

Sampling the GGX Distribution of Visible Normals

Eric Heitz
Unity Technologies

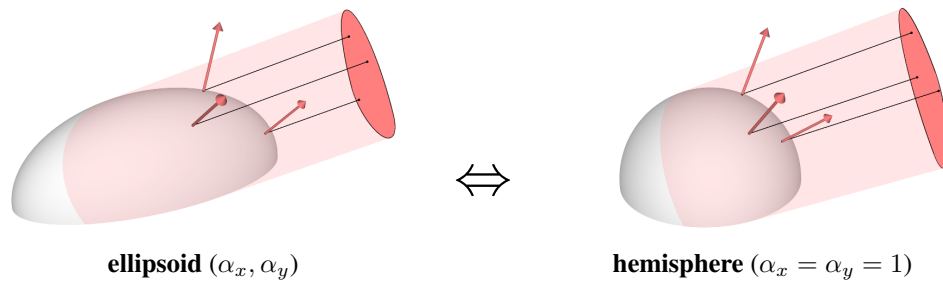


Figure 1. Sampling the GGX distribution of visible normals (VNDF) is equivalent to sampling the projected area of an ellipsoid, which can be mapped to sampling the projected area of a hemisphere.

Abstract

Importance sampling microfacet bidirectional scattering distribution functions (BSDFs) using their distribution of visible normals (VNDF) yields significant variance reduction in Monte Carlo rendering. In this article, we describe an efficient and exact sampling routine for the VNDF of the GGX microfacet distribution. This routine leverages the property that GGX is the distribution of normals of a truncated ellipsoid, and sampling the GGX VNDF is equivalent to sampling the 2D projection of this truncated ellipsoid. To do that, we simplify the problem by using the linear transformation that maps the truncated ellipsoid to a hemisphere. Since linear transformations preserve the uniformity of projected areas, sampling in the hemisphere configuration and transforming the samples back to the ellipsoid configuration yields valid samples from the GGX VNDF.

1. Introduction and Previous Work

1.1. The GGX Distribution

The GGX distribution is the normal distribution function (NDF) of an ellipsoid, i.e., it measures the density of a given normal orientation on the surface of the ellipsoid. Formally, it is a distribution D such that if Ω is a solid-angle domain, then $\int_{\Omega} D(\omega) \partial\omega$ is the area of the surface of the ellipsoid whose normals are oriented within Ω . The idea

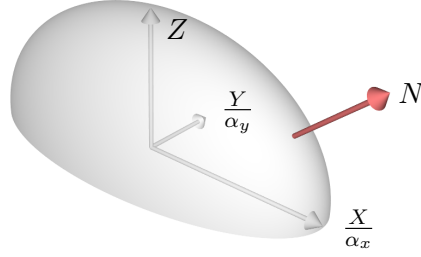


Figure 2. The GGX distribution is the distribution of normals of an ellipsoid.

of representing a distribution of normals using an ellipsoid shape was invented independently multiple times. To our knowledge, it was first introduced by Trowbridge and Reitz [1975] in the physics literature. In computer graphics, Neyret derived an equivalent distribution to represent volumetric materials [Neyret 1995; Neyret 1998]. Later, the same distribution was derived again by Walter et al. [2007] to model the scattering of glass and named GGX, which stands for “ground glass unknown.” Currently, this microfacet distribution is one of the most widely used in the rendering industry.

The GGX distribution uses only the upper part of the ellipsoid according to a local frame. As shown in Figure 2, the ellipsoid is truncated to the upper hemisphere in direction $Z = (0, 0, 1)$, and it is described by two scaling (or roughness) parameters α_x and α_y that represent the inverse lengths of the principal axes of the ellipsoid on the $X = (1, 0, 0)$ and $Y = (0, 1, 0)$ directions. For a given normal defined in this frame by $N = (x_n, y_n, z_n)$, the GGX distribution is given by

$$D(N) = \frac{1}{\pi \alpha_x \alpha_y \left(\frac{x_n^2}{\alpha_x^2} + \frac{y_n^2}{\alpha_y^2} + z_n^2 \right)^2}. \quad (1)$$

