Eurographics Symposium on Rendering 2017
P. Sander and M. Zwicker
(Guest Editors)

Volume 36 (2017), Number 4

Stochastic Light Culling for VPLs on GGX Microsurfaces

Yusuke Tokuyoshi[1] and Takahiro Harada[2]

[1]Square Enix Co., Ltd.
[2]Advanced Micro Devices, Inc.

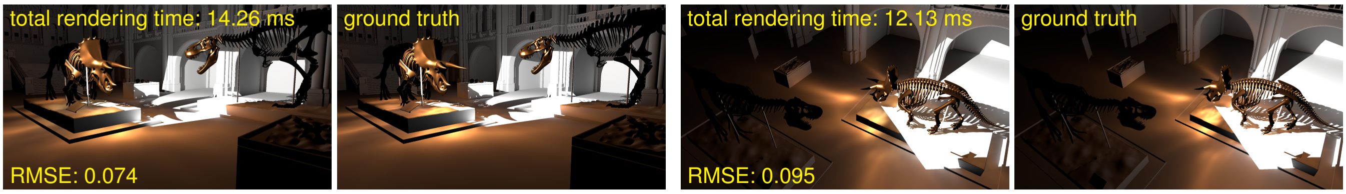

total rendering time: 14.26 ms
RMSE: 0.074
ground truth
total rendering time: 12.13 ms
RMSE: 0.095
ground truth

Figure 1: *Dynamic single-bounce diffuse-to-diffuse and glossy-to-diffuse indirect illumination using 65536 virtual point lights (VPLs) performed on an NVIDIA® GeForce® GTX 970 GPU (1.47M triangle scene, 1920×1080 screen resolution). Glossy caustics created by GGX microsurfaces are rendered efficiently with our method culling VPLs using random light ranges for GGX reflection.*

Abstract

This paper introduces a real-time rendering method for single-bounce glossy caustics created by GGX microsurfaces. Our method is based on stochastic light culling of virtual point lights (VPLs), which is an unbiased culling method that randomly determines the range of influence of light for each VPL. While the original stochastic light culling method uses a bounding sphere defined by that light range for coarse culling (e.g., tiled culling), we have further extended the method by calculating a tighter bounding ellipsoid for glossy VPLs. Such bounding ellipsoids can be calculated analytically under the classic Phong reflection model which cannot be applied to physically plausible materials used in modern computer graphics productions. In order to use stochastic light culling for such modern materials, this paper derives a simple analytical solution to generate a tighter bounding ellipsoid for VPLs on GGX microsurfaces. This paper also presents an efficient implementation for culling bounding ellipsoids in the context of tiled culling. When stochastic light culling is combined with interleaved sampling for a scene with tens of thousands of VPLs, this tiled culling is faster than conservative rasterization-based clustered shading which is a state-of-the-art culling technique that supports bounding ellipsoids. Using these techniques, VPLs are culled efficiently for completely dynamic single-bounce glossy caustics reflected from GGX microsurfaces.

CCS Concepts
•***Computing methodologies*** → ***Rendering;***

1. Introduction

While there are various real-time global illumination methods, rendering completely dynamic caustics at real-time frame rates is still a challenging problem. This paper proposes an unbiased light culling method for glossy caustics lit by *virtual point lights* (VPLs) [Kel97]. VPLs are often used in real-time applications including video games [Lef12, Xu16] for dynamic single-bounce indirect illumination. To render many light sources such as VPLs, light culling techniques (e.g., splatting [DS06], tiled culling [OA11, Har12, Ste15] and clustered shading [OBA12, OP16, OC17]) have been developed. However, these culling techniques produced noticeable darkening bias because of the need to restrict the range of influence of light. To avoid this limitation, an unbiased culling method referred to as *stochastic light culling* [TH16] was proposed recently. This method randomly determines the influence range of each light source based on *Russian roulette* [AK90]. Then light sources are culled using bounding volumes of the above light ranges without introducing the darkening bias. However, in order to utilize existing tiled culling techniques which rely on sphere bounding volumes, this method assumed the radiant intensity of a light source has a low-frequency directional distribution. Therefore, VPLs on glossy surfaces (which have high-frequency directional distributions) could not be culled efficiently.

To render caustics lit by such glossy VPLs (shown in Fig. 1), this paper presents a stochastic light culling method taking a microfacet bidirectional reflectance distribution function (BRDF) into

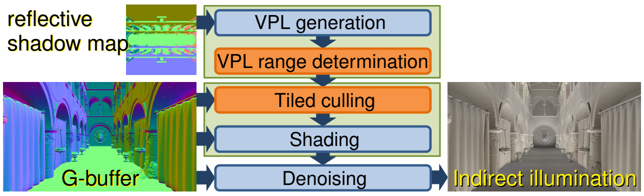


Figure 2: *Rendering pipeline based on tiled deferred shading [And11, Ste15] for real-time single-bounce indirect illumination. Our contributions are implemented in orange stages. Each green box is implemented with a single-pass compute shader.*

account. Similar to Dachsbacher and Stamminger [DS06], we use ellipsoids instead of spheres to bound the light range. This is a simple extension for light culling, but the difficulty lies in determining a tighter bounding ellipsoid for physically plausible materials used in modern computer graphics productions. Dachsbacher and Stamminger [DS06] introduced a bounding ellipsoid of the isosurface of a glossy VPL for the Phong reflection model [Pho75] which does not satisfy physically based constraints such as the Helmholtz reciprocity. As an alternative, this paper introduces a bounding ellipsoid for the influence range of a VPL reflected from the GGX microfacet BRDF [WMLT07]. Our bounding ellipsoid is simple, and easy to calculate without any precomputation. In addition, this paper also presents an efficient tiled culling implementation for bounding ellipsoids, which can be faster than rasterization-based culling techniques [DS06, OP16]. Using our method, we are able to render single-bounce glossy caustics reflected from GGX microsurfaces at real-time frame rates.

Our contributions are as follows.

- The stochastic light culling method is extended to use a direction-dependent influence range for a glossy VPL (Sec. 3).
- The bounding ellipsoid of the light range is derived analytically for a VPL reflected from the GGX microfacet BRDF (Sec. 4).
- This paper presents an efficient tiled culling implementation for bounding ellipsoids (Sec. 5).

Our rendering pipeline is shown in Fig. 2. The above contributions are implemented in the orange stages of this pipeline.

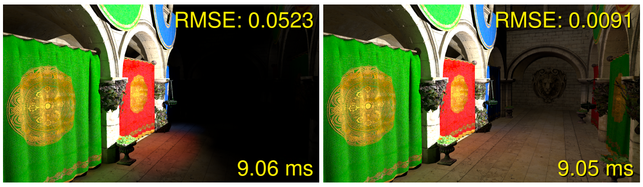
2. Related Work

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

VPLs are often used for representing indirect illumination [Kel97]. Single-bounce VPLs lit from point or directional lights can be generated at real-time frame rates by rendering reflective shadow maps [DS05]. Visibilities for VPLs can be approximated using imperfect shadow maps [RGK*08, REH*11, BBH13]. To sample important VPLs from the generated VPLs, sophisticated methods such as lightcuts [WFA*05, DGS12] have been developed for offline rendering. For real-time rendering, interleaved sampling [WKB*02, SIMP06] has often been used to reduce the number of VPLs per pixel. Furthermore, hundreds or thousands of VPLs were often resampled from a reflective shadow map according to an image-based probability density function [DS06, REH*11]. Since the probability density function can change dynamically if we take view-dependent importance into account, this resampling can increase flickering artifacts. To improve the temporal coherence, various studies have been conducted [LSK*07, BBH13, HKL16]. While the original VPL method is theoretically unbiased, variance is visible as spiky artifacts especially for glossy materials [KFB10]. These artifacts can be suppressed by clamping estimated radiance for each VPL, however darkening bias can be visible if the number of VPLs is insufficient. To avoid such artifacts, rich-VPLs [SHD15] were developed primarily for offline rendering. For real-time or interactive rendering, VPLs were often clustered and then approximated using a smaller number of virtual lights [PKD12, Tok15, Tok16]. Novák et al. [NED11] proposed a screen-space bias compensation. In this paper, we alleviate the above artifacts by generating tens of thousands of VPLs without resampling. Instead, this paper uses stochastic light culling to reduce the computational burden.

