



Enhanced Direct Lighting Using Visibility-Aware Light Sampling

Geonu Noh , Hajin Choi , and Bochang Moon ^(✉) 

Gwangju Institute of Science and Technology, Gwangju 61005, South Korea
bmoon@gist.ac.kr

Abstract. Next event estimation has been widely applied to Monte Carlo rendering methods such as path tracing since estimating direct and indirect lighting separately often enables finding light paths from the eye to the lights effectively. Its success heavily relies on light sampling for direct lighting when a scene contains multiple light sources since each light can contribute differently to the reflected radiance on a surface point. We present a light sampling technique that can guide such a light selection to improve direct lighting. We estimate a spatially-varying function that approximates the contribution of each light on surface points within a discretized local area (i.e., a voxel in an adaptive octree) while considering the visibility between lights and surface points. We then construct a probability distribution function for sampling lights per voxel, which is proportional to our estimated function. We demonstrate that our light sampling technique can significantly improve rendering quality thanks to improved direct lighting with our light sampling.

Keywords: light sampling · direct lighting · next event estimation

1 Introduction

Global illumination methods such as path tracing [16] and bidirectional path tracing [31] have been widely adopted in production rendering scenarios [5, 9, 10], where simulating accurate light propagation between lights and 3D virtual models should be conducted. While such Monte Carlo (MC) integration-based algorithms can reduce errors in their rendering images by increasing the number of samples, it often requires non-trivial rendering times to generate a visually acceptable rendering output without noticeable noise.

A commonly adopted technique for reducing the variance in the rendered images is to split the light transport equation into direct and indirect illumination parts and then attempt to connect the eye subpaths, determined by ray tracing from the eye, to lights by casting shadow rays from a surface point (i.e., direct lighting). This simple but effective strategy is commonly referred to as next event estimation (NEE) and has been widely employed in rendering frameworks [15, 25] since it often reduces the MC variance drastically.

While this direct lighting is conceptually simple and can be easily implemented, it technically requires determining a sampling probability for selecting a light source when multiple light sources exist in a scene. A straightforward light selection approach is uniform sampling (i.e., allocating the equal probability for each light), but it can be ineffective since the contribution of light sources can vary significantly per light and surface point.

A natural extension to uniform sampling is to assume that all light sources are visible from surface points and to adjust the light selection probability of being proportional to a potential contribution (e.g., light power) of each light source, referred to as spatial sampling [29]. This simple light selection strategy, spatial sampling, can be effective when all light sources are unoccluded from surface points. However, it is often suboptimal when selected light sources are invisible from the points. Unfortunately, exact visibility information between two points (e.g., a surface point and a point on a light) cannot be pre-determined since it requires tracing a shadow ray that is typically computationally expensive. As a result, a computationally efficient process that closely approximates the visibility information is a technical requirement for an ideal light selection process.

As a recent example of guiding light sampling using visibility information, Guo et al. [12] discretized the scene space, i.e., a space determined by the axis-aligned bounding box of a scene, into a 3D regular grid and estimated the visibility between two voxels using uniformly generated but with a small number of visibility samples. Then the stored visibility was used as the light selection probability so that the lights with high probability, which can be likely visible from a surface point, could be more selected than the other lights with low probability. It enables avoiding unnecessary sample allocations to invisible lights, but it can also be sub-optimal when the contributions of multiple visible lights are significantly different. One may extend this method to consider the potential contributions of lights together with their estimated visibility via multiple importance sampling [28,30], but its performance improvement can still be restricted due to the sparse visibility approximation with a uniform grid structure.

This paper also addresses the light sampling problem (i.e., selecting a light for direct lighting at a surface point). However, we approximate more comprehensive information, a potential light contribution (e.g., light power) together with visibility, so that we can guide the light selection more effectively by forming a light sampling probability of being proportional to the contribution of light sources to a surface point. To this end, we estimate a spatially-varying function that approximates the contribution of each light source per local region using an adaptively constructed octree.

We demonstrate that our approach can produce more accurate rendering results (e.g., up to $7\times$ lower errors) than the existing approaches (e.g., spatial sampling and [12]) given equal-time budgets, thanks to improved direct lighting with our light selection for various rendering scenes where multiple lights exist.

