DOI: 10.1111/cgf.14867
High-Performance Graphics 2023
J. Bikker and C. Gribble
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 42 (2023), Number 8
e14867 *pp. 1 - 6*

Sampling Visible GGX Normals with Spherical Caps

Jonathan Dupuy and Anis Benyoub

Intel Corporation

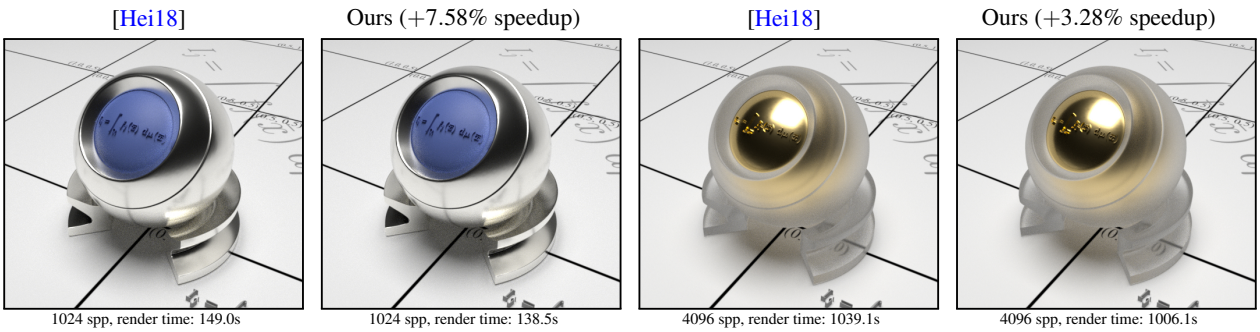

Figure 1: *We introduce a novel importance-sampling algorithm for GGX microfacet BSDFs. Our algorithm offers the same variance reduction as the state-of-the-art method [Hei18] but with lower computational overhead. This translates into slight but systematic performance gains for the rendering of, e.g., (left) rough conductors and (right) rough dielectrics. Scene credit: Yasutoshi Mori for the PBRT-v4 renderer.*

Abstract

Importance sampling the distribution of visible GGX normals requires sampling those of a hemisphere. In this work, we introduce a novel method for sampling such visible normals. Our method builds upon the insight that a hemispherical mirror reflects parallel light rays uniformly within a solid angle shaped as a spherical cap. This spherical cap has the same apex as the hemispherical mirror, and its aperture given by the angle formed by the orientation of that apex and the direction of incident light rays. Based on this insight, we sample GGX visible normals as halfway vectors between a given incident direction and directions drawn from its associated spherical cap. Our resulting implementation is even simpler than that of Heitz and leads to systematic speed-ups in our benchmarks.

CCS Concepts

• ***Computing methodologies*** → ***Reflectance modeling; Ray tracing;***

1. Introduction

The GGX microfacet BRDF is the most successful reflectance model in modern computer graphics history: Over the last decade, it has been implemented in many realtime game engines including (but not restricted to) Frostbite [HMD*14], Unreal [MHM*13], and Unity [AMG*18], as well as in offline path-tracers such as those of Walt Disney Animation Studios [MHH*12], Pixar [MHM*13], DreamWorks Animation [HMC*17], and Sony Pictures Imageworks [HMC*17].

In this work, we introduce a novel importance sampling algorithm for the GGX microfacet BRDF. Our algorithm offers the same variance reduction as state-of-the-art methods albeit at faster execution times as shown in Figure 1. Furthermore, it is straightforward to integrate within an existing implementation. It can thus immediately benefit to a large number of physically based renderers including those just mentioned.

Our algorithm effectively offers an alternative way to sample the *GGX distribution of visible normals (VNDF)* as described by Heitz [Hei18]—this particular method lies foundation for our own algorithm and so we explain it in detail in Section 2. We arrived at our alternative method by linking visible GGX normals to directions enclosed within solid angles shaped as spherical caps. We describe this link and our contributions throughout the remainder of this article as follows:

- In Section 3, we link the GGX VNDF to spherical caps and leverage this link to derive our novel importance sampling algorithm.
- In Section 4, we validate and benchmark our algorithm against that of Heitz both on the CPU and the GPU.

