Bidirectional Path Tracing Using Backward Stochastic Light Culling (Supplementary Material)

Yusuke Tokuyoshi
SQUARE ENIX CO., LTD.
tokuyosh@square-enix.com

Takahiro Harada
Advanced Micro Devices, Inc.
Takahiro.Harada@amd.com

ACM Reference Format:
Yusuke Tokuyoshi and Takahiro Harada. 2018. Bidirectional Path Tracing Using Backward Stochastic Light Culling (Supplementary Material). In *Proceedings of SIGGRAPH '18 Talks.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3214745.3214750

1 CONSTANT TERM TO CONTROL THE VARIANCE

This paper samples connections via backward stochastic light culling and naïve uniform sampling for glossy reflections and other scattering effects, respectively. These two sampling techniques are combined using multiple importance sampling (MIS) [10]. For backward stochastic light culling, the probability of Russian roulette is

$$p\left(\mathbf{x}, \mathbf{y}_i\right) = \min\left(\frac{c_i f\left(\mathbf{x}, \boldsymbol{\omega}', \boldsymbol{\omega}_i\right) \max\left(\boldsymbol{\omega}_i \cdot \mathbf{n}, 0\right)}{\|\mathbf{x} - \mathbf{y}_i\|^2}, 1\right), \tag{1}$$

where $\mathbf{x}$ is the eye vertex, $\mathbf{y}_i$ is the ith light vertex in a light vertex cache, $f(\cdot)$ is the specular BRDF, $\boldsymbol{\omega}'$ is the direction from $\mathbf{x}$ to the previous vertex in the eye subpath, $\boldsymbol{\omega}_i$ is the direction from $\mathbf{x}$ to $\mathbf{y}_i$, $\mathbf{n}$ is the surface normal at $\mathbf{x}$, and c_i is a constant term to control the variance. For ease of parameter tuning, this paper uses the following c_i:

$$\boxed{c_i = \frac{\Phi_i k_i}{M\delta}}, \tag{2}$$

where Φ_i is the radiant flux of $\mathbf{y}_i$ that is propagated by light subpath tracing, k_i is the reflectance at $\mathbf{y}_i$, M is the number of light subpaths, and δ is the user-specified parameter to control variance ($\delta = 16$ in this paper). Since this Russian roulette is performed for all the light vertices (i.e., all the light subpaths), the sampling density is given as

$$Mp\left(\mathbf{x}, \mathbf{y}_i\right) = \min\left(\frac{\Phi_i k_i f\left(\mathbf{x}, \boldsymbol{\omega}', \boldsymbol{\omega}_i\right) \max\left(\mathbf{n} \cdot \boldsymbol{\omega}_i, 0\right)}{\delta\|\mathbf{x} - \mathbf{y}_i\|^2}, M\right). \tag{3}$$

When $M \to \infty$, this density will be independent from M. In other words, the number of sampled light vertices is sublinear to M. Hence, our method is efficient for many light vertices. Eq. (3) is also used to calculate the weight of MIS.

Option for Simplification. If a recursive calculation is used for the MIS weight [4, 9], we define Φ_i as the total radiant flux of light sources (which is a vertex-independent value). We can also omit k_i from Eq. (2) so that c_i is an identical value. When using an identical c_i, it is not necessary to store the maximum of c_i into the tree.

2 BOUNDING ELLIPSOID FOR GLOSSY REFLECTION

2.1 Classic Phong BRDF

The classic Phong BRDF [5] is modeled as

$$f\left(\mathbf{x}, \boldsymbol{\omega}', \boldsymbol{\omega}\right) = \frac{\max\left(\boldsymbol{\omega} \cdot \boldsymbol{\omega}_u, 0\right)^\eta}{|\boldsymbol{\omega} \cdot \mathbf{n}|}, \tag{4}$$

where $\boldsymbol{\omega}_u$ is the perfect specular reflection direction, and η is the Phong exponent. The bounding ellipsoid for this BRDF model was derived by Dachsbacher and Stamminger [3]. The semiaxes of this ellipsoid are