2 Related Work

To solve the direct illumination integral, one typically applies a Monte Carlo (MC) integration (i.e., a numerical approximation of the integral), and its

approximation quality (i.e., the difference between the approximate value and the unknown ground truth), mainly depends on the randomly chosen light samples. One of the most popular approaches is to reduce the approximation error (i.e., the variance of the MC integration) using carefully selected light samples by adjusting the probability of generating such samples [29]. Such techniques often constructed their sampling probability by taking terms in the direct light transport equation into account (e.g., the emissive light energy [1, 23], the bidirectional reflectance distribution function (BRDF) [19], and both the light energy and BRDF [4, 6]). Ghosh and Heidrich [11] presented a two-stage sampling method where they applied a Monte Carlo sampling using the BRDF and lights at the first stage and then used a mutation strategy to allocate more samples to partially-occluded regions in the later stage.

Importance sampling for direct lighting has also received attention for rendering scenarios where many lights exist. A well-known approach to making the sampling scalable against many lights (e.g., hundreds of thousands of lights) is to use light clusters maintained in an acceleration structure (e.g., a light tree) and select a light according to their importance (i.e., the radiance contribution at a surface point) [7, 17, 20, 35, 36]. Additionally, progressively refining the sample distribution was recently explored [18, 24, 33] so that the sample distribution for direct lighting can have a similar shape to the unknown radiance contribution as collecting more samples in rendering.

The sophisticated importance sampling aforementioned can be necessary when rendering a scene with environment lights or thousands of lights. Nonetheless, it can be preferable in practice to choose a simpler alternative that does not require an expensive data structure (e.g., light trees) for typical scenes with moderate numbers of small lights. For example, one can select a light according to the light power, the area, and the distance to a surface point, and then sample a point on the selected light [29]. A variant of this technique was implemented, referred to as spatial sampling, in a well-known rendering framework [25]. While this simple approach can behave well when all lights are visible to the shading point, its efficiency gain over the most straightforward approach (i.e., uniform light selection) can disappear when the visibilities between shading points and lights vary significantly. Guo et al. [12] employed a computationally efficient data structure (i.e., a uniform grid), to contain estimated visibility between two voxels, and showed that this estimated visibility could be exploited for selecting a visible light. However, this approach did not consider the relative importance of unoccluded light sources (e.g., light power). Like these simple methods, we focus on rendering scenarios where the number of lights is moderate and propose an improved light sampling using a more comprehensive estimation of light contributions while maintaining a low computational overhead so that our technique can be a practical choice for rendering scenarios where multiple lights exist.

Path Guiding. Importance sampling for indirect lighting also has been actively explored. A widely adopted strategy is to adjust the sampling density of secondary rays to be proportional to the incoming radiance from the rays per surface point [30], often called path guiding. Well-known approaches include

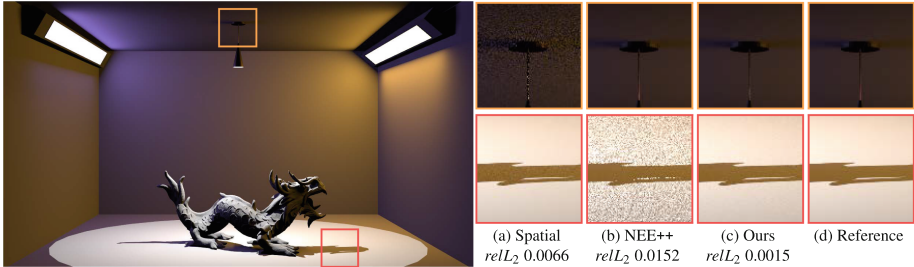


Fig. 1. Equal-time comparisons (in twenty secs) of light sampling techniques for a scene with three lights. We render the scene without indirect illumination to show the visual differences among the tested methods clearly. We measured the numerical accuracy of rendered images using a relative L_2 error ($relL_2$) [27].

approximating the contribution of indirect light radiance with the Gaussian mixture model [13, 34] and a spatial-directional tree [8, 21]. Rath et al. [26] recently showed that considering the variance of the pixel estimator can make path guiding effective. A popular alternative to such approaches is to devise a deep neural network that can guide the sampling of indirect light paths [2, 14, 22, 37–39]. The path guiding techniques are orthogonal to importance sampling for direct lighting, including ours, and can be used together for more effective rendering.

