The Ellipsoid Normal Distribution Function

Bruce Walter, Zhao Dong, Steve Marschner and Donald P. Greenberg
November 2014 (last revised August 2015)
Supplemental material for ACM TOG paper:
"Predicting Surface Appearance from Measured Microgeometry of Metal Surfaces"

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 micro-facet BRDF to model the reflection and refraction of light from surfaces. This new 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 the sample weights never exceed one.

1 Introduction to Ellipsoid NDF

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

$$D(\mathbf{m}) = \frac{\chi_+(\mathbf{m}\cdot\mathbf{n})}{\pi\,|\mathbf{A}|\;\|\mathbf{A}\,\mathbf{n}\|\;\|\mathbf{A}^{-\mathsf{T}}\mathbf{m}\|^4} \tag{1}$$

where $\mathbf{m}$ is the micro-facet normal, $\mathbf{n}$ is the geometric normal of the macro-surface, and $\mathbf{A}$ is an 3x3 matrix with determinant $|\mathbf{A}|$ and whose inverse transpose is denoted as $\mathbf{A}^{-\mathsf{T}}$. 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. $\chi_+(x) = 1$ if $x \geq 0$ and is zero otherwise).

The 3x3 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 specify the matrix $\mathbf{A}$ as the product of a orthogonal rotation matrix $\mathbf{R}$ and a diagonal scaling matrix $\mathbf{S}$ as:

$$\mathbf{A} = \mathbf{S}\,\mathbf{R} \;\text{ where }\; \mathbf{S} = \begin{bmatrix}\alpha_x & 0 & 0\\ 0 & \alpha_y & 0\\ 0 & 0 & 1\end{bmatrix} \;\text{ and }\; \mathbf{R}^{\mathsf{T}}\mathbf{R} = \mathbf{I} \tag{2}$$

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. 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 exactly 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 α 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 work in a coordinate system where the macro-surface normal is aligned with the z-axis (i.e. $\mathbf{n} = \mathbf{z}$) and express the rotation as a product of three axial rotations:

$$\mathbf{R} = \mathbf{R}_x(\theta_x)\,\mathbf{R}_y(\theta_y)\,\mathbf{R}_z(\theta_z) \tag{3}$$
$$= \begin{bmatrix}1 & 0 & 0\\ 0 & \cos\theta_x & -\sin\theta_x\\ 0 & \sin\theta_x & \cos\theta_x\end{bmatrix}\begin{bmatrix}\cos\theta_y & 0 & \sin\theta_y\\ 0 & 1 & 0\\ -\sin\theta_y & 0 & \cos\theta_y\end{bmatrix}\begin{bmatrix}\cos\theta_z & -\sin\theta_z & 0\\ \sin\theta_z & \cos\theta_z & 0\\ 0 & 0 & 1\end{bmatrix}$$

In this space the three rotation parameters and their effects on the distribution are easy to understand. θ_z rotates the distribution in the tangent plane to align the axes of anisotropy (whose roughness is controlled by α_x and α_y respectively) with the desired principal directions. Then θ_x and θ_y allow shifting the maximal value of the distribution away from the macro-surface normal direction $\mathbf{n}$. Thus if θ_x or θ_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 θ_x and θ_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.

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 surfaces normal distribution function as [Walter et al. 2007]:

$$f_\mathrm{r}(\psi,\omega) = \frac{D(\mathbf{h})\,G(\psi,\omega,\mathbf{h})\,F(\psi\cdot\mathbf{h})}{4\,|\psi\cdot\mathbf{n}|\;|\omega\cdot\mathbf{n}|} \;\text{ where }\; \mathbf{h} = \frac{\psi+\omega}{\|\psi+\omega\|} \tag{4}$$

for light which arrives from direction ψ and is reflected in direction ω. 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:

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

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

We now have all 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, ψ or ω, we want to be able to generate the other direction with a probability that is roughly proportional to $f_\mathrm{r}(\psi,\omega)$. In section 4, we describe high quality sampling schemes for the ellipsoid BRDFs. Note since our BRDFs obey reciprocity (i.e. $f_\mathrm{r}(\psi,\omega) = f_\mathrm{r}(\omega,\psi)$), the sampling methods are the same for both directions.

