Computational Visual Media
DOI 10.1007/s41095-016-0063-3

Research Article

Modified filtered importance sampling for virtual spherical Gaussian lights

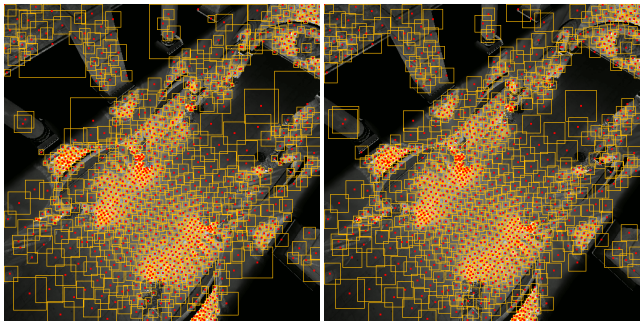
Yusuke Tokuyoshi[1] (✉)

Abstract This paper proposes a modification of the filtered importance sampling method, and improves the quality of virtual spherical Gaussian light (VSGL) based real-time glossy indirect illumination using this modification. The original filtered importance sampling method produces large overlaps of and gaps between filtering kernels for high-frequency probability density functions (PDFs). This is because the size of the filtering kernel is determined using the PDF at the sampled center of the kernel. To reduce those overlaps and gaps, this paper determines the kernel size using the integral of the PDF within the filtering kernel. Our key insight is that these integrals are approximately constant, if kernel centers are sampled using stratified sampling. Therefore, an appropriate kernel size can be obtained by solving this integral equation. Using the proposed kernel size for filtered importance sampling-based VSGL generation, undesirable artifacts are significantly reduced with a negligibly small overhead.

Keywords filtered importance sampling, real-time rendering, global illumination, virtual point lights.

1 Introduction

The filtered importance sampling method [19] is a variance reduction technique of Monte Carlo integration often used for real-time or interactive rendering, which uses filtering kernels instead of sample points. This paper proposes a modification of filtered importance sampling, and improves the quality of virtual spherical Gaussian light (VSGL) [30] based real-time glossy indirect illumination using this modification. The original filtered importance sampling first samples the center of each filtering kernel according to a probability density function (PDF), and then determines the size of each filtering kernel using the PDF at the sampled center. However, this kernel size determination produces large overlaps of and gaps between filtering kernels for high-frequency PDFs (Fig. 1). This is because the kernel size can be too large when the sampled center is at a local minimum of a high-frequency PDF. Therefore, this paper introduces an appropriate kernel size of filtered importance sampling to reduce these overlaps and gaps.



(a) Previous filtering kernels (b) Our filtering kernels

Fig. 1 Sampling VPL clusters from a reflective shadow map based on filtered importance sampling. Red points: kernel centers sampled according to the PDF (brightness). Orange squares: filtering kernels (i.e., VPL clusters). The previous filtered importance sampling (a) produces significant overlaps of and gaps between filtering kernels. Our method (b) reduces these overlaps and gaps.

One effective application of our method is generation of VSGLs using reflective shadow maps [8]. Reflective shadow map-based global illumination is well established for real-time rendering. However, stochastic sampling of virtual point lights (VPLs) [18] (i.e., texels of reflective shadow maps which represent one-bounce light subpaths) produces noticeable variance especially for glossy interreflections. To

1 Square Enix Co., Ltd., Tokyo, 160-8430, Japan. E-mail: tokuyosh@square-enix.com

Manuscript received: 2016-08-31; accepted: 2016-09-21.

reduce this variance, VSGLs were introduced recently. A VSGL approximates a cluster of VPLs using a Gaussian-based representation. Thanks to this representation, the distribution of VPLs can be filtered with a simple summation operation (e.g., mipmaping). In addition, this representation has an analytical solution of the rendering integral for each VSGL. Therefore, if VSGLs are generated from reflective shadow maps inexpensively, we are able to render one-bounce glossy indirect illumination at real-time frame rates.

Tokuyoshi [30] sampled VPL clusters as VSGLs from a reflective shadow map based on filtered importance sampling to achieve real-time frame rates. However, while this approach is simpler and faster than k-means-based VPL clustering [11, 23], it does induce flickering structured artifacts due to the previously mentioned overlaps and gaps. This problem is noticeable when a bidirectional reflective shadow mapping method [25] is used to build the PDF. This is because the bidirectional reflective shadow mapping method produces a dynamic and high-frequency PDF. Using our kernel size, we are able to reduce flickering artifacts significantly for such a high-frequency PDF.

The contributions of our work are as follows:

- An appropriate kernel size of filtered importance sampling is introduced to reduce undesirable overlaps of and gaps between filtering kernels.
- For image-based PDFs, the above kernel size is computed using a simple numerical approach with negligibly small overhead.
- Using the proposed filtered importance sampling method, flickering artifacts are reduced for VSGL-based real-time glossy indirect illumination.

2 Related work

2.1 Sampling with pre-integration

Sampling pre-integrated values is often used for image-based lighting. Structured importance sampling [1] stratifies samples hierarchically, and then the illumination is pre-integrated within each stratum. Debevec [10] subdivided an environment map into regions of equal energy using a median cut algorithm. This method approximates each region with a directional light source whose color is the sum of pixel values within the region. Filtered importance sampling [19] is introduced for glossy materials under environment maps. This technique samples pre-filtered values using a mipmap, thus it performs at real-time frame rates. This paper modifies the kernel size of filtered importance sampling to improve the sampling quality.

2.2 Real-time global illumination

In this paper, we apply the proposed filtered importance sampling technique to real-time global illumination. Interactive global illumination algorithms were surveyed by Ritschel et al. [24]. For a comprehensive survey of VPL-based rendering, we refer the readers to Dachsbacher et al. [7]. Here we pay attention only to the most relevant works.

Virtual point lights (VPLs) are often used for representing indirect illumination [18]. For real-time rendering, single-bounce VPLs lit from point or directional lights can be generated by using reflective shadow maps [8]. Hundreds or thousands of VPLs are often resampled from a reflective shadow map by hierarchical sample warping [5] according to a mipmapped image-based PDF [9]. To generate shadow maps for so many lights, Ritschel et al. [26] proposed imperfect shadow maps. In their follow-up paper [25], a bidirectional reflective shadow mapping method was introduced to estimate a view-dependent importance for the image-based PDF. This method roughly computes the contribution of light paths from the eye to each reflective shadow map texel, and thus generates dynamic and high-frequency PDFs. Therefore, it can increase flickering artifacts, even though the total variance can be reduced. To take such dynamic PDFs into account, Barák et al. [2] introduced a temporally coherent sampling technique based on the Metropolis-Hastings algorithm for static light sources. They also proposed tessellation-based imperfect shadow maps to reduce memory usage. While the original VPL method is theoretically unbiased, variance is visible as spiky artifacts especially for glossy materials [20]. To avoid this problem, VPLs are often clustered and then represented using a smaller number of area lights for interactive rendering.

Area light approximation via VPL clustering. Dong et al. [11] clustered VPLs using the k-means algorithm, and then approximated visibilities of VPLs using a soft shadow map for each cluster. Prutkin et al. [23] clustered texels of a reflective shadow map based on k-means similar to Dong et al., while they approximated the clusters with area lights for analytical radiance evaluation. They sampled cluster centers using the bidirectional reflective shadow mapping method to improve the quality. Luksch et

al. [21] clustered VPLs using a kd-tree to generate virtual polygon lights to update light maps. These virtual area lights were evaluated using analytical form factors which cannot reduce variance caused by high-frequency bidirectional reflectance distribution functions (BRDFs). To allow for all-frequency BRDFs, recently spherical Gaussians have been used.

Spherical Gaussians [32, 38] are often used for approximating the rendering of various types of materials under environment maps or area lights [14, 15, 34, 37, 39]. This is because they have closed-form solutions for the integral, product, and product integral, which are fundamental operations to evaluate rendering integrals. Hence, all-frequency materials can be rendered efficiently. To represent static environment maps using spherical Gaussians, they have been fitted in preprocessing. For dynamic indirect illumination performed at near-interactive frame rates, Xu et al. [36] approximated the outgoing radiance using spherical Gaussians for each triangle primitive lit from distant light sources. Virtual spherical Gaussian lights (VSGLs) [30] were introduced to approximate a set of VPLs. To generate thousands of VSGLs at real-time frame rates, a filtered importance sampling-based approach was used with mipmapped reflective shadow maps. To generate a few VSGLs for more time-sensitive applications such as video games, the total value of all the reflective shadow map texels were computed instead of filtered importance sampling [29]. These methods can render caustics, unlike eye-path tracing-based methods (e.g., cone tracing [6, 35]).

However, filtered importance sampling-based VSGL generation induces a flickering error for high-frequency PDFs due to inappropriate kernel sizes. In this paper, we introduce an appropriate kernel size of filtered importance sampling.