3 Problem Statement and Motivation

This paper aims to reduce the variance of direct lighting by carefully choosing lights to be sampled according to their estimated contribution. This section provides a background on direct lighting and present a light selection problem for direct lighting, followed by a motivation for our light selection approach. The computation of the direct illumination can be formulated into an area form [29]:

$$L_d(x, \omega) = \int_A L_e(x, x') \rho(x, x', \omega) V(x, x') G(x, x') dx', \quad (1)$$

which produces the outgoing radiance $L_d(x, \omega)$ on a surface point x with direction ω by integrating the emitted radiance $L_e(x, x')$ from the points x' on the surface A of light sources. $\rho(x, x', \omega)$ is the bidirectional reflectance distribution function (BRDF), $V(x, x')$ is the visibility (one if x and x' are mutually visible and zero otherwise), and the geometric term $G(x, x') = \frac{\cos \theta_x \cos \theta_{x'}}{\|x - x'\|^2}$. A Monte Carlo (MC) estimator for the direct lighting (Eq. 1) can be written by

$$\langle L_d(x, \omega) \rangle = \frac{L_e(x, x') \rho(x, x', \omega) V(x, x') G(x, x')}{p(x'|x)}, \quad (2)$$

which provides an unbiased estimate of the ground truth radiance $L_d(x, \omega)$ by randomly selecting a light sample x' given the surface point x . This sampling

process is controlled by the selection probability, probability density function (PDF) $p(x'|x)$. Note that one can simply take an averaged value of the estimates when selecting multiple light samples.

The PDF $p(x'|x)$ can be decomposed into two terms, one for selecting a light and the other for sampling a point on the chosen light, i.e., $p(x'|x) = p(E_l|x)p(x'|E_l)$ where E_l is a l -th light ($l \in [1, L]$) given L lights in the scene.

A common choice of choosing E_l (i.e., $p(E_l|x)$) is to consider the contribution of the light into the outgoing radiance $L_d(x, \omega)$. For example, a well-known renderer, PBRT [25], exploits the spatial sampling strategy that varies the PDF of selecting a light per a discretized scene region. Specifically, it divides a scene space into M voxels, each of which contains a PDF (e.g., $p(E_l|x)$ for surface points x in the m -th voxel V_m). The spatial sampling constructs the probability $p(E_l|x)$ for selecting the l -th light E_l to be proportional to the following function:

$$f(x, E_l|x \in V_m) = \frac{1}{N_m} \sum_{i=1}^{N_m} \frac{L_e(x_i, x'_i)G(x_i, x'_i)}{p(x'_i|E_l)}, \quad (3)$$

which is computed using N_m number of randomly generated points x_i inside the voxel and light samples x'_i on the l -th light. The function $f(\cdot)$ varies per voxel and light, and thus the function should be evaluated for each voxel and light using the randomly selected points x_i and x'_i . Then, the PDF $p(E_l|x)$ at a surface point x in the m -th voxel is determined to be proportional to the $f(\cdot)$, i.e., $p(E_l|x) \propto f(\cdot)$.

The PDF $p(x'|E_l)$ of selecting a specific point x' on the l -th light depends on the properties of the light (e.g., the shape) and was well-established in an early method [29]. We employ the existing technique for the PDF $p(x'|E_l)$ and refer to PBRT [25] for more details on this sampling.

Our Motivation for the Light Selection $p(E_l|x)$. The estimated direct illumination term ($\langle L_d(x, \omega) \rangle$) in Eq. 2) is an unbiased estimate of the ground truth $L_d(x, \omega)$. Still, it typically suffers from non-trivial variance (thus a noisy estimate) unless many samples are used. As a result, a careful light selection strategy, whose PDF varies per surface point x , is required to minimize the variance of the estimate.

Spatial sampling offers a simple means that builds a spatially-varying PDF using a discretized data structure (i.e., voxels), but it does not consider the visibility $V(x, x')$ between two points x and x' (see Eq. 3). Similarly, Guo et al. [12] proposed NEE++ that is an improved direct light sampling considering voxel-to-voxel visibility. Then, given a surface point, it tries to choose only visible lights that can contribute to the radiance but does not consider relative importance among visible lights introduced by the other terms such as $L_e(x_i, x'_i)$ in Eq. 2.

