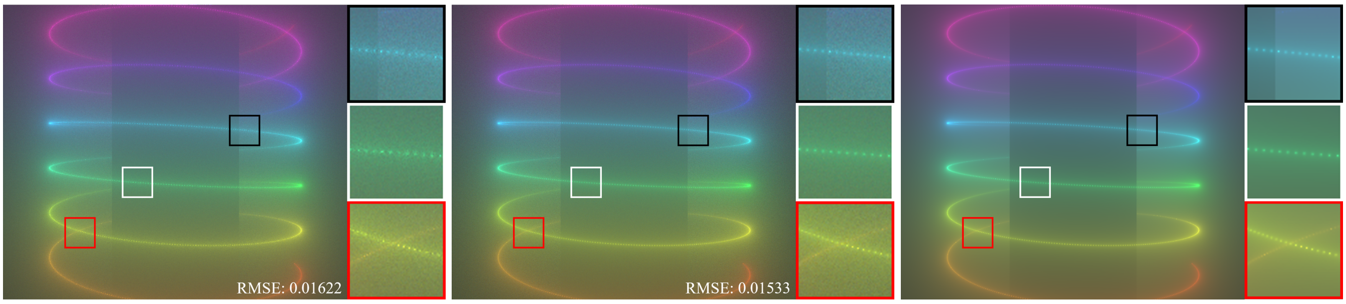
Stochastic Light Culling for Single Scattering in Participating Media

Shin Fujieda Yusuke Tokuyoshi Takahiro Harada

Advanced Micro Devices, Inc.



(a) *Reservoir sampling [Cha82]* **(b)** *Our method + reservoir sampling* **(c)** *Reference*

Figure 1: *Equal-time comparisons (1 sec) for single scattering of 2,560 point light sources in a homogeneous medium. (a) Reservoir sampling to choose one light for each iteration with equiangular sampling [KF12] (416 samples/pixel). (b) Our method produces higher-quality results than using only reservoir sampling for selecting a light (380 samples/pixel).*

Abstract

We introduce a simple but efficient method to compute single scattering from point and arbitrarily shaped area light sources in participating media. Our method extends the stochastic light culling method to volume rendering by considering the intersection of a ray and spherical bounds of light influence ranges. For primary rays, this allows simple computation of the lighting in participating media without hierarchical data structures such as a light tree. First, we show how to combine equiangular sampling with the proposed light culling method in a simple case of point lights. We then apply it to arbitrarily shaped area lights by considering virtual point lights on the surface of area lights. Using our method, we are able to improve the rendering quality for scenes with many lights without tree construction and traversal.

CCS Concepts
• ***Computing methodologies*** $\rightarrow$ ***Rendering;***

1. Introduction

Tree-based light sampling techniques are typically used in production rendering for scenes containing many light sources. In order to choose lights used for shading from a large set, it requires building and traversing a hierarchical structure such as a light tree [CK18]. Although it is useful for shading in the object space, it is sometimes unnecessary when we need to consider scattering in participating media only for primary rays to add volumetric effects such as light shafts.

In this paper, we propose a simple method to compute single scattering only visible from a camera without processing a hierarchical data structure. Our method extends the *stochastic light culling* method [TH16] to participating media lit by point or arbitrarily shaped area light sources. Using stochastic light culling, we restrict the range of influence of each light in an unbiased way. This allows us to cull lights with less importance for shading as pre-processing. We also derive an efficient sampling approach for a scene with uniform volume lit by point and area lights.

The contributions of our work are as follows:

- We extend stochastic light culling to participating media by considering the intersection of a ray and two types of bounding spheres of light influence ranges.
- We introduce a combination of equiangular sampling [KF12] and our stochastic light culling.

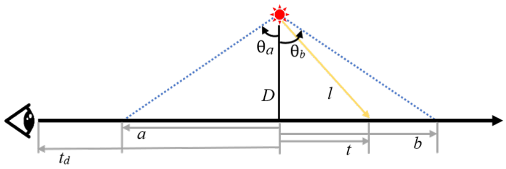


Figure 2: *Settings of equiangular sampling.*

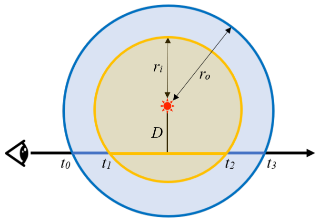


Figure 3: *Spherical bounds for the point light case.*

We conclude this paper by demonstrating the effectiveness of our method for primary rays in a homogeneous medium.

