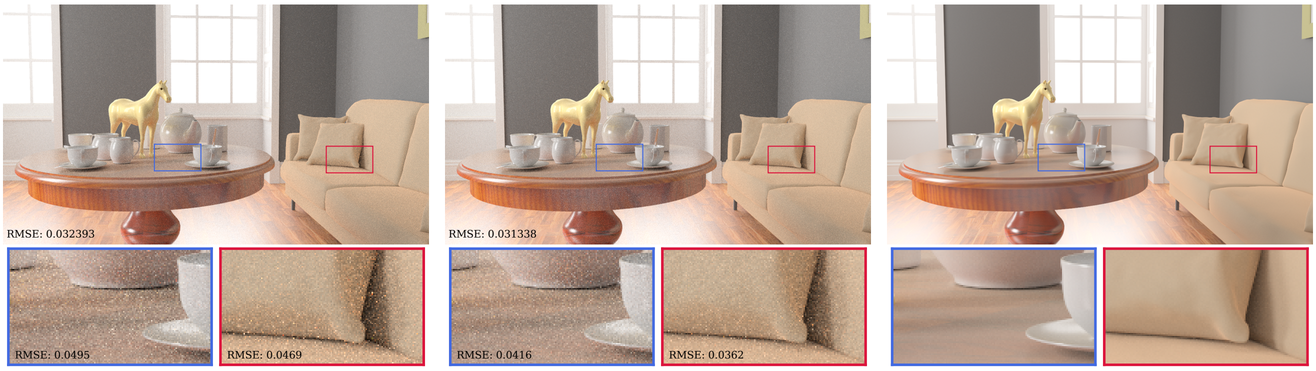
EUROGRAPHICS 2025 / B. Wang and A. Wilkie
(Guest Editors)

Volume 44 (2025), Number 4

Importance Sampling of the Micrograin Visible NDF

S. Lucas[1,2,3] R. Pacanowski[2] and P. Barla[2]

[1] Université de Bordeaux, France; [2]Inria, France; [3] Université Côte d'Azur, France



(a) NDF sampling: 387 spp, 512.7 sec. (b) vNDF sampling: 346 spp, 513.3 sec. (c) Reference: $21k$ spp, 546 min.

Figure 1: *Multiple renderings of a living-room scene showing several objects covered by different types of micrograins, using (a) NDF sampling or (b) vNDF sampling, with (c) a reference rendering. The zoom insets highlight the impact of vNDF sampling with a noticeable noise reduction mainly at grazing angles and in regions of inter-reflections, for near constant-time renderings.*

Abstract

Importance sampling of visible normal distribution functions (vNDF) is a required ingredient for the efficient rendering of microfacet-based materials. In this paper, we explain how to sample the vNDF for the micrograin material model [LRPB23], which has been recently improved to handle height-normal correlations through a new Geometric Attenuation Factor (GAF) [LRPB24], leading to a stronger impact on appearance compared to the earlier Smith approximation. To this end, we make two contributions: we derive analytic expressions for the marginal and conditional cumulative distribution functions (CDFs) of the vNDF; we provide efficient methods for inverting these CDFs based respectively on a 2D lookup table and on the triangle-cut method [Hei20].

CCS Concepts

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

1. Motivation and related work

Most physically-based rendering engines solve the rendering equation through Monte Carlo integration. This process produces noisy images that can only be removed by computing more samples, or by using denoisers (e.g., [SFA*24]) or more efficient importance sampling algorithms. Even though denoisers are nowadays efficient, the final image depends on the quality of the input image, which is why the development of low variance sampling algorithms is still required.

The sampling of Bidirectional Scattering Distribution Functions (BSDF) has a significant impact on noise, which is mitigated through importance sampling. For microfacet models [TR75, WMLT07], state-of-the-art BSDF sampling algorithms rely on the sampling of the visible normal distribution function (vNDF) [Hd14]. This method aims at avoiding the sampling of microfacet normals that create invalid light paths. Naive methods only sample the normal distribution function (NDF) whereby half of the sampled normals at grazing angles are rejected, making this sampling scheme inefficient.

Our goal in this paper is to introduce efficient importance sampling for the micrograin formalism [LRPB23, LRPB24], used for the rendering of porous layers. The current solution is limited to

NDF importance sampling for both diffuse and specular micrograins. We extend their approach to vNDF importance sampling.

Multiple algorithms have been proposed to sample the vNDF. Heitz et al. [Hd14] introduced an analytical inversion of the cumulative vNDF based on fitted functions for the Trowbridge-Reitz (GGX) distribution and on a piecewise inversion for the Beckmann microfacet distribution [BS63]. Later, Jakob [Jak14] used an iterative process based on Newton's method for the Beckmann microfacet distribution. Heitz [Hei18] proposed a simpler approach relying on a geometric interpretation of the Trowbridge-Reitz (GGX) NDF to generate microfacet normals (micronormals) in a unit transformed space. This method has been recently improved by Dupuy et al. [DB23] and Tokuyoshi et al. [TE24], who proposed algorithms based on the sampling of a spherical cap. The vNDF of the Student-T distribution has also been derived through an analytical formulation [d'E23]. Other methods rely on tabulated data, such as in the work of Lawrence et al. [LRR04] for mono-lobe distributions and Chermain et al. [CSDD21] for multi-lobe distributions.

These methods are either based on a geometric construction or on an inversion of the CDF which may be analytic, numeric or tabulated. An alternative consists in using the Triangle Cut method [Hei20], which allows for the analytic sampling of a PDF without having to know the analytic expression of the corresponding inverse CDF. The core idea of this method is to transform biased random samples generated from a "sufficiently good approximation" $\tilde{P}$ of the PDF P, into valid samples. All that is required is the PDF P, its derivative P', its CDF C, and a way to sample the approximated PDF $\tilde{P}$. Since the vNDF is bivariate, this method should be applied twice, once for the marginal CDF, and once for the conditional CDF.

Unfortunately, the application of the Triangle-Cut method to the micrograin model [LRPB24] is not straightforward. First and foremost, no analytical formula for the marginal and conditional CDFs for the vNDF have been derived yet. Second, the handling of masking and shadowing is more involved due to dependencies between heights and normals.

In this paper, we show how the vNDF of micrograin distributions may be importance-sampled (see Figure 1 for a comparison to NDF sampling). We first provide a vNDF formula for porous microsurfaces, which differs from non-porous cases such as GGX distributions (Section 3.1). We then derive the exact marginal and conditional CDFs for this vNDF using the shape-invariance property of micrograin structures (Section 3.2). Our sampling routine relies on a 2D look-up table for the importance sampling of the zenithal angle θ and on the Triangle-Cut method for the azimuthal angle ϕ (Section 4). We next show that our method is more efficient than previous methods, both for diffuse and specular micrograins (Section 5).

