Monte Carlo Rendering with Natural Illumination

Category: Research

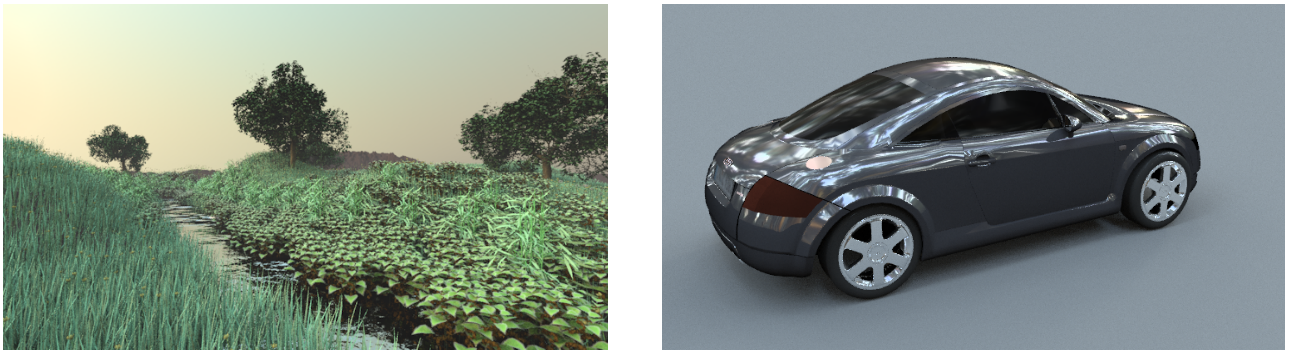

Figure 1: Two scenes rendered with our technique using environment maps for illumination with 4 pixel samples and 16 light samples per pixel sample. Preprocessing the environment map before rendering required under 0.03 seconds on a modern CPU and a negligible fraction of rendering time was spent selecting light samples.

We present a new sampling technique for using environment maps for illumination in a Monte Carlo ray tracer, based on directly importance sampling the continuous two-dimensional distribution of illumination as a function of direction. Unlike previous techniques for high-quality rendering with environment maps, our method requires very little precomputation, is easy to implement, and is compatible with existing approaches for Monte Carlo variance reduction. We achieve equivalent or lower error in rendered images than previous methods and robustly handle a wider variety of reflection models for the same amount of rendering time, though our technique uses three orders of magnitude less precomputation. For interactive rendering, our sampling methods can be used to generate a set of directional light sources that accurately approximate the environment map's illumination in a few hundredths of a second.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Raytracing;

Keywords: environment maps, illumination, Monte Carlo integration, importance sampling

1 Introduction

Image-based lighting can substantially improve the realism of rendered images when used in place of idealized approximations like point and directional lights. This is especially true when the lighting imagery is captured from a real environment and accurately represents high dynamic range lighting features. Unfortunately, rendering images with this light representation is computationally expensive, since it is necessary to evaluate the reflection integral over the entire sphere of incident illumination, rather than computing a sum over a set of discrete light sources. The reflection integral is impossible to evaluate analytically when including accurate visibility computations and arbitrary geometry, incident illumination, and reflection models, so some form of numerical integration or other approximation must be used.

This paper describes a new sampling technique for Monte Carlo lighting calculations with environment map light sources. It handles general reflection models and scene geometry and efficiently computes high-quality results (see Figure 1). Our technique uses the lighting image to define a two-dimensional probability density function over the sphere of directions and directly samples directions according to this distribution. This has a number of advantages over previous techniques: it is very easy to implement, uses little additional storage, and requires approximately three orders of magnitude less precomputation time. For environments with many small bright light sources, our technique produces results that are both visually and statistically similar to previous techniques, and for environments where important illumination is arriving from a wider range of directions, our technique gives less error. In addition, our approach is unbiased and fits naturally into classic Monte Carlo integration methods, so it is possible to apply a number of effective variance reduction methods that are less easily used with previous techniques, such as adaptive sampling, low-discrepancy sampling patterns, and multiple importance sampling.

2 Background and Previous Work

Image-based lighting (IBL) dates to Blinn and Newell, who used environment maps to shade perfectly specular surfaces [Blinn and Newell 1976]. Williams and Greene described how to filter environment maps to reduce aliasing artifacts [Williams 1983; Greene 1986]. Miller and Hoffman were the first to use environment maps to illuminate non-specular objects [Miller and Hoffman 1984].