In microfacet BSDFs, the GGX distribution is usually used with the Smith shadowing model. The Smith shadowing function associated with the GGX distribution was first introduced by Walter et al. [2007] for isotropic distributions ($\alpha_x = \alpha_y$) and generalized to anisotropic GGX distributions by Heitz [2014]. For a given view direction $V = (x_v, y_v, z_v)$, the Smith anisotropic GGX shadowing function is

$$G_1(V) = \frac{1}{1 + \Lambda(V)}, \quad \text{with } \Lambda(V) = \frac{-1 + \sqrt{1 + \frac{\alpha_x^2 x_v^2 + \alpha_y^2 y_v^2}{z_v^2}}}{2}. \quad (2)$$

1.2. Importance Sampling Using the VNDF

Using microfacet BSDFs in Monte Carlo renderers requires importance-sampling techniques. Historically, the classic approach consists of sampling microfacets using the NDF, but Heitz and d’Eon [2014] showed that using the distribution of visible

normals (VNDF)

$$D_V(N) = \frac{G_1(V) \max(0, V \cdot N) D(N)}{V \cdot Z} \quad (3)$$

instead of the NDF provides significant variance reduction. This is because the VNDF contains visibility terms that appear in the BSDF expression that cancel out in the importance-sampling weights, thus reducing the variance (see Appendix B).

2. Previous Work

VNDF sampling. Heitz and d'Eon [2014] provide analytic solutions for sampling the VNDFs of the Beckmann and GGX distributions in their supplemental material. To facilitate the VNDF sampling, their algorithms map the configuration to a simpler configuration of unit roughness ($\alpha_x = \alpha_y = 1$). Unfortunately, even with unit roughness, their solution for GGX is only approximate because it requires a fitted curve: the derivations are made in slope space and the conditional inverse CDF of GGX does not have a closed form in this space.

Sampling the projected area of an ellipsoid. The key observation used in this article is that sampling the GGX distribution can be done exactly by sampling the projected area of the associated ellipsoid. The idea of sampling the projected area of the ellipsoid was introduced by Neyret [1998] and Heitz et al. [2015] showed that it was equivalent to sampling the VNDF of the ellipsoid's NDF. However, they performed this operation for a complete ellipsoid (also called SGGX distribution), while in the case of the GGX distribution the ellipsoid is truncated to the upper hemisphere.

Sampling the projected area of a truncated ellipsoid. In this article, we use the sampling algorithm introduced in the supplemental material [Walter et al. 2015] of Dong et al. [2015]. They introduce a truncated-ellipsoid NDF that is a generalized GGX distribution whose associated ellipsoid is transformed not only by linear scaling but also by skewness factors (the matrix A is non-diagonal). The idea behind their sampling algorithm is to keep Heitz and d'Eon's unit-roughness transformation that transforms the truncated ellipsoid into a hemisphere, as we explain in Section 3. With this transformation, the problem boils down to sampling the projected area of a hemisphere, and they provide an area-preserving parameterization to do this, which we explain in Section 4. In summary, the purpose of this article is not to introduce new ideas, as they can already be found in substance in Walter and Dong's supplemental material. Our main contribution is to bring these ideas together into a simple and well-documented routine for sampling the GGX VNDF.

Previous version of this article. An early version of this work was made available as a non-peer-reviewed technical report [Heitz 2017]. However, this previous version uses

another parameterization that cannot be used with view directions located in the lower hemisphere ($V \cdot Z < 0$). This is never the case with classic microfacet BSDF models, but there are special cases where this occurs, for instance Smith multiple-scattering BSDFs model incident rays that can be located in the lower hemisphere after their first bounce on the microsurface [Heitz et al. 2016]. In this article, we replaced the previous parameterization by the one of Walter et al. [2015] that is not subject to this limitation and can thus be used with multiple-scattering BSDFs.

3. Ellipsoid-hemisphere Transformation

In this section, we explain the transformation that maps an ellipsoid configuration to a hemisphere configuration, as shown in Figure 3.