2. Background

Lucas et al [LRPB23] introduced the micrograin-based BSDF model for the rendering of porous layers such as dust, pollen or spray paint. Micrograins define a stochastic geometry similarly to microfacets or microflakes: they are opaque half-ellipsoids randomly distributed on a surface plane. The appearance of the base surface is arbitrary whereas micrograins may be either specular or diffuse. Thanks to the shape-invariance property of micrograins, the model has been further improved to handle anisotropy by Lucas et al. [LRPB24], where they introduced an exact masking and shadowing term – also called Geometric Attenuation Factor.

The micrograin BSDF model is defined as a light and view dependent combination of a micrograin layer BRDF f_S and an arbitrary bulk surface BSDF f_B:

$$f(\mathbf{i},\mathbf{o}) = \tau_0 f_S(\mathbf{i},\mathbf{o}) + V_p(\mathbf{i},\mathbf{o}) f_B(\mathbf{i},\mathbf{o}). \tag{1}$$

The proportion of the surface plane covered by micrograins is controlled by the filling factor τ_0. The bulk surface is only visible through pores between micrograins, which is characterized by the pore visibility term $V_p(\mathbf{i},\mathbf{o})$.

The importance sampling for this BSDF is defined in the work of Lucas et al. [LRPB24]. Since the base surface is arbitrary, only the sampling of f_S is considered. The micrograin layer BRDF consists of a sum of diffuse and specular terms. Both involve the Normal Distribution Function (NDF) $D(\mathbf{m})$ and the Geometric Attenuation Factor (GAF) $G(\mathbf{i},\mathbf{o},h)$ with $\mathbf{m}$ the micronormal, h the height of the micronormal, $\mathbf{i}$ and $\mathbf{o}$ respectively the ingoing and outgoing directions. One important aspect of the micrograin model compared to microfacet models is that the micrograin normal corresponds to the normal of a point on an half-ellipsoid; hence the heights and normals are coupled through $h = \cos\theta_m$. The NDF is defined as :

$$D(\mathbf{m}) \;=\; \frac{1}{\cos^4\theta_{\mathbf{m}}}\frac{1}{\beta_x^2\beta_y^2}\frac{1}{\tau_0}\frac{\rho e^{-\rho\frac{\pi}{\beta_x\beta_y}\frac{t(\mathbf{m})^2}{t(\mathbf{m})^2+1}}}{(t(\mathbf{m})^2+1)^2}, \tag{2}$$

where $t(\mathbf{m}) = \tan\theta_{\mathbf{m}}\sqrt{\frac{\cos^2\phi_{\mathbf{m}}}{\beta_x^2}+\frac{\sin^2\phi_{\mathbf{m}}}{\beta_y^2}}$, while $\frac{1}{\beta_x}$ and $\frac{1}{\beta_y}$ are the semi-axis lengths of the micrograins (they provide controls similar to roughness parameters). The filling factor is then given by $\tau_0 = 1 - e^{-\rho\frac{\pi}{\beta_x\beta_y}}$. It is used as a control parameter since it is conveniently bounded between 0 and 1. However, the density ρ of micrograins is easily retrieved by using $\rho = \frac{-\ln(1-\tau_0)\beta_x\beta_y}{\pi}$. Note that the division by τ_0 in Equation 2 is necessary to normalize the NDF, since micrograins do not cover the entire surface plane. This is further explained in Section 3 and illustrated in Figure 4a.

The height-correlated GAF is defined by $G(\mathbf{i},\mathbf{o},h) = \chi^+(\mathbf{i}\cdot\mathbf{n})\chi^+(\mathbf{o}\cdot\mathbf{n})G_{dist}(\mathbf{i},\mathbf{o},h)$, with χ^+ the Heaviside function and G_{dist} the bidirectional distant GAF. For importance sampling, we only need the unidirectional distant GAF:

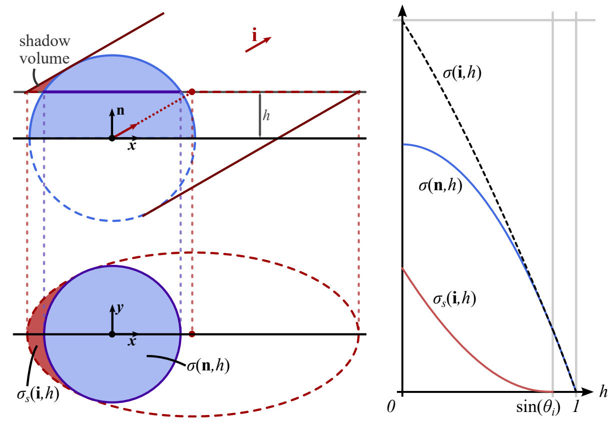
$$G_{dist}(\mathbf{i},h) = \frac{e^{-\rho\sigma(\mathbf{i},h)}}{e^{-\rho\sigma(\mathbf{n},h)}}, \tag{3}$$

where $\sigma(\mathbf{i},h)$ is the projected area from direction $\mathbf{i}$ of an ellipsoid cap, obtained by cutting a micrograin by a plane at height h. We refer the interested reader to the work of Lucas et al. [LRPB24] for more details on the fully-correlated bidirectional GAF.

A key contribution of Lucas et al. [LRPB24] is to show that for random micrograin distributions, the filling factor is generalized to:

$$\tau(\mathbf{i},h) = 1 - e^{-\rho\sigma(\mathbf{i},h)}. \tag{4}$$

Using this equation, the filling factor parameter τ_0 can be written as $\tau(\mathbf{n},0)$ with $\mathbf{n}$ the geometric normal.



(a) (b)

Figure 2: *(a) Geometric interpretation of the micrograin projected area σ and projected shadow σ_s as provided by Lucas et al. [LRPB24]. (b) Plot of $\sigma(\mathbf{i},h)$ computed as the sum of $\sigma(\mathbf{n},h)$ and $\sigma_s(\mathbf{i},h)$ for $\theta_\mathbf{i} = 60°$. The projected shadow area σ_s vanishes for $h \geq \sin\theta_\mathbf{i}$.*

$\mathbf{n}$	geometric normal
$\mathbf{m}$	micronormal
τ_0	filling factor
ρ	micrograin density
β_x,β_y	anfractuosity
$\mathbf{x}'$	vector $\mathbf{x}$ in the unit transformed space
$D^1(\mathbf{m}')$	unit NDF
$D^1_\mathbf{v}(\mathbf{m}')$	unit vNDF
$P^1_\mathbf{v}(\mathbf{i}',\theta)$	unit marginal PDF
$C^1_\mathbf{v}(\mathbf{i}',\theta)$	unit marginal CDF
$P^1_\mathbf{v}(\mathbf{i}',\phi \mid \theta)$	unit conditional PDF
$C^1_\mathbf{v}(\mathbf{i}',\phi \mid \theta)$	unit conditional CDF
$\sigma^1(\mathbf{i}',h')$	unit micrograin projected area
$\tau^1(\mathbf{i}',h')$	unit visible filling factor