Debevec's work on capturing illumination from real-world environments has recently rekindled interest in image-based lighting [Debevec 1998]. He first used the *Radiance* rendering system [Ward 1994], applying the lighting image as a texture map onto distant geometry and using *Radiance*'s built-in Monte Carlo sampling, which has no specialized sampling methods for environment illumination. He reported that high sampling rates were necessary to compute high-quality imagery. Cohen and Debevec later developed *LightGen*, which uses the *k*-means clustering algorithm to convert environment maps into a set of directional light sources in an offline preprocess, taking a few minutes to create a hundred lights from a low-resolution environment map [Cohen and Debevec 2001]. Directional lights like these are not a suitable representation for rendering very glossy surfaces.

More recently, a number of techniques have been developed for stratifying environment maps and preintegrating the illumination within each stratum, with strata chosen to achieve higher sampling density in areas with relatively bright illumination. Kollig and Keller use Lloyd's relaxation algorithm to choose a fixed number of strata and compute reflection from a surface with a quadrature rule [Kollig and Keller 2003]. They reported preprocessing times of 20–75 seconds per environment map.

Agarwal et al. have developed a technique called structured importance sampling (SIS), where strata are chosen based on an analysis of expected variance due to illumination variation and visibility [Agarwal et al. 2003]. Their algorithm first computes a small set of nested brightness levels. Then, the total number of desired samples are divided among the levels based based on a statistical analysis and finally the deterministic Hochbaum-Shmoys algorithm is used to place the sample points within each level. Each sample point gives rise to a stratum corresponding to its Voronoi cell. Their method requires roughly 50–100 seconds of precomputation for a typical environment map.

Cohen has developed a technique based on an adaptive tessellation of the unit sphere [Cohen 2003]. For each of a set of surface normal directions, he integrates cosine-weighted radiance over each spherical triangle. Then, given a point to be lit and its surface normal, this information is used to derive a sampling method. This approach is only applicable to reflection from Lambertian surfaces, and requires approximately 20 minutes of precomputation.

There has recently been great interest in interactive IBL with complex materials [Ramamoorthi and Hanrahan 2002; Sloan et al. 2002; Ng et al. 2003]. These techniques project the lighting onto a basis function for fast evaluation of the reflection equation, seeking to find methods that are fast enough for real-time use, rather than to supporting completely general scenes and reflection models.

2.1 Monte Carlo Direct Lighting

We would like to estimate the value of the direct lighting equation

$$L_o(p,\omega_o) = \int_{\mathscr{S}^2} f(p,\omega_i \to \omega_o) V(p,\omega_i) L_d(p,\omega_i) |\cos\theta_i| \, \mathrm{d}\omega_i,$$

where $L_o(p,\omega_o)$ is the outgoing radiance at a point p in direction ω_o, $\mathscr{S}^2$ is the unit sphere, f is the bidirectional scattering distribution function (BSDF), V is a binary visibility term that is zero if the ray (p,ω_i) intersects scene geometry and one otherwise, and L_d is the incident radiance arriving along ω_i at p. Here, we consider only the direct illumination component of radiance from the environment map, and assume that other global illumination algorithms are used for multiply scattered illumination if necessary. We also make the usual assumption in IBL that L_d is a function of direction only (i.e., the light source is infinitely far away).[1]

Monte Carlo integration has been shown to be an effective technique for evaluating the direct lighting equation in graphics, handling arbitrary BSDF and light source models as well as general scene geometry. See for example Jensen et al.'s course notes for information about Monte Carlo integration in computer graphics, including references to further resources [Jensen et al. 2003]. The *Monte Carlo estimator* gives the expected value of an integral of a function f as the average of N separate estimates,

$$E\left[\int f(x)\,dx\right] = \frac{1}{N}\sum_{i=1}^{N}\frac{f(X_i)}{p(X_i)},$$

where X_i are random variables drawn from some sampling distribution p. Mathematically, any valid probability distribution can be used as long as $p(x) > 0$ whenever $f(x) \neq 0$.

The choice of sampling distribution p can dramatically affect the amount of variance in the Monte Carlo estimate. *Importance sampling* is a variance reduction technique that draws samples from a distribution p that is similar to f. When the sampling distribution matches the shape of the integrand, importance sampling can be an extremely effective optimization. However, if the sampling distribution under-samples locations where the integrand's value is large, variance increases substantially. Such a poorly chosen sampling distribution can substantially *increase* variance. For a modern discussion of this issue, see Owen and Zhou [2000].

Multiple importance sampling (MIS) is a generalization of importance sampling that addresses this problem by combining samples drawn from multiple distributions [Veach and Guibas 1995; Veach 1997]. The key property that a sampling algorithm must possess to be compatible with MIS is that it must be possible to compute the probability density $p(X_i)$ of generating any given sample value X_i with that technique, even if it was generated with some other method. We will not present the details of how to use this density in MIS, but instead refer the reader to Veach's original papers.