3 Modified filtered importance sampling

3.1 Filtered importance sampling

Filtered importance sampling can be used when the integrand has a 2D image $f(\mathbf{x})$, where $\mathbf{x} \in [0,1]^2$ is the image-space position. This method first samples each kernel center $\mathbf{x}_i \in [0,1]^2$ according to a PDF $p(\mathbf{x})$, and then a filtered value of $f(\mathbf{x})$ is used as each sample value instead of $f(\mathbf{x}_i)$. This filtered value is given by a pre-filtered mipmap as follows:

$$\int_{[0,1]^2} f(\mathbf{x}) \frac{g((\mathbf{x}-\mathbf{x}_i)/s_i)}{a_i} \mathrm{d}\mathbf{x} \approx \bar{f}(\mathbf{x}_i, l_i),$$

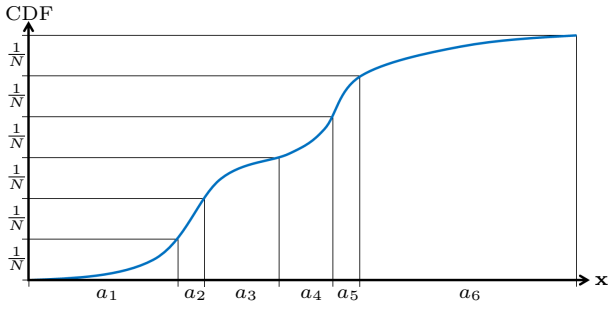


Fig. 2 CDF (blue line) and stratified sampled kernels.

where $g((\mathbf{x}-\mathbf{x}_i)/s_i)$ is the unnormalized filtering kernel which has a fixed maximum, s_i is the kernel size, a_i is the filtering area (i.e., normalization factor) given by $a_i = \int_{[0,1]^2} g((\mathbf{x}-\mathbf{x}_i)/s_i)\mathrm{d}\mathbf{x}$, and $\bar{f}(\mathbf{x}_i, l_i)$ is the mipmapped value of $f(\mathbf{x})$ at mip level l_i. Let M be the number of texels of $f(\mathbf{x})$, then the filtering area a_i is also written as a function of mip level l_i: $a_i = \frac{4^{l_i}}{M}$. Křivánek and Colbert [19] determined mip level l_i by representing this filtering area using the inverse of the density at sampled center $\mathbf{x}_i$ as

$$a_i = \frac{4^{l_i}}{M} = \min\left(\frac{K}{Np(\mathbf{x}_i)}, 1\right), \tag{1}$$

where N is the number of samples, and K is a user-specified parameter to tweak the kernel size (Křivánek and Colbert used $K = 4$). However, this mip level determination is sensitive to the sampled center $\mathbf{x}_i$. When $\mathbf{x}_i$ is at a local minimum of a high-frequency PDF, the filtering kernel can be too large. Conversely, the filtering kernel can also be too small when $\mathbf{x}_i$ is at a local maximum. Therefore, undesirable overlaps of and gaps between filtering kernels can be produced.

3.2 Our filtering kernels

This paper introduces an appropriate kernel size to reduce overlaps of and gaps between filtering kernels for filtered importance sampling. Sampling according to a PDF is done by computing the inverse cumulative distribution function (CDF) of the PDF. As shown in Fig. 2, a sampling interval of the vertical axis is the integral of the PDF within each filtering kernel. Therefore, if kernel centers are sampled using stratified sampling, this integral is almost $\frac{1}{N}$. Hence, an appropriate kernel size s_i is obtained by solving the following integral equation:

$$\int_{[0,1]^2} p(\mathbf{x}) g((\mathbf{x}-\mathbf{x}_i)/s_i)\mathrm{d}\mathbf{x} = \frac{1}{N}. \tag{2}$$

Since the left side is monotonically increasing with respect to the kernel size, we can obtain the kernel size using a bisection method. When PDF $p(\mathbf{x})$ is given by a 2D image, we can use the mipmap of this PDF, which

is also used for sampling $\mathbf{x}_i$ via hierarchical sample warping [5]. Using this mipmap, Eq. (2) is rewritten as

$$\boxed{\frac{4^{l_i}}{M}\bar{p}(\mathbf{x}_i, l_i) = \frac{1}{N}}, \tag{3}$$

where $\bar{p}(\mathbf{x}_i, l_i) \approx \int_{[0,1]^2} p(\mathbf{x})\frac{g((\mathbf{x}-\mathbf{x}_i)/s_i)}{a_i}\mathrm{d}\mathbf{x}$ is the mipmapped value of $p(\mathbf{x})$. In this paper, l_i is calculated using the bisection method with an iteration count of 12.

4 Application to virtual spherical Gaussian lights (VSGLs)

In this paper, we demonstrate generation of VSGLs as an effective application of our filtered sampling. A VSGL represents the positional distribution and total radiant intensity of VPLs using a Gaussian and spherical Gaussians, respectively. Since spherical Gaussians have closed-form solutions to evaluate rendering integrals, all-frequency illumination is computed analytically for each VSGL. The VSGL algorithm is composed of the following five phases: reflective shadow map rendering, PDF building, VSGL generation, shadow map rendering, and shading. This paper improves only on the VSGL generation phase. For the detail of VSGLs, please refer to Appendix A, B, and C.

4.1 Mipmap-based VSGL generation

To generate VSGLs, VPLs are first clustered. Then VPL powers are summed and VPL distributions are averaged for each cluster (Fig. 3). To represent VPL distributions with a Gaussian and spherical Gaussians, weighted averages of emission directions, VPL positions, and squared VPL positions weighted by each VPL power are required (for the detail, please refer to Appendix B). Therefore, a reflective shadow map to store the above VPL power and weighted distribution parameters is generated, and then they are mipmapped to approximately obtain total texel values (Fig. 4). Let $f(\mathbf{x})$ be the reflective shadow map, then the total texel value within the ith VPL cluster centered at $\mathbf{x}_i$ is approximated using the mipmap $\bar{f}(\mathbf{x}_i, l_i)$ as follows:

$$\int_{[0,1]^2} f(\mathbf{x})g((\mathbf{x}-\mathbf{x}_i)/s_i)\mathrm{d}\mathbf{x} \approx \frac{4^{l_i}}{M}\bar{f}(\mathbf{x}_i, l_i),$$

where filtering kernel $g((\mathbf{x}-\mathbf{x}_i)/s_i)$ represents the VPL cluster. To sample the kernel center $\mathbf{x}_i$ and mip level l_i, a filtered importance sampling-based approach can be used. The kernel center $\mathbf{x}_i$ is sampled according to a dynamic and high-frequency view-dependent PDF $p(\mathbf{x})$ given by the bidirectional reflective shadow mapping method. Tokuyoshi [30] determined l_i using Eq. (1) with $K = 1$ according to the previous filtered importance sampling. Using this mip level determination, the total value of reflective shadow map texels within each cluster is given by

$$\int_{[0,1]^2} f(\mathbf{x})g((\mathbf{x}-\mathbf{x}_i)/s_i)\mathrm{d}\mathbf{x} \approx \frac{\bar{f}(\mathbf{x}_i, l_i)}{\max\left(N\bar{p}(\mathbf{x}_i), 1\right)}. \tag{4}$$

However, since the numerator is filtered while the denominator is not, this sampling method can induce an intensive error due to overlaps of and gaps between filtering kernels.

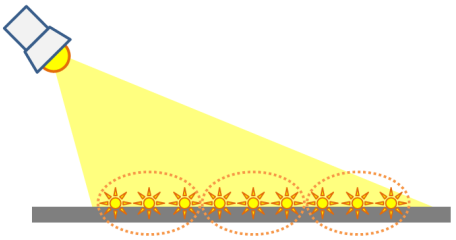


Fig. 3 Clustered VPLs. Each cluster is approximated with a VSGL by computing the total VPL power and averaged VPL distributions within the cluster. These operations are done by filtered sampling on the reflective shadow map.