Table 1: *Notations for micrograin vNDF sampling.*

The projected area of a micrograin $\sigma(\mathbf{i},h)$ is defined as:

$$\sigma(\mathbf{i},h) = \begin{cases} \sigma(\mathbf{n},h) & \text{if } h \geq \sin\theta_\mathbf{i}, \\ \sigma(\mathbf{n},h) + \sigma_s(\mathbf{i},h) & \text{otherwise,} \end{cases} \tag{5}$$

with $\sigma(\mathbf{n},h)$ the area of the intersection between the microgain and the plane at height h, and $\sigma_s(\mathbf{n},h)$ the area of the micrograin shadow, which exists only when $h < \sin\theta_\mathbf{i}$. For further information on the definition and computation details of σ, we refer the interested readers to the work of Lucas et al. [LRPB24]. Figure 2 illustrates the geometric decomposition behind Equation 5, as well as a plot for $\theta_\mathbf{i} = 60°$.

The main notations regarding the importance sampling of micrograin normals are summarized in Table 1.

3. Cumulative distribution functions

The micrograin distribution is shape-invariant. For simplicity, the following derivations thus focus on a distribution of unit spherical micrograins. More general cases (including anisotropic distributions) are easily handled through space transformations. As shown in Figure 3, anisotropic micrograin distributions can be transformed into a unit space micrograin distribution.

3.1. Visible normal distribution function

Classically (e.g., [Hei14]), the visible density of a normal $\mathbf{m}$ is computed from its density $D(\mathbf{m})$ multiplied by the probability of being visible from direction $\mathbf{i}$, given by $G_{dist}(\mathbf{i},h)$. Then, the visible density is transformed by a factor $\frac{\langle \mathbf{i}\cdot\mathbf{m}\rangle}{\langle \mathbf{i}\cdot\mathbf{n}\rangle}$ such that the distribution is normalised based on the projected area of a microfacet in direction $\mathbf{i}$. In the micrograin model, the surface plane is not entirely filled. Similarly to Equation 8, where the filling factor at normal incidence τ_0 is used to normalize the distribution, pore visibility have to be taken into account in the vNDF, which is achieved by:

$$D_\mathbf{v}(\mathbf{i},\mathbf{m}) = \frac{\tau_0}{\tau(\mathbf{i},0)} \frac{D(\mathbf{m})G_{dist}(\mathbf{i},\mathbf{m})\,\langle \mathbf{i}\cdot\mathbf{m}\rangle}{\langle \mathbf{i}\cdot\mathbf{n}\rangle}. \tag{6}$$

Intuitively, the multiplication by τ_0 cancels the normalization in D, while the division by $\tau(\mathbf{i},0)$ reintroduces the proper normalization due to the filling factor from direction $\mathbf{i}$.

In order to express the vNDF in the unit transformed space, we need to transform each of its terms in turn. First, as shown by Heitz et al. [HDHN16] and further extended by Atanasov et al. [AKDW22], the distribution of normals $D(\mathbf{m})$ is defined from a distribution $D^1(\mathbf{m}')$ in a unit transformed space by:

$$D(\mathbf{m}) = \frac{\det(M)}{||M^T\mathbf{m}||^4} D^1(\mathbf{m}'), \tag{7}$$

where $\mathbf{m}'$ is the transformed micronormal computed as $\mathbf{m}' = \frac{M^T\mathbf{m}}{||M^T\mathbf{m}||}$ with $M^{-1} = \mathrm{diag}(\beta_x,\beta_y,1)$ the transformation that turns half-spheres into micrograins of semi-axis β_x and β_y. To derive D^1, we simply consider $\beta_x = 1$ and $\beta_y = 1$ in Equation 2, yielding:

$$D^1(\mathbf{m}') = \frac{\rho e^{-\rho\pi\sin^2\theta_{\mathbf{m}'}}}{\tau_0} = \frac{\rho e^{-\rho\sigma^1(\mathbf{n},h')}}{1-e^{-\rho\sigma^1(\mathbf{n},0)}}, \tag{8}$$

where $h' = \mathbf{m}'\cdot\mathbf{n}$ is the height of the point on a micrograin with normal $\mathbf{m}'$ and $\sigma^1(\mathbf{v},h')$ is the projected area of a *unit spherical* cap of height h' from direction $\mathbf{v}$.

As shown by Heitz [Hei14], the shadowing term can be computed in the unit transformed space as long as the directions are also transformed. Thus, we express $G_{dist}(\mathbf{i},\mathbf{m})$ in the unit transformed space as:

$$G_{dist}(\mathbf{i},\mathbf{m}) = G^1_{dist}(\mathbf{i}',\mathbf{m}') = \frac{e^{-\rho\sigma^1(\mathbf{i}',h')}}{e^{-\rho\sigma^1(\mathbf{n},h')}}, \tag{9}$$

where $\mathbf{i}' = \frac{M^{-1}\mathbf{i}}{||M^{-1}\mathbf{i}||}$ is the transformation of $\mathbf{i}$ in the unit transformed space. Similarly to G_{dist}, we rewrite the visible filling factor as:

$$\tau(\mathbf{i},h) = \tau^1(\mathbf{i}',h') = 1-e^{-\rho\sigma^1(\mathbf{i}',h')}. \tag{10}$$

Based on the relation between $\mathbf{i}$ and $\mathbf{i}'$, we have $\langle \mathbf{i}\cdot\mathbf{n}\rangle =$

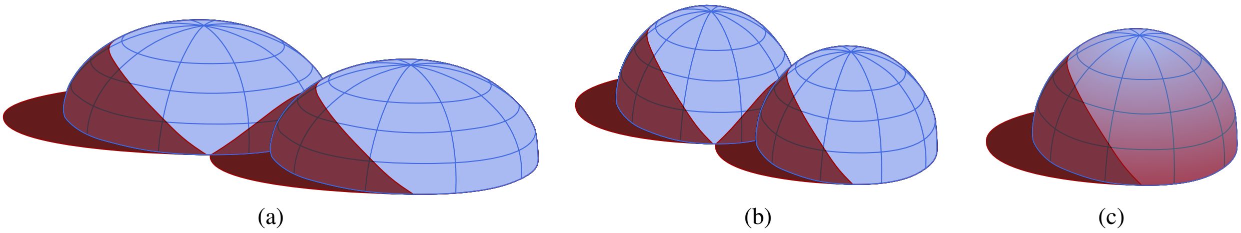


Figure 3: *Transformation of an anisotropic micrograin distribution (a) into of a unit space micrograin distribution (b). The micrograin visibility is illustrated in (c) as a color gradient between visible (in blue) and not visible (in red) locations.*

