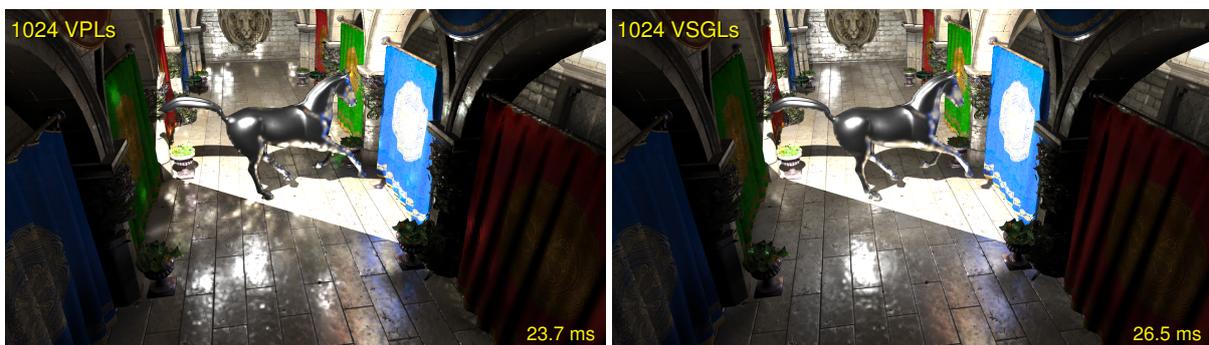


# Virtual Spherical Gaussian Lights for Real-time Glossy Indirect Illumination

Yusuke Tokuyoshi<sup>†</sup>

Square Enix Co., Ltd., Japan



**Figure 1:** One-bounce global illumination for a dynamic scene with normal mapped glossy surfaces (264K triangles, resolution: 1920×1088, GPU: AMD Radeon™ R9 290X). Real-time rendering using virtual point lights (VPLs) [REH\* 11] produces spiky artifacts (left), while our virtual spherical Gaussian lights (VSGLs) do not (right).

## Abstract

Virtual point lights (VPLs) are well established for real-time global illumination. However, this method suffers from spiky artifacts and flickering caused by singularities of VPLs, highly glossy materials, high-frequency textures, and discontinuous geometries. To avoid these artifacts, this paper introduces a virtual spherical Gaussian light (VSGL) which roughly represents a set of VPLs. For a VSGL, the total radiant intensity and positional distribution of VPLs are approximated using spherical Gaussians and a Gaussian distribution, respectively. Since this approximation can be computed using summations of VPL parameters, VSGLs can be dynamically generated using mipmapped reflective shadow maps. Our VSGL generation is simple and independent from any scene geometries. In addition, reflected radiance for a VSGL is calculated using an analytic formula. Hence, we are able to render one-bounce glossy interreflections at real-time frame rates with smaller artifacts.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Color, shading, shadowing, and texture

## 1. Introduction

Indirect illumination effects are perceptually important for interactive applications such as video games, lighting de-

sign and virtual reality systems. Although virtual point light (VPL) [Kel97] based global illumination methods are well established for real-time applications, they have considerable problems such as spiky artifacts and flickering. These are due to the variance of Monte Carlo integration, and caused by singularities of VPLs, highly glossy materials, high-frequency textures, and discontinuous geometries (Fig.

<sup>†</sup> tokuyosh@square-enix.com

1). Virtual spherical lights [HKWB09] addressed this issue for offline rendering, but this method is unsuitable for real-time rendering. This is because virtual spherical lights uses expensive methods such as the  $k$ -nearest neighbor algorithm and numerical radiance evaluation for each virtual light.

On the other hand, *spherical Gaussian lights* are often used in order to approximate all-frequency rendering such as static environmental lighting [TS06]. The illumination due to these lights is evaluated using the product integral of spherical Gaussians which has an analytical solution. Recently, Xu et al. [XCM\*14] proposed an indirect illumination algorithm based on spherical Gaussian lights whose performance ranges from near-interactive to a few seconds per frame. Although their algorithm can render high-quality images, it is unsuitable for time-sensitive applications with high-frequency textures. In addition, their light sources are restricted to distant lights. Real-time global illumination for all-frequency materials is still a challenging problem with high industrial impact.

This paper proposes a real-time frame rate method to dynamically generate *virtual spherical Gaussian lights* (VSGLs) which can represent all-frequency indirect illumination with smaller variance than VPLs. Unlike Xu et al.'s algorithm, this technique supports indirect illumination lit from point or spot lights. In this paper, a set of VPLs are approximated with a VSGL. A VSGL represents the total radiant intensity of VPLs using spherical Gaussians. Furthermore, the distribution of VPL positions is represented with an isotropic Gaussian distribution. Using VSGLs, the incoming radiance at each shading point can be approximated with spherical Gaussians, and thus the reflected radiance can be analytically calculated for a VSGL.

The advantage of our VSGLs is that they are generated using summations of VPL parameters. Therefore, VSGLs can be obtained from mipmapped reflective shadow maps [DS05] inexpensively. In this paper, VSGLs are sampled from the reflective shadow maps using filtered importance sampling [KC08]. Our VSGL generation does not require any expensive data structures. Hence, it is simple, easy to integrate in existing reflective shadow map-based frameworks, and completely dynamic for one-bounce indirect illumination including caustics.

The contributions of our work are as follows:

- A VSGL, which represents a set of VPLs using spherical Gaussians and a Gaussian distribution, is introduced to handle highly glossy materials, high-frequency textures, and discontinuous geometries. A simple spherical Gaussian approximation of incoming radiance from a VSGL is also derived.
- VSGLs are dynamically generated using filtered importance sampling of reflective shadow maps, which is an image-space technique and independent from scene geometries.

- This paper demonstrates dynamic caustics at real-time frame rates.

## 2. Background

### 2.1. Related Work

Interactive global illumination algorithms were surveyed by Ritschel et al. [RDGK12]. For a comprehensive survey of VPL-based rendering, we refer the readers to Dachsbacher et al. [DKH\*14]. Here we pay attention only to the most relevant works.