$$\boxed{\begin{bmatrix} r_u \\ r_v \\ r_w \end{bmatrix} = \bar{r}_i \begin{bmatrix} \max(t, 1-t) \\ W(t) \\ W(t) \end{bmatrix}}, \tag{5}$$

where $\bar{r}_i$ is the size of the ellipsoid ($\bar{r}_i = \sqrt{\frac{c_i}{\xi_i}}$ in this paper), $W(\cdot)$ is the width of the isosurface of the Phong reflection, and t is the parameter that maximizes $W(t)$. These t and $W(t)$ are given by the following equations:

$$t = \left(\frac{\eta}{\eta + 2}\right)^{\frac{\eta+2}{4}}, \tag{6}$$

$$W(t) = t^{\frac{\eta}{\eta+2}}\sqrt{1 - t^{\frac{4}{\eta+2}}}. \tag{7}$$

The rotation matrix of this ellipsoid is

$$\boxed{\mathbf{R} = \begin{bmatrix} \boldsymbol{\omega}_u & \boldsymbol{\omega}_w \times \boldsymbol{\omega}_u & \boldsymbol{\omega}_w \end{bmatrix}}, \tag{8}$$

where $\boldsymbol{\omega}_w$ is a unit vector orthogonal to $\boldsymbol{\omega}_u$ and $\mathbf{n}$ (i.e., $\boldsymbol{\omega}_w = \frac{\boldsymbol{\omega}_u \times \mathbf{n}}{\|\boldsymbol{\omega}_u \times \mathbf{n}\|}$ for $\boldsymbol{\omega}_u \neq \mathbf{n}$). The center of the ellipsoid is

$$\boxed{\mathbf{C} = \mathbf{x} + t\bar{r}_i\boldsymbol{\omega}_u}. \tag{9}$$

2.2 GGX Microfacet BRDF

The microfacet BRDF [2] is modeled as

$$f\left(\mathbf{x}, \boldsymbol{\omega}', \boldsymbol{\omega}\right) = \frac{F\left(\boldsymbol{\omega}' \cdot \boldsymbol{\omega}_h\right) G\left(\boldsymbol{\omega}', \boldsymbol{\omega}\right) D(\boldsymbol{\omega}_h \cdot \mathbf{n})}{4\left|\boldsymbol{\omega}' \cdot \mathbf{n}\right|\left|\boldsymbol{\omega} \cdot \mathbf{n}\right|}, \tag{10}$$

where $\boldsymbol{\omega}_h = \frac{\boldsymbol{\omega}' + \boldsymbol{\omega}}{\|\boldsymbol{\omega}' + \boldsymbol{\omega}\|}$ is the halfvector, $F\left(\boldsymbol{\omega}' \cdot \boldsymbol{\omega}_h\right) \in [0, 1]$ is the Fresnel factor, $G\left(\boldsymbol{\omega}', \boldsymbol{\omega}\right) \in [0, 1]$ is the masking-shadowing function, and $D(\boldsymbol{\omega}_h \cdot \mathbf{n}) \in [0, \infty]$ is the normal distribution function (NDF). The bounding ellipsoid for the microfacet BRDF model with the GGX NDF [8, 11] was derived by Tokuyoshi and Harada [7]. The semiaxes of this ellipsoid are

$$\boxed{\begin{bmatrix} r_u \\ r_v \\ r_w \end{bmatrix} = \bar{r}_i \begin{bmatrix} \frac{1+\alpha^2}{2\alpha} \\ 1 \\ 1 \end{bmatrix}}, \tag{11}$$

SIGGRAPH '18 Talks, 2018, Vancouver

where α is the GGX roughness parameter. The rotation matrix of this ellipsoid is the same as Eq. (8). The center of the ellipsoid is

$$\boxed{\mathbf{C} = \mathbf{x} + \frac{1-\alpha^2}{2\alpha}\bar{r}_i \boldsymbol{\omega}_u}. \tag{12}$$

For the microfacet BRDF, the size $\bar{r}_i$ takes the maximum of the Fresnel factor $F_{\max}(\boldsymbol{\omega}')$ and the maximum of the masking-shadowing function $G_{\max}(\boldsymbol{\omega}')$ as follows:

$$\bar{r}_i = \sqrt{\frac{c_i F_{\max}(\boldsymbol{\omega}')\, G_{\max}(\boldsymbol{\omega}')}{4\pi\xi_i\, |\boldsymbol{\omega}' \cdot \mathbf{n}|}}. \tag{13}$$

Although $F_{\max}(\boldsymbol{\omega}') = 1$ and $G_{\max}(\boldsymbol{\omega}') = 1$ can be used for the worst case, smaller upper bounds are available for some models.