To make the direct lighting more effective, we extend the spatial sampling into a more comprehensive one that includes the visibility term additionally while considering the original terms ($L_e(x_i, x'_i)$ and $G(x_i, x'_i)$). Figure 1 shows an example result where the existing techniques (spatial sampling [25] and NEE++ [12]) do not effectively reduce the noise from the direct illumination due to their incomplete consideration. On the other hand, our technique shows a rendering

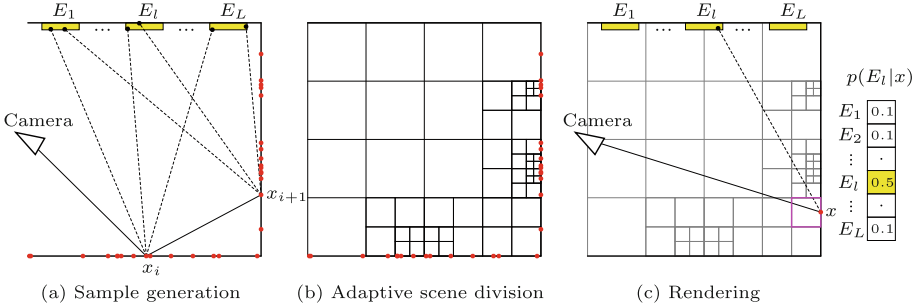


Fig. 2. As a pre-processing step, we determine the surface points through path tracing with one sample per pixel (a) and split the space adaptively according to the density of vertices to estimate the contribution of each local region from each light (b). Then, we perform the original path tracing with direct lighting whose light selection probability is proportional to our target function stored in the octree (c).

result with much-reduced noise thanks to a more effective light sampling, which will be presented in the following section.

4 Visibility-Aware Light Sampling with an Adaptive Octree

This section presents a simple but effective light sampling by estimating spatially-varying light contributions. Analogously in the existing sampling strategy (e.g., spatial sampling), we discretize the scene space into multiple voxels, each of which contains a localized target function:

$$g(x, E_l | x \in V_m) = \sum_{i=1}^{N_m} \frac{L_e(x_i, x'_i) G(x_i, x'_i) V(x_i, x'_i)}{N_m p(x'_i | E_l)}, \quad (4)$$

which approximates the l -th light contribution to the points within a local space (i.e., the m -th voxel V_m). The major modification to the existing one (Eq. 3) is that we incorporate the visibility term $V(x_i, x'_i)$ so that visible lights can have more chance to be selected for direct lighting.

Our next task is to compute the localized function per voxel for each light using N_m samples so that the light selection PDF $p(E_l | x)$ (i.e., a probability of selecting l -th light at the surface point x within m -th voxel) can be constructed to be proportional to the target function (i.e., $p(E_l | x) \propto g(\cdot)$). Figure 2 illustrates our framework where we estimate our localized target function as a pre-processing (Sect. 4.1) and perform the rendering using direct lighting with our light sampling.

4.1 Estimation of Our Localized Target Function

We estimate the localized target function $g(x, E_l | x \in V_m)$ (Eq. 4) to locally vary the light selection probability $p(E_l | x)$ for direct lighting in the original

rendering. Analogously in spatial sampling (in Sect. 3), we discretize the scene space into disjoint voxels where each voxel contains L light contributions, i.e., $g(x, E_1|x \in V_m), \dots, g(x, E_L|x \in V_m)$. Note that all the points x within a voxel V_m share the same target function, $g(x, E_l|x \in V_m)$, and thus we can reduce its approximation errors by creating more voxels. Unfortunately, estimation of the target function requires non-trivial computational efforts since each voxel should evaluate N_m samples (see Eq. 4). As a result, we should employ a simple but compact data structure that approximates the light contributions appropriately without forming a large number of voxels.

To this end, we present an octree-based estimation where each leaf node (a voxel) estimates L light contributions. Specifically, we perform path tracing to collect samples (L -dimensional vectors) for constructing the octree. We generate one sample per pixel to make the computational overhead of this process small and set the maximum ray depth of the path tracing to three to approximate light contributions for scene areas where direct lighting can be potentially performed, i.e., not just the region intersected by the camera rays. Whenever we find an intersection point x_i , we cast a shadow ray towards a light sample x'_i for each light and evaluate

$$g(x_i, E_l) = \frac{L_e(x_i, x'_i)G(x_i, x'_i)V(x_i, x'_i)}{p(x'_i|E_l)}. \quad (5)$$

Note that the $g(x_i, E_l)$ is computed for each light (i.e., for E_1, \dots, E_L) per a surface point x_i and thus we have an L -dimensional value at a surface point x_i . We treat the $\mathbf{g}(x_i) = \{g(x_i, E_1), \dots, g(x_i, E_L)\}$ as a sample in \mathbb{R}^L for our octree construction.

