Adaptive Numerical Cumulative Distribution Functions for Efficient Importance Sampling

Jason Lawrence   Szymon Rusinkiewicz   Ravi Ramamoorthi

Princeton University   Princeton University   Columbia University

Abstract

As image-based surface reflectance and illumination gain wider use in physically-based rendering systems, it is becoming more critical to provide representations that allow sampling light paths according to the distribution of energy in these high-dimensional measured functions. In this paper, we apply algorithms traditionally used for curve approximation to reduce the size of a multidimensional tabulated Cumulative Distribution Function (CDF) by one to three orders of magnitude without compromising its fidelity. These adaptive representations enable new algorithms for sampling environment maps according to the local orientation of the surface and for multiple importance sampling of image-based lighting and measured BRDFs.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism I.3.6 [Computer Graphics]: Methodology and Techniques

1. Introduction

Techniques are now common for accurately measuring real-world surface reflectance and illumination. As a result, densely sampled tabular representations of lighting and Bi-directional Reflectance Distribution Functions (BRDFs) appear in off-line, physically-based rendering pipelines. Because global illumination algorithms typically use Monte Carlo integration, employing common variance reduction techniques is critical to achieving a feasible rate of convergence. Consequently, it is important that representations of measured environment maps and BRDFs provide efficient importance sampling.

While several specific representations of measured environment maps and BRDFs do allow direct sampling, there is still no single representation that is appropriate for general multidimensional measured functions. As higher-dimensional datasets find their way into rendered scenes (e.g. light fields, reflectance fields, etc.), a general method for sampling them will become more important. Moreover, for the specific case of environment maps, existing representations do not account for the local orientation of the surface (i.e. the *cosine term* in the rendering equation). This property can limit the effectiveness of environment map sampling in reducing variance for many scenes. Another important draw-
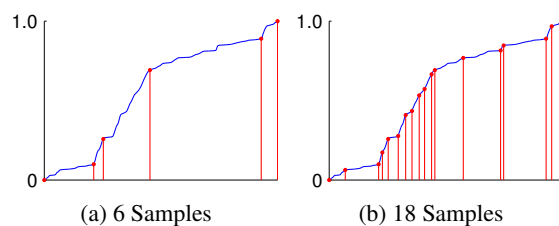

Figure 1: *We represent a 1D CDF with a set of non-uniformly spaced samples of the original function. This results in a more compact yet accurate approximation of the original function than uniform spacing would allow. In addition, the final CDF maintains many key properties necessary for unbiased multiple importance sampling.*

back of some existing environment map representations is that they are not readily incorporated into a multiple importance sampling framework [VG95].

In this paper, we apply a curve approximation algorithm to the task of compressing multidimensional tabular Cumulative Distribution Functions (CDFs) derived from measured datasets. Assume we have a 1D CDF, $P(x)$, sampled uniformly in x. In order to compress this function, we lift the restriction that the samples must be uniformly spaced, as shown in Figure 1. We use the Douglas-Peucker greedy al-

gorithm for polygonal approximation of 2D curves [DP73, HS92] to compute the location of these *adaptive* samples. We further extend this algorithm to represent multidimensional CDFs. To accomplish this, we compute marginal 1D CDFs in each dimension by summing the energy contained across the orthogonal dimensions. Each of these 1D CDFs is represented by non-uniformly spaced samples and the resulting set of these "cascading CDFs" approximates the original high-dimensional distribution. There are several benefits of using this adaptive numerical representation:

- Allowing placement of non-uniformly spaced samples reduces the number that must be stored to accurately represent the original CDF. This is especially true for multidimensional distributions because the storage grows exponentially with the number of dimensions. Significant reduction is also achieved for common "peaky" distributions, for which many methods require $O(n)$ storage.
- Generating directions according to the distribution, accomplished using numerical inversion of the CDF, simply requires a binary search over the sorted samples of $P(x)$. This is essentially the same algorithm as is used for uniformly sampled CDFs, but with the position of each sample along the domain stored explicitly.
- Storing a "cascading set" of conditional 1D CDFs, each represented by non-uniformly spaced samples of the original functions, promotes a direct implementation of unbiased stratified importance sampling. This results from the fact that each dimension can be sampled independently.
- The probability of a sample not drawn from the CDF itself can be efficiently computed from the final representation. This property is critical for combining distributions with standard multiple importance sampling algorithms.

To demonstrate the benefit of our adaptive representation, we present a novel algorithm for sampling measured environment maps in an orientation-dependent manner. This is accomplished by sampling the 4D function that results from modulating an environment map with the horizon-clipped *cosine term* in the rendering equation. This algorithm is more efficient than existing techniques that sample only a single spherical distribution. Lastly, we show how our adaptive representation can be used within a multiple importance sampling framework.

2. Related Work

Monte Carlo importance sampling has a long history in Computer Graphics [Vea97]. For stratified sampling from 2D CDFs on a manifold (in the practical examples of this paper, the manifold is a sphere or hemisphere), Arvo [Arv01] describes a particular recipe when an analytic description of the function is available, with analytic sampling strategies available in some cases [Arv95]. When dealing with measured illumination or reflectance data, as in this paper, non-parametric or "numerical" CDFs are unavoidable, and it is important to compress them.

One possible approach is to use general function compression methods, such as wavelets or Gaussian mixture models. Wavelets have been previously used for importance sampling of BRDFs [CPB03,LF97]. However, the computational cost for generating a sample can be significant, especially if non-Haar wavelets are used (as is necessary to avoid many kinds of blocking artifacts). Additionally, the implementation for multidimensional functions such as measured illumination and BRDFs can be difficult, requiring sparse wavelet data structures and a hexa-decary search.

A second approach to compact CDF representation that has been applied for BRDFs is factorization [LRR04]. This method takes advantage of the structure of the BRDF to factor it into 1D and 2D pieces, thereby reducing dimensionality while still allowing for accurate representation and efficient importance sampling. The technique proposed here differs in considering compression of general tabulated CDFs, and is not limited to BRDF sampling. Additionally, the CDF compression considered here is independent of dimension and orthogonal to any factorizations of the input data. It could therefore be applied equally well to methods that use a full tabular BRDF representation and sampling scheme (as shown in this paper), or to lower dimensional components.

