

The SGGX Microflake Distribution

Eric Heitz^{1,2}

Jonathan Dupuy³

Cyril Crassin²

Carsten Dachsbacher¹

¹Karlsruhe Institute of Technology

²NVIDIA

³Univ. Montréal; LIRIS, Univ. Lyon 1

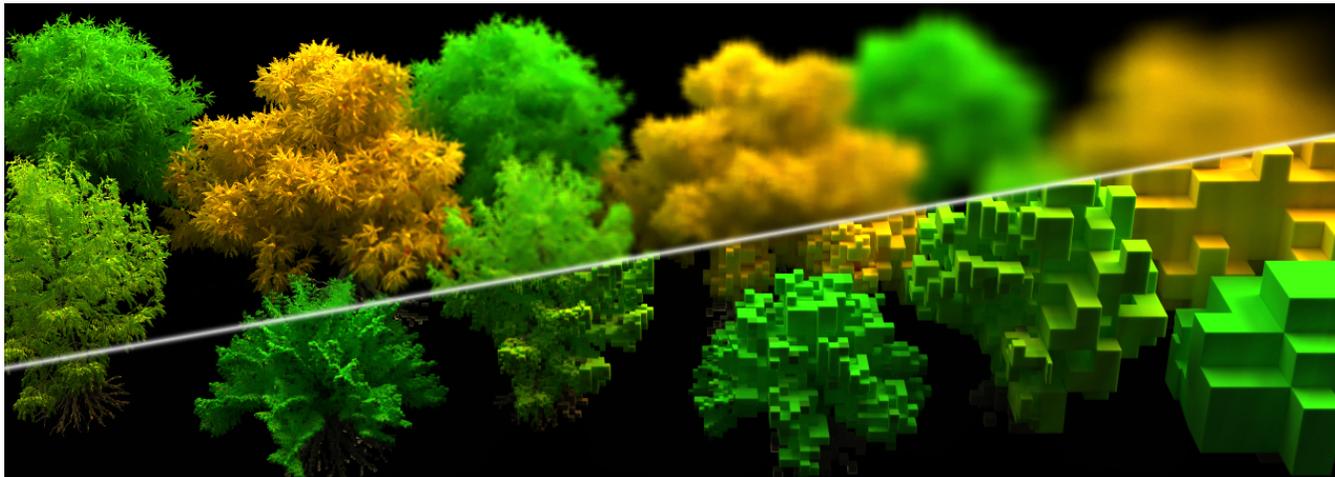


Figure 1: Top-left: rendering a voxelized forest at decreasing levels of detail (left to right). Bottom-right: visualization of the voxel structure at the matching resolutions. We use the SGGX microflake distribution to represent volumetric anisotropic materials. Our representation supports downscaling and interpolation, resulting in smooth and antialiased renderings at multiple scales.

Abstract

We introduce the *Symmetric GGX* (SGGX) distribution to represent spatially-varying properties of anisotropic microflake participating media. Our key theoretical insight is to represent a microflake distribution by the projected area of the microflakes. We use the projected area to parameterize the shape of an ellipsoid, from which we recover a distribution of normals. The representation based on the projected area allows for robust linear interpolation and prefiltering, and thanks to its geometric interpretation, we derive closed form expressions for all operations used in the microflake framework.

We also incorporate microflakes with diffuse reflectance in our theoretical framework. This allows us to model the appearance of rough diffuse materials in addition to rough specular materials. Finally, we use the idea of sampling the distribution of visible normals to design a perfect importance sampling technique for our SGGX microflake phase functions. It is analytic, deterministic, simple to implement, and one order of magnitude faster than previous work.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Ray tracing;

Keywords: microflake theory, global illumination, light transport.

1 Introduction

The importance of rendering volumetric models has dramatically increased in recent years. While geometry has been used as the standard rendering primitive for decades, it is hard to represent materials like hair, fur, or fabric as surfaces. Instead, using volumetric data to describe spatially varying microstructures has proven to be a promising alternative [Zhao et al. 2011]. Furthermore, volumetric representations are also well-suited for level-of-detail (LOD) techniques that replace complex visual data (e.g. distant fine geometry) with a more compact and efficient representation of its appearance. This is an important and pressing challenge as the complexity of scenes in production rendering has reached unprecedented heights.

A physically-based definition of volumetric scattering with arbitrary microstructures, the “microflake theory”, has recently been introduced by Jakob et al. [2010]. Analogous to the microfacet theory which has been formulated for surface materials (BRDFs), this theoretical framework is used to describe volumetric scattering. It provides a phase function deduced from a given set of statistically distributed microflakes, and ensures physical correctness by constraints for energy conservation and reciprocity. While this new framework opened up new possibilities for rendering, it is currently limited by the existing microflake representations, which lack closed form operators, e.g. for evaluating the phase function and importance sampling.

In this paper we introduce a novel microflake distribution which resolves these shortcomings and thus increases the practical benefits of the microflake framework in general.

Our SGGX distribution is the first that provides lightweight storage of microstructure representation, linear parameter interpolation, and analytical evaluation and importance sampling. SGGX parameters can be generated by converting from existing data in other microflake models [Zhao et al. 2011] or created from polygonal geometry, and can be filtered to obtain volumetric level-of-detail representations.

In particular our contributions are:

- A novel microflake representation to represent isotropic and anisotropic microstructures. Its parameters are simple to initialize, simple to convert from previous representations, and can be robustly interpolated.
- An extension of the microflake framework to incorporate not only specular but also diffuse microflakes.
- An analytic and deterministic importance sampling technique for the resulting phase functions, which is one order of magnitude faster than previous work.

We also explore how the SGGX distribution can be used in future multi-scale rendering applications. Such frameworks require filtering of both the microflake representation *and* the complex visibility effects emerging at coarser levels of detail from the organization of the finest details. We show that our method is able to accurately prefilter the microflake representation and discuss the challenges for future work.

2 Previous Work

Our approach connects volumetric filtering techniques to the physically-based rendering framework of the microflake formulation. The realization of our multi-scale framework is also inspired by recent LOD techniques for surface details.

Microflake Framework Jakob et al. [2010] were the first to introduce a physically-based radiative transfer framework for anisotropic participating media. It is based on a volume scattering model analogous to microfacet surface reflection models, but using oriented non-spherical particles in a volume. They also derive the constraints for reciprocity and energy conservation for the phase function. This framework has been used to reproduce the appearance of woven fabric designed at a micron-scale resolution [Zhao et al. 2011; Zhao et al. 2012]. Schröder et al. [2011] proposed a different volumetric approach using virtual scattering events computed from statistical distributions of fibers in woven materials. A recent course [Schröder et al. 2012] provides a comprehensive overview over reproducing the appearance of cloth. Note that the current microflake framework does not incorporate transmission scattering effects, which are especially important for realistic hair rendering [Marschner et al. 2003]. Hence, our model is also subject to this limitation.

Complexity of Microflake Rendering Unfortunately, using Monte Carlo path tracing requires hundreds of hours of computation time even for moderately complex scenes and resolution [Zhao et al. 2012]. Zhao et al. [2013] propose to accelerate rendering by precomputing and storing radiance transfer for repeated structures that are present in many volumetric materials (e.g. fabrics). While this technique provides significant speedup, its application is limited and it cannot be applied to generic non-repetitive assets.

Another strategy to accelerate rendering would be to design a multi-scale representation that is able to adapt the resolution of the data to the image resolution (or pixel footprint). Unfortunately, existing representations [Jakob et al. 2010; Zhao et al. 2011] are not suitable in this case: the distributions of normals (used to represent a collection of microflakes) lack a filtering operator, which in turn is required to obtain the parameters of the distribution across different scales. Another limitation is that they can represent fiber-like *or* surface-like material, but not both at the same time. Interpolating and filtering different materials is thus not possible. In contrast, our framework can describe and operate on both in a unified manner.

Lastly, they also lack practicability as, for example, analytic importance sampling is not possible and requires a costly rejection based algorithm. Furthermore, the projected area, which needs to be evaluated at each step during ray marching, is not computed analytically nor efficiently. Creating multi-scale representations for these models has been considered an important challenge [Bala 2014]. In this paper, we present a solution to the aforementioned problems related to microflake representations.

The GGX Distribution The idea of representing a distribution of normals (of microfacets or microflakes) using an ellipsoid shape was first introduced by Trowbridge and Reitz [1975] in the physics literature. Neyret derived an equivalent distribution to represent volumetric materials [Neyret 1995; Neyret 1998]. He defined the associated ellipsoid surface as the set of points P such that $P^T Q P = 1$ (also called tensor ellipsoid), where Q is a 3×3 symmetric positive definite matrix. Neyret proposed to interpolate and filter the coefficients of Q^{-1} . Unfortunately these operations do not preserve the roughness of the input distributions and often result in obviously wrong results (see Fig. 10). Later, the same distribution was independently proposed by Walter et al. [2007] and named GGX; it is nowadays widely used in microfacet BRDFs [Hill et al. 2014]. However, the GGX distribution uses only the upper part of the ellipsoid according to a local frame and does not allow for filtering or interpolation when the different input distributions are not defined in the same frame.