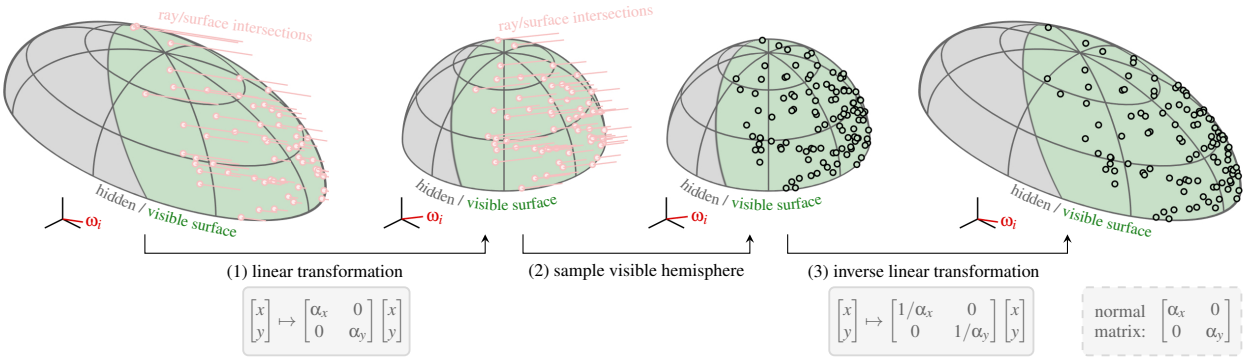

Figure 2: *GGX VNDF sampling overview.*

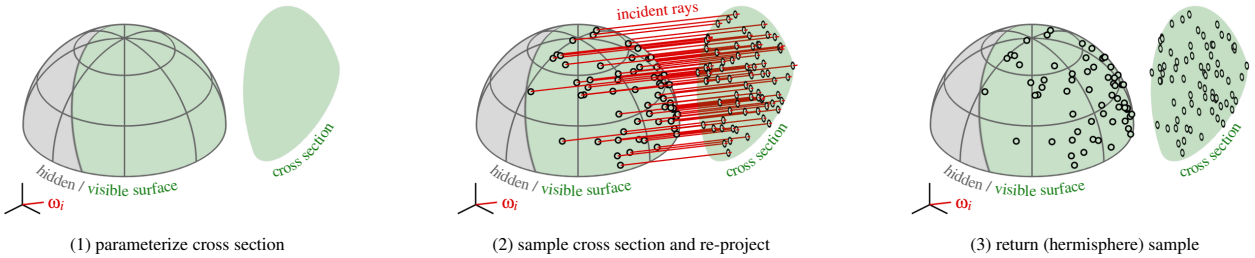

Figure 3: *Sampling the visible hemisphere using its cross section.*

2. Background on Sampling GGX Visible Normals

In this section, we provide the preliminary background suitable for deriving a state-of-the-art GGX VNDF sampler. Note that we refrain from providing more general background on the GGX microfacet BRDF here as this is largely unnessecary for our derivations. We refer the interested reader to Appendix A for such supplementary information.

Definition The GGX VNDF refers to the distribution of normals that lie at the surface formed by the intersection of parallel rays and a truncated ellipsoid [Hei18]. This intersected surface is shaded in green in Figure 2 and varies according to the direction of the parallel rays as well as the shape of the ellipsoid. More formally, the VNDF is parameterized by an incident direction $\omega_i \in \mathcal{S}^2$ that, by convention, points towards the opposite direction of the incident rays, as well as two scaling parameters $\alpha_x > 0$ and $\alpha_y > 0$ such that the matrices

$$M = \begin{bmatrix} \frac{1}{\alpha_x} & 0 & 0 \\ 0 & \frac{1}{\alpha_y} & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \text{and} \quad M^{-1} = \begin{bmatrix} \alpha_x & 0 & 0 \\ 0 & \alpha_y & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

respectively map a unit hemisphere to the ellipsoid and vice versa. Note that this parameterization implicitely assumes that the ellipsoid is formed by stretching the unit hemisphere centered at the origin and lying on the $z = 0$ plane. Next, we describe how to sample this distribution.