2. Related Work

Distance Sampling in Participating Media. One of the major approaches for distance sampling distributes samples proportional to transmittance [PJH16]. To focus on more samples close to the light, equiangular sampling [KF12] designs a probability density function (pdf) proportional to the fall-off function: $f(l) = 1/l^2$ where l is the distance from a light source. For this equiangular sampling, we sample a distance $t(\xi)$ according to the following pdf $p(t)$:

$$p(t) = \frac{D}{(\theta_b - \theta_a)(D^2 + t^2)}, \tag{1}$$

$$t(\xi) = D \tan\left((1-\xi)\theta_a + \xi\theta_b\right), \tag{2}$$

where $\xi \in [0,1)$ is a uniform random number. Fig. 2 describes the involved parameters.

Light Culling. We only need to evaluate the shading point in a finite range if we use light culling which restricts the influence range of light. Dachsbacher and Stamminger [DS06] rendered indirect illumination by splatting bounding geometries around virtual point lights (VPLs) [Kel97]. To create the bounding geometry, they derived a bounding ellipsoid of the isosurface for indirect diffuse and glossy lights. Tile-based methods [Ste15] are a widely-used acceleration technique in real-time rendering. In this method, lights are binned into 2D screen-space tiles considering the limited light range. These light-culling approaches have not been used for accurate physically-based rendering because such approaches assume limited influence ranges of lights which can cause noticeable darkening bias. Tokuyoshi and Harada [TH16; TH17] avoided this problem by introducing a stochastic fall-off function.

Stochastic Light Culling. Stochastic light culling randomly determines the influence range of each light using *Russian roulette* [AK90]. For physically based lights, the incoming radiance at a shading point is described as $L(l) = I(\omega)/l^2$ where $I(\omega)$ is the radiant intensity and ω is the direction from the light source to the shading point. A Russian roulette technique randomly samples $L(l)$ with the following probability $P(l) \in [0,1]$ which is proportional to $L(l)$ for efficiency as follows:

$$P(l) = \min\left(\frac{L(l)}{\alpha}, 1\right), \tag{3}$$

where α is a user-defined constant value to control variance. Then, if the light is accepted, the fall-off function can be approximated with $P(l)$ as follows:

$$L'(l) \approx \begin{cases} \frac{L(l)}{P(l)} = \max(\alpha, L(l)), & \text{if } P(l) > \xi \\ 0, & \text{otherwise} \end{cases}. \tag{4}$$

Drawing a single random number ξ for each light source instead of each shading point, this method allows us to bound the influence range of each light and also to cull the unimportant lights outside the bound in an unbiased way.

3. Stochastic Light Culling in Participating Media

Applying a stochastic light culling method described in Sec. 2 for single scattering in participating media, we can simply calculate the lighting without tree construction and traversal. We first consider the simple case of point lights, and then we will extend our method to area lights. Finally, we will discuss how to optimize the implementation of our method on the GPU for primary rays.

3.1. Point and Sphere Lights

The stochastic fall-off function in Eq. 4 is discontinuous. So we can partition the range of this function into three regions which are separated by two spheres, as shown in Fig. 3. We introduce the radius of the inner sphere here while that of the outer sphere is derived in Tokuyoshi and Harada [TH16] as follows:

$$r_i = \sqrt{\frac{\max_\omega(I(\omega))}{\alpha}}, \quad r_o = \sqrt{\frac{\max_\omega(I(\omega))}{\alpha\xi}}. \tag{5}$$

$L'(l)$ is equal to α or zero if $r_i < l < r_o$, while $L'(l)$ is always zero if $l \geq r_o$. These two spheres allow us to cull the light before shading. In this case, we consider the following three situations for how the ray intersects these spheres.

(a) Not intersect to the outer sphere.
(b) Intersect to the outer sphere but not to the inner sphere.
(c) Intersect to both spheres.

In the case (a), we just reject its light for shading. In the case (b), we just need to sample a distance t from a constant pdf because the fall-off function is constant. Thus, we sample the shading location t uniformly between t_0 and t_1 with a pdf, $1/(t_1 - t_0)$. And the case (c) is the most complicated case to compute because we need to integrate a pdf over each region to decide the sample position. So we will focus on this case in the following.