Once the samples are generated, we construct an adaptive octree whose leaf node contains roughly the same number of samples $\mathbf{g}(x_i)$. Note that the density of surface points x_i is typically non-uniform, and thus the voxel size of each leaf node needs to be adaptively determined by the density of the points. To control this adaptive process, we take an input parameter that controls the number of leaf nodes, which will be detailed in the subsequent paragraph.

After collecting N_m samples for each voxel (i.e., a leaf node in the tree), we compute our localized target function $g(x, E_l|x \in V_m)$ (Eq. 4) using the N_m samples (i.e., $\mathbf{g}(x_i)$ whose $x_i \in V_m$). Note that once the tree is generated (i.e., evaluating the target function per each leaf node), we do not need to store the individual samples. Once this tree-building process is completed, we perform the original rendering, i.e., a standard path tracing with direct lighting. For direct lighting in the rendering process, we select a light according to the light selection probability $p(E_l|x)$ that is proportional to our target function.

Implementation Details. We initially divide the bounding box of the scene space into $4 \times 4 \times 4$ equal-sized voxels (i.e., a uniform octree with depth two). Then, we recursively divide the voxels by limiting the number of samples inside the voxels. Specifically, we split a voxel only if the number of samples within the voxel is larger than $\frac{4N}{T}$ where N and T are the total sample count and a user-specified parameter that controls the number of leaf nodes, respectively. We observed that

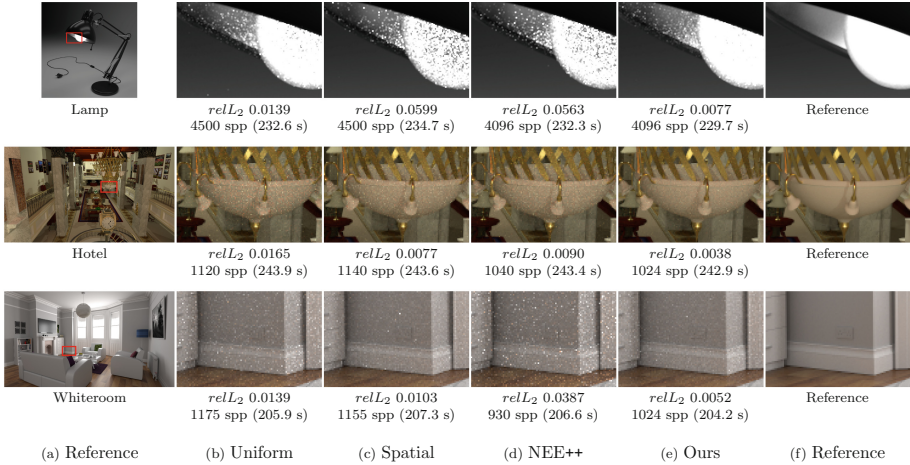


Fig. 3. Equal-time comparisons of light sampling techniques. We vary the samples per pixel (spp) per each method for the same-time test. The Lamp scene has two lights (in the bulb and on the ceiling), the Hotel has nine area lights, and the Whiteroom has eight area lights. The times in the parenthesis are the total rendering times, including the overheads of tested methods.

the number of leaf nodes increases roughly proportional to the parameter T with this setting. We set the T to 4096 unless otherwise specified. We integrated our light selection into a well-known rendering framework, PBRT [25]. The original implementation of PBRT treats each primitive of a mesh light as an individual light source, increasing the number of lights L unnecessarily. We modified the existing implementation to treat a mesh light as a single light source.

5 Results and Discussion

We compare our method with existing light sampling techniques, spatial sampling [25] and NEE++ [12]. As a baseline, we also test the most straightforward choice, uniform sampling, which assigns the same probability for all lights. We use path tracing while varying the light sampling strategy for direct lighting. All tests have been done on a PC with an AMD Ryzen 3900X CPU.

