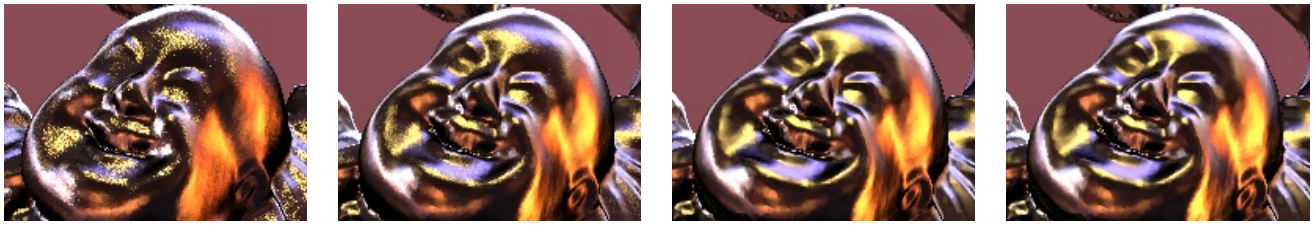
Structured Importance Sampling of Environment Maps

Sameer Agarwal* Ravi Ramamoorthi† Serge Belongie* Henrik Wann Jensen*

*University of California, San Diego

†Columbia University



Importance w/ 300 samples | Importance w/ 3000 samples | Structured importance w/ 300 samples | Structured importance w/ 4.7 rays/pixel

Figure 1: *Close-up rendering of a glossy buddha in the grace cathedral environment. The two images on the left have been rendered using stratified importance sampling with 300 and 3000 samples, while the two images on the right show the result of structured importance sampling using 300 samples, and after further rendering optimizations an average of 4.7 rays per pixel to evaluate the 300 possible samples.*

Abstract

We introduce *structured importance sampling*, a new technique for efficiently rendering scenes illuminated by distant natural illumination given in an environment map. Our method handles occlusion, high-frequency lighting, and is significantly faster than alternative methods based on Monte Carlo sampling. We achieve this speedup as a result of several ideas. First, we present a new metric for stratifying and sampling an environment map taking into account both the illumination intensity as well as the expected variance due to occlusion within the scene. We then present a novel hierarchical stratification algorithm that uses our metric to automatically stratify the environment map into regular strata. This approach enables a number of rendering optimizations, such as pre-integrating the illumination within each stratum to eliminate noise at the cost of adding bias, and sorting the strata to reduce the number of sample rays. We have rendered several scenes illuminated by natural lighting, and our results indicate that structured importance sampling is better than the best previous Monte Carlo techniques, requiring one to two orders of magnitude fewer samples for the same image quality.

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

Keywords: Rendering, Image Synthesis, Illumination, Ray Tracing, Monte Carlo Techniques, Shadow Algorithms, Global Illumination, Environment Mapping

1 Introduction

To capture realistic natural lighting, it is common to use environment maps, a representation of the distant illumination at a point. Environment map rendering has a long history in graphics, going back to seminal work by Blinn and Newell [1976], Miller and Hoffman [1984], Greene [1986] and Cabral et al. [1987], as well as recent work on high-dynamic range imagery by Debevec [1998], and extensions of the basic environment mapping ideas by Cabral et al. [1999], Kautz and McCool [2000], Kautz et al. [2000], Ramamoorthi and Hanrahan [2001; 2002], and others.

Most of the previous environment mapping techniques [Miller and Hoffman 1984; Greene 1986; Ramamoorthi and Hanrahan 2001] are intended for real-time applications, and ignore visibility. They usually require an expensive pre-computation or pre-filtering step, where an irradiance environment map is obtained by convolving the incident illumination with the Lambertian or more complex reflection function. Ramamoorthi and Hanrahan [2001; 2002] propose fast pre-filtering methods using spherical harmonics, but their methods also make the common assumption of no cast shadows. In recent work, Sloan et al. [2002] have demonstrated real-time rendering taking visibility effects into account, but their technique is limited to static scenes with low-frequency lighting, and requires a slow pre-computation step involving ray tracing and detailed sampling of visibility.

In this paper, we address the problem of efficiently rendering high quality images of scenes illuminated by arbitrary environment maps. Our method specifically optimizes the integration of distant illumination on surfaces with Lambertian and semi-glossy BRDFs, it correctly accounts for occlusion within the scene (such as shadows due to bright lights in the environment map), and it handles scenes with changing geometry. In terms of global illumination research, our method can be viewed as an efficient technique for sampling millions of distant lights corresponding to pixels in an environment map. We seek to estimate the integral of a product of the visibility and the illumination. One of these, the illumination, is known, and is the same for every surface point in the scene, and may also be reused for multiple scenes or multiple frames of an animation. Therefore, unlike many previous image synthesis problems, it is feasible to perform extensive preprocessing on the environment map without degrading performance. Visibility, on the other hand can be complicated and changes throughout the scene, requiring sampling for general scenes. Naive Monte Carlo sampling such as

path tracing [Kajiya 1986] is well-suited for sampling visibility, but in the presence of high-frequency environment maps it results in significant noise, since it does not take the variation of the illumination into account. In this situation it is better to use importance sampling based on the illumination in the environment map. Even though importance sampling is significantly better than path tracing, it is not deterministic and results in significant sampling noise as shown in Figure 1. Pure illumination based importance sampling also tends to use too many samples on small bright lights such as the sun in the blue sky even though it is very small, within which the variation in visibility is mostly insignificant.

To understand how to sample an environment map, we present a novel analysis of visibility variance, and develop a metric for sampling both visibility and illumination efficiently. We also introduce a general and automatic hierarchical stratification algorithm for partitioning environment maps into a set of area light sources. The algorithm performs hierarchical thresholding of the map and uses our importance metric to deterministically allocate samples to each level. The samples are then placed inside each level using the Hochbaum-Shmoys clustering algorithm [Hochbaum and Shmoys 1985], which has strong runtime and quality guarantees associated with it.

Our stratification algorithm ensures a good sampling pattern of the environment map, and in addition it enables a number of rendering optimizations that are difficult to include in standard Monte Carlo techniques. We can eliminate sampling noise and make the method completely deterministic by pre-integrating the illumination in each stratum — effectively collapsing the stratum into a directional light source. This results in a set of lights approximating the environment similar to the output of the LightGen plug-in for HDRShop [Cohen and Debevec 2001].

