

# The Ellipsoid Normal Distribution Function

Bruce Walter, Zhao Dong, Steve Marschner and Donald P. Greenberg

Supplemental material for ACM TOG paper:

“Predicting Surface Appearance from Measured Microgeometry of Metal Surfaces”

November 2014 (last revised May 2016)

## Abstract

The Ellipsoid NDF is a new normal distribution function (NDF) that we introduce and use in the accompanying paper. It can be used with microfacet BRDFs to model the reflection and refraction of light from surfaces. This distribution is a generalization of the widely used isotropic GGX/Trowbridge-Reitz distribution. The Ellipsoid model is based on the surface statistics of an arbitrary 3D ellipsoid allowing both anisotropy and rotations of the distribution. This document describes the derivation of the ellipsoid NDF along with a corresponding shadowing/masking term, which is needed for energy conservation, and a low-variance importance sampling strategy that guarantees that the sample weights never exceed one.

## 1 Introduction to Ellipsoid NDF

The Ellipsoid normal distribution function (NDF) can be compactly expressed as:

$$D(\mathbf{m}) = \frac{\mathcal{X}_+(\mathbf{m} \cdot \mathbf{n})}{\pi |\mathbf{A}| \|\mathbf{A} \mathbf{n}\| \|\mathbf{A}^{-\top} \mathbf{m}\|^4} \quad (1)$$

where  $\mathbf{m}$  is the microfacet normal,  $\mathbf{n}$  is the geometric normal of the macrosurface, and  $\mathbf{A}$  is an  $3 \times 3$  matrix with determinant  $|\mathbf{A}|$  and whose inverse transpose is denoted as  $\mathbf{A}^{-\top}$ . The normals are represented as 3 element column vectors with unit Euclidian norm (i.e.  $\|\mathbf{m}\| = \|\mathbf{n}\| = 1$ ). The numerator restricts  $\mathbf{m}$  to the hemisphere centered around  $\mathbf{n}$  using the indicator function for positive numbers (i.e.  $\mathcal{X}_+(x) = 1$  if  $x \geq 0$  and is zero otherwise).

The  $3 \times 3$  matrix  $\mathbf{A}$  controls the shape of the distribution. Although the matrix contains 9 elements, there are actually only 5 useful degrees of freedom. Scaling the matrix by a constant (e.g.,  $c \mathbf{A}$ ) and left multiplication by an orthogonal matrix (e.g.,  $\mathbf{Q} \mathbf{A}$ ) does not change the distribution, thus many different matrices can produce the same distribution. It is convenient to work in a coordinate system where the macrosurface normal  $\mathbf{n}$  is aligned with the z-axis and then specify the matrix  $\mathbf{A}$  as the product of a orthogonal rotation matrix  $\mathbf{R}$  and a diagonal scaling matrix  $\mathbf{S}$  as:

$$\mathbf{A} = \mathbf{S} \mathbf{R} \quad \text{where} \quad \mathbf{S} = \begin{bmatrix} \alpha_x & 0 & 0 \\ 0 & \alpha_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{R}^\top \mathbf{R} = \mathbf{I} \quad (2)$$

A 3D rotation matrix provides three degrees of freedom while the scaling matrix provides another two to span the space of all possible ellipsoidal distributions. In this case  $|\mathbf{A}| = |\mathbf{S}| = \alpha_x \alpha_y$ . If we set the rotation matrix to just be the identity (i.e.  $\mathbf{R} = \mathbf{I}$  or no rotation), then the distribution reduces to be the same as the anisotropic distribution GTR2aniso in [Burley 2012, Eq. 13]. And if we further set  $\alpha_x = \alpha_y = \alpha$ , then the distribution becomes identical to the isotropic GGX/Trowbridge-Reitz distribution [Walter et al. 2007; Trowbridge and Reitz 1975]. As in those previous models, the  $\alpha$  parameters control the width of the distribution in two orthogonal directions and correspond to notions of surface roughness.

There are many ways to parameterize the space of 3D rotations. One that we have found convenient is to express the rotation as a

product of three axial rotations:

$$\mathbf{R} = \mathbf{R}_x(\theta_x) \mathbf{R}_y(\theta_y) \mathbf{R}_z(\theta_z) \quad (3)$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_x & -\sin \theta_x \\ 0 & \sin \theta_x & \cos \theta_x \end{bmatrix} \begin{bmatrix} \cos \theta_y & 0 & \sin \theta_y \\ 0 & 1 & 0 \\ -\sin \theta_y & 0 & \cos \theta_y \end{bmatrix} \begin{bmatrix} \cos \theta_z & -\sin \theta_z & 0 \\ \sin \theta_z & \cos \theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

In this space the three rotation parameters and their effects on the distribution are easy to understand.  $\theta_z$  rotates the distribution in the tangent plane to align the axes of anisotropy (whose roughness is controlled by  $\alpha_x$  and  $\alpha_y$  respectively) with the desired principal directions. Then  $\theta_x$  and  $\theta_y$  allow shifting the maximal value of the distribution away from the macrosurface normal direction  $\mathbf{n}$ . Thus if  $\theta_x$  or  $\theta_y$  are non-zero, then we get an asymmetric or skewed distribution. This is an effect that is not usually supported by NDFs in graphics but something that we have observed in some of our measured data. This parameterization is simple and intuitive to control when  $\theta_x$  and  $\theta_y$  are small, which has been generally true in our data. For larger rotations though, alternative representations for rotations such as quaternions might be preferable. For the symmetric cases, such as GGX, when  $\theta_x = 0$  and  $\theta_y = 0$  then we have  $\mathbf{A} \mathbf{n} = \mathbf{n}$  and  $\|\mathbf{A} \mathbf{n}\| = 1$ .

### 1.1 Ellipsoid BRDF

The reflection pattern from a surface is described by its bidirectional reflectance distribution function (BRDF), denoted  $f_r$ . Microfacet theory [Torrance and Sparrow 1967; Blinn 1977; Cook and Torrance 1982] approximates the BRDF in terms of the surface’s normal distribution function as [Walter et al. 2007]:

$$f_r(\psi, \omega) = \frac{D(\mathbf{h}) G(\psi, \omega, \mathbf{h}) F(\psi \cdot \mathbf{h})}{4 |\psi \cdot \mathbf{n}| |\omega \cdot \mathbf{n}|} \quad \text{where} \quad \mathbf{h} = \frac{\psi + \omega}{\|\psi + \omega\|} \quad (4)$$

for light which arrives from direction  $\psi$  and is reflected in direction  $\omega$ . The derived direction  $\mathbf{h}$  is often called the half-direction (or half-vector). We can use the ellipsoid NDF for  $D$ . The fresnel factor  $F$  can be computed using the standard fresnel equations based on the material’s complex index of refraction. The only other piece we need is a suitable shadowing-masking term  $G$ .

