Production Monte Carlo Rendering

Villemin, Ryusuke
Pixar Animation Studios

出版情報：MI lecture note series. 64, pp.80-88, 2015-09-18. 九州大学マス・フォア・インダストリ研究所

MATHEMATICAL PROGRESS IN EXPRESSIVE IMAGE SYNTHESIS 2015
Sep 25-27, 2015, Nishijin Plaza Kyushu University, Fukuoka

Production Monte Carlo Rendering

Ryusuke Villemin*
Pixar Animation Studios

Abstract Production rendering changed radically in last few years. With ever increasing computing power, it is now possible to use raytracing and physically based lighting and shading in production. The tools evolved (surfaces are now using normalized brdfs, lighting is using sampled arealights) and rendering is becoming more and more a sampling problem. Even in real-time rendering, we begin to see the use of such algorithms (like sampled ambient occlusion). Monte Carlo integration is the most common method of solving the rendering equation. Although the basic version does not have a remarkable convergence behaviour, numerous techniques have been added to the toolbox of the shader writer to make it practical.

Keywords: Monte Carlo, Sampling, Statistical, Lighting, Shading, Raytracing

1 Introduction

Rendering is rapidly becoming a sampling problem. In the first part of this paper, we will first present and expose the problem, which has a very compact and clean form in the rendering equation. Then we will show some of the most common techniques used in production: importance sampling and multiple importance sampling. Finally, we will go through a few practical cases, direct lighting, and bsdf and volume sampling.

2 Rendering Equation

The rendering equation was first introduced by Kajiya [7], in the following form:

$$L_o(x,\omega) = L_e(x,\omega) + L_r(x,\omega) \tag{1}$$

$$L_o(x,\omega) = L_e(x,\omega) + \int_\Omega f_r(x,\omega',\omega) L_i(x,\omega')(\omega'.n)d\omega' \tag{2}$$

which can be reformulated as an area integral over all surfaces in the scene:

$$L_o(x',x) = L_e(x',x) + \int_S f_r(x'',x',x)L_i(x'',x)V(x'',x')G(x'',x')dA'' \tag{3}$$

*rvillemin@pixar.com

with

$$
\begin{aligned}
V(x', x) &= \text{1, if x' and x are mutually visible, 0 otherwise} && (4)\\
G(x', x) &= \frac{(\omega.n')(\omega'.n)}{||x' - x||^2} && (5)
\end{aligned}
$$

Depending on the sampling strategy used, one or the other formulation will be preferred.

Although this equation already presents a few interesting challenges, this is in fact a simplified version that does not account for volumetric effects or complex subsurface scattering. Readers interested in the full form should look at Glassner [2].

Table 1: Notation

L_o	outgoing radiance
L_i	incoming radiance
L_e	emitted radiance
f_r	brdf
x	position on surface
n	normal at x
ω	outgoing direction
ω'	incoming direction
Ω	hemishere fo directions

3 Monte Carlo Sampling

3.1 Introduction

A PDF (probability density function) p is a function that describes the distribution of values a random variable x can take on a domain X. It has 2 main properties:

$$\forall x \in X, p(x) \geq 0 \tag{6}$$

$$\int_X p(x) = 1 \tag{7}$$

Now let's say we want to evaluate the following integral:

$$F = \int_X f(x)dx \tag{8}$$

Even if we don't have an analytical solution, if we can evaluate f for any point in X, then we can compute F with Monte Carlo sampling. Suppose we have a random variable Y with a PDF p defined on X, and a function g such as the expected value of $g(Y)$ is F, then:

$$
\begin{aligned}
E[g(Y)] &= \int_X g(y)p(y)dy && (9)\\
&= \int_X f(y)dy && (10)\\
&= F && (11)
\end{aligned}
$$

The integration problem is now a mean estimation problem, if we take n samples, then the estimate will be:

$$\hat{F} = \frac{1}{n}\sum_{i=0}^{n-1} g(y_i) \tag{12}$$

where y_i are random values from the distribution p.