Light culling is a well-established acceleration technique in recent video games. This technique restricts the light range, and then performs shading only inside that range. Dachsbacher and Stamminger [DS06] rendered glossy caustics by splatting bounding geometries around VPLs. To create the bounding geometry, they derived a bounding ellipsoid of an isosurface of reflected light for the Phong reflection model and Lambert reflection model. Nichols and Wyman [NW10] proposed an adaptive multiresolution approach to reduce the fill rate of splatting. Tiled culling [OA11, Har12, Ste15] is a compute-based technique using a depth buffer. In this technique, lights are binned into 2D screen-space tiles taking the range of depth in each tile into account. Clustered shading [OBA12] uses 3D binning (i.e., clustering in the depth axis for each tile) to improve the culling precision. Donell and Chajdas [OC17] improved this depth clustering using tiled light trees. Generally, these compute-based culling techniques use bounding spheres. To support arbitrarily shaped convex bounding volumes, Örtegren and Persson [OP16] proposed clustered shading using conservative rasterization. While light culling improves the performance significantly, it was an inconsistent estimator and produced undesirable darkening bias due to the limited light range. To avoid this bias, stochastic light culling [TH16] using random light ranges was introduced. Laurent et al. [LDdLRB16] also chose their support functions randomly for splatting. However, these methods assumed diffuse-to-diffuse indirect illumination. This paper extends stochastic light culling to render glossy caustics reflected from the GGX microfacet BRDF.

The microfacet BRDF was introduced to computer graphics by Cook and Torrance [CT82]. Unlike the Phong reflection model, this BRDF satisfies physically based constraints [Hei14] using an appropriate masking-shadowing function such as the Smith model [Smi67]. The GGX normal distribution function (NDF) [TR75, WMLT07] is nowadays widely used for the microfacet BRDF in computer graphics productions [Bur12, Lag14]. However, as far as we know, a bounding ellipsoid of an isosurface of reflected light for the microfacet BRDF has not been presented



Light culling with fixed ranges

Stochastic light culling

Figure 3: *Stochastic light culling avoids darkening bias which occurs far from a VPL.*

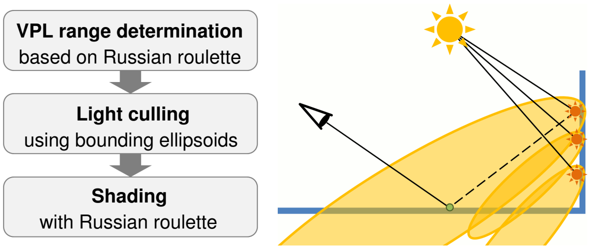


Figure 4: *Stochastic light culling for glossy VPLs. The influence range of each VPL is randomly determined based on Russian roulette according to the BRDF. Then VPLs are culled using bounding ellipsoids of those random ranges before shading.*

yet. Therefore, this paper derives the bounding ellipsoid using a property of the GGX distribution.

3. Stochastic Light Culling for Glossy VPLs

3.1. Stochastic light culling

The stochastic light culling method randomly rejects unimportant lights by determining their influence ranges based on Russian roulette. For each light, Russian roulette decides whether to accept or reject according to a probability proportional to the radiance at the shading point. For a point light, this radiance is given by

$$L(\boldsymbol{\omega}_o, l) = \frac{I(\boldsymbol{\omega}_o)}{l^2}, \tag{1}$$

where $I(\boldsymbol{\omega}_o)$ is the radiant intensity, and $\boldsymbol{\omega}_o \in \mathcal{S}^2$ and $l \in [0, \infty)$ are the direction and distance from the point light to the shading point, respectively. In this paper, we use the following probability:

$$p(\boldsymbol{\omega}_o, l) = \min\left(\frac{I(\boldsymbol{\omega}_o)}{\delta l^2}, 1\right), \tag{2}$$

where $\delta \in (0, \infty)$ is a user-specified parameter to control variance ($\delta = 0.001$ is used in this paper). If a point light is accepted, the radiance $L(\boldsymbol{\omega}_o, l)$ is divided by the probability $p(\boldsymbol{\omega}_o, l)$ as follows:

$$L(\boldsymbol{\omega}_o, l) \approx \begin{cases} \frac{L(\boldsymbol{\omega}_o, l)}{p(\boldsymbol{\omega}_o, l)} & (p(\boldsymbol{\omega}_o, l) > \xi) \\ 0 & (\text{otherwise}) \end{cases}, \tag{3}$$

where $\xi \in [0, 1)$ is a uniform random number. The stochastic light culling method uses a single random number ξ for each light source, and all the shading points will use the same ξ for a given light. The advantage of using a single random number is that we can bound the influence range for each light which allows us to utilize existing culling techniques in an unbiased fashion (Fig. 3).

3.2. BRDF-dependent influence range for a VPL

While Tokuyoshi and Harada [TH16] ignored the directionality of the probability $p(\boldsymbol{\omega}_o, l)$ in order to use bounding spheres and existing tiled culling methods, this paper takes the directionality into account for glossy VPLs (as shown in Fig. 4). The radiant intensity of a VPL is calculated by

$$I(\boldsymbol{\omega}_o) = \Phi f(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o) \max(\boldsymbol{\omega}_o \cdot \mathbf{n}, 0), \tag{4}$$

where Φ is the radiant flux of the photon that has arrived at the VPL location, $f(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o)$ is the BRDF, $\boldsymbol{\omega}_i \in \mathcal{S}^2$ is the incoming direction of the photon, and $\mathbf{n} \in \mathcal{S}^2$ is the geometry normal at the VPL. Thus, the VPL range shown in Fig. 4 is derived as

$$l_{\max}(\boldsymbol{\omega}_o) = p^{-1}(\xi) = \sqrt{\frac{\Phi f(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o) \max(\boldsymbol{\omega}_o \cdot \mathbf{n}, 0)}{\delta \xi}}. \tag{5}$$

This $l_{\max}(\boldsymbol{\omega}_o)$ is an isosurface of the radiance reflected by the BRDF $f(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o)$. To bound this isosurface, we use an ellipsoid similar to Dachsbacher and Stamminger [DS06]. Unlike them, we introduce a bounding ellipsoid for the GGX microfacet BRDF.