Another specialized CDF compression approach, which has been applied to environment maps, is to decompose the function into piece-wise constant Voronoi or Penrose regions on the sphere [ARBJ03,KK03,ODJ04]. As compared to our method, these techniques offer more optimal stratification, but do not directly extend to multidimensional distributions. Another drawback of these representations is that they are difficult to use with standard multiple importance sampling algorithms that require computing the probability of a direction generated from a separate distribution. Lastly, these representations ignore the fact that half of the environment is always clipped against the horizon of the surface and that the illumination is scaled by a cosine term (Figure 10).

3. Background

In this paper we address the problem of importance sampling. That is, we seek to generate samples according to some Probability Density Function (PDF), p, which by definition is non-negative and normalized (i.e., it integrates to 1). We accomplish this using the *inversion method*, which precomputes the corresponding Cumulative Distribution Function (CDF)

$$P(x) = \int_{-\infty}^{x} dx'\, p(x') \tag{1}$$

and evaluates its inverse $P^{-1}(\zeta)$ at locations given by a uniformly distributed random variable $\zeta \in [0,1]$.

We are interested in the case of a PDF specified numerically: in 1D, we assume that we are given probabilities p_i at locations x_i. We precompute the corresponding CDF val-
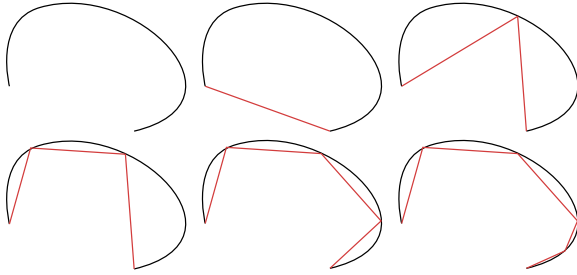
Figure 2: *The Douglas-Peucker algorithm greedily computes a polyline approximation of a smooth 2D curve. It works by inserting the next sample in the approximation at the point of maximum deviation between the (black) original curve and the (red) current polyline approximation.*

ues P_i and, at run-time, invert the CDF by performing a binary search for the interval $[P_i, P_{i+1}]$ that contains the random value ζ. Note that this search is required whether or not the x_i are spaced uniformly. This will be the key property used by our representation: we can represent many functions more efficiently by having non-uniformly spaced x_i without increasing the run-time cost of importance sampling.

In 2D, the situation is more complex. We must first decompose the 2D PDF $p(x, y)$ into two pieces, one dependent only on x and the other on y:

$$\tilde{p}(x) \;=\; \int_{-\infty}^{\infty} dy \, p(x, y) \qquad (2)$$

$$p(y|x) \;=\; \frac{p(x, y)}{\tilde{p}(x)} \qquad (3)$$

The numerical representation then consists of a discretized version of $\tilde{p}$, given as samples $\tilde{p}_i$ at locations x_i, together with a *collection* of discretized conditional probability functions $p_i(y|x_i)$. This technique generalizes naturally to any number of dimensions, producing a "cascading set" of CDFs where a value in each dimension is generated sequentially using the appropriate 1D marginal CDF at each step [Sch94]. As an important special case, we note that functions on a sphere may be represented using the parameterization $p(z, \phi)$, where the usual change of variables $z = \cos\theta$ is used to normalize for the area measure $d\omega = \sin\theta \, d\theta \, d\phi$.

If the CDFs are uniformly sampled along their domain, the total size of this set of CDFs will be slightly larger than the size of the original function. In Computer Graphics, it is often the case that these functions can be both high-dimensional and measured at high resolutions. Consequently, the combined size of the resulting 1D CDFs can quickly become prohibitively large. This motivates our investigation into efficient techniques for compressing these sampled functions without compromising their accuracy or utility.

4. Numerical CDF Compression

We use polygonal curve approximation algorithms to compress a densely sampled CDF by representing it with a reduced set of non-uniformly spaced samples selected to minimize the reconstruction error.

4.1. Polygonal Curve Approximation

With early roots in cartography, several efficient algorithms have been developed for computing polygonal approximations of digitized curves. Polygonal approximation algorithms take as input a curve represented by an N-segment polyline and produce an M-segment polyline with vertices so as to minimize the error between the two (typically $M \ll N$).

Although algorithms exist that output the optimal solution [CC96, Goo94, CD03], we instead use the Douglas-Peucker [DP73, HS92] greedy algorithm because of its simplicity and speed. It has also been shown that these greedy algorithms typically produce results within 80% accuracy of the optimal solution [Ros97].

The Douglas-Peucker curve approximation algorithm works by iteratively selecting the vertex furthest from the current polyline as the next vertex to insert into the approximation (Figure 2). Initially only the endpoints of the curve are selected, and the algorithm iteratively updates this approximation until either an error threshold is reached or some maximum number of vertices have been used. For curves derived from numerical CDFs, we found this algorithm sufficient for producing near-optimal approximations with few samples.

4.2. Applying Curve Approximation to CDFs

There are several ways of applying the above curve approximation algorithms to the task of representing numerical probability functions. First, we can apply them to yield a piecewise linear approximation of the CDF, which is equivalent to a *piecewise constant* approximation of the corresponding PDF. Because the Douglas-Peucker algorithm, when applied to the CDF, is guaranteed to yield a nondecreasing function with a range of $[0..1]$, the resulting approximation may be used directly as a CDF and differentiated to find the corresponding PDF.