The estimate is unbiased:

$$\begin{aligned}
E[\hat{F}] &= \frac{1}{n}\sum_{i=0}^{n-1} E[g(Y_i)] &(13)\\
E[\hat{F}] &= E[g(Y)] &(14)\\
&= \int_X f(x)dx &(15)
\end{aligned}$$

And the variance is:

$$\begin{aligned}
E[\hat{F}] &= \frac{1}{n}\sum_{i=0}^{n-1} V[g(Y_i)] &(16)\\
E[\hat{F}] &= V[g(Y)] &(17)\\
&= \int_X \left(g(x) - \int_X f(t)dt\right)^2 dx &(18)
\end{aligned}$$

3.2 Importance Sampling

In the previous section, we showed a way to compute the integral:

$$F = \int_X g(x)p(x)dx \tag{19}$$

Unfortunately this is not exactly the equation we are after, but a simple rewrite can get us back to the original problem:

$$\begin{aligned}
F &= \int_X f(x)dx &(20)\\
&= \int_X \left(\frac{f(x)}{p(x)}\right) p(x)dy &(21)
\end{aligned}$$

then the estimate is :

$$\hat{F} = \frac{1}{n}\sum_{i=0}^{n-1} \frac{f(y_i)}{p(y_i)} \tag{22}$$

To have a good estimate, we can increase the number of samples n, but also try to reduce the variance of each f/p (p should match f). We still need a way to generate those y_i samples based on p, a few techniques are available to us, the most interesting one and usable in practice, is the CDF inversion.

3.3 CDF Inversion

One question is: how do we distribute samples according to $p(y)$? For the random variable Y, the CDF (cumulative distribution function) is defined as :

$$P_Y(y) = P_{probalility}(Y \leq y) = \int_{y_{min}}^{y} p(x)dx \tag{23}$$

If Y is a scalar random variable with CDF P_Y, then the random variable U defined by:

$$U = P_Y(Y) \tag{24}$$

is the uniform distribution. If we invert this equation:

$$Y = P_Y^{-1}(U) \tag{25}$$

then Y is distributed with density p. This transformation is called the inverse CDF technique. This means that if you can compute the inverse of the integral of a PDF, then you can use the inverse CDF method. Other methods like rejection sampling are also available, but not very popular in practice, since there is no way to bound the computational cost, which is always a variable we try to minimize in rendering.

3.4 Multiple Importance Sampling

To use the CDF inversion method efficiently we need to find a PDF that will closely match the function f we are trying to integrate. Unfortunately in real scenarios, it is usually not possible to find such functions, but instead we can find multiple PDFs, each of them roughly matching some part of f. We could let the user choose the best technique for each scenario but it's not really practical. Therefore we use multiple importance sampling (MIS) to automatically combine PDFs. If we have M techniques sampling N samples, then:

$$\hat{F} = \frac{1}{N} \sum_{j=0}^{M-1} \sum_{i=0}^{N_j} \omega_j(y_i) \frac{f(y_i)}{p_j(y_i)} \tag{26}$$

the estimate will remain unbiased if the weights obey the following:

$$\sum_{j=0}^{M-1} \omega_j(y) = 1 \tag{27}$$

we could just take $\omega_j = 1/M$ but this would just give an average of all the techniques, Veach [11] instead proposed a very simple, yet efficient method called the balance heuristic:

$$\omega_i(y) = \frac{p_i(y)}{\sum_{j=0}^{M-1} p_j(y)} \tag{28}$$

4 Applications

4.1 Direct Lighting

For direct lighting, we have the following equation which is a further simplified version of the general rendering equation:

$$L_o(x, \omega) = \int_\Omega f_r(x, \omega', \omega) L_{light}(x, \omega')(\omega'.n) d\omega' \tag{29}$$

and in the three-point form:

$$L_o(x', x) = \int_S f_r(x'', x', x) L_{light}(x'', x) V(x'', x') G(x'', x') dA'' \tag{30}$$

L_i in the integral is replaced by L_{light}, the energy emitted directly by light sources. The implicit recursion is gone and what's left is just a 2D integral of a product of 3 functions L_{light}, f_r, and V. The visibility V is scene dependent and usually there is no good way to compute it except to sample blindly; however we can use our knowledge of the material f_r and the light source L_{light} to guide our sampling. And by using multiple sampling, we don't have to choose between the two.