**Virtual point lights.** For real-time rendering, single-bounce VPLs can be generated by rendering reflective shadow maps. Dachsbacher and Stamminger [DS06] rendered caustics by combining a splatting algorithm and reflective shadow maps. To generate shadow maps for so many VPLs, Ritschel et al. [RGK\*08] proposed imperfect shadow maps using point-based rendering. They also proposed a bidirectional reflective shadow mapping and adaptive imperfect shadow maps to take view-dependent importance into account [REH\*11]. In their algorithm, thousands of VPLs were resampled from reflective shadow maps according to the importance. Dong et al. [DGR\*09] clustered VPLs into tens of area lights using  $k$ -means, and then approximated visibilities of VPLs using a soft shadow map for each cluster. To achieve real-time frame rates, interleaved sampling and denoising filtering have often been used [WKB\*02]. While the VPL method is theoretically unbiased, variance is visible as spiky artifacts especially for glossy materials [KFB10]. In addition, the variance is also visible as temporal flickering. To alleviate flickering, Barák et al. [BBH13] proposed temporally coherent sampling based on the Metropolis-Hastings algorithm. Simon et al. [SHD15] introduced rich-VPLs to avoid spiky artifacts on glossy materials for offline multi-bounce global illumination. To avoid spiky artifacts for interactive or real-time rendering, virtual area light approximation has been developed.

**Virtual area lights.** Virtual spherical lights [HKWB09] were proposed to avoid singularities of VPLs focusing on offline rendering using a numerical approach. Since this numerical approach is unsuitable for interactive rendering, analytical approximations have been developed. Prutkin et al. [PKD12] clustered pixels of a reflective shadow map based on  $k$ -means similar to Dong et al. [DGR\*09], while they approximated the clusters with area lights for analytical radiance evaluation. Luksch et al. [LTH\*13] clustered VPLs using a kd-tree to generate virtual polygon lights to update light maps. These lights were evaluated using analytical form factors which cannot reduce variance caused by high-frequency bidirectional reflectance distribution functions (BRDFs). To take all-frequency materials into account, Xu et al. [XCM\*14] approximated the outgoing radiance using spherical Gaussians for each triangle primitive lit from

distant light sources. They also constructed a tree of these virtual triangle lights for lightcuts [WFA\*05] like radiance evaluation. This algorithm can produce high-quality images in a few seconds. Furthermore, caustics can be rendered unlike eye-path tracing-based methods such as final gathering using voxel cone tracing [CNS\*11]. However, their triangle-based hierarchy depends on scene geometries, and needs dynamic subdivision to handle textures.

Instead of area light approximations, this paper represents the positional distribution of VPLs with an isotropic Gaussian distribution. This representation enables a simpler implementation which is independent from scene geometries and not restricted to distant light sources. Our VSGL generation is performed in reflective shadow map space, and does not require additional expensive data structures. In addition, our approach also has an analytical solution of radiance evaluation unlike virtual spherical lights. Hence, it is more suitable for time-sensitive applications.

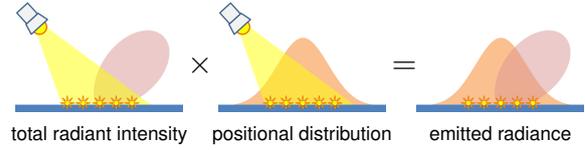
**Spherical Gaussian lighting.** Spherical Gaussians and anisotropic spherical Gaussians [XSD\*13] are often used for approximating the rendering of various types of materials under environment maps or area lights [WRG\*09, XMR\*11, YZXW12, IDN12, IMDN14]. This is because they have closed-form solutions for the integral, product, and product integral, which are fundamental operations to evaluate rendering integrals. In addition, since a normalized spherical Gaussian is equivalent to the von Mises-Fisher distribution, several spherical Gaussians can be merged into a single spherical Gaussian in an analytic way [BDGS05]. Laurijssen et al. [LWDB10] merged specular lobes for variance reduction. Instead of this merging, more inexpensive Toksvig's filtering [Tok05] is also usable for spherical Gaussians. Since microfacet BRDFs can be analytically approximated with spherical Gaussians and anisotropic spherical Gaussians [WRG\*09], approximation of incoming radiance is now the main issue for real-time rendering. For static environment maps, fitting has been used in preprocessing. For indirect illumination, Xu et al. [XCM\*14] proposed the near-interactive frame rate method using the triangle-based hierarchy. This paper proposes a real-time frame rate method for completely dynamic glossy interreflections.

## 2.2. Spherical Gaussians

A spherical Gaussian is a type of spherical function for a direction  $\omega$  and is represented by using a Gaussian function  $g$  as follows:

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

where  $\xi$  is the lobe axis, and  $\lambda$  is the lobe sharpness.  $\xi$  and  $\frac{1}{\lambda}$  correspond to the mean and variance for the Gaussian function, respectively. The integral of a spherical Gaussian



**Figure 2:** Our virtual spherical Gaussian light (VSGL) representation. The total radiant intensity and positional distribution of VPLs are approximated using spherical Gaussians and a Gaussian distribution, respectively. The emitted radiance of the VSGL is given by the product of both.

is given by

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

In this paper, a normalized spherical Gaussian  $\frac{G(\omega, \xi, \lambda)}{A(\lambda)}$  is used for representing the distribution of radiant intensity of a virtual light.

## 3. Virtual Spherical Gaussian Lights

To reduce the variance of radiance estimation, this paper approximates a set of VPLs with a virtual spherical Gaussian light (VSGL). For a VSGL, the total radiant intensity and positional distribution of VPLs are respectively represented using spherical Gaussians and a Gaussian distribution as shown in Fig. 2. This representation is computed using a simple summation operation which can be inexpensively calculated by mipmapping on the GPU.

### 3.1. Virtual light representation

#### 3.1.1. Radiant intensity

Xu et al. [XCM\*14] approximated the directional term of the emitted radiance of virtual light using spherical Gaussians. Unlike Xu et al., this paper approximates the radiant intensity using spherical Gaussians to take the distribution of surface normals into account. The radiant intensity of the  $i$ th VPL for a direction  $\omega$  is given as

$$I_i(\omega) = \Phi_i \rho(\mathbf{x}_i, \omega'_i, \omega) \langle \omega, \mathbf{n}_i \rangle,$$

where  $\Phi_i$  is the power of the  $i$ th photon emitted from the light source,  $\omega'_i$  is the incoming direction of the photon, and  $\mathbf{n}_i$  is the surface normal at the VPL position  $\mathbf{x}_i$ ,  $\rho(\mathbf{x}_i, \omega'_i, \omega)$  is the BRDF, and  $\langle \omega, \mathbf{n}_i \rangle = \max(\omega \cdot \mathbf{n}_i, 0)$ . This paper first divides this BRDF into diffuse and specular components. Then, the total radiant intensity of a set of VPLs is approximated with a single spherical Gaussian for each component by using Toksvig's filtering. Therefore, two spherical Gaussians are used for a VSGL on diffuse-specular surfaces. For ease of explanation, this section hereafter describes only a single BRDF component. The total radiant intensity of a set

of VPLs  $\mathbb{S}$  is represented as

$$I_v(\boldsymbol{\omega}) = \sum_{i \in \mathbb{S}} I_i(\boldsymbol{\omega}) \approx c_v G(\boldsymbol{\omega}, \boldsymbol{\xi}_v, \lambda_v).$$