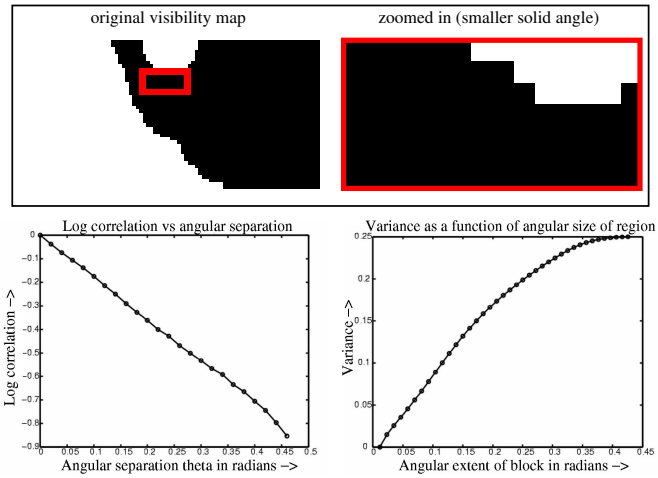


Figure 2: *Analysis of a representative visibility map, confirming the qualitative results expected, and validating our quantitative analysis.* **Top:** *Part of a visibility map for one pixel on the ground plane with a teapot casting shadows. On the left is the original binary visibility map. This region has approximately equal visible and shadowed regions and the variance over the whole region is the maximum,* $1/4 = 0.25$. *We zoom in on the red rectangle in the right figure. Even though this is one of the most complex regions in the original visibility map, it is clear that over a smaller solid angle, visibility is much simpler, and the variance drops down to less than* 0.14. **Bottom:** *Quantitative analysis of above effect. On the left, we plot the log of the correlation function* $\tilde{\alpha}(\theta)$ *as it varies with with the angular separation* θ. *Confirming our theoretical analysis, we find a straight line, showing that correlation decays exponentially with* θ. *The correlation angle* T *is estimated from this plot as* $T \approx 0.5$. *On the right, we plot the variance as a function of the angular separation. For small* θ, *the variance increases linearly, fairly accurately obeying Equation 6. As* θ/T *approaches* $3/4$, *the graph tails off, with the maximum variance of* $1/4$ *being approached.*

2 Monte Carlo Sampling and Importance

In this section, we analyze Monte Carlo integration of irradiance due to environment maps in more detail with the purpose of defining an appropriate importance metric. The irradiance, E, at a given surface location, x is computed as

$$E(x) = \int_{\Omega_{2\pi}} L_i(\vec{\omega}) S(x,\vec{\omega})(\vec{\omega}\cdot\vec{n})\, d\vec{\omega}\ , \qquad (1)$$

where $\Omega_{2\pi}$ is the hemisphere of directions above x, L_i is the incident radiance or environment map, indexed only as a function of angle $\vec{\omega}$, S is the (binary) visibility in the direction $\vec{\omega}$, and $\vec{n}$ is the surface normal at x. For our analysis we will ignore the surface orientation and focus on the illumination and visibility. A complete rendering algorithm is presented in section 4. Our goal is to compute this integral efficiently using Monte Carlo sampling, where L_i is a known function that is easy to evaluate and the same for all points x, while S requires sampling since it is unknown and depends on x.

To understand how to distribute samples in the domain of the integrand, i.e. the illumination sphere, we introduce a new importance metric Γ. The two key competing strategies here are area-based stratified sampling and illumination-based importance sampling. To unify these two extremes and intermediate possibilities within a common framework, we use the following general metric for distributing samples in a region of solid angle $\Delta\omega$, with net (integrated) illumination L,

$$\Gamma(L,\Delta\omega) = L^a \Delta\omega^b\ . \qquad (2)$$

Here, a and b are parameters we seek to determine. First, consider the extreme cases: $a = 1, b = 0$ corresponds to standard illumination-based importance regardless of area (this technique over-samples small bright lights), and $a = 0, b = 1$ corresponds to pure area-based stratification without considering illumination (this technique under-samples the bright lights). It would appear at first glance that both extremes have problems, and an intermediate parameter setting is better. To determine the parameters for the optimal metric we first analyze the variance due to visibility.

2.1 Variance Analysis for Visibility

We now present a novel preliminary theoretical and empirical analysis showing that variance in visibility is proportional to the angular extent (square root of solid angle) of the region in question, providing a basis for a new importance metric.

We analyze the expected variance of the visibility function $S(x,\vec{\omega})$ in a region subtending solid angle $\Delta\omega$, corresponding to a cluster or light source. Intuitively, we expect some coherence in the visibility function S, at least over small regions, so we expect the variance to be a function of $\Delta\omega$, with less variance in smaller regions and more in larger regions. Figure 2 shows some empirical tests on a representative visibility map—we have carried out experimental tests on approximately 10 visibility maps. In particular, in the top left, we show a relatively complex section of the visibility map (here, part of one face of a binary cubemap). Even in the most complex regions, the visibility function is much more coherent when zooming in on a smaller region, as shown in the top right of Figure 2.

For further quantitative analysis, we assume a correlation model for visibility. Mathematically, we can define a correlation function,

$$\alpha(\theta) = P(S(\vec{\omega}_1) = S(\vec{\omega}_2) | \parallel \vec{\omega}_1 - \vec{\omega}_2 \parallel = \theta)\ , \qquad (3)$$

which simply measures the probability that the visibility function S is the same for two points separated by a distance (angle) θ. Assuming the worst case that the mean visibility β is $1/2$, or that overall $P(S=0) = P(S=1) = 1/2$, gives the highest overall vari-

ance of 1/4. Under this assumption as $\theta \to 0$, $\alpha(\theta) = 1$ (neighboring points are the same with high probability), while as θ becomes large $\alpha(\theta) \to 1/2$, i.e. things become random. We often prefer a measure going from 0 to 1 instead, and define

$$\tilde{\alpha}(\theta) = 2\alpha(\theta) - 1 \,, \tag{4}$$

We will use an exponential model for $\tilde{\alpha}(\theta)$

$$\tilde{\alpha}(\theta) = e^{-\theta/T} \,. \tag{5}$$

The equation above has a simple explanation—the probability that the visibility at points is correlated decays exponentially with the angular separation between the points. T is the correlation angle for visibility and measures the average visibility feature size. It can be estimated from the above equation for a given visibility map, but the precise value will turn out not to be of significant practical importance for sampling.