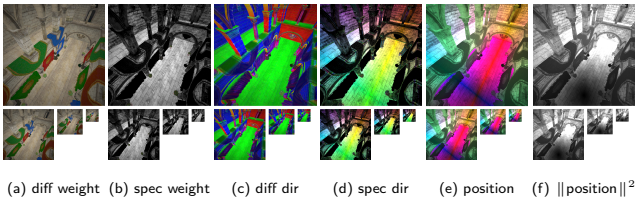


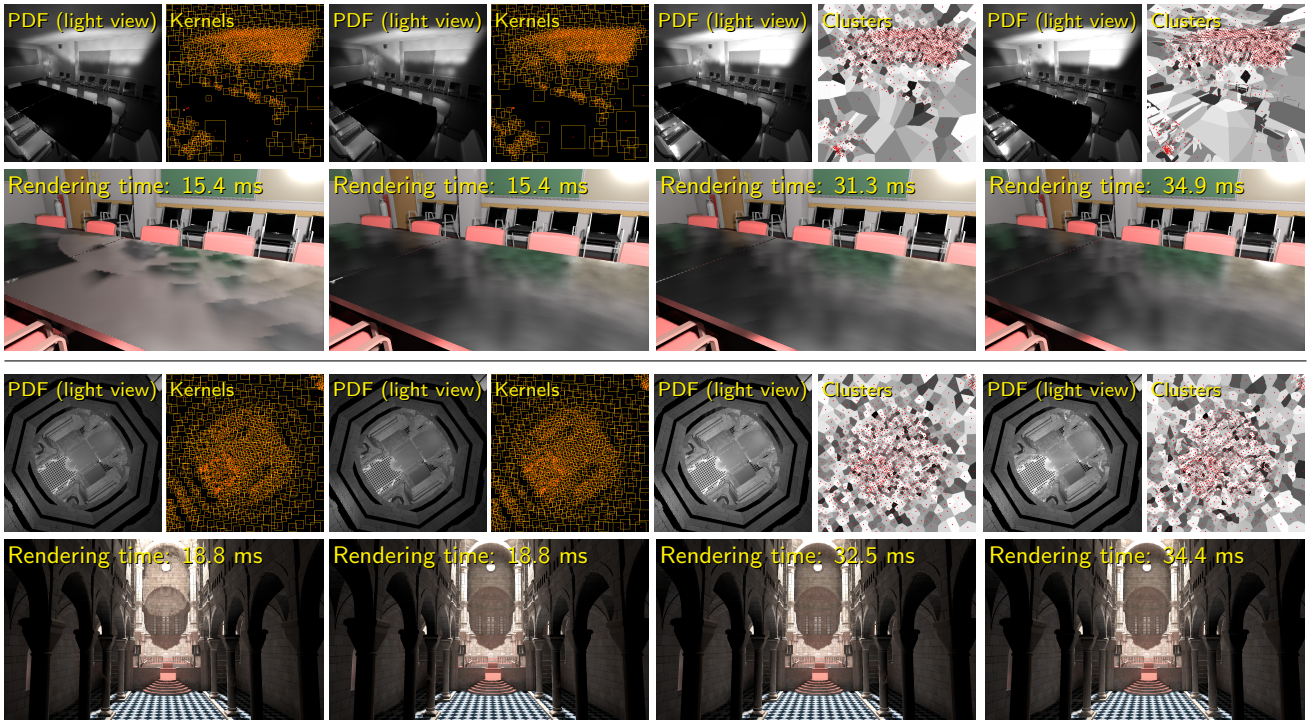
Fig. 4 Mipmapped reflective shadow map for VSGL generation. Average emission directions (c)(d) and positions (e)(f) are weighted by VPL powers (a)(b). VSGLs are sampled from this reflective shadow map based on filtered importance sampling.

4.2 VSGL generation using our filtering kernels

To obtain an appropriate mip level l_i, this paper employs Eq. (3) instead of Eq. (1) for filtered importance sampling-based VSGL generation. Using this mip level l_i, the total texel value within each cluster is approximated as follows:

$$\boxed{\int_{[0,1]^2} f(\mathbf{x})g((\mathbf{x}-\mathbf{x}_i)/s_i)\mathrm{d}\mathbf{x} \approx \frac{\bar{f}(\mathbf{x}_i, l_i)}{N\bar{p}(\mathbf{x}_i, l_i)}}. \tag{5}$$

Unlike Eq. (4), both the numerator and denominator are filtered using the same kernel. Hence, temporal coherence is improved for a dynamic high-frequency PDF. Furthermore, the approximation error can be reduced if PDF $p(\mathbf{x})$ is approximately proportional to $f(\mathbf{x})$, similar to standard importance sampling.



(a) Tokuyoshi 2015 [30] (b) Ours (c) k-means (2D image space) (d) k-means (3D world space)

Fig. 5 Rendered images using different VSGL generation methods for 331k triangles scene (upper row) and 75k triangles scene (lower row). When a sampled kernel center is at a local minimum of the PDF, the previous method (a) produces too bright of a VSGL with low probability. On the other hand, our method (b) does not produce such an error similar to k-means-based approaches (c)(d). Aliasing artifacts on the glossy table in the upper row are the shadow acne of imperfect shadow maps.

Controlling the kernel size. For Eq. (5), the mip level l_i affects only the filter bandwidth. Therefore, the user-specified parameter K can also be used for calculating l_i in our case. This is implemented using $\frac{K}{N}$ instead of $\frac{1}{N}$ in Eq. (3). Using $K > 1$, the temporal coherence is improved, though an overblurring error is induced. This overblurring error is reduced by increasing the number of samples N, similar to the original filtered importance sampling.

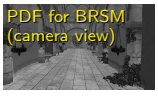
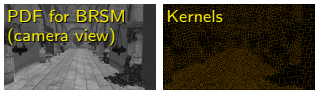
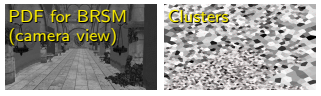
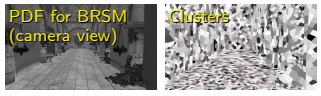
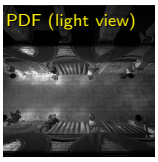
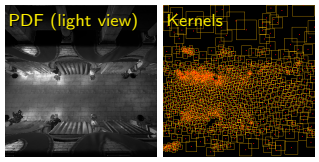
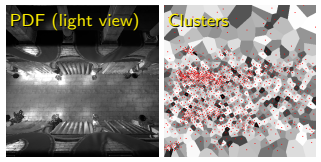
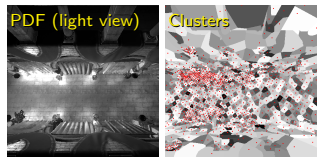
5 Experimental results

Here we present rendering results using 1024 VSGLs generated using our filtered importance sampling with $K = 1$ on an NVIDIA® GeForce® GTX™ 970 GPU. The frame buffer and reflective shadow map (RSM) resolutions are 1920×1088 and 512^2, respectively. A tessellation-based imperfect shadow map [2] of resolution 64^2 is employed to evaluate the visibility of each VSGL. To estimate a view-dependent PDF on the reflective shadow map using the bidirectional reflective shadow mapping method, 2048 VSGLs without shadow maps are generated on the G-buffer. For the PDF on the G-buffer, reflectance is used. To perform stratified sampling, the Fibonacci lattice point set using a golden ratio approximation [27] is employed as a quasi-random number. For comparison, this paper uses k-means clustering using 2D image space and 3D world space. In these k-means-based approaches, once clusters are assigned to all the texels, those texels are sorted by cluster ID. Then, to compute the total value of clustered texels, a thread is dispatched for each cluster similar to Prutkin et al. [23]. For implementation details, please refer to Appendix D.

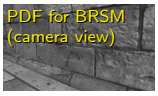
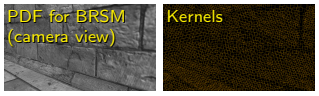
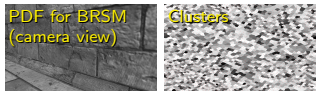
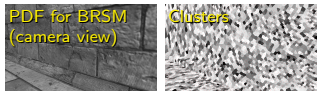
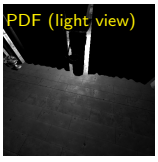
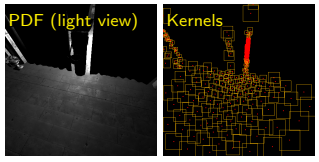
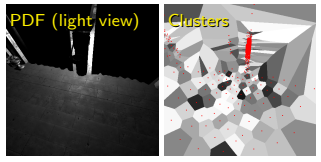
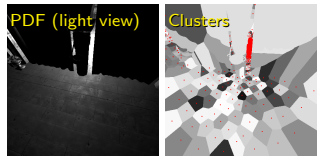
Quality. Fig. 5 shows rendered images using different VSGL generation methods. Using the previous kernel size (a), intense artifacts can be produced with low probability, though this sampling method is faster than the k-means-based approaches (c)(d). This is because too large of a filtering kernel is produced when the sampled kernel center is at a local minimum of the PDF. On the other hand, our kernel size (b) does not produce these undesirable filtering kernels nor does it noticeably sacrifice performance.

Performance. Tab. 1 shows the computation times of VSGL generation both for bidirectional reflective shadow mapping (BRSM) (upper row) and final

Tab. 1 Computation times of VSGL generation (ms).