Filterable Distribution of Normals The LEAN representation [Olano and Baker 2010; Dupuy et al. 2013] for normal or displacement maps enables filtering and interpolation, and served as inspiration for our work. The core idea of LEAN is to find a space where the anisotropic distribution of microfacet normals can be sufficiently well represented with a small set of parameters. LEAN data can be interpolated linearly as it is represented as a symmetric 2×2 matrix, the covariance matrix of the distribution of slopes. The key insight of our representation is that the “volumetric analogon” to this distribution is the projected area of the microflakes, and that this again can be compactly represented with a symmetric matrix.

3 Background: Microflake Theory

In this section we briefly summarize the practical aspects of the microflake theory [Jakob et al. 2010]; Table 1 shows our notation. A microflake material is defined by a *distribution of normals* (in accordance with previous work we use the abbreviation NDF which stands for normal distribution function) and during rendering the model is used to compute the volumetric *attenuation* and the *phase function*. The choice and the implementation of the integrator (e.g. path tracing) is independent and out of the scope of this paper; we refer the reader to [Pharr and Humphreys 2010].

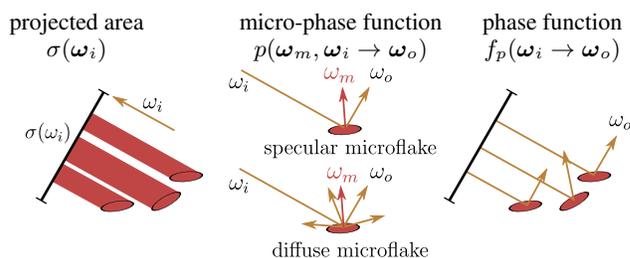


Figure 2: Illustration of the notation used in microflake theory.

ω_i, ω_o	incident and outgoing directions
ω_m	microflake normal
$D(\omega_m)$	distribution of normals (NDF)
$D\omega_i(\omega_m)$	distribution of visible normals (VNDF)
α	direction-independent albedo of microflakes
$p(\omega_m, \omega_i \rightarrow \omega_o)$	phase function of one microflake
$f_p(\omega_i \rightarrow \omega_o)$	phase function of all microflakes
ρ	volumetric density
$\sigma(\omega_i)$	microflake projected area
$\sigma_t(\omega_i)$	volumetric attenuation coefficient
$\sigma_s(\omega_i)$	volumetric scattering coefficient
S	SGGX parameters (3 × 3 matrix)
$(- \cdot -)$	dot product
$\langle -, - \rangle$	clamped dot product

Table 1: The notation used in this paper. Note that all quantities can be spatially-varying, which we omit for better readability.

3.1 Microflake Volumes

The Anisotropic RTE The light transport in an anisotropic participating medium is described by the anisotropic Radiative Transfer Equation (RTE) [Jakob et al. 2010]:

$$\begin{aligned} & (\omega_i \cdot \nabla) L(\omega_i) + \sigma_t(\omega_i) L(\omega_i) \\ & = \sigma_s(\omega_i) \int_{\Omega} f_p(\omega_i \rightarrow \omega_o) L(\omega_o) d\omega_o + Q(\omega_i). \end{aligned} \quad (1)$$

Note that we omit spatial parameters in the quantities arising in this equation, which are further explained in Table 1.

Microflake Distribution The microflakes' statistical orientation is defined by the distribution of normals (NDF) D . Previous work [Jakob et al. 2010; Zhao et al. 2011] used normalized NDFs, i.e. $\int_{\Omega} D(\omega_m) d\omega_m = 1$. However, this normalization constraint is not mandatory for the framework as we show in Section 4.2.

Microflake Projected Area The volumetric attenuation $\sigma_t(\omega_i)$ and scattering $\sigma_s(\omega_i)$ coefficients are used by the integrator to sample distances within the volume, and to evaluate visibility. In a microflake volume we have:

$$\sigma_t(\omega_i) = \rho \sigma(\omega_i), \quad (2)$$

$$\sigma_s(\omega_i) = \alpha \rho \sigma(\omega_i), \quad (3)$$

where ρ is the volume density, α is the direction-independent albedo (as we assume that all microflakes at a point in the volume have the same material), and $\sigma(\omega_i)$ is the projected area of the microflakes in direction ω_i :

$$\sigma(\omega_i) = \int_{\Omega} \langle \omega_i, \omega_m \rangle D(\omega_m) d\omega_m. \quad (4)$$

The evaluation of the projected area is important for our representation and further discussed in Section 4.

Microflake Reflectance The microflake phase function $f_p(\omega_i \rightarrow \omega_o)$ represents the interaction of light with a collection of microflakes present in the volume. Each individual microflake can be seen as a small two-sided surface-like patch with its own BRDF. This BRDF, weighted with the cosine of the incident direction, is the *phase function* of the microflake $p(\cdot, \cdot)$. Note that we name $p(\cdot, \cdot)$ phase function, as it is a probability distribution function (PDF) used for computing outgoing directions upon scattering. To simplify notation in this paper, we assume that the phase

function $p(\cdot, \cdot)$ of a single microflake is either based on a purely diffuse or purely specular BRDF with a direction-independent albedo α . However, nothing prevents us from using a blend of diffuse and specular microflakes and/or direction-dependent albedos.

Microflake Phase Function The phase function for a collection of microflakes is

$$f_p(\omega_i \rightarrow \omega_o) = \frac{1}{\sigma(\omega_i)} \int_{\Omega} p(\omega_m, \omega_i \rightarrow \omega_o) \langle \omega_i, \omega_m \rangle D(\omega_m) d\omega_m. \quad (5)$$

The phase function is used to evaluate the light scattering at one point in the medium as

$$L(\omega_i) = \alpha \int_{\Omega} f_p(\omega_i \rightarrow \omega_o) L(\omega_o) d\omega_o. \quad (6)$$

The implementation of a microflake material must provide two additional functions: *evaluation* and *sampling* of $f_p(\omega_i \rightarrow \omega_o)$. We define these operators for the SGGX distribution in Section 5.

3.2 Microflake Constraints

A physically-based microflake material is both energy conserving and reciprocal. *Energy conservation* requires a normalized phase function, i.e. the distribution of outgoing directions ω_o given an incident direction ω_i is a PDF:

$$\int_{\Omega} f_p(\omega_i \rightarrow \omega_o) d\omega_o = 1. \quad (7)$$

The *reciprocity* constraint relates the projected area σ and the phase function f_p (with a direction-independent microflake albedo α):

$$\sigma(\omega_i) f_p(\omega_i \rightarrow \omega_o) = \sigma(\omega_o) f_p(\omega_o \rightarrow \omega_i). \quad (8)$$

4 The SGGX Microflake Distribution

In this section we introduce our new representation for SGGX distributions and discuss the initialization of its parameters.

4.1 Preliminary Considerations

Our new representation of SGGX distributions is motivated by three important observations.

1) Ellipsoidal NDF As shown in Fig. 3, the GGX distribution is the distribution of normals (NDF) of an ellipsoid clamped to a hemisphere. The SGGX distribution is simply the NDF of the ellipsoid over the entire domain (S stands for symmetric).

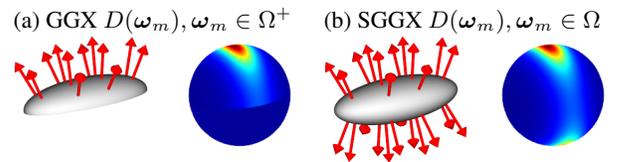


Figure 3: Ellipsoidal distributions. The SGGX distribution is a symmetrized GGX, i.e. the distribution of normals of the surface of a non-clamped ellipsoid. The color-coded spheres illustrate the distributions of normals.

2) A distribution of normals does not need to be normalized In Section 5 we show that a microflake phase function f_p depends on the distribution of *visible* microflakes $D\omega_i$, which is always normalized, even if D is not. Intuitively, scaling D by an arbitrary factor is equivalent to scaling the number, or the size, of microflakes in a volume; both still represent a valid medium. Consequently, the normalization of D and/or the evaluation of its norm are unnecessary for the derivation of a microflake model.

3) The projected area should be conserved The projected area is an important aspect in both microfacet and microflake theory as it is the normalization factor of the resulting BRDFs and phase functions. Furthermore, in microflake theory, it affects the probability of light-matter interaction and thus the opacity of the material. Any filtering operation should conserve this quantity.