Shadowing-masking terms are essential in microfacet models to preserve energy conservation and prevent unrealistic behavior at near-grazing angles. We recommend the following shading-masking term, which we show later guarantees energy conservation when used with the ellipsoid NDF:

$$G(\psi, \omega, \mathbf{m}) = G_1(\psi, \mathbf{m}) G_1(\omega, \mathbf{m}) \quad (5)$$

$$G_1(\mathbf{u}, \mathbf{m}) = \min \left( 1, \frac{2 \|\mathbf{A} \mathbf{n}\|^2 |\mathbf{u} \cdot \mathbf{n}|}{\|\mathbf{A} \mathbf{u}\| \|\mathbf{A} \mathbf{n}\| + (\mathbf{A} \mathbf{u}) \cdot (\mathbf{A} \mathbf{n})} \right) \mathcal{X}_+(\mathbf{u} \cdot \mathbf{m}) \quad (6)$$

If desired, transmission or refraction through the surface could be modeled using the microfacet BSDF definitions from [Walter et al. 2007] along with an extended shadowing/masking term:

$$G(\psi, \omega, \mathbf{m}) = G_1(\text{sgn}(\psi \cdot \mathbf{n}) \psi, \mathbf{m}) G_1(\text{sgn}(\omega \cdot \mathbf{n}) \omega, \mathbf{m}) \quad (7)$$

$\psi$	Direction from which light arrives at surface
$\omega$	Direction into which light is scattered
$\mathbf{n}$	Large-scale, or average, surface normal
$\mathbf{m}$	Local microsurface normal
$D$	Normal distribution function (NDF)
$G$	Bidirectional shadowing-masking function
$G_1$	Monodirectional shadowing function
$\mathbf{A}$	Ellipsoid shape matrix
$C_e$	Constant related to ellipsoid size
$A_e^\perp$	Projected area of ellipsoid
$A_\ell^\perp$	Projected area of ellipsoidal lune
$\mathcal{X}_+(x)$	Positive indicator function ( $= 1$ if $x \geq 0$ else $0$ )

**Figure 1:** List of important symbols.

where  $\text{sgn}$  the sign function which is  $+1$  for positive numbers and  $-1$  for negative ones.

We now have all of the definitions needed evaluate the ellipsoid microfacet BRDF. However in many applications, it is also very useful to be able to randomly sample a BRDF (e.g., in Monte Carlo rendering algorithms). Given one of the directions,  $\psi$  or  $\omega$ , we want to be able to generate the other direction with a probability that is roughly proportional to  $f_r(\psi, \omega)$ . In section 4, we describe high quality sampling schemes for the ellipsoid BRDFs. Note since our BRDFs obey reciprocity (i.e.  $f_r(\psi, \omega) = f_r(\omega, \psi)$ ), the sampling methods are the same for both directions.

## 2 Derivation of Ellipsoid NDF

This section derives the normal distribution of a 3D ellipsoid, which together with a normalization constraint for microfacet NDFs, defines the ellipsoid NDF. While many different NDFs have been proposed, two of the most widely used in computer graphics are the Beckmann and GGX distributions. The Beckmann distribution is derived from assuming Gaussian random statistics for the surface and has proved a good model for some surfaces. In a previous paper [Walter et al. 2007], we measured several surfaces and noted that Beckmann was unable to provide a good fit for the rougher surfaces in our dataset. We tried many different functions until we found one that was analytically tractable and provided a good fit for our ground glass sample, which we termed the GGX<sup>1</sup> distribution. However GGX is actually mathematically identical to an earlier NDF model proposed by Trowbridge and Reitz [Trowbridge and Reitz 1975]<sup>2</sup>, which coincidentally they also matched to measured data from a ground glass sample. Unlike GGX, TrowbridgeReitz was derived by computing the NDF of a special class of ellipsoids, called spheroids, or ellipsoids of revolution. Unfortunately their method does not generalize to arbitrary, or triaxial, ellipsoids which are not surfaces of revolution. Instead we use an alternate approach based on implicit surfaces to derive the NDF for general ellipsoids, which naturally extends the isotropic GGX/TrowbridgeReitz distributions to handle anisotropy and rotation.

Many different surfaces can share the same NDF, so we need not assume our surface actually consists of ellipsoids, but only that it has a similar NDF to one. Deriving NDFs from a simple convex shape, such as a ellipsoid, is intuitively appealing and also allows us to solve some of the related integrals geometrically, which may be easier. One natural way to define an arbitrary ellipsoid is using

<sup>1</sup>The name GGX originally stood for ground glass unknown.

<sup>2</sup>To my knowledge, Brent Burley was the first person notice the equivalence of GGX and Trowbridge-Reitz in 2011, using trig identities.

an implicit surface of all points  $\bar{\mathbf{p}}$  that satisfy this condition:

$$f(\bar{\mathbf{p}}) = \bar{\mathbf{p}}^\top \mathbf{A}^\top \mathbf{A} \bar{\mathbf{p}} - C_e^2 = 0 \quad (8)$$

where  $\mathbf{A}$  is a  $3 \times 3$  matrix as discussed earlier and  $C_e$  is a constant related to the size of the ellipsoid.

The NDF appears in microfacet theory because it is used to convert an area integral into an integral over microsurface normals. This transformation is sometimes called a Gauss map, and its Jacobian is given by the Gaussian curvature,  $K_g$ . The NDF is the Jacobian needed for this change of variables from surface-area to normal-density along with a normalization term and can be defined as:

$$D(\mathbf{m}) = \frac{1}{A_e^\perp(\mathbf{n})} \frac{dA}{d\mathbf{m}} = \frac{1}{A_e^\perp(\mathbf{n}) K_g(\mathbf{m})} \quad (9)$$

where the normalization factor  $A_e^\perp(\mathbf{n})$  is the projected area of the ellipsoid in the direction  $\mathbf{n}$ ,  $dA$  is the area measure over the microsurface,  $d\mathbf{m}$  is the solid angle measure over surface normals, and  $K_g(\mathbf{m})$  is the surface's Gaussian curvature at the point where its local surface normal is equal to  $\mathbf{m}$ . This is well defined since for non-degenerate ellipsoids, each surface normal  $\mathbf{m}$  occurs at only a single point on the ellipsoid.