$\langle \mathbf{i}' \cdot \mathbf{n} \rangle \, ||M^{-1}\mathbf{i}||$. We show in supplemental material that $\langle \mathbf{i} \cdot \mathbf{m} \rangle = \langle \mathbf{i}' \cdot \mathbf{m}' \rangle \, ||M^{-1}\mathbf{i}|| \, ||M^T \mathbf{m}||$. We may now rewrite Equation 6 as:

$$D_{\mathbf{v}}(\mathbf{i},\mathbf{m}) = \frac{\det(M)}{||M^T\mathbf{m}||^3} \frac{D^1(\mathbf{m}')G^1_{dist}(\mathbf{i}',\mathbf{m}')\langle \mathbf{i}' \cdot \mathbf{m}' \rangle}{\langle \mathbf{i}' \cdot \mathbf{n} \rangle} \frac{\tau_0}{\tau^1(\mathbf{i}',0)} \tag{11}$$

$$= \frac{\det(M)}{||M^T\mathbf{m}||^3} D^1_{\mathbf{v}}(\mathbf{i}',\mathbf{m}'), \tag{12}$$

where $D^1_{\mathbf{v}}$ – the visible normal distribution function in the unit transformed space – is obtained by using Equations 8, 9 and 10:

$$\boxed{D^1_{\mathbf{v}}(\mathbf{i}',\mathbf{m}') = \frac{\rho e^{-\rho\sigma^1(\mathbf{i}',h')}}{1-e^{-\rho\sigma^1(\mathbf{i}',0)}} \frac{\langle \mathbf{i}' \cdot \mathbf{m}' \rangle}{\langle \mathbf{i}' \cdot \mathbf{n} \rangle}. \tag{13}}$$

Note that in Equation 13, the variation in $\phi_{\mathbf{m}'}$ only comes from $\langle \mathbf{i}' \cdot \mathbf{m}' \rangle$. As we will see in Section 4, this observation is key to the efficient sampling of the vNDF.

Common vNDF sampling routine are built upon the Smith GAF assumptions, where the visibility is independent of the micronormal thus simplifying the sampling procedure. In our case, the GAF cannot be neglected as illustrated in Figure 3(c).

3.2. Marginal and Conditional Cumulative Distributions

In previous work [LRPB23, LRPB24], importance sampling of the micrograin model was done on the NDF of Equation 2. In the case of unit spherical micrograins (see Equation 8), the corresponding CDF is given by:

$$C^1(\theta) = \frac{1-e^{-\rho\sigma^1(\mathbf{n},\cos\theta)}}{1-e^{-\rho\sigma^1(\mathbf{n},0)}} = \frac{\tau^1(\mathbf{n},\cos\theta)}{\tau_0}, \tag{14}$$

which is the ratio between the visible filling factors at heights $\cos\theta$ and 0 respectively, both at normal incidence. This ratio has a geometric interpretation as seen in Figure 4a. Note that Equation 14 does not depend on any azimuthal micronormal angle ϕ, since micrograins are fully visible at normal incidence.

In contrast, the vNDF of Equation 13 does depend on a micronormal azimuthal angle, since micrograins are not fully visible away from normal incidence. We thus need to compute a marginal cumulative distribution function (mCDF) and its conditional distribution function (cCDF). These two functions, once inverted, are then used to generate (θ, ϕ) samples from the vNDF, first by sampling θ and then by sampling ϕ knowing θ (see Section 4).

Marginal Cumulative Distribution. In order to compute the mCDF, we first define the marginal probability density function:

$$P^1_{\mathbf{v}}(\mathbf{i}',\theta_{\mathbf{m}'}) = \int_{-\pi}^{\pi} D^1_{\mathbf{v}}(\mathbf{i}',\mathbf{m}') \sin\theta_{\mathbf{m}'} \, \mathrm{d}\phi_{\mathbf{m}'} \tag{15}$$

$$= \frac{\rho e^{-\rho\sigma^1(\mathbf{i}',\cos\theta_{\mathbf{m}'})}}{1-e^{-\rho\sigma^1(\mathbf{i}',0)}} \frac{\sin\theta_{\mathbf{m}'}}{\langle \mathbf{i}' \cdot \mathbf{n} \rangle} \int_{-\pi}^{\pi} \langle \mathbf{i}' \cdot \mathbf{m}' \rangle \, \mathrm{d}\phi_{\mathbf{m}'}, \tag{16}$$

where we show in the supplemental document that :

$$\frac{\sin\theta_{\mathbf{m}'}}{\langle \mathbf{i}' \cdot \mathbf{n} \rangle} \int_{-\pi}^{\pi} \langle \mathbf{i}' \cdot \mathbf{m}' \rangle \, \mathrm{d}\phi_{\mathbf{m}'} = \frac{\mathrm{d}\sigma^1(\mathbf{i}',\cos\theta_{\mathbf{m}'})}{\mathrm{d}\theta_{\mathbf{m}'}}. \tag{17}$$

By integrating the marginal density function using the equality $\int_0^X u'(x)e^{-u(x)}dx = 1-e^{-u(X)}$, we retrieve the following mCDF:

$$\boxed{\begin{aligned} C^1_{\mathbf{v}}(\mathbf{i}',\theta) &= \int_0^{\theta} P^1_{\mathbf{v}}(\mathbf{i}',\theta_{\mathbf{m}'}) d\theta_{\mathbf{m}'} && (18) \\ &= \frac{1-e^{-\rho\sigma^1(\mathbf{i}',\cos\theta)}}{1-e^{-\rho\sigma^1(\mathbf{i}',0)}} = \frac{\tau^1(\mathbf{i}',\cos\theta)}{\tau^1(\mathbf{i}',0)}. && (19) \end{aligned}}$$

This generalizes Equation 14 to any direction $\mathbf{i}'$. Figure 4b gives a geometric interpretation of this generalization.