Tokuyoshi 2015 [30]		Ours		k-means (2D image space)		k-means (3D world space)	
							
Additional G-buffer:	0.816	Additional G-buffer:	0.816	Cluster assignment:	3.004	Cluster assignment:	5.009
Mipmapping:	0.968	Mipmapping:	0.968	Sort:	1.541	Sort:	1.517
Filtered sampling:	0.090	Filtered sampling:	0.092	Sum:	9.096	Sum:	9.856
							
Additional RSM:	0.165	Additional RSM:	0.165	Cluster assignment:	0.619	Cluster assignment:	0.860
Mipmapping:	0.201	Mipmapping:	0.201	Sort:	0.306	Sort:	0.299
Filtered sampling:	0.084	Filtered sampling:	0.088	Sum:	1.995	Sum:	4.231
Total:	2.324	Total:	2.330	Total:	16.561	Total:	21.772

Tab. 2 Computation times of VSGL generation (ms).

Tokuyoshi 2015 [30]		Ours		k-means (2D image space)		k-means (3D world space)	
							
Additional G-buffer:	0.752	Additional G-buffer:	0.752	Cluster assignment:	3.262	Cluster assignment:	5.756
Mipmapping:	0.985	Mipmapping:	0.985	Sort:	1.523	Sort:	1.499
Filtered sampling:	0.087	Filtered sampling:	0.089	Sum:	8.850	Sum:	8.735
							
Additional RSM:	0.154	Additional RSM:	0.154	Cluster assignment:	1.172	Cluster assignment:	0.899
Mipmapping:	0.199	Mipmapping:	0.199	Sort:	0.306	Sort:	0.305
Filtered sampling:	0.065	Filtered sampling:	0.067	Sum:	5.275	Sum:	20.489
Total:	2.242	Total:	2.246	Total:	20.388	Total:	37.683

Tab. 3 Code size of VSGL generation (C++ and HLSL)

	Tokuyoshi 2015 [30]	Ours	k-means (2D)	k-means (3D)
Number of shaders	2	2	9	9
Number of dispatch calls	2	2	39	39
Lines of code	222	232	1143	1172

shading (lower row). Our contribution is written in red. Although our method is a numerical approach, its overhead is a total of about five microseconds more compared to the previous filtered importance sampling-based generation. In addition, our method is about 7-9 times faster than the k-means-based approaches. The difference is significant especially for the bidirectional reflective shadow mapping method, which uses a higher-resolution G-buffer (1920×1088) than the reflective shadow map (512^2). Tab. 2 shows the computation times using different PDFs. For these PDFs, the performance of k-means-based approaches is more expensive than Tab. 1. This is because the last pass "Sum" (which is the summation of texel

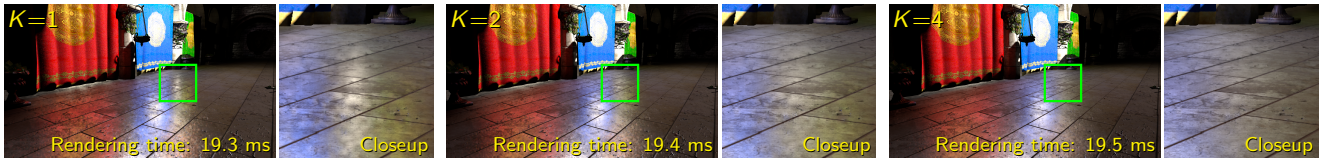
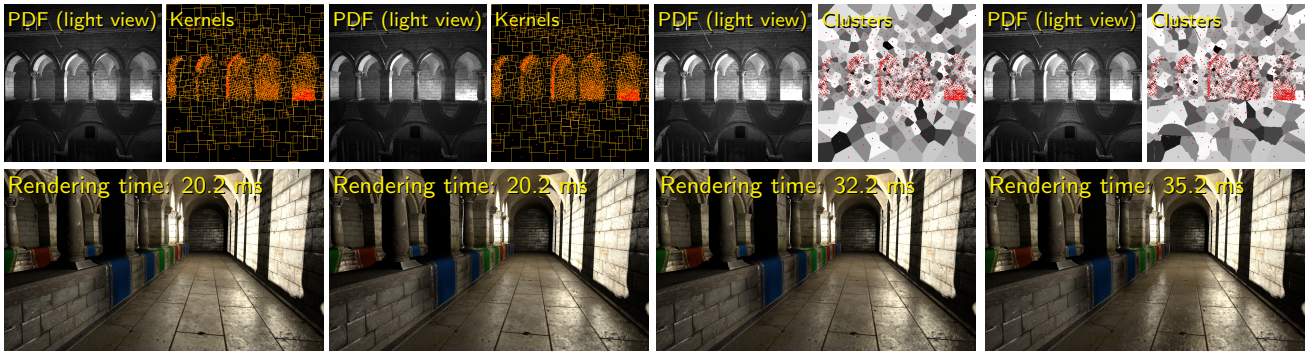


Fig. 6 Unlike k-means-based approaches, our approach can control the kernel size using the user-specified parameter K. By increasing K, the temporal coherence is improved, while some illumination appearance is overblurred (262k triangles scene).



(a) Tokuyoshi 2015 [30] (b) Ours (c) k-means (2D image space) (d) k-means (3D world space)

Fig. 7 Light occluded by columns (262k triangles scene). Since filtered importance sampling-based approaches (a)(b) ignore the difference of world space positions similar to the k-means-based approach using image space (c), they blur some indirect illumination for this scene. These low-frequency errors are visually acceptable compared to high-frequency artifacts.

values based on Prutkin et al.'s implementation) has a linear complexity with respect to the number of texels within a cluster. Comparatively, the performance of our approach is almost independent of the PDFs, because it uses pre-filtered mipmaps. Hence, the proposed method is suitable for applications which require stable performance.

Code size. Tab. 3 shows the code size of VSGL generation in our implementation. Filtered importance sampling-based approaches use only two compute shaders. One is for the calculation of the additional reflective shadow map (or G-buffer), and the other is for the filtered sampling of VSGLs. The difference of our method from the previous method [30] is only the mip level determination (10 lines of code). On the other hand, the k-means-based approaches require more compute shaders than ours. In addition, some of them are dispatched iteratively for the GPU sort. Our method is about five times fewer lines of code than the k-means-based approaches.

Kernel size controlling. As shown in Fig. 6, the kernel size of our method is controllable by using the user-specified parameter K unlike k-means-based approaches. Although some illumination appearance is overblurred by using $K > 1$, the temporal coherence is improved. The parameter K can be used to balance illumination details and temporal coherence according to the liking of a user.

6 Limitations

Feature space. As shown in Fig. 7, filtered importance sampling-based approaches (a)(b) ignore the difference of world space positions similar to the k-means-based-approach using image space (c). If VPLs are clustered ignoring such high-dimensional features, some indirect illumination is blurred when using VSGLs. These low-frequency errors are a limitation of filtered importance sampling-based VSGL generation to achieve real-time frame rates, but they are more visually acceptable than high-frequency artifacts (e.g., flickering and spiky artifacts).

Overlaps and gaps. Although our method reduces overlaps of and gaps between filtering kernels, they cannot be removed completely for inhomogeneous sample distributions. This problem is alleviated by using stratified sampling.

PDF. Since our method requires a given PDF, it cannot be applied to sampling strategies without the PDF (e.g., sequential Monte Carlo instant radiosity [13]).

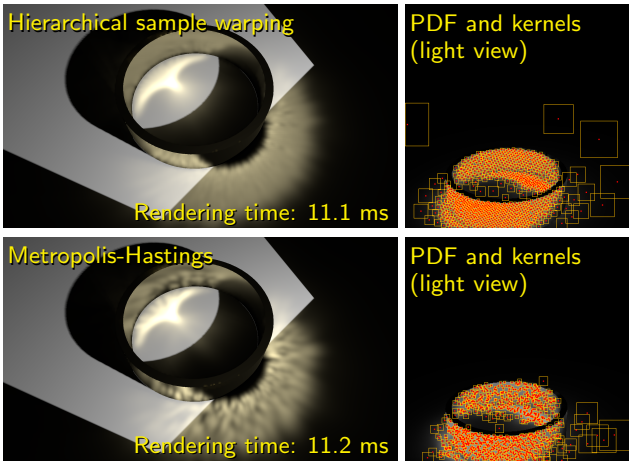


Fig. 8 Caustics rendered using our method with the same PDF (514 triangles scene). Kernel centers of the upper row and lower row are generated using hierarchical sample warping [5] and the Metropolis-Hastings-based temporally coherent sampling [2], respectively. Due to a lack of stratification, Metropolis-Hastings produces noticeable artifacts.

Temporal coherence. Since our method improves only the kernel size, kernel centers can still be temporally incoherent for dynamic PDFs. Although the Metropolis-Hastings algorithm can be used for temporally coherent sampling [2], it is limited to static light sources and has a lack of stratification. This problem induces noticeable artifacts especially for caustics (Fig. 8). Therefore, this paper employs hierarchical sample warping for stratified sampling. If the temporal coherence is more important than detailed illumination, $K > 1$ can be used for our method to improve the temporal coherence.