In summary, in order to compute estimates of the direct lighting integral, we would like sampling strategies to find directions ω_i from distributions that match components of the integrand. Because real-world illumination ranges over multiple orders of magnitude in intensity, the L_d term generally is the main contributor to the shape of the integrand. Therefore, importance sampling from its distribution is an efficient way to compute estimates of reflected radiance.

3 Sampling Environment Maps

In the derivation below we will assume that illumination is represented in an image with a (θ,ϕ) "latitude-longitude" parameterization given by $x = r\sin\theta\cos\phi$, $y = r\sin\theta\sin\phi$, and $z = r\cos\theta$. Given a direction ω, the radiance value $L_d(\omega)$ can be found by inverting the spherical coordinate formulas to find (θ,ϕ) from a direction $\omega = (x,y,z)$ and then interpolating among the nearby texels in the image map (we use bilinear interpolation).[2]

This map representation has a simple relationship to the angles (θ,ϕ), but it is distorted when mapped to the sphere of directions, especially at the poles of the sphere.[3] Due to this distortion, we cannot directly sample from the (θ,ϕ) distribution, since directions near the poles of the sphere would be oversampled. Trying to define a sampling distribution directly on the unit sphere is also difficult, since bilinear interpolation in (θ,ϕ) yields a non-linear interpolant on the surface of the sphere.

Our technique approaches this problem by sampling from a 2D (θ,ϕ) distribution, the most convenient domain to sample from, but then transform (θ,ϕ) to the appropriate density on the unit sphere $p(\omega)$. There are three main steps to our approach:

[1] The techniques described in this paper could easily be applied to a spherical or cube light source with finite radius that surrounded the scene.

[2] If the source image is in a different representation like a light probe or cube map, it can be warped and resampled into this representation, or the methods used here could be applied to other representations.

[3] This is a problem with *any* mapping from a region on the plane to the surface of a sphere. See Snyder and Mitchell's report for an analysis of distortion in environment map representations [Snyder and Mitchell 2001].

```
precompute1D(f[nf], out pf[nf], out Pf[nf+1]):
  I = sum(f[0] ... f[nf-1])
  for (i = 0 to nf-1):
    pf[i] = f[i] / I
  Pf[0] = 0
  for (i = 1 to nf-1):
    Pf[i] = Pf[i-1] + pf[i-1]
  Pf[nf] = 1

sample1D(pf[nf], Pf[nf+1], unif, out x, out p):
  i = binarySearch(Pf, unif) // Pf[i]<=unif<Pf[i+1]
  t = (Pf[i+1] - unif) / (Pf[i+1] - Pf[i])
  x = (1-t) * i + t * (i+1)
  p = pf[i]
```

Figure 2: Pseudocode for precomputing the PDF p_f and CDF P_f for a piecewise constant 1D function defined by an array of values `f[]` and sampling from its distribution. Using the precomputed information, the sampling function transforms a uniform random variable ξ to a sample from the distribution, returning both the value x and the value of the PDF $p(x)$.

- Define a piecewise constant probability density $p(\theta,\phi)$ based on the environment map's luminance distribution.
- Apply a sampling method that transforms random numbers over $[0,1]^2$ to samples drawn from $p(\theta,\phi)$.
- Derive a probability density function on the unit sphere $p(\omega)$ based on the probability density over (θ,ϕ).

These three steps allows us to sample directions ω from a distribution that is close to L_d, compute $p(\omega)$, and apply either the standard Monte Carlo estimator or the MIS estimator to evaluate the reflection equation with low error thanks to a sampling distribution that generally matches the integrand well.

Before we describe the algorithm in full, we will first present the necessary building blocks from probability theory. For a more complete introduction to probability distributions, see Ross [2002], and for a more graphics-centric view of these topics see Veach's thesis [1997] or Jensen et al.'s course notes [2003].

3.1 Sampling 1D Piecewise Constant Functions

We will represent piecewise constant functions $f(x)$ as sets of values f_i where i is an integer and f_i gives the value of the function over the range $[i,i+1)$. Given such a function, the integral $I_f = \int f(x)\,dx$ is easily seen to be $\sum_i f_i$. To find the probability distribution function p (PDF) that describes f, we must scale f so that the p integrates to one, obtaining $p_f(x) = f(x)/I_f = f(x)/\sum_i f_i$.