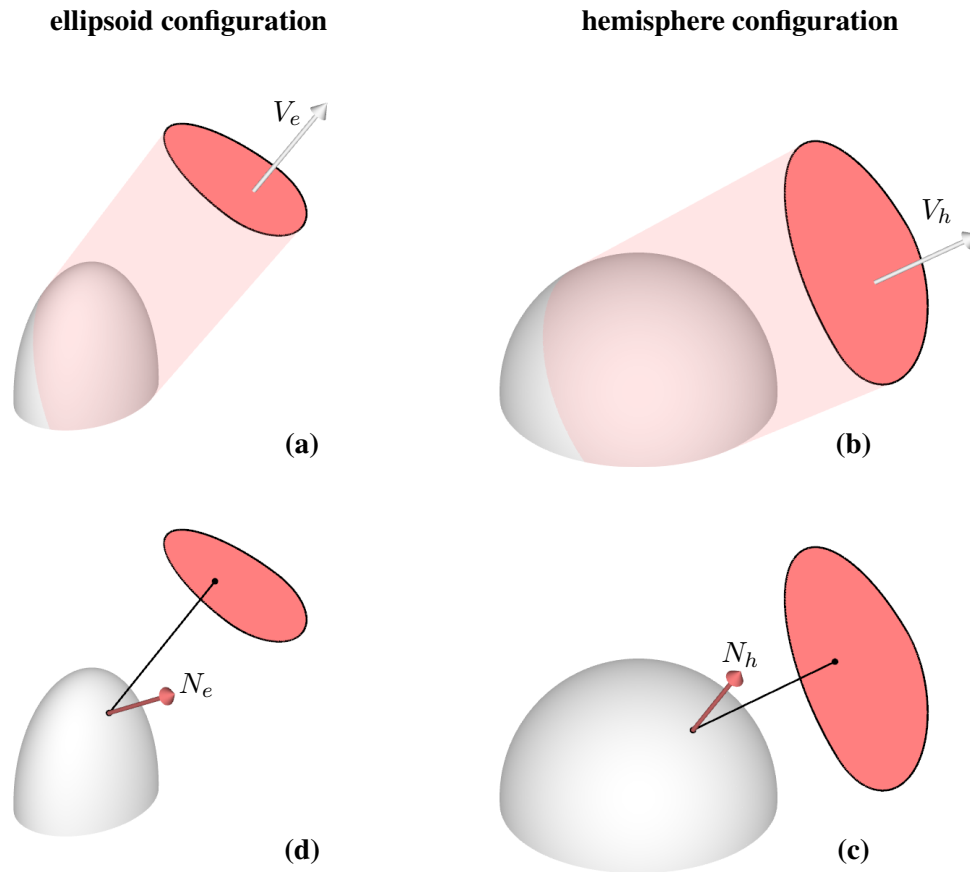


Figure 3. Ellipsoid-hemisphere transformation. We start by transforming the view vector V_e of the ellipsoid configuration (a) to a view vector V_h in the hemisphere configuration (b). In the hemisphere configuration, we sample a normal N_h by sampling the projected area of the hemisphere (c). By transforming this normal back to the ellipsoid configuration, we obtain a normal N_e sampled from the distribution of visible normals of the ellipsoid (d).

3.1. Initial Ellipsoid Configuration

The initial ellipsoid configuration is shown in Figure 3(a). V_e denotes the view direction, and the ellipsoid is given by the roughness parameters α_x and α_y . We also consider two uniform random numbers U_1 and U_2 that we are going to use in the sampling procedure.

```
vec3 sampleGGXVNDF(vec3 Ve, float alpha_x, float alpha_y, float U1, float U2)
{
    ...
}
```

3.2. Transforming the View Direction to the Hemisphere Configuration

The linear transformation that maps the ellipsoid of roughness α_x and α_y to the hemisphere is represented by a 3×3 matrix

$$A = \begin{bmatrix} \alpha_x & 0 & 0 \\ 0 & \alpha_y & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (4)$$

To move from Figure 3(a) to Figure 3(b), we compute a view vector in the hemisphere configuration V_h by transforming the view vector V_e in the ellipsoid configuration:

$$V_h = \frac{A V_e}{\|A V_e\|}. \quad (5)$$

```
// Section 3.2: transforming the view direction to the hemisphere configuration
vec3 Vh = normalize(vec3(alpha_x * Ve.x, alpha_y * Ve.y, Ve.z));
```

3.3. Sampling the Projected Area of the Hemisphere

In Figure 3(c), we sample a normal N_h by sampling the projected area of the hemisphere. This part of the algorithm is explained in Section 4.

3.4. Transforming the Normal Back to the Ellipsoid Configuration

To move from Figure 3(c) to Figure 3(d), we transform the normal N_h in the hemisphere configuration to obtain a normal in the ellipsoid configuration V_e . The geometric transformation is the inverse transformation as before, i.e., A^{-1} . However, since normals are not vectors but covectors, they are not transformed by A^{-1} but by its inverse transpose matrix, which is $(A^{-1})^{-T} = A$. The normal in the ellipsoid configuration is thus

$$N_e = \frac{A N_h}{\|A N_h\|}. \quad (6)$$

Note that in the code we clamp the z -component to 0 to prevent numerical errors.

```
// Section 3.4: transforming the normal back to the ellipsoid configuration
vec3 Ne = normalize(vec3(alpha_x * Nh.x, alpha_y * Nh.y, max(0.0, Nh.z)));
```

4. Sampling the Projected Area of a Hemisphere

In this section, we derive an area-preserving parameterization that we use to sample the projected area of the hemisphere.