### 2.1 Gaussian Curvature of Ellipsoids

The Gaussian curvature of an implicit surface is given by [Goldman 2005, Eq. 4.1]:

$$K_g = \frac{(\nabla f)^\top \text{adj}(\mathbf{H}) \nabla f}{\|\nabla f\|^4} \quad (10)$$

where  $\nabla f$  is the gradient of the implicit function and  $\text{adj}(\mathbf{H})$  is the adjugate of its Hessian which, in this case, can be expressed as:

$$\text{adj}(\mathbf{H}) = |\mathbf{H}| \mathbf{H}^{-1} \quad \text{when } |\mathbf{H}| \neq 0 \quad (11)$$

$$\mathbf{H} = \nabla^2 f = 2 \mathbf{A}^\top \mathbf{A} \quad (12)$$

$$\nabla f = 2 \mathbf{A}^\top \mathbf{A} \bar{\mathbf{p}} = \mathbf{H} \bar{\mathbf{p}} \quad (13)$$

Substituting these values, we get:

$$K_g = \frac{|\mathbf{H}| (\mathbf{H} \bar{\mathbf{p}})^\top \mathbf{H}^{-1} (\mathbf{H} \bar{\mathbf{p}})}{\|\mathbf{H} \bar{\mathbf{p}}\|^4} \quad (14)$$

This gives us the Gaussian curvature except, that we want it expressed in terms of the microsurface normal  $\mathbf{m}$  instead of surface position  $\bar{\mathbf{p}}$  which are related by:

$$\mathbf{m} = \frac{\nabla f}{\|\nabla f\|} = \frac{\mathbf{H} \bar{\mathbf{p}}}{\|\mathbf{H} \bar{\mathbf{p}}\|} \quad (15)$$

Substituting this into the numerator we get:

$$K_g = \frac{|\mathbf{H}| \mathbf{m}^\top \mathbf{H}^{-1} \mathbf{m}}{\|\mathbf{H} \bar{\mathbf{p}}\|^2} \quad (16)$$

To get rid of the  $\bar{\mathbf{p}}$  in the denominator, consider the following expansion:

$$\begin{aligned} \mathbf{m}^\top \mathbf{H}^{-1} \mathbf{m} &= \frac{(\mathbf{H} \bar{\mathbf{p}})^\top \mathbf{H}^{-1} (\mathbf{H} \bar{\mathbf{p}})}{\|\mathbf{H} \bar{\mathbf{p}}\|^2} = \frac{2 \bar{\mathbf{p}}^\top \mathbf{A}^\top \mathbf{A} \bar{\mathbf{p}}}{\|\mathbf{H} \bar{\mathbf{p}}\|^2} = \frac{2 C_e^2}{\|\mathbf{H} \bar{\mathbf{p}}\|^2} \\ \implies \|\mathbf{H} \bar{\mathbf{p}}\|^2 &= \frac{2 C_e^2}{\mathbf{m}^\top \mathbf{H}^{-1} \mathbf{m}} \end{aligned} \quad (17)$$

We can then express the Gaussian curvature as:

$$K_g = \frac{|\mathbf{H}| (\mathbf{m}^T \mathbf{H}^{-1} \mathbf{m})^2}{2C_e^2} \quad (18)$$

Next we use the relations that  $|\mathbf{H}| = 8|\mathbf{A}|^2$  and:

$$\mathbf{m}^T \mathbf{H}^{-1} \mathbf{m} = \frac{1}{2} \mathbf{m}^T \mathbf{A}^{-1} \mathbf{A}^{-T} \mathbf{m} = \frac{1}{2} \|\mathbf{A}^{-T} \mathbf{m}\|^2 \quad (19)$$

to express the Gaussian curvature as:

$$K_g = \frac{|\mathbf{A}|^2 \|\mathbf{A}^{-T} \mathbf{m}\|^4}{C_e^2} \quad (20)$$

Now that we have a suitable expression for the Gaussian curvature, next we need to compute the projected area of an ellipsoid.

## 2.2 Projected Area of Ellipsoids

The projected area of a convex shape, such as an ellipsoid, in a direction  $\mathbf{u}$  can be defined by an integral over its surface  $\mathcal{S}$ :

$$A_e^+(\mathbf{u}) = \int_{\mathcal{S}} \chi_+(\mathbf{u} \cdot \mathbf{m}) (\mathbf{u} \cdot \mathbf{m}) d\bar{\mathbf{p}} \quad (21)$$

where

$$\mathbf{m} = \frac{\mathbf{A}^T \mathbf{A} \bar{\mathbf{p}}}{\|\mathbf{A}^T \mathbf{A} \bar{\mathbf{p}}\|} \quad (22)$$

from equations 15 and 12. This integral is difficult to solve directly for general ellipsoids, so we will instead use a more geometric solution approach. We can find the silhouette of the ellipsoid by taking all points where the local surface normal is perpendicular to a view direction  $\mathbf{u}$  which is equivalent to solving for the condition:

$$\begin{aligned} 0 &= \mathbf{u} \cdot \mathbf{m} \\ 0 &= \mathbf{u} \cdot (\mathbf{A}^T \mathbf{A} \bar{\mathbf{p}}) = (\mathbf{A}^T \mathbf{A} \mathbf{u}) \cdot \bar{\mathbf{p}} \end{aligned} \quad (23)$$

Notice that this is also the equation for a plane passing through the origin and perpendicular to the vector  $\mathbf{A}^T \mathbf{A} \mathbf{u}$ . Hence the silhouette must be an ellipse lying in this plane (since the intersection of a plane and an ellipsoid is always an ellipse). This silhouette ellipse necessarily has the same projected area as the ellipsoid.

To compute the area of this ellipse, let us consider points  $\bar{\mathbf{p}}_o$  in a space transformed by the affine transform  $\mathbf{A}$  and defined by:

$$\bar{\mathbf{p}}_o = \mathbf{A} \bar{\mathbf{p}} \quad \text{and} \quad \bar{\mathbf{p}} = \mathbf{A}^{-1} \bar{\mathbf{p}}_o \quad (24)$$

In this transformed space, the implicit equation for the ellipsoid becomes  $\bar{\mathbf{p}}_o^T \bar{\mathbf{p}}_o - C_e^2 = 0$ , which means the ellipsoid has been transformed into a sphere with radius  $C_e$ . Spheres are much easier to analyze, so we can start our computation in this sphere-space and then transform the results back to ellipsoid-space. The silhouette in this space is an intersection of a sphere and a plane passing through its center, so it is a circle with area  $\pi C_e^2$ . To compute the un-transformed ellipse area, we can use the following identity for cross-products:

$$(\mathbf{M} \bar{\mathbf{a}}) \times (\mathbf{M} \bar{\mathbf{b}}) = |\mathbf{M}| \mathbf{M}^T (\bar{\mathbf{a}} \times \bar{\mathbf{b}}) \quad (25)$$

where  $\mathbf{M}$  is any non-singular matrix and  $\bar{\mathbf{a}}$  and  $\bar{\mathbf{b}}$  are 3D vectors. Cross-products transform in the same way as the perpendiculars to planes (i.e. both are bivectors or covariant), and their lengths are proportional to the corresponding areas in their respective spaces.