Smith Masking Function. For the Smith microsurface model [6], the upper bound of the masking-shadowing function is obtained as follows:

$$\begin{aligned} G_{\max}(\boldsymbol{\omega}') &= \frac{|\boldsymbol{\omega}' \cdot \mathbf{n}|}{\int_{S^2} D(\boldsymbol{\omega} \cdot \mathbf{n}) \max(\boldsymbol{\omega}' \cdot \boldsymbol{\omega}, 0)\, \mathrm{d}\boldsymbol{\omega}} \\ &= \frac{2\,|\boldsymbol{\omega}' \cdot \mathbf{n}|}{|\boldsymbol{\omega}' \cdot \mathbf{n}| + \sqrt{(1-\alpha^2)(\boldsymbol{\omega}' \cdot \mathbf{n})^2 + \alpha^2}}. \end{aligned} \tag{14}$$

Schlick Fresnel Factor. For Schlick Fresnel factor, the upper bound is obtained by

$$F_{\max}(\boldsymbol{\omega}') = F\left(\cos\left(\theta_h^{\max}(\boldsymbol{\omega}')\right)\right), \tag{15}$$

where $\theta_h^{\max}(\boldsymbol{\omega}') = \frac{1}{2}\arccos(\boldsymbol{\omega}' \cdot \mathbf{n}) + \frac{\pi}{4}$ is the maximum of the halfvector angle, and $\cos\left(\theta_h^{\max}(\boldsymbol{\omega}')\right) = \sqrt{\frac{1-\sqrt{1-(\boldsymbol{\omega}' \cdot \mathbf{n})^2}}{2}}$.

Unpolarized Fresnel Factor for Dielectrics. For dielectrics with unpolarized light [2], the upper bound of the Fresnel factor is obtained by

$$F_{\max}(\boldsymbol{\omega}') = \max\left(F(1), F\left(\cos\left(\theta_h^{\max}(\boldsymbol{\omega}')\right)\right)\right). \tag{16}$$

2.3 Other Microfacet BRDFs with Bell-Shaped NDFs

Bell-shaped NDFs are often used for microfacet BRDFs. Since the GGX NDF is also bell-shaped, a bell-shaped NDF can be replaced with the GGX NDF in the probability of Russian roulette (Eq. (1)). For example, Beckman NDF [1] can be approximated with the GGX NDF using the same roughness parameter. Therefore, the ellipsoid mentioned in §2.2 is also used for other microfacet BRDFs.

3 MINIMUM RANDOM NUMBER CALCULATION

Subsection §3.1 first derives the $O(1)$ form to calculate the minimum of many non-stratified random numbers. Then, we describe practical issues for the on-the-fly random number assignment using this form in tree traversal. Subsection §3.2 describes how these issues are addressed in our semi-stratified random number assignment.

3.1 Minimum of Non-Stratified Random Numbers

3.1.1 Derivation. The probability that a uniform random number $z \in [0, 1)$ is the minimum within N random numbers is given by the following recursion:

$$P_N(z) = (1-z)P_{N-1}(z), \tag{17}$$

$$P_1(z) = 1, \tag{18}$$

where $1 - z$ is the probability that the other random number is larger than z. Expanding this recursion, we obtain the following probability:

$$P_N(z) = (1-z)^{N-1}. \tag{19}$$

Since z is uniformly distributed, the probability density function (PDF) of the minimum random number is obtained by normalizing Eq. (19) as follows:

$$\hat{P}_N(z) = \frac{P_N(z)}{\int_0^1 P_N(z')\,\mathrm{d}z'} = N(1-z)^{N-1}. \tag{20}$$