A second way of using curve approximation algorithms is to apply them directly to the PDF to obtain a *piecewise linear* approximation (which implies a piecewise quadratic CDF). In this case, the resulting approximation is not guaranteed to integrate to one, and must be normalized before it can be used as a probability function. Figure 3, bottom, compares these two strategies on a (relatively smooth) function: note that the two approaches result in samples being placed at different locations in the domain. For comparison, Figure 3, top, shows piecewise constant and piecewise linear approximations using uniform sample spacing.
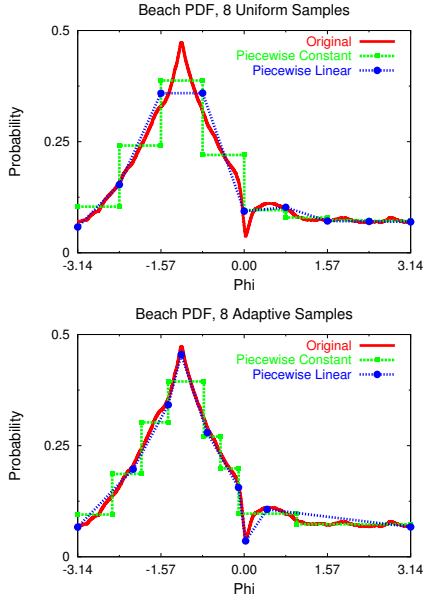
Figure 3: *A probability density function (corresponding to environment map in Figure 7) and its piecewise linear and piecewise constant approximations with 8 samples placed uniformly (top) and computed by the Douglas-Peucker algorithm (bottom). The piecewise constant approximation was computed by running Douglas-Peucker on the integral of the PDF (i.e. the CDF). Note that, for this relatively smooth function, the piecewise linear approximation is closer to the original.*

One important difference between uniformly-sampled and adaptively-sampled CDFs is the cost of reconstructing the value of the approximated function (i.e., evaluating the probability) at an arbitrary position. This property is necessary for combining several distributions using multiple importance sampling algorithms [Vea97]. When the samples are uniformly spaced the cost is $O(1)$, whereas adaptively sampled representations require $O(\log N)$ time (here N refers to the number of non-uniform samples). This increased complexity results from having to perform a binary search over the values of the function sorted along the domain to find the desired interval. Because adaptive representations provide such large compression rates, however, N is typically small enough to make this added cost insignificant in practice. In addition, the time complexity of *generating* a sample (as opposed to evaluating the probability) remains the same at $O(\log N)$ in both cases.

In our experiments, we always used a piecewise constant approximation of the PDF (i.e. piecewise linear CDF). Although this results in a slightly larger representation, in our experience this drawback was outweighed by the simpler implementation required for sampling a piecewise constant approximation.

5. Multidimensional CDFs: The Cascading Douglas-Peucker Algorithm

In the previous section, we discussed how to apply curve approximation algorithms to the task of efficiently representing numerical 1D CDFs. In this section, we extend these ideas to accomodate distributions of higher dimension. For the sake of explanation, we first restrict our discussion to the 2D case and provide an example with synthetic data in Figure 4. Extending these techniques to higher dimensions is straightforward and briefly discussed at the end of the section.

Recall that we can convert any 2D distribution (Figure 4 top) into a single marginal CDF plus a set of conditional CDFs according to Equations 1, 2 and 3. In order to generate (x, y) pairs with probability proportional to the magnitude of the original function, we first generate a value of x from the marginal CDF $\tilde{P}(x)$ (Figure 4 bottom, red curve) and then generate a value of y from the corresponding conditional CDF $P(y|x)$ (not shown in Figure 4).

As described previously, we use the Douglas-Peucker algorithm to select a set of non-uniformly spaced samples that accurately represent the marginal 1D CDF, $\tilde{P}(x)$. For the example in Figure 4, we can perfectly approximate the marginal CDF with samples at the endpoints A and E and at internal locations B and D. Next, we would compute a set of conditional CDFs, $P(y|x)$; one for each of these regions in x (e.g. in Figure 4 these regions are AB, $\boldsymbol{BD}$ and DE). Each conditional CDF is the average across its associated range:

$$p(y|x_i) = \frac{1}{x_i - x_{i-1}} \int_{x_{i-1}}^{x_i} dx' \, \frac{p(x', y)}{\tilde{p}(x')}. \qquad (4)$$

For all the examples in this paper on measured data, building a cascading set of CDFs according to Equation 4 was sufficient for accurately approximating the original distribution. However, there are potential situations where this approach alone ignores error introduced by approximating the distribution of energy within a region with a single CDF. Figure 4 illustrates such a situation. In this case, the distribution of energy within the region BD would be poorly approximated by a single conditional distribution because the two area light sources are at different heights. In order to address this issue, we must also consider the gradient in the x-direction of the original distribution:

$$g(x) = \int_{-\infty}^{\infty} dy \left| \frac{\partial p(x, y)}{\partial x} \right| \qquad (5)$$

When the function $g(x)$ is large this indicates locations in x where the conditional CDFs, $P(y|x)$, would *not* be well approximated by a single distribution. Therefore, after our first application of the Douglas-Peucker algorithm to represent $\tilde{P}(x)$, we add additional samples according to this gradient function. Specifically, we can compute a numerical CDF from $g(x)$ and generate a fixed number of stratified samples along the domain (e.g. the x-axis) such that they occur at locations where this function is large. Adding samples ac-
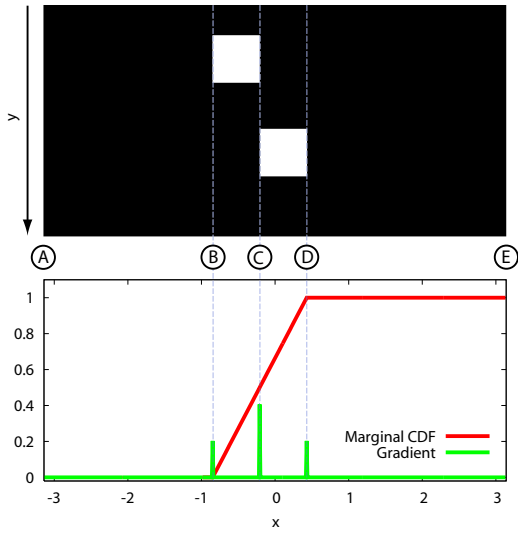
Figure 4: *Efficiently approximating multi-dimensional distributions requires computing a cascading set of 1D marginal and conditional CDFs. Here we show (top) a synthetic environment map that contains only two equal sized area light sources. We compute (bottom, red curve) a marginal CDF in x by summing the total energy across y. We also consider (bottom, green curve) the average gradient in the x-direction. We place non-uniformly spaced samples according to the Douglas-Peucker algorithm at positions A, B, D and E and any additional points where the gradient function is large (i.e. at position C).*

cording to the gradient guarantees that both $\tilde{P}(x)$ is well represented by the non-uniformly spaced samples and that the conditional CDFs computed for each region, $P(y|x_i)$, well approximate the variation present in the orthogonal dimensions. In the example in Figure 4, we additionally sample the marginal CDF at location *C*, separating the 2D distribution into a total of four regions (*AB*, *BC*, *CD* and *DE*), where each region is now well approximated by a single CDF.