4.1. Orthonormal Basis

We start by constructing an orthonormal basis (V_h, T_1, T_2) (see Figure 4), where T_1 is in the tangent plane orthogonal to $Z = (0, 0, 1)$:

$$T_1 = \frac{Z \times V_h}{\|Z \times V_h\|}, \quad (7)$$

$$T_2 = V_h \times T_1. \quad (8)$$

```
// Section 4.1: orthonormal basis
vec3 T1 = (Vh.z < 0.9999) ? normalize(cross(vec3(0, 0, 1), Vh)) : vec3(1, 0, 0);
vec3 T2 = cross(Vh, T1);
```

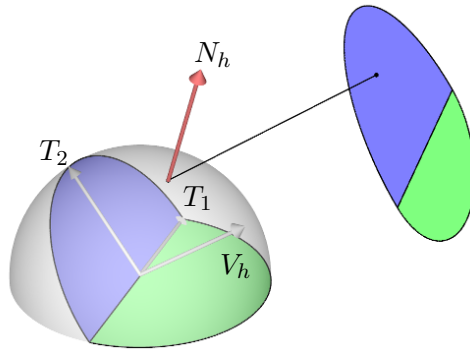


Figure 4. Orthonormal basis for sampling the projected area of the hemisphere.

4.2. Parameterization of the Projected Area

Shape of the projected area. Figure 5 shows the shape of the projected area of the hemisphere. It is the signed sum of the projected areas of the two half disks. The projected area of the half disk located in the tangent plane (in green) is proportional to $Z \cdot V_h = z_v$, and the projected area of the other half disk (in blue) is proportional to $V_h \cdot V_h = 1$. If the view direction is below the horizon ($V_h \cdot Z < 0$), the green disk partially masks the blue disk. In all configurations, the shape of the projected area is

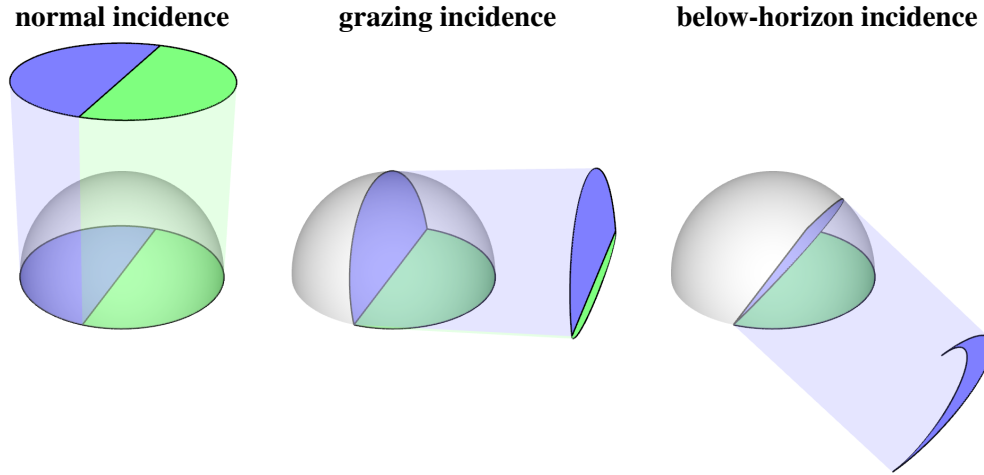


Figure 5. Shape of the projected area of the hemisphere. The projected area of the hemisphere is the signed sum of the projected areas of the two half disks. If the view direction is below the horizon the green disk partially masks the blue disk.

thus a disk whose lower boundary is moving and producing a uniform vertical scaling such that the vertical segment $[-\sqrt{1-t_1^2}, +\sqrt{1-t_1^2}]$ of abscissa t_1 is uniformly remapped to the interval $[-(V_h \cdot Z) \sqrt{1-t_1^2}, +\sqrt{1-t_1^2}]$. The scaling factor of this remapping is $s = \frac{1+(V_h \cdot Z)}{2}$.