Thus if our ellipse is in a plane perpendicular to  $\mathbf{A}^T \mathbf{A} \mathbf{u}$ , then the corresponding circle will be perpendicular to  $|\mathbf{A}| \mathbf{A}^{-T} \mathbf{A}^T \mathbf{A} \mathbf{u}$  in the sphere-space. Moreover the ratio between the lengths of these vectors is equal to the ratio between the areas of the silhouette ellipse and circle in their respective spaces.

Finally we have to account for the fact that the ellipse silhouette plane may not be perpendicular to  $\mathbf{u}$  by multiplying by the dot product between its unit plane normal and  $\mathbf{u}$ . Putting these three terms together we get:

$$\begin{aligned} A_e^+(\mathbf{u}) &= (\pi C_e^2) \left( \frac{\|\mathbf{A}^T \mathbf{A} \mathbf{u}\|}{|\mathbf{A}| \|\mathbf{A}^{-T} \mathbf{A}^T \mathbf{A} \mathbf{u}\|} \right) \left( \frac{(\mathbf{A}^T \mathbf{A} \mathbf{u}) \cdot \mathbf{u}}{\|\mathbf{A}^T \mathbf{A} \mathbf{u}\|} \right) \\ &= \frac{\pi C_e^2 \|\mathbf{A} \mathbf{u}\|}{|\mathbf{A}|} \end{aligned} \quad (26)$$

This agrees with previous results for the projected area of an ellipsoid (e.g., [Vickers 1996] albeit in different notation).<sup>3</sup> Now we can combine equations 9, 20, and 26, to complete our derivation of equation 1 for the ellipsoid NDF.  $\square$

## 2.3 Ellipsoidal Lunes and Their Projected Area

Next we generalize the result above to compute projected areas of partial ellipsoids, which we call *ellipsoidal lunes*. Although not needed to define the ellipsoid NDF itself, we will need to do this for related functions such as the shadowing-masking and importance sampling. When using the ellipsoid NDF in a microfacet model, we actually use only half of the ellipsoid because we are only interested in portions where  $\mathbf{m} \cdot \mathbf{n} > 0$ . As per the discussion above, this corresponds to cutting the ellipsoid in half by a plane that is perpendicular to the vector  $\mathbf{A}^T \mathbf{A} \mathbf{n}$ . It is useful to be able to compute the projected area of such a half ellipsoid along an arbitrary direction  $\mathbf{u}$ , where we additionally restrict ourselves to portions of the ellipsoid where  $\mathbf{m} \cdot \mathbf{u} > 0$ . This corresponds to cutting the ellipsoid by two planes, both of which pass through its center, and keeping only the portion on the positive side of both planes. On a sphere such a region is called a spherical lune and by extension we will refer to such a region as an ellipsoidal lune.

We can define the projected area of such an ellipsoidal lune by an integral over the ellipsoid surface  $\mathcal{S}$ :

$$A_\ell^+(\mathbf{u}, \mathbf{v}) = \int_{\mathcal{S}} \chi_+(\mathbf{u} \cdot \mathbf{m}) \chi_+(\mathbf{v} \cdot \mathbf{m}) (\mathbf{u} \cdot \mathbf{m}) d\bar{\mathbf{p}} \quad (27)$$

Comparing to equation 21, we can see that  $A_e^+(\mathbf{u}) = A_\ell^+(\mathbf{u}, \mathbf{u})$ , so this is a generalization of the ellipsoid projected area problem. We follow the same solution approach as before. Using equation 24, we start by working in a transformed space where the ellipsoid becomes a sphere with radius  $C_e$ , and the ellipsoidal lune becomes a spherical lune. Note that for a sphere, the silhouette curves are simply great circles in planes perpendicular to the corresponding projection directions, which is not generally true for ellipsoids. Let the angle between the normals of the two planes defining a spherical lune be  $\theta_\ell$ . Then the projected area of a spherical lune along one of these normals is given by  $\frac{\pi}{2} C_e^2 (1 + \cos \theta_\ell)$ . The vectors  $\mathbf{A} \mathbf{u}$  and  $\mathbf{A} \mathbf{v}$  are perpendiculars to these planes in sphere space, so we can normalize and take their dot product to compute the  $\cos \theta_\ell$ .

Once we know the projected area of the spherical lune, we can account for the change in area when transforming back to the ellipsoid

<sup>3</sup>We are fortunate that the projected area can be expressed so simply. The total surface area of a general ellipsoid has no solution in terms of elementary functions and is instead expressed in terms of elliptic integrals.

space, and compute the corresponding projected area, in the same manner as in equation 26 to get:

$$\begin{aligned} A_\ell^\pm(\mathbf{u}, \mathbf{v}) &= \frac{\pi}{2} C_e^2 \left( 1 + \frac{\mathbf{A}\mathbf{u} \cdot \mathbf{A}\mathbf{v}}{\|\mathbf{A}\mathbf{u}\| \|\mathbf{A}\mathbf{v}\|} \right) \left( \frac{\|\mathbf{A}^\top \mathbf{A}\mathbf{u}\|}{|\mathbf{A}| \|\mathbf{A}\mathbf{u}\|} \right) \left( \frac{(\mathbf{A}^\top \mathbf{A}\mathbf{u}) \cdot \mathbf{u}}{\|\mathbf{A}^\top \mathbf{A}\mathbf{u}\|} \right) \\ &= \frac{\pi C_e^2 (\|\mathbf{A}\mathbf{u}\| \|\mathbf{A}\mathbf{v}\| + \mathbf{A}\mathbf{u} \cdot \mathbf{A}\mathbf{v})}{2 |\mathbf{A}| \|\mathbf{A}\mathbf{v}\|} \end{aligned} \quad (28)$$

## 2.4 Conventions for $C_e$

The ellipsoid size constant  $C_e$  appears in many of our intermediate equations, but not in our final expressions. This is because the NDF and its related functions are defined using ratios between areas such that the  $C_e$  factors always cancel out. During computations however, it is often useful to select a particular value for  $C_e$  and we are free to choose any convenient value. One convention we suggest is to pick  $C_e$  such that  $A_\ell^\pm(\mathbf{n}) = 1$  by inverting equation 26:

$$C_e = \sqrt{\frac{|\mathbf{A}|}{\pi \|\mathbf{A}\mathbf{n}\|}} \quad (29)$$

which conceptually simplifies some of the equations. However computationally is sometimes cheaper to simply choose:

$$C_e = 1 \quad (30)$$

as we do in section 4.