Lastly, we extend this sampling algorithm to arbitrary dimensions by simply expanding the integrals over the entire range of free variables (as opposed to just *y* for the 2D example considered above). For an N-dimensional distribution, $p(x_1, x_2, \ldots x_N)$, both the marginal and conditional CDFs are proportional to the integral across the remaining free variables (note: we omit the normalization constant for clarity):

$$p(x_i|x_1 \ldots x_{i-1}) \propto \int_{-\infty}^{\infty} dx_{i+1} \ldots \int_{-\infty}^{\infty} dx_N \, p(x_1 \ldots x_N),$$

and the gradient function would be computed similarly:

$$g(x_i|x_1 \ldots x_{i-1}) = \int_{-\infty}^{\infty} dx_{i+1} \ldots \int_{-\infty}^{\infty} dx_N \left| \frac{\partial p(x_1 \ldots x_N)}{\partial x_i} \right|.$$

6. Evaluation of Algorithm

In general, global illumination algorithms perform numerical integration of the rendering equation:

$$L_o(x, \omega_o) = L_e(x, \omega_o) + \int_{\Omega_{2\pi}} d\omega_i \, L_i(x, \omega_i) \, \rho(x, \omega_i, \omega_o) \, (\omega_i \cdot \mathbf{n}).$$

A common approach to estimating the value of this integral is to perform Monte Carlo integration over the space of incoming directions. Because the entire integrand is usually not known *a priori*, a reasonable strategy is to sample according to the terms that are known. For example, if the incident illumination L_i is represented by an environment map, we may perform environment sampling. BRDF sampling, on the other hand, generates samples according to either ρ itself or $\rho \cdot (\omega_i \cdot \mathbf{n})$. Although algorithms exist for sampling BRDFs and environment maps, these functions provide a convenient platform to evaluate our representation. Moreover, our approach has several desirable properties that these existing techniques lack. These enable novel applications that we present in Section 7.

6.1. Environment Map Sampling

One direct approach for generating samples according to a measured environmet map [Deb98], is simply to compute a family of numerical 1D CDFs directly from the 2D spherical function [PH04]. Recall that one CDF will quantify the distribution along ϕ, $\tilde{P}(\phi)$ and a set of 1D CDFs will control the distribution of samples along θ at each sampled location of ϕ, $P_i(\theta|\phi_i)$. These are derived from the *intensity* of each pixel in the environment map (i.e. weighted average of color values) using the method described in Section 3.

If the resolution of these CDFs is proportional to that of the environment map (as it should be to avoid aliasing) this representation will be slightly larger then the original measured dataset itself. Therefore, there is significant opportunity for compression using our adaptive representation. Figure 5 shows false-color visualizations on a logarithmic scale of the full-resolution 1000×1000 ($\theta \times \phi$) PDF of the Grace Cathedral environment (http://www.debevec.org/Probes/), together with 16×16 and 64×64 approximations using both uniform and non-uniform sample selection. As compared to uniform sampling, adaptive sample placement results in a significantly more accurate approximation of the original distribution.

Figure 6 compares the error of our adaptive numerical representation with uniform sample placement on two distributions with qualitatively different behaviors. The upper graphs show a single scanline (i.e. varying *phi* for a constant *theta*) of the environment map, while the graphs at bottom plot the RMS error of the approximation as a function of the number of samples used (note that the horizontal axis is logarithmic). At left, we consider a relatively smooth function. In this case, the gain from nonuniform placement of samples
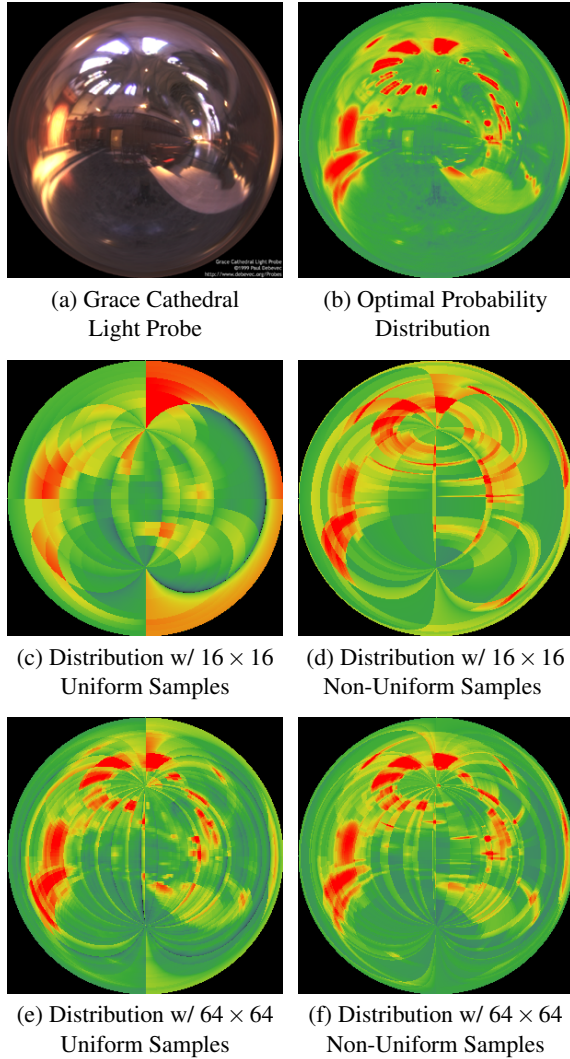
(a) Grace Cathedral
Light Probe