To compute  $c_v$ ,  $\boldsymbol{\xi}_v$ , and  $\lambda_v$  efficiently, each reflection lobe is approximated using the product of a reflectance and normalized spherical Gaussian (derived in the supplemental material) as follows:

$$\begin{aligned} I_v(\boldsymbol{\omega}) &\approx \sum_{i \in \mathbb{S}} \Phi_i R_i \frac{G(\boldsymbol{\omega}, \boldsymbol{\xi}_i, \lambda_i)}{A(\lambda_i)} \\ &= \left( \sum_{i \in \mathbb{S}} \Phi_i R_i \right) \frac{\sum_{i \in \mathbb{S}} \Phi_i R_i \frac{G(\boldsymbol{\omega}, \boldsymbol{\xi}_i, \lambda_i)}{A(\lambda_i)}}{\sum_{i \in \mathbb{S}} \Phi_i R_i}, \end{aligned}$$

where  $R_i$  is the reflectance, and  $\boldsymbol{\xi}_i$  and  $\lambda_i$  are the axis and sharpness of the reflection lobe at the  $i$ th VPL. Then, the weighted average of normalized spherical Gaussians weighted by  $\Phi_i R_i$  is approximated with a single spherical Gaussian as

$$\frac{\sum_{i \in \mathbb{S}} \Phi_i R_i \frac{G(\boldsymbol{\omega}, \boldsymbol{\xi}_i, \lambda_i)}{A(\lambda_i)}}{\sum_{i \in \mathbb{S}} \Phi_i R_i} \approx \frac{G(\boldsymbol{\omega}, \boldsymbol{\xi}_v, \lambda_v)}{A(\lambda_v)}.$$

Using Toksvig's filtering, the  $i$ th normalized spherical Gaussian is first approximately converted into its averaged direction as  $\bar{\boldsymbol{\xi}}_i = \frac{\lambda_i}{\lambda_i + 1} \boldsymbol{\xi}_i$ . Next, the weighted average of the directions is computed by  $\bar{\boldsymbol{\xi}}_v = \frac{\sum_{i \in \mathbb{S}} \Phi_i R_i \bar{\boldsymbol{\xi}}_i}{\sum_{i \in \mathbb{S}} \Phi_i R_i}$ . 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  $c_v = \frac{\sum_{i \in \mathbb{S}} \Phi_i R_i}{A(\lambda_v)}$ .

### 3.1.2. 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 to avoid the increase of visibility tests (i.e., shadow maps). The weighted mean of VPL positions is computed by

$$\boldsymbol{\mu}_v = \frac{\sum_{i \in \mathbb{S}} \Phi_i (R_{d,i} + R_{s,i}) \mathbf{x}_i}{\sum_{i \in \mathbb{S}} \Phi_i (R_{d,i} + R_{s,i})},$$

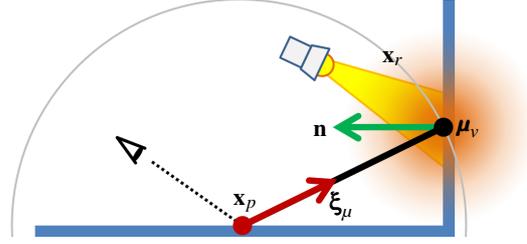
where  $R_{d,i}$  and  $R_{s,i}$  are the diffuse reflectance and specular reflectance at the  $i$ th VPL, respectively. The positional variance is also calculated using weighted average as

$$\sigma_v^2 = \frac{\sum_{i \in \mathbb{S}} \Phi_i (R_{d,i} + R_{s,i}) \|\mathbf{x}_i\|^2}{\sum_{i \in \mathbb{S}} \Phi_i (R_{d,i} + R_{s,i})} - \|\boldsymbol{\mu}_v\|^2.$$

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

$$L_e(\mathbf{x}, \boldsymbol{\omega}) \approx \frac{I_v(\boldsymbol{\omega})}{2\pi\sigma_v^2 |\boldsymbol{\omega} \cdot \mathbf{n}|} g(\|\mathbf{x} - \boldsymbol{\mu}_v\|, \sigma_v^2), \quad (1)$$

where  $\mathbf{n}$  is the surface normal which will be eliminated in shading (§3.2). Therefore, a VSGL is represented with the



**Figure 3:** Since the VPL positions are represented using an isotropic Gaussian distribution, the spherical region of the VSGL viewed from the shading point  $\mathbf{x}_p$  is approximated with a spherical Gaussian.

coefficient  $\frac{c_v}{2\pi\sigma_v^2}$ , lobe axis  $\boldsymbol{\xi}_v$ , lobe sharpness  $\lambda_v$ , positional mean  $\boldsymbol{\mu}_v$  and positional variance  $\sigma_v^2$ . Since these VSGL parameters are computed using a summation operation, they can be inexpensively computed by mipmapping on the GPU.

### 3.2. Shading

For each shading point  $\mathbf{x}_p$  with view direction  $\boldsymbol{\omega}_p$ , the reflected radiance is calculated using the rendering equation [Kaj86] defined by

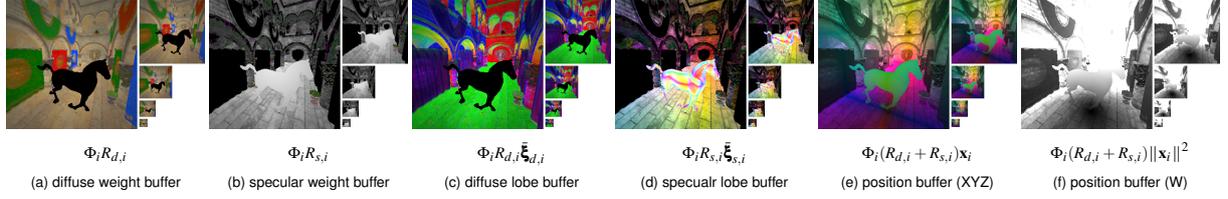
$$L(\mathbf{x}_p, \boldsymbol{\omega}_p) = \int_{\mathbb{S}^2} L_{in}(\mathbf{x}_p, \boldsymbol{\omega}) \rho(\mathbf{x}_p, \boldsymbol{\omega}_p, \boldsymbol{\omega}) \langle \boldsymbol{\omega}, \mathbf{n}_p \rangle d\boldsymbol{\omega}, \quad (2)$$

where  $L_{in}(\mathbf{x}_p, \boldsymbol{\omega})$  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 [WRG\*09, XSD\*13]. Using Eq. 1, the approximated incoming radiance is given by