The cumulative distribution function (CDF) for f, $P_f(x) = \int_0^x p_f(x')\,dx'$, is a piecewise linear function where $P_f(0) = 0$, $P_f(n) = 1$, and for integer i, $P_f(i) = P_f(i-1) + p_f(i-1) = P_f(i-1) + f_{i-1}/I_f$.

To generate a sample from this distribution using a uniform random number ξ, we must find x such that $P_f(x) = \xi$. This can be done efficiently by precomputing the CDF for integer i values using the recurrence above and performing a binary search for i such that $P_f(i) \le \xi < P_f(i+1)$. Then x can be found by linearly interpolating between i and $i+1$ by amount $t = (\xi - P_f(i))/(P_f(i+1) - P_f(i))$. This process is summarized in Figure 2.

3.2 Sampling 2D Piecewise Constant Functions

Drawing a sample from a a 2D distribution $p(u,v)$ is more complicated than sampling a 1D function (unless p is separable into the product of two 1D functions in u and v). For general multidimensional joint probability distributions, each dimension must be sampled in turn, based on the values chosen for previous dimensions. Given a 2D density function $p(u,v)$, the *marginal density function* $p_u(u)$ is obtained by "integrating out" the v dimension:

$$p_u(u) = \int p(u,v)\,dv.$$

$p_u(u)$ can be thought of as the density function for u alone; more precisely, it is the average density for a particular u over all possible v values. The *conditional density function* $p_v(v|u)$ is the density function for v given that some particular u has been chosen,

$$p_v(v|u) = \frac{p(u,v)}{p_u(u)}.$$

To sample from a non-separable 2D joint distribution, one must first compute the marginal density and draw a sample from that density using standard 1D techniques such as the one described in the previous section. Once that sample is known, the corresponding conditional density function is obtained and sampled, again using standard 1D techniques.

For a piecewise constant 2D distribution, this process is particularly straightforward. Consider a function $f(u,v)$ defined by a set of $n_u n_v$ values $f_{i,j}$ where $f_{i,j}$ gives the value of f over the range $[i,i+1)\times[j,j+1)$. The joint 2D distribution that describes f's distribution is $p(u,v) = f(u,v)/\iint f(u,v)\,du\,dv = f(u,v)/I_f$, where $I_f = \sum_i\sum_j f_{i,j}$. The marginal density $p_u(u)$ is easily found as a sum of $f_{i,j}$ values, $p_u(u) = \int p(u,v)dv = \sum_j f_{i,j}/I_f$, where $i \le u < i+1$. Note that $p_u(u)$ is itself a piecewise constant function that can be quickly computed in a preprocessing step, and thus u samples can be taken as described in the previous section.

Given such a u sample, the conditional density $p_v(v|u)$ is $(f_{i,j}/I_f)/p_u(u)$. If the piecewise constant $p_u(u)$ function is represented as a set of values g_i with $i \le u < i+1$, we have $p_v(v|u) = (f_{i,j}/I_f)/g_i$, itself a piecewise constant function that can be sampled with the one-dimensional approach. This is summarized in the pseudocode in Figure 3.

```
precompute2D(f[nu][nv], out pu[nu], out Pu[nu+1],
             out pv[nu][nv], Pv[nu][nv+1]):
  for (u = 0 to nu-1):
    precompute1D(f[u], pv[u], Pv[u])
    colsum[u] = sum(f[u][0] ... f[u][nv-1])
  precompute1D(colsum, pu, Pu)

sample2D(pu[nu], Pu[nu+1], pv[nu][nv], Pv[nu][nv+1],
         unif1, unif2, out u, out v, out pdf):
  sample1D(pu, Pu, unif1, u, pdfu)
  sample1D(pv[nu*u], Pv[nu*u], unif2, v, pdfv)
  pdf = pdfu * pdfv
```

Figure 3: Precomputation and sampling pseudocode for 2D sampling. For precomputation, we first compute all of the conditional densities into the `pv` and `Pv` arrays, and then use those to precompute the marginal density into `pu` and `Pu`. Sampling is done by first sampling the 1D marginal density, and using that value to choose the appropriate 1D conditional density to sample.

3.3 Transforming Between Distributions

It is frequently the case that we are given a multi-dimensional random variable X that was sampled from some distribution $p(X)$ (e.g., a uniform distribution over $[0,1]^2$), but we would like to transform this variable to a random variable X' over some other domain (e.g., the surface of the unit sphere). It is easy to compute the density $p'(X')$ of the new random variable in terms of the original density $p(X)$ and the bijection T that transforms $X \to X'$. The probability density of the new random variable X' can be shown to be

$$p'(X') = p'(T(X)) = \frac{p(X)}{|J_T(X)|}, \quad (1)$$

where $|J_T|$ is the absolute value of the determinant of T's Jacobian,

$$\begin{vmatrix} \partial T_1/\partial x_1 & \cdots & \partial T_1/\partial x_n \\ \vdots & \ddots & \vdots \\ \partial T_n/\partial x_1 & \cdots & \partial T_n/\partial x_n \end{vmatrix},$$

and T_i are defined by $T(x) = (T_1(x), \ldots, T_n(x))$.