From these observations we conclude that representing a set of microflakes by its projected area, i.e. the sum over the projected areas of all individual microflakes, instead of its distribution of normals, is beneficial for operations such as interpolation or filtering. However, we have to be able to recover the NDF from the projected area.

4.2 Novel Definition of the SGGX Distribution

We introduce the SGGX distribution that is defined by its projected area, from which its NDF can be recovered.

Definition of the SGGX Distribution Based on Projected Area Our SGGX distribution is best visualized in its eigenspace. We define it as a 3×3 symmetric positive definite matrix S such that

$$S = (\omega_1, \omega_2, \omega_3) \begin{pmatrix} S_{11} & 0 & 0 \\ 0 & S_{22} & 0 \\ 0 & 0 & S_{33} \end{pmatrix} (\omega_1, \omega_2, \omega_3)^T, \quad (9)$$

where $S_{11} = \sigma^2(\omega_1)$, $S_{22} = \sigma^2(\omega_2)$ and $S_{33} = \sigma^2(\omega_3)$ are positive eigenvalues that are equal to the squared projected areas of the ellipsoid in the directions given by the orthonormal eigenvectors (ω_1 , ω_2 and ω_3) as shown in Fig. 4. The eigenvectors are also the principal axes of the ellipsoid.

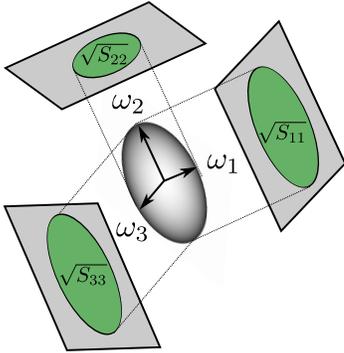


Figure 4: Eigenvectors and eigenvalues of the matrix S . The projected areas (area of green surfaces) are the square roots of the matrix eigenvalues.

Definition of the SGGX Distribution in the Canonical Basis We can write the development of Eq. 9 to get the expression of

the SGGX matrix in the canonical basis

$$S = \begin{pmatrix} S_{xx} & S_{xy} & S_{xz} \\ S_{xy} & S_{yy} & S_{yz} \\ S_{xz} & S_{yz} & S_{zz} \end{pmatrix}, \quad (10)$$

where the 6 coefficients $S_{xx}, S_{yy}, S_{zz}, S_{xy}, S_{xz}$ and S_{yz} are the parameters that we use for computations, as we show in Section 4.4.

Projected Area Given S (Eq. 9) we can compute the projected area as

$$\sigma(\omega_i) = \int_{\Omega} \langle \omega_i, \omega_m \rangle D(\omega_m) d\omega_m = \sqrt{\omega_i^T S \omega_i}. \quad (11)$$

From this we can observe that interpolating the matrix S is equivalent to interpolating the squared projected area $\sigma(\omega_i)^2$. This ensures a convex and thus robust interpolation of the projected area, as we discuss in Section 4.4.

Distribution of Normals For an ellipsoid defined by S (Eq. 9), we can also obtain the distribution of normals (see the supplemental material for a derivation) as:

$$D(\omega_m) = \frac{1}{\pi \sqrt{|S|} (\omega_m^T S^{-1} \omega_m)^2}. \quad (12)$$

The evaluation of D can be implemented efficiently, as S is a 3×3 symmetric positive definite matrix and is thus analytically invertible. Note that in general D is not normalized, i.e. $\int_{\Omega} D(\omega_m) d\omega_m \neq 1$, and its norm has no closed-form expression. However, this is not an issue for microflake models as we explained before.

4.3 Initializing the SGGX Representation

General Approach The form of Eq. 9 enables us to easily initialize the SGGX matrix S as long as its principal directions ω_1 , ω_2 , and ω_3 and the respective projected areas $\sigma(\omega_1)$, $\sigma(\omega_2)$, and $\sigma(\omega_3)$ are available. We use this approach to initialize both surface- and fiber-like materials, convert existing data to SGGX, and design a parameter estimation procedure for arbitrary input distributions.

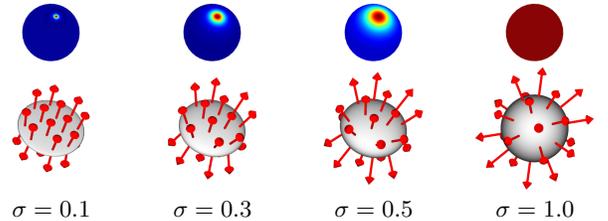


Figure 5: SGGX Initialization. Initializing a surface-like SGGX distribution with roughness σ (top-row: normal distribution, bottom-row: ellipsoids).

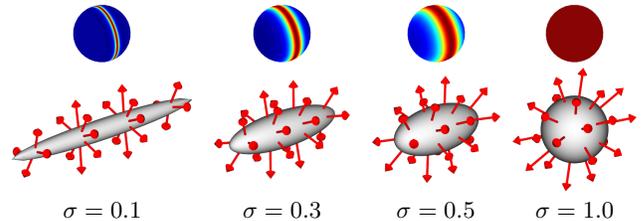


Figure 6: SGGX Initialization. Initializing a fiber-like SGGX distribution with roughness σ (top-row: normal distribution, bottom-row: ellipsoids).

Surface-like Distributions A surface-like SGGX distribution is defined by its normal direction ω_3 and a roughness parameter $\sigma \in [0, 1]$, which is the projected area onto orthogonal tangent directions ω_1 and ω_2 . The projected area onto ω_3 is 1 by convention. The eigenvalues are

$$\sigma^2(\omega_1) = \sigma^2, \quad \sigma^2(\omega_2) = \sigma^2, \quad \sigma^2(\omega_3) = 1. \quad (13)$$

By expanding Eq. 9 with $\omega_3 = (x, y, z)$, we obtain

$$S = \begin{pmatrix} x^2 & xy & xz \\ xy & y^2 & yz \\ xz & yz & z^2 \end{pmatrix} + \sigma^2 \begin{pmatrix} y^2 + z^2 & -xy & -xz \\ -xy & x^2 + z^2 & -yz \\ -xz & -yz & x^2 + y^2 \end{pmatrix}. \quad (14)$$

Note that the parameter σ matches the roughness parameter (usually denoted α) of the hemispherical GGX distribution used in microfacet BRDFs [Walter et al. 2007]. The shape of the ellipsoid varies from a disk ($\sigma = 0$), where D is a delta Dirac distribution, to a sphere ($\sigma = 1$) with a uniform distribution of normals, as shown in Fig. 5.

Fiber-like Distributions A fiber-like SGGX distribution is defined by its tangent direction ω_3 and a roughness parameter $\sigma \in [0, 1]$, which is the projected area onto ω_3 . The projected area onto orthogonal normal directions ω_1 and ω_2 is 1 by convention. The eigenvalues are

$$\sigma^2(\omega_1) = 1, \quad \sigma^2(\omega_2) = 1, \quad \sigma^2(\omega_3) = \sigma^2. \quad (15)$$

By expanding Eq. 9 with $\omega_3 = (x, y, z)$, we obtain

$$S = \sigma^2 \begin{pmatrix} x^2 & xy & xz \\ xy & y^2 & yz \\ xz & yz & z^2 \end{pmatrix} + \begin{pmatrix} y^2 + z^2 & -xy & -xz \\ -xy & x^2 + z^2 & -yz \\ -xz & -yz & x^2 + y^2 \end{pmatrix}. \quad (16)$$

The shape of the ellipsoid varies from a cylinder ($\sigma = 0$) to a sphere ($\sigma = 1$), as shown in Fig. 6.

Converting Existing Distributions Zhao et al. [2011] represent fiber-like materials with angular Gaussian distributions D^{zhao} parameterized by a tangent direction $\omega_3 = (x, y, z)$ and a roughness coefficient γ . We can easily convert such data to SGGX distributions. To this end, we first evaluate the projected areas of D^{zhao} onto the tangent direction ω_3 and onto the orthogonal directions $(\omega_1, \omega_2) \perp \omega_3$ by using Eq. 4:

$$\begin{aligned} \sigma(\omega_3) &= \int_{\Omega} \langle \omega_3, \omega_m \rangle D^{\text{zhao}}(\omega_m) d\omega_m, \\ \sigma(\omega_1) &= \sigma(\omega_2) = \int_{\Omega} \langle \omega_1, \omega_m \rangle D^{\text{zhao}}(\omega_m) d\omega_m. \end{aligned} \quad (17)$$

We then obtain a SGGX distribution using Eq. 9. The results of two conversions are shown in Fig. 7. Note that the SGGX distribution has a sharper peak and a wider tail than the angular Gaussian distribution, which is a well-known property of the GGX distribution [Walter et al. 2007].