## 3 Shadowing-masking Approximation

For a rough surface, generally some areas will not be visible from the lighting direction (i.e. shadowed) and some will not be visible from the viewing direction (i.e. masked). Ignoring this effect would result in unrealistic behavior, especially at near-grazing angles. The shadowing-masking term  $G(\psi, \omega, \mathbf{m})$  is defined as the fraction of the surface with local normal  $\mathbf{m}$  that is visible from both the light  $\psi$  and viewing  $\omega$  directions. Thus it produces values in the range  $[0, 1]$ . The exact  $G$  function for a surface is highly dependent on its fine-scale details and can be very different even for surfaces with the same NDF. In practice the exact  $G$  function is almost always unknown, and most models settle instead for an approximation that is energy-conserving and has plausible behavior. One common energy-conserving  $G$  is the V-Groove function [Torrance and Sparrow 1967], however its behavior is atypical of most real world surfaces, and thus we do not recommend using it. Instead we use an approximation approach that is often called the Smith shadowing-masking.

We start with the assumption that the bi-directional shadowing-masking term  $G$  can be well approximated as the separable product of two mono-directional shadowing terms  $G_1$  (see equation 5). We further approximate  $G_1$  as being independent of the local surface normal  $\mathbf{m}$  (except for a check to see if it is back-facing with respect to the relevant direction). Under these approximations,  $G_1$  can be computed using an integral of the NDF [Smith 1967; Walter et al. 2007]. Unfortunately, for the ellipsoid NDF we were not able to directly solve this integral analytically. As discussed in [Smith 1967; Heitz 2014], this integral gives exactly the term needed to make the projected area of the visible (i.e. unshadowed) part of the micro-surface equal to the projected area of the macro-surface, under the assumptions above. This provides us with an alternate geometric way to define  $G_1$ .

When viewing the surface from direction  $\mathbf{u}$ , surface locations which are back-facing (i.e. where  $\mathbf{u} \cdot \mathbf{m} < 0$ ) must necessarily be in shadow (i.e.  $G_1 = 0$  in this case). For the rest of the surface, the visible

projected area constraint for the ellipsoid NDF can be expressed in terms of the projected area of the ellipsoidal lune as:

$$A_\ell^\pm(\mathbf{u}, \mathbf{n}) G_1(\mathbf{u}) \leq A_e^\pm(\mathbf{n}) |\mathbf{u} \cdot \mathbf{n}| \quad (31)$$

where the left-hand side is the visible projected area of the micro-surface and the right-hand side is the projected area of the corresponding macro-surface (i.e. a flat surface perpendicular to  $\mathbf{n}$ ). Ideally we would like to enforce this as an equality constraint, but for some parameter settings of the ellipsoid NDF we will settle for the inequality<sup>4</sup>, which is still sufficient to ensure energy conservation. The following expression for  $G_1$  satisfies these constraints:

$$G_1(\mathbf{u}, \mathbf{m}) = \min \left( 1, \frac{A_e^\pm(\mathbf{n}) |\mathbf{u} \cdot \mathbf{n}|}{A_\ell^\pm(\mathbf{u}, \mathbf{n})} \right) \mathcal{X}_+(\mathbf{u} \cdot \mathbf{m}) \quad (32)$$

where the  $\mathcal{X}_+(\cdot)$  term ensures that back-facing portions of the surface are shadowed and the minimum operation ensures  $G_1 \leq 1$  (visible surface cannot be larger than the total surface). Substituting equations 26 and 28 into this equation gives the ellipsoid shadowing term presented in equation 6.  $\square$

## 4 Importance Sampling

Often we know one of the two directions,  $\psi$  or  $\omega$ , and would like to randomly generate the other direction with a probability based on the BRDF  $f_r(\psi, \omega)$ . This process is referred to as importance sampling. Ideally the probability would be exactly proportional to the BRDF (times a cosine factor), but in practice, we generally settle for a function that is only approximately proportional. Because our BRDFs obey reciprocity (i.e.  $f_r(\psi, \omega) = f_r(\omega, \psi)$ ), we can use the same method regardless of which of the two directions we are sampling. Without loss of generality, we will assume here that  $\psi$  is known and we want to randomly sample  $\omega$ .

The standard method for sampling microfacet BRDFs is to first randomly sample a half direction  $\mathbf{m}$ , and then use it to determine the other desired direction. The original sampling method for GGX [Walter et al. 2007] generates half directions with a probability given by:  $p_h(\mathbf{m}) = D(\mathbf{m}) |\mathbf{m} \cdot \mathbf{n}|$ . This probability is independent of the known direction  $\psi$  and works best when  $\psi$  is close to  $\mathbf{n}$ . However for grazing angles, the sampling can sometimes be poor (i.e. exhibit high variance). A better sampling strategy is to choose half directions with a probability proportional to  $D(\mathbf{m}) |\mathbf{m} \cdot \psi|$  and such improved sampling methods are already available for the GGX and Beckmann NDFs [Heitz and d'Eon 2014]. In this section we develop such a sampling method for the ellipsoid NDF.

Working out the normalization term, since probability density functions must integrate to one, our half angle sampling probability for the ellipsoid NDF will be:

$$p_h(\mathbf{m}) = \frac{A_e^\pm(\mathbf{n})}{A_\ell^\pm(\psi, \mathbf{n})} \mathcal{X}_+(\mathbf{m} \cdot \psi) D(\mathbf{m}) |\mathbf{m} \cdot \psi| \quad (33)$$

By design, this normalization term is closely related to our mono-directional shadowing term  $G_1$ . Our ellipsoid importance sampling can also be used for sampling GGX and GTR2aniso NDFs, since these can be viewed as special cases of the ellipsoid NDF. In these cases we achieve similar sampling quality to prior methods [Heitz and d'Eon 2014], though with a few small advantages that we will discuss later.

<sup>4</sup>The inequality is only needed where the NDF is asymmetric, such as when  $\theta_x \neq 0$  or  $\theta_y \neq 0$ . In such cases the microsurface projected area can be smaller than the projected area of the macrosurface for some directions. To keep  $G_1$  physically plausible, we then add a constraint that  $G_1 \leq 1$ .