The cumulative distribution function (CDF) of this PDF is yielded as

$$C_N(z) = \int_0^z \hat{P}_N(z')\,\mathrm{d}z' = 1-(1-z)^N. \tag{21}$$

Using the inverse function of this CDF, the minimum random number is obtained using a single random number $\xi \in [0, 1)$ as follows:

$$C_N^{-1}(\xi) = 1-(1-\xi)^{\frac{1}{N}}. \tag{22}$$

3.1.2 On-the-fly Assignment in Tree Traversal. In our BVH, one child node has the same minimum random number as its parent. Thus this child node is randomly selected with the probability $\frac{|L_0|}{|L_0|+|L_1|}$, where $|L_0|$ is the number of leaves covered by the selected child, and $|L_1|$ is the number of leaves covered by the other child. When one child is selected, the other child node’s minimum random number must be equal to or greater than the parent’s value. Let $\xi_{\min}$ be the minimum random number at the parent node; then the other child node’s value $\xi'_{\min}$ is the minimum within $[\xi_{\min}, 1)$ as follows:

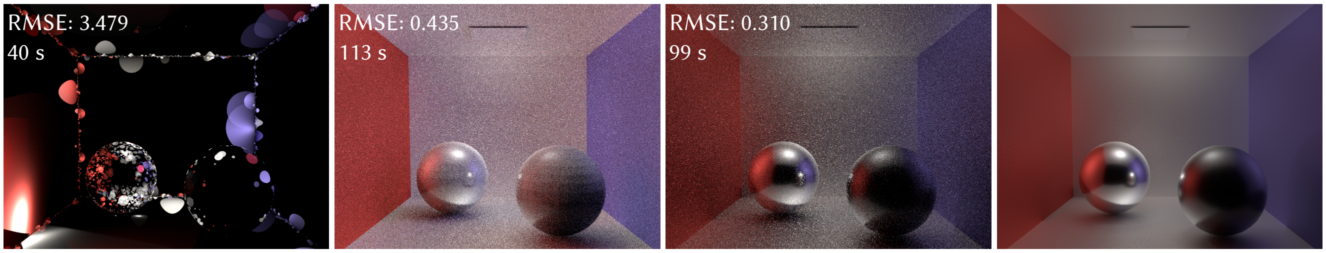
$$\begin{aligned} \xi'_{\min} = \min_{i \in L_1} \xi_i &= \xi_{\min} + (1-\xi_{\min})C_{|L_1|}^{-1}(\xi) \\ &= 1-(1-\xi_{\min})(1-\xi)^{\frac{1}{|L_1|}}. \end{aligned} \tag{23}$$

For the root node, $\xi_{\min} = 0$ is used.

3.1.3 Issues. Although the minimum random number for a node is generated using Eq. (23), a large precision error of floating point calculation is produced for tens of millions of L_1 (Fig. 1b). This error is propagated and accumulated into descendant nodes. In addition, stratification of random numbers is difficult for this algorithm. Therefore, we introduce on-the-fly semi-stratified random number assignment which uses a simpler and more precise calculation.

3.2 Minimum of Semi-Stratified Random Numbers

3.2.1 Derivation. For stratified sampling, the minimum of random numbers is within the lowest stratum. Therefore, the probability that a stratified random number $z \in [0, 1)$ is the minimum within



(a) Precomputed [7] **(b) Non-stratified (Eq. (23))** **(c) Semi-stratified (Eq. (27))** **(d) Reference**

Figure 1: Results of backward stochastic light culling (without MIS weights) using different random number assignment algorithms ($N = 45039313$ light vertices). A precomputed single random number for each light vertex (a) produces undesirable artifacts due to the correlation of variance. On-the-fly assignment of non-stratified random numbers (b) is unbiased in theory, however it produces noticeable bias in practice due to the precision error of floating point calculation. On-the-fly assignment of semi-stratified random numbers (c) avoids this precision error in addition to slightly improving the stratification.