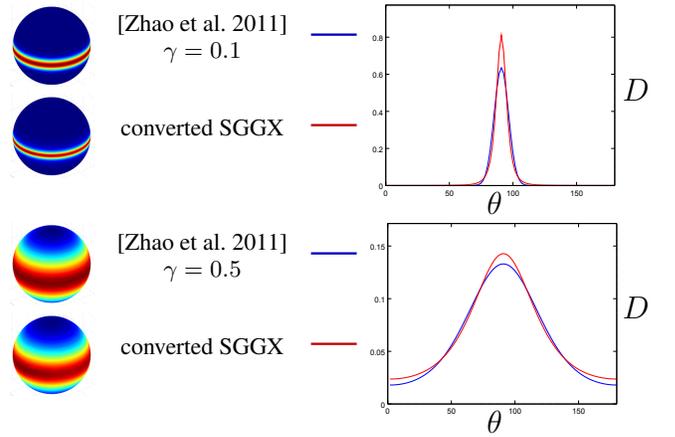


Figure 7: SGGX conversion. We obtain a fast conversion from the distribution of Zhao et al. to SGGX by computing the projected areas in the principal directions.

Parameter Estimation from Arbitrary Distributions Note that, in general, estimating the parameters of spherical distributions requires costly non-linear optimization procedures with parameter space exploration [Xu et al. 2013]. In contrast, our parameter estimation procedure based on the geometric properties of SGGX distributions is efficient, deterministic, and simple to implement. The goal of the parameter estimation procedure is to compute 3 eigenvectors $(\omega_1, \omega_2, \omega_3)$ and 3 associated projected areas $(\sigma_1, \sigma_2, \sigma_3)$ from an input spherical distribution D . We recall that a valid microflake distribution D should be symmetric, i.e. $D(\omega_m) = D(-\omega_m)$.

First, we extract the eigenvectors. Because of the ellipsoidal symmetry, the eigenvectors of a SGGX distribution are the same than the eigenvectors of its 3D covariance matrix ¹

$$\Sigma = \begin{pmatrix} \mathbb{E}[x^2] & \mathbb{E}[xy] & \mathbb{E}[xz] \\ \mathbb{E}[xy] & \mathbb{E}[y^2] & \mathbb{E}[yz] \\ \mathbb{E}[xz] & \mathbb{E}[yz] & \mathbb{E}[z^2] \end{pmatrix}, \quad (18)$$

where the expectations are the second-order moments of the coordinates $\omega_m = (x, y, z)$. For instance $\mathbb{E}[x^2] = \int_{\Omega} x^2 D(\omega_m) d\omega_m$ and $\mathbb{E}[xy] = \int_{\Omega} xy D(\omega_m) d\omega_m$. We compute the covariance matrix Σ numerically and extract its eigenvectors $(\omega_1, \omega_2, \omega_3)$. This last operation is simple because the matrix is of size 3×3 and symmetric positive definite. Next, we obtain the associated eigenvalues by computing the projected areas $\sigma(\omega_1)$, $\sigma(\omega_2)$, and $\sigma(\omega_3)$ of the distribution on these three directions, by using Eq. 4. Finally, we combine the eigenvectors and the eigenvalues following Eq. 9. The estimated SGGX distribution has exactly the same projected area as the input data in these three directions but not necessarily in other directions.

An important property of our parameter estimation procedure is that by construction the estimation is exact if the input distribution is a SGGX distribution (Fig. 8). Indeed, if the input can be represented exactly with a SGGX distribution, i.e. with 3 eigenvectors and 3 eigenvalues, then the output of our prefiltering algorithm is exact, as it retrieves these eigenvectors and eigenvalues. This property guarantees robust and accurate parametric estimation. As illustrated in Fig. 9, our algorithm captures the scaling and the anisotropy of the input distribution. In contrast, Neyret's [1998] filtering procedure fails in both respects.

¹Note that the SGGX matrix and its covariance matrix have the same eigenvectors but are not equal in general: $S \neq \Sigma$. They are equal only in special cases, e.g. perfect spheres or perfect disk-like ellipsoids.

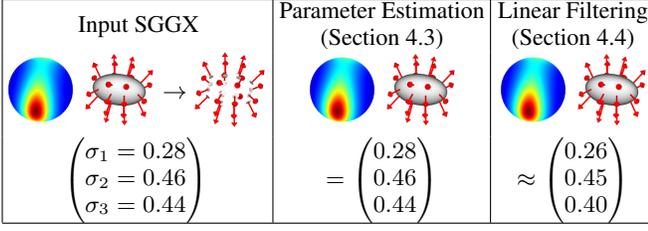


Figure 8: Parameter estimation validation. *If the input distribution can be represented by a SGGX distribution, then our parameter estimation procedure outputs exactly this SGGX distribution. In contrast, linearly filtering the input data does not provide the best parameter estimation but it remains a fair and simple approximation. The σ values are the projected areas onto the eigenvectors.*

Input	Parameter Estimation SGGX (4.3)	Linear Filtering Neyret SGGX (4.4)	
		$\frac{1}{N} \sum_{n=1..N} Q_n^{-1}$ 	$\frac{1}{N} \sum_{n=1..N} S_n$
$\begin{pmatrix} \sigma_x = 1.54 \\ \sigma_y = 4.88 \\ \sigma_z = 4.81 \end{pmatrix}$	$\begin{pmatrix} 1.55 \\ 4.84 \\ 4.85 \end{pmatrix}$	$\begin{pmatrix} 5.97 \\ 25.48 \\ 18.87 \end{pmatrix}$	$\begin{pmatrix} 1.73 \\ 4.89 \\ 4.81 \end{pmatrix}$
$\begin{pmatrix} \sigma_x = 2.13 \\ \sigma_y = 1.25 \\ \sigma_z = 4.03 \end{pmatrix}$	$\begin{pmatrix} 2.17 \\ 1.27 \\ 4.07 \end{pmatrix}$	$\begin{pmatrix} 2.53 \\ 2.35 \\ 4.18 \end{pmatrix}$	$\begin{pmatrix} 2.51 \\ 1.29 \\ 4.14 \end{pmatrix}$

Figure 9: Parameter estimation from arbitrary distributions. *We robustly estimate the parameters of various input distributions. The σ values are the projected areas onto the canonical directions x , y and z .*

4.4 Memory Representation of the SGGX Parameters

Compact Storage Volumetric data can easily exceed the available memory. Thus, a compact representation for storage (and for use during rendering) is desirable. Since the 6 coefficients of the matrix S from Eq. 10 represent squared values, a linear quantization would be wasteful. We found that the following transformation yields a set of parameters which are well-suited for storing SGGX parameters:

$$\begin{aligned} \sigma_x &= \sqrt{S_{xx}}, & \sigma_y &= \sqrt{S_{yy}}, & \sigma_z &= \sqrt{S_{zz}} \\ r_{xy} &= \frac{S_{xy}}{\sqrt{S_{xx}S_{yy}}}, & r_{xz} &= \frac{S_{xz}}{\sqrt{S_{xx}S_{zz}}}, & r_{yz} &= \frac{S_{yz}}{\sqrt{S_{yy}S_{zz}}}. \end{aligned} \quad (19)$$

Note that the σ -values are linearly distributed in $[0, 1]$, and the r -parameters are distributed linearly in $[-1, 1]$. In our implementation, we store each parameter using 1 Byte. Thus, our SGGX representation requires 6 Bytes per voxel.

Linear Interpolation The compact representation from Eq. 19 cannot be linearly interpolated because the resulting SGGX matrix is not necessarily positive definite. However, the 6 parameters S_{xx} ,

S_{yy} , S_{zz} , S_{xy} , S_{xz} and S_{yz} from Eq. 10 can be linearly interpolated. When fetching the values during rendering, we first reconstruct the 6 parameters of S :

$$\begin{aligned} S_{xx} &= \sigma_x^2, & S_{yy} &= \sigma_y^2, & S_{zz} &= \sigma_z^2 \\ S_{xy} &= r_{xy}\sigma_x\sigma_y, & S_{xz} &= r_{xz}\sigma_x\sigma_z, & S_{yz} &= r_{yz}\sigma_y\sigma_z. \end{aligned} \quad (20)$$

With these we can safely interpolate: the convex combinations of symmetric positive definite matrices is always a symmetric positive definite matrix. Thus, linearly interpolating the SGGX matrices provides always a (mathematically) valid result. Furthermore, in Section 4.2 we have shown that averaging the coefficients of SGGX matrices is equivalent to averaging the squared projected areas. The projected area of an interpolated distribution is a convex combination of the inputs. This property ensures that the interpolated distribution is always well-defined, visually well-behaved, and accurate. Fig. 10 visually compares our interpolation scheme to Neyret’s. His method fails at interpolating fiber-like materials because it does not preserve the projected area (the roughness of the material). In this example, it blends fiber-like ellipsoids into an almost flat surface-like ellipsoid, i.e. a highly specular surface-like material. In contrast, our linear interpolation preserves the average projected area and the main directions. However, while our interpolation scheme preserves the average appearance, it slightly smoothens the results. Note that LEAN/LEADR [Olano and Baker 2010; Dupuy et al. 2013] interpolation schemes, which are based on a similar idea, share this limitation.