While the simple form of Equation 5 is only plausible, and cannot be rigorously proven, our empirical tests on approximately 10 visibility maps, with T ranging from .02 to .6, indicate that it is a reasonably good approximation. A representative example is shown in Figure 2. The bottom left of Figure 2 plots $\log \tilde{\alpha}(\theta)$ as a function of θ for one visibility map. We see that this graph is almost a straight line, confirming the exponential correlation model.

In the appendix, we derive that the expected variance for a small angular extent θ (corresponding to a solid angle $\Delta\omega \approx \pi\theta^2$) is

$$V[S, \Delta\omega] \approx \frac{\theta}{3T} \,. \tag{6}$$

Note that the above equation is valid only when $\theta/T \ll 3/4$; as θ/T becomes larger, visibility becomes essentially random and uncorrelated, so the variance will tend toward the limit of its maximal or worst-case value of $1/4$. The bottom right of Figure 2 plots the variance as a function of angular extent θ. Equation 6 is quite closely followed for $\theta/T \ll 3/4$, and the variance increases linearly with θ. At some solid angle $\Delta\omega_0$, at which θ/T is close to $3/4$, the variance tails off, approaching the limit of $1/4$. Here $\Delta\omega_0 \propto T^2$.

The appendix goes further in using Equation 6 to determine the optimal distribution of samples. In particular, the variance of a region of total illumination L subtending a solid angle $\Delta\omega$ is proportional to $L^2\Delta\omega^{1/2}$, assuming uniform lighting. Hence, the number of samples is proportional to $\sqrt{L^2\Delta\omega^{1/2}} = L\Delta\omega^{1/4}$. Notice that this formula applies only when $\Delta\omega \leq \Delta\omega_0$, since visibility variance tends to a constant for $\Delta\omega > \Delta\omega_0$. Our results do not seem to be sensitive to the precise value used to estimate T, either in the distribution of the samples or the corresponding value of $\Delta\omega_0$.

2.2 An Importance Metric for Environment Maps

Based on the visibility analysis, we use $a = 1$, $b = 1/4$ for our importance metric (Equation 2). This has the nice property of reducing the number of samples allocated to small (in the limit point-like[1]) sources, while it reduces to illumination importance ($a = 1$) for equal area regions. A major consequence of our analysis and empirical results is that visibility coherence is significant only for small light sources, and consequently the main impact of our metric is to reduce the number of samples for these lights while using an essentially illumination-based metric in other regions. In particular, we use the the following modified version of Equation 2:

$$\Gamma(L, \Delta\omega) = L \cdot \left(\min(\Delta\omega, \Delta\omega_0)\right)^{1/4} . \tag{7}$$

[1] As the area goes to zero the number of samples allocated for a light can become less than one. Since visibility sampling requires at least one ray, we ensure that we always allocate one sample for a light if illumination importance would assign one or more samples to this light.

As $\Delta\omega_0 \to 0$, our metric becomes similar to illumination based importance, and as $\Delta\omega_0 \to 4\pi$, the metric penalizes small lights more and more, resulting in a distribution which is closer to area based stratification. As mentioned in the previous section, we found that the correlation angle T typically varies between 0.02 and .6, and we make a conservative choice of $\Delta\omega_0 = .01$.

3 Hierarchical Environment Map Stratification

In this section, we describe how to efficiently stratify an environment map using the visibility importance metric from the previous section. Our stratification algorithm consists of two steps: a hierarchical thresholding procedure that assigns samples to different regions in the map, and a stratification algorithm to subdivide the regions into strata that can be sampled during rendering. This algorithm is applicable to general irregular multidimensional spaces and may have applications beyond sampling of environment maps.

3.1 Hierarchical Thresholding

Environment maps with natural lighting (e.g. [Debevec 1998]) have illumination that varies significantly throughout the map. Elements such as small bright lights, large bright windows, and dark regions mean that the importance of different regions and consequently the desired number of samples or strata is highly non-uniform.

To create this non-uniform stratification we use a hierarchical thresholding in which we threshold the map at given illumination intensities to create levels in the map of approximately equal intensity that can be assigned an appropriate number of strata.

Our thresholding algorithm uses the standard deviation σ of the illumination in the map to define a standard scale that is independent of the dynamic range of the map

$$t_i = i\sigma \quad i = 0, ..., d-1 \,. \tag{8}$$

Here, t_i is the threshold value for level i, and d is the number of hierarchies (We use $d = 6$ for all our examples). The above scheme defines an increasing sequence of threshold values, and since the intensity values are positive it starts at $t_0 = 0$. While it is possible to define more sophisticated thresholding schemes, this scheme works well in practice and we use it for all our experiments.

To assign samples, we first compute our metric for the entire map as:

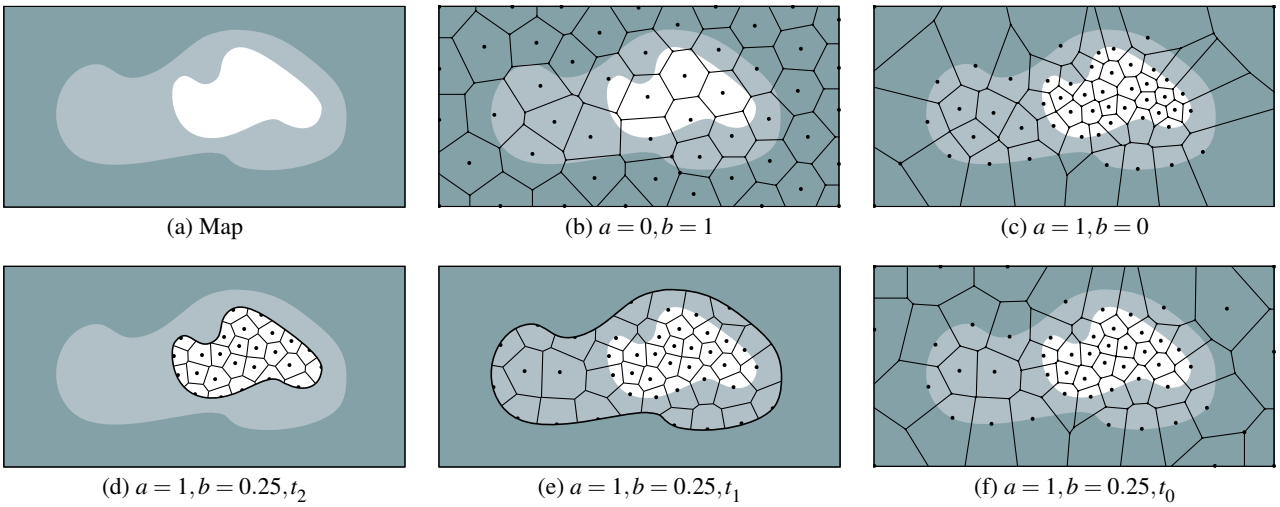
$$\Gamma_{4\pi} = \Gamma(\sum_i L_i, \Delta\omega_0) = L\Delta\omega_0^{1/4} \,, \tag{9}$$