N stratified random numbers is given by the following equation:

$$P_N(z) = \begin{cases} 1 & \text{if } z < \frac{1}{N} \\ 0 & \text{otherwise.} \end{cases} \tag{24}$$

Thus, the PDF is yielded as

$$\hat{P}_N(z) = \frac{P_N(z)}{\int_0^1 P_N(z')\,\mathrm{d}z'} = \begin{cases} N & \text{if } z < \frac{1}{N} \\ 0 & \text{otherwise.} \end{cases} \tag{25}$$

Hence, the inverse of the CDF is trivially derived as follows:

$$C_N^{-1}(\xi) = \frac{\xi}{N}. \tag{26}$$

Compared to Eq. (22), this calculation is simpler and does not increase the precision error for large N.

3.2.2 On-the-fly Assignment in Tree Traversal. Similar to §3.1.2, the parent's minimum random number is transferred into one randomly selected child node. When one child is selected, the other child node's minimum random number is generated. Unlike §3.1.2, since stratified sampling is assumed for each node, this child node's value $\xi'_{\min}$ must be equal to or greater than the upper bound of the parent's stratum. Therefore, let s be the lower bound of $\xi'_{\min}$ (which is the upper bound of the parent's stratum); then $\xi'_{\min}$ is obtained as

$$\begin{aligned} \xi'_{\min} = \min_{i \in L_1} \xi_i &= s + (1-s)C^{-1}_{|L_1|}(\xi) \\ &= \boxed{s - (1-s)\frac{\xi}{|L_1|}}. \end{aligned} \tag{27}$$

The lower bound s' for the next minimum random number is calculated by

$$\boxed{s' = s - (1-s)\frac{1}{|L_1|}}. \tag{28}$$

For the root node, $s = 0$ is used. Compared to Eq. (23), the above calculations do not use the parent's minimum random number as the lower bound, and the lower bound is computed precisely due to the simple form. Hence, the precision error is negligible for our on-the-fly random number assignment algorithm (Fig. 1c). Although stratified sampling is assumed for each node, generated random numbers for leaves are not stratified completely. However, they are still uniformly distributed and partially stratified. Thus, stratification is slightly improved compared to non-stratified random numbers.

ACKNOWLEDGMENTS

The polygon model in the extended abstract is courtesy of Toshiya Hachisuka.

REFERENCES

[1] P. Beckmann and A. Spizzichino. 1963. *Scattering of Electromagnetic Waves from Rough Surfaces.* MacMillan.
[2] R. L. Cook and K. E. Torrance. 1982. A Reflectance Model for Computer Graphics. *ACM Trans. Graph.* 1, 1 (1982), 7–24.
[3] C. Dachsbacher and M. Stamminger. 2006. Splatting Indirect Illumination. In *I3D'06.* 93–100.
[4] I. Georgiev. 2012. *Implementing Vertex Connection and Merging.* Technical Report. Saarland U.
[5] B. T. Phong. 1975. Illumination for Computer Generated Pictures. *Commun. ACM* 18, 6 (1975), 311–317.
[6] B. G. Smith. 1967. Geometrical Shadowing of a Random Rough Surface. *IEEE Trans. Antennas and Propagation* 15, 5 (1967), 668–671.
[7] Y. Tokuyoshi and T. Harada. 2017. Stochastic Light Culling for VPLs on GGX Microsurfaces. *Comput. Graph. Forum* 36, 4 (2017), 55–63.
[8] T. S. Trowbridge and K. P. Reitz. 1975. Average Irregularity Representation of a Rough Surface for Ray Reflection. *J. Opt. Soc. Am* 65, 5 (1975), 531–536.
[9] D. v. Antwerpen. 2011. *Recursive MIS Computation for Streaming BDPT on the GPU.* Technical Report. Delft U. Technol.
[10] E. Veach and L. J. Guibas. 1995. Optimally Combining Sampling Techniques for Monte Carlo Rendering. In *SIGGRAPH '95.* 419–428.
[11] B. Walter, S. R. Marschner, H. Li, and K. E. Torrance. 2007. Microfacet Models for Refraction Through Rough Surfaces. In *EGSR'07.* 195–206.