ψ	Direction from which light arrives at surface
ω	Direction into which light is scattered
$\mathbf{n}$	Large-scale, or average, surface normal
$\mathbf{m}$	Local micro-surface 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
$A_\mathrm{e}^\perp$	Projected area of ellipsoid
$A_\ell^\perp$	Projected area of ellipsoidal lune
$\chi_+(x)$	Positive indicator function (= 1 if $x \geq 0$ else 0)

Figure 1: *List of important symbols.*

2 Derivation of Ellipsoid NDF

In this section we show how to derive the normal distribution of a 3D ellipsoid, which together with a normalization constraint for micro-facet NDFs, defines the ellipsoid NDF. While many different NDFs have been proposed, the two most widely used ones 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, 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[1] distribution. However GGX is actually mathematically identical to an earlier NDF model proposed by Trowbridge and Reitz [Trowbridge and Reitz 1975][2], which coincidently 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 the related integrals geometrically which may be easier. One natural way to define an arbitrary ellipsoid is using an implicit surface of all points $\vec{\mathbf{p}}$ that satisfy this condition:

$$f(\vec{\mathbf{p}}) = \vec{\mathbf{p}}^\mathsf{T}\mathbf{A}^\mathsf{T}\mathbf{A}\,\vec{\mathbf{p}} - C_\mathrm{e}^2 = 0 \tag{7}$$

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

The NDF appears in micro-facet theory because it is used to convert an area integral into an integral over micro-surface 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:

$$D(\mathbf{m}) = \frac{1}{A_\mathrm{e}^\perp(\mathbf{n})}\frac{dA}{d\mathbf{m}} = \frac{1}{A_\mathrm{e}^\perp(\mathbf{n})\,K_\mathrm{g}(\mathbf{m})} \tag{8}$$

[1]The name GGX was originally stood for ground glass unknown.

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

where the normalization factor $A_\mathrm{e}^\perp(\mathbf{n})$ is the projected area of the ellipsoid in the direction $\mathbf{n}$, dA is the area measure over the micro-surface, $d\mathbf{m}$ is the solid angle measure over surface normals, and $K_\mathrm{g}(\mathbf{m})$ 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]:

$$K_\mathrm{g} = \frac{(\nabla f)^\mathsf{T}\mathrm{adj}(\mathbf{H})\nabla f}{\|\nabla f\|^4} \tag{9}$$

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

$$\mathrm{adj}(\mathbf{H}) = |\mathbf{H}|\;\mathbf{H}^{-1} \quad \text{when} \quad |\mathbf{H}| \neq 0 \tag{10}$$

$$\mathbf{H} = \nabla^2 f = 2\,\mathbf{A}^\mathsf{T}\mathbf{A} \tag{11}$$

$$\nabla f = 2\,\mathbf{A}^\mathsf{T}\mathbf{A}\vec{\mathbf{p}} = \mathbf{H}\vec{\mathbf{p}} \tag{12}$$

Subsituting these values, we get:

$$K_\mathrm{g} = \frac{|\mathbf{H}|\,(\mathbf{H}\vec{\mathbf{p}})^\mathsf{T}\mathbf{H}^{-1}(\mathbf{H}\vec{\mathbf{p}})}{\|\mathbf{H}\vec{\mathbf{p}}\|^4} \tag{13}$$

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

$$\mathbf{m} = \frac{\nabla f}{\|\nabla f\|} = \frac{\mathbf{H}\vec{\mathbf{p}}}{\|\mathbf{H}\vec{\mathbf{p}}\|} \tag{14}$$

Substituting this into the numerator we get:

$$K_\mathrm{g} = \frac{|\mathbf{H}|\,\mathbf{m}^\mathsf{T}\mathbf{H}^{-1}\mathbf{m}}{\|\mathbf{H}\vec{\mathbf{p}}\|^2} \tag{15}$$

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

$$\mathbf{m}^\mathsf{T}\mathbf{H}^{-1}\mathbf{m} = \frac{(\mathbf{H}\vec{\mathbf{p}})^\mathsf{T}\mathbf{H}^{-1}(\mathbf{H}\vec{\mathbf{p}})}{\|\mathbf{H}\vec{\mathbf{p}}\|^2} = \frac{2\vec{\mathbf{p}}^\mathsf{T}\mathbf{A}^\mathsf{T}\mathbf{A}\vec{\mathbf{p}}}{\|\mathbf{H}\vec{\mathbf{p}}\|^2} = \frac{2C_\mathrm{e}^2}{\|\mathbf{H}\vec{\mathbf{p}}\|^2}$$

$$\implies \|\mathbf{H}\vec{\mathbf{p}}\|^2 = \frac{2C_\mathrm{e}^2}{\mathbf{m}^\mathsf{T}\mathbf{H}^{-1}\mathbf{m}} \tag{16}$$