Conditional Cumulative Distribution. Considering θ and assuming without loss of generality that $\phi_{i'} = 0$, we compute the following conditional cumulative distribution function for $\phi \in [-\phi_q, \phi_q]$:

$$C^1_{\mathbf{v}}(\mathbf{i}',\phi \mid \theta) = \int_{-\phi_q}^{\phi} \frac{D^1_{\mathbf{v}}(\mathbf{i}',\hat{\mathbf{m}}') \sin\theta}{P^1_{\mathbf{v}}(\mathbf{i}',\theta)} d\phi_{\mathbf{m}'} \tag{20}$$

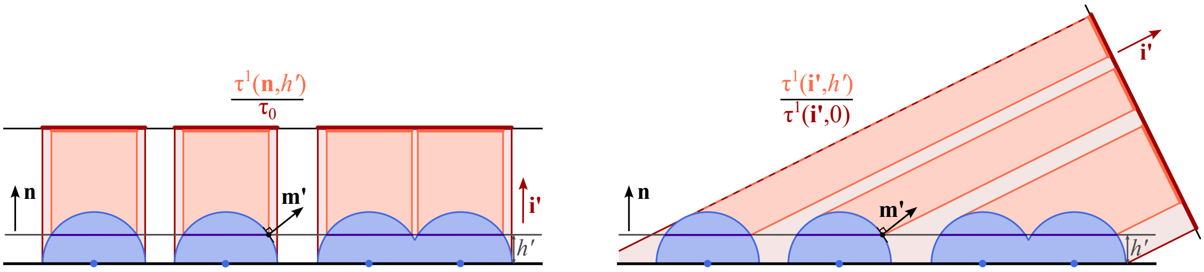
Using Equation 15 and Equation 13, most of the terms can be cancelled out, thus the equation is simplified into:

$$C^1_{\mathbf{v}}(\mathbf{i}',\phi \mid \theta) = \frac{\int_{-\phi_q}^{\phi} \langle \mathbf{i}' \cdot \hat{\mathbf{m}}' \rangle d\phi_{\mathbf{m}'}}{\int_{-\phi_q}^{\phi_q} \langle \mathbf{i}' \cdot \hat{\mathbf{m}}' \rangle d\phi_{\mathbf{m}'}}, \tag{21}$$

where $\pm\phi_q = \pm\arccos\left(-\frac{1}{\tan\theta\tan\theta_{\mathbf{i}'}}\right)$ bound the visible portion of the micrograin [LRPB24] (i.e., $\mathbf{i}' \cdot \mathbf{m}' = 0$ when $\phi_{\mathbf{m}'} = \pm\phi_q$), and $\hat{\mathbf{m}}'$ is the vector of zenithal angle θ and azimuthal angle $\phi_{\mathbf{m}'}$.

As detailed in Supplemental Material through a straightforward derivation, $C^1_{\mathbf{v}}$ is given in analytical form:

$$\boxed{C^1_{\mathbf{v}}(\mathbf{i}',\phi \mid \theta) = \frac{1}{2} + \frac{1}{2} \frac{\sin\phi \sin\theta_{\mathbf{i}'} \sin\theta + \phi\cos\theta_{\mathbf{i}'} \cos\theta}{\sin\phi_q \sin\theta_{\mathbf{i}'} \sin\theta + \phi_q \cos\theta_{\mathbf{i}'} \cos\theta}. \tag{22}}$$


$\frac{\tau^1(\mathbf{n},h')}{\tau_0}$
n
m'
i'
h'
$\frac{\tau^1(\mathbf{i}',h')}{\tau^1(\mathbf{i}',0)}$
i'
n
m'
h'

(a) (b)

Figure 4: *Geometric interpretation of the Cumulative Distribution Function (CDF) given by the ratio of the visible filling factor at height h over the visible filling factor at height* 0 *for both directions* **n** *(a) and* $\mathbf{i}'$ *(b).*

4. Importance sampling

We now look at how to sample θ in Section 4.1; then knowing θ, we explain how to sample ϕ in Section 4.2.

4.1. Sampling θ

By using a uniform random number ξ_0 between 0 and 1 and setting $\xi_0 = C^1_{\mathbf{v}}(\mathbf{i}',\theta)$, we may try to derive the inverse function of Equation 19 to retrieve θ sampled according to the vNDF. Sadly, the inversion is soon halted at:

$$\pi \frac{\ln\left(1-\tau^1(\mathbf{i}',0)\xi_0\right)}{\ln(1-\tau_0)} = \sigma^1(\mathbf{i}',\cos\theta), \tag{23}$$

since σ^1 cannot be inverted for an arbitrary θ. If we now write $\Xi = \pi\frac{\ln(1-\tau^1(\mathbf{i}',0)\xi_0)}{\ln(1-\tau_0)}$ as a *non-uniform* random sample defined in the $[0,\sigma^1(\mathbf{i}',0)]$ range, the sampling of $\cos\theta$ is then given by:

$$\sigma^{-1}\left(\mathbf{i}',\Xi\right) = \cos\theta, \tag{24}$$

where σ^{-1} is the inverse projected area function.

According to Equation 5, we may invert σ^1 in two different ways, depending on the existence of a micrograin shadow:

- When there is no micrograin shadow (i.e., $\cos\theta \geq \sin\theta_{\mathbf{i}'}$) the definition of σ^1 is simplified to $\sigma^1(\mathbf{i}',\cos\theta) = \sigma^1(\mathbf{n},\cos\theta) = \pi(1-\cos^2\theta)$; hence σ^{-1} is given in analytical form:

$$\sigma^{-1}(\mathbf{i}',\Xi) = \sqrt{1-\frac{\Xi}{\pi}}, \tag{25}$$

- In the presence of a micrograin shadow, due to the complexity of its area σ_s, we did not find an analytical solution for its inverse. One way to get around this issue is to store σ^{-1} into a 2D lookup table. Note that this table has to be computed only once since it does not depend on material parameters, only on Ξ and $\theta_{\mathbf{i}}$. The use of a low-dimensional lookup table is possible here because we reduced the problem of sampling θ to the inversion of σ^1, which by definition does not depend on material parameters. We provide the lookup table in supplemental material as well as the code to generate it.

The full sampling for the cosine of θ is described in Listing 1.

```
vec3 sample_cos_theta(vec3 i, float tau_0, float xi){
  float Xi = pi*ln(1-xi*tau(i,0))/ln(1-tau_0);
  if(Xi <= pi*i.z*i.z) { // if no shadow
    return sqrt(1-sigma*inv_pi);
  }
  return inv_sigma_tab.get(i.z, Xi);
}
```

Listing 1: *Sampling routine for* $\cos\theta$.

4.2. Sampling ϕ knowing θ

Ideally ϕ would be retrieved by inverting the conditional cumulative distribution function. However we need an alternative solution since $C^1_{\mathbf{v}}$ cannot be inverted. While θ could be sampled through a 2D lookup table, ϕ requires a 3D lookup table since it also depends on θ. We therefore opt for the triangle cut method to sample ϕ.

The triangle cut method relies on an area-preserving 2D mapping to transform biased samples from an approximated PDF, into valid samples for the base PDF. To create samples with this method, we need: the PDF (Eq. 26), its derivative (Eq. 27), its CDF (Eq. 22), and an approximation of the PDF (Eq. 28) with a way to sample it.