Comparisons of Light Sampling Methods. Figure 3 shows equal-time comparisons of our method and existing light sampling techniques. Spatial sampling and NEE++ produce higher errors than the uniform sampling method for the Lamp and Whiteroom scene. It indicates that considering a partial term (e.g., light sampling without visibility consideration or only using visibility estimation) is not robust. Ours, however, consistently generates lower errors than the baseline and the existing methods, thanks to our more accurate estimation of light contribution. In addition, NEE++ fails to capture the spatially varying visibility due to their use of sparse uniform structure, worsening the results even than the

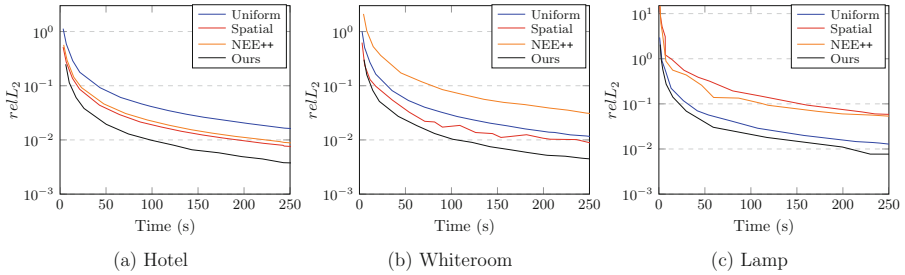


Fig. 4. Numerical convergences of path tracing with different light selection strategies.

Table 1. Breakdowns of our total rendering times (in secs) for the results in Fig. 3.

Scene	spp	Sample generation	Octree building	Rendering	Total
Lamp	4096	0.12	0.47	229.1	229.7
Hotel	1024	0.74	1.78	240.4	242.9
Whiteroom	1024	0.63	1.21	202.3	204.2

uniform sampling method even in the relatively simple scene with respect to the visibility (i.e., the Lamp). On the other hand, our selection using an adaptive octree structure makes a much-improved result than the existing methods, e.g., more than $7\times$ lower errors than NEE++. We also test the numerical convergence of the tested methods over time in Fig. 4. As shown in the figure, the benefit of our technique for path tracing is consistently maintained over time.

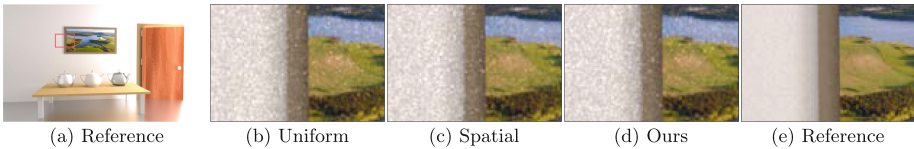
Our Computational Overhead. Table 1 shows a breakdown of the total rendering times reported in Fig. 3. Note that our pre-processing is decomposed into two stages, the sample generation and the octree construction (see Fig. 2). As seen in the table, our computational overhead is minor (e.g., 0.257% to 1.046%) in the total times, given the offline settings.

Analysis of Our Voxel Granularity. Our method controls the voxel granularity of the octree by a user-specified parameter T (discussed in Sect. 4). Table 2 shows our computational and memory overheads with rendering accuracy by varying the T . Using a large T (and thus many voxels) does not necessarily improve the rendering quality since the number of samples, $\mathbf{g}(x_i)$ used to construct the localized target function within a voxel, decreases and thus leads to noisy estimates of the light contributions. Also, with a small T , the rendering quality degrades as our light selection strategy cannot appropriately capture the locally varying light contributions due to a high discretization error. Consequently, we set the T to the chosen number, 4096, for all tested scenes.

Limitations and Future Work. The main limitation of the proposed method is that, like the other light selection methods, it cannot drastically improve the rendering quality for the scenes where most of the scene regions are lit by indirect

Table 2. Analysis of the voxel granularity for our adaptive octree.

Scene	T	Number of leaf nodes	$relL_2$	Memory (KB)	Time (s)
					Overhead
Lamp	512	778	0.0102	35	0.50
	4096	5349	0.0077	239	0.59
	32768	43527	0.0081	1945	0.77
Hotel	512	596	0.0051	43	1.97
	4096	4789	0.0038	345	2.54
	32768	38914	0.0037	2801	2.99
Whiteroom	512	505	0.0073	34	1.47
	4096	4537	0.0052	309	1.90
	32768	39250	0.0089	2672	2.21

**Fig. 5.** Failure cases of the light sampling techniques for the Veach-Ajar scene with 4K spp. The spatial sampling ($relL_2$: 0.104) and ours ($relL_2$: 0.103) become ineffective and do not improve the uniform sampling ($relL_2$: 0.106).

lighting. Figure 5 shows a clear example where our method does not improve the rendering quality compared to the baseline (i.e., uniform sampling). Given the specific scene setting, light sources are located behind the door, and thus our light selection for direct lighting cannot be effective. Note that this is also a counter-example of the use of direct lighting.