Importance Sampling Sampling the GGX VNDF according to the method of Heitz [Hei18] refers to mapping two uniform random numbers $u_1, u_2 \in [0, 1)$ to a normal ω_m. Intuitively, this mapping selects a ray that intersects the truncated ellipsoid and returns the normal lying at the intersection point. In practise, the GGX VNDF sampler leverages two key components. First, an invariance to linear transformations, which allows to systematically remap the intersection configuration to that of a hemisphere. Second, a routine to sample the visible normals of a hemisphere. The sampling algorithm then consists of the three following steps, which are illustrated in Figure 2 (see also Listing 1 for a GLSL implementation):

(1) Warp the entire space using the inverse matrix M^{-1}. This maps the ellipsoid to the unit hemisphere and the incident direction as follows

$$\omega_i \mapsto \frac{M^{-1}\,\omega_i}{\|M^{-1}\,\omega_i\|}.$$

(2) Intersect the unit hemisphere based on the new incident direction and record the position of the intersection points. The normal associated to each intersection point is equal to its position. In order to sample such intersection points, Heitz derives an algorithm based on the parameterization of the cross section of the hemisphere. We illustrate this algorithm in Figure 3 and provide a GLSL implementation in Listing 2. Note that this particular algorithm is the one we improve in this work.

(3) Transform each intersection point back to the ellipsoid configuration using the matrix M. The normals must be warped by the inverse transpose M^{-T} and effectively sample the GGX VNDF.

```glsl
vec3 SampleVndf_GGX(vec2 u, vec3 wi, vec2 alpha)
{
    // warp to the hemisphere configuration
    vec3 wiStd = normalize(vec3(wi.xy * alpha, wi.z));
    // sample the hemisphere (see implementation 2 or 3)
    vec3 wmStd = SampleVndf_Hemisphere(u, wiStd);
    // warp back to the ellipsoid configuration
    vec3 wm = normalize(vec3(wi.xy * alpha, wi.z));
    // return final normal
    return wm;
}
```

Listing 1: *General GGX VNDF sampling routine.*

```glsl
// Sampling the visible hemisphere using its cross section
vec3 SampleVndf_Hemisphere(vec2 u, vec3 wi)
{
    // orthonormal basis (with special case if cross product is 0)
    float tmp = wi.x * wi.x + wi.y * wi.y;
    vec3 w1 = tmp > 0.0f ? vec3(-wi.y, wi.x, 0) * inversesqrt(tmp)
                         : vec3(1, 0, 0);
    vec3 w2 = cross(wi, w1);
    // parameterization of the cross section
    float phi = 2.0f * M_PI * u.x;
    float r = sqrt(u.y);
    float t1 = r * cos(phi);
    float t2 = r * sin(phi);
    float s = (1.0f + wi.z) / 2.0f;
    t2 = (1.0f - s) * sqrt(1.0f - t1 * t1) + s * t2;
    float ti = sqrt(max(1.0f - t1 * t1 - t2 * t2, 0.0f));
    // reprojection onto hemisphere
    vec3 wm = t1 * w1 + t2 * w2 + ti * wi;
    // return hemispherical sample
    return wm;
}
```

Listing 2: *Sampling routine for the hemisphere. This is a refactored version of the code distributed by Heitz [Hei18].*

Probability Distribution Function The probability distribution function (PDF) of the GGX VNDF can be written as the following linearly-transformed spherical distribution

$$D_{\text{vis}}(\omega_m,\omega_i) = \underbrace{D_{\text{vis,std}}\left(\frac{M^T\omega_m}{\|M^T\omega_m\|},\frac{M^{-1}\omega_i}{\|M^{-1}\omega_i\|}\right)}_{\text{hemisphere VNDF}}\underbrace{\frac{|\det M^T|}{\|M^T\omega_m\|^3}}_{\text{Jacobian}}. \quad (1)$$

Here, the Jacobian is that induced by the linear transformation we introduced in the previous paragraph [HDHN16], and the term $D_{\text{vis,std}}$ refers to the VNDF of the unit hemisphere. This term is defined as [Hd14, Hei14]

$$D_{\text{vis,std}}(\omega_m,\omega_i) = \frac{\max(\omega_m\cdot\omega_i,0)\,D_{\text{std}}(\omega_m)}{\sigma_{\text{std}}(\omega_i)}, \quad (2)$$

$$D_{\text{std}}(\omega_m) = \begin{cases}\frac{1}{\pi} & \text{if } z_m>0,\\ 0 & \text{otherwise,}\end{cases} \quad (3)$$

$$\sigma_{\text{std}}(\omega_i) := \int_{\mathcal{S}^2}\max(\omega_m\cdot\omega_i,0)\,D_{\text{std}}(\omega_m)\,d\omega_m = \frac{1+z_i}{2}, \quad (4)$$

where D_{std} denotes the normal distribution function (NDF) of the hemisphere [WMLT07], and σ_{std} normalizes the PDF by measuring the cross-sectional area of the hemisphere shown in Figure 3. Note that Heitz needs to compute this area for his VNDF sampling algorithm (see line 14 in Listing 2).