(b) Optimal Probability
Distribution

(c) Distribution w/ 16×16
Uniform Samples

(d) Distribution w/ 16×16
Non-Uniform Samples

(e) Distribution w/ 64×64
Uniform Samples

(f) Distribution w/ 64×64
Non-Uniform Samples

Figure 5: *False-color visualizations of spherical probability density functions on a logarithmic scale (red = largest probability, green = smallest probability). Directions are mapped to the unit circle according to the parameterization used by Debevec [Deb98]. (a) A measured environment map of the inside of Grace Cathedral. (b) The probability density resulting from using a numerically tabulated CDF sampled uniformly at the same resolution of the original map. The probability distribution of numerical CDFs computed from (c) 16×16 uniform samples (d) 16×16 non-uniform samples (e) 64×64 uniform samples and (f) 64×64 non-uniform samples.*

is relatively modest. At right, we show a "peakier" function that is easier to compress with nonuniform sample placement. In this example, our adaptive representation reduces the number of samples required at equal approximation error by a factor of 16 compared to uniform downsampling.
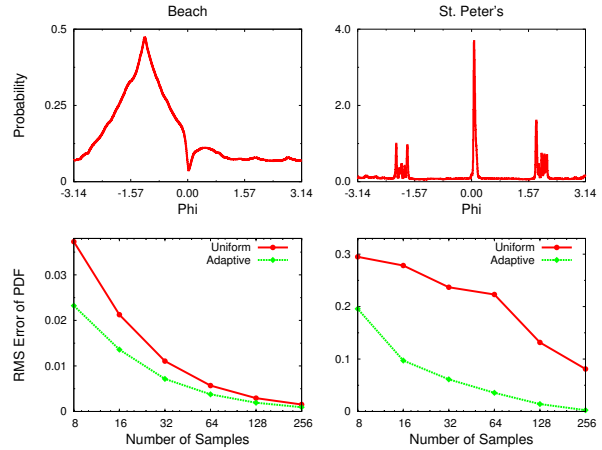

Figure 6: *Two different probability distribution functions and the RMS error in approximating them using different numbers of points and different sampling strategies. The different sampling algorithms use either uniform or adaptive placement of sample locations.*

6.2. BRDF Sampling

The BRDF gives the ratio of reflected light to incident light for every pair of incoming and outgoing directions: $\rho(\omega_o, \omega_i)$. For glossy materials, it is advantageous to sample the environment according to the distribution of energy in the BRDF. Because this is a 4D function (3D if the BRDF is isotropic), a tabular representation at a modest resolution would still be quite large. Consequently, we apply our adaptive representation to the task of efficiently storing numerical CDFs derived from measured BRDFs.

We compared the size and accuracy of this representation with a standard approach of pre-computing the CDFs at their full resolution [Mat03] for the same set of viewing directions (Figure 7). We evaluated the efficiency of generating samples using an adaptive numerical CDF computed from two measured BRDFs [MPBM03]: *nickel* and *metallic-blue*.

For these results, we first reparameterized the BRDF into a view/half-angle frame in order to maximize the redundancy among *slices* of the function giving greater opportunity for compression [LRR04]. Each uniformly-sampled CDF had a resolution of $32 \times 16 \times 256 \times 32$ ($\theta_o \times \phi_o \times \theta_h \times \phi_h$) and occupied 65MB. Here, θ_h and ϕ_h are the elevation and azimuthal angles of the half-angle vector respectively. To compute the corresponding adaptive numerical CDFs required, on average, roughly 30 samples in θ_h and 10 samples in ϕ_h. Using the Douglas-Peucker algorithm, these adaptive samples were selected from an initial set of 2048×1024 ($\theta_h \times \phi_h$) uniformly-spaced samples—a resolution prohibitively expensive for the fully tabulated CDFs. It required 20 minutes of processing time to compute the adaptive representation for each BRDF.
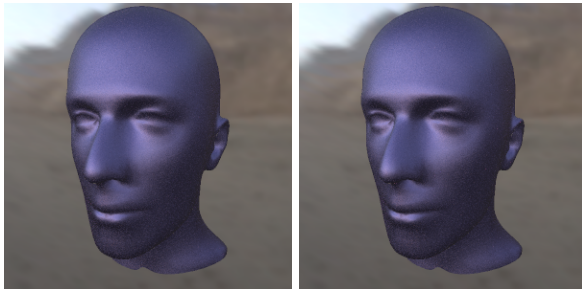
Measured Nickel BRDF

Original (65MB)   Compressed (3.9MB)

Measured Metallic-Blue BRDF

Original (65MB)   Compressed (2.3MB)

Figure 7: *BRDF importance sampling with adaptive numerical CDFs. We compare the variance in images rendered using a path tracer that generates samples using the fully tabulated CDF and the adaptive CDF. In all cases we estimate the radiance with 80 paths/pixel. We also list the total size of the probability representation below each image.*

We found that for these BRDFs, sampling the adaptive numerical CDF is nearly as efficient as the full tabular approach. For the measured nickel BRDF, the compact CDF actually produces slightly less variance in the image because the uniform sampling was not sufficiently dense to capture the very sharp highlight.

7. Novel Applications

In this section we present a new algorithm for sampling illumination from an environment map according to the local orientation of the surface. Additionally, we demonstrate how our representation facilitates multiple importance sampling of both illumination and the BRDF.