$$\begin{aligned} L_{in}(\mathbf{x}_p, \boldsymbol{\omega}) &= V(\mathbf{x}_p, \mathbf{x}) L_e(\mathbf{x}, -\boldsymbol{\omega}) \\ &\approx \frac{V(\mathbf{x}_p, \boldsymbol{\mu}_v) I_v(-\boldsymbol{\omega})}{2\pi\sigma_v^2 |\boldsymbol{\omega} \cdot \mathbf{n}|} g(\|\mathbf{x} - \boldsymbol{\mu}_v\|, \sigma_v^2), \quad (3) \end{aligned}$$

where  $\boldsymbol{\omega} = \frac{\mathbf{x} - \mathbf{x}_p}{\|\mathbf{x} - \mathbf{x}_p\|}$ , and  $V(\mathbf{x}_p, \boldsymbol{\mu}_v)$  is the visibility between  $\mathbf{x}_p$  and  $\boldsymbol{\mu}_v$  obtained from a shadow map.  $\mathbf{x}$  is assumed to be on the planar surface defined by the normal  $\mathbf{n}$  and position  $\boldsymbol{\mu}_v$ .

Here this paper introduces virtual spherical light-like approximation [HKWB09]. Virtual spherical lights do not take the surface orientation into account for the shape viewed from a shading point. Instead,  $|\boldsymbol{\omega} \cdot \mathbf{n}|$  is multiplied to correct the energy. Similar to this, this paper approximates the surface orientation by multiplying  $|\boldsymbol{\omega} \cdot \mathbf{n}|$ . For our case, it is divided by  $|\boldsymbol{\omega} \cdot \mathbf{n}|$ , and thus  $\mathbf{n}$  is eliminated. This is reasonable because the actual surface normal distribution is taken into account by the radiant intensity  $I_v(-\boldsymbol{\omega})$ . Our representation of the positional distribution is an isotropic Gaussian distribution which is directionally independent similar to a spherical light. The shape of a virtual spherical light viewed



**Figure 4:** Our mipmapped reflective shadow map.  $\bar{\xi}_{d,i}$  and  $\bar{\xi}_{s,i}$  are average directions of diffuse and specular reflection lobes, respectively. The diffuse weight buffer (a) and specular weight buffer (b) have RGB channels. In our implementation, the total of these RGB channels is used as the weight of the diffuse lobe buffer (c), specular lobe buffer (d), and position buffer (e)(f) for simplicity. The weighted position (e) and weighted squared norm of the position (f) are packed into a single position buffer.

from a shading point is represented by projecting the spherical light onto the sphere centered at the shading point. Therefore, the viewed shape of a VSGL is also approximated by projecting onto this sphere. Hence, Eq. 3 is approximated as follows:

$$L_{in}(\mathbf{x}_p, \boldsymbol{\omega}) \approx \frac{V(\mathbf{x}_p, \boldsymbol{\mu}_v) I_v(-\boldsymbol{\omega})}{2\pi\sigma_v^2} g(\|\mathbf{x}_r - \boldsymbol{\mu}_v\|, \sigma_v^2),$$

where  $\boldsymbol{\omega} = \frac{\mathbf{x}_r - \mathbf{x}_p}{\|\mathbf{x}_r - \mathbf{x}_p\|}$ , and  $\mathbf{x}_r$  is the position on the sphere defined by the center  $\mathbf{x}_p$  and radius  $\|\boldsymbol{\mu}_v - \mathbf{x}_p\|$  as shown in Fig. 3. This is derived assuming a small  $\sigma_v$  or large radius, but it does not produce noticeable artifacts in practice for a large  $\sigma_v$  and small radius. Furthermore, the Gaussian term can be rewritten into a spherical Gaussian as

$$g(\|\mathbf{x}_r - \boldsymbol{\mu}_v\|, \sigma_v^2) = G(\boldsymbol{\omega}, \boldsymbol{\xi}_\mu, \lambda_\sigma), \quad (4)$$

where  $\boldsymbol{\xi}_\mu = \frac{\boldsymbol{\mu}_v - \mathbf{x}_p}{\|\boldsymbol{\mu}_v - \mathbf{x}_p\|}$ , and  $\lambda_\sigma = \frac{\|\boldsymbol{\mu}_v - \mathbf{x}_p\|^2}{\sigma_v^2}$ . This spherical Gaussian represents the spherical region of the VSGL viewed from  $\mathbf{x}_p$ . Using Eq. 4, the incoming radiance is approximated with the product of two spherical Gaussians which yields a spherical Gaussian as follows:

$$\begin{aligned} L_{in}(\mathbf{x}_p, \boldsymbol{\omega}) &\approx \frac{V(\mathbf{x}_p, \boldsymbol{\mu}_v) c_v}{2\pi\sigma_v^2} G(\boldsymbol{\omega}, -\boldsymbol{\xi}_v, \lambda_v) G(\boldsymbol{\omega}, \boldsymbol{\xi}_\mu, \lambda_\sigma) \\ &= \boxed{c_{in} G(\boldsymbol{\omega}, \boldsymbol{\xi}_{in}, \lambda_{in})}, \end{aligned} \quad (5)$$

where  $\boldsymbol{\xi}_{in} = \frac{\lambda_\sigma \boldsymbol{\xi}_\mu - \lambda_v \boldsymbol{\xi}_v}{\|\lambda_\sigma \boldsymbol{\xi}_\mu - \lambda_v \boldsymbol{\xi}_v\|}$ ,  $\lambda_{in} = \|\lambda_\sigma \boldsymbol{\xi}_\mu - \lambda_v \boldsymbol{\xi}_v\|$ , and  $c_{in} = \frac{V(\mathbf{x}_p, \boldsymbol{\mu}_v) c_v}{2\pi\sigma_v^2} e^{\lambda_{in} - \lambda_v - \lambda_\sigma}$ . Since the reflection lobe  $\rho(\mathbf{x}_p, \boldsymbol{\omega}_p, \boldsymbol{\omega})(\boldsymbol{\omega}, \mathbf{n}_p)$  can be approximated using anisotropic spherical Gaussians, Eq. 2 can be calculated using the analytical product integral of the spherical Gaussian and anisotropic spherical Gaussians (please refer to the supplemental material).