Importance sampling an NDF is conceptually quite easy if we have an explicit representation for the microsurface, such as our ellipsoid. We can generate half directions  $\mathbf{m}$  with the probability above by shooting random rays at the surface along the known direction  $\psi$  and using the  $\mathbf{m}$  from the corresponding points they hit on the surface. The computational cost however will vary depending on how we implement this process. Below we briefly outline three ways to implement this importance sampling:

**Version 1.** One way to generate such rays is to randomly sample points on the silhouette of a bounding volume for the surface. Then we generate a ray<sup>5</sup> passing through that point with direction  $-\psi$ . If the ray intersects the surface (i.e. the ellipsoid) at a valid point (i.e. inside the lune where  $\mathbf{m} \cdot \mathbf{n} \geq 0$  and  $\mathbf{m} \cdot \psi \geq 0$ ), we use the corresponding intersection point's surface normal as  $\mathbf{m}$ . Otherwise we “reject” this ray and repeat the process until a valid hit point is found. Thus this is a form of rejection sampling.

If we use a sphere for the bounding volume, then the silhouette is a circle whose area can be easily sampled using standard techniques. The drawback with this version is that the rejection rate may be quite high and we may have to generate and test many random rays to create each sample.

**Version 2.** We can improve the version above by using the ellipsoid as its own bounding volume, which will usually be much tighter than a bounding sphere. To generate random samples on the ellipsoid's silhouette, we can use the transformed space from equation 24. Recall that in this space the ellipsoid becomes a sphere with radius  $C_e$  and that the direction  $\psi$  corresponds to the direction  $\mathbf{A}\psi$  in this space. We generate a point  $\bar{\mathbf{p}}_o$  by randomly sampling a point in the circle perpendicular to  $\mathbf{A}\psi$  and then transform this back using equation 24 to create a random point on the ellipsoid's silhouette. This point plus the direction  $-\psi$  defines the random ray, and then we can proceed as above, testing random rays until we get a valid point.

Even better, we can project the point onto the sphere along the direction  $\mathbf{A}\psi$  before transforming back to the ellipsoid space. For a sphere this is trivial to do analytically, and then we are guaranteed the point is already on the ellipsoid's surface, removing the need for ray intersection testing. However the point may lie outside of the valid ellipsoidal lune, so we still may need to generate multiple points before we get a valid one. Thus we still have to use rejection sampling, but the rejection rate is typically much lower than with the first version.

**Version 3.** We can further improve this procedure to get rid of the need for rejection sampling, by only generating points inside the valid ellipsoidal lune. In the transformed sphere-space, the ellipsoidal lune become a spherical lune. The projected area, or silhouette, of a spherical lune is a crescent shape bounded on one side by a hemi-circle and on the other by a hemi-ellipse. Its area is equal to an ellipse formed by compressing a circle along one axis by a factor of  $\frac{1}{2}(1 + \cos \theta_\ell)$  (see Section 2.3). Moreover we can transform this ellipse to the crescent shape using an area-preserving skew transform. Thus the algorithm proceeds as follows. In sphere space, we generate a random point on the silhouette circle, then we apply a 2D affine compress-and-skew transform to convert this to a random point on the crescent. Next we project it along the direction  $\mathbf{A}\psi$  to get a point on the sphere, and then transform this into the ellipsoid space. This is guaranteed to generate a point on the desired ellipsoidal lune and with the desired probability distribution, and thus testing and rejection sampling are no longer needed.

<sup>5</sup>Note these rays are really oriented lines.

## 4.1 Importance Sampling Procedure

In this section gives a set of equations implementing the sampling strategy outlined above (version 3). This is divided into 5 steps below (a-e). For this section, we set  $C_e = 1$  to simplify the equations and assume the determinant of  $\mathbf{A}$  is positive.

**(a) Sample a random point on the unit disk.** Let  $\xi_1$  and  $\xi_2$  be two uniform random numbers in the range  $[0, 1]$ . Then we can generate a random point  $[x_a, y_a]$  on the unit radius disk in the XY plane via:

$$a = 2\xi_1 - 1 \quad (34)$$

$$b = 2\xi_2 - 1 \quad (35)$$

$$x_a = \begin{cases} a \cos\left(\frac{\pi b}{4a}\right) & \text{if } |a| > |b| \\ b \sin\left(\frac{\pi a}{4b}\right) & \text{otherwise} \end{cases} \quad (36)$$

$$y_a = \begin{cases} a \sin\left(\frac{\pi b}{4a}\right) & \text{if } |a| > |b| \\ b \cos\left(\frac{\pi a}{4b}\right) & \text{otherwise} \end{cases} \quad (37)$$

where we have used the concentric mapping [Shirley and Chiu 1997]. To avoid divide-by-zero issues implementations will also need to check if  $a$  and  $b$  are both effectively zero (within some floating point precision) and set  $x_a = y_a = 0$  in that case. The concentric mapping has much lower distortion than alternatives such as a simple polar mapping. This is beneficial if you are using stratified, quasi-random, blue-noise, or other structured sampling patterns leading to reduced noise and faster convergence. You can use any uniform-area mapping from the square to disk but we recommend using the concentric mapping.

**(b) Compress the disk to a crescent and project to 3D spherical lune.** We are working in a sphere space (i.e. a space where ellipsoid has been warped to a sphere) and we want to map from the disk (i.e. the silhouette of a sphere) to a crescent (i.e. the silhouette of a spherical lune). We can accomplish this by compressing the disk in  $x$  by a factor of  $s = \frac{1}{2}(1 + \cos \theta_\ell)$  and applying a  $y$ -dependent translation to  $x$  via:

$$s = \frac{1}{2} \left( 1 + \frac{\mathbf{A}\psi \cdot \mathbf{A}\mathbf{n}}{\|\mathbf{A}\psi\| \|\mathbf{A}\mathbf{n}\|} \right) \quad (38)$$

$$x_b = s x_a + (1 - s) \sqrt{1 - y_a^2} \quad (39)$$

$$y_b = y_a \quad (40)$$

$$z_b = \sqrt{1 - x_b^2 - y_b^2} \quad (41)$$

where we also selected a  $z$  coordinate to project the point from the 2D crescent onto the 3D sphere.