4. Bounding Ellipsoid for GGX Reflection

4.1. GGX microfacet BRDF

The microfacet BRDF [CT82] is modeled to represent light reflection from rough surfaces as follows:

$$f(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o) = \frac{F(\boldsymbol{\omega}_i \cdot \boldsymbol{\omega}_h) G_2(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o) D(\boldsymbol{\omega}_h \cdot \mathbf{n})}{4|\boldsymbol{\omega}_i \cdot \mathbf{n}||\boldsymbol{\omega}_o \cdot \mathbf{n}|}, \tag{6}$$

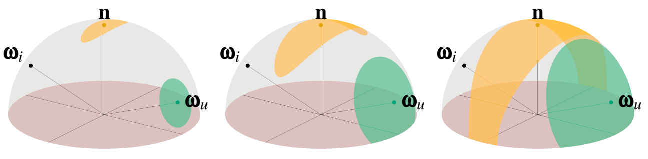
where $\boldsymbol{\omega}_h = \frac{\boldsymbol{\omega}_i + \boldsymbol{\omega}_o}{\|\boldsymbol{\omega}_i + \boldsymbol{\omega}_o\|}$ is the half vector, $F(\boldsymbol{\omega}_i \cdot \boldsymbol{\omega}_h) \in [0, 1]$ is the Fresnel factor, and $D(\boldsymbol{\omega}_h \cdot \mathbf{n})$ is the NDF to represent the distribution of microfacet normals. The GGX NDF [TR75, WMLT07] is a bell-shaped function defined by

$$D(\cos\theta_m) = \frac{\alpha^2 \chi^+(\cos\theta_m)}{\pi\left(\alpha^2 \cos^2\theta_m + \sin^2\theta_m\right)^2}, \tag{7}$$

where α is the roughness parameter, and $\chi^+(\cos\theta_m)$ is the Heaviside function: 1 if $\cos\theta_m > 0$ and 0 if $\cos\theta_m \leq 0$. This paper assumes $\alpha \in (0, 1]$ which is often used in computer graphics productions for data compression and ease of artists control. If $\alpha \in (0, 1]$, the GGX NDF is monotonically decreasing for $\theta_m \in [0, \frac{\pi}{2}]$. $G_2(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o) \in [0, 1]$ is the masking-shadowing function. In this paper, we use the Smith microsurface model [Smi67] whose masking function is a separable form: $G_1(\boldsymbol{\omega}_i, \boldsymbol{\omega}_h) = \chi^+(\boldsymbol{\omega}_i \cdot \boldsymbol{\omega}_h) G_1^{\text{dist}}(\boldsymbol{\omega}_i)$, where $\chi^+(\boldsymbol{\omega}_i \cdot \boldsymbol{\omega}_h)$ is the binary visibility of frontfacing microfacets, and $G_1^{\text{dist}}(\boldsymbol{\omega}_i)$ is independent from $\boldsymbol{\omega}_h$. From the constraint of the visible normal distribution [Hei14], $G_1^{\text{dist}}(\boldsymbol{\omega}_i)$ is given by

$$\begin{aligned} G_1^{\text{dist}}(\boldsymbol{\omega}_i) &= \frac{|\boldsymbol{\omega}_i \cdot \mathbf{n}|}{\int_{\mathcal{S}^2} D(\boldsymbol{\omega} \cdot \mathbf{n}) \max(\boldsymbol{\omega}_i \cdot \boldsymbol{\omega}, 0)\, d\boldsymbol{\omega}} \\ &= \frac{2|\boldsymbol{\omega}_i \cdot \mathbf{n}|}{|\boldsymbol{\omega}_i \cdot \mathbf{n}| + \sqrt{(1 - \alpha^2)(\boldsymbol{\omega}_i \cdot \mathbf{n})^2 + \alpha^2}}. \end{aligned} \tag{8}$$

There are several forms of the Smith masking-shadowing function such as the height-correlated form [Hei14], but any form satisfies $G_2(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o) \leq G_1^{\text{dist}}(\boldsymbol{\omega}_i)$.



(a) $\theta = 15°$ **(b)** $\theta = 30°$ **(c)** $\theta = 45°$

Figure 5: *When outgoing direction $\boldsymbol{\omega}_o$ is on the green spherical circle centered at $\boldsymbol{\omega}_u$, the halfvector $\boldsymbol{\omega}_h$ is on the orange spherical ellipse (a,b) or hyperbola (c). Thus the angular lower bound between $\boldsymbol{\omega}_h$ and* **n** *is obtained from the radius θ of the green circle.*

4.2. Our bounding ellipsoid

Since the microfacet BRDF is rather complex, we consider a simpler function equal to or greater than the reflection lobe as follows:

$$f(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o)\max(\boldsymbol{\omega}_o \cdot \mathbf{n}, 0) \leq \frac{F_{\max}(\boldsymbol{\omega}_i) G_1^{\text{dist}}(\boldsymbol{\omega}_i) D(\boldsymbol{\omega}_h \cdot \mathbf{n})}{4|\boldsymbol{\omega}_i \cdot \mathbf{n}|}, \tag{9}$$

where $F_{\max}(\boldsymbol{\omega}_i)$ is the maximum Fresnel factor for $\boldsymbol{\omega}_i$ (for the detail, please refer to the supplemental material). In the right side, the $\boldsymbol{\omega}_o$-dependent term is only the NDF $D(\boldsymbol{\omega}_h \cdot \mathbf{n})$. When the outgoing direction $\boldsymbol{\omega}_o$ is on a spherical circle centered at the perfect specular reflection direction $\boldsymbol{\omega}_u = 2(\boldsymbol{\omega}_i \cdot \mathbf{n})\mathbf{n} - \boldsymbol{\omega}_i$ with radius $\theta = \arccos(\boldsymbol{\omega}_o \cdot \boldsymbol{\omega}_u)$, $\boldsymbol{\omega}_h$ is on a spherical ellipse with semi-minor axis $\frac{\theta}{2}$ (Figs. 5a and 5b), or a spherical hyperbola with semi-major axis $\frac{\theta}{2}$ (Fig. 5c) [JHY*14]. Therefore, we obtain the following angular lower bound between $\boldsymbol{\omega}_h$ and **n**:

$$\arccos(\boldsymbol{\omega}_h \cdot \mathbf{n}) \geq \frac{\theta}{2}. \tag{10}$$

This inequality becomes equal when the outgoing direction $\boldsymbol{\omega}_o$ is on the great circle passing through the incoming direction $\boldsymbol{\omega}_i$ and normal **n**. For $\alpha \in (0, 1]$, since the GGX NDF is monotonically decreasing, Eq. (10) yields the following inequality:

$$D(\boldsymbol{\omega}_h \cdot \mathbf{n}) \leq D\left(\cos\frac{\theta}{2}\right). \tag{11}$$

Hence, the isosurface of the GGX-based glossy reflection (Eq. (5)) is enclosed by the following surface $s(\boldsymbol{\omega}_o)$:

$$\begin{aligned} l_{\max}(\boldsymbol{\omega}_o) \leq s(\boldsymbol{\omega}_o) &= \sqrt{\frac{\Phi F_{\max}(\boldsymbol{\omega}_i) G_1^{\text{dist}}(\boldsymbol{\omega}_i) D\left(\cos\frac{\theta}{2}\right)}{4\delta\xi|\boldsymbol{\omega}_i \cdot \mathbf{n}|}} \\ &= r\sqrt{\pi D\left(\cos\frac{\theta}{2}\right)}, \end{aligned} \tag{12}$$

where $r = \sqrt{\frac{\Phi F_{\max}(\boldsymbol{\omega}_i) G_1^{\text{dist}}(\boldsymbol{\omega}_i)}{4\pi\delta\xi|\boldsymbol{\omega}_i \cdot \mathbf{n}|}}$. Surprisingly, this surface $s(\boldsymbol{\omega}_o)$ is a spheroid as shown in Fig. 6. For the derivation, please refer to Appendix A. The semiaxes of this spheroid are

$$\boxed{(r_u, r_v, r_w) = \left(\frac{1+\alpha^2}{2\alpha} r, r, r\right)}. \tag{13}$$