We can then express the gaussian curvature as:

$$K_\mathrm{g} = \frac{|\mathbf{H}|\left(\mathbf{m}^\mathsf{T}\mathbf{H}^{-1}\mathbf{m}\right)^2}{2C_\mathrm{e}^2} \tag{17}$$

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

$$\mathbf{m}^\mathsf{T}\mathbf{H}^{-1}\mathbf{m} = \frac{1}{2}\mathbf{m}^\mathsf{T}\mathbf{A}^{-1}\mathbf{A}^{-\mathsf{T}}\mathbf{m} = \frac{1}{2}\left\|\mathbf{A}^{-\mathsf{T}}\mathbf{m}\right\|^2 \tag{18}$$

to express the gaussian curvature as:

$$K_\mathrm{g} = \frac{|\mathbf{A}|^2\left\|\mathbf{A}^{-\mathsf{T}}\mathbf{m}\right\|^4}{C_\mathrm{e}^2} \tag{19}$$

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 $\mathcal{S}$:

$$A_{\mathrm{e}}^{\perp}(\mathbf{u}) = \int_{\mathcal{S}} \chi_{+}(\mathbf{u}\cdot\mathbf{m})\ (\mathbf{u}\cdot\mathbf{m})\ d\vec{\mathbf{p}} \tag{20}$$

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:

$$\begin{aligned} 0 &= \mathbf{u}\cdot\mathbf{m} \\ 0 &= \mathbf{u}\cdot(\mathbf{A}^{\mathsf{T}}\mathbf{A}\vec{\mathbf{p}}) = (\mathbf{A}^{\mathsf{T}}\mathbf{A}\mathbf{u})\cdot\vec{\mathbf{p}} \end{aligned} \tag{21}$$

Notice that this is also the equation for a plane passing through the origin and perpendicular to the vector $\mathbf{A}^{\mathsf{T}}\mathbf{A}\mathbf{u}$. 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 $\vec{\mathbf{p}}_{\circ}$ in a space transformed by the affine transform $\mathbf{A}$ and defined by:

$$\vec{\mathbf{p}}_{\circ} = \mathbf{A}\vec{\mathbf{p}} \quad \text{and} \quad \vec{\mathbf{p}} = \mathbf{A}^{-1}\vec{\mathbf{p}}_{\circ} \tag{22}$$

In this transformed space, the implicit equation for the ellipsoid becomes $\vec{\mathbf{p}}_{\circ}{}^{\mathsf{T}}\vec{\mathbf{p}}_{\circ} - C_{\mathrm{e}}^2 = 0$, 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 its area is πC_{e}^2. To compute the un-transformed area, we can use the following identity for cross-products:

$$(\mathbf{M}\vec{\mathbf{a}}) \times (\mathbf{M}\vec{\mathbf{b}}) = |\mathbf{M}|\,\mathbf{M}^{-\mathsf{T}}(\vec{\mathbf{a}} \times \vec{\mathbf{b}}) \tag{23}$$

where $\mathbf{M}$ is any non-singular matrix and $\vec{\mathbf{a}}$ and $\vec{\mathbf{b}}$ are 3D vectors. Cross-products transform in the same way as the perpendiculars to planes (i.e. both are contra-variant), and their lengths are proportional to the corresponding areas in their respective spaces. Thus if our ellipse is perpendicular to $\mathbf{A}^{\mathsf{T}}\mathbf{A}\mathbf{u}$, then the corresponding circle will be perpendicular to $|\mathbf{A}|\,\mathbf{A}^{-\mathsf{T}}\mathbf{A}^{\mathsf{T}}\mathbf{A}\mathbf{u}$ 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 the unit plane normal and $\mathbf{u}$. Putting these three terms together we get:

$$\begin{aligned} A_{\mathrm{e}}^{\perp}(\mathbf{u}) &= \left(\pi C_{\mathrm{e}}^2\right)\left(\frac{\|\mathbf{A}^{\mathsf{T}}\mathbf{A}\mathbf{u}\|}{|\mathbf{A}|\,\|\mathbf{A}^{-\mathsf{T}}\mathbf{A}^{\mathsf{T}}\mathbf{A}\mathbf{u}\|}\right)\left(\frac{(\mathbf{A}^{\mathsf{T}}\mathbf{A}\mathbf{u})\cdot\mathbf{u}}{\|\mathbf{A}^{\mathsf{T}}\mathbf{A}\mathbf{u}\|}\right) \\ &= \frac{\pi C_{\mathrm{e}}^2\|\mathbf{A}\mathbf{u}\|}{|\mathbf{A}|} \end{aligned} \tag{24}$$

This agrees with previous results for the projected area of an ellipsoid (e.g., [Vickers 1996] albeit in different notation).[3] Now we can combine equations 8, 19, and 24, to complete our derivation of equation 1 for the ellipsoid NDF. □

[3] 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.