Here is an example of a dome light, and a GGX specular brdf, using light importance sampling for infinite arealights [10], and microfacets based sampling [5] for the brdf:

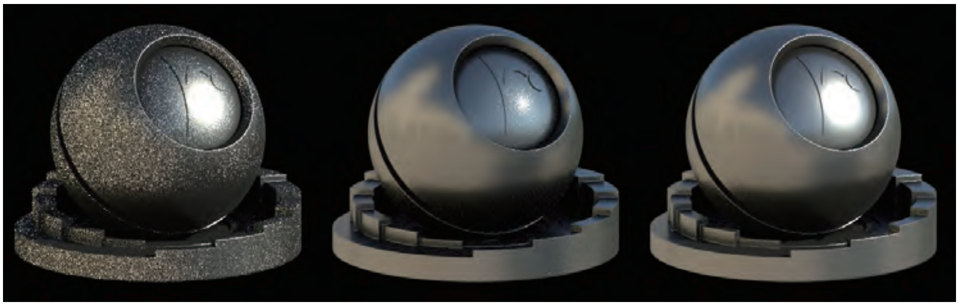

Figure 1: LightSampling, BrdfSampling, MIS(from left to right, equal time renders)

When using only brdf sampling the reflections (Eq. 29) are clean but not the highlights, with light sampling only (Eq. 30), the highlights are nicely resolved but the rest is noisy. Because each sampling technique covers different parts of the integral, when using MIS we get the best result everywhere. We are effectively doubling the number of techniques, since we are shooting twice the number of samples, but even with this additional cost the result is more converged at equal render time (Fig. 1).

4.2 BSDF Sampling

In a pathtracer, to maintain good interactivity, we usually want only one path per iteration, i-e a camera ray hitting a surface will generate only one indirect ray to continue the path. However a material used in production will have multiple lobes, such as clearcoat, diffuse, specular, ..., so we need to choose one at every hit. We could just pick and evaluate one lobe, but one way to further reduce the variance is to use one-sample MIS [11]. The computation is the same as normal MIS, the only difference is that here we will pick one sampling strategy j with probability c_j to choose our unique sample.

$$F(y_i) = \frac{\omega_j f(y_i)}{c_j p_j(y_i)} \tag{31}$$

We will choose one lobe to pick the direction and still compute the whole contribution, and then use the balance heuristic on the PDF. For a brdf with a diffuse, specular and clearcoat:

$$f_r(x,\omega,\omega') = f_d(x,\omega,\omega') + f_s(x,\omega,\omega') + f_c(x,\omega,\omega') \tag{32}$$

In this example, we have 3 sampling strategies we can choose from, choosing c_j proportional to each albedo, the final PDF p will be independent of the chosen strategy:

$$\frac{1}{p(x)} = \frac{1}{c_d p_d(x)} \frac{c_d p_d(x)}{c_d p_d(x) + c_s p_s(x) + c_c p_c(x)} \text{when choosing diffuse} \tag{33}$$

$$\frac{1}{p(x)} = \frac{1}{c_s p_s(x)} \frac{c_s p_s(x)}{c_d p_d(x) + c_s p_s(x) + c_c p_c(x)} \text{when choosing specular} \tag{34}$$

$$\frac{1}{p(x)} = \frac{1}{c_c p_c(x)} \frac{c_c p_c(x)}{c_d p_d(x) + c_s p_s(x) + c_c p_c(x)} \text{when choosing clearcoat} \tag{35}$$

$$\frac{1}{p(x)} = \frac{1}{c_d p_d(x) + c_s p_s(x) + c_c p_c(x)} \tag{36}$$

The cost here is virtually the same as brdf evaluation and sampling represent a very small portion of an actual render ($< 1\%$), for a cleaner result (Fig. 2).