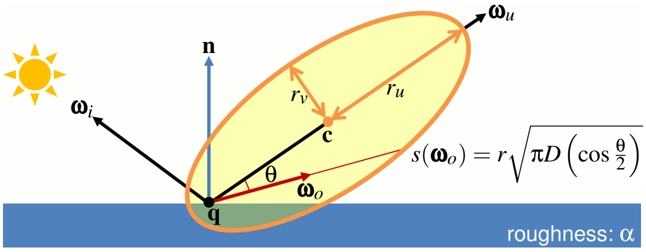
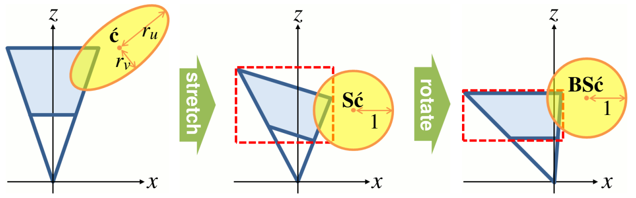


Figure 6: *Isosurface of microfacet-based reflection is enclosed by surface $s(\boldsymbol{\omega}_o)$. This surface is a spheroid for the GGX NDF.*



(a) *view space* **(b)** *stretched space* **(c)** *our test space*

Figure 7: *Ellipsoid-frustum intersection test for tiled culling (a) can be calculated using a sphere-frustum intersection test in the stretched space (b). This paper additionally rotates the test space for optimization (c).*

The rotation matrix of this spheroid is

$$\boxed{\mathbf{R} = \begin{bmatrix} \boldsymbol{\omega}_u & \boldsymbol{\omega}_w \times \boldsymbol{\omega}_u & \boldsymbol{\omega}_w \end{bmatrix}}, \tag{14}$$

where $\boldsymbol{\omega}_w \in \mathcal{S}^2$ is a unit vector orthogonal to $\boldsymbol{\omega}_u$ and **n** (i.e., $\boldsymbol{\omega}_w = \frac{\boldsymbol{\omega}_u \times \mathbf{n}}{\|\boldsymbol{\omega}_u \times \mathbf{n}\|}$ for $\boldsymbol{\omega}_u \neq \mathbf{n}$). The center of the spheroid is

$$\boxed{\mathbf{c} = \mathbf{q} + \frac{1-\alpha^2}{2\alpha} r \boldsymbol{\omega}_u}, \tag{15}$$

where $\mathbf{q} \in \mathbb{R}^3$ is the position of the VPL.

5. Tiled Culling Using Bounding Ellipsoids

Although rasterization-based culling methods [DS06, OP16] support bounding ellipsoids, this paper extends compute-based tiled culling to use the ellipsoids. This compute-based culling is less expensive than those rasterization-based culling methods (as shown in the experimental results Sec. 7), when stochastic light culling is combined with interleaved sampling described in Sec. 6 for tens of thousands of light sources. This extension is mathematically trivial, but its calculation cost is higher than bounding spheres. Therefore, this paper introduces an optimization technique.

5.1. Ellipsoid-frustum intersection test

Tiled culling performs a rough intersection test of a bounding volume and frustum for each tile in view space (Fig. 7a). The view-space ellipsoid is represented using the rotation matrix $\acute{\mathbf{R}} = \mathbf{V}\mathbf{R}$ and center position $\acute{\mathbf{c}} = \mathbf{V}\mathbf{c} + \mathbf{o}$, where the 3×3 matrix **V** and $\mathbf{o} \in \mathbb{R}^3$ are

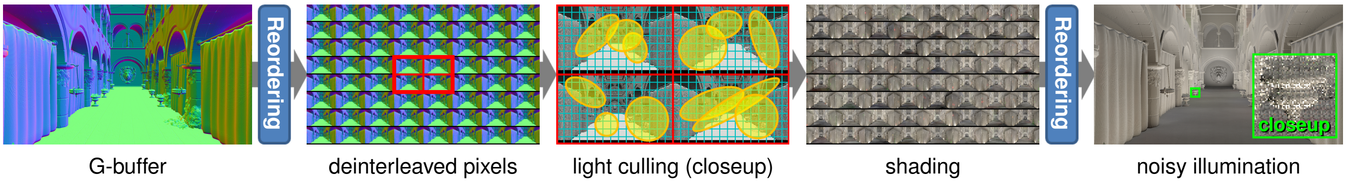


Figure 8: *Combination of interleaved sampling and stochastic light culling. Pixels are reordered into 8×8 subregions for interleaved sampling of 65536 VPLs. Then, our stochastic light culling is performed for each subregion with a different subset of VPLs (i.e., 1024 VPLs per subregion). For this algorithm, variance is visible as noise which can be removed using a denoising filter.*

the rotation and translation from world space to view space. The intersection test of this ellipsoid and a frustum can be equivalently expressed as a sphere-frustum intersection test by stretching the space using the following transformation matrix:

$$\mathbf{S} = \acute{\mathbf{R}} \begin{bmatrix} \frac{1}{r_u} & 0 & 0 \\ 0 & \frac{1}{r_v} & 0 \\ 0 & 0 & \frac{1}{r_w} \end{bmatrix} \acute{\mathbf{R}}^{\mathrm{T}}. \tag{16}$$

Using this matrix, each bounding ellipsoid is stretched into a unit sphere (Fig. 7b). Therefore, by transforming the test space for each light source, we can reuse existing sphere-based culling methods with a slight modification for bounding ellipsoids. This paper employs the Modified HalfZ culling method [Ste15] which uses two depth clusters and an axis-aligned bounding box (AABB) of a frustum for the rough sphere-frustum intersection test.

5.2. Acceleration by rotating the test space

For code optimization, most existing culling implementations assume the depth plane of the frustum is perpendicular to the z-axis. However, our stretching transformation violates this assumption. This increases the code complexity of AABB calculation for the rough intersection test. In addition, it captures too many false positives because of the mismatch between the stretched frustum and AABB (Fig. 7b). To alleviate these problems, this paper additionally rotates the test space using the following rotation matrix:

$$\mathbf{B} = \begin{cases} \begin{bmatrix} \frac{\mathbf{b}_x}{\|\mathbf{b}_x\|} & \frac{\mathbf{b}_z \times \mathbf{b}_x}{\|\mathbf{b}_z \times \mathbf{b}_x\|} & \frac{\mathbf{b}_z}{\|\mathbf{b}_z\|} \end{bmatrix}^{\mathrm{T}} & (\|\mathbf{b}_x\| > \|\mathbf{b}_y\|) \\ \begin{bmatrix} \frac{\mathbf{b}_y \times \mathbf{b}_z}{\|\mathbf{b}_y \times \mathbf{b}_z\|} & \frac{\mathbf{b}_y}{\|\mathbf{b}_y\|} & \frac{\mathbf{b}_z}{\|\mathbf{b}_z\|} \end{bmatrix}^{\mathrm{T}} & (\text{otherwise}) \end{cases}, \tag{17}$$

where $\mathbf{b}_x = \mathbf{S}\begin{bmatrix}1 & 0 & 0\end{bmatrix}^{\mathrm{T}}$, $\mathbf{b}_y = \mathbf{S}\begin{bmatrix}0 & 1 & 0\end{bmatrix}^{\mathrm{T}}$, and $\mathbf{b}_z = \mathbf{b}_x \times \mathbf{b}_y$. Rotation matrix $\mathbf{B}$ is designed so that the depth plane of the transformed frustum is perpendicular to the z-axis (Fig. 7c). This allows us to simplify the AABB calculation code and to make a tighter AABB for the sheared frustum. In this paper, matrix $\mathbf{BS}$ and position $\mathbf{BS}\acute{\mathbf{c}}$ are computed and stored into memory for each light source. Then, the light culling stage loads these $\mathbf{BS}$ and $\mathbf{BS}\acute{\mathbf{c}}$ for each ellipsoid-frustum intersection test. The source code of our culling implementation is described in the supplemental material.