where the sum is over all pixels in the map (L is the net illumination of the whole map), and $\Delta\omega_0 \ll 4\pi$ is the area of the smallest lights for which area importance is used, as per the discussion at the end of the previous section. Our sample assignment proceeds hierarchically by first assigning samples to the brightest regions as given by the threshold value t_{d-1}. We detect all pixels M_{d-1} brighter than t_{d-1}. For all pixels in M_{d-1} we find the connected components C_j [Gonzalez and Woods 2001]. For each connected component C_j we evaluate our metric for all pixels in the component as $\Gamma_j = \Gamma(\sum_{i \in C_j} L_i, \sum_{i \in C_j} \Delta\omega_i)$, where $\Delta\omega_i$ is the solid angle of pixel i in C_j. The number of samples assigned to the component can then be computed as

$$N_j = N\frac{\Gamma_j}{\Gamma_{4\pi}} \,, \tag{10}$$

where N is the total number of samples used.

Once the samples have been assigned to the components corresponding to t_{d-1}, we proceed to the next threshold level t_{d-2}. All the components at this level contain the components from the



(a) Map (b) $a = 0, b = 1$ (c) $a = 1, b = 0$

(d) $a = 1, b = 0.25, t_2$ (e) $a = 1, b = 0.25, t_1$ (f) $a = 1, b = 0.25, t_0$

Figure 3: *This figure illustrates our hierarchical stratification algorithm on a plane map using three different metrics. (a) shows the map that is being stratified, after the hierarchical thresholding. It contains three regions, with the illumination constant in each region and decreasing as we move away from the center. (b) shows the result of stratifying the map based purely on area. Note how few samples the central bright region gets. (c) shows the result of stratifying the map based solely on illumination, while (f) shows the result using our metric. We see that while our stratification is largely similar to one based on illumination importance, we allocate a somewhat smaller number of samples to the small bright central region (25 vs 35) because of increased visibility coherence. Finally, figures (d)-(f) demonstrate how our algorithm hierarchically operates on each illumination level, stratifying it and carrying the strata centers to the next level to ensure a good global stratification.*

previous (brighter) level. In effect, we grow the components and possibly create new components. Our assignment of the total number of samples to the individual components proceeds exactly as described before and we compute N_j for each component. Finally, we compute the number of samples to add to this component at this level by subtracting the number of samples already assigned to the elements of the previous hierarchy within the component.

This continues until we reach t_0 at which point we include all the remaining illumination in the map. The result is a hierarchy of levels with individual components each assigned a fixed number of samples. The next section describes an efficient algorithm for stratifying the entire map utilizing this hierarchical structure.

3.2 Hierarchical Stratification

The hierarchical thresholding of the environment map results in a number of samples assigned to each component at each level. Next we have to stratify these irregularly shaped and non-Euclidean components into the appropriate number of compact strata of approximately equal area. In other words, we must partition an arbitrary set of connected points (pixels in the environment map) into k disjoint partitions, where k is the number of samples allocated this component.

A good metric for this partitioning is to minimize the maximum distance between the center of each partition and any point inside it. This is a well-studied problem in theoretical computer science, and is known to be NP-hard [Nemhauser and Wolsey 1988]. Given the number of strata, the task of stratifying a region of the illumination map is equivalent to solving the k-centers problem on the surface of a sphere. Given the NP-Hardness of the problem, there is little hope of a polynomial time solution. However, it is possible to get high quality approximations in polynomial time. We use the pairwise clustering method proposed by Hochbaum and Shmoys [1985]. This algorithm produces a 2-approximation, i.e. the quality of the stratification returned by the algorithm is at most 2 times worse than the optimal stratification. In fact it can proved that this is the best approximation possible in polynomial time [Feder and Greene 1988]. Figure 4 gives the pseudocode for the algorithm.

The Hochbaum-Shmoys algorithm takes as input a set of points

Dataset $Y = \{y_1, y_2, \ldots, y_n\}$,X={}
Hochbaum-Shmoys Algorithm

Pick an arbitrary point in Y and label it x_1
Add x_1 to X.
For $i = 2, 3, \ldots, k$

$$x_i = \underset{y_p}{\operatorname{argmax}} \left[\min_q \left[d(y_p, x_q) \right] \right]$$

Add x_i to X.
For $j = 1, 2, \ldots, k$

$$S_j = \{y : y \in Y, x_j = \underset{x_l}{\operatorname{argmin}} [d(y, x_l)]\}$$

Return $X, \{S_j\}$

Figure 4: *The Hochbaum-Shmoys Algorithm.*

(in our case the set of points in a patch C_j) and a function $d(x,y)$ which gives the distance between any pair of points in that set. The algorithm performs a *farthest-first* traversal of the dataset. Starting with an arbitrarily chosen point x_1 and adding it to the set X, the algorithm in each iteration picks that point in C_j which is farthest away from the points in X and adds it to X. At the end of k iterations, each point in X acts as a center for a stratum and the result of the algorithm is a disjoint partitioning obtained by assigning each point in C_j to its nearest point in X.

A very useful property of the Hochbaum-Shmoys algorithm is that the position of the centers x_i does not change across iterations; hence it can be used for adding centers to a region which already contains a number of centers assigned to it. This means that the algorithm integrates directly with the hierarchical assignment algorithm outlined in the previous section, since it can add more strata centers to a component that already has a number of points assigned to previous hierarchy levels.

Figure 3 illustrates our algorithm and compares the stratification obtained by using our proposed importance metric with illumination and area based importance.

4 Rendering Optimizations

The previous section described how to stratify the environment map based on our importance metric. In this section, we describe how this stratification can be used to efficiently render arbitrary geometry with complex BRDFs.