3. Sampling GGX Visible Normals with Spherical Caps

In this section, we introduce our alternative method to sample the GGX VNDF. Our method differs from that of Heitz only in the way we sample the visible normals of the unit hemisphere. Our resulting GLSL implementation is shown in Listing 3. As can be seen from a direct comparison with the implementation of Heitz in Listing 2, our code is considerably simpler. We emphasize that this simplification is not due to aggressive optimizations of the original code of Heitz but to a novel insight that leads to an entirely different implementation. We share this insight and detail the implementation it leads to in the remainder of this section.

```glsl
// Sampling the visible hemisphere as half vectors (our method)
vec3 SampleVndf_Hemisphere(vec2 u, vec3 wi)
{
   // sample a spherical cap in (-wi.z, 1]
   float phi = 2.0f * M_PI * u.x;
   float z = fma((1.0f - u.y), (1.0f + wi.z), -wi.z);
   float sinTheta = sqrt(clamp(1.0f - z * z, 0.0f, 1.0f));
   float x = sinTheta * cos(phi);
   float y = sinTheta * sin(phi);
   vec3 c = vec3(x, y, z);
   // compute halfway direction;
   vec3 h = c + wi;
   // return without normalization (as this is done later)
   return h;
}
```

Listing 3: *Our novel sampling routine for the hemisphere.*

3.1. Key Insight: The Specular Reflections of a Hemisphere are Distributed as Spherical Caps

We make the key observation that if the hemisphere used for sampling the GGX VNDF acted as a perfect mirror, it would reflect parallel rays uniformly within a spherical cap. We illustrate this property in Figure 4 for two specific direction of incidence—notice how the direction of incidence only affects the height of the cap that shapes the directional distribution of reflected rays. Based on this insight, we devise our sampling algorithm using this spherical cap rather than the cross section of the hemisphere as done by Heitz. Before diving into the details of our algorithm, we first formally prove that our insight is mathematically sound. This is the focus of the next subsection.

3.2. Mathematical Proof

Let us consider that the hemisphere used for sampling the GGX VNDF acts as a perfect mirror, i.e., it deviates *incident* ray directions ω_i towards *outgoing* ray directions ω_o according to the law of specular reflection

$$\omega_o = 2(\omega_m\cdot\omega_i)\omega_m - \omega_i. \quad (5)$$

In order to derive the distribution of these outgoing ray directions, we first need to invert Equation (5) to determine the location at which the hemisphere deflects ω_i towards ω_o. This location corresponds to the point on the hemisphere whose normal ω_m is aligned with the half-vector

$$\omega_h = \frac{\omega_i+\omega_o}{\|\omega_i+\omega_o\|}. \quad (6)$$

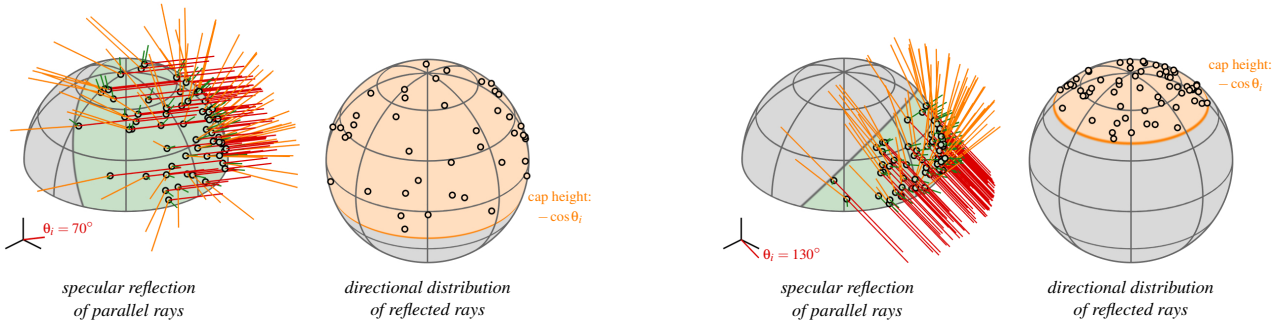

Figure 4: *Main insight: a hemispherical mirror reflects parallel light rays towards directions enclosed within a spherical cap.*