4.3 Volume Sampling

Another interesting use case of one sample MIS is volume sampling. The simplest way to sample a volume is to use density sampling, i-e sample the transmittance. The contribution of a ray passing though a scattering volume is :

$$L_e(x',\omega) = \int_S \tau(t)\epsilon(x',\omega,t)dt \tag{37}$$

with the transmittance $\tau(t)$ defined as:

$$\tau(t) = \int_{t'=0}^{t} e^{-\sigma_t t'} dt' \tag{38}$$

if we use a PDF proportional to τ, for a random number χ, the CDF inversion gives us:

$$t = \frac{-1}{\sigma_t} \ln(1 - (1 - e^{-\sigma_t S})\chi) \tag{39}$$

the problem here is that the extinction σ_t is usually a color, so we need to choose one channel to do the sampling. We can pick the max, the min of some kind of average, but whatever choice we

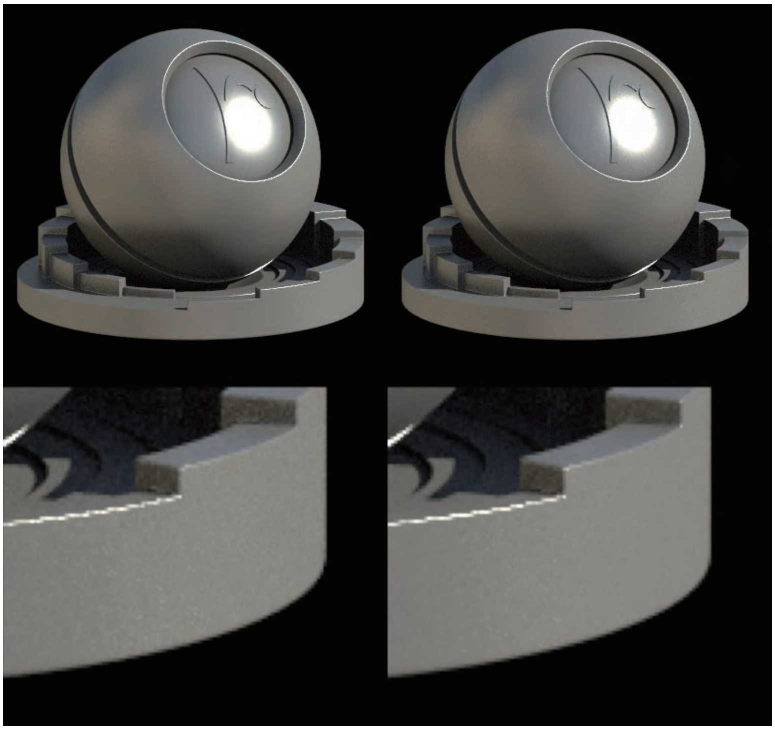

Figure 2: single lobe sampling, one-sample MIS (from left to right, equal time renders)

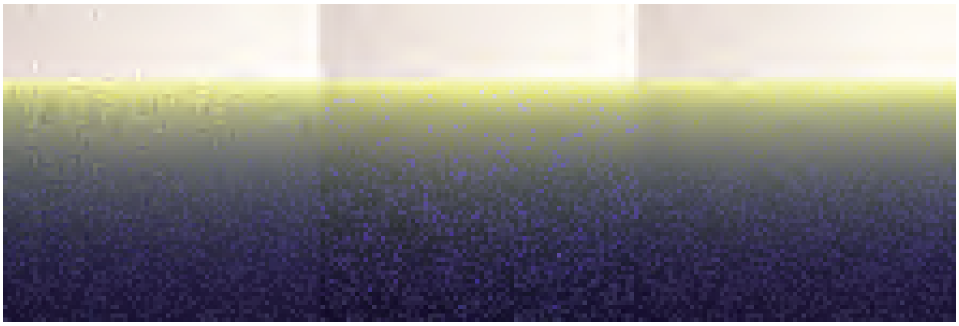

Figure 3: Slice of a homogeneous volume illuminated from above: max sampling, min-sampling, MIS (from left to right, equal time renders)

make there are cases where the PDF does not match the extinction anymore and gives high variance results. The problem becomes bigger the more satured exctinction is. Here again, by doing MIS between channels we can get a better image.