3.4 Our Algorithm

Bringing these techniques together, our sampling algorithm creates a piecewise constant function over (θ, ϕ) by applying a slight Gaussian blur to the image and computing pixel luminances to define a piecewise-constant function $f(u,v)$. We set the number of function values $f_{i,j}$ from the original map resolution, though a lower resolution could be used as well. We then precompute the marginal density $p_u(u)$ by summing the values in the columns of the image and normalizing by I_f. Finally, we find the piecewise constant conditional densities for each column as shown in Figure 3. This precomputation requires less than 100 lines of C++ code.

Given a pair of uniformly distributed random variables (ξ_1, ξ_2) over $[0,1]^2$, we can draw a sample from the precomputed densities using the sampling algorithm in Figure 3, which simultaneously gives a (u,v) value and the value of the PDF $p(u,v)$. The (u,v) sample is mapped to a direction (θ, ϕ) on the unit sphere by scaling by $(\pi/n_u, 2\pi/n_v)$ and then spherical coordinates give a direction $\omega = (x,y,z)$.

We also need to convert the probability density for sampling (u,v) to one expressed in terms of solid angle on the sphere using the transformation from Section 3.3. Consider the function g that maps from (u,v) to (θ,ϕ), $g(u,v) = (\pi u/n_u, 2\pi v/n_v)$. The absolute value of the determinant of the Jacobian $|J_g|$ is $2\pi^2/(n_u n_v)$. Applying Equation 1, $p(\theta,\phi) = p(u,v) n_u n_v / 2\pi^2$.

Using the definition of spherical coordinates, the absolute value of the Jacobian for the mapping from (r,θ,ϕ) to (x,y,z) is $r^2 \sin\theta$. Since we are interested in the unit sphere, $r = 1$, and again applying Equation 1 to find the final relationship between probability densities in terms of the probability density for the sample from the 2D piecewise constant function to the direction on the sphere,

$$p(\omega) = p(u,v) \frac{n_u n_v}{2\pi^2 \sin\theta}.$$

This is the key relationship for applying our technique: it lets us sample from the piecewise constant distribution and transform the sample and its probability density to the unit sphere. Because we have access to this probability density over the appropriate measure, we can easily apply multiple importance sampling if desired.

The above algorithm gives a correct sampling technique, but we can improve it slightly by multiplying each $f_{i,j}$ value by a $\sin\theta$ term to account for the latitude-longitude parameterization's distortion. This is still a valid sampling distribution, though we will (correctly) tend to take fewer samples near the poles. Note that it is still necessary to include the $\sin\theta$ term in the density conversion even if this improvement is applied; the transformation terms do not depend on the density, just the relationship between domains.

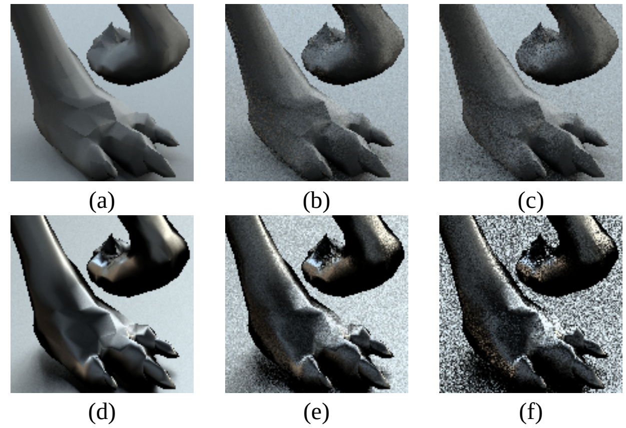

Figure 4: A sampling of images from our experiments. The scene is a zoomed-in portion of the scene in Figure 8, using a Cook-Torrance BRDF and lit with the Eucalyptus grove map. Top row: specular exponent of 2.5. (a) Reference image. (b) Our light sampling method. (c) Structured importance sampling. Bottom row: specular exponent of 100. (d) Reference image. (e) Multiple importance sampling using our light sampling algorithm. (f) Structured importance sampling. 64 samples were taken for all images.