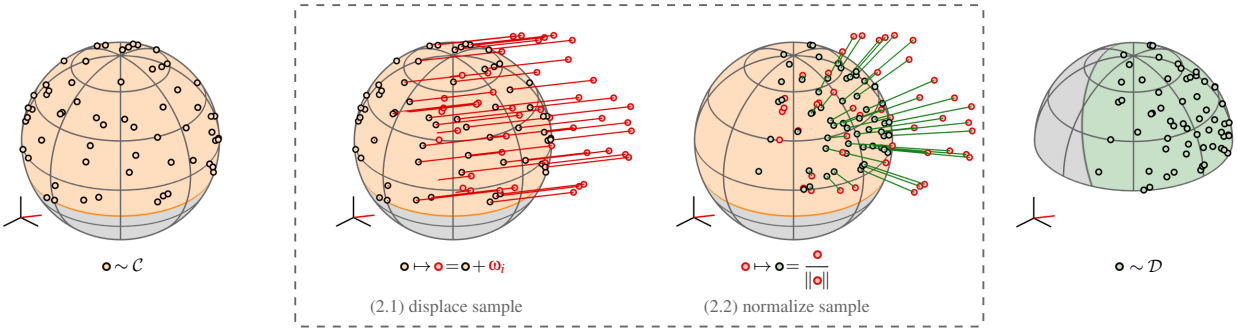

Figure 5: *Sampling the visible hemisphere using spherical caps (our method).*

Thanks to Equation (6), we can relate the density f_p of outgoing ray directions oriented within an infinitesimal solid angle $d\omega_o$ to the probability that the incident rays will intersect the infinitesimal surface $d\omega_h$:

$$f_p(\omega_o, \omega_i)\, d\omega_o = D_{\text{vis,std}}(\omega_h, \omega_i)\, d\omega_h.$$

Consequently, we get

$$f_p(\omega_o, \omega_i) = D_{\text{vis,std}}(\omega_h, \omega_i) \left\| \frac{d\omega_h}{d\omega_o} \right\|. \tag{7}$$

Now, to finalize our proof, we show that Equation (7) reduces to a uniform density over a domain bounded by a spherical cap. We first use the fact that the reflection Jacobian satisfies [WMLT07]

$$\left\| \frac{d\omega_h}{d\omega_o} \right\| = \frac{1}{4\,|\omega_o \cdot \omega_h|}. \tag{8}$$

Then, plugging Equations (2) and (8) into Equation (7) and using the fact that, by definition, $(\omega_o \cdot \omega_h) = (\omega_i \cdot \omega_h) \geq 0$, we get

$$f_p(\omega_o, \omega_i) = \frac{D_{\text{std}}(\omega_h)}{4\,\sigma_{\text{std}}(\omega_i)}. \tag{9}$$

Finally, using Equations (3) and (4) we obtain the uniform density

$$f_p(\omega_o, \omega_i) = \begin{cases} \frac{1}{2\pi(1+z_i)} & \text{if } z_h > 0 \Rightarrow z_o > -z_i \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

Such a uniform density is enclosed within a spherical cap oriented upwards and cutoff at the $z = -z_i$ plane, thus concluding our proof. As a final note, we mention that the normalization constant in Equation (10) corresponds to the solid angle of the cap that bounds the distribution.

3.3. Sampling Algorithm

A corollary result of our insight is that the half-vectors we retrieve from a spherical cap cutoff at the $z = -z_i$ plane are necessarily distributed according to the normals of the hemisphere that are visible from direction ω_i. This corollary result lies at the core of our sampling algorithm, which is illustrated in Figure 5 and consists of the following steps:

(1) Sample a direction ω_o from a spherical cap with elevation $-z_i$.
(2) Compute the half-vector between the sample ω_o and ω_i.
(3) Return this half-vector as a visible normal.

We provide a GLSL implementation of our algorithm in Listing 3, which is meant to be used in conjunction with Listing 1. An important note here is that we purposely omit the normalization of the half-vector at line (14) of Listing 3 (this is equivalent to avoiding step (2.2) in Figure 5). We do this because we know that the direction is normalized later, specifically at line (8) of Listing 1. If one was to use Listing 3 in a stand-alone fashion, then the normalization would be necessary.