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 this for related functions such as the shadowing-masking and importance sampling. When using the ellipsoid NDF in a microfacet model for a surface reflectance, 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 $\mathbf{A}^{\mathsf{T}}\mathbf{A}\mathbf{n}$. 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 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 $\mathcal{S}$:

$$A_{\ell}^{\perp}(\mathbf{u},\mathbf{v}) = \int_{\mathcal{S}} \chi_{+}(\mathbf{u}\cdot\mathbf{m})\ \chi_{+}(\mathbf{v}\cdot\mathbf{m})\ (\mathbf{u}\cdot\mathbf{m})\ d\vec{\mathbf{p}} \tag{25}$$

Comparing to equation 20, we can see that $A_{\mathrm{e}}^{\perp}(\mathbf{u}) = A_{\ell}^{\perp}(\mathbf{u},\mathbf{u})$, so this is generalization of the ellipsoid projected area problem. We follow the same solution approach as before. Using equation 22, 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 projection directions, which is not generally true for ellipsoids. Let the angle between the normals of the two planes defining a spherical lune be θ_{ℓ}. Then the projected area of a spherical lune along one of these normals is given by $\frac{\pi}{2}C_{\mathrm{e}}^2(1+\cos\theta_{\ell})$. 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 space, and compute the corresponding projected area, in the same manner as in equation 24 to get:

$$\begin{aligned} A_{\ell}^{\perp}(\mathbf{u},\mathbf{v}) &= \frac{\pi}{2}C_{\mathrm{e}}^2\left(1+\frac{\mathbf{A}\mathbf{u}\cdot\mathbf{A}\mathbf{v}}{\|\mathbf{A}\mathbf{u}\|\,\|\mathbf{A}\mathbf{v}\|}\right)\left(\frac{\|\mathbf{A}^{\mathsf{T}}\mathbf{A}\mathbf{u}\|}{|\mathbf{A}|\,\|\mathbf{A}\mathbf{u}\|}\right)\left(\frac{(\mathbf{A}^{\mathsf{T}}\mathbf{A}\mathbf{u})\cdot\mathbf{u}}{\|\mathbf{A}^{\mathsf{T}}\mathbf{A}\mathbf{u}\|}\right) \\ &= \frac{\pi C_{\mathrm{e}}^2\left(\|\mathbf{A}\mathbf{u}\|\,\|\mathbf{A}\mathbf{v}\| + \mathbf{A}\mathbf{u}\cdot\mathbf{A}\mathbf{v}\right)}{2\,|\mathbf{A}|\,\|\mathbf{A}\mathbf{v}\|} \end{aligned} \tag{26}$$

2.4 Convention 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. The convention we suggest is to pick C_{e} such that $A_{\mathrm{e}}^{\perp}(\mathbf{n}) = 1$ by inverting equation 24:

$$C_{\mathrm{e}} = \sqrt{\frac{|\mathbf{A}|}{\pi\,\|\mathbf{A}\mathbf{n}\|}} \tag{27}$$

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 not visi-

ble 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 ψ and viewing ω 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 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:

$$A^{\perp}_{\ell}(\mathbf{u},\mathbf{n})\, G_1(\mathbf{u}) \le A^{\perp}_{\mathrm{e}}(\mathbf{n})\ |\mathbf{u}\cdot\mathbf{n}| \tag{28}$$

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 enforce this as an equality constraint, but for some parameter settings of the ellipsoid NDF we will settle for the inequality, which is still sufficient to ensure energy conservation. The following expression for G_1 satisfies these constraints:

$$G_1(\mathbf{u},\mathbf{m}) = \min\left(1, \frac{A^{\perp}_{\mathrm{e}}(\mathbf{n})\ |\mathbf{u}\cdot\mathbf{n}|}{A^{\perp}_{\ell}(\mathbf{u},\mathbf{n})}\right) \chi_+(\mathbf{u}\cdot\mathbf{m}) \tag{29}$$

where the $\chi_+()$ term ensures that back-facing portions of the surface are shadowed and the minimum operation ensures $G_1 \le 1$ (visible surface cannot be larger than the total surface). Substituting equations 24 and 26 into this equation gives the ellipsoid shadowing term presented in equation 6. □