**Comparison with area light approximation.** For the spherical region of a virtual area light viewed from a shading point, Xu et al. [XCM\*14] introduced the following spherical Gaussian approximation:

$$2G\left(\boldsymbol{\omega}, \boldsymbol{\xi}_a, \frac{4\pi}{\|\Omega_a\|}\right), \quad (6)$$

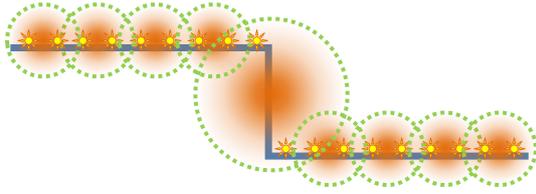
where  $\boldsymbol{\xi}_a = \frac{\mathbf{x}_a - \mathbf{x}_p}{\|\mathbf{x}_a - \mathbf{x}_p\|}$ , and  $\mathbf{x}_a$  is the center of the virtual light.  $\|\Omega_a\|$  is the solid angle computed as  $\|\Omega_a\| \approx \frac{\Delta_a \langle \boldsymbol{\xi}_a, \mathbf{n}_a \rangle}{\|\mathbf{x}_a - \mathbf{x}_p\|^2}$ , where  $\Delta_a$  and  $\mathbf{n}_a$  are the area and average normal of the virtual light, respectively. Since they approximated the directional term of emitted radiance using spherical Gaussians, the surface orientation had to be taken into account by the spherical region. On the other hand, our method eliminates the surface normal for the spherical region because the distribution of surface normals is taken into account by the radiant intensity  $I_v(-\boldsymbol{\omega})$ . This approach avoids flipping of the average normal on complex objects which is noticeable when using a few VSGLs. Eq. 4 has no coefficient, and thus it is lower-frequency and simpler than Eq. 6, while both approximately preserve energy. In addition, this paper does not need area  $\Delta_a$ , which is calculated using a bounding sphere [HKWB09]; 2D convex hull [LTH\*13]; or total of triangle areas [XCM\*14]. Instead of the area, we use variance  $\sigma_v^2$  obtained by mipmapping.

#### 4. Generating Virtual Spherical Gaussian Lights

Since our virtual spherical Gaussian lights (VSGLs) are calculated using summations of VPL parameters (i.e., pixels of reflective shadow maps), they can be generated by mipmapping of reflective shadow maps for completely dynamic scenes.

##### 4.1. Our mipmapped reflective shadow maps

Our reflective shadow map has five buffers shown in Fig. 4. Once reflective shadow maps are rendered, their mipmaps are generated. Via mipmapping, VPLs are implicitly clustered with a quadtree in reflective shadow map space. Thus, each total value of a cluster  $\mathbb{S}$  can be obtained using a pixel position  $\mathbf{s}$  and mip level  $l$ . For example, let  $B(\mathbf{s}, l)$  be the value of diffuse weight buffer, then the total diffuse weight is given by  $\sum_{i \in \mathbb{S}} \Phi_i R_{d,i} = 4^l B(\mathbf{s}, l)$ . While this mipmap-based clustering ignores difference of depths, it does not produce undesirable high-frequency artifacts. This is because lower-frequency VSGLs are generated on such discontinuous geometries as shown in Fig. 5.



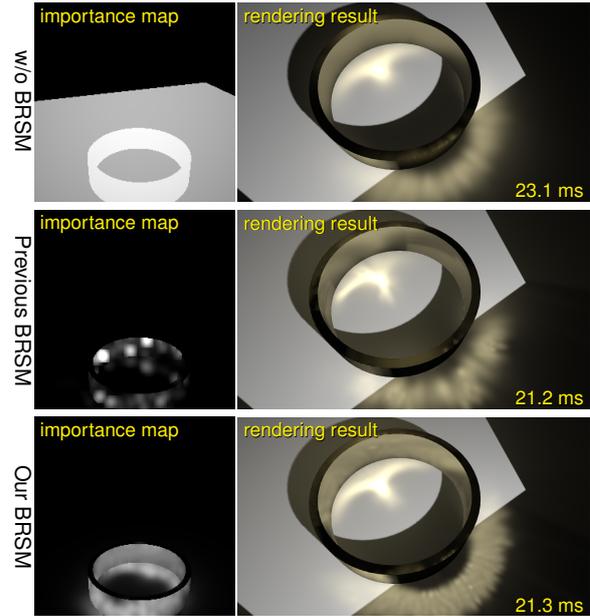
**Figure 5:** Since VPLs are sparsely distributed on depth discontinuous regions, lower-frequency virtual spherical Gaussian lights (VSGLs) are generated on such regions. These VSGLs produce a smoother illumination appearance, and alleviate temporal flickering.

#### 4.2. Filtered importance sampling

For VPLs, Ritschel et al. [REH\*11] sampled pixel positions of reflective shadow map according to a probability density function represented by an importance map. Instead of this standard importance sampling, this paper employs filtered importance sampling [KC08] to select  $\mathbf{s}$  and  $l$  for VSGLs. Filtered importance sampling smoothes samples on unimportant regions via mipmapping, and it is often used for variance reduction of environmental lighting. We employ it for the mipmapped reflective shadow maps to approximate indirect illumination. The pixel position  $\mathbf{s}$  is sampled using standard importance sampling performed by hierarchical sample warping [CJAMJ05]. The mip level  $l$  is determined based on the sample density as  $l = l_{max} - \frac{1}{2} \log_2 Np(\mathbf{s})$ , where  $l_{max}$  is the top mip level of the reflective shadow map,  $N$  is the number of VSGLs, and  $p(\mathbf{s})$  is the probability density function. This paper uses  $p(\mathbf{s}) \propto \Phi_i(R_{d,i} + R_{s,i})$  as default. While it is simple, inexpensive, and almost temporally coherent, it can blur some high-frequency illumination such as caustics.

#### 4.3. Importance estimation using virtual spherical Gaussian lights

To render high-frequency illumination such as caustics, this paper proposes an improved bidirectional reflective shadow map technique as an option. The probability density on the reflective shadow map (i.e., importance of the VPL) is estimated by computing indirect illumination exchanging reflective shadow maps and a G-buffer. However, this importance estimation can have high variance, especially for glossy interreflections. Therefore, to reduce variance, our VSGLs are also used for the importance estimation as shown in Fig. 6. For this bidirectional reflective shadow mapping, a diffuse lobe buffer, specular lobe buffer, and position buffer are added into the G-buffer. Unlike reflective shadow maps, only reflectances are used as weights. The G-buffer is mipmapped similarly to reflective shadow maps, and then VSGLs are generated on the view side to estimate the probability density on the reflective shadow map. One limitation of bidirectional reflective shadow map techniques is that sample points are often temporally incoherent, even if the importance es-

