi-Monte Carlo Methods*, pp. 257–273. Springer, Secaucus, NJ (2000)
9. Ghosh, A., Heidrich, W.: Correlated visibility sampling for direct illumination. *Visual Comput.* **22**(9), 693–701 (2006)
10. Gorski, K.M., Hivon, E., Banday, A.J., Wandelt, B.D., Hansen, F.K., Reinecke, M., Bartelmann, M.: HEALPix – a framework for high resolution discretization, and fast analysis of data distributed on the sphere. *Astrophys. J.* **622**, 759–771 (2005)
11. Jensen, H.W., Arvo, J., Dutré, P., Keller, A., Owen, A., Pharr, M., Shirley, P.: Monte Carlo ray tracing. In: *ACM SIGGRAPH 2003 Course Notes*. ACM, New York (2003)
12. Kajiya, J.T.: The rendering equation. *Comput. Graph. (Proc. ACM SIGGRAPH 86)* **20**(4), 143–150 (1986)
13. Keller, A.: Quasi-Monte Carlo Methods for Photorealistic Image Synthesis. Ph.D. thesis, University of Kaiserslautern (1998)
14. Kollig, T., Keller, A.: Efficient illumination by high dynamic range images. In: *Rendering Techniques 2003: Eurographics Symposium on Rendering*, pp. 45–50. Eurographics Association, Aire-la-Ville, Switzerland (2003)
15. Lawrence, J., Rusinkiewicz, S., Ramamoorthi, R.: Efficient BRDF importance sampling using a factored representation. *ACM Trans. Graph.* **23**(3), 496–505 (2004)
16. Matusik, W., Pfister, H., Brand, M., McMillan, L.: A data-driven reflectance model. *ACM Trans. Graph.* **22**(3), 759–769 (2003)
17. Niederreiter, H.: *Random Number Generation and Quasi-Monte Carlo Methods*. Society for Industrial and Applied Mathematics, Philadelphia, PA (1992)
18. Ostromoukhov, V.: Sampling with polyominoes. *ACM Trans. Graph.* **26**(3), 78 (2007)
19. Ostromoukhov, V., Donohue, C., Jodoin, P.M.: Fast hierarchical importance sampling with blue noise properties. *ACM Trans. Graph.* **23**(3), 488–495 (2004)
20. Owen, A.B.: Randomly permuted  $(t, m, s)$ -nets and  $(t, s)$ -sequences. *Lect. Notes Stat.* **107**, 299–317 (1995)
21. Pharr, M., Humphreys, G.: *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann Publishers Inc., San Francisco, CA (2004)
22. Talbot, J., Cline, D., Egbert, P.: Importance resampling for global illumination. In: *Rendering Techniques 2005: Eurographics Symposium on Rendering*, pp. 139–146. Eurographics Association, Aire-la-Ville, Switzerland (2005)
23. Veach, E., Guibas, L.J.: Optimally combining sampling techniques for Monte Carlo rendering. In: *Proc. ACM SIGGRAPH 95*, pp. 419–428. ACM, New York, NY (1995)
24. Wang, R., Zhou, K., Snyder, J., Liu, X., Bao, H., Peng, Q., Guo, B.: Variational sphere set approximation for solid objects. *Visual Comput.* **22**(9), 612–621 (2006)
25. Ward, G.J.: Measuring and modeling anisotropic reflection. *Comput. Graph. (Proc. ACM SIGGRAPH 92)* **26**(2), 265–272 (1992)



FABRICE ROUSSELLE received his engineering degree from the Ecole Polytechnique de Montréal, Canada in 2000 and his M.Sc. degree from Université de Montréal, Canada in 2007. He is now a Ph.D. student at the Peripheral Systems Laboratory of the Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland. He is interested in computer graphics and in modeling the interaction of light, inks and paper.

PETRIK CLARBERG is a Ph.D. student in the Department of Computer Science at Lund University, Sweden. He received his M.Sc. in Computer Engineering from Lund University in 2005. Petrik has also been a visitor at UCSD and Université de Montréal. His research interests include photo-realistic rendering, global illumination, and Monte Carlo techniques.

LUC LEBLANC is a Ph.D. candidate in Computer Graphics at the Université de Montréal.

After completing his M.Sc. degree at the Université de Montréal, he worked for 4 years at Autodesk in Montreal. His research interests are in procedural modeling, real-time rendering, illumination, and animation.

VICTOR OSTROMOUKHOV studied mathematics, physics and computer science at Moscow Phys-Tech (MIPT). After graduating in 1980, he spent several years with prominent European and American industrial companies (SG2, Paris; Olivetti, Paris and Milan; Canon Information Systems, Cupertino, CA) as a research scientist and/or computer engineer. He completed his Ph.D. in CS at Swiss Federal Institute of Technology (EPFL, Lausanne, 1995), where he continued to work as a lecturer and senior researcher. Invited professor at University of Washington, Seattle, in 1997, and at iMAGIS/INRIA, Grenoble, France, in 2002. Research scientist at Massachusetts

Institute of Technology, Cambridge, MA, in 1999–2000. Associate Professor at University of Montreal, since August 2000. Victor's research interests are mainly in computer graphics, and more specifically in sampling theory, photorealistic and non-photorealistic rendering, color science, halftoning, and digital art. URL <http://www.iro.umontreal.ca/~ostrom/>.

PIERRE POULIN is a full professor in the Computer Science and Operations Research department of the Université de Montréal. He holds a Ph.D. from the University of British Columbia and a M.Sc. from the University of Toronto, both in Computer Science. He has served on program committees of more than 30 international conferences. His research interests cover a wide range of topics, including image synthesis, image-based modeling, procedural modeling, natural phenomena, scientific visualization, and computer animation.