7 Conclusions

This paper improved the kernel size of filtered importance sampling to reduce overlaps of and gaps between filtering kernels. Using this modification for VSGL generation, we are able to render glossy indirect illumination with fewer artifacts than the previous VSGL generation. The overhead of our method is about five microseconds for thousands of VSGLs on a commodity GPU. Although the filtered importance sampling-based approach cannot take into account the difference of higher-dimensional features (e.g., world position) unlike k-means-based approaches, it is simple, fast, and has stable performance. This paper has demonstrated VSGL-based dynamic glossy indirect illumination, but our method is also usable for spherical Gaussian light generation for dynamic environment maps. Since environment maps are 2D light distribution, it might be more suitable than VSGL generation. We would like to investigate its efficiency in the future.

A Spherical Gaussians

A spherical Gaussian is a type of spherical function and is represented using a Gaussian function γ with respect to a direction vector $\boldsymbol{\omega} \in \mathrm{S}^2$ as follows:

$$G(\boldsymbol{\omega}, \boldsymbol{\xi}, \lambda) = \gamma\left(\|\boldsymbol{\omega} - \boldsymbol{\xi}\|, \frac{1}{\lambda}\right) = \mathrm{e}^{-\frac{\lambda}{2}\|\boldsymbol{\omega}-\boldsymbol{\xi}\|^2} = \mathrm{e}^{\lambda((\boldsymbol{\omega}\cdot\boldsymbol{\xi})-1)},$$

where $\boldsymbol{\xi} \in \mathrm{S}^2$ is the lobe axis, and λ is the lobe sharpness. $\boldsymbol{\xi}$ and $\frac{1}{\lambda}$ correspond to the mean and variance for the Gaussian function, respectively. The integral of a spherical Gaussian is given by

$$A(\lambda) = \int_{\mathrm{S}^2} G(\boldsymbol{\omega}, \boldsymbol{\xi}, \lambda)\mathrm{d}\boldsymbol{\omega} = \frac{2\pi}{\lambda}\left(1 - \mathrm{e}^{-2\lambda}\right).$$

A normalized spherical Gaussian $\frac{G(\boldsymbol{\omega},\boldsymbol{\xi},\lambda)}{A(\lambda)}$ is known as the Von Mises-Fisher distribution. For VSGLs, this distribution is used for representing reflection lobes.

A.1 Spherical Gaussian approximation of reflection lobes

Diffuse lobes. For the Lambert BRDF ρ_d, the diffuse reflection lobe can be approximated with a spherical Gaussian taking energy conservation into account as follows:

$$\boxed{\rho_d(\mathbf{y}, \boldsymbol{\omega}', \boldsymbol{\omega})\langle\boldsymbol{\omega}, \mathbf{n}\rangle \approx R_d \frac{G(\boldsymbol{\omega}, \mathbf{n}, \lambda_d)}{A(\lambda_d)}}, \tag{6}$$

where $\rho_d(\mathbf{y}, \boldsymbol{\omega}', \boldsymbol{\omega}) = \frac{R_d}{\pi}$, R_d is the diffuse reflectance, $\boldsymbol{\omega}' \in \mathrm{S}^2$ is the incoming direction, $\mathbf{n} \in \mathrm{S}^2$ is the surface normal at the world position $\mathbf{y} \in \mathbb{R}^3$, $\langle\boldsymbol{\omega}, \mathbf{n}\rangle = \max(\boldsymbol{\omega} \cdot \mathbf{n}, 0)$ is the clamped dot product, and $\lambda_d \approx 2$ is the sharpness of the diffuse lobe which is obtained by using the least square method.

Specular lobes. For the microfacet BRDF ρ_s, the specular reflection lobe is fitted with a single spherical Gaussian by using Wang et al. [34]'s analytical approximation. The BRDF is separated into two factors: the unnormalized normal distribution function $D(\boldsymbol{\omega}_h)$ whose maximum is one, and the rest of the factors $C(\boldsymbol{\omega})$ as follows:

$$\rho_s(\mathbf{y}, \boldsymbol{\omega}', \boldsymbol{\omega})\langle\boldsymbol{\omega}, \mathbf{n}\rangle = C(\boldsymbol{\omega})D(\boldsymbol{\omega}_h),$$

where $\boldsymbol{\omega}_h = \frac{\boldsymbol{\omega}+\boldsymbol{\omega}'}{\|\boldsymbol{\omega}+\boldsymbol{\omega}'\|}$ is the halfway vector of the incoming direction and outgoing direction. Bell-shaped normal distribution functions (e.g., Phong [4], Beckmann [3] and GGX [31, 33] distributions) can be approximated with a spherical Gaussian as

$$D(\boldsymbol{\omega}_h) \approx G(\boldsymbol{\omega}_h, \mathbf{n}, \lambda_h).$$

For Beckmann or GGX normal distribution functions, $\lambda_h = \frac{2}{\alpha^2}$ where α is the roughness parameter. Using spherical warping, this can be approximated with a function of $\boldsymbol{\omega}$ as

$$G(\boldsymbol{\omega}_h, \mathbf{n}, \lambda_h) \approx G(\boldsymbol{\omega}, \boldsymbol{\xi}_s, \lambda_s),$$

where $\boldsymbol{\xi}_s$ is the reflection vector given by $\boldsymbol{\xi}_s = 2(\boldsymbol{\omega}' \cdot \mathbf{n})\mathbf{n} - \boldsymbol{\omega}'$, and $\lambda_s = \frac{\lambda_h}{4|\boldsymbol{\xi}_s \cdot \mathbf{n}|}$. Hence, the specular lobe is approximated with the following equation:

$$\rho_s(\mathbf{y}, \boldsymbol{\omega}', \boldsymbol{\omega})\langle \boldsymbol{\omega}, \mathbf{n}\rangle \approx C(\boldsymbol{\omega}) G(\boldsymbol{\omega}, \boldsymbol{\xi}_s, \lambda_s).$$

Moreover, since microfacet BRDFs mostly preserve energy for highly glossy surfaces, the specular lobe can be approximated using a normalized spherical Gaussian as follows:

$$\boxed{\rho_s(\mathbf{y}, \boldsymbol{\omega}', \boldsymbol{\omega})\langle \boldsymbol{\omega}, \mathbf{n}\rangle \approx R_s \frac{G(\boldsymbol{\omega}, \boldsymbol{\xi}_s, \lambda_s)}{A(\lambda_s)}}, \tag{7}$$

where R_s is the specular reflectance. Anisotropic spherical Gaussians [38] are also usable in the same manner.

B Virtual spherical Gaussian lights (VSGLs)

This paper approximates a cluster of VPLs with a VSGL. For a VSGL, the total radiant intensity and positional distribution of VPLs are represented using a spherical Gaussian and isotropic Gaussian distribution respectively. This representation can be computed using a simple summation operation.

B.1 Radiant intensity

The radiant intensity of the jth VPL is given as

$$I_j(\boldsymbol{\omega}) = \Phi_j \rho(\mathbf{y}_j, \boldsymbol{\omega}'_j, \boldsymbol{\omega})\langle \boldsymbol{\omega}, \mathbf{n}_j\rangle,$$

where Φ_j is the power of the jth photon emitted from the light source, $\boldsymbol{\omega}'_j \in \mathrm{S}^2$ is the incoming direction of the photon, and $\mathbf{n}_j \in \mathrm{S}^2$ is the surface normal at the VPL position $\mathbf{y}_j \in \mathbb{R}^3$, and $\rho(\mathbf{y}_j, \boldsymbol{\omega}'_j, \boldsymbol{\omega})$ is the BRDF. This paper first divides this BRDF into diffuse and specular components (i.e., ρ_d and ρ_s). Then, the total radiant intensity of clustered VPLs is approximated with a single spherical Gaussian for each component by using Toksvig [28]'s filtering. For ease of explanation, this subsection hereafter describes only a single BRDF component. The total radiant intensity of a VPL cluster $\mathbb{S}$ is represented as

$$I_v(\boldsymbol{\omega}) = \sum_{j \in \mathbb{S}} I_j(\boldsymbol{\omega}) \approx c_v G\left(\boldsymbol{\omega}, \boldsymbol{\xi}_v, \lambda_v\right).$$