4 Results

In order to evaluate the effectiveness of our sampling technique, we computed the L^2 error of images rendered with a number of techniques against a high quality reference image. For these experiments, we rendered the scene shown in Figure 4 with a Cook–Torrance BRDF. This scene incorporates a variety of surface orientations so that different sections of the environment map will be important for different image pixels. It also exhibits complex visibility and shadowing, so that a sampling method that suffered from excessive clumping of samples in bright regions would perform poorly.

The three approaches we compared were: importance sampling from the light's distribution using our technique, multiple importance sampling with half of the samples taken from the light's distribution and half taken from the BRDF's, and structured importance sampling.[4] For the approaches based on our sampling technique, we used randomized low-discrepancy point generated with Kollig and Keller's technique [2002] as (ξ_1, ξ_2) values for sampling directions from the light and BRDF distributions. For the structured importance sampling comparisons, we used the preintegrated illumination within each stratum and jittered the directions of the shadow rays traced within the cone of directions subtended by the stratum. Later in this section we will compare our technique to SIS for generating a set of directional lights.

For our tests, we used two environment maps downloaded from *www.debevec.org/Probes*: Galileo's tomb and the Eucalyptus grove, both at 1024×512 resolution (results when using other maps were similar). These maps are representative of the two most common (and challenging) types of environment maps: those with the majority of their illumination concentrated in a set of small bright regions (Galileo), and those with illumination spread out over a broader area while still having substantial areas with low contribution (Eucalyptus). Our sampling technique required less than 0.03 seconds of preprocessing for these environment maps on a 2.8GHz

[4] We also did experiments with sampling from the BRDF's distribution alone, but for environment map illumination this was only competitive for highly glossy surfaces and otherwise had significantly higher error, so we have not included those results here.

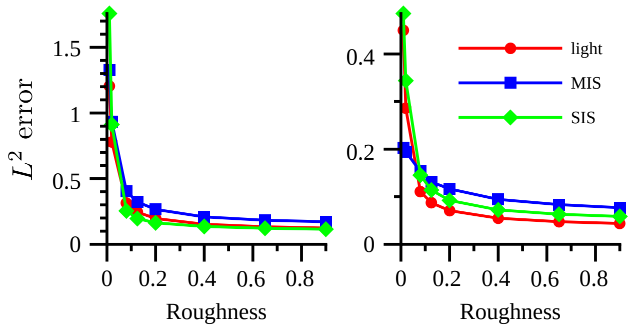

Figure 5: Graphs showing the L^2 error for images of the test scene with the three sampling techniques as the glossiness of the surface being rendered varies, using 64 light samples per pixel. The left graph is Galileo's tomb, and the right is the Eucalyptus grove. Our method consistently performs as well as or better than structured importance sampling. For very glossy surfaces, applying multiple importance sampling is the most effective approach.

Pentium 4 CPU.

Figure 5 compares the L^2 image error of the three techniques for images rendered with 64 light samples per pixel. For the Galileo's tomb environment map, SIS and our method give similar error, with slightly less from SIS. For the Eucalyptus grove, our technique has less error than SIS. For all tests that we ran, multiple importance sampling was only better when the object was very glossy. It is likely that adaptively assigning proportions of samples to the lighting and the BRDF based on the object's reflection properties would make MIS an appropriate default for all scenes.

4.1 Discussion

In addition to requiring very little preprocessing, the time spent generating samples from the light source distribution in our method is negligible: each sample requires only two binary searches of less than ten comparisons each, and a few trigonometric function evaluations. For scenes with complex geometry and shading, the time spent on sampling is a tiny fraction of overall rendering time.

One reason for the low error of images computed with our method is that we can leverage classic Monte Carlo variance reduction techniques. Because our method transforms random variables defined over $[0,1]^2$ to directions on the sphere while preserving stratification reasonably well, the low-discrepancy point set we use for sampling gives a well-stratified set of spherical directions without any additional computation. It is well known that preserving good distribution properties when transforming between domains yields lower variance [Shirley and Chiu 1997]. Other approaches cannot take advantage of low discrepancy samples as easily or can be stratified in only one dimension, and thus cannot take advantage of the variance reduction afforded by these sampling patterns. In particular, the straightforward approach of transforming the environment map into a large set of directional lights and sampling from that 1D distribution does not preserve stratification on the sphere.

In order to apply multiple importance sampling, it is necessary to be able to compute the probability densities associated with each sampling method, even if the sample point is given *a priori*. It is likely that MIS would help other light source sampling methods handle glossy objects as it does our method, but it is not clear how to compute the appropriate probability densities for existing techniques such as SIS or Kollig and Keller's.