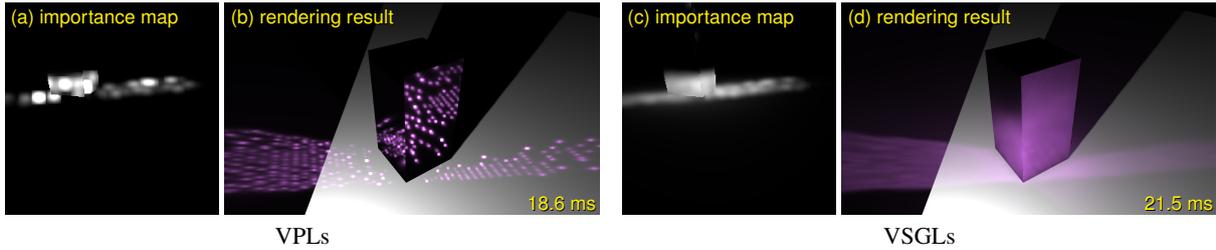


**Figure 6:** Caustics using 1024 virtual spherical Gaussian lights (514 triangles scene). Rendering without the bidirectional reflective shadow mapping (BRSM) (top) suppresses flickering, while it also blurs illumination appearance. The previous BRSM (middle) cannot estimate an accurate importance for specular surfaces, and thus produces large errors. Our BRSM (bottom) renders more accurate results than the previous method.

timation is converged. Although our VSGLs alleviate this limitation using mipmapping filtering, it cannot be avoided completely. Therefore, this paper recommends the bidirectional reflective shadow mapping for applications which give priority to detailed illumination over temporal coherence, such as lighting design.

## 5. Experimental Results

We show rendering results using 1024 virtual spherical Gaussian lights (VSGLs) performed on an AMD Radeon R9 290X. The frame buffer and reflective shadow map resolutions are  $1920 \times 1088$  and  $256^2$ , respectively. To evaluate the visibility of each VSGL, an adaptive imperfect shadow map of resolution  $64^2$  is employed. For each adaptive imperfect shadow map, 8192 shadow caster points are used, and 1/16 of them are updated for each frame in a round-robin fashion.  $8 \times 8$  interleaved sampling and  $17 \times 17$  denoising filtering are used to alleviate the shading cost. For bidirectional reflective shadow mapping, 4096 VSGLs without shadow maps are generated on the G-buffer. For comparison, the same number of samples as VSGLs is used for VPLs. The GGX normal distribution function [WMLT07] is used for glossy BRDFs. In this paper, the diffuse lobe buffer, specular lobe



**Figure 7:** Caustics using bidirectional reflective shadow mapping (BRSM) (14 triangles scene, roughness: 0.004). Point sampling-based BRSM produces large errors (a), and thus generates VPLs with high variance. In addition, VPLs also produces intense spiky artifacts for the resulting image (b). Using virtual spherical Gaussian lights (VSGLs), these artifacts are reduced significantly (c)(d).



**Figure 8:** Textured dynamic scene with normal maps (264K triangles, minimum roughness: 0.04). Virtual spherical Gaussian lights (VSGLs) without bidirectional reflective shadow mapping (BRSM) (c) suppress spiky artifacts and flickering, while it has the side-effect of possibly overblurring some illumination details. Such illumination details can be rendered using BRSM (d), though flickering is remained slightly.

buffer and position buffer of a reflective shadow map and G-buffer are generated using a compute shader after rendering the standard reflective shadow map and G-buffer (e.g., depth, normal, roughness, diffuse weight and specular weight) for acceleration.

**Caustics.** Fig. 7 shows caustics rendered using VPLs and VSGLs with bidirectional reflective shadow mapping at real-time frame rates. The previous bidirectional reflective shadow mapping (a) is unsuitable for such scenes since the importance map can be estimated with large variance which induces additional errors for resulting images. In addition,

the VPL evaluation also has large variance. Hence, intense spiky artifacts and flickering are produced (b). On the other hand, our method renders such caustics with less variance (c)(d).

**Dynamic scenes.** Fig. 8 shows a textured dynamic scene with complex geometries and normal maps. For this scene, 1024 VPLs are insufficient to accurately represent indirect illumination (b). On the other hand, our method significantly reduces spiky artifacts and flickering (c)(d). For such insufficient samples, although our approach can produce lower-frequency illumination appearance than is actu-

**Table 1:** Averaged computation times for indirect illumination (ms).

	14 triangles			331K triangles			264K triangles		
	VPLs	VSGLs	VSGLs	VPLs	VSGLs	VSGLs	VPLs	VSGLs	VSGLs
	(w/o BRSM)		(with BRSM)	(w/o BRSM)		(with BRSM)	(w/o BRSM)		(with BRSM)
G-buffer	0.15	0.15	0.15	0.42	0.42	0.42	0.54	0.54	0.54
Reflective shadow map	0.29	0.29	0.29	0.35	0.35	0.35	0.30	0.30	0.30
<b>Additional G-buffer</b>	-	-	<b>1.49</b>	-	-	<b>1.48</b>	-	-	<b>1.49</b>
<b>Additional reflective shadow map</b>	-	<b>0.19</b>	<b>0.19</b>	-	<b>0.19</b>	<b>0.19</b>	-	<b>0.19</b>	<b>0.19</b>
<b>Importance map</b>	<b>0.70</b>	<b>0.04</b>	<b>0.79</b>	<b>0.69</b>	<b>0.04</b>	<b>0.78</b>	<b>0.70</b>	<b>0.04</b>	<b>0.79</b>
<b>VPL/VSGL sampling</b>	<b>0.02</b>	<b>0.02</b>	<b>0.02</b>	<b>0.02</b>	<b>0.02</b>	<b>0.02</b>	<b>0.02</b>	<b>0.02</b>	<b>0.02</b>
Adaptive imperfect shadow maps	11.19	10.58	10.87	15.15	15.41	15.71	14.30	16.21	13.93
<b>Radiance evaluation</b>	<b>3.44</b>	<b>4.50</b>	<b>4.60</b>	<b>3.25</b>	<b>4.44</b>	<b>4.44</b>	<b>3.60</b>	<b>4.46</b>	<b>4.65</b>
Denosing filter	2.68	2.68	2.68	2.68	2.68	2.68	2.68	2.68	2.68