4. Results and Validation

In this section, we validate and benchmark our method against that of Heitz through several experiments that we detail below.

CPU Synthetic Benchmark We wrote a C++ program that calls Listing 1 within a loop and measured the median time it takes to run over 100 invocations. We ran our program on an Intel i7-13700K and measured a speed-up of 37.67% in favor of our method. This result is consistent across roughness values and incident directions.

GPU Synthetic Benchmark We wrote a DirectX 12 compute shader that calls Listing 1 64 times per lane and measured the median time taken for the shader to run over 100 invocations. We ran our program on an Intel Arc A770 and an NVIDIA RTX 2080 and measured a speed-up of respectively 52.96% and 39.25% in favor of our method. Again, this result is consistent across roughness values and incident directions.

Rendering Comparisons We validated that our method converges to the same results as the method of Heitz within state-of-the-art path-tracers. We provide an example of such a validation in Figure 1, where we show PBRT-v4 renderings of a scene based on two different GGX microfacet BSDFs. As expected, our sampling algorithm does indeed converge towards the same result as Heitz.

Rendering Profiling We also profiled the execution of a PBRT rendering for scenes of varying geometric complexity. For each scene, we set all materials to either rough conductors or rough dielectric. Under this setup we report the relative time spent for BSDF importance sampling in Table 1. As demonstrated by the reported numbers, our sampling scheme systematically reduces the relative time spent for BSDF importance sampling by at least a factor of two compared to that of Heitz. Despite this advantage in relative time, we mention that, depending on the scene, one may not necessarily observe significant boosts in rendering times as reported, e.g., in Figure 1. This is because path-traced rendering is usually largely bottlenecked by other computations such as, e.g., ray intersection queries and/or memory accesses. Note that this is already visible in Table 1 as the relative time spent for BSDF sampling is systematically lower than 4% for all the scenes we tested.

	relative time		
scene name	[Hei18]	(ours)	speed-up
pbrt-book	1.34%	0.54%	×2.48
lte-orb (conductor)	3.71%	1.49%	×2.49
lte-orb (dielectric)	1.89%	0.74%	×2.55
sportscar	0.48%	0.20%	×2.40

Table 1: *Relative time spent for BSDF importance sampling during a PBRT rendering. The scenes are available at the PBRT repository:* `https://github.com/mmp/pbrt-v4-scenes`.

5. Conclusion

We introduced a novel importance sampling algorithm for GGX microfacet BSDFs. Our method is easy to implement and offers systematic speed-ups over that of Heitz.

Appendix A: Background on the GGX Microfacet BRDF

The GGX microfacet BRDF refers to an analytic *bidirectional reflectance distribution function* (BRDF), which results from microfacet theory. We provide here some background and the mathematical expressions suitable for evaluating and importance sampling such a BRDF via the GGX VNDF introduced in Section 2.

Microfacet Theory Intuitively, microfacet theory describes surface reflectance as the result of interactions between incident light and microscopic mirrors (which compose the actual surface) that deviate this incident light. If we only consider single-scattering interactions, then the amount of light a surface reflects towards any given direction becomes proportional to the probability of intersecting a mirror suitably oriented to produce such a reflection. This relation is formally expressed by the Cook-Torrance equation [CT82]

$$f_r(\omega_i,\omega_o) = \frac{F(\omega_h\cdot\omega_o)\,G_2(\omega_h,\omega_i,\omega_o)\,D(\omega_h)}{4\cos\theta_i\cos\theta_o}, \tag{11}$$

where F, G_2, and D respectively denote the Fresnel term, the microfacet shadowing-and-masking function and the microfacet normal distribution function (NDF) [WMLT07].