4 Importance Sampling

Often we know one of the two directions, ψ or ω, and would like randomly generate the other direction with a probability based on $f_{\mathrm{r}}(\psi,\omega)$. This process is referred to as importance sampling the BRDF. Ideally the probability would be exactly proportional to the BRDF, but in practice, we generally settle for a function that is only approximately proportional. Because our BRDFs obey reciprocity (i.e. $f_{\mathrm{r}}(\psi,\omega) = f_{\mathrm{r}}(\omega,\psi)$), 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 ψ is known and we want to randomly sample ω.

The standard method for sampling microfacet BRDFs is to first randomly sample a half direction $\mathbf{h}$, and then use that determine the other desired direction (e.g., see [Walter et al. 2007]). The usual sampling methods for the GGX and Beckmann distributions generates half vectors with a probability given by: $p(\mathbf{h}) = D(\mathbf{h})\,|\mathbf{h}\cdot\mathbf{n}|$ [Walter et al. 2007]. This means the probability is actually independent of the known direction ψ, and tends to work best when ψ is close to $\mathbf{n}$. For grazing angles, however such sampling can sometimes be poor (i.e. exhibit high variance).

A better sampling method is to choose half directions with a probability proportional to $D(\mathbf{h})\,|\mathbf{h}\cdot\psi|$. Working out the normalization term, since probability density functions must integrate to one, this means our half angle sampling probability for the ellipsoid NDF will be:

$$p(\mathbf{h}) = \frac{A^{\perp}_{\mathrm{e}}(\mathbf{n})}{A^{\perp}_{\ell}(\mathbf{u},\mathbf{n})} \chi_+(\mathbf{h}\cdot\psi)\, D(\mathbf{h})\,|\mathbf{h}\cdot\psi| \tag{30}$$

By design, this normalization term is closely related to our mono-directional shadowing term G_1. In cases where the ellipsoid NDF reduces to the prior NDFs, GGX or GTR2aniso, the sampling methods presented here provide a better importance sampling than the usual approaches which do not take ψ into account.

Importance sampling an NDF is quite easy if we have an explicit representation for the micro-surface, such as our ellipsoid. We can generate directions $\mathbf{m}$ with the probability above by shooting random rays at the surface along the known direction ψ and using the $\mathbf{m}$ from the corresponding points they hit on the surface. 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[4] passing through that point with direction -ψ. If the ray intersects the surface (i.e. the ellipsoid) at a valid point (i.e. inside the lune where $\mathbf{m}\cdot\mathbf{n} \ge 0$ and $\mathbf{m}\cdot\psi \ge 0$), we use the corresponding hit point's $\mathbf{m}$, 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 just 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 get a 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 22. Recall that in this space the ellipsoid becomes a sphere with radius C_{e} and that the direction ψ corresponds to the direction $\mathbf{A}\psi$ in this space. We generate a point $\vec{\mathbf{p}}_{\circ}$ by randomly sampling a point in the circle perpendicular to $\mathbf{A}\psi$ and then transform this back using equation 22 to create a random point on the ellipsoid's silhouette. This point plus the direction -ψ 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 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

[4] Note these rays are really just oriented lines.

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 modifying the above procedure to only generate 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 desired ellipsoidal lune and with the desired probability distribution, and thus testing and rejection sampling is no longer needed.

5 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 21, we can see this is equivalent to slicing the ellipsoid in half by a plane perpendicular to $\mathbf{A}^\mathsf{T}\mathbf{An}$, which means that the average surface normal of our microsurface is $\mathbf{A}^\mathsf{T}\mathbf{An}$. However when θ_x or θ_y are non-zero, then in general $\mathbf{A}^\mathsf{T}\mathbf{An} \neq \mathbf{n}$ 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 θ_x and θ_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 (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.

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.

GOLDMAN, R. 2005. Curvature formulas for implicit curves and surfaces. *Comput. Aided Geom. Des. 22*, 7 (Oct.), 632–658.

HEITZ, E. 2014. Understanding the masking-shadowing function in microfacet-based brdfs. *Journal of Computer Graphics Techniques (JCGT) 3*, 2 (June), 32–91.

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.