Linear Prefiltering The ability to initialize a new SGGX distribution by blending multiple distributions is a key operation required for downsampling volumetric data. To achieve this, we can estimate the parameters of the blended distribution with the procedure introduced in Section 4.3. However, in practice, we perform 3D MIP mapping, i.e. we downsample the data by linearly filtering the matrices S and building a LOD hierarchy. Note that in theory linear filtering does not guarantee an optimal parameter estimation as shown in Fig. 8. However, it is much simpler to implement and reasonably accurate for practical purposes as shown in Fig. 9. More comparisons are available in our supplemental material.

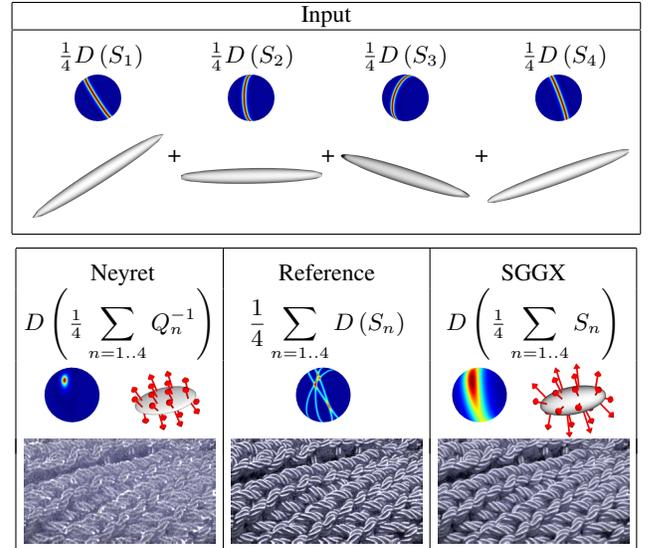


Figure 10: Interpolation. *We compare Neyret’s interpolation and SGGX interpolation against a reference configuration.*

5 The SGGX Phase Functions

In this section we develop microflake phase functions based on the distribution of visible normals (Section 5.1). This concept was first introduced in the context of microfacet BRDFs [Heitz 2014]. Based on this we rederive the phase function for specular microflakes from the result of Jakob et al. [2010] and also derive a new phase function for diffuse microflakes (Section 5.2). One advantage of Heitz’ construction is that it enables us to establish a link between the distribution of visible normals and the phase function, which we leverage to develop an efficient importance sampling technique for SGGX phase functions (Section 5.3).

5.1 The Distribution of Visible Normals

The distribution of visible normals (VNDF) for a collection of microflakes is

$$D_{\omega_i}(\omega_m) = \frac{\langle \omega_i, \omega_m \rangle D(\omega_m)}{\int_{\Omega} \langle \omega_i, \omega_m \rangle D(\omega_m) d\omega_m} = \frac{\langle \omega_i, \omega_m \rangle D(\omega_m)}{\sigma(\omega_i)}, \quad (21)$$

where the projected area $\sigma(\omega_i)$ is the normalization factor of the distribution² and ensures that it is normalized (i.e. it is a PDF):

$$\int_{\Omega} D_{\omega_i}(\omega_m) d\omega_m = 1. \quad (22)$$

Fig. 11 illustrates the NDF and the VNDF of a SGGX distribution.

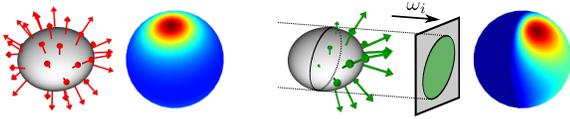


Figure 11: The distribution of normals D (left), and the distribution of visible normals D_{ω_i} (right) of a SGGX distribution.

5.2 Construction of the Phase Functions

In this section, we derive the SGGX phase function starting from the VNDF. When a ray with direction ω_i intersects a microflake, the normal ω_m of this microflake is chosen according to the PDF $D_{\omega_i}(\omega_m)$. The individual microflake’s phase function $p(\omega_m, \omega_i \rightarrow \omega_o)$ is the PDF used to determine the reflected direction ω_o of the ray (depending on the incident direction ω_i and ω_m). We can rewrite Eq. 5 with the distribution of visible normals:

$$f_p(\omega_i \rightarrow \omega_o) = \int_{\Omega} p(\omega_m, \omega_i \rightarrow \omega_o) D_{\omega_i}(\omega_m) d\omega_m. \quad (23)$$

Intuitively, the specular and diffuse SGGX phase functions, which we derive next, describe how a specular, respectively diffuse, ellipsoid would reflect the incident light (shown in Fig. 12).

²Note that this normalization factor plays the same role and is defined under the same assumption as the Smith masking function in microfacet BRDFs: it assumes that the visibility and the orientation of the normals are independent [Heitz 2014].

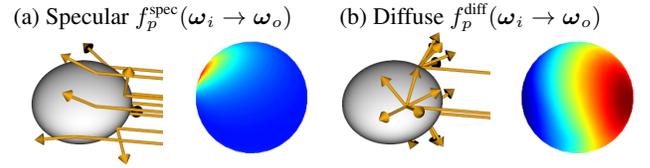


Figure 12: The SGGX phase functions $f_p(\omega_i \rightarrow \omega_o)$ are given by the distributions of rays reflected by the surface of the ellipsoid.

Specular Phase Function The SGGX phase function for specular microflakes is the distribution of reflection directions of an ellipsoid with a specular surface (Fig. 12(a)). For specular microflakes the micro-phase function is $p^{\text{spec}}(\omega_m, \omega_i \rightarrow \omega_o) = \frac{\delta_{\omega_h}(\omega_m)}{4|\omega_i \cdot \omega_h|}$, where ω_h is the half-vector. Inserting it in Eq. 23 we obtain the same result as Jakob et al. [2010]:

$$f_p^{\text{spec}}(\omega_i \rightarrow \omega_o) = \frac{D(\omega_h)}{4\sigma(\omega_i)}, \quad (24)$$

which satisfies both energy conservation (Eq. 7) and reciprocity (Eq. 8). f_p^{spec} can be evaluated using the analytic form of D from Eq. 12 and of σ from Eq. 11.

Diffuse Phase Function Similarly, the SGGX phase function for diffuse microflakes is the distribution of directions reflected by an ellipsoid with a Lambertian surface (Fig. 12(b)). In this case, the microflakes’ micro-phase function is $p^{\text{diff}}(\omega_m, \omega_i \rightarrow \omega_o) = \frac{1}{\pi} \langle \omega_o, \omega_m \rangle$. Inserting it into Eq. 23 yields:

$$\begin{aligned} f_p^{\text{diff}}(\omega_i \rightarrow \omega_o) &= \frac{1}{\pi} \int_{\Omega} \langle \omega_o, \omega_m \rangle D_{\omega_i}(\omega_m) d\omega_m \\ &= \frac{1}{\pi\sigma(\omega_i)} \int_{\Omega} \langle \omega_o, \omega_m \rangle \langle \omega_i, \omega_m \rangle D(\omega_m) d\omega_m, \end{aligned} \quad (25)$$

which also satisfies energy conservation and reciprocity. The evaluation of f_p^{diff} is more involved than for specular microflakes. In Section 5.4 we present a practical evaluation procedure.

5.3 Importance Sampling the Phase Function

Distribution of Visible Normals Heitz and d’Eon [2014] showed that microfacet BRDF models can be efficiently importance sampled by first sampling a normal from the distribution of visible normals D_{ω_i} , and then sampling the micro-BRDF of the material aligned with this normal. The same idea can be applied to importance sampling microflake phase functions: if we importance sample the VNDF D_{ω_i} to generate a sample ω_m , and then sample an outgoing direction ω_o with the micro-phase function $p(\omega_m, \omega_i \rightarrow \omega_o)$, then ω_o follows the PDF given by the dot product of $D_{\omega_i}(\omega_m)$ and $p(\omega_m, \omega_i \rightarrow \omega_o)$. Note that this dot product is the microflake phase function $f_p(\omega_m, \omega_i \rightarrow \omega_o)$ (Eq. 23).

For microfacet BRDFs, this importance sampling technique is not perfect as it does not account for the shadowing probability which is then represented in the weight of the sample. However, for microflake phase functions there is no shadowing³ but only masking.

³Shadowing in microfacet BRDFs removes rays that would scatter multiple times on the microsurface, i.e. only single scattering is modelled. However, in microflake theory, further interactions of a ray with the material will be accounted for when it scatters into the outgoing direction ω_o , i.e. multiple scattering is automatically computed by the integrator. This explains the absence of shadowing and the unusual reciprocity constraint in Eq. 8.