Parameterization of the projected area of the hemisphere. Figure 6 shows that a point (t_1, t_2) uniformly sampled in the unit disk using a polar parameterization,

$$(r, \phi) = \left(\sqrt{\mathcal{U}_1}, 2\pi \mathcal{U}_2 \right), \quad (9)$$

$$(t_1, t_2) = (r \cos \phi, r \sin \phi), \quad (10)$$

is mapped to a point (t_1, t'_2) uniformly sampled in the shape of the projected area by applying the vertical remapping:

$$t'_2 = (1-s) \sqrt{1-t_1^2} + s t_2. \quad (11)$$

```
// Section 4.2: parameterization of the projected area
float r = sqrt(U1);
float phi = 2.0 * M_PI * U2;
float t1 = r * cos(phi);
float t2 = r * sin(phi);
float s = 0.5 * (1.0 + Vh.z);
t2 = (1.0 - s)*sqrt(1.0 - t1*t1) + s*t2;
```

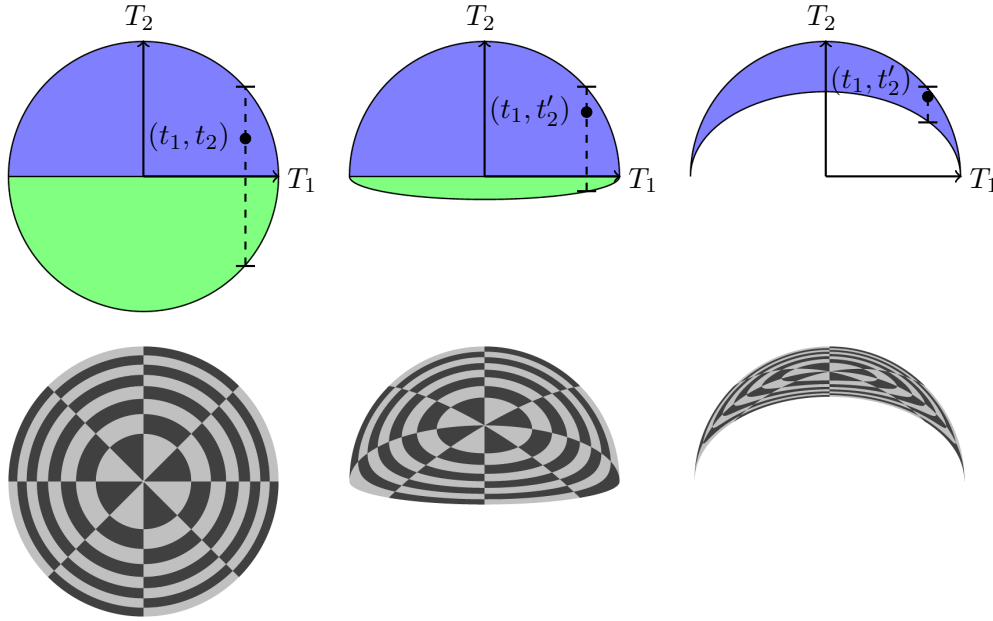


Figure 6. Parameterization of the projected area of the hemisphere. The shape of the projected area in all configurations is a disk whose lower boundary is moving and producing a uniform vertical scaling. Hence, a polar parameterization of the unit disk compressed by a uniform vertical scaling yields an area-preserving parameterization of the projected area.

4.3. Reprojection on the Hemisphere

Finally, we reproject the point $t_1 T_1 + t_2 T_2$ onto the hemisphere to obtain its normal. To do this, we compute the component v_h in direction V_h such that the point

$$P_h = t_1 T_1 + t_2 T_2 + v_h V_h \quad (12)$$

is normalized, i.e., we compute

$$v_h = \sqrt{1 - t_1^2 - t_2^2}. \quad (13)$$

Since the point P_h is on a hemisphere, the associated normal is $N_h = P_h$. Note that in our implementation we clamp $1 - t_1^2 - t_2^2$ to 0 in order to avoid numerical-precision errors.

```
// Section 4.3: reprojection onto hemisphere
vec3 Nh = t1*T1 + t2*T2 + sqrt(max(0.0, 1.0 - t1*t1 - t2*t2))*Vh;
```

5. Evaluation

Performance. We compared the performance of our final implementation from Listing 1 to the implementation provided by Heitz and d'Eon [2014] by generating 10 mil-

lion random samples on a Intel(R) Core(TM) i7-5960X. The generation takes 1.46s with our implementation and 2.31s with Heitz and d'Eon's implementation. Our implementation is 58% faster.