6. Implementation Details

Interleaved sampling. In addition to stochastic light culling, this paper employs a GPU-friendly interleaved sampling technique [SIMP06] to reduce the number of VPLs per pixel (Fig. 8). This technique first deinterleaves pixels in a regular sampling pattern into subregions of the screen. Then, for each subregion, shading is performed using a different subset of VPLs. For example, using 8×8 interleaved sampling for 65536 VPLs, the number of VPLs per subregion is reduced to 1024. In this paper, our stochastic light culling is applied for each subregion. This interleaved sampling technique does not only reduce the shading time, but also the culling time. We implement this algorithm in a single pass based on tiled deferred shading [And11]. Although the variance is visible as noise in the rendered image, this noise is removed using a cross bilateral filter [PSA*04, ED04] in postprocessing.

Diffuse VPLs. Although a bounding ellipsoid can also be used for a diffuse VPL, this shape (derived by Dachsbacher and Stamminger [DS06]) is almost a sphere. Therefore, for a diffuse VPL, we use a bounding sphere whose radius is equal to the longest semiaxis of the ellipsoid. In this paper, the center and radius of this sphere are stochastically given by $\mathbf{q} + \left(\frac{1}{3}\right)^{\frac{3}{4}} \sqrt{\frac{\Phi k}{\pi \delta \xi}} \mathbf{n}$ and $\left(\frac{4}{27}\right)^{\frac{1}{4}} \sqrt{\frac{\Phi k}{\pi \delta \xi}}$, respectively, where k is the diffuse reflectance.

7. Experimental Results

Here we present single-bounce diffuse-to-diffuse and glossy-to-diffuse indirect illumination using our method. All images are rendered at 1920×1080 screen resolution on an NVIDIA GeForce GTX 970 GPU. A VPL is generated from each texel of a reflective shadow map of 256^2 resolution (i.e., 65536 VPLs in total). While Tokuyoshi and Harada [TH16] generated an imperfect shadow map for each VPL, this paper omits these shadow maps to quantitatively evaluate only the sampling efficiency for VPLs. 8×8 interleaved sampling and cross bilateral filtering of kernel radius 8 are employed to reduce the number of VPLs per pixel. The tile size is 16×16 pixels for tiled culling. The image quality is evaluated using the root mean squared error (RMSE) metric.

False positives. Fig. 9 shows the number of glossy VPLs per pixel and false discovery rate for different scenes. For comparison, this paper uses two types of bounding spheres. One is a sphere centered at the VPL position $\mathbf{q}$ based on the previous work [TH16] which uses the maximum of radiant intensity for their error bound-based radius. For this radius, we use $\frac{r}{\alpha}$ (which is derived using the maximum of the NDF instead of the NDF for Eq. (9)). Another one is a bounding sphere enclosing our bounding ellipsoid, given by radius r_u and center $\mathbf{c}$. While there are a small number of

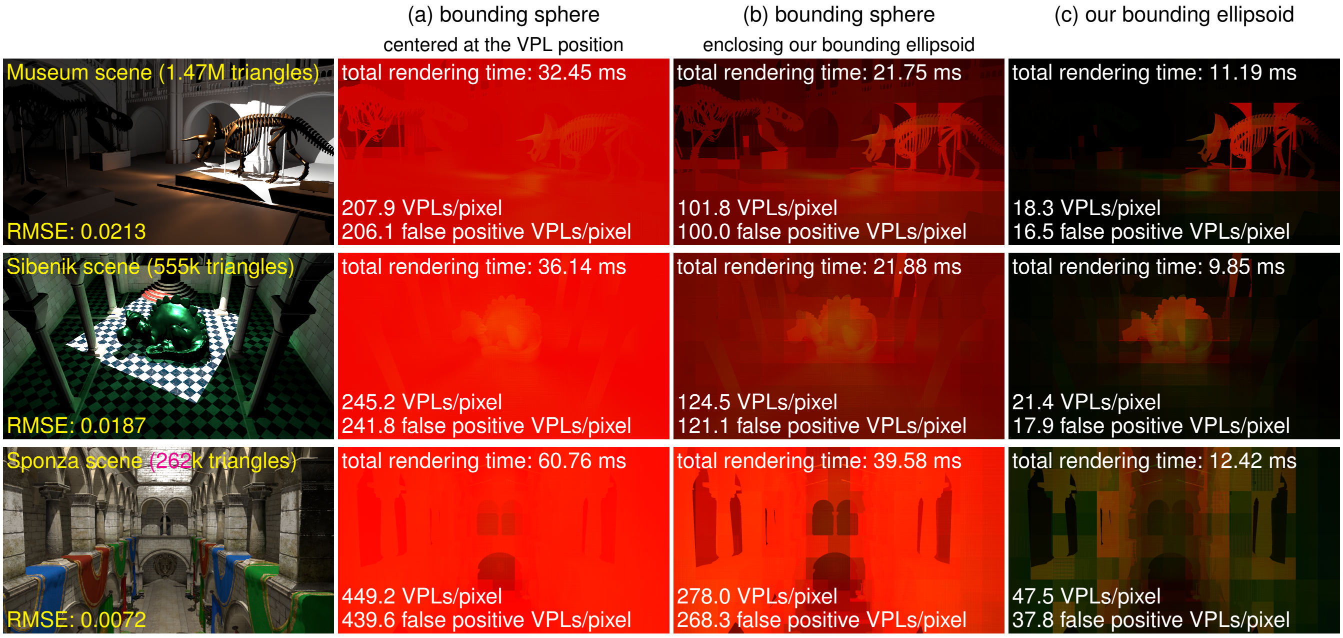

(a) bounding sphere
centered at the VPL position
(b) bounding sphere
enclosing our bounding ellipsoid
(c) our bounding ellipsoid
Museum scene (1.47M triangles)
RMSE: 0.0213
total rendering time: 32.45 ms
207.9 VPLs/pixel
206.1 false positive VPLs/pixel
total rendering time: 21.75 ms
101.8 VPLs/pixel
100.0 false positive VPLs/pixel
total rendering time: 11.19 ms
18.3 VPLs/pixel
16.5 false positive VPLs/pixel
Sibenik scene (555k triangles)
RMSE: 0.0187
total rendering time: 36.14 ms
245.2 VPLs/pixel
241.8 false positive VPLs/pixel
total rendering time: 21.88 ms
124.5 VPLs/pixel
121.1 false positive VPLs/pixel
total rendering time: 9.85 ms
21.4 VPLs/pixel
17.9 false positive VPLs/pixel
Sponza scene (262k triangles)
RMSE: 0.0072
total rendering time: 60.76 ms
449.2 VPLs/pixel
439.6 false positive VPLs/pixel
total rendering time: 39.58 ms
278.0 VPLs/pixel
268.3 false positive VPLs/pixel
total rendering time: 12.42 ms
47.5 VPLs/pixel
37.8 false positive VPLs/pixel

Figure 9: *Visualization of the number of glossy VPLs per pixel (brightness, maximum: 256 VPLs) and false discovery rate (spectrum).*