Let us first consider Lambertian surfaces for which we are evaluating Equation 1 scaled by the diffuse reflectance. A straightforward evaluation of this integral would be to pick a random location within each stratum and evaluate the illumination and BRDF (cosine term for Lambertian surfaces) at this location, as well as the visibility S using ray tracing. An unbiased estimate of the contribution from that stratum can then be obtained by scaling this result by the area of the stratum. This approach improves on naive Monte Carlo methods since we have a good stratification of the environment map, based on our importance metric.

However, we can improve on this as a consequence of our stratification, which enables a number of optimizations that are difficult to include in standard importance sampling methods. Some of these introduce a bias (but maintain consistency) in order to significantly decrease variance, and are "bias for variance" optimizations.

Pre-integrating the illumination: The first optimization is pre-integrating the illumination within each stratum—effectively combining all the pixels in the stratum into a single directional light source located at the center. This method introduces bias by ignoring the variation of the BRDF and the surface orientation within each stratum. The surface orientation could be included using the nine coefficient spherical harmonics approximation of Ramamoorthi and Hanrahan [2001]. However, in practice, our strata are sufficiently small that the error of just using a directional light source is negligible. An exception is highly glossy materials for which the BRDF itself can vary significantly within the stratum. This makes the point source approximation less accurate, and it would be better to use a large number of spherical harmonic coefficients as described in [Ramamoorthi and Hanrahan 2002].

Since this optimization creates a set of directional lights, it also allows us to use environment maps with unmodified renderers. We evaluate the BRDF for the directional light at the center of the stratum and add the pre-integrated illumination in case the stratum is visible. This integration technique adds extra bias, but it eliminates sampling noise and makes the evaluation of the illumination completely deterministic. These noise-reducing optimizations are not easy to incorporate into standard importance sampling since naive approaches to stratification do not give compact regular strata limited to a single light source.

Jittering: One disadvantage with the above approach is that a small number of strata can introduce unwanted banding near shadow boundaries due to strata centers suddenly becoming visible or invisible. This banding can be eliminated by jittering the direction of the ray that is used to test visibility, by randomly choosing a location in the stratum—this eliminates banding, and instead adds noise. Note that, unlike standard Monte Carlo evaluation, this noise is added only where the visibility varies, such as shadow boundaries, since we are pre-integrating the illumination contribution of the stratum.

Sorting: The third optimization is sorting and adaptively sampling the strata based on the potential contribution taking surface orientation and the BRDF into account. For this purpose, we use a variant of Ward's adaptive shadow testing method [Ward 1991] which samples all the lights deterministically in order of contribution until the contrast that the remaining lights can add is below a certain threshold. In Ward's method, the contribution from the remaining lights is added based on prior statistics from the sampling. This approach is less useful to us since we have light entering from all directions and the contribution from the different strata can vary significantly from one location of an object to another. Instead, we randomly sample a fraction of the remaining lights, and use the fractional visibility obtained by this sampling as an estimate of the visibility of the remaining unsampled lights. This optimization is particularly powerful for glossy materials where only a few strata contribute significantly to the reflected radiance. Again, this optimization is difficult to include with importance sampling, since there is no notion of directional lights.

5 Results and Discussion

We show the results of our implementation of structured importance sampling and compare it to previous Monte Carlo techniques as well as LightGen [Cohen and Debevec 2001]. All images have been rendered on a 2.4GHz P4 PC using a Monte Carlo ray tracer.

We use three different Monte Carlo techniques for comparison with our results. Stratified sampling involves sampling visibility by generating ray directions that are distributed uniformly over the visible hemisphere. Importance sampling is implemented by considering the environment map in raster scan order as a one dimensional function and points are sampled on the sphere with probability proportional to their irradiance. Stratified importance sampling uses a stratified random number generator combined with standard importance sampling–This results in a superior sample distribution as compared to pure importance sampling. Finally we compare our results with those produced by using LightGen. LightGen is an unpublished environment map approximation algorithm which clusters the environment map by using a weighted spherical k-means algorithm. The irradiance of a pixel is used as its weight. The output of the algorithm is a set of directional light sources.

Figure 5 compares structured importance sampling with the existing rendering techniques on a teapot illuminated in Galileo's tomb [Debevec 1998]. As the images demonstrate, structured importance sampling produces significantly better results than stratified importance sampling as well as LightGen. Stratified importance sampling is still noisy with 300 samples, and to obtain reasonably noise free results similar to our method requires at least 3000 samples. The figure also shows the sample distribution produced by LightGen as well as our method for this map, and these two distributions demonstrate that that structured importance sampling gives better results due to a detailed sampling of the lights in the model, while LightGen's samples are closer to an area-based stratified sampling of the map. To check the convergence of LightGen, we created 3000 lights (which took 10 hours), but to our surprise having 10 times as many lights did not improve the quality much. We suspect this is because k-means is an iterative local search based clustering algorithm and it got stuck in a local minimum. The structured importance sampling images were rendered by pre-integrating the illumination in each stratum and treating it as a directional source. Note that we did not use sorting or jittering in Figure 5.

The use of jittering makes it possible to reduce the number of samples even further as shown in Figure 7. Here, the shadow from the teapot has been rendered using just 50 samples for the entire map. This results in banding if these samples are used as point-lights, but as shown in the figure this banding can be eliminated at the cost of noise along the shadow boundaries by jittering the direction of the ray used for shadow testing.

The Galileo map was pre-processed at its full resolution of 1024×1024 to produce 300 lights. Our stratification algorithm took about a minute to process the map while LightGen took about an hour. For Figure 5 the rendering time for the full teapot image at 500×500 with 300 samples was 10 seconds with structured importance sampling (as well as LightGen), while it was 70 seconds

for stratified importance sampling (the increased time is due to the search for the next sample in a histogram of the environment map). Sorting the samples on the teapot enabled us to reduce the number of samples by 75%, however the total render time was nearly the same since the overhead of sorting outweighed the advantage of using fewer sample rays. Interestingly, our experiments also showed that the lights produced by LightGen are much less amenable to sorting for the Galileo map, since they mostly have uniform intensity.