Fig. 3 describes the situation of the case (c) where t_0, t_1, t_2 and t_3 denote the distances for the intersection locations of a ray to the two spherical bounds. In order to sample efficiently in this case, we use a different sampling technique for each region. Integrating pdf over segments $[t_0, t_1]$, $[t_1, t_2]$ and $[t_2, t_3]$, we can use it as the importance of each segment. In the segments $[t_0, t_1]$ and $[t_2, t_3]$, pdf is uniform because the fall-off function is canceled out with the Russian roulette probability (Eq. 4). And also in the segment $[t_1, t_2]$, pdf is proportional to the fall-off function. Therefore, we use $1/r_i^2$ and $1/(D^2+t^2)$ for the segments $[t_0, t_1]$ and $[t_2, t_3]$, and the segment $[t_1, t_2]$ for each. Then, integrating them over each region, the importance can be obtained as follows:

$$w_0 = \int_{t_0}^{t_1} \frac{1}{r_i^2} \mathrm{d}t = \frac{1}{r_i^2}(t_1 - t_0), \tag{6}$$

$$w_1 = \int_{t_1}^{t_2} \frac{1}{D^2+t^2} \mathrm{d}t = \frac{1}{D}(\theta_{t_2} - \theta_{t_1}), \tag{7}$$

$$w_2 = \int_{t_2}^{t_3} \frac{1}{r_i^2} \mathrm{d}t = \frac{1}{r_i^2}(t_3 - t_2), \tag{8}$$

where θ_{t_1} and θ_{t_2} correspond to θ_a and θ_b for each in Fig. 2. With these importance, we define how to sample the shading position according to where the uniform random number, ξ, falls into. Considering the sum of importance as $w = w_0 + w_1 + w_2$, if $\xi < w_0/w$, it is sampled uniformly between t_0 and t_1. If $w_0/w < \xi \wedge \xi < (w_0+w_1)/w$, we do equiangular sampling on the segment between t_1 and t_2. And if $\xi > (w_0+w_1)/w$, we also sample it uniformly between t_2 and t_3. For each case, we sample it on the segment with the following pdf:

$$p_0(t) = \frac{w_0}{w} \frac{1}{t_1 - t_0}, \tag{9}$$

$$p_1(t) = \frac{w_1}{w} \frac{D}{(\theta_{t_2} - \theta_{t_1})(D^2+t^2)}, \tag{10}$$

$$p_2(t) = \frac{w_2}{w} \frac{1}{t_3 - t_2}. \tag{11}$$

3.2. Area Lights with Diffuse Emission Profile

Our method is applicable to arbitrary shapes of area lights by generating VPLs on the area light surface. Since the radiant intensity for these VPLs depends on the area light normal $\mathbf{n}$ similar to diffuse VPLs, we can use tighter bounds for diffuse VPLs [DS06; TH17] instead of bounding spheres centered at the VPL position. The isosurface of the radiance emitted from the VPL is $s(\omega) = \sqrt{\frac{\Phi(\mathbf{n}\cdot\omega)}{\pi\alpha\xi}}$ where Φ is the radiant flux of the VPL. We use a bounding sphere for this isosurface (Fig. 4 Left), which is derived in Tokuyoshi and Harada [TH17]. With this bounding sphere, we define two bounds for area lights as shown in Fig. 4 Right, whose radii are

$$r_i^{\text{area}} = \left(\frac{4}{27}\right)^{\frac{1}{4}} \sqrt{\frac{\Phi}{\pi\alpha}}, \quad r_o^{\text{area}} = \left(\frac{4}{27}\right)^{\frac{1}{4}} \sqrt{\frac{\Phi}{\pi\alpha\xi}}. \tag{12}$$

And these centers are given by

$$\mathbf{c}_i = \mathbf{x} + \left(\frac{1}{3}\right)^{\frac{3}{4}} \sqrt{\frac{\Phi}{\pi\alpha}}\mathbf{n}, \quad \mathbf{c}_o = \mathbf{x} + \left(\frac{1}{3}\right)^{\frac{3}{4}} \sqrt{\frac{\Phi}{\pi\alpha\xi}}\mathbf{n}, \tag{13}$$

where $\mathbf{x}$ is the VPL position. Then, for sampling of the shading position, we can handle area lights in a similar fashion as point lights in Sec. 3.1. We thereby use the same weights as in Eqs. 6–8,