Visualizing the parameterization. To visualize the parameterizations used by the sampling algorithms, we use them to warp a unit-square checker and a point set in Figure 7. To ease the visualization, we display the results in the 2D Cartesian slope space by converting the 3D normal directions to 2D slope values: $(\tilde{x}, \tilde{y}) = \left(\frac{-x_n}{z_n}, \frac{-y_n}{z_n}\right)$. Since our algorithm and the one of Heitz and d'Eon [2014] use the same linear transformation to map the configuration to the unit-roughness or hemisphere configuration (where $\alpha_x = \alpha_y = 1$) and their difference is how they operate in this configuration, we display them in this configuration for varying view angles.

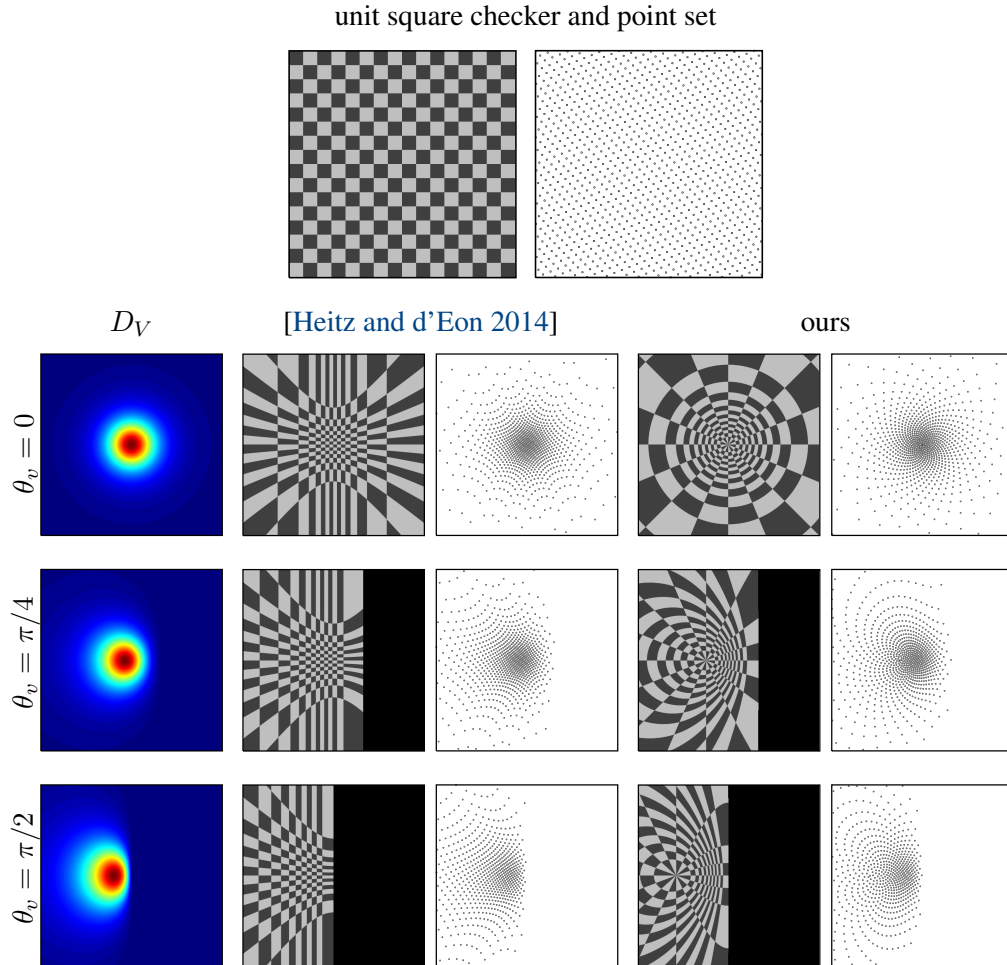


Figure 7. Visualizing the parameterization. We warp the unit-square checker and point set using the sampling algorithms. We compute the warping in the unit-roughness (or hemisphere) configuration and visualize it in slope space.