To compute spherical Gaussian parameters c_v, $\boldsymbol{\xi}_v$ and λ_v efficiently, each reflection lobe is approximated using Eq. (6) or Eq. (7) as follows:

$$\begin{aligned} I_v(\boldsymbol{\omega}) &= \sum_{j \in \mathbb{S}} \Phi_j \rho(\mathbf{y}_j, \boldsymbol{\omega}'_j, \boldsymbol{\omega})\langle \boldsymbol{\omega}, \mathbf{n}_j\rangle \\ &\approx \sum_{j \in \mathbb{S}} \Phi_j R_j \frac{G\left(\boldsymbol{\omega}, \boldsymbol{\xi}_j, \lambda_j\right)}{A(\lambda_j)} \\ &= \left(\sum_{j \in \mathbb{S}} \Phi_j R_j\right) \frac{\sum_{j \in \mathbb{S}} \Phi_j R_j \frac{G\left(\boldsymbol{\omega}, \boldsymbol{\xi}_j, \lambda_j\right)}{A(\lambda_j)}}{\sum_{j \in \mathbb{S}} \Phi_j R_j}, \end{aligned}$$

where R_j is the reflectance, and $\boldsymbol{\xi}_j$ and λ_j are the axis and sharpness of the reflection lobe at the jth VPL. Then, the weighted average of the normalized spherical Gaussians weighted by the VPL power $\Phi_j R_j$ is approximated with a single spherical Gaussian as

$$\frac{\sum_{j \in \mathbb{S}} \Phi_j R_j \frac{G\left(\boldsymbol{\omega}, \boldsymbol{\xi}_j, \lambda_j\right)}{A(\lambda_j)}}{\sum_{j \in \mathbb{S}} \Phi_j R_j} \approx \frac{G\left(\boldsymbol{\omega}, \boldsymbol{\xi}_v, \lambda_v\right)}{A(\lambda_v)}.$$

Using Toksvig's filtering, the jth normalized spherical Gaussian is first approximately converted into its averaged direction as $\bar{\boldsymbol{\xi}}_j = \frac{\lambda_j}{\lambda_j + 1}\boldsymbol{\xi}_j$. Next, the weighted average of the directions is computed by

$$\bar{\boldsymbol{\xi}}_v = \frac{\sum_{j \in \mathbb{S}} \Phi_j R_j \bar{\boldsymbol{\xi}}_j}{\sum_{j \in \mathbb{S}} \Phi_j R_j}.$$

Finally, the filtered spherical Gaussian is obtained from the weighted average direction as $\boldsymbol{\xi}_v = \frac{\bar{\boldsymbol{\xi}}_v}{\|\bar{\boldsymbol{\xi}}_v\|}$, $\lambda_v = \frac{\|\bar{\boldsymbol{\xi}}_v\|}{1 - \|\bar{\boldsymbol{\xi}}_v\|}$. The coefficient c_v is given by $c_v = \frac{\sum_{j \in \mathbb{S}} \Phi_j R_j}{A(\lambda_v)}$.

B.2 Positional distribution

In this paper, the positional distribution of VPLs is represented with a single isotropic Gaussian distribution for a VSGL. Unlike radiant intensity, this distribution is not divided into diffuse and specular components in order to avoid the increase of visibility tests (i.e., shadow maps). The weighted mean of VPL positions is computed by

$$\boldsymbol{\mu}_v = \frac{\sum_{j \in \mathbb{S}} \Phi_j (R_{d,j} + R_{s,j}) \mathbf{y}_j}{\sum_{j \in \mathbb{S}} \Phi_j (R_{d,j} + R_{s,j})},$$

where $R_{d,j}$ and $R_{s,j}$ are the diffuse reflectance and specular reflectance at the jth VPL, respectively. The positional variance is also calculated using weighted average as

$$\sigma_v^2 = \frac{\sum_{j \in \mathbb{S}} \Phi_j (R_{d,j} + R_{s,j}) \|\mathbf{y}_j\|^2}{\sum_{j \in \mathbb{S}} \Phi_j (R_{d,j} + R_{s,j})} - \|\boldsymbol{\mu}_v\|^2.$$

Assuming VPLs are distributed on a planar surface, the emitted radiance of a VSGL is represented as follows:

$$\boxed{L_e(\mathbf{y}, \boldsymbol{\omega}) \approx \frac{I_v(\boldsymbol{\omega})}{2\pi\sigma_v^2 |\boldsymbol{\omega} \cdot \mathbf{n}|} \gamma\left(\|\mathbf{y} - \boldsymbol{\mu}_v\|, \sigma_v^2\right)}, \tag{8}$$

where $\mathbf{n}$ is the surface normal which will be eliminated in shading (§C.1).

B.3 VSGL generation using reflective shadow maps

As mentioned in §B.1 and B.2, a VSGL is generated by calculating the total VPL power $\sum_{j\in\mathbb{S}}\Phi_j R_j$, total weighted emission direction $\sum_{j\in\mathbb{S}}\Phi_j R_j\boldsymbol{\xi}_j$, total weighted position $\sum_{j\in\mathbb{S}}\Phi_j(R_{d,j}+R_{s,j})\mathbf{y}_j$, and total weighted squared norm of the position $\sum_{j\in\mathbb{S}}\Phi_j(R_{d,j}+R_{s,j})\|\mathbf{y}_j\|^2$. Therefore, these values are stored into reflective shadow maps, and then they are mipmapped to obtain the total values. The ith VPL cluster is represented by the unnormalized filtering kernel $g\left((\mathbf{x}-\mathbf{x}_i)/s_i\right)$ on the reflective shadow map. For example, let $f(\mathbf{x})$ be VPL power stored in the reflective shadow map, then the total VPL power of the ith VPL cluster is given by

$$\sum_{j\in\mathbb{S}}\Phi_j R_j = \int_{[0,1]^2} f(\mathbf{x})g\left((\mathbf{x}-\mathbf{x}_i)/s_i\right)\mathrm{d}\mathbf{x} \approx \frac{4^{l_i}}{M}\bar{f}(\mathbf{x}_i,l_i).$$

We are also able to calculate the total weighted emission direction, total weighted position, and total weighted squared norm of the position in the same manner. In this paper, the image-space position $\mathbf{x}_i$ and mip level l_i are sampled based on filtered importance sampling.

C Shading

For each shading point $\mathbf{y}_p$ with view direction $\boldsymbol{\omega}_p$, the reflected radiance is calculated using the rendering equation [16] defined by

$$L(\mathbf{y}_p,\boldsymbol{\omega}_p) = \int_{\mathrm{S}^2} L_{in}(\mathbf{y}_p,\boldsymbol{\omega})\rho(\mathbf{y}_p,\boldsymbol{\omega}_p,\boldsymbol{\omega})\langle\boldsymbol{\omega},\mathbf{n}_p\rangle\mathrm{d}\boldsymbol{\omega}, \quad (9)$$

where $L_{in}(\mathbf{y}_p,\boldsymbol{\omega})$ is the incoming radiance, and $\mathbf{n}_p$ is the surface normal at the shading point. This paper approximates the incoming radiance using spherical Gaussians for the analytical approximation of the rendering integral [34, 38].

C.1 Incoming radiance

Using Eq. (8), the approximated incoming radiance is given by

$$\begin{aligned} L_{in}(\mathbf{y}_p,\boldsymbol{\omega}) &= V(\mathbf{y}_p,\mathbf{y})L_e(\mathbf{y},-\boldsymbol{\omega}) \\ &\approx \frac{V(\mathbf{y}_p,\boldsymbol{\mu}_v)I_v(-\boldsymbol{\omega})}{2\pi\sigma_v^2|\boldsymbol{\omega}\cdot\mathbf{n}|}\gamma\left(\|\mathbf{y}-\boldsymbol{\mu}_v\|,\sigma_v^2\right), \end{aligned} \quad (10)$$

where $\boldsymbol{\omega}=\frac{\mathbf{y}-\mathbf{y}_p}{\|\mathbf{y}-\mathbf{y}_p\|}$, $V(\mathbf{y}_p,\boldsymbol{\mu}_v)$ is the visibility between $\mathbf{y}_p$ and $\boldsymbol{\mu}_v$ obtained from a shadow map. The position $\mathbf{y}$ is assumed to be on the planar surface defined by the normal $\mathbf{n}$ and position $\boldsymbol{\mu}_v$. However, $\mathbf{n}$ and $\mathbf{y}$ are unknown for shading. Therefore, we project the positional distribution onto a sphere centered at a shading point instead. To correct the energy for this projection, $|\boldsymbol{\omega}\cdot\mathbf{n}|$ is multiplied similar to virtual spherical lights [12]. Since it is divided by $|\boldsymbol{\omega}\cdot\mathbf{n}|$, $\mathbf{n}$ is eliminated. This is reasonable because the actual surface normal distribution is taken into account by the radiant intensity $I_v(-\boldsymbol{\omega})$. Therefore, Eq. (10) is approximated with the following equation:

$$L_{in}(\mathbf{y}_p,\boldsymbol{\omega}) \approx \frac{I_v(-\boldsymbol{\omega})}{2\pi\sigma_v^2}\gamma\left(\|\mathbf{y}_r-\boldsymbol{\mu}_v\|,\sigma_v^2\right),$$