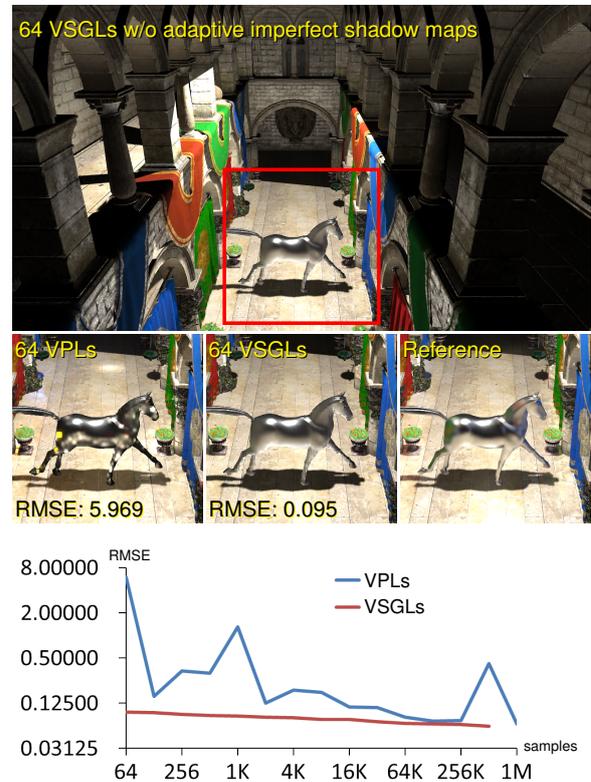
ally the case, it is visually acceptable compared to the VPL-based method which produces high-frequency artifacts. Fig. 8(d) is rendered using bidirectional reflective shadow mapping. Although the bidirectional reflective shadow mapping can produce more detailed illumination, slight flickering still can be produced compared to without bidirectional reflective shadow mapping (c) (please refer to the supplemental video).

**Performance.** Table 1 shows the computation time of each procedure. The performance of our contributions (written in bold) is independent from the triangle count and number of dynamic objects, and is not the main bottleneck. The difference of adaptive imperfect shadow maps is due to sampled mean positions  $\mu_v$ . The overhead of our method is due to the additional buffer generation, mipmapping, and radiance evaluation using spherical Gaussians. Compared to VPLs, the total overhead of virtual spherical Gaussian lights (VSGLs) with bidirectional reflective shadow mapping (BRSM) is 2-4 ms.

**Accuracy.** Fig. 9 shows the quality comparison using the root mean squared error metric. For this experiment, a resolution of  $4096^2$  reflective shadow map is used. In addition, adaptive imperfect shadow maps are omitted to remove errors of adaptive imperfect shadow maps (which are not our contribution). VSGLs reduce errors, especially for a smaller number of samples. Although our method is an inconsistent estimator, it has lower errors even for hundreds of thousands of samples which perform at a few seconds per frame.

## 6. Limitations and Future Work

**Blurring.** Unlike VPLs, the proposed method has bias which can blur illumination appearance. However, this blurring is more visually acceptable than spiky artifacts and



**Figure 9:** Quality comparison. Virtual spherical Gaussian lights (VSGLs) produce significantly lower root mean squared errors (RMSEs) than VPLs for smaller number of samples. Even for hundreds of thousands of samples, our method still has lower errors.

flickering, especially for a smaller number of virtual spherical Gaussian lights (VSGLs). Although our method is also an inconsistent estimator, undesirable errors are reduced at real-time frame rates. For accurate rendering, the bias can be removed by canceling spherical Gaussian approximations while increasing the number of VSGLs. We would like to investigate the effectiveness of this approach in the future.

**Multi-lobe BRDFs.** Although a larger number of spherical Gaussians can be used for multi-lobe BRDFs, the complexity of the product integral of two spherical Gaussian-mixtures is  $O(K^2)$ , where  $K$  is the size of the spherical Gaussian-mixture. Therefore, this paper restricts  $K = 2$  for diffuse-specular BRDFs. For multi-lobe BRDFs, the number of lobes can be reduced on-the-fly using inexpensive Toksvig's filtering.

**Light leaks.** Due to the rough representation of positional distribution, our method can produce light leaks. These leaks are often noticeable when the true distribution is multimodal for a cluster  $S$ . This problem can be alleviated by dividing the positional distribution into diffuse and specular components. However, this approach requires twice the number of shadow maps.

**Shadow maps.** Our implementation uses adaptive imperfect shadow maps to evaluate visibilities of VSGLs. Imperfect shadow map-based methods also have high bias due to their imperfectness. In addition, when a scene has dynamic objects, adaptive imperfect shadow maps can produce wavering shadows even for static objects. This can be avoided by using non-adaptive imperfect shadow maps with more shadow caster points which can sacrifice performance. This issue is problematic not only for our method, but also most real-time many-lights algorithms.

**Temporal coherence of bidirectional reflective shadow mapping.** Although the importance estimation using bidirectional reflective shadow mapping is sufficiently converged, flickering can still be produced on the resulting image. This paper inherits this limitation from bidirectional reflective shadow mapping. However, this flickering can be avoided by temporally coherent Metropolis-Hastings sampling [BBH13]. For future work, we would like to investigate such sampling methods to moreover reduce flickering.

## 7. Conclusions

This paper has presented a simple approximation of a set of VPLs called virtual spherical Gaussian light (VSGL) for real-time applications. Since our VSGL representation is calculated using a summation operation, they are simply generated by mipmapping of reflective shadow maps. In addition, VSGL-based radiance estimation is also usable for bidirectional reflective shadow mapping. Hence, we are able

to render one-bounce glossy interreflections including caustics with a few milliseconds overhead as compared to VPLs. When the number of VPLs is insufficient to approximate indirect illumination, the proposed method exchanges undesirable variance for visually acceptable bias in a simplistic way. Although our method suppresses artifacts significantly, flickering caused by bidirectional reflective shadow mapping cannot be avoided completely. For future work, we would like to investigate the effectiveness of combining with temporally coherent sampling for bidirectional reflective shadow mapping. In addition, since our VSGL generation performs in an image space, it is also potentially applicable for dynamic environment maps. We would like to implement an on-the-fly spherical Gaussian lights generation for environmental lighting in the future.

## Acknowledgements

The polygon models are courtesy of M. Dabrovic, F. Meinel, R. W. Sumner and J. Popovic, and A. Grynberg and G. Ward. The authors would like to thank the anonymous reviewers for valuable comments and helpful suggestions.