6. Conclusion

We have described an efficient implementation of a GGX VNDF sampling routine. In contrast to the routine of Heitz and d'Eon [2014] that used a fitted curve, this new routine is exact. Furthermore, it is about 58% faster according to our experiments and simpler to implement. Finally, it can be used with view directions located in the lower hemisphere, which makes it usable for sampling multiple-scattering BSDFs based on the GGX distribution [Heitz et al. 2016].

Acknowledgements

I thank Stephen Hill for motivating me to write this paper and for his helpful feedback.

A. Complete Implementation of the GGX VNDF Sampling Routine

```
// Input Ve: view direction
// Input alpha_x, alpha_y: roughness parameters
// Input U1, U2: uniform random numbers
// Output Ne: normal sampled with PDF D_Ve(Ne) = G1(Ve) * max(0, dot(Ve, Ne)) * D(Ne) / Ve.z
vec3 sampleGGXVNDf(vec3 Ve, float alpha_x, float alpha_y, float U1, float U2)
{
    // Section 3.2: transforming the view direction to the hemisphere configuration
    vec3 Vh = normalize(vec3(alpha_x * Ve.x, alpha_y * Ve.y, Ve.z));
    // Section 4.1: orthonormal basis
    vec3 T1 = (Vh.z < 0.9999) ? normalize(cross(vec3(0, 0, 1), Vh)) : vec3(1, 0, 0);
    vec3 T2 = cross(Vh, T1);
    // Section 4.2: parameterization of the projected area
    float r = sqrt(U1);
    float phi = 2.0 * M_PI * U2;
    float t1 = r * cos(phi);
    float t2 = r * sin(phi);
    float s = 0.5 * (1.0 + Vh.z);
    t2 = (1.0 - s) * sqrt(1.0 - t1*t1) + s*t2;
    // Section 4.3: reprojection onto hemisphere
    vec3 Nh = t1*T1 + t2*T2 + sqrt(max(0.0, 1.0 - t1*t1 - t2*t2))*Vh;
    // Section 3.4: transforming the normal back to the ellipsoid configuration
    vec3 Ne = normalize(vec3(alpha_x * Nh.x, alpha_y * Nh.y, std::max<float>(0.0, Nh.z)));
    return Ne;
}
```

Listing 1. Sampling the GGX VNDF: complete implementation.

B. Usage in a Monte Carlo Renderer

In this section, we briefly recall how to use the routine of Listing 1 to compute a Monte Carlo estimator of the direct illumination,

$$\bar{I} = \int_{\Omega} I(L) \rho(V, L) (L \cdot Z) d\omega_L, \quad (14)$$

where V is the view direction, $I(L)$ is the radiance arriving at the shading point from direction L , and $\rho(V, L)$ is a GGX microfacet BRDF.

GGX microfacet BRDF. Typically, a microfacet BRDF has the following expression:

$$\rho(V, L) = \frac{F(V, H) D(H) G_2(V, L)}{4 \cos \theta_V \cos \theta_L}, \quad (15)$$

where H is the half-vector between V and L , $F(V, H)$ is a Fresnel term, $D(H)$ is the GGX distribution, and $G_2(V, L)$ is the shadowing-and-masking function [Heitz 2014].

Importance sampling using the VNDF. The sampling routine of Listing 1 generates microfacet samples N_i whose PDF is the VNDF $D_V(N_i)$ of Equation (3). By reflecting the view direction, we obtain light samples L_i :

$$L_i = \text{reflect}(V, N_i), \quad (16)$$

whose PDF is the VNDF weighted by the Jacobian of the reflection operator:

$$\text{PDF}(L_i) = \frac{D_V(N_i)}{4 (V \cdot N_i)}. \quad (17)$$

Monte Carlo estimator of the direct illumination. With this importance sampling procedure, we obtain a stochastic estimator of Equation (14):

$$\bar{I} \approx \frac{1}{n} \sum_{i=1}^n I(L_i) \frac{\rho(V, L_i) (L_i \cdot Z)}{\text{PDF}(L_i)} \quad (18)$$

$$= \frac{1}{n} \sum_{i=1}^n I(L_i) \frac{F(V, L_i) G_2(V, L_i)}{G_1(V)}. \quad (19)$$

This estimator usually has low variance because most of the microfacet BRDF terms cancel with the PDF terms and the remaining fraction, $\frac{F(V, L_i) G_2(V, L_i)}{G_1(V)}$, takes values in $[0, 1]$. For more details on these derivations, we refer the reader to the original VNDF-sampling article [Heitz and d'Eon 2014].