When choosing the maximum extinction, the shallow scattering is not resolved enough and we get a noisy yellow scattering. If on the other side we choose to sample the minimum extinction, then in that case the deep scattering is not sampled correctly and we get blue fireflies. The MIS image shows a nice convergence thoughout the thickness of the volume (Fig. 3).

5 Conclusion

At first glance it may seem that we only use very basic sampling techniques in production, whereas every year we see novel new algorithms being published. There are a few reasons for this. Contrary to architectural renders, time contraints are severe in movie production where we have to render animations, and even with large render farms, single frame renders need to stay typically within a few hours to few dozen hours. The other important limitation is that when working interactively, lighters get noisy unconverged images quickly but these should nonetheless give them a good estimate of the final result. That means the noise needs to be predictable, and this is one of the reasons we have yet to see good usage of MCMC techniques in production rendering.

One additional variable is that a large part of the scene construction is up to the artist, who can use procedural geometry and patterns on surfaces and volumes, and this introduces complexity and unpredictability.

Only a few algorithms can fulfill all the above contraints and MIS is one of them. It is still possible to sometimes find special cases that are important enough to get special treatment (for example, usage of control variates for diffuse and domelights [6]), but those are the exception. And even MIS is not without problems, because it's just a combination of many sampling techniques, its cost increases with the number of strategies. We are still therefore looking to try new techniques, from the light integration point of view (better integration of difficult light paths [3] [4]), to better bsdf [1], hair [9], volume models, but also more fundamental sampling improvements [8]. Fortunately experimentation is now much easier since the whole rendering pipeline has moved to raytracing and physically based rendering.

References

[1] Per H. Christensen, Brent Burley: Approximate Reflectance Profiles for Efficient Subsurface Scattering *Siggraph 2015*: `http://graphics.pixar.com/library/ApproxBSSRDF/SIGGRAPH_Tech_Talk.pdf`, 2015.

[2] A.S. Glassner: Principles of Digital Image Synthesis 1. *Morgan Kaufmann*: `http://books.google.com/books?id=wOt9IBunyrkC`, 1995.

[3] Iliyan Georgiev, Jaroslav Krivnek, Toms Davidovis, and Philipp Slusallek: Light Transport Simulation with Vertex Connection and Merging *Siggraph Asia 2012*: `http://cgg.mff.cuni.cz/ jaroslav/papers/2012-vcm/`, 2012.

[4] T. Hachisuka, J. Pantaleoni, and H. W. Jensen: A Path Space Extension for Robust Light Transport Simulation *Siggraph Asia 2012*: `http://www.ci.i.u-tokyo.ac.jp/ hachisuka/ups.pdf`, 2012.

[5] Eric Heitz and Eugene dÉon: Importance Sampling Microfacet-Based BSDFs using the Distribution of Visible Normals *EGSR 2014*: `http://hal.inria.fr/hal-00996995/en`, 2014.

[6] Christophe Hery and Ryusuke Villemin: Physically Based Lighting at Pixar *Siggraph Course 2013*: `http://graphics.pixar.com/library/PhysicallyBasedLighting/paper.pdf`, 2013.

[7] James T. Kajiya: The Rendering Equation. In *SIGGRAPH Comput. Graph.*: `http://portal.acm.org`, 1986.

[8] Andrew Kensler: Correlated Multi-Jittered Sampling. In *Pixar Technical Memo.*: `http://graphics.pixar.com/library/MultiJitteredSampling/paper.pdf`, 2013.

[9] Leonid Pekelis, Christophe Hery, Ryusuke Villemin, Junyi Ling: A Data-Driven Light Scattering Model for Hair. In *Pixar Technical Memo.*: `http://graphics.pixar.com/library/DataDrivenHairScattering/paper.pdf`, 2015.

[10] Matt Pharr and Greg Humphreys: Infinite Area Light Source with Importance Sampling: `http://www.pbrt.org/plugins/infinitesample.pdf`, 2004.

[11] Eric Veach and Leonidas J. Guibas: Optimally combining sampling techniques for Monte Carlo rendering *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*: `http://doi.acm.org/10.1145/218380.218498`, 1995.