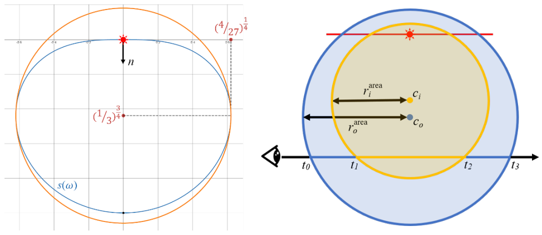


Figure 4: *Left: Diffuse emission profile. Blue shows the isosurface of the emitted radiance and orange shows the bounding sphere of this diffuse VPL. Right: Two spherical bounds for area light.*

which ignores the directionality of the radiant intensity for brevity. For the detail, please see the supplemental document.

3.3. Tile-based Light Culling for Primary Rays

Thanks to stochastic light culling, our proposed method has the limited influence range for each light in an unbiased way. We can reduce the list of lights which we need to evaluate for each primary ray by culling the lights on the screen space. As per-ray culling is computationally expensive, we can use tile-based light culling. Tile-based light culling is executed on the screen space by a thread block for a tile. First, a frustum is constructed for a tile and then, each light in the scene is tested against the frustum to check the overlaps between lights and the frustum on the tile. This overlapping test is executed with the number of all threads in a thread block in parallel. And overlapping lights are stored on the list in shared memory, where all threads in a thread block have access, using an atomic operation. Then, we evaluate only the lights in the shared list per tile and accumulate the contributions.

4. Results and Discussion

We benchmark our method using tile-based light culling with a tile size of 16×16. All experiments in this paper are executed on an AMD Radeon™ RX 6900 XT GPU at 1280×720 screen resolution. In order to execute a fair comparison, we choose one light used for shading for each primary ray with reservoir sampling [Cha82], and then sample a distance using equiangular sampling. Since we assume there is no acceleration structure for lights, we, at first, select 32 light sample candidates uniformly. And then, one light is selected from the candidates using $(\theta_b - \theta_a)/D$ (Eq. 7) as the weight for reservoir sampling (Fig. 1a and Fig. 6 Left). On the other hand, our method selects one light from the shared list per tile with the weight for reservoir sampling, $w = w_0 + w_1 + w_2$, for each light sample (Fig. 1b and Fig. 6 Center). The image quality is evaluated with the root mean squared error (RMSE) metric.

Point Lights. Fig. 1 shows an equal-time comparison on a scene where 2,560 point light sources are placed in a homogeneous medium. Our method achieves a lower RMSE compared to reservoir sampling with the same rendering time of one second. Although our method produces higher-quality results, the overhead is small compared to the variance reduction (416 vs 380 samples per pixel, respectively) because the light culling is computationally inexpensive.

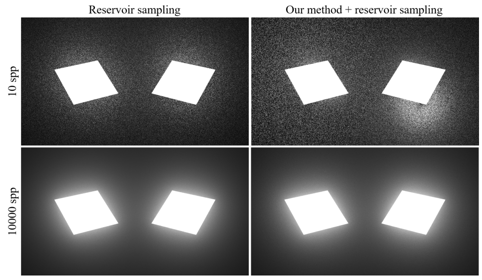


Figure 5: *Equal-sample comparisons on a four triangle light scene with reservoir sampling and our method.*

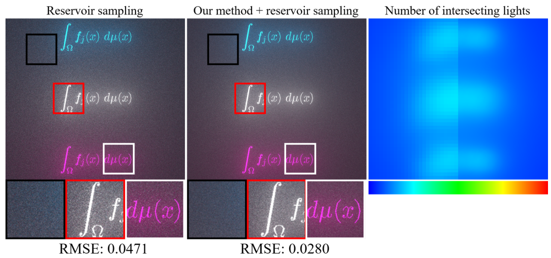


Figure 6: *Equal-sample comparisons on a 2,997 triangle light scene with reservoir sampling and our method. Right shows the number of lights intersecting to the frustum and the outer sphere.*