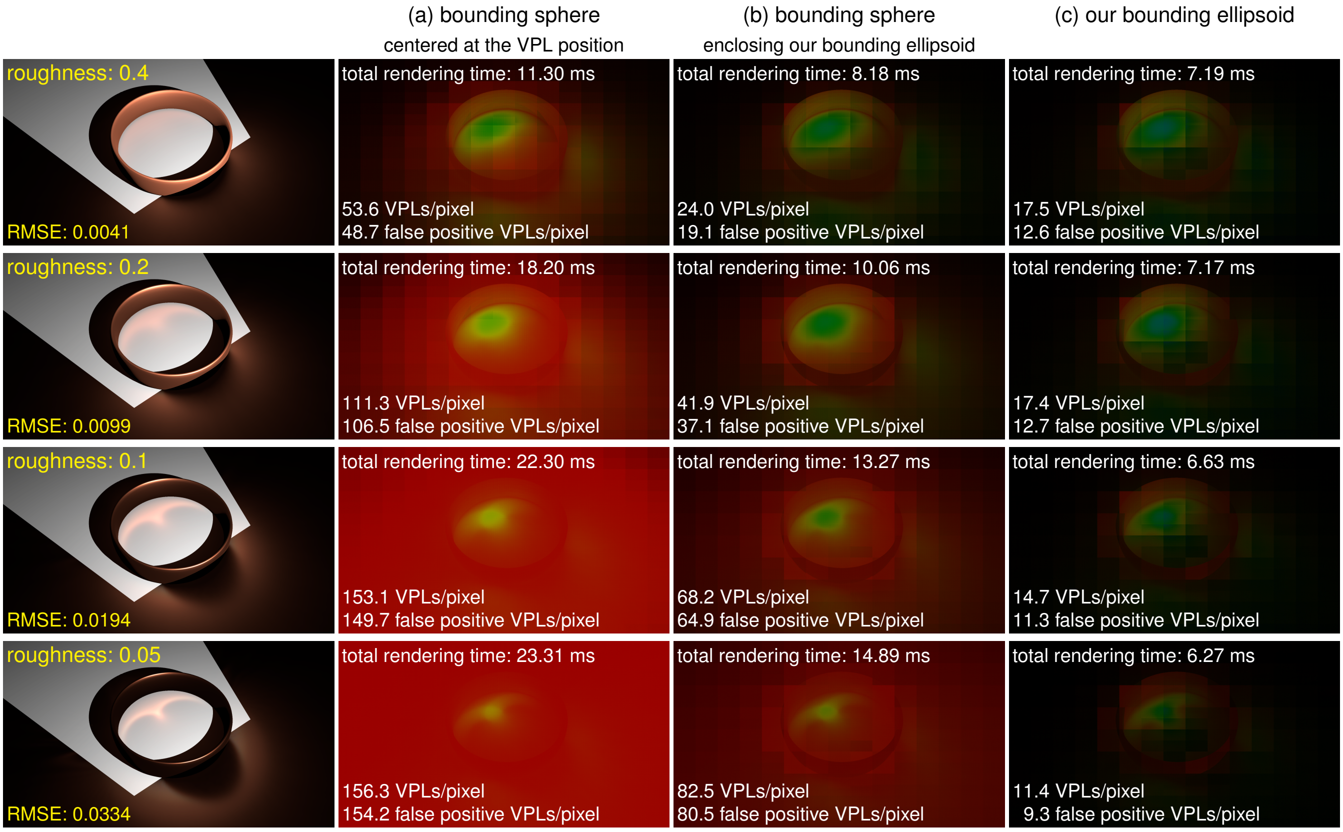

(a) bounding sphere
centered at the VPL position
(b) bounding sphere
enclosing our bounding ellipsoid
(c) our bounding ellipsoid
roughness: 0.4
RMSE: 0.0041
total rendering time: 11.30 ms
53.6 VPLs/pixel
48.7 false positive VPLs/pixel
total rendering time: 8.18 ms
24.0 VPLs/pixel
19.1 false positive VPLs/pixel
total rendering time: 7.19 ms
17.5 VPLs/pixel
12.6 false positive VPLs/pixel
roughness: 0.2
RMSE: 0.0099
total rendering time: 18.20 ms
111.3 VPLs/pixel
106.5 false positive VPLs/pixel
total rendering time: 10.06 ms
41.9 VPLs/pixel
37.1 false positive VPLs/pixel
total rendering time: 7.17 ms
17.4 VPLs/pixel
12.7 false positive VPLs/pixel
roughness: 0.1
RMSE: 0.0194
total rendering time: 22.30 ms
153.1 VPLs/pixel
149.7 false positive VPLs/pixel
total rendering time: 13.27 ms
68.2 VPLs/pixel
64.9 false positive VPLs/pixel
total rendering time: 6.63 ms
14.7 VPLs/pixel
11.3 false positive VPLs/pixel
roughness: 0.05
RMSE: 0.0334
total rendering time: 23.31 ms
156.3 VPLs/pixel
154.2 false positive VPLs/pixel
total rendering time: 14.89 ms
82.5 VPLs/pixel
80.5 false positive VPLs/pixel
total rendering time: 6.27 ms
11.4 VPLs/pixel
9.3 false positive VPLs/pixel

Figure 10: *Visualization of the number of glossy VPLs per pixel (brightness, maximum: 256 VPLs) and false discovery rate (spectrum) for different roughness (8.19k triangles).*

	Museum	Sibenik	Sponza
G-buffer	0.986	0.618	0.628
Reflective shadow map	0.610	0.270	0.121
VPL generation & range determination	0.136	0.140	0.140
Culling & shading	7.398	6.960	9.745
Denoising (cross bilateral filter)	1.320	1.328	1.320
Total	10.450	9.176	11.956

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

true positive VPLs for glossy caustics, sphere-based tiled culling produces numerous false positives. Using our bounding ellipsoid, false positives are reduced by over 90% which leads to a significant performance improvement. Fig. 10 shows false positives for different roughnesses. The use of bounding spheres produces more false positives for smaller roughness, and thus it is inefficient to render sharp caustics. On the other hand, the use of our bounding ellipsoid is more effective for smaller roughness.

Computation time. Table 1 shows the computation time for each pass in our method. Our contribution is VPL generation and range determination pass and culling and shading pass. These passes are independent from the triangle count of the scene. The VPL generation and range determination time is almost constant and negligible for 65536 VPLs. The main bottleneck is the culling and shading pass for VPLs. This computation time depends on the distribution of VPLs.

Culling. Table 2 shows the performance comparison between splatting [DS06], clustered shading using conservative rasterization [OP16], and our tiled culling (described in Sec. 5) for the combination of interleaved sampling and stochastic light culling. This table describes only the computation times of the culling and shading pass for diffuse VPLs only (i.e., bounding spheres) and glossy VPLs only (i.e., bounding ellipsoids). For a bounding geometry, 80 triangles are used in splatting, and 20 triangles are used in clustered shading. For clustered shading, this paper employs 32 depth clusters for each 16×16 pixel tile, and allocates 256MB for the node buffer of the light list. These parameters were obtained empirically. In this experiment, our tiled culling implementation is faster than these rasterization-based culling methods both for diffuse VPLs and glossy VPLs. This is because interleaved sampling reduces the iteration count of the main loop in compute-based tiled culling. On the other hand, in rasterization-based culling, interleaved sampling reduces only the pixel shader cost. In addition, the optimal tile size of clustered shading for the low-resolution subregion (i.e., 240×135 pixels) is smaller than the tile size for a high-resolution screen. For such a small tile size (e.g., 16×16 pixels), tiled culling can be more efficient than clustered shading. Hence, our tiled culling is more suitable to interleaved sampling than rasterization-based culling for tens of thousands of VPLs.