The conditional probability density function is defined as the derivative of Equation 22:

$$P^1_{\mathbf{v}}(\mathbf{i}',\phi\mid\theta) = \frac{1}{2}\frac{\cos\phi\sin\theta_{\mathbf{i}'}\sin\theta+\cos\theta_{\mathbf{i}'}\cos\theta}{\sin\phi_q\sin\theta_{\mathbf{i}'}\sin\theta+\phi_q\cos\theta_{\mathbf{i}'}\cos\theta}, \tag{26}$$

and its derivative with respect to ϕ is given by:

$$\frac{\mathrm{d}P^1_{\mathbf{v}}(\mathbf{i}',\phi\mid\theta)}{\mathrm{d}\phi} = \frac{1}{2}\frac{-\sin\phi\sin\theta_{\mathbf{i}'}\sin\theta}{\sin\phi_q\sin\theta_{\mathbf{i}'}\sin\theta+\phi_q\cos\theta_{\mathbf{i}'}\cos\theta}. \tag{27}$$

As shown in Figure 5, we approximate $P^1_{\mathbf{v}}$ with a tent function:

$$\tilde{P}^1_{\mathbf{v}}(\mathbf{i}',\phi\mid\theta) = \frac{h_1-h_0}{\phi_q}|\phi|+h_1. \tag{28}$$

The tent is centered on $\phi = 0$, bounded by $\pm\phi_q$, with $h_0 = P^1_{\mathbf{v}}(\mathbf{i}',\phi_q\mid\theta)$ and $h_1 = \frac{1}{\phi_q} - h_0$ respectively the minimum and maximum height. The cumulative distribution function of Equation 28 is :

$$\tilde{C}^1_{\mathbf{v}}(\mathbf{i}',\phi \mid \theta) = \int_{-\phi_q}^{\phi} \tilde{P}^1_{\mathbf{v}}(\mathbf{i}',\phi_0 \mid \theta)\mathrm{d}\phi_0 \tag{29}$$

$$= \frac{1}{2} + \phi\left(|\phi|\,\frac{h_0\phi_q - \frac{1}{2}}{\phi_q^2} + h_1\right), \tag{30}$$

which properly integrates to 1 when $\phi = \phi_q$. Thus the inverse CDF is given by:

$$\tilde{C}^{-1}_{\mathbf{v}}(\xi_1) = \begin{cases} \frac{\phi_q}{h_1 - h_0}\left(\sqrt{2h_1^2\xi_1 + h_0^2(1-2\xi_1)} - h_1\right) & \text{if } \xi_1 \leq \frac{1}{2}, \\ \frac{\phi_q}{h_1 - h_0}\left(h_1 - \sqrt{\frac{h_1 - h_0}{\phi_q}(1-2\xi_1) + h_1^2}\right) & \text{otherwise.} \end{cases} \tag{31}$$

```
void sample_tent(float xi ,float phi_q, float h0,
                     float& phi, float& pdf)
{
  float h1 = 1./phi_q - h0;
  float h1_sqr = h1*h1;
  float h0_sqr = h0*h0;
  phi = phi_q/(h1-h0);
  if(xi < 0.5)
    phi *= sqrt(2*h1_sqr*xi +h0_sqr*(1-2*xi))-h1;
  else
    phi *= h1 -sqrt((h1-h0)*(1.-2.*xi)/phi_q + h1_sqr);
  pdf = (h0-h1)/phi_q *abs(phi) +h1;
}
```

Listing 2: *Sampling routine for* $\tilde{P}^1_{\mathbf{v}}(\mathbf{i}',\phi \mid \theta)$.

Two conditions have to be satisfied for the approximated function. First, the domains of definition should be identical between $P^1_{\mathbf{v}}$ and $\tilde{P}^1_{\mathbf{v}}$, which is true by construction. Second, the mapping should be bijective and preserve areas. On the right side of Figure 5, we show the transformed unit square grid with our triangle cut mapping. We did not find any overlap that would have resulted in purple lines crossing each others indicating surjections. On the left side of Figure 5, the blue histogram always closely matches the PDF (Eq. 26). Additional test cases are provided in the Supplemental materials. Given our experiments, we consider that the approximated function satisfies both conditions.

The validation of the sampling algorithm for ϕ is done empirically by testing the entire range of input parameters. More validations are available in the supplemental material. The sampling algorithm for ϕ is provided in Listing 3 where we used notations from Listing 1 of the Triangle-Cut method paper [Hei20] .

The full sampling for $\mathbf{m}'$ is described in Listing 4. Note that since we assumed in Equation 22 that $\phi_{i'} = 0$, we need to rotate $\mathbf{m}'$ around the geometric normal $\mathbf{n}$ such that it is defined with respect to $\mathbf{i}'$ (as done in the last line).

```
1   vec3 sample_phi(vec3 i, float cos_th, vec2 xi){
2     float sin_thi = sqrt(1 - wi_u.z * wi_u.z);
3     float tan_thi = sin_thi / wi_u.z;
4     float sin_th  = sqrt(1 - cos_th * cos_th);
5     float tan_th  = sin_th  / cos_th;
6     float phq = pi;
7     if(sin_thi > cos_th)
8       phq = acos(-1. / (tan_thi * tan_th));
9     float cos_phq = cos(phq), sin_phq = sin(phq);
10    float cos_thi_th = cos_thi*cos_th;
11    float sin_thi_th = sin_thi*sin_th;
12
13    float cst = 1./(2.*(sin_phq*sin_thi_th+phq*cos_thi_th));
14    float h0 = cst*(cos_phq*sin_thi_th+cos_thi_th);
15    float x_a, g_x_a;
16    sample_tent(xi.x, phq, h0, x_a, g_x_a);
17
18    float sin_ph = sin(x_a), cos_ph = cos(x_a);
19    float F_x_a = 0.5+cst*(sin_ph*sin_thi_th+x_a*cos_thi_th);
20    float f_x_a = cst*(cos_ph*sin_thi_th+cos_thi_th);
21    float fprime_x_a = -cst*sin_ph*sin_thi_th;
22
23    return tricut_remove_bias( xi, x_a, f_x_a,
24                               F_x_a, fprime_x_a, g_x_a);
25  }
```

Listing 3: *Sampling routine for* $P^1_{\mathbf{v}}(\mathbf{i}',\phi \mid \theta)$.

```
vec3 sample_m_unit(vec3 i, float tau_0, vec3 xi){
  float cos_th = sample_cos_theta(i, tau_0, xi.x);
  float sin_th = sqrt(1-cos_th*cos_th);
  float phi = sample_phi(cos_th, xi.yz);
  vec3 m = vec3(cos(phi)*sin_th, sin(phi)*sin_th, cos_th);
  vec2 r = normalize(i.xy);
  // apply rotation
  return vec3(m.x*r.x+m.y*r.y, m.x*r.y-m.y*r.x, m.z);
}
```

Listing 4: *Sampling routine for the visible micronormal* $\mathbf{m}'$ *in the unit transformed space.*

5. Results

All results are generated with an 8-core, 3.60 GHz AMD Ryzen 7 3700X CPU except for Figure 1 generated with an 8-core, 2.30 GHz Intel Core i7-11800H CPU. Rendered images in Figures 6, 7, 8 and 9 are computed with a custom pathtracer using a BRDF integrator and MIS integrator for Figure 1. We report RMSE for tone-mapped images in all results. The sampling of the bulk BRDF component impacts variance in these results, but since it is kept unchanged between NDF and vNDF sampling, comparisons remain valid.