Area Lights. Fig. 5 shows an equal-sample comparison for four triangle lights in a homogeneous medium. In this experiment, we generate one VPL per triangle. We choose one light out of four lights with reservoir sampling while our method similarly picks up one light from the shared light list. With a few samples per pixel, our method represents more variances than reservoir sampling. This is because stochastic light culling is an acceleration technique for many lights, and it does not improve the efficiency for a few lights. On the other hand, with enough sample count, our method achieves the equivalent result because it is an unbiased and consistent estimator. The efficiency of our method can be improved by generating many VPLs on area lights. The results for the more complex scene with much more lights are shown in Fig. 6. There are 2,997 triangle lights in this scene, and we generate one VPL per triangle also for this scene. This time, our method obtains a lower RMSE than reservoir sampling with 256 samples and both methods show a similar performance of 0.56 seconds. Moreover, Fig. 6 Right represents the visualization of the number of lights collected in each tile (left) and intersecting the outer spherical bound (right). The heat color from blue to red represents 0 to 2,997 lights. So as more distant from lights, fewer lights are considered for the contribution. Note that for sampling area lights (Sec. 3.2), our way of ignoring the directionality of the radiant intensity can produce additional variance. Variance reduction for this case is to be addressed in future work.

5. Conclusion and Future Work

We presented an efficient approach to compute single scattering in media using stochastic light culling without a hierarchical data structure. Our proposed method can be implemented with further optimization on the GPU with tile-based culling thanks to the limited light range. We performed experiments that show our method can obtain higher-quality images compared to reservoir sampling for the light selection with equiangular sampling.

In this paper, we demonstrated our method in homogeneous media with an isotropic phase function. However, we can combine our method with transmittance-based importance sampling via multiple importance sampling to reduce the variance for heterogeneous media. Stochastic light culling does not take into account the phase function on ray segments similar to equiangular sampling. Thus, it could produce a variance for highly anisotropic media. We also showed the application of the method in direct illumination for volume rendering for primary ray only, which is a relatively simple environment. However, we believe that our work opens up many interesting future works. For example, we can render indirect illumination by applying our proposed method to VPLs in participating media where the radiant intensity of VPLs depends on the phase function. Then, we can use a bounding ellipsoid for the GGX distribution [TH17] instead of a bounding sphere if we assume this phase function is the SGGX distribution [HDCD15]. Another one would be an extension to distance sampling taking into account the transmittance on the ray segment to reduce the noise further, which is possible because of the uniform importance which we derived. Our method is also applicable to virtual ray lights [NNDJ12] by exchanging the ray segment and a point light. We would like to investigate these extensions in the future.

References

[AK90] Arvo, J. and Kirk, D. "Particle Transport and Image Synthesis". SIGGRAPH Comput. Graph. 24.4 (1990), 63–66 2.

[Cha82] Chao, M. "A General Purpose Unequal Probability Sampling Plan". Biometrika 69.3 (1982), 653–656 1, 3.

[CK18] Conty Estevez, A. and Kulla, C. "Importance Sampling of Many Lights with Adaptive Tree Splitting". Proc. ACM Comput. Graph. Interact. Tech. 1.2 (2018) 1.

[DS06] Dachsbacher, C. and Stamminger, M. "Splatting Indirect Illumination". I3D '06. 2006, 93–100 2, 3.

[HDCD15] Heitz, E., Dupuy, J., Crassin, C., and Dachsbacher, C. "The SGGX Microflake Distribution". ACM Trans. Graph. 34.4 (2015) 4.

[Kel97] Keller, A. "Instant Radiosity". SIGGRAPH '97. 1997, 49–56 2.

[KF12] Kulla, C. and Fajardo, M. "Importance Sampling Techniques for Path Tracing in Participating Media". Comput. Graph. Forum 31.4 (2012), 1519–1528 1, 2.

[NNDJ12] Novák, J., Nowrouzezahrai, D., Dachsbacher, C., and Jarosz, W. "Virtual Ray Lights for Rendering Scenes with Participating Media". ACM Trans. Graph. 31.4 (2012) 4.

[PJH16] Pharr, M., Jakob, W., and Humphreys, G. Physically Based Rendering: From Theory to Implementation. 3rd. Morgan Kaufmann Publishers Inc., 2016 2.

[Ste15] Stewart, J. "Compute-Based Tiled Culling". GPU Pro 6: Advanced Rendering Techniques. A K Peters/CRC Press, 2015, 435–458 2.

[TH16] Tokuyoshi, Y. and Harada, T. "Stochastic Light Culling". J. Comput. Graph. Tech. 5.1 (2016), 35–60 1, 2.

[TH17] Tokuyoshi, Y. and Harada, T. "Stochastic Light Culling for VPLs on GGX Microsurfaces". Comput. Graph. Forum 36 (2017), 55–63 2–4.