Derivation of the GGX BRDF The GGX microfacet BRDF is a specialized form of Equation (11), which follows from the two following hypothesis. The first hypothesis is that the microfacets are oriented according to the NDF of a truncated ellipsoid [TR75, Hei18]. In the spirit of Section 2, we write this NDF as a function of that of a hemisphere. This leads to the following expression

$$D(\omega_m) = \underbrace{D_{\text{std}}\left(\frac{M^T\omega_m}{\|M^T\omega_m\|}\right)}_{\substack{\text{hemisphere NDF}\\ \text{see Equation (3)}}}\underbrace{\frac{|\det M^T|}{\|M^T\omega_m\|^4}}_{\substack{\text{Jacobian}\\ \text{[AKDW22]}}}, \tag{12}$$

where M is the matrix that maps the hemisphere to the GGX ellipsoids as defined in Section 2. The second hypothesis is that the microfacets are subject to the Smith shadowing model [Smi67]. This leads to the following shadowing term [Hei14, DHI*15]

$$G_2(\omega_m,\omega_i,\omega_o) = \frac{G_i\,G_o}{G_i+G_o-G_i\,G_o}, \tag{13}$$

$$G_k = G_1\left(\omega_m,\omega_k\right), \tag{14}$$

$$G_1(\omega_m,\omega_k) = \frac{\chi^+(\omega_m\cdot\omega_k)\,z_k}{\int_{\mathcal{S}^2}\max(\omega_m\cdot\omega_k,0)\,D(\omega_m)\,\mathrm{d}\omega_m}, \tag{15}$$

where G_1 and χ^+ respectively denote the Smith monostatic shadowing term and the Heaviside step function. Again, we provide an analytic expression for Equation (15) based on the shadowing term of the hemispere [AKDW22]

$$G_1(\omega_m,\omega_i) = G_{1,\text{std}}\left(\frac{M^T\omega_m}{\|M^T\omega_m\|},\frac{M^{-1}\omega_i}{\|M^{-1}\omega_i\|}\right) \tag{16}$$

$$G_{1,\text{std}} = \frac{\chi^+(\omega_i\cdot\omega_m)\,z_i}{\sigma_{\text{std}}(\omega_i)}, \tag{17}$$

where σ_{std} is the cross-sectional area of the hemisphere defined in Equation (4) and illustrated in Figure 3.

Importance Sampling Using the GGX VNDF The GGX microfacet BRDF typically occurs in the direct illumination equation

$$L(\omega_a) = \int_{\mathcal{S}^2} L(\omega_b)\, f_r(\omega_b, \omega_a)\, |\cos\theta_b|\, \mathrm{d}\omega_b, \tag{18}$$

where L denotes incident radiance. The GGX VNDF provides a way to solve this equation through the Monte Carlo estimator

$$L(\omega_a) \approx I(\omega_a) = \frac{1}{N} \sum_{j=0}^{N-1} \frac{L(\omega_{b_j})\, f_r(\omega_{b_j}, \omega_a)\, |\cos\theta_{b_j}|}{\mathrm{PDF}(\omega_{b_j}, \omega_a)}, \tag{19}$$

where ω_{b_j} denotes the j-th sample of the estimator. This sample is produced in the two following steps:

(1) Sample the GGX VNDF from direction ω_a using Listing 1. This produces a normal ω_{m_j} whose PDF is given in Equation (1).
(2) Use this normal to reflect the incident direction ω_a according to the reflection equation $\omega_{b_j} = 2(\omega_{m_j} \cdot \omega_a)\omega_{m_j} - \omega_a$.

The resulting PDF is given by the VNDF weighted by the Jacobian of the reflection operator as provided in Equation (8):

$$\mathrm{PDF}(\omega_{b_j}, \omega_a) = \frac{G_1(\omega_{m_j}, \omega_a)\, D(\omega_{m_j})}{4\cos\theta_a}. \tag{20}$$

By sampling the GGX VNDF in this way, Equation (19) effectively provides an estimator with state-of-the-art variance [Hd14].

References

[AKDW22] ATANASOV A., KOYLAZOV V., DIMOV R., WILKIE A.: Microsurface transformations. *Computer Graphics Forum 41*, 4 (2022), 105–116. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.14590, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.14590, doi:https://doi.org/10.1111/cgf.14590. 5

[AMG*18] ABADIE G., MCAULEY S., GOLUBEV E., HILL S., LAGARDE S.: Advances in real-time rendering in games. In *ACM SIGGRAPH 2018 Courses* (New York, NY, USA, 2018), SIGGRAPH '18, Association for Computing Machinery. URL: https://doi.org/10.1145/3214834.3264541, doi:10.1145/3214834.3264541. 1

[CT82] COOK R. L., TORRANCE K. E.: A reflectance model for computer graphics. *ACM Trans. Graph. 1*, 1 (jan 1982), 7–24. URL: https://doi.org/10.1145/357290.357293, doi:10.1145/357290.357293. 5