Overall, as shown in Table 2, vNDF sampling is between 3 and 3.9 times slower than naive NDF sampling using a lookup table of resolution 1024 × 1024. The vNDF sampling cost increases with the angle of incidence both during the sampling of θ_m and $\phi_{\mathbf{m}}$. When sampling $\theta_{\mathbf{m}}$ the tabulated inversion method (line 6 in Listing 1) is invoked more frequently. Similarly for $\phi_{\mathbf{m}}$, the use of the `acos` function (line 8 in Listing 3) is more frequent due to the presence of shadows. Cache misses can also account for some of the increased costs due to random access to the lookup table. This implies that the sampling cost increases with anfractuosity, as micrograins cast larger shadows. Note that since the triangle cut method relies on a 2D mapping, the sampling of ϕ is done using two random variables, leading to additional computation overhead.

Figure 6 shows the estimated variance as well as the efficiency of the vNDF and NDF importance sampling routines (both averaged over 300 realizations). Note that the estimated variance does not converge to 0, but to the variance *of the estimator*. These metrics have been computed for multiple pixels from Figure 7. The estimated variance is systematically below the previous sampling routine. However, its cost also has to be considered with respect to

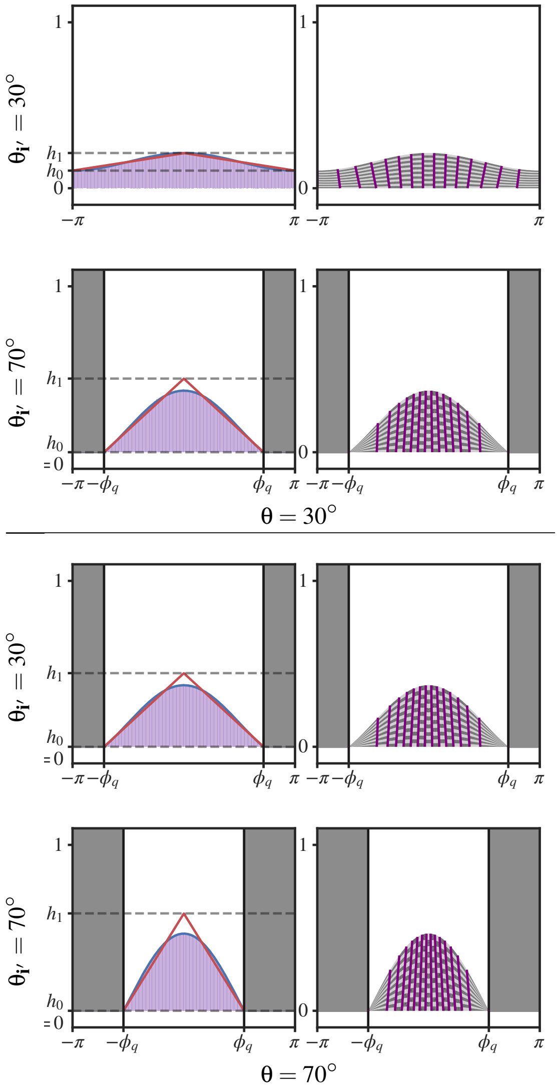


Figure 5: *Triangle-Cut validation for the generation of ϕ samples using a tent function for the approximation of $P^1_{\mathbf{v}}(\mathbf{i}',\phi|\theta)$. The sampling histogram is shown in purple and matches the PDF in blue, the approximation PDF is in red.*

$\theta_{\mathbf{i}'}$	20°	40°	60°	80°
NDF (ns)	76.42	76.39	76.32	76.25
vNDF (ns)	226.81	252.43	276.29	289.33
$\cos\theta_{\mathbf{m}'}$ (ns)	86.25	107.0	126.36	136.87
$\phi_{\mathbf{m}'}$ (ns)	107.91	112.89	117.4	120.18
rotation (ns)	32.66	32.53	32.53	32.28
vNDF/NDF	2.97	3.3	3.62	3.79

Table 2: *CPU mean sampling costs for the NDF and the vNDF sampling ($\mathbf{m}'$) for multiple angles of incidence. Mean sampling costs are given in nanosecond and averaged over 10^7 samples. We split the vNDF sampling cost in three group corresponding to the sampling of $\cos\theta_{\mathbf{m}'}$ given in Listing 1, the sampling of $\phi_{\mathbf{m}'}$ given in Listing 3 and additional computation cost required to rotate $\mathbf{m}'$ around the geometric normal by $\phi_{\mathbf{i}'}$ given in Listing 4. Note that despite a higher sampling cost, vNDF sampling yields a better overall efficiency, as shown in Figure 6.*

the entire rendering cost, which is why we also compare the two methods in terms of efficiency (defined by $\frac{1}{Variance \cdot Time}$ [Vea98]). Efficiency is a relevant metric as it measures simultaneously how well a Monte-Carlo estimator reduces noise (i.e., the variance) and how fast it is to achieve that noise reduction. The blue curves of Figure 6 show that, for a fixed number of samples (x-axis), our method is consistently more efficient (e.g., at 100 spp, 245% more efficient in (a), 126% more efficient in (b), 77% more efficient in (c)), except for near normal incidence (d), where the vNDF importance sampling is slightly less efficient than regular NDF sampling (e.g., at 100 spp, 0.15% less efficient). As detailed in supplemental material, the efficiency of vNDF estimator is 30% more efficient than the NDF estimator when applied to all pixels belonging to the object.