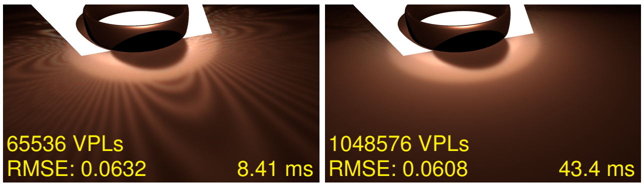


Figure 11: *Caustics created by highly specular surfaces (roughness: 0.02). A larger number of VPLs has to be generated for higher-frequency BRDFs to reduce the error.*

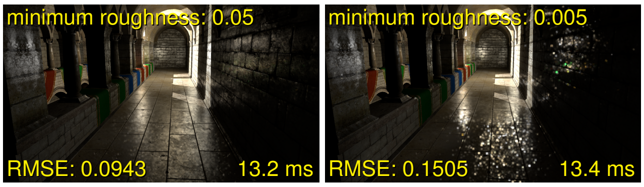


Figure 12: *Our method produces noticeable variance for diffuse-to-glossy and glossy-to-glossy indirect illumination, especially when a scene has highly specular BRDFs.*

8. Limitations

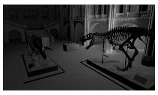
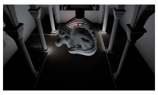
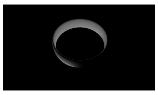
Highly specular surfaces. While our method culls VPLs according to the contribution of each VPL, the rendering quality is limited by the density of VPLs before culling, similar to lightcuts. Therefore, it is necessary to generate more VPLs for higher-frequency BRDFs (Fig. 11). For such a large number of VPLs, the culling stage before shading can be a bottleneck. In addition, denoising postprocessing can blur detailed caustics. Our method cannot render caustics reflected from perfect specular surfaces (i.e., $\alpha = 0$).

Glossy-to-glossy interreflections. While this paper improves the sampling efficiency of glossy VPLs, the BRDF at a shading point is still ignored for the sampling probability. This limitation can produce significant variance if the shading point has a highly specular BRDF (Fig. 12). For diffuse-to-glossy and glossy-to-glossy interreflections, biased approximation methods such as screen-space reflection [Sta15] can be used instead of our method for practical use. Future research would investigate an unbiased culling method taking glossy-to-glossy interreflections into account.

Anisotropic reflection lobes. Our bounding spheroid is well fitted on the plane defined by the incoming direction $\boldsymbol{\omega}_i$ and normal $\mathbf{n}$. On the other hand, it can be loose for the direction $\boldsymbol{\omega}_w$ (which is orthogonal to that plane) because the reflection lobe of the microfacet BRDF model is anisotropic for a grazing incoming direction. Therefore, there might be room to reduce the semiaxis r_w. Furthermore, our bounding spheroid does not take anisotropic NDFs into account. Although a bounding spheroid can be calculated using the maximum roughness for an anisotropic NDF, it can produce a loose bounding volume. Shrinking the bounding ellipsoid according to this anisotropic reflection is our future work.

9. Conclusion

This paper has presented a real-time unbiased culling method for glossy caustic paths. We have found $\sqrt{D\left(\cos\frac{\theta}{2}\right)}$ is a spheroid, if

Diffuse VPLs only:							
Splatting [DS06]	17.05	12.96	16.48	53.03	21.42	12.39	7.22
Clustered shading [OP16]	12.69	10.89	12.36	10.81	6.54	13.06	7.19
Our culling & shading	**3.75**	**3.05**	**3.27**	**3.83**	**2.56**	**3.23**	**1.98**

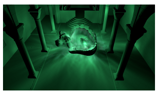
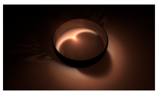
Glossy VPLs only:							
Splatting [DS06]	10.12	8.91	17.67	5.93	58.81	14.68	13.02
Clustered shading [OP16]	5.92	4.99	8.13	3.92	15.66	10.99	5.40
Our culling & shading	**3.73**	**2.87**	**3.94**	**2.15**	**7.28**	**5.14**	**2.47**

Table 2: *Computation times of the culling and shading pass for diffuse VPLs only (upper table) and glossy VPLs only (lower table) (ms). Each image is indirect illumination after denoising.*

$D(\cos\theta)$ is the GGX distribution. Thanks to this property, the influence range of a VPL on GGX microsurfaces can now be enclosed by a spheroid for stochastic light culling. To perform tiled culling using such bounding ellipsoids, this paper also introduced an efficient implementation. Our implementation outperforms a state-of-the-art rasterization-based culling technique when combined with interleaved sampling for tens of thousands of light sources. Using these techniques, we are able to render dynamic single-bounce caustics at real-time frame rates for a scene with the GGX microfacet BRDF. For future work, we would like to further shrink the bounding volume for anisotropic reflection lobes. In addition, we would also like to investigate efficient culling techniques for a larger number of VPLs to render caustics created by more highly specular surfaces.

Acknowledgements

The polygon models are courtesy of A. L. Bautista, J. Andersdon, M. Dabrovic, F. Meinl, UTIA, Academy of Sciences of the Czech Republic, and CGG, Czech Technical University in Prague.

Appendix A: Derivation of Our Bounding Ellipsoid

Since $D\left(\cos\frac{\theta}{2}\right)$ is an isotropic distribution and r is constant for $s(\boldsymbol{\omega}_o)$, the surface $s(\boldsymbol{\omega}_o) = r\sqrt{\pi D\left(\cos\frac{\theta}{2}\right)}$ can be represented using a 2D Cartesian coordinate system as follows:

$$\begin{aligned} u &= s(\boldsymbol{\omega}_o)\cos\theta, && (18)\\ v &= s(\boldsymbol{\omega}_o)\sin\theta. && (19) \end{aligned}$$

Eq. (18) is a function of $\cos\theta$ as follows:

$$\begin{aligned} u &= r\sqrt{\pi D\left(\cos\frac{\theta}{2}\right)}\cos\theta \\ &= \frac{r\alpha\cos\theta}{\alpha^2\cos^2\frac{\theta}{2}+\sin^2\frac{\theta}{2}} \\ &= \frac{2r\alpha\cos\theta}{\alpha^2(1+\cos\theta)+1-\cos\theta}. \end{aligned} \tag{20}$$

Thus, $\cos\theta$ is represented using u as follows:

$$\cos\theta = \frac{\left(1+\alpha^2\right)u}{\left(1-\alpha^2\right)u+2r\alpha}. \tag{21}$$

Assigning this equation into Eq. (19), we obtain the following equation:

$$\begin{aligned} v^2 &= \left(s(\boldsymbol{\omega}_o)\right)^2\left(1-\cos^2\theta\right) \\ &= \frac{u^2}{\cos^2\theta}-u^2 \\ &= \left(\frac{\left(1-\alpha^2\right)u+2r\alpha}{1+\alpha^2}\right)^2-u^2 \\ &= \frac{4(r-\alpha u)\left(r\alpha^2+\alpha u\right)}{\left(1+\alpha^2\right)^2}. \end{aligned} \tag{22}$$

Let $u = t+\frac{1-\alpha^2}{2\alpha}r$, then it yields the equation of an ellipse as follows:

$$\begin{aligned} v^2 &= \frac{4\left(r-\alpha\left(t+\frac{1-\alpha^2}{2\alpha}r\right)\right)\left(r\alpha^2+\alpha\left(t+\frac{1-\alpha^2}{2\alpha}r\right)\right)}{\left(1+\alpha^2\right)^2} \\ &= r^2-\frac{t^2}{\left(\frac{1+\alpha^2}{2\alpha}\right)^2} \\ \frac{v^2}{r^2} &= 1-\frac{t^2}{\left(\frac{1+\alpha^2}{2\alpha}r\right)^2} \\ 1 &= \frac{t^2}{\left(\frac{1+\alpha^2}{2\alpha}r\right)^2}+\frac{v^2}{r^2}. \end{aligned} \tag{23}$$

Since $s(\boldsymbol{\omega}_o)$ is an isotropic function centered at $\boldsymbol{\omega}_u$ (i.e., u-axis), it is a spheroid in 3D space.