where $\boldsymbol{\omega}=\frac{\mathbf{y}_r-\mathbf{y}_p}{\|\mathbf{y}_r-\mathbf{y}_p\|}$, and $\mathbf{y}_r$ is the position on the sphere defined by the center $\mathbf{y}_p$ and radius $\|\boldsymbol{\mu}_r-\mathbf{y}_p\|$. This is derived assuming a small σ_v or large radius, but it does not produce noticeable artifacts in practice for a large σ_v and small radius. The Gaussian term can be rewritten into a spherical Gaussian as

$$\gamma\left(\|\mathbf{y}_r-\boldsymbol{\mu}_v\|,\sigma_v^2\right) = G(\boldsymbol{\omega},\boldsymbol{\xi}_\mu,\lambda_\sigma), \quad (11)$$

where $\boldsymbol{\xi}_\mu=\frac{\boldsymbol{\mu}_v-\mathbf{y}_p}{\|\boldsymbol{\mu}_v-\mathbf{y}_p\|}$, and $\lambda_\sigma=\frac{\|\boldsymbol{\mu}_v-\mathbf{y}_p\|^2}{\sigma_v^2}$. This spherical Gaussian represents the spherical region of the VSGL viewed from $\mathbf{y}_p$. Using Eq. (11), the incoming radiance is approximated with the product of two spherical Gaussians which yields a spherical Gaussian as follows:

$$\begin{aligned} L_{in}(\mathbf{y}_p,\boldsymbol{\omega}) &\approx \frac{c_v}{2\pi\sigma_v^2}G\left(\boldsymbol{\omega},-\boldsymbol{\xi}_v,\lambda_v\right)G(\boldsymbol{\omega},\boldsymbol{\xi}_\mu,\lambda_\sigma) \\ &= \boxed{c_{in}G\left(\boldsymbol{\omega},\boldsymbol{\xi}_{in},\lambda_{in}\right)}, \end{aligned} \quad (12)$$

where $\xi_{in}=\frac{\lambda_\sigma\boldsymbol{\xi}_\mu-\lambda_v\boldsymbol{\xi}_v}{\|\lambda_\sigma\boldsymbol{\xi}_\mu-\lambda_v\boldsymbol{\xi}_v\|}$, $\lambda_{in}=\|\lambda_\sigma\boldsymbol{\xi}_\mu-\lambda_v\boldsymbol{\xi}_v\|$, and $c_{in}=\frac{c_v}{2\pi\sigma_v^2}\mathrm{e}^{\lambda_{in}-\lambda_v-\lambda_\sigma}$.

C.2 Shading via product integrals of spherical Gaussians

Since the reflection lobe $\rho(\mathbf{y}_p,\boldsymbol{\omega}_p,\boldsymbol{\omega})\langle\boldsymbol{\omega},\mathbf{n}_p\rangle$ can be approximated using spherical Gaussians and anisotropic spherical Gaussians, Eq. (9) can be calculated using the analytical product integral.

Diffuse reflection. Using Eq. (6) and Eq. (12), the rendering integral of the diffuse component is calculated using the analytical product integral of two spherical Gaussians. This approach is efficient for a few VSGLs [29]. However, a light leak error caused by the spherical Gaussian approximation of reflection lobes cannot be reduced by increasing the number of VSGLs. Unlike the secondary bounce represented by VSGLs, light leaks are noticeable at the first bounce which is more visually important. Therefore, for thousands of VSGLs, the cosine factor at the first bounce is assumed to be a constant and pulled out of the integral [34] as follows:

$$\begin{aligned} L_d(\mathbf{x}_p,\boldsymbol{\omega}_p) &= \int_{\mathrm{S}^2} L_{in}(\mathbf{x}_p,\boldsymbol{\omega})\rho_d(\mathbf{x}_p,\boldsymbol{\omega}_p,\boldsymbol{\omega})\langle\boldsymbol{\omega},\mathbf{n}_p\rangle\mathrm{d}\boldsymbol{\omega} \\ &\approx \frac{c_{in}R_d}{\pi}A(\lambda_{in})\langle\boldsymbol{\xi}_{in},\mathbf{n}_p\rangle. \end{aligned}$$

In addition, when λ_{in} is not small, $A(\lambda_{in}) \approx \frac{2\pi}{\lambda_{in}}$ can be assumed [14]. Therefore, diffuse reflection is inexpensively calculated using the following equation:

$$\boxed{L_d(\mathbf{x}_p, \boldsymbol{\omega}_p) \approx \frac{2c_{in}R_d}{\lambda_{in}}\langle \boldsymbol{\xi}_{in}, \mathbf{n}_p\rangle}.$$

Specular reflection. While spherical Gaussians are used for VSGLs, this paper employs an anisotropic spherical Gaussian to approximate a specular lobe at a shading point. This is because a specular lobe can be anisotropic even if it is an isotropic BRDF model, especially for shallow grazing angles. For simplicity, anisotropic spherical Gaussians are used only for the first bounce which is more visually important than the second bounce. In addition, the product integral of a spherical Gaussian and anisotropic spherical Gaussian [38] has a reasonable computation cost. An anisotropic spherical Gaussian is defined as

$$\acute{G}(\boldsymbol{\omega}, \boldsymbol{\xi}_x, \boldsymbol{\xi}_y, \boldsymbol{\xi}_z, \eta_x, \eta_y) = \langle \boldsymbol{\omega}, \boldsymbol{\xi}_z\rangle \mathrm{e}^{-\eta_x(\boldsymbol{\omega}\cdot\boldsymbol{\xi}_x)^2 - \eta_y(\boldsymbol{\omega}\cdot\boldsymbol{\xi}_y)^2},$$

where $\boldsymbol{\xi}_x, \boldsymbol{\xi}_y, \boldsymbol{\xi}_z$ are orthonormal vectors, and η_x, η_y are the bandwidth parameters. Since a specular lobe is approximated with an anisotropic spherical Gaussian as $\rho_s(\mathbf{y}_p, \boldsymbol{\omega}_p, \boldsymbol{\omega})\langle \boldsymbol{\omega}, \mathbf{n}_p\rangle \approx C(\boldsymbol{\omega})\acute{G}(\boldsymbol{\omega}, \boldsymbol{\xi}_x, \boldsymbol{\xi}_y, \boldsymbol{\xi}_z, \eta_x, \eta_y)$, the rendering integral is calculated as

$$\begin{aligned} L_s(\mathbf{y}_p, \boldsymbol{\omega}_p) &= \int_{\mathrm{S}^2} L_{in}(\mathbf{y}_p, \boldsymbol{\omega})\rho_s(\mathbf{y}_p, \boldsymbol{\omega}_p, \boldsymbol{\omega})\langle \boldsymbol{\omega}, \mathbf{n}_p\rangle \mathrm{d}\boldsymbol{\omega} \\ &\approx c_{in}C(\boldsymbol{\xi}_{in}) \int_{\mathrm{S}^2} G(\boldsymbol{\omega}, \boldsymbol{\xi}_{in}, \lambda_{in})\acute{G}\left(\boldsymbol{\omega}, \boldsymbol{\xi}_x, \boldsymbol{\xi}_y, \boldsymbol{\xi}_z, \eta_x, \eta_y\right) \mathrm{d}\boldsymbol{\omega} \\ &\approx \boxed{\frac{\pi c_{in}C(\boldsymbol{\xi}_{in})\acute{G}\left(\boldsymbol{\xi}_{in}, \boldsymbol{\xi}_x, \boldsymbol{\xi}_y, \boldsymbol{\xi}_z, \frac{\eta_x\nu}{\eta_x+\nu}, \frac{\eta_y\nu}{\eta_y+\nu}\right)}{\sqrt{(\eta_x+\nu)(\eta_y+\nu)}}}, \end{aligned}$$

where $\nu = \frac{\lambda_{in}}{2}$.

D Implementation details of VSGL generation

VSGL generation using filtered importance sampling. Our implementation is based on Tokuyoshi [30], and uses DirectX® 11. After rendering a reflective shadow map, an additional reflective shadow map (which stores VPL positions, squared VPL positions, and average emission directions to calculate VSGL parameters) is generated using a compute shader. Then, these reflective shadow maps are mipmapped using a graphics API (i.e., *GenerateMips* of DirectX). Finally, VSGLs are generated based on filtered importance sampling. The proposed mip level l_i is calculated using Alg. 1.