We have found that sampling the light's distribution does not result in an excessive number of samples allocated to very bright regions of the image; Figure 6 shows 128 sample points generated by our algorithm in the St. Peters environment map. Additionally, any clumping that does occur can be offset by our compatibility with multiple importance sampling, since half of the samples will be drawn from BSDF's distribution and thus cannot clump due to properties of the lighting.

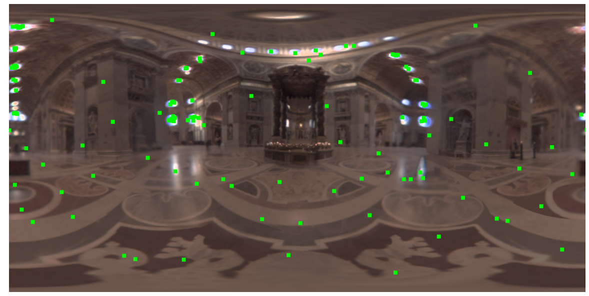

Figure 6: 128 sample points placed in the St. Peters environment map by warping a low discrepancy point set in the unit square to the distribution of illumination using our technique. Note that samples are more likely to be taken in bright parts of the environment map, though there is not excessive clumping of samples in bright areas.

Most previous techniques for sampling environment maps have been able to turn the incident lighting into a set of directional lights for fast, zero-variance rendering. Our methods are compatible with this approach, as shown in Figure 8. Thanks to the efficiency of our algorithm, a set of directional lights can be found in real time, making it possible, for example, to use real-time video [Kang et al. 2003] as input for GPU-based interactive rendering system.

5 Conclusion and Future Work

We have presented a new sampling algorithm for image-based lighting in Monte Carlo rendering. Our algorithm is unbiased, easy to implement, and is fully compatible with standard Monte Carlo variance reduction techniques such as low discrepancy sampling and multiple importance sampling. It computes results with similar or less error than previous techniques for the same number of samples while requiring precomputation time measured in hundredths of seconds, rather than tens of seconds or minutes. The ability to use it with multiple importance sampling improves its robustness in the presence of highly specular reflection.

We have not investigated further optimization techniques for reducing the number of light samples taken (for example, like Agarwal et al.'s adaptation of Ward's probabilistic handling of large numbers of light sources [Ward 1991].) However, we believe that ideas like these can equally well be applied to our technique, with the potential for similar substantial reductions in the number of rays traced.

Preliminary experiments with using tone mapping algorithms to slightly reduce the dynamic range of the data in the sampling distribution suggest that this may provide a successful way to improve stratification over the area of the environment map. Using an efficient algorithm such as the one described by Reinhard et al. [2002] would likely be most appropriate. This may provide a way to further reduce the error with our approach by better sampling the visibility component of the reflection equation.

References

AGARWAL, S., RAMAMOORTHI, R., BELONGIE, S., AND JENSEN, H. W. 2003. Structured importance sampling of environment maps. *ACM Transactions on Graphics 22*, 3 (July), 605–612.

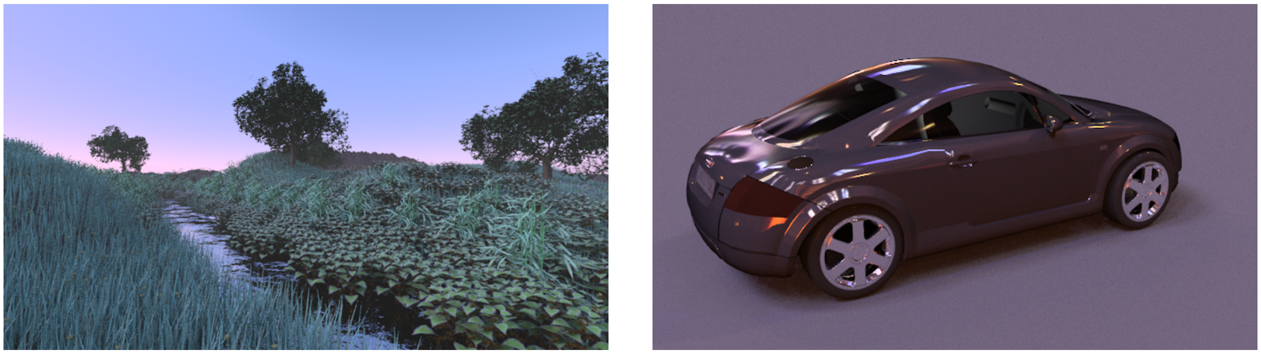

Figure 7: Ecosystem scene lit by another simulated sky environment, and the TT car model lit by the Grace Cathedral map.