The second test scene shown in Figure 6 (and Figure 1) compares the convergence rates of structured importance sampling and stratified importance sampling for a glossy buddha in the Grace cathedral. This figure shows how structured importance sampling requires an order of magnitude fewer samples than stratified importance sampling to generate images of comparable quality. Even at 1000 samples per pixel, the best Monte Carlo techniques are noisy for this model. We also rendered an image by importance sampling based on the glossy BRDF (a normalized Phong model with an exponent of 50), but this image is even more noisy than that of importance sampling. It is possible to use the multiple importance sampling technique of Veach and Guibas [1995] to ensure that the best of either illumination based or BRDF based sampling is selected, but given that both illumination as well as BRDF based importance sampling perform much worse than structured importance sampling the combination of the two cannot perform better than our method.

The glossy buddha is a good example of a model where sorting can reduce the number of samples substantially by selecting and sampling only the few lights that contribute significantly to the BRDF. As shown in Figure 6, we were able to achieve an image with a quality matching the 300-samples version using an average of just 4.7 sample rays per pixel. Note that the number of sample rays for structured importance sampling and importance sampling is roughly half the number of samples reported, since on average only half of the selected sample directions are visible at a given point. Sorting reduced the rendering time for a 500×500 image of the buddha to 12 seconds while rendering with all 300 samples took 55 seconds. The rendering time using stratified importance sampling was 83 seconds.

Our final example is a complex snow covered mountain consisting of more than 2 million triangles illuminated at sunrise. This model has been rendered with full global illumination (using irradiance caching [Ward et al. 1988]). Global illumination and indirect lighting create a glow of indirect illumination on the left mountain, from the bright sunlit right mountain. For this 640x512 image with 4 samples per pixel the use of sorting reduced the render time by more than a factor of 3 making it 74 seconds instead of 233 seconds. In the model, sorting enabled a reduction in the number of sample rays, since the sun is the primary light source. Furthermore, sorting can be performed more aggressively for rays sampling indirect illumination which further reduces the number of secondary shadow rays.

6 Conclusions and Future Work

We have presented a new method, structured importance sampling, for integrating environment maps, taking visibility into account. For this purpose we have introduced—a novel visibility variance analysis that leads to a new metric for sampling illumination maps taking visibility into consideration, a hierarchical stratification algorithm for sampling the environment map according to our metric, and finally a number of rendering optimizations such as sorting and jittering making it possible to reduce the number of samples significantly and still obtain high quality images. The stratification algorithm is fast and deterministic, and the resulting strata can be used for animations of moving objects, changing materials, etc., illuminated by this map (this can be seen in parts of the animations on the CD-ROM that have been rendered using structured importance sampling).

In future work, we would like to apply structured importance sampling to rapid computation of surface light fields and other multidimensional data, integrating fast pre-filtering [Ramamoorthi and Hanrahan 2001; Ramamoorthi and Hanrahan 2002] and visibility computations. We would also like to extend our theoretical analysis of visibility into a complete statistical analysis of visibility maps, just as for images [Field 1987; Huang and Mumford 1999] and illumination [Dror et al. 2001]. Finally, we would like to integrate our technique into real-time rendering approaches to speed up the pre-computation phase of methods like Sloan et al. [2002], or to incorporate shadows into real-time environment mapping [Miller and Hoffman 1984; Greene 1986; Ramamoorthi and Hanrahan 2001].

7 Acknowledgments

This work was partially supported under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under contract No. W-7405-ENG-48 to S.B. The authors would like to thank the anonymous reviewers for their comments and suggestions and Theo Engell-Nielsen for help with producing the accompanying video. We would also like to thank Paul Debevec for providing us with high resolution environment maps and the University of Utah for the dataset for the mountain scene.

References

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

CABRAL, B., MAX, N., AND SPRINGMEYER, R. 1987. Bidirectional reflection functions from surface bump maps. In *SIGGRAPH 87*, 273–281.

CABRAL, B., OLANO, M., AND NEMEC, P. 1999. Reflection space image based rendering. In *SIGGRAPH 99*, 165–170.

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

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

DROR, R., LEUNG, T., ADELSON, E., AND WILLSKY, A. 2001. Statistics of real-world illumination. In *CVPR 01*, 164–171.

FEDER, T., AND GREENE, D. 1988. Optimal algorithms for approximate clustering. In *ACM Symposium on Theory of Computing*.

FIELD, D. 1987. Relations between the statistics of natural images and the response properties of cortical cells. *JOSA A*, 4, 2379–2394.

GONZALEZ, R. C., AND WOODS, R. E. 2001. *Digital Image Processing*. Prentice Hall.

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

HOCHBAUM, D., AND SHMOYS, D. 1985. A best possible heuristic for the k-center problem. *Mathematics of Operations Research*.

HUANG, J., AND MUMFORD, D. 1999. Statistics of natural images and models. In *CVPR*, 541–547.

KAJIYA, J. 1986. The rendering equation. In *SIGGRAPH 86*, 143–150.

KAUTZ, J., AND MCCOOL, M. 2000. Approximation of glossy reflection with prefiltered environment maps. In *Graphics Interface*, 119–126.

KAUTZ, J., VÁZQUEZ, P., HEIDRICH, W., AND SEIDEL, H. 2000. A unified approach to prefiltered environment maps. In *EGRW 00*, 185–196.

MILLER, G., AND HOFFMAN, C. 1984. Illumination and reflection maps: Simulated objects in simulated and real environments. *SIGGRAPH 84 Advanced Computer Graphics Animation seminar notes*.

NEMHAUSER, G. L., AND WOLSEY, L. A. 1988. *Integer and Combinatorial Optimization*. John Wiley and Sons, Inc.

RAMAMOORTHI, R., AND HANRAHAN, P. 2001. An efficient representation for irradiance environment maps. In *SIGGRAPH 01*, 497–500.

RAMAMOORTHI, R., AND HANRAHAN, P. 2002. Frequency space environment map rendering. In *SIGGRAPH 02*, 517–526.

SLOAN, P., KAUTZ, J., AND SNYDER, J. 2002. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. In *SIGGRAPH 02*, 527–536.

VEACH, E., AND GUIBAS, L. J. 1995. Optimally combining sampling techniques for monte carlo rendering. In *SIGGRAPH '95*, 419–428.

WARD, G. J., RUBINSTEIN, F. M., AND CLEAR, R. D. 1988. A ray tracing solution for diffuse interreflection. In *Computer Graphics (SIGGRAPH '88 Proceedings)*, J. Dill, Ed., vol. 22, 85–92.