Algorithm 1 Importance sampling the VNDF with SGGX

- 1: **function** SAMPLEVNDF($\omega_k, S, \mathcal{U}_1, \mathcal{U}_2$)
 - 2: compute an orthonormal basis ($\omega_k, \omega_j, \omega_i$) around ω_i
 - 3: project S in this basis

$$S^{kji} = \begin{pmatrix} S_{kk} & S_{kj} & S_{ki} \\ S_{kj} & S_{jj} & S_{ji} \\ S_{ki} & S_{ji} & S_{ii} \end{pmatrix} = \begin{pmatrix} \omega_k^T S \omega_k & \omega_k^T S \omega_j & \omega_k^T S \omega_i \\ \omega_j^T S \omega_k & \omega_j^T S \omega_j & \omega_j^T S \omega_i \\ \omega_i^T S \omega_k & \omega_i^T S \omega_j & \omega_i^T S \omega_i \end{pmatrix}$$
 - 4: compute vectors $M_k = \begin{pmatrix} \frac{|S^{kji}|}{\sqrt{S_{jj}S_{ii}-S_{ji}^2}} \\ 0 \\ 0 \end{pmatrix}$,

$$M_j = \frac{1}{\sqrt{S_{ii}}} \begin{pmatrix} -\frac{S_{ki}S_{ji}-S_{kj}S_{ii}}{\sqrt{S_{jj}S_{ii}-S_{ji}^2}} \\ \sqrt{S_{jj}S_{ii}-S_{ji}^2} \\ 0 \end{pmatrix}, M_i = \frac{1}{\sqrt{S_{ii}}} \begin{pmatrix} S_{ki} \\ S_{ji} \\ S_{ii} \end{pmatrix}$$
 - 5: generate random 3D point (u, v, w) on the visible sphere

$$u = \sqrt{\mathcal{U}_1} \cos(2\pi \mathcal{U}_2)$$

$$v = \sqrt{\mathcal{U}_1} \sin(2\pi \mathcal{U}_2)$$

$$w = \sqrt{1-u^2-v^2}$$
 - 6: compute normal and rotate to world space

$$\omega_m^{kji} = \frac{uM_k+vM_j+wM_i}{\|uM_k+vM_j+wM_i\|}$$

$$\omega_m = (\omega_k \omega_j \omega_i) \omega_m^{kji}$$
 - 7: **return** ω_m
 - 8: **end function**
-

Consequently, this importance sampling technique is perfect and the weight of the samples is always 1.

The crucial step of the importance sampling is the sampling of the distribution of visible normals, which we describe next.

Importance Sampling with the SGGX Distribution As we will see, the fact that the SGGX distribution is based on an ellipsoid to represent the distribution of normals will enable us to efficiently importance sample the resulting phase functions. The sampling of the VNDF is detailed in Algorithm 1 and the final steps of our sampling algorithm are illustrated in Fig. 13.

Sampling the VNDF First, we compute an orthonormal basis ($\omega_k, \omega_j, \omega_i$) aligned with the incident direction ω_i and rotate the SGGX matrix S in this basis yielding a new SGGX matrix S^{kji} . By using a Cholesky decomposition we can transform 3D points on a unit-sphere to points on the ellipsoid; the normal of these points can be computed as $\omega_m^{kji} = \frac{uM_k+vM_j+wM_i}{\|uM_k+vM_j+wM_i\|}$, where the vectors M_k, M_j , and M_i are computed from the Cholesky decomposition. To account for visibility, we generate the point (u, v, w) on the visible hemisphere. We achieve this by using two random numbers to sample a 2D point (u, v) on the unit disk and we project it onto the sphere to obtain its third coordinate w . Finally, we rotate the normal ω_m^{kji} back to world space. More details about this derivation and our implementation are provided in the supplemental material.

Importance Sampling the Specular Phase Function To sample f_p^{spec} , we generate a sample ω_m from D_{ω_i} and reflect ω_i at ω_m to generate the outgoing direction $\omega_o = -\omega_i + 2\omega_m \langle \omega_i, \omega_m \rangle$ (Fig. 13a).

Importance Sampling the Diffuse Phase Function To sample f_p^{diff} , we generate a sample ω_m from D_{ω_i} and sample a diffuse reflected direction ω_o in the hemisphere given by ω_m (Fig. 13b).

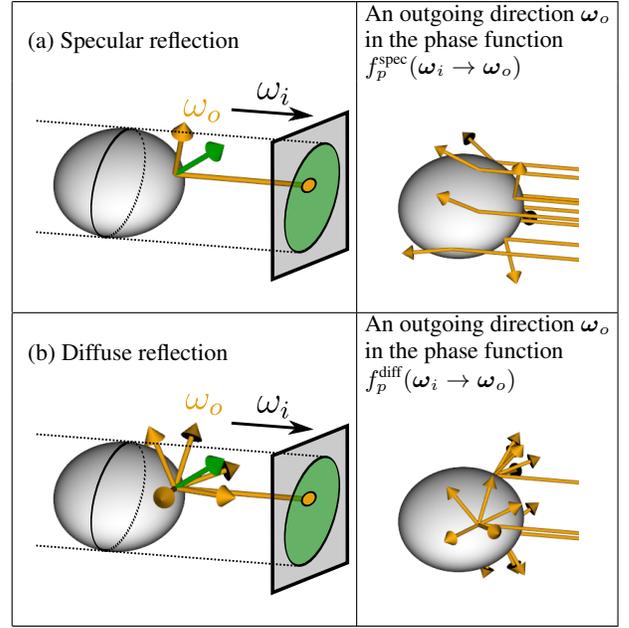


Figure 13: Computing the outgoing directions in our importance sampling algorithm.

Validation We validated our importance sampling technique experimentally with the χ^2 test provided by Mitsuba [Jakob 2010]. The test shows that the distribution generated by importance sampling effectively converges towards the correct phase function.

Comparison Our SGGX importance sampling scheme has beneficial properties and significantly improved performance compared to previous specular microflake phase functions:

	[Jakob et al. 2010]	[Zhao et al. 2011]	SGGX
analytic	✗	✓	✓
deterministic	✓	✗	✓
2 random numbers	✓	✗	✓
performance	✗	✗	×10

5.4 Evaluating the Diffuse Phase Function

As mentioned above, the evaluation of the diffuse phase function requires the computation of the integral in Eq. 25 which, unfortunately, is not possible analytically. A numerical evaluation would introduce a (considerable) bias unless the integration is very accurate and thus very costly. In the context of Monte Carlo rendering (which we target), we can achieve an unbiased evaluation at reasonable cost by evaluating the integral stochastically with an unbiased estimator. We observe that

$$f_p^{\text{diff}}(\omega_i \rightarrow \omega_o) = \frac{1}{\pi} \int_{\Omega} \langle \omega_o, \omega_m \rangle D_{\omega_i}(\omega_m) d\omega_m$$

$$= \lim_{N \rightarrow +\infty} \frac{1}{N} \sum_{n=1}^N \frac{1}{\pi} \langle \omega_o, \omega_m(n) \rangle \quad (26)$$

where $\omega_m(n)$ is the n -th sample from the distribution of visible normals D_{ω_i} . That is, an unbiased estimator of the diffuse phase function is obtained by sampling a normal ω_m from D_{ω_i} with the procedure explained in Section 5.3 and evaluating the diffuse contribution of the light source to this normal $\frac{1}{\pi} \langle \omega_o, \omega_m \rangle$.

6 Results

We generated our results on an Intel Core i7-3770 CPU with 20 GB memory and an NVIDIA Quadro M6000 GPU. Our main implementation is integrated into Mitsuba [Jakob 2010], and we introduce our main results in the following paragraphs. Additionally, we implemented a GPU renderer in CUDA that exploits an octree structure [Crassin et al. 2009], which we used to generate Fig. 1. This GPU implementation is able to trace 1.15M paths per second and shows that our method is simple enough to be implemented on such platforms.

Specular vs. Diffuse Microflakes Fig. 15 compares the appearance of a triangle mesh and surface scattering models (diffuse and GGX microfacet BRDFs) to the appearance of density volumes with our SGGX phase functions. The comparison shows that the specular and diffuse SGGX phase functions allow for modelling volumetric appearance beyond simple isotropic scattering.

Performance Fig. 14 compares the performance of the SGGX operators to those of the Gaussian microflake distribution by Zhao et al. [2011]. We used the implementation provided in the current release of Mitsuba, where we also integrated our method for faithful comparison. We generated 1 million random configurations with different Gaussian microflake distributions and different incident and outgoing directions. The SGGX parameters were *converted on-the-fly* as described in Section 4.3 and the cost of the conversion is part of the measured timings.