References

- DONG, Z., WALTER, B., MARSCHNER, S., AND GREENBERG, D. P. 2015. Predicting appearance from measured microgeometry of metal surfaces. *ACM Trans. Graph.* 35, 1 (Dec.), 9:1–9:13. URL: <http://doi.acm.org/10.1145/2815618>. 3
- HEITZ, E., AND D'EON, E. 2014. Importance sampling microfacet-based BSDFs using the distribution of visible normals. In *Proceedings of the 25th Eurographics Symposium on Rendering*, Eurographics Association, Aire-la-Ville, Switzerland, EGSR '14, 103–112. URL: <https://hal.inria.fr/hal-00996995v2>. 2, 3, 8, 9, 10, 11
- HEITZ, E., DUPUY, J., CRASSIN, C., AND DACHSBACHER, C. 2015. The SGGX microflake distribution. *ACM Trans. Graph.* 34, 4 (July), 48:1–48:11. URL: <http://doi.acm.org/10.1145/2766988>. 3

- HEITZ, E., HANIKA, J., D'EON, E., AND DACHSBACHER, C. 2016. Multiple-scattering microfacet BSDFs with the Smith model. *ACM Trans. Graph.* 35, 4 (July), 58:1–58:14. URL: <http://doi.acm.org/10.1145/2897824.2925943>. 4, 10
- HEITZ, E. 2014. Understanding the masking-shadowing function in microfacet-based BRDFs. *Journal of Computer Graphics Techniques (JCGT)* 3, 2 (June), 48–107. URL: <http://jcgt.org/published/0003/02/03/>. 2, 11
- HEITZ, E. 2017. A Simpler and Exact Sampling Routine for the GGX Distribution of Visible Normals. Research report. URL: <http://hal.archives-ouvertes.fr/hal-01509746>. 3
- NEYRET, F. 1995. A General and Multiscale Model for Volumetric Textures. In *Graphics Interface*, Canadian Human-Computer Communications Society, Toronto, Ontario, Canada, 83–91. URL: <https://hal.inria.fr/inria-00588886>. 2
- NEYRET, F. 1998. Modeling Animating and Rendering Complex Scenes using Volumetric Textures. *IEEE Transactions on Visualization and Computer Graphics* 4, 1, 55 – 70. URL: <https://hal.inria.fr/inria-00537523>. 2, 3
- TROWBRIDGE, T. S., AND REITZ, K. P. 1975. Average irregularity representation of a rough surface for ray reflection. *J. Opt. Soc. Am.* 65, 5 (May), 531–536. URL: <http://www.osapublishing.org/abstract.cfm?URI=josa-65-5-531>. 2
- WALTER, B., MARSCHNER, S. R., LI, H., AND TORRANCE, K. E. 2007. Microfacet models for refraction through rough surfaces. In *Proceedings of the 18th Eurographics Conference on Rendering Techniques*, Eurographics Association, Aire-la-Ville, Switzerland, EGSR'07, 195–206. URL: <https://www.cs.cornell.edu/~srm/publications/EGSR07-btdf.html>. 2
- WALTER, B., DONG, Z., MARSCHNER, S., AND GREENBERG, D. 2015. The Ellipsoid Normal Distribution Function. Supplemental material of Predicting Appearance from Measured Microgeometry of Metal Surfaces, *ACM Trans. Graph.* 35, 4, 9:1–9:13. May 2016 updated version. URL: <http://pdfs.semanticscholar.org/fc79/f6c7340937c2589cf22165f2c4fc9acb2ba2.pdf>. 3, 4

Author Contact Information

Eric Heitz
Unity Technologies
eric@unity3d.com

Eric Heitz, Sampling the GGX Distribution of Visible Normals, *Journal of Computer Graphics Techniques (JCGT)*, vol. 7, no. 4, 1–13, 2018
<http://jcgt.org/published/0007/04/01/>

Received: 2018-07-16

Recommended: 2018-09-07

Published: 2018-11-30

Corresponding Editor: Wojciech Jarosz

Editor-in-Chief: Marc Olano

© 2018 Eric Heitz (the Authors).

The Authors provide this document (the Work) under the Creative Commons CC BY-ND 3.0 license available online at <http://creativecommons.org/licenses/by-nd/3.0/>. The Authors further grant permission for reuse of images and text from the first page of the Work, provided that the reuse is for the purpose of promoting and/or summarizing the Work in scholarly venues and that any reuse is accompanied by a scientific citation to the Work.