7.1. Local Environment Map Sampling

Using adaptive numerical CDFs, we introduce a novel algorithm for sampling an environment map in an orientation-dependent manner. In previous methods of sampling environment maps, incoming directions are drawn from a single
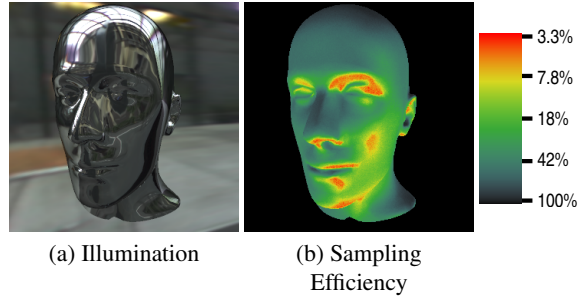
(a) Illumination   (b) Sampling Efficiency

Figure 8: *For some orientations and lighting, sampling from a single distribution will be inefficient because most of the energy is occluded by the horizon. (a) We examine this inefficiency for an example in which the majority of light is above and slightly behind the object being rendered. (b) A false-color image visualizes the percentage of samples that will be generated above the horizon and, consequently, make a positive contribution to the radiance estimate at that pixel. In many regions of this image only 5% of the samples are generated above the horizon.*

spherical distribution [ARBJ03, KK03, ODJ04, PH04]. This approach is inefficient when a significant amount of light in the scene happens to fall below the horizon for a large number of pixels. In Figure 8, there are many regions of the image where as few as 5% of the samples are generated above the horizon—this also indicates the inefficacy of standard techniques like rejection sampling to address this problem. Furthermore, sampling from a single spherical distribution cannot consider the *cosine term* that appears in the rendering equation (i.e. $\max(0, \mathbf{n} \cdot \omega_i)$). Accounting for this cosine-falloff would require sampling from a 4D function (i.e. there are two degrees of freedom in the incoming direction and two in the normal direction). We show several 2D slices of this function for different normal directions in Figure 9. As with BRDFs, representing a 4D distribution even at a modest resolution could require prohibitively large storage.

We can store the 4D distribution that results from modulating an environment map by the cosine term using our adaptive CDF representation. During rendering, each pixel corresponds to a normal direction that becomes an index into the 4D distribution, producing a 2D distribution over incoming directions that we sample from. In our experiments, we evaluated the *local* environment map distribution at 25×10 ($\phi \times \theta$) normal directions and 1000×2000 ($\phi \times \theta$) incoming directions. Storing this tabular CDF directly would require *approx.* 4GB of space. In contrast, our representation requires 10-20MB of storage and 1-2 hours of compute time to provide an acceptable approximation.

We compared *local environment map sampling* with jittered sampling of a stratified representation [ARBJ03] and sampling from a uniformly-spaced CDF [PH04] (see Figure 10). Jittered sampling (Figure 10 left) performed the
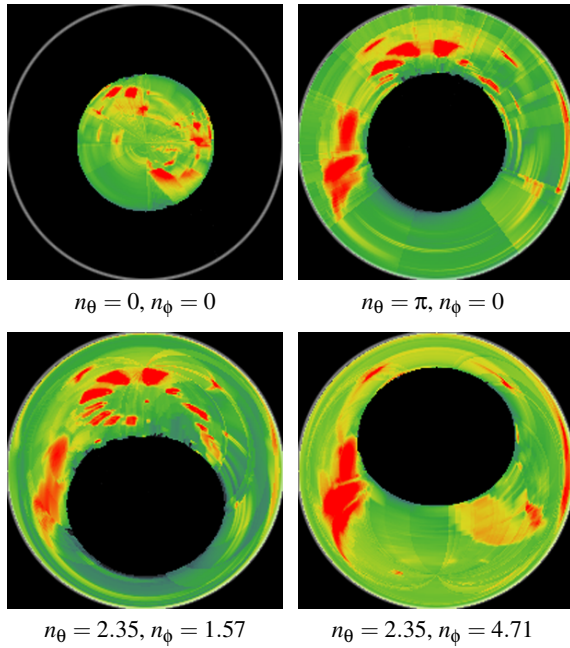
$n_\theta = 0, n_\phi = 0$ $n_\theta = \pi, n_\phi = 0$

$n_\theta = 2.35, n_\phi = 1.57$ $n_\theta = 2.35, n_\phi = 4.71$

Figure 9: *False-color visualizations of several CDFs computed at different surface orientations. Each distribution is visualized on a logarithmic scale as in Figure 5. For each surface normal considered we clip the environment map to the visible hemisphere and multiply each radiance value by $(\mathbf{n} \cdot \omega_i)$ before computing an adaptive CDF representation of the resulting 4D distribution.*

St. Peter's Basilica

Galileo's Tomb

Grace Cathedral

Jittered   Full CDF   Local
Sampling   With Rejection   Env. Sampling

Figure 10: *We compare the variance of a Monte Carlo estimator computed according to (left column) jittered sampling of a stratified representation [ARBJ03], (middle column) a uniformly-sampled CDF [PH04] where we reject samples that fall below the horizon and (right column) using our local environment map sampling algorithm. We have rendered a perfectly diffuse object at 20 samples/pixel in three different environments that all exhibit high-frequency lighting. Cutouts include a magnified region of the image and a variance image (note: these are false-color visualizations of the logarithm of RMS error in image intensity where black ≤ 0.135 and red ≥ 20.08). All three sampling methods required approximately 15 seconds to render these images.*

worst mainly because this technique is ineffective for such low sample counts (note: we are using only 20 samples here). Moreover, there is significant error due to the bias introduced by approximating each strata with a radial disk. Although unbiased jittering is not impossible to achieve, it is not a simple extension to published algorithms and has not been reported in previous work. We also compared our algorithm to sampling from a uniformly-sampled CDF and rejecting those samples that fell below the horizon [PH04] (Figure 10 middle). This strategy is most comparable in quality to our own, but because it does not account for the horizon-clipped cosine term in the rendering equation, it fails to achieve the same rate of convergence. Quantitatively, *local environment map sampling* achieved *approx.* 5 times lower variance than sampling a single CDF computed at full resolution and *approx.* 20 times better than jittered sampling for these test scenes.