VSGL generation using k-means. For comparison, this paper uses k-means VPL clustering using 2D image space and 3D world space. The k-means algorithm first samples the cluster center according to the PDF, and then the closest center is computed for each VPL. In our implementation, all the texels are assigned to clusters for high-frequency geometries and textured glossy materials unlike Dong et al. [11]. To accelerate the search of the closest cluster center for each texel, a kd-tree of cluster centers is built using parallel construction of a binary radix tree [17]. For densely distributed cluster centers, this tree-based search is more efficient than using a 2D uniform grid proposed by Prutkin et al. [23]. Once clusters are assigned to all the texels, those texels are sorted by cluster ID. Then, to compute the total value of clustered texels, a thread is dispatched for each cluster similar to Prutkin et al. [23]. Unlike Prutkin et al., we use a GPU radix sort [22] instead of bitonic sort for the high-resolution reflective shadow map and G-buffer. Although k-means clustering can be improved by updating cluster centers in an iterative fashion, we do not update iteratively in this paper.

Algorithm 1 Mip level calculation using the bisection method.

```
l_min ← 0
l_max ← the top mip level of p̄
for k = 1 to the user-specified iteration count do
  l ← (l_min + l_max)/2
  if (4^l/M) p̄(x_i, l) < K/N then
    l_min ← l
  else
    l_max ← l
  end if
end for
l_i ← (l_min + l_max)/2
```

Acknowledgements

The polygon models are courtesy of M. Dabrovic, F. Meinl, A. Grynberg and G. Ward. The author would like to thank the anonymous reviewers for valuable comments and helpful suggestions.

References

[1] S. Agarwal, R. Ramamoorthi, S. Belongie, and H. W. Jensen. Structured importance sampling of environment maps. *ACM Trans. Graph.*, 22(3):605–612, 2003.

[2] T. Barák, J. Bittner, and V. Havran. Temporally coherent adaptive sampling for imperfect shadow maps. *Comput. Graph. Forum*, 32(4):87–96, 2013.

[3] P. Beckmann and A. Spizzichino. *Scattering of Electromagnetic Waves from Rough Surfaces*. MacMillan, 1963.

[4] J. F. Blinn. Models of light reflection for computer synthesized pictures. *SIGGRAPH Comput. Graph.*, 11(2):192–198, 1977.

[5] P. Clarberg, W. Jarosz, T. Akenine-Möller, and H. W. Jensen. Wavelet importance sampling: Efficiently evaluating products of complex functions. *ACM Trans. Graph.*, 24(3):1166–1175, 2005.

[6] C. Crassin, F. Neyret, M. Sainz, S. Green, and E. Eisemann. Interactive indirect illumination using voxel cone tracing. *Comput. Graph. Forum*, 30(7):1921–1930, 2011.

[7] C. Dachsbacher, J. Křivánek, M. Hašan, A. Arbree, B. Walter, and J. Novák. Scalable realistic rendering with many-light methods. *Comput. Graph. Forum*, 33(1):88–104, 2014.

[8] C. Dachsbacher and M. Stamminger. Reflective shadow maps. In *I3D'05*, pages 203–231, 2005.

[9] C. Dachsbacher and M. Stamminger. Splatting indirect illumination. In *I3D'06*, pages 93–100, 2006.

[10] P. Debevec. A median cut algorithm for light probe sampling. In *SIGGRAPH'05 Posters*, 2005.

[11] Z. Dong, T. Grosch, T. Ritschel, J. Kautz, and H.-P. Seidel. Real-time indirect illumination with clustered visibility. In *VMV'09*, pages 187–196, 2009.

[12] M. Hašan, J. Křivánek, B. Walter, and K. Bala. Virtual spherical lights for many-light rendering of glossy scenes. *ACM Trans. Graph.*, 28(5):143:1–143:6, 2009.

[13] P. Hedman, T. Karras, and J. Lehtinen. Sequential monte carlo instant radiosity. In *I3D'16*, pages 121–128, 2016.

[14] K. Iwasaki, Y. Dobashi, and T. Nishita. Interactive bi-scale editing of highly glossy materials. *ACM Trans. Graph.*, 31(6):144:1–144:7, 2012.

[15] K. Iwasaki, K. Mizutani, Y. Dobashi, and T. Nishita. Interactive cloth rendering of microcylinder appearance model under environment lighting. *Comput. Graph. Forum*, 33(2):333–340, 2014.

[16] J. T. Kajiya. The rendering equation. *SIGGRAPH Comput. Graph.*, 20(4):143–150, 1986.

[17] T. Karras. Maximizing parallelism in the construction of BVHs, octrees, and k-d trees. In *HPG'12*, pages 33–37, 2012.

[18] A. Keller. Instant radiosity. In *SIGGRAPH'97*, pages 49–56, 1997.

[19] J. Křivánek and M. Colbert. Real-time shading with filtered importance sampling. *Comput. Graph. Forum*, 27(4):1147–1154, 2008.

[20] J. Křivánek, J. A. Ferwerda, and K. Bala. Effects of global illumination approximations on material appearance. *ACM Trans. Graph.*, 29(4):112:1–112:10, 2010.

[21] C. Luksch, R. F. Tobler, R. Habel, M. Schwärzler, and M. Wimmer. Fast light-map computation with virtual polygon lights. In *I3D'13*, pages 87–94, 2013.

[22] D. G. Merrill and A. S. Grimshaw. Revisiting sorting for GPGPU stream architectures. In *PACT'10*, pages 545–546, 2010.

[23] R. Prutkin, A. S. Kaplanyan, and C. Dachsbacher. Reflective shadow map clustering for real-time global illumination. In *EG'12 Short Papers*, pages 9–12, 2012.

[24] T. Ritschel, C. Dachsbacher, T. Grosch, and J. Kautz. The state of the art in interactive global illumination. *Comput. Graph. Forum*, 31(1):160–188, 2012.

[25] T. Ritschel, E. Eisemann, I. Ha, J. D. Kim, and H.-P. Seidel. Making imperfect shadow maps view-adaptive: High-quality global illumination in large dynamic scenes. *Comput. Graph. Forum*, 30(8):2258–2269, 2011.

[26] T. Ritschel, T. Grosch, M. H. Kim, H.-P. Seidel, C. Dachsbacher, and J. Kautz. Imperfect shadow maps for efficient computation of indirect illumination. *ACM Trans. Graph.*, 27(5):129:1–129:8, 2008.

[27] I. H. Sloan and S. Joe. *Lattice Methods for Multiple Integration*. Clarendon Press, Oxford, 1994.

[28] M. Toksvig. Mipmapping normal maps. *J. Graph. Tools*, 10(3):65–71, 2005.

[29] Y. Tokuyoshi. Fast indirect illumination using two virtual spherical Gaussian lights. In *SIGGRAPH Asia '15 Posters*, pages 12:1–12:1, 2015.

[30] Y. Tokuyoshi. Virtual spherical Gaussian lights for real-time glossy indirect illumination. *Comput. Graph. Forum*, 34(7):89–98, 2015.

[31] T. S. Trowbridge and K. P. Reitz. Average irregularity representation of a rough surface for ray reflection. *J. Opt. Soc. Am*, 65(5):531–536, 1975.

[32] Y.-T. Tsai and Z.-C. Shih. All-frequency precomputed radiance transfer using spherical radial basis functions and clustered tensor approximation. *ACM Trans. Graph.*, 25(3):967–976, 2006.

[33] B. Walter, S. R. Marschner, H. Li, and K. E. Torrance. Microfacet models for refraction through rough surfaces. In *EGSR'07*, pages 195–206, 2007.

[34] J. Wang, P. Ren, M. Gong, J. Snyder, and B. Guo. All-frequency rendering of dynamic, spatially-varying reflectance. *ACM Trans. Graph.*, 28(5):133:1–133:10, 2009.

[35] C. Xu, R. Wang, and H. Bao. Realtime rendering glossy to glossy reflections in screen space. *Comput. Graph. Forum*, 34(7):57–66, 2015.

[36] K. Xu, Y.-P. Cao, L.-Q. Ma, Z. Dong, R. Wang, and S.-M. Hu. A practical algorithm for rendering interreflections with all-frequency BRDFs. *ACM Trans. Graph.*, 33(1):10:1–10:16, 2014.

[37] K. Xu, L.-Q. Ma, B. Ren, R. Wang, and S.-M. Hu. Interactive hair rendering and appearance editing under environment lighting. *ACM Trans. Graph.*, 30(6):173:1–173:10, 2011.

[38] K. Xu, W.-L. Sun, Z. Dong, D.-Y. Zhao, R.-D. Wu, and S.-M. Hu. Anisotropic spherical Gaussians. *ACM Trans. Graph.*, 32(6):209:1–209:11, 2013.

[39] L.-Q. Yan, Y. Zhou, K. Xu, and R. Wang. Accurate translucent material rendering under spherical Gaussian lights. *Comput. Graph. Forum*, 31(7):2267–2276, 2012.

Yusuke Tokuyoshi is a senior researcher at Square Enix. He received his Ph.D in engineering from Shinshu University in 2007. Before joining Square Enix, he engaged in R&D on compiler optimization at Hitachi, Ltd. His interests include global illumination algorithms and real-time rendering.