## References

- [BBH13] BARÁK T., BITTNER J., HAVRAN V.: Temporally coherent adaptive sampling for imperfect shadow maps. *Comput. Graph. Forum* 32, 4 (2013), 87–96. 2, 9
- [BDGS05] BANERJEE A., DHILLON I. S., GHOSH J., SRA S.: Clustering on the unit hypersphere using von Mises-Fisher distributions. *J. Mach. Learn. Res.* 6 (2005), 1345–1382. 3
- [CJAMJ05] CLARBERG P., JAROSZ W., AKENINE-MÖLLER T., JENSEN H. W.: Wavelet importance sampling: Efficiently evaluating products of complex functions. *ACM Trans. Graph.* 24, 3 (2005), 1166–1175. 6
- [CNS\*11] CRASSIN C., NEYRET F., SAINZ M., GREEN S., EISEMANN E.: Interactive indirect illumination using voxel cone tracing. *Comput. Graph. Forum* 30, 7 (2011), 1921–1930. 3
- [DGR\*09] DONG Z., GROSCH T., RITSCHER T., KAUTZ J., SEIDEL H.-P.: Real-time indirect illumination with clustered visibility. In *Vision, Modeling, and Visualization Workshop* (2009), pp. 187–196. 2
- [DKH\*14] DACHSBACHER C., KRÍVÁNEK J., HAŠAN M., ARBREE A., WALTER B., NOVÁK J.: Scalable realistic rendering with many-light methods. *Comput. Graph. Forum* 33, 1 (2014), 88–104. 2
- [DS05] DACHSBACHER C., STAMMINGER M.: Reflective shadow maps. In *I3D '05* (2005), pp. 203–231. 2
- [DS06] DACHSBACHER C., STAMMINGER M.: Splatting indirect illumination. In *I3D '06* (2006), pp. 93–100. 2
- [HKWB09] HAŠAN M., KRÍVÁNEK J., WALTER B., BALÁ K.: Virtual spherical lights for many-light rendering of glossy scenes. *ACM Trans. Graph.* 28, 5 (2009), 143:1–143:6. 2, 4, 5
- [IDN12] IWASAKI K., DOBASHI Y., NISHITA T.: Interactive bi-scale editing of highly glossy materials. *ACM Trans. Graph.* 31, 6 (2012), 144:1–144:7. 3
- [IMDN14] IWASAKI K., MIZUTANI K., DOBASHI Y., NISHITA T.: Interactive cloth rendering of microcylinder appearance model under environment lighting. *Comput. Graph. Forum* 33, 2 (2014), 333–340. 3

- [Kaj86] KAJIYA J. T.: The rendering equation. *SIGGRAPH Comput. Graph.* 20, 4 (1986), 143–150. 4
- [KC08] KRIVÁNEK J., COLBERT M.: Real-time shading with filtered importance sampling. *Comput. Graph. Forum* 27, 4 (2008), 1147–1154. 2, 6
- [Kel97] KELLER A.: Instant radiosity. In *SIGGRAPH'97* (1997), pp. 49–56. 1
- [KFB10] KRIVÁNEK J., FERWERDA J. A., BALA K.: Effects of global illumination approximations on material appearance. *ACM Trans. Graph.* 29, 4 (2010), 112:1–112:10. 2
- [LTH\*13] LUKSCH C., TOBLER R. F., HABEL R., SCHWÄRZLER M., WIMMER M.: Fast light-map computation with virtual polygon lights. In *ISD '13* (2013), pp. 87–94. 2, 5
- [LWDB10] LAURIJSSSEN J., WANG R., DUTRÉ P., BROWN B. J.: Fast estimation and rendering of indirect highlights. *Comput. Graph. Forum* 29, 4 (2010), 1305–1313. 3
- [PKD12] PRUTKIN R., KAPLANYAN A. S., DACHSBACHER C.: Reflective shadow map clustering for real-time global illumination. In *Eurographics 2012 Short Papers* (2012), pp. 9–12. 2
- [RDGK12] RITSCHER T., DACHSBACHER C., GROSCH T., KAUTZ J.: The state of the art in interactive global illumination. *Comput. Graph. Forum* 31, 1 (2012), 160–188. 2
- [REH\*11] RITSCHER T., EISEMANN E., HA I., KIM J. D., SEIDEL H.-P.: Making imperfect shadow maps view-adaptive: High-quality global illumination in large dynamic scenes. *Comput. Graph. Forum* 30, 8 (2011), 2258–2269. 1, 2, 6
- [RGK\*08] RITSCHER T., GROSCH T., KIM M. H., SEIDEL H.-P., DACHSBACHER C., KAUTZ J.: Imperfect shadow maps for efficient computation of indirect illumination. *ACM Trans. Graph.* 27, 5 (2008), 129:1–129:8. 2
- [SHD15] SIMON F., HANIKA J., DACHSBACHER C.: Rich-VPLs for improving the versatility of many-light methods. *Comput. Graph. Forum* 34, 2 (2015), 575–584. 2
- [Tok05] TOKSVIG M.: Mipmapping normal maps. *J. Graph. Tools* 10, 3 (2005), 65–71. 3
- [TS06] TSAI Y.-T., SHIH Z.-C.: All-frequency precomputed radiance transfer using spherical radial basis functions and clustered tensor approximation. *ACM Trans. Graph.* 25, 3 (2006), 967–976. 2
- [WFA\*05] WALTER B., FERNANDEZ S., ARBREE A., BALA K., DONIKIAN M., GREENBERG D. P.: Lightcuts: A scalable approach to illumination. *ACM Trans. Graph.* 24, 3 (2005), 1098–1107. 3
- [WKB\*02] WALD I., KOLLIG T., BENTHIN C., KELLER A., SLUSALLEK P.: Interactive global illumination using fast ray tracing. In *EGWR'02* (2002), pp. 15–24. 2
- [WMLT07] WALTER B., MARSCHNER S. R., LI H., TORRANCE K. E.: Microfacet models for refraction through rough surfaces. In *EGSR '07* (2007), pp. 195–206. 6
- [WRG\*09] WANG J., REN P., GONG M., SNYDER J., GUO B.: All-frequency rendering of dynamic, spatially-varying reflectance. *ACM Trans. Graph.* 28, 5 (2009), 133:1–133:10. 3, 4
- [XCM\*14] XU K., CAO Y.-P., MA L.-Q., DONG Z., WANG R., HU S.-M.: A practical algorithm for rendering interreflections with all-frequency BRDFs. *ACM Trans. Graph.* 33, 1 (2014), 10:1–10:16. 2, 3, 5
- [XMR\*11] XU K., MA L.-Q., REN B., WANG R., HU S.-M.: Interactive hair rendering and appearance editing under environment lighting. *ACM Trans. Graph.* 30, 6 (2011), 173:1–173:10. 3
- [XSD\*13] XU K., SUN W.-L., DONG Z., ZHAO D.-Y., WU R.-D., HU S.-M.: Anisotropic spherical Gaussians. *ACM Trans. Graph.* 32, 6 (2013), 209:1–209:11. 3, 4
- [YZXW12] YAN L.-Q., ZHOU Y., XU K., WANG R.: Accurate translucent material rendering under spherical Gaussian lights. *Comput. Graph. Forum* 31, 7 (2012), 2267–2276. 3