In addition, our target function (Eq. 4) does not fully consider the direct lighting integrand (Eq. 2) since we do not consider the BRDF. Note that our method is unbiased, like the other light selection methods, and this partial consideration does not indicate that one cannot use general materials. Nevertheless, it is desirable to extend our method to a more comprehensive one with BRDF for further improving direct lighting. Also, we test our light sampling with unidirectional path tracing, but it would also be interesting to integrate our sampling into other light transport techniques (e.g., bidirectional path tracing [31] and metropolis light transport [32]) and path-reusing techniques (e.g., [3]). We leave such extensions to future work.

Acknowledgements. We thank the anonymous reviewers for feedback on our paper and the following authors and artists for the 3D models or scenes: Asian Dragon (the Stanford Computer Graphics Laboratory), Lamp (UP3D), Whiteroom (Jay-Artist), and Veach-Ajar (Benedikt Bitterli). The Hotel scene was bought from TurboSquid.

This work was supported in part by Ministry of Culture, Sports and Tourism and Korea Creative Content Agency (No. R2021080001) and by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2023-00207939).

References

1. Agarwal, S., Ramamoorthi, R., Belongie, S., Jensen, H.W.: Structured importance sampling of environment maps. *ACM Trans. Graph.* **22**(3), 605–612 (2003)
2. Bako, S., Meyer, M., DeRose, T., Sen, P.: Offline deep importance sampling for Monte Carlo path tracing. *Comput. Graph. Forum* **38**(7), 527–542 (2019)
3. Bitterli, B., Wyman, C., Pharr, M., Shirley, P., Lefohn, A., Jarosz, W.: Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting. *ACM Trans. Graph.* **39**(4), 1–17 (2020)
4. Burke, D., Ghosh, A., Heidrich, W.: Bidirectional importance sampling for direct illumination. In: *Proceedings of the 16th Eurographics Conference on Rendering Techniques*, pp. 147–156 (2005)
5. Burley, B., et al.: The design and evolution of Disney’s Hyperion renderer. *ACM Trans. Graph.* **37**(3), 1–22 (2018)
6. Clarberg, P., Akenine-Möllery, T.: Practical product importance sampling for direct illumination. *Comput. Graph. Forum* **27**(2), 681–690 (2008)
7. Conty Estevez, A., Kulla, C.: Importance sampling of many lights with adaptive tree splitting. *Proc. ACM Comput. Graph. Interact. Tech.* **1**(2), 1–17 (2018)
8. Diolatzis, S., Gruson, A., Jakob, W., Nowrouzezahrai, D., Drettakis, G.: Practical product path guiding using linearly transformed cosines. *Comput. Graph. Forum* **39**(4), 23–33 (2020)
9. Fascione, L., et al.: Manuka: a batch-shading architecture for spectral path tracing in movie production. *ACM Trans. Graph.* **37**(3), 1–18 (2018)
10. Georgiev, I., et al.: Arnold: a brute-force production path tracer. *ACM Trans. Graph.* **37**(3), 1–12 (2018)
11. Ghosh, A., Heidrich, W.: Correlated visibility sampling for direct illumination. *Vis. Comput.* **22**(9), 693–701 (2006)
12. Guo, J.J., Eisemann, M., Eisemann, E.: Next event estimation++: visibility mapping for efficient light transport simulation. *Comput. Graph. Forum* **39**(7), 205–217 (2020)
13. Herholz, S., Elek, O., Vorba, J., Lensch, H., Krivánek, J.: Product importance sampling for light transport path guiding. *Comput. Graph. Forum* **35**(4), 67–77 (2016)
14. Huo, Y., Wang, R., Zheng, R., Xu, H., Bao, H., Yoon, S.E.: Adaptive incident radiance field sampling and reconstruction using deep reinforcement learning. *ACM Trans. Graph.* **39**(1), 1–17 (2020)
15. Jakob, W.: Mitsuba renderer (2010). <http://www.mitsuba-renderer.org>
16. Kajiyaya, J.T.: The rendering equation. *SIGGRAPH Comput. Graph.* **20**(4), 143–150 (1986)
17. Ki, H., Oh, K.: A GPU-based light hierarchy for real-time approximate illumination. *Vis. Comput.* **24**(7), 649–658 (2008)
18. Lai, Y.C., Chou, H.T., Chen, K.W., Fan, S.: Robust and efficient adaptive direct lighting estimation. *Vis. Comput.* **31**(1), 83–91 (2015)
19. Lawrence, J., Rusinkiewicz, S., Ramamoorthi, R.: Efficient BRDF importance sampling using a factored representation. *ACM Trans. Graph.* **23**(3), 496–505 (2004)