References

[AK90] Arvo J., Kirk D.: Particle transport and image synthesis. *SIGGRAPH Comput. Graph. 24*, 4 (1990), 63–66. 1

[And11] ANDERSSON J.: DirectX 11 rendering in Battlefield 3. In *GDC '11* (2011). 2, 5

[BBH13] BARÁK T., BITTNER J., HAVRAN V.: Temporally coherent adaptive sampling for imperfect shadow maps. *Comput. Graph. Forum 32*, 4 (2013), 87–96. 2

[Bur12] BURLEY B.: Physically-based shading at Disney. In *SIGGRAPH '12 Course: Practical Physically-based Shading in Film and Game Productio* (2012), pp. 10:1–10:7. 2

[CT82] COOK R. L., TORRANCE K. E.: A reflectance model for computer graphics. *ACM Trans. Graph. 1*, 1 (1982), 7–24. 2, 3

[DGS12] DAVIDOVIČ T., GEORGIEV I., SLUSALLEK P.: Progressive lightcuts for GPU. In *SIGGRAPH '12 Talks* (2012), pp. 1:1–1:1. 2

[DKH*14] DACHSBACHER C., KŘIVÁNEK 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. 1, 2, 3, 4, 5, 7, 8

[ED04] EISEMANN E., DURAND F.: Flash photography enhancement via intrinsic relighting. *ACM Trans. Graph. 23*, 3 (2004), 673–678. 5

[Har12] HARADA T.: A 2.5D culling for Forward+. In *SIGGRAPH Asia '12 Tech. Briefs* (2012), pp. 18:1–18:4. 1, 2

[Hei14] HEITZ E.: Understanding the masking-shadowing function in microfacet-based BRDFs. *J. Comput. Graph. Tech. 3*, 2 (2014), 48–107. 2, 3

[HKL16] HEDMAN P., KARRAS T., LEHTINEN J.: Sequential monte carlo instant radiosity. In *I3D'16* (2016), pp. 121–128. 2

[JHY*14] JAKOB W., HAŠAN M., YAN L.-Q., LAWRENCE J., RAMAMOORTHI R., MARSCHNER S.: Discrete stochastic microfacet models. *ACM Trans. Graph. 33*, 4 (2014), 115:1–115:10. 4

[Kel97] KELLER A.: Instant radiosity. In *SIGGRAPH'97* (1997), pp. 49–56. 1, 2

[KFB10] KŘIVÁNEK 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

[Lag14] LAGARDE S.: Moving Frostbite to physically based rendering. In *SIGGRAPH '14 Course: Physically Based Shading in Theory and Practice* (2014), pp. 23:1–23:8. 2

[LDdLRB16] LAURENT G., DELALANDRE C., DE LA RIVIÈRE G., BOUBEKEUR T.: Forward light cuts: A scalable approach to real-time global illumination. *Comput. Graph. Forum 35*, 4 (2016), 79–88. 2

[Lef12] LEFOHN A.: Introduction to beyond programmable shading 2012. In *SIGGRAPH '12 Course: Beyond Programmable Shading* (2012). 1

[LSK*07] LAINE S., SARANSAARI H., KONTKANEN J., LEHTINEN J., AILA T.: Incremental instant radiosity for real-time indirect illumination. In *EGSR'07* (2007), pp. 277–286. 2

[NED11] NOVÁK J., ENGELHARDT T., DACHSBACHER C.: Screen-space bias compensation for interactive high-quality global illumination with virtual point lights. In *I3D '11* (2011), pp. 119–124. 2

[NW10] NICHOLS G., WYMAN C.: Interactive indirect illumination using adaptive multiresolution splatting. *IEEE Trans. Vis. Comput. Graph. 16*, 5 (2010), 729–741. 2

[OA11] OLSSON O., ASSARSSON U.: Tiled shading. *J. Graph. GPU, and Game Tools* (2011), 235–251. 1, 2

[OBA12] OLSSON O., BILLETER M., ASSARSSON U.: Clustered deferred and forward shading. In *HPG '12* (2012), pp. 87–96. 1, 2

[OC17] O'DONNELL Y., CHAJDAS M. G.: Tiled light trees. In *I3D '17* (2017), pp. 1:1–1:7. 1, 2

[OP16] ÖRTEGREN K., PERSSON E.: Clustered shading: Assigning lights using conservative rasterization in DirectX 12. In *GPU Pro 7: Advanced Rendering Techniques*. A K Peters/CRC Press, 2016, pp. 43–81. 1, 2, 4, 7, 8

[Pho75] PHONG B. T.: Illumination for computer generated pictures. *Commun. ACM 18*, 6 (1975), 311–317. 2

[PKD12] PRUTKIN R., KAPLANYAN A. S., DACHSBACHER C.: Reflective shadow map clustering for real-time global illumination. In *EG'12 Short Papers* (2012), pp. 9–12. 2

[PSA*04] PETSCHNIGG G., SZELISKI R., AGRAWALA M., COHEN M., HOPPE H., TOYAMA K.: Digital photography with flash and no-flash image pairs. *ACM Trans. Graph. 23*, 3 (2004), 664–672. 5

[RDGK12] RITSCHEL 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] RITSCHEL 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. 2

[RGK*08] RITSCHEL 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

[SIMP06] SEGOVIA B., IEHL J.-C., MITANCHEY R., PÉROCHE B.: Non-interleaved deferred shading of interleaved sample patterns. In *Graph. Hardw. '06* (2006), pp. 53–60. 2, 5

[Smi67] SMITH B. G.: Geometrical shadowing of a random rough surface. *IEEE Trans. Antennas and Propagation 15*, 5 (1967), 668–671. 2, 3

[Sta15] STACHOWIAK T.: Stochastic screen-space reflections. In *SIGGRAPH '15 Course: Advances in Real Time Rendering* (2015). 7

[Ste15] STEWART J.: Compute-based tiled culling. In *GPU Pro 6: Advanced Rendering Techniques*. A K Peters/CRC Press, 2015, pp. 435–458. 1, 2, 5

[TH16] TOKUYOSHI Y., HARADA T.: Stochastic light culling. *J. Comput. Graph. Tech. 5*, 1 (2016), 35–60. 1, 2, 3, 5

[Tok15] TOKUYOSHI Y.: Virtual spherical Gaussian lights for real-time glossy indirect illumination. *Comput. Graph. Forum 34*, 7 (2015), 89–98. 2

[Tok16] TOKUYOSHI Y.: Modified filtered importance sampling for virtual spherical Gaussian lights. *Comput. Vis. Media 2*, 4 (2016), 343–355. 2

[TR75] TROWBRIDGE T. S., REITZ K. P.: Average irregularity representation of a rough surface for ray reflection. *J. Opt. Soc. Am 65*, 5 (1975), 531–536. 2, 3

[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. 2

[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. 2, 3

[Xu16] XU K.: Temporal antialiasing in Uncharted 4. In *SIGGRAPH '16 Course: Advances in Real Time Rendering* (2016). 1