The results show that the evaluation of the projected area $\sigma(\omega_i)$ and the specular phase function $f_p^{\text{spec}}(\omega_i \rightarrow \omega_o)$ are about twice as fast with SGGX while being easier to implement. This is due to our analytic SGGX operators, while the implementation of Zhao et al. [2011] uses precomputed data that must be fetched and interpolated. For both methods, importance sampling is the most expensive operation. For the Gaussian distribution, it requires a non-deterministic rejection sampling scheme which uses costly numerical CDF inversions in each iteration. Our importance sampling is more than one order of magnitude faster. Furthermore, it follows a predictable instruction sequence, which makes it efficient on GPU architectures. Finally, note that our importance sampling takes a constant number of random numbers as input and provides a continuous parameterization of the sampling space, which also makes it better suited for more elaborate global illumination methods such as Metropolis light transport.

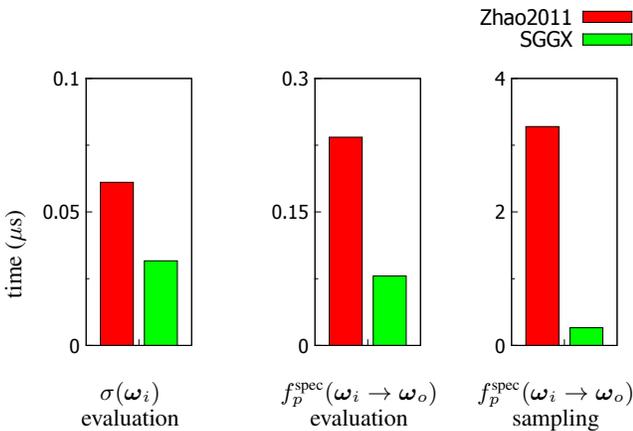


Figure 14: Performance. We compare the performance of the microflake operators of Zhao et al. [2011] against SGGX.

Compatibility with Existing Data Fig. 16 shows SGGX conversion examples performed on-the-fly (Section 4.3) using data from Zhao et al.’s [2011] method as input. Although the images do not converge exactly to the same result (because the representations are not identical), the appearance remains consistent. The rendering performance for the scarf model is approximately 25% faster due to our more efficient SGGX operators. We found that the bottleneck is the I/O overhead for accessing the volume data in Mitsuba. The third example shows a spatially-constant volume. In this case the rendering is not slowed down by I/O and the speedup is about 2× to 3×. Multi-scale rendering put aside, the SGGX operators thus improve rendering performance.

Filtering SGGX Distributions Fig. 17 shows the high resolution volumetric models of a hairball and a tree that we used to evaluate the prefiltering of our SGGX representation. We generated these datasets procedurally and through surface voxelization, respectively. Fig. 18 shows the models viewed at a distance. We computed the reference with the full resolution model (1024^3 voxels) and the downsampled SGGX data with 3D MIP mapping. As shown by Kraus and Bürger [2008], direct density MIP mapping yields poor results. Hence, we rely on their work to filter both density and albedo. We compare several levels of detail; the number of the LOD denotes how many times the model has been downsampled (LOD0: full resolution, LOD2: downsampled by a factor of 4 in all dimensions).

In both views we chose the LOD for SGGX to match a one-voxel-per-pixel ratio, and varied the LOD for density. SGGX downscaling with full resolution density works well and results in stable appearance. However, we observed that downsampling the density generally alters the result. This is because the density downsampling procedure does not account for correlation of visibility along a light path. Another limitation of our approach is that we neglect the correlation between visibility of the voxels and material properties.

Still, we observe that if SGGX data matches a one-voxel-per-pixel ratio and the density is downsampled less aggressively, then we obtain results close to the reference. For instance, the combination “SGGX LOD3 with density LOD2” in the second row is close to the reference. Thus, for distant views, only 15MB of memory are enough to produce similar appearance.

7 Conclusion

We introduced the SGGX microflake distribution, which provides closed-form analytical expressions for the operators used in the microflake framework. In addition to being simple to implement and use, it significantly accelerates the rendering of microflake volumes. Furthermore, it can represent specular and diffuse microflakes in a unified manner.

Our representation can be linearly filtered. We have seen that our prefiltered distributions are able to preserve the roughened appearances that emerge from shiny materials. We have also seen that, at a distance, a small amount of memory contains enough information to restore the appearance of complex volumetric objects.

The most important limitation of our approach is the way we pre-filter density. In many configurations, the visibility of the material can be correlated with itself or with spatially-varying material properties. Density downsampling thus remains an important problem for future work.

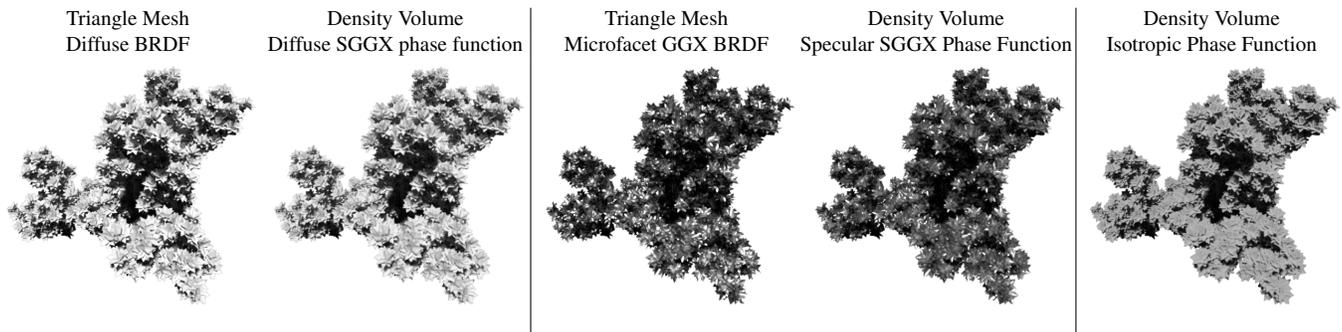


Figure 15: Specular and diffuse SGGX phase functions.

Acknowledgements

The authors wish to thank Fabrice Neyret for his valuable advice on ellipsoidal distributions and for sharing his vision of multi-scale representations. Eric Heitz and Cyril Crassin thank David Luebke and Aaron Lefohn at NVIDIA Research for their support of this research. Jonathan Dupuy acknowledges financial support from GRAND and the ANR AMCQMCSCGA.

References

- BALA, K. 2014. Modeling cloth at micron resolution. In *Proc. SPIE*, vol. 9018, 90180J–90180J–6.
- CRASSIN, C., NEYRET, F., LEFEBVRE, S., AND EISEMANN, E. 2009. Gigavoxels : Ray-guided streaming for efficient and detailed voxel rendering. In *Proc. ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, 15–22.
- DUPUY, J., HEITZ, E., IEHL, J.-C., POULIN, P., NEYRET, F., AND OSTROMOUKHOV, V. 2013. Linear efficient antialiased displacement and reflectance mapping. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)* 32, 6, 211:1–211:11.
- HEITZ, E., AND D’EON, E. 2014. Importance sampling microfacet-based BSDFs using the distribution of visible normals. In *Proc. Eurographics Symposium on Rendering*, 103–112.
- HEITZ, E. 2014. Understanding the masking-shadowing function in microfacet-based BRDFs. *Journal of Computer Graphics Techniques* 3, 2, 32–91.
- HILL, S., MCAULEY, S., DUPUY, J., GOTANDA, Y., HEITZ, E., HOFFMAN, N., LAGARDE, S., LANGLANDS, A., MEGIBBEN, I., RAYANI, F., AND DE ROUSIERS, C. 2014. Physically based shading in theory and practice. In *ACM SIGGRAPH Courses*.
- JAKOB, W., ARBREE, A., MOON, J. T., BALA, K., AND MARSCHNER, S. 2010. A radiative transfer framework for rendering materials with anisotropic structure. *ACM Transactions on Graphics (Proc. SIGGRAPH)* 29, 4, 53:1–53:13.
- JAKOB, W., 2010. Mitsuba renderer. <http://www.mitsuba-renderer.org>.
- KRAUS, M., AND BÜRGER, K. 2008. Interpolating and downsampling rgba volume data. In *Proceedings of the Vision, Modeling, and Visualization Conference*, 323–332.
- MARSCHNER, S. R., JENSEN, H. W., CAMMARANO, M., WORLEY, S., AND HANRAHAN, P. 2003. Light scattering from human hair fibers. *ACM Transactions on Graphics (Proc. SIGGRAPH)* 22, 3, 780–791.
- NEYRET, F. 1995. A general and multiscale model for volumetric textures. In *Graphics Interface*, 83–91.
- NEYRET, F. 1998. Modeling animating and rendering complex scenes using volumetric textures. *IEEE Transactions on Visualization and Computer Graphics* 4, 1, 55–70.
- OLANO, M., AND BAKER, D. 2010. Lean mapping. In *Proc. ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, 181–188.
- PHARR, M., AND HUMPHREYS, G. 2010. *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann Publishers.
- SCHRÖDER, K., KLEIN, R., AND ZINKE, A. 2011. A volumetric approach to predictive rendering of fabrics. *Computer Graphics Forum (Proc. Eurographics Symposium on Rendering)* 30, 4, 1277–1286.
- SCHRÖDER, K., ZHAO, S., AND ZINKE, A. 2012. Recent advances in physically-based appearance modeling of cloth. *ACM SIGGRAPH Asia 2012: Course Notes*.
- TROWBRIDGE, T. S., AND REITZ, K. P. 1975. Average irregularity representation of a rough surface for ray reflection. *Journal of the Optical Society of America* 65, 5, 531–536.
- WALTER, B., MARSCHNER, S. R., LI, H., AND TORRANCE, K. E. 2007. Microfacet models for refraction through rough surfaces. In *Proc. Eurographics Symposium on Rendering*, 195–206.
- XU, K., SUN, W.-L., DONG, Z., ZHAO, D.-Y., WU, R.-D., AND HU, S.-M. 2013. Anisotropic spherical gaussians. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)* 32, 6, 209:1–209:11.
- ZHAO, S., JAKOB, W., MARSCHNER, S., AND BALA, K. 2011. Building volumetric appearance models of fabric using micro CT imaging. *ACM Transactions on Graphics (Proc. SIGGRAPH)* 30, 4, 44:1–44:10.
- ZHAO, S., JAKOB, W., MARSCHNER, S., AND BALA, K. 2012. Structure-aware synthesis for predictive woven fabric appearance. *ACM Transactions on Graphics (Proc. SIGGRAPH)* 31, 4, 75:1–75:10.
- ZHAO, S., HAŠAN, M., RAMAMOORTHY, R., AND BALA, K. 2013. Modular flux transfer: Efficient rendering of high-resolution volumes with repeated structures. *ACM Transactions on Graphics (Proc. SIGGRAPH)* 32, 4, 131:1–131:12.