**(c) Rotate spherical lune into correct alignment.** Next we rotate the spherical lune to align with the  $\psi$  and  $\mathbf{n}$  directions by constructing appropriate orthonormal basis vectors:  $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ . We set  $\mathbf{e}_3$  to be collinear with  $\mathbf{A}\psi$ , (i.e. the incident direction in the sphere space) while setting  $\mathbf{e}_1$  to be the perpendicular direction closest to  $\mathbf{A}\mathbf{n}$ .

$$\mathbf{e}_3 = \frac{\mathbf{A}\psi}{\|\mathbf{A}\psi\|} \quad (42)$$

$$\mathbf{e}_2 = \frac{\mathbf{A}\psi \times \mathbf{A}\mathbf{n}}{\|\mathbf{A}\psi \times \mathbf{A}\mathbf{n}\|} = \frac{\mathbf{A}^{-\top}(\psi \times \mathbf{n})}{\|\mathbf{A}^{-\top}(\psi \times \mathbf{n})\|} \quad (43)$$

$$\mathbf{e}_1 = \mathbf{e}_2 \times \mathbf{e}_3 \quad (44)$$

Note that if  $\psi$  and  $\mathbf{n}$  are collinear then their cross product will be zero and the construction of  $\mathbf{e}_2$  above will fail. In this case we

can choose  $\mathbf{e}_2$  to be any direction perpendicular to  $\mathbf{e}_3$ . Practical implementations will need to detect this case and handle it (e.g., using [Frisvad 2012]).

**(d) Map back to ellipsoid to get corresponding micronormal  $\mathbf{m}$ .**

$$\mathbf{p}_c = x_b \mathbf{e}_1 + y_b \mathbf{e}_2 + z_b \mathbf{e}_3 \quad (45)$$

$$\mathbf{m} = \frac{\mathbf{A}^\top \mathbf{p}_c}{\|\mathbf{A}^\top \mathbf{p}_c\|} \quad (46)$$

where  $\mathbf{p}_c$  is our random point on the spherical lune,  $\mathbf{A}\mathbf{p}_c$  would be the corresponding point on the ellipsoidal lune, and  $\mathbf{m}$  is its surface normal on the ellipsoid. This procedure generates random half directions  $\mathbf{m}$  with our target probability density (equation 33).

**(e) Generate outgoing direction  $\omega$  from  $\mathbf{m}$ .** Now that we have a sampled half direction, we can follow the standard procedures for generating the outgoing direction  $\omega$  by reflecting the incident direction  $\psi$  across  $\mathbf{m}$  (e.g., see [Walter et al. 2007]).

$$\omega = 2(\psi \cdot \mathbf{m})\mathbf{m} - \psi \quad (47)$$

$$p_o(\omega) = \frac{p_h(\mathbf{m})}{4|\psi \cdot \mathbf{m}|} = \frac{A_e^+(\mathbf{n})}{4A_e^+(\psi, \mathbf{n})} \mathcal{X}_+(\mathbf{m} \cdot \psi) D(\mathbf{m}) \quad (48)$$

The corresponding weight for this sampled direction  $\omega$  is:

$$\text{weight}(\omega) = \frac{f_r(\psi, \omega) |\omega \cdot \mathbf{n}|}{p_o(\omega)} \quad (49)$$

$$= \frac{G_1(\psi, \mathbf{m}) G_1(\omega, \mathbf{m}) F(\psi \cdot \mathbf{m}) A_e^+(\psi, \mathbf{n})}{|\psi \cdot \mathbf{n}| A_e^+(\mathbf{n})} \quad (50)$$

$$\leq G_1(\omega, \mathbf{m}) F(\psi \cdot \mathbf{m}) \quad (51)$$

$$\leq 1 \quad (52)$$

where we have used the microfacet BRDF (equation 4) along with our suggested ellipsoid shadowing/masking functions. The first inequality is due to the min operation in our definition of  $G_1$  (equation 32). If  $\theta_x$  and  $\theta_y$  are zero, such as for a GGX or GTR2aniso NDF, then the min operation is unnecessary and this is actually an equality. The second inequality follows from the fact that both  $G_1$  and  $F$  produce values in the range zero to one.

## 4.2 Reduction to importance sampling GGX

If there is no skew (e.g.,  $\theta_x = 0$  and  $\theta_y = 0$ ) then the ellipsoid NDF reduces to GGX or its anisotropic extension GTR2aniso, and the above procedure can be used to importance sample these NDFs. In this case we have  $\frac{\mathbf{A}\mathbf{n}}{\|\mathbf{A}\mathbf{n}\|} = \mathbf{n}$  and  $\text{weight}(\omega) = G_1(\omega, \mathbf{m}) F(\psi \cdot \mathbf{m})$  which simplifies some computations, and they can be further simplified if we work in a coordinate system aligned with  $\mathbf{n}$  and the principal directions of roughness where  $\mathbf{A}$  is just a diagonal matrix.

High quality importance sampling of GGX was first described in [Heitz and d'Eon 2014] using the algebraic approach of solving for and inverting the related one dimensional cumulative distribution functions (CDF). While our geometry-based importance sampling mapping and the prior algebra-based one are different, they both target the same probability density function (PDF) for  $\mathbf{m}$  and thus are mostly equivalent. Our sphere-space transform is essentially the same as their slope stretching operation except performed in a different space.

Nevertheless there are a few advantages to our geometry-based importance sampling procedure. In the algebraic approach, one of the CDFs could not be inverted analytically so a rational approximation is used instead. This causes a small discrepancy between the target

and actual importance sampling PDFs. In our test there was an average relative error of 0.4% in the algebra-based method's sampling probabilities. This small error is likely insignificant for most graphics applications but could be an issue for applications with higher accuracy requirements. Our approach does involve any approximations and thus is more accurate. Also when used with the concentric mapping as we suggest, our mapping has lower distortion which is advantageous if used with structured sampling methods (though if used with a simple polar mapping then our distortion would be higher than for the algebraic approach).

## 5 Discussion

### 5.1 Open Issues

**Manufacturability.** In order to ensure our microsurface is a height field, we defined the ellipsoid NDF using the half of an ellipsoid where  $\mathbf{m} \cdot \mathbf{n} \geq 0$ . From equation 23, we can see this is equivalent to slicing the ellipsoid in half by a plane perpendicular to  $\mathbf{A}^\top \mathbf{A}\mathbf{n}$ , which means that the average surface normal of our microsurface is  $\mathbf{A}^\top \mathbf{A}\mathbf{n}$ . However when  $\theta_x$  or  $\theta_y$  are non-zero, then in general  $\mathbf{A}^\top \mathbf{A}\mathbf{n} \neq \mathbf{n}$  which can be a contradiction since we claimed that  $\mathbf{n}$  was the average or large-scale normal of our surface. For rendering, as long as  $\theta_x$  and  $\theta_y$  are small the discrepancy is likely not significant, but if the angles are large or we actually wanted to manufacture such surfaces, this could be problematic if the discrepancy persists across larger regions of the surface.