7.2. Multiple Importance Sampling

In practice, neither the BRDF nor the incident illumination alone determine the final shape of the integrand in the rendering equation. Therefore, it is critical that a CDF representation supports multiple importance sampling [VG95].
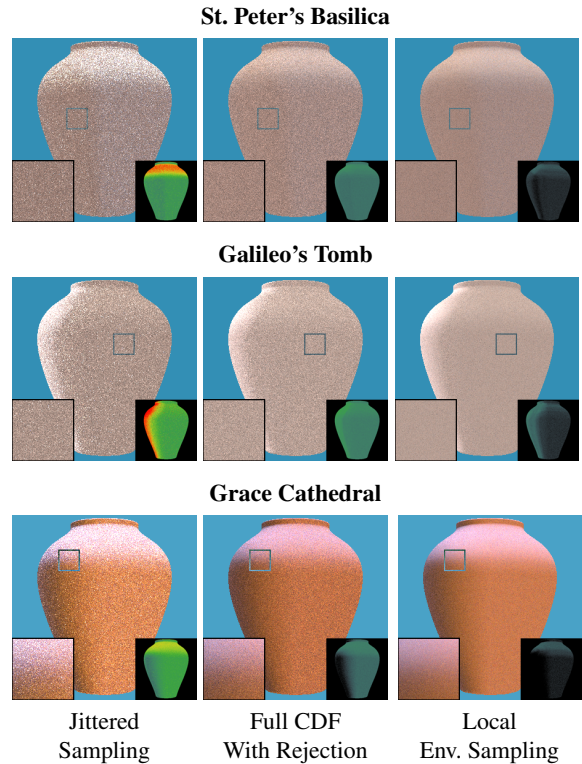
The main criterion this imposes is that the representation must allow efficient computation of the probability of a direction that was not generated from the distribution itself. Algorithms that decompose environment maps into non-overlapping strata [ARBJ03, KK03, ODJ04], for example, do not readily provide this property because determining the probability of an arbitrary direction would require searching over the strata. Although not impossible, making this search efficient has not previously been demonstrated and could be one direction of future work. With our adaptive numerical CDF, however, the probability of an arbitrary direction can be computed in $O(\log N)$ where N is the number of non-
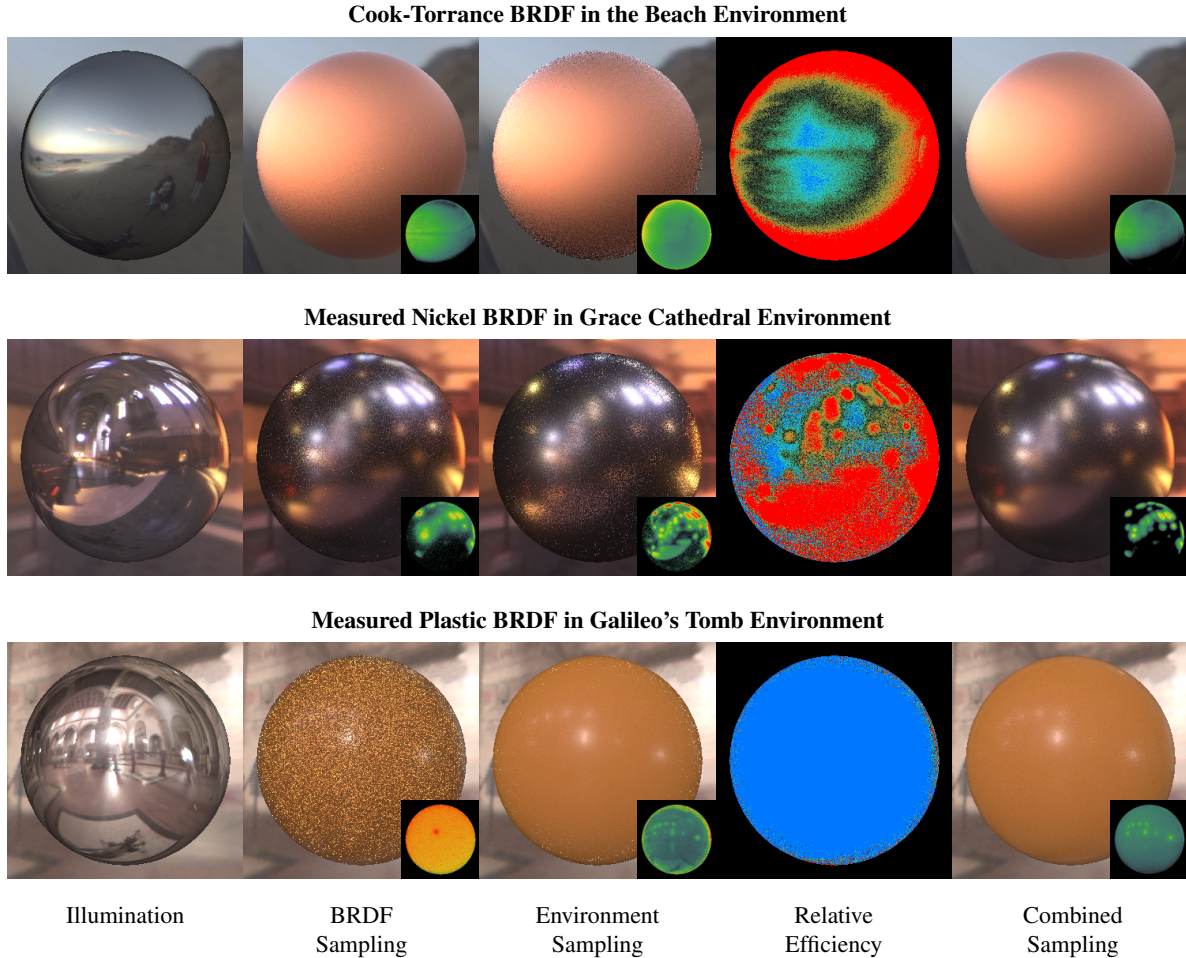
Cook-Torrance BRDF in the Beach Environment

Measured Nickel BRDF in Grace Cathedral Environment

Measured Plastic BRDF in Galileo's Tomb Environment

Illumination	BRDF Sampling	Environment Sampling	Relative Efficiency	Combined Sampling

Figure 11: *Multiple importance sampling using adaptive numerical CDFs computed from both BRDFs and image-based lighting. The 4th column visualizes the relative reduction in variance using environment sampling vs. BRDF sampling: red = BRDF sampling has 8x less variance than environment map sampling, blue = environment sampling has 8x less variance than BRDF sampling. For these scenes, sampling from either the BRDF or environment alone will not effectively reduce variance over the entire image and performing multiple importance sampling is critical.*

uniformly spaced samples. Moreover, because of the compression ratios possible with our representation, N is typically small enough to make this operation inexpensive in practice.