[DHI*15] DUPUY J., HEITZ E., IEHL J.-C., POULIN P., OSTROMOUKHOV V.: Extracting microfacet-based brdf parameters from arbitrary materials with power iterations. In *Computer Graphics Forum* (2015), vol. 34, Wiley Online Library, pp. 21–30. 5

[Hd14] HEITZ E., D'EON E.: Importance sampling microfacet-based bsdfs using the distribution of visible normals. *Computer Graphics Forum 33*, 4 (2014), 103–112. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.12417, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.12417, doi:https://doi.org/10.1111/cgf.12417. 3, 6

[HDHN16] HEITZ E., DUPUY J., HILL S., NEUBELT D.: Real-time polygonal-light shading with linearly transformed cosines. *ACM Trans. Graph. 35*, 4 (jul 2016). URL: https://doi.org/10.1145/2897824.2925895, doi:10.1145/2897824.2925895. 3

[Hei14] HEITZ E.: Understanding the masking-shadowing function in microfacet-based brdfs. *Journal of Computer Graphics Techniques (JCGT) 3*, 2 (June 2014), 48–107. 3, 5

[Hei18] HEITZ E.: Sampling the ggx distribution of visible normals. *Journal of Computer Graphics Techniques (JCGT) 7*, 4 (November 2018), 1–13. URL: http://jcgt.org/published/0007/04/01/. 1, 2, 3, 5

[HMC*17] HILL S., MCAULEY S., CONTY A., DROBOT M., HEITZ E., HERY C., KULLA C., LANZ J., LING J., WALSTER N., XIE F., MICCIULLA A., VILLEMIN R.: Physically based shading in theory and practice. In *ACM SIGGRAPH 2017 Courses* (New York, NY, USA, 2017), SIGGRAPH '17, Association for Computing Machinery. URL: https://doi.org/10.1145/3084873.3084893, doi:10.1145/3084873.3084893. 1

[HMD*14] HILL S., MCAULEY S., DUPUY J., GOTANDA Y., HEITZ E., HOFFMAN N., LAGARDE S., LANGLANDS A., MEGIBBEN I., RAYANI F., DE ROUSIERS C.: Physically based shading in theory and practice. In *ACM SIGGRAPH 2014 Courses* (New York, NY, USA, 2014), SIGGRAPH '14, Association for Computing Machinery. URL: https://doi.org/10.1145/2614028.2615431, doi:10.1145/2614028.2615431. 1

[MHH*12] MCAULEY S., HILL S., HOFFMAN N., GOTANDA Y., SMITS B., BURLEY B., MARTINEZ A.: Practical physically-based shading in film and game production. In *SIGGRAPH 2012 Courses* (2012), ACM, pp. 10:1–7. URL: http://doi.acm.org/10.1145/2343483.2343493, doi:10.1145/2343483.2343493. 1

[MHM*13] MCAULEY S., HILL S., MARTINEZ A., VILLEMIN R., PETTINEO M., LAZAROV D., NEUBELT D., KARIS B., HERY C., HOFFMAN N., ZAP ANDERSSON H.: Physically based shading in theory and practice. In *SIGGRAPH 2013 Courses* (2013), ACM, pp. 22:1–8. URL: http://doi.acm.org/10.1145/2504435.2504457, doi:10.1145/2504435.2504457. 1

[Smi67] SMITH B.: Geometrical shadowing of a random rough surface. *IEEE Trans. on Antennas and Propagation 15*, 5 (1967), 668–671. doi:10.1109/TAP.1967.1138991. 5

[TR75] TROWBRIDGE T. S., REITZ K. P.: Average irregularity representation of a rough surface for ray reflection. *J. Opt. Soc. Am. 65*, 5 (May 1975), 531–536. URL: http://www.opticsinfobase.org/abstract.cfm?URI=josa-65-5-531, doi:10.1364/JOSA.65.000531. 5

[WMLT07] WALTER B., MARSCHNER S. R., LI H., TORRANCE K. E.: Microfacet models for refraction through rough surfaces. In *Proc. Eurographics Symposium on Rendering* (2007), EGSR'07, pp. 195–206. URL: http://dx.doi.org/10.2312/EGWR/EGSR07/195-206, doi:10.2312/EGWR/EGSR07/195-206. 3, 4, 5