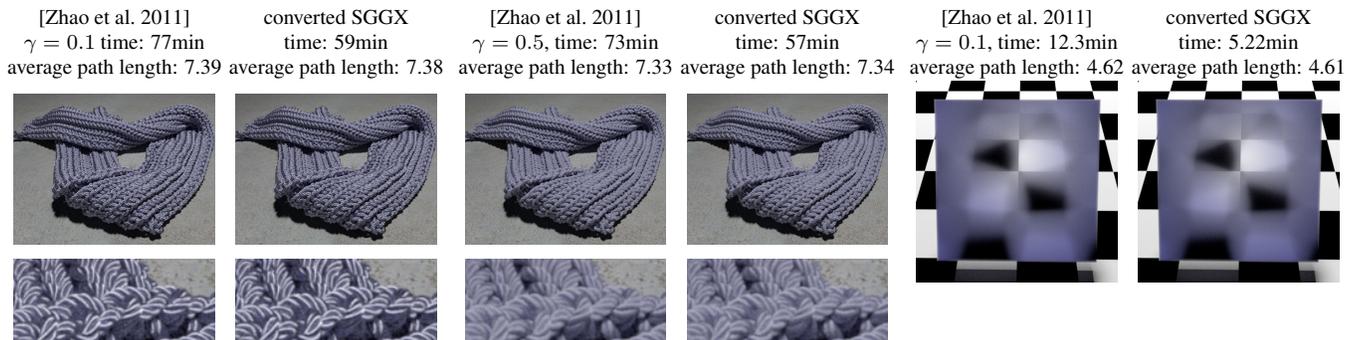


Figure 16: Converting existing data to SGGX. *Left: The scarf model rendered with Mitsuba’s implementation of Zhao et al.’s phase function and the same data converted to our SGGX representation on-the-fly, which we use for importance sampling and closed form evaluations. This yields almost identical result images with better performance. Right: A homogeneous microflake medium with high anisotropy rendered with Zhao et al.’s phase function and SGGX. All six images were computed at 1024×1024 resolution with Mitsuba’s volumetric forward pathtracer using 256 samples per pixel.*

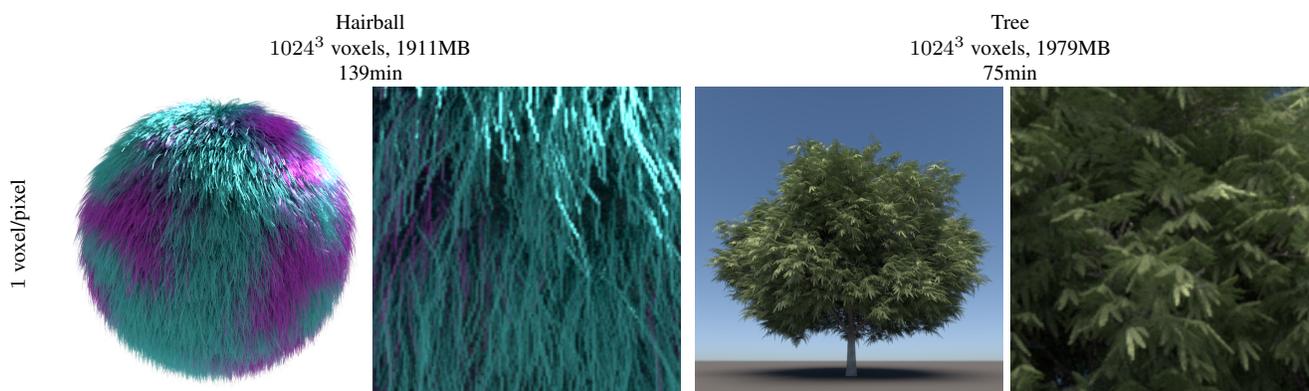


Figure 17: High-resolution volumetric models. *The images were computed at 1024×1024 resolution with Mitsuba’s volumetric forward pathtracer using 256 samples per pixel.*

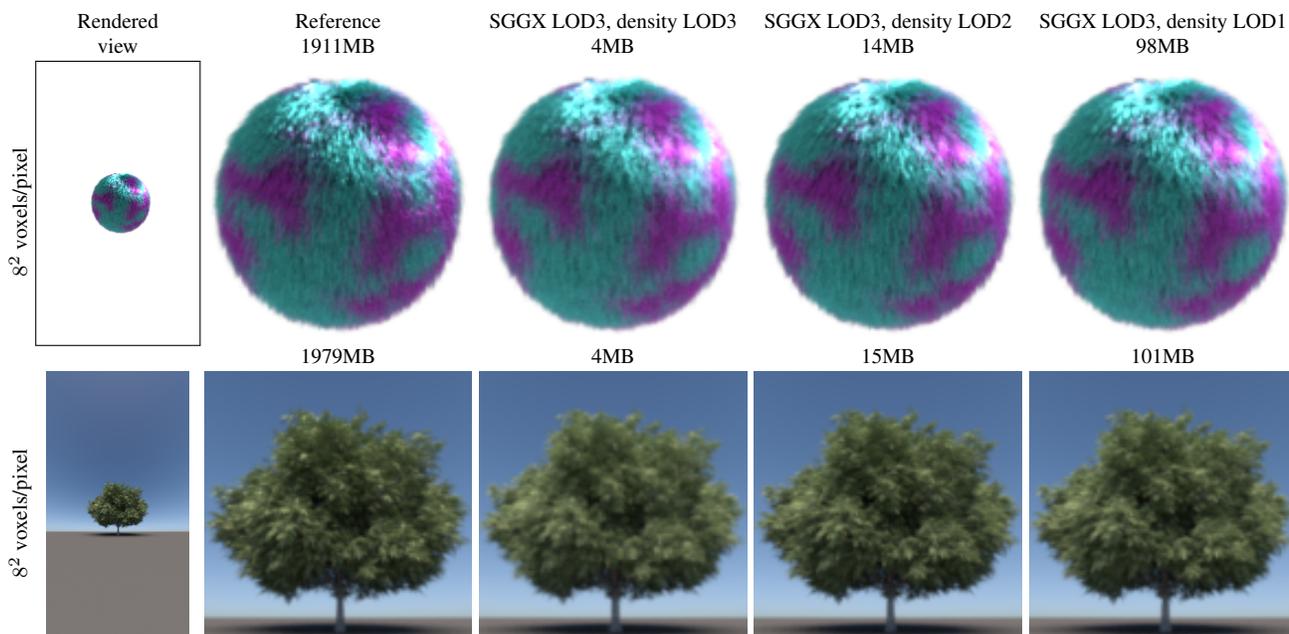


Figure 18: Prefiltering the volumetric models. *The left column shows the rendered view at 512×512 resolution, all other images show $8 \times$ magnifications, rendered with the reference (full resolution) and different levels of detail settings.*