There are many different ways one could modify the NDF to get the correct the average normal. For example, we could always split the ellipsoid by a plane perpendicular to  $\mathbf{n}$ , though in this case the microsurface would no longer always be a height field. A better alternative might be to add some additional vertical microsurface area (i.e. where  $\mathbf{m} \cdot \mathbf{n} = 0$ ) to the NDF. Essentially a vertical extension from the cut plane of the ellipsoid back to a plane perpendicular to  $\mathbf{n}$ . Since vertical surfaces do not affect  $A_e^+(\mathbf{n})$  and do not generate valid reflection directions (would reflect light downward into the wrong hemisphere), only the shadowing-masking term would need to be modified (at least for reflections, it might not work as well for refraction). It remains future work to determine if and when this issue may be significant and what the best remedies may be.

**Relation to normal maps.** Normal maps are often used to model meso-scale surface normals that are closely related NDF skew phenomena the ellipsoid NDF is designed to support. And normal maps can similarly cause a shifting of the maximum of a BRDF away from the direction of the surface normal. Compared to skewed NDFs, normal maps have the advantage that they are widely used and can be applied to any BRDF model, not just microfacet ones. However they also have some significant disadvantages. Unlike our skewed NDF model, normal maps often violate energy conservation, break reciprocity, and can lead to singularities or ill-defined behavior in some cases (e.g., for directions that are at or below the horizon of the shading normal). In future, it would be interesting to perform a more detailed comparison between the skewed NDF and normal map approaches.

### 5.2 Relation to the SGGX microflake distribution

The SGGX microflake model [Heitz et al. 2015] was also derived based on the surface normals of arbitrary 3D ellipsoids. Our development of the ellipsoid NDF was conducted independently of and concurrently with the development of the SGGX model.<sup>6</sup> Since

<sup>6</sup>SGGX was published first but this was after both papers had been submitted and accepted for publication.

both are based on the same underlying mathematical model, SSGX and our ellipsoid NDF are closely related with many similarities. However because they are applied to somewhat different domains there are several differences between them. SGGX is a 3D volumetric scattering model (i.e. a microflake distribution), while our ellipsoid NDF is a 2D surface scattering model (i.e. a microfacet distribution).

The SGGX defines its ellipsoids by a symmetric matrix  $S$  while we use a non-symmetric matrix  $A$ , however the two are convertible via the relation  $S = A^T A$  (and their SGGX matrix  $S$  should not be confused with our scaling matrix  $S$ ). Normalizing microflake distributions can be problematic so SGGX shows how to avoid this issue by combining the scattering coefficient with the volumetric phase function. Microfacet distributions, however, do need to be normalized which is why our NDF includes a normalizing term of  $\|An\|$  in the denominator that is not present in SGGX.

For the volumetric ellipsoid case there is no masking and the shadowing is relatively simple. One only needs to ensure the micro-normal is not back-facing with respect to the incident direction (e.g.,  $\psi \cdot \mathbf{m} \geq 0$ ). The situation is more complicated for surface microfacet distributions because we also have to consider the macro-surface normal  $\mathbf{n}$ . This double constraint (e.g.,  $\psi \cdot \mathbf{m} \geq 0$  and  $\mathbf{n} \cdot \mathbf{m} \geq 0$ ) affects both the shadowing/masking and importance sampling and required us to develop methods for computing and sampling ellipsoid lunes, which are not needed for SGGX. Conversely the SGGX work develops multiple techniques that are not considered here but that are essential for their volumetric scattering applications, including projection operators from other distribution, filtering steps needed for multi-scale computation, and consideration of diffuse scattering cases.

## A Revision History

**September 2015:** Version that appears in the ACM digital library as supplemental material for paper "Predicting Surface Appearance from Measured Microgeometry of Metal Surfaces".

**May 2016:** Fixed typo in equation 33 and changed surrounding text to use  $\mathbf{m}$  instead of  $\mathbf{h}$  to be more consistent with the rest of the text. Significantly expanded section 4 to more clearly describe the importance sampling procedure and better reference prior work. Added discussion in section 5.2 of relationship to the concurrent SGGX work. Other small text changes throughout.

## References

- BLINN, J. F. 1977. Models of light reflection for computer synthesized pictures. In *Computer Graphics (Proceedings of SIGGRAPH 77)*, 192–198.
- BURLEY, B. 2012. Physically-based shading at disney. In *ACM SIGGRAPH 2012 Course: Practical Physically-based Shading in Film and Game Production*, SIGGRAPH '12.
- COOK, R. L., AND TORRANCE, K. E. 1982. A reflectance model for computer graphics. *ACM Trans. Graph.* 1, 1 (Jan.), 7–24.
- FRISVAD, J. R. 2012. Building an orthonormal basis from a 3d unit vector without normalization. *Journal of Graphics Tools* 16, 3, 151–159.
- GOLDMAN, R. 2005. Curvature formulas for implicit curves and surfaces. *Comput. Aided Geom. Des.* 22, 7 (Oct.), 632–658.
- HEITZ, E., AND D'EON, E. 2014. Importance sampling microfacet-based bsdfs using the distribution of visible normals. *Computer Graphics Forum* 33, 4, 103–112.

- HEITZ, E., DUPUY, J., CRASSIN, C., AND DACHSBACHER, C. 2015. The sgx microflake distribution. *ACM Trans. Graph.* 34, 4 (July), 48:1–48:11.
- HEITZ, E. 2014. Understanding the masking-shadowing function in microfacet-based brdfs. *Journal of Computer Graphics Techniques (JCGT)* 3, 2 (June), 32–91.
- SHIRLEY, P., AND CHIU, K. 1997. A low distortion map between disk and square. *Journal of Graphics Tools* 2, 3, 45–52.
- SMITH, B. 1967. Geometrical shadowing of a random rough surface. *IEEE Trans. on Antennas and Propagation* 15, 5 (september), 668–671.
- TORRANCE, K. E., AND SPARROW, E. M. 1967. Theory for off-specular reflection from roughened surfaces. *J. Opt. Soc. Am. A* 9, 1105–1114.
- 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.
- VICKERS, G. 1996. The projected areas of ellipsoids and cylinders. *Powder Technology* 86, 2, 195–200.
- WALTER, B., MARSCHNER, S. R., LI, H., AND TORRANCE, K. E. 2007. Microfacet models for refraction through rough surfaces. In *Proc. of EGSR*, 195–206.