Figure 8: Using our method and structured importance sampling to generate directional light sources: from left, high-quality reference image, image rendered with 32 lights created by sampling from the light's distribution using our method, and image rendered with 32 lights found with SIS. Both approaches do reasonably well at capturing the ground shadows and highlights on the creature, even with a small number of light sources. The lights created using SIS give a better match to the reference image (compare the highlights on the head, for example), though our method takes less than 0.025 seconds for both precomputation and selecting the light directions, while SIS takes 76 seconds.

BLINN, J. F., AND NEWELL, M. E. 1976. Texture and reflection in computer generated images. *Communications of the ACM 19*, 542–546.

COHEN, J., AND DEBEVEC, P., 2001. LightGEN plugin. http://www.ict.usc.edu/~jcohen/lightgen/lightgen.html.

COHEN, J. M., 2003. Estimating reflected radiance under complex distant illumination. Rhythm and Hues Studios Technical Report RH-TR-2003-1.

DEBEVEC, P. 1998. Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *Proceedings of SIGGRAPH 98*, ACM SIGGRAPH / Addison Wesley, Orlando, Florida, 189–198.

GREENE, N. 1986. Environment mapping and other applications of world projections. *IEEE Computer Graphics & Applications 6*, 11 (November), 21–29.

JENSEN, H. W., ARVO, J., DUTRÉ, P., KELLER, A., OWEN, A., PHARR, M., AND SHIRLEY, P., 2003. Monte Carlo ray tracing, July. SIGGRAPH '2003 Course, San Diego.

KANG, S. B., UYTTENDAELE, M., WINDER, S., AND SZELISKI, R. 2003. High dynamic range video. *ACM Transactions on Graphics 22*, 3 (July), 319–325.

KOLLIG, T., AND KELLER, A. 2002. Efficient multidimensional sampling. In *Computer Graphics Forum (Proceedings of Eurographics 2002)*, G. Drettakis and H.-P. Seidel, Eds., vol. 21, 557–563.

KOLLIG, T., AND KELLER, A. 2003. Efficient illumination by high dynamic range images. In *Eurographics Symposium on Rendering: 14th Eurographics Workshop on Rendering*, 45–51.

MILLER, G. S., AND HOFFMAN, C. R., 1984. Illumination and reflection maps: Simulated objects in simulated and real environments. Course Notes for Advanced Computer Graphics Animation, SIGGRAPH 84.

NG, R., RAMAMOORTHI, R., AND HANRAHAN, P. 2003. All-frequency shadows using non-linear wavelet lighting approximation. *ACM Transactions on Graphics 22*, 3 (July), 376–381.

OWEN, A., AND ZHOU, Y. 2000. Safe and effective importance sampling. *Journal of the American Statistical Association 95*, 449, 135–.

RAMAMOORTHI, R., AND HANRAHAN, P. 2002. Frequency space environment map rendering. *ACM Transactions on Graphics 21*, 3 (July), 517–526.

REINHARD, E., STARK, M., SHIRLEY, P., AND FERWERDA, J. 2002. Photographic tone reproduction for digital images. *ACM Transactions on Graphics 21*, 3, 267–276. Proceedings of ACM SIGGRAPH 2002.

ROSS, S. M. 2002. *Introduction to Probability Models*, 8 ed. Academic Press.

SHIRLEY, P., AND CHIU, K. 1997. A low distortion map between disk and square. *Journal of Graphics Tools 2*, 3, 45–52. ISSN 1086-7651.

SLOAN, P.-P., KAUTZ, J., AND SNYDER, J. 2002. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. *ACM Transactions on Graphics 21*, 3 (July), 527–536.

SNYDER, J., AND MITCHELL, D., 2001. Sampling-efficient mapping of spherical images. Microsoft Technical Report, available at http://www.mentallandscape.com/Papers_01spheremap.pdf.

VEACH, E., AND GUIBAS, L. J. 1995. Optimally combining sampling techniques for Monte Carlo rendering. In *Computer Graphics (SIGGRAPH Proceedings)*, 419–428.

VEACH, E. 1997. *Robust Monte Carlo Methods for Light Transport Simulation*. PhD thesis, Stanford University.

WARD, G. 1991. Adaptive shadow testing for ray tracing. In *Eurographics Workshop on Rendering*.

WARD, G. J. 1994. The Radiance lighting simulation and rendering system. In *Proceedings of SIGGRAPH '94*, A. Glassner, Ed., 459–472.

WILLIAMS, L. 1983. Pyramidal parametrics. In *Computer Graphics (SIGGRAPH '83 Proceedings)*, vol. 17, 1–11.