We show several scenes for which multiple importance sampling is critical (Figure 11). In these results, we use the *balance heuristic* introduced by [VG95] to combine 50 samples of the BRDF with 50 samples of the environment. The BRDF samples are generated using our adaptive CDF discussed in Section 6.2 and the illumination samples are generated using *local environment map sampling* (see Section 7.1). To demonstrate the benefit of a representation that supports multiple importance sampling, we also compare these images to those rendered using 100 samples drawn from either the BRDF or environment alone.

8. Conclusions and Future Work

We have applied traditional curve approximation algorithms to compress the size of numerically tabulated Cumulative Distribution Functions (CDFs) for efficient importance sampling. This representation results in a drastic reduction in the storage cost of a numerical CDF without sacrificing significant accuracy in the reconstructed Probability Density Function (PDF). We investigated the benefit of using adaptive numerical CDFs to sample image-based lighting and measured BRDFs. We also introduced *local environment map sampling*, which accounts for the orientation dependence of the illumination. Lastly, we have demonstrated multiple importance sampling using adaptive numerical CDFs to represent distributions for both the BRDFs and environment map in the scene.

Because of the generality of our adaptive representation, it has potential applications in many other problems that rely on sampling from tabulated measured data. For example, the technique might be used to represent light fields, which would then be used to illuminate a scene or to sample from the full product of a cosine-weighted environment map with a measured BRDF. A second class of applications might involve synthesis of textures, video, or animation based on probability distributions learned by example.

9. Acknowledgements

References

[ARBJ03] AGARWAL S., RAMAMOORTHI R., BELONGIE S., JENSEN H. W.: Structured importance sampling of environment maps. In *SIGGRAPH 03* (2003), pp. 605–612. 2, 7, 8

[Arv95] ARVO J.: Stratified sampling of spherical triangles. In *SIGGRAPH 95* (1995), pp. 437–438. 2

[Arv01] ARVO J.: Stratified sampling of 2-manifolds. In *SIGGRAPH 2001 Course Notes, volume 29* (2001). 2

[CC96] CHAN W., CHIN F.: On approximation of polygonal curves with minimum number of line segments or minimum error. *Int. J. Comput. Geom. Appl. 6* (1996), 59–77. 3

[CD03] CHEN D. Z., DAESCU O.: Space-efficient algorithms for approximating polygonal curves in two-dimensional space. *International Journal of Computational Geometry & Applications 13*, 2 (2003), 95—111. 3

[CPB03] CLAUSTRES L., PAULIN M., BOUCHER Y.: Brdf measurement modeling using wavelets for efficient path tracing. *Computer Graphics Forum 12*, 4 (2003), 1–16. 2

[Deb98] DEBEVEC P.: Rendering synthetic objects into real scenes: bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques* (1998), ACM Press, pp. 189–198. 5, 6

[DP73] DOUGLAS D., PEUCKER T.: Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *The Canadian Cartographer 10*, 2 (1973), 112–122. 2, 3

[Goo94] GOODRICH M. T.: Efficient piecewise-linear function approximation using the uniform metric: (preliminary version). In *Proceedings of the tenth annual symposium on Computational geometry* (1994), ACM Press, pp. 322–331. 3

[HS92] HERSHBERGER J., SNOEYINK J.: Speeding up the douglas-peucker line-simplification algorithm. In *Proceedings of the 5th International Symposium on Spatial Data Handling* (Charleston, South Carolina, 1992), vol. 1, pp. 134–143. 2, 3

[KK03] KOLLIG T., KELLER A.: Efficient illumination by high dynamic range images. In *Eurographics Symposium on Rendering 03* (2003), pp. 45–51. 2, 7, 8

[LF97] LALONDE P., FOURNIER A.: Generating reflected directions from brdf data. *Computer Graphics Forum 16*, 3 (1997), 293–300. 2

[LRR04] LAWRENCE J., RUSINKIEWICZ S., RAMAMOORTHI R.: Efficient BRDF importance sampling using a factored representation. *ACM Transactions on Graphics (ACM SIGGRAPH 2004) 23*, 3 (2004). 2, 6

[Mat03] MATUSIK W.: *A Data-Driven Reflectance Model*. PhD thesis, Massachussetts Institute of Technology, 2003. 6

[MPBM03] MATUSIK W., PFISTER H., BRAND M., MCMILLAN L.: A data-driven reflectance model. In *SIGGRAPH 03* (2003), pp. 759–769. 6

[ODJ04] OSTROMOUKHOV V., DONOHUE C., JODOIN P.-M.: Fast hierarchical importance sampling with blue noise properties. *ACM Transactions on Graphics 23*, 3 (2004), 488–495. Proc. SIGGRAPH 2004. 2, 7, 8

[PH04] PHARR M., HUMPHREYS G.: *Physically Based Rendering : From Theory to Implementation*. Morgan Kaufmann, 2004. 5, 7, 8

[Ros97] ROSIN P. L.: Techniques for assessing polygonal approximations of curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence 19*, 6 (1997), 659–666. 3

[Sch94] SCHEAFFER R. L.: *Introduction to Probability and Its Applications (Statistics)*, 2nd ed. Duxbury Press, 1994. 3

[Vea97] VEACH E.: *Robust Monte Carlo Methods for Light Transport Simulation*. PhD thesis, Stanford University, 1997. 2, 4

[VG95] VEACH E., GUIBAS L.: Optimally combining sampling techniques for Monte Carlo rendering. In *SIGGRAPH 95* (1995), pp. 419–428. 1, 8, 9