20. Liu, Y., Xu, K., Yan, L.Q.: Adaptive BRDF-oriented multiple importance sampling of many lights. *Comput. Graph. Forum* **38**(4), 123–133 (2019)
21. Müller, T., Gross, M., Novák, J.: Practical path guiding for efficient light-transport simulation. *Comput. Graph. Forum* **36**(4), 91–100 (2017)
22. Müller, T., McWilliams, B., Rousselle, F., Gross, M., Novák, J.: Neural importance sampling. *ACM Trans. Graph.* **38**(5), 1–19 (2019)
23. Ostromoukhov, V., Donohue, C., Jodoin, P.M.: Fast hierarchical importance sampling with blue noise properties. *ACM Trans. Graph.* **23**(3), 488–495 (2004)
24. Pantaleoni, J.: Importance sampling of many lights with reinforcement lightcuts learning. arXiv preprint [arXiv:1911.10217](https://arxiv.org/abs/1911.10217) (2019)
25. Pharr, M., Jakob, W., Humphreys, G.: *Physically Based Rendering: From Theory to Implementation*, 3rd edn. Morgan Kaufmann Publishers Inc., Burlington (2016)
26. Rath, A., Grittmann, P., Herholz, S., Vévoda, P., Slusallek, P., Krivánek, J.: Variance-aware path guiding. *ACM Trans. Graph.* **39**(4), 1–12 (2020)
27. Rousselle, F., Knaus, C., Zwicker, M.: Adaptive sampling and reconstruction using greedy error minimization. *ACM Trans. Graph.* **30**(6), 1–12 (2011)
28. Sbert, M., Havran, V.: Adaptive multiple importance sampling for general functions. *Vis. Comput.* **33**(6), 845–855 (2017)
29. Shirley, P., Wang, C., Zimmerman, K.: Monte Carlo techniques for direct lighting calculations. *ACM Trans. Graph.* **15**(1), 1–36 (1996)
30. Veach, E.: *Robust Monte Carlo methods for light transport simulation*. Stanford University (1998)
31. Veach, E., Guibas, L.: Bidirectional estimators for light transport. In: Sakas, G., Müller, S., Shirley, P. (eds.) *Photorealistic Rendering Techniques*, pp. 145–167. Springer, Heidelberg (1995). https://doi.org/10.1007/978-3-642-87825-1_11
32. Veach, E., Guibas, L.J.: Metropolis light transport. In: *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1997*, pp. 65–76. ACM Press/Addison-Wesley Publishing Co. (1997)
33. Vévoda, P., Kondapaneni, I., Krivánek, J.: Bayesian online regression for adaptive direct illumination sampling. *ACM Trans. Graph.* **37**(4), 1–12 (2018)
34. Vorba, J., Karlík, O., Šik, M., Ritschel, T., Krivánek, J.: On-line learning of parametric mixture models for light transport simulation. *ACM Trans. Graph.* **33**(4), 1–11 (2014)
35. Walter, B., Fernandez, S., Arbree, A., Bala, K., Donikian, M., Greenberg, D.P.: Lightcuts: a scalable approach to illumination. *ACM Trans. Graph.* **24**(3), 1098–1107 (2005)
36. Wang, Y.C., Wu, Y.T., Li, T.M., Chuang, Y.Y.: Learning to cluster for rendering with many lights. *ACM Trans. Graph.* **40**(6), 1–10 (2021)
37. Zheng, Q., Zwicker, M.: Learning to importance sample in primary sample space. *Comput. Graph. Forum* **38**(2), 169–179 (2019)
38. Zhu, S., et al.: Hierarchical neural reconstruction for path guiding using hybrid path and photon samples. *ACM Trans. Graph.* **40**(4), 1–16 (2021)
39. Zhu, S., et al.: Photon-driven neural reconstruction for path guiding. *ACM Trans. Graph.* **41**(1), 1–15 (2021)