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

Appendix

Standard Monte Carlo theory tells us that the variance of a function (random variable) is given by

$$V[S] = E[S^2] - E[S]^2. \tag{11}$$

Since shadows are binary, $S^2 = S$, and $E[S^2] = E[S]$. Letting $\beta = E[S]$, we get $V[S] = \beta - \beta^2 = \beta(1-\beta)$.

We want to compute the expected variance (for one sample) of the visibility function S for a small solid angle $\Delta\omega$. The expected value of the variance (where for clarity, we denote expected values over all regions by an overline) will be given by

$$\overline{V[S,\Delta\omega]} = \overline{E[S,\Delta\omega] - E[S,\Delta\omega]^2} = \overline{\beta(\Delta\omega)(1-\beta(\Delta\omega))}, \tag{12}$$

where we define $\beta(\Delta\omega)$ as the average or expected value of S over solid angle $\Delta\omega$.

The key expression above is $\beta(\Delta\omega)(1-\beta(\Delta\omega))$. Consider the case when $\beta = 1$ (the entire region of interest is visible). The above expression is then 0, and there is no variance. Similarly, if $\beta = 0$, the variance is 0. In fact, the worst case is $\beta = 1/2$, corresponding to a variance of $1/4$. Over the entire image, we will assume the worst case of random visibility, $P(S=0) = P(S=1) = 1/2$.

We want to consider the expected variance as a function of solid angle $\Delta\omega$. The intuition is that as $\Delta\omega$ becomes smaller than the feature size of visibility, the probability distribution for $\beta(\Delta\omega)$ is bimodal, i.e. either $\beta = 0$ or $\beta = 1$, and the corresponding variance tends to zero. Note that we must consider the average or expected value of the variance $\overline{\beta(1-\beta)}$ and not separately $\overline{\beta} \times \overline{1-\beta}$, which tends to $1/4$ as $\Delta\omega$ tends to 0, since β has equal probability of being 0 or 1.

To analyze further, we assume a correlation model for visibility, as per Equation 5. Now assume that the central point of the region of interest, subtending solid angle $\Delta\omega$, is visible, i.e. $S(0) = 1$. Analysis with $S(0) = 0$ is symmetric. We will assume that the visibility at a point making an angle θ with the central point is described using the simple correlation model above, so that the expected value, given $S(0) = 1$ is simply $\alpha(\theta) = \frac{1}{2}(1 + \tilde{\alpha}(\theta))$. In other words,

$$E[S(\theta) \mid S(0) = 1] = \frac{1}{2}\left(1 + e^{-\theta/T}\right), \tag{13}$$

which, as expected, tends to 1 as $\theta \to 0$, and tends to $1/2$ as θ becomes large.

Now, all that remains is to compute the expected variance. For this, it suffices to compute the expected value of $\beta(\Delta\omega)$. The variance, under the assumption $S(0) = 1$ will simply be $\beta(\Delta\omega)(1 - \beta(\Delta\omega))$. A similar argument holds for $S(0) = 0$, so this is the quantity we seek. Now, $\beta(\Delta\omega)$ is the expected value of S over the whole solid angle $\Delta\omega$, which is the same as integrating the expected value of $S(\theta)$ at each point. Finally, the solid angle $\Delta\omega = \pi\theta^2$, assuming the maximum angle θ with respect to the central point is small.

$$\begin{aligned} \beta(\Delta\omega) &= \frac{1}{\pi\theta^2}\int_0^\theta \int_0^{2\pi} \frac{1}{2}\left(1 + e^{-u/T}\right) u\, du\, d\phi \\ &= \frac{1}{2} - \frac{T}{\theta}e^{-\theta/T} + \frac{T^2}{\theta^2}(1 - e^{-\theta/T}). \end{aligned} \tag{14}$$

Assuming, $\theta \ll T$ and taking the Taylor series expansion for $e^{-\theta/T}$ we obtain

$$\begin{aligned} \beta &\approx \frac{1}{2} - \frac{T}{\theta}\left(1 - \frac{\theta}{T} + \frac{\theta^2}{2T^2}\right) + \frac{T^2}{\theta^2}\left(\frac{\theta}{T} - \frac{\theta^2}{2T^2} + \frac{\theta^3}{6T^3}\right) \\ &\approx 1 - \frac{\theta}{3T}\,. \end{aligned} \tag{15}$$

Now, the variance is given by $\beta(\Delta\omega)(1-\beta(\Delta\omega))$ and this becomes

$$\overline{V[S,\Delta\omega]} \approx \frac{\theta}{3T}(1 - \frac{\theta}{3T}) \approx \frac{\theta}{3T}. \tag{16}$$

Note that in these equations, the solid angle for small θ is given by $\Delta\omega = \pi\theta^2$, so $\theta = \sqrt{\Delta\omega/\pi}$, and the variance is proportional to the square root of the solid angle.

Finally, we seek to determine the optimal distribution of samples, and we will do so by attempting to minimize the variance of the net integral. For this, consider two regions of variance V_1 and V_2, with N samples to be distributed between them in the ratio ρN and $(1-\rho)N$. Assuming variance decreases at the rate of N^{-1},

$$V[N] = \frac{1}{N}\left(\frac{V_1}{\rho} + \frac{V_2}{(1-\rho)}\right), \tag{17}$$

which we can differentiate with respect to ρ, obtaining

$$\frac{\rho}{1-\rho} = \sqrt{\frac{V_1}{V_2}}, \tag{18}$$

and with more than two regions, this ratio must generally be followed, i.e. the number of samples is proportional to $\sqrt{V}$. As an example, consider $V = L^2$, assuming simple scaling of variance by illumination magnitude. In that case, the number of samples $\rho \sim L$, as for standard importance-based stratification. Now, consider the variance in a region of solid angle $\Delta\omega$. The visibility variance is proportional to $\Delta\omega^{1/2}$, but the net variance must be scaled by net illumination intensity L^2, assuming uniform lighting (this assumption also builds on our hierarchical thresholding scheme which creates levels of approximately equal illumination intensity). Hence, the number of samples is proportional to $\sqrt{L^2\Delta\omega^{1/2}} = L\Delta\omega^{1/4}$.

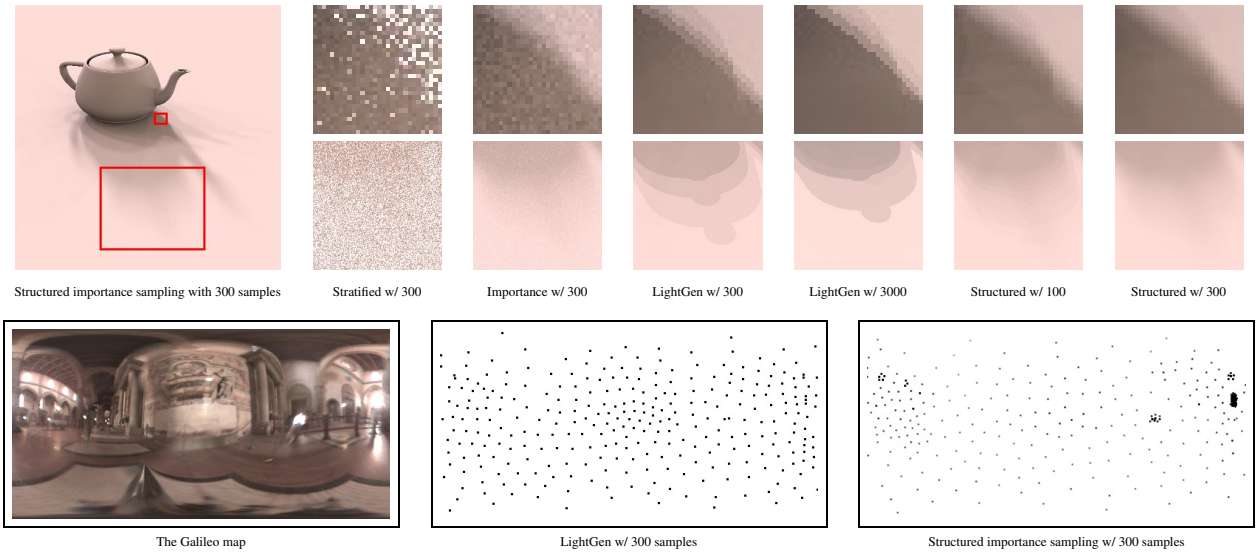


Figure 5: *A teapot in Galileo's tomb rendered using different sampling strategies. No sorting or jittering has been used for this comparison. The large image in the top row has been computed using structured importance sampling with 300 samples, which we verified to be indistinguishable from a reference image computed with 100,000 samples using standard Monte Carlo sampling. The red squares show two regions that have been rendered using different sampling techniques as close-ups in the small images on the right. From left to right these images have been rendered using naive stratified sampling, illumination based stratified importance sampling and using LightGen with 300 samples, LightGen with 3000 samples, structured importance sampling with 100 samples and 300 samples. Both Monte Carlo techniques produce significant statistical noise even for this simple model, LightGen shows banding in the shadows with both 300 and 3000 samples (since too few samples are placed at the bright lights), structured importance sampling looks convincing with just 100 samples and with 300 samples the result is indistinguishable from a reference image. The bottom row shows from left to right, the Galileo map, the lights created by LightGen, and the stratum centers created using our method. Note how our stratification method samples the bright lights much more densely than LightGen. This is the reason why the shadows with structured importance sampling are more accurate.*

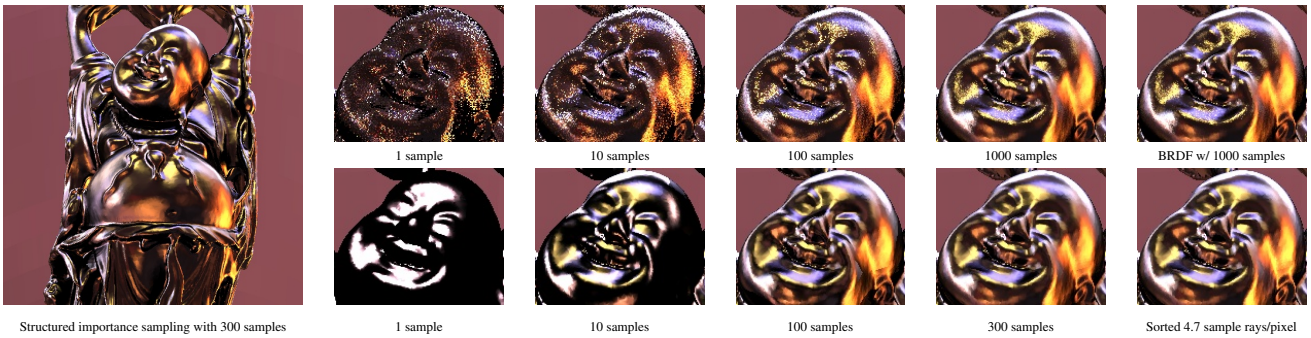


Figure 6: *A glossy buddha in the Grace environment map. The large image on the left is our sampling technique with 300 samples, which is practically indistinguishable from a reference image. The two rows show close-ups of the head rendered with an increasing number of samples. The top row is stratified importance sampling with 1, 10, 100 and 1000 samples as well as BRDF based importance sampling with 1000 samples. The bottom row shows structured importance sampling with 1, 10, 100, and 300 samples per pixel, as well as a version rendered with sorting and thresholding resulting in an average of just 4.7 samples per pixel. Note how structured importance sampling results in noise free images and quickly converges to the final result while the best Monte Carlo sampling techniques are noisy even when using 1000 samples.*

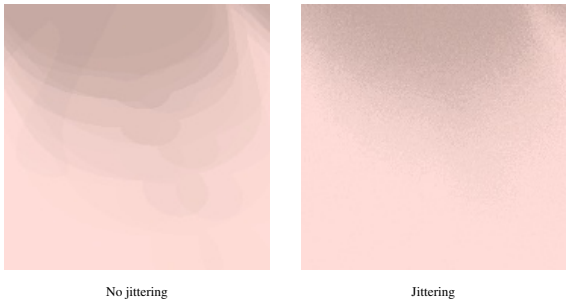


Figure 7: *Jittering can be used to eliminate banding at low sample counts at the cost of adding noise along the shadow boundaries. This image is the same close-up of the shadow as in Figure 5 using just 50 samples. The image on the left is without jittering and the image on the right has been rendered using jittering of the shadow ray.*

Figure 8: *A snow covered mountain model illuminated at sunrise. This model has more than 2 million triangles, and the image has been rendered in 640x512 with full global illumination in 75 seconds.*