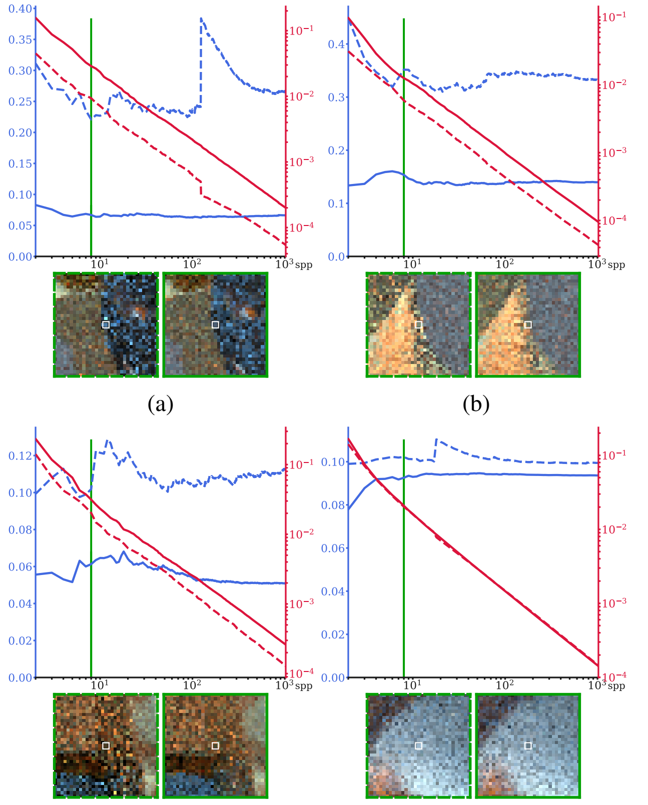
Scenes in Figure 8 and 9 are made of a gray diffuse probe ($K_d = 0.5$) covered by micrograins and lit by a low-frequency environment map. When using green diffuse micrograins in Figure 8, the noise is much less noticeable with our vNDF sampling routine compared to NDF sampling at near constant rendering time, especially at grazing angles. In the same context, the noise is also reduced for gold conductor micrograins as shown in Figure 9. We also provide FLIP error images and Root-Mean Square Error (RMSE) values computed with respect to the converged renders, highlighting that our vNDF sampling routine achieves better convergence with less noise at high roughness levels. The benefits of our method are more noticeable at large filling factors, where a greater number of samples are generated from the micrograin layer rather than the base layer. More results are provided in supplemental material.

One of the main advantage of using the vNDF importance sampling is to have a bounded sample weight $w = \frac{F(\mathbf{i}\cdot\mathbf{m})G(\mathbf{i},\mathbf{o},h)}{G(\mathbf{i},h)} \in [0, F(\mathbf{i}\cdot\mathbf{m})]$, rather than an unbounded weight $w = \frac{F(\mathbf{i}\cdot\mathbf{m})G(\mathbf{i},\mathbf{o},h)|\mathbf{i}\cdot\mathbf{m}|}{|\mathbf{m}\cdot\mathbf{n}||\mathbf{i}\cdot\mathbf{n}|} \in [0,+\infty]$ in the case of the NDF importance sampling [WMLT07]. This is particularly important for more complex scenes with multiple light bounces, such as in Figures 1 or 7, to avoid fireflies due to uncontrolled light path contributions. In this context, our sampling routine provides lower variance and better results for equal-time renderings.

6. Discussion and Future Work

In this work, we provided the theorical foundations to sample the porous micrograin vNDF. We used those to define the first vNDF sampling routine for micrograins based on a 2D lookup table and the Triangle-Cut method. Our sampling routine provides better efficiency and convergence rate than previous methods.

We did not find an analytical way to sample θ, even when using the Triangle-Cut method. However, we derived all the necessary



(a)

(b)

(c)

(d)

Figure 6: *Comparisons of the estimated variance (blue) and efficiency (red, log scale) for different pixels from Figure 7 between vNDF (solid) and NDF (dashed) sampling. Comparisons are done over 1024 samples per pixel (spp). The estimated variance is averaged over 300 realisations. Insets illustrate the selected pixels from one of the realizations at the 8^{th} spp (green vertical line). In most cases, our sampling routine is more efficient at the exception of sampling near normal incidence (d).*

functions in the supplemental document such that the only missing ingredient is a sufficiently good approximation of $P^{1}_{\mathbf{v}}(\mathbf{i}',\theta)$ which we have not found yet.

Performance-wise, our current sampling routine requires three random numbers to sample θ and ϕ. Finding a way to reduce this number would directly result in a gain in performance.

Finally, the main limitation of the micrograin BRDF is the lack of multiple scattering. We consider that our work is a step toward this goal. Indeed this is a required ingredient for a stochastic evaluation of multiple scattering in microfacet models [HHdD16]. What remains to be found is an efficient method for sampling the vNDF at an arbitrary height and direction inside the micrograin structure.

References

[AKDW22] Atanasov A., Koylazov V., Dimov R., Wilkie A.: Microsurface transformations. *Computer Graphics Forum 41*, 4 (2022), 105–116. 3

[BS63] Beckmann P., Spizzichino A.: *The scattering of electromagnetic waves from rough surfaces*. Pergamon Press, 1963. 2

[CSDD21] Chermain X., Sauvage B., Dischler J.-M., Dachsbacher C.: Importance Sampling of Glittering BSDFs based on Finite Mixture Distributions . In *Eurographics Symposium on Rendering - DL-only Track* (2021), Bousseau A., McGuire M., (Eds.), The Eurographics Association. 2

[DB23] Dupuy J., Benyoub A.: Sampling visible ggx normals with spherical caps. *Computer Graphics Forum 42*, 8 (2023), e14867. 2

[d'E23] d'Eon E.: Student-t and beyond: Practical tools for multiple-scattering bsdfs with general ndfs. In *SIGGRAPH 2023 Talks* (New York, NY, USA, 2023), Association for Computing Machinery. 2

[Hd14] Heitz E., d'Eon E.: Importance Sampling Microfacet-Based BSDFs using the Distribution of Visible Normals. *Computer Graphics Forum 33*, 4 (July 2014), 103–112. 1, 2

[HDHN16] Heitz E., Dupuy J., Hill S., Neubelt D.: Real-time polygonal-light shading with linearly transformed cosines. *ACM Trans. Graph. 35*, 4 (July 2016). 3

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

[Hei18] Heitz E.: Sampling the ggx distribution of visible normals. *Journal of Computer Graphics Techniques (JCGT) 7*, 4 (November 2018), 1–13. 2

[Hei20] Heitz E.: Can't Invert the CDF? The Triangle-Cut Parameterization of the Region under the Curve. *Computer Graphics Forum 39*, 4 (2020), 121–132. 1, 2, 6

[HHdD16] Heitz E., Hanika J., d'Eon E., Dachsbacher C.: Multiple-scattering microfacet bsdfs with the smith model. *ACM Trans. Graph. 35*, 4 (jul 2016). 8

[Jak14] Jakob W.: *An Improved Visible Normal Sampling Routine for the Beckmann Distribution*. Tech. rep., 2014. 2

[LRPB23] Lucas S., Ribardiere M., Pacanowski R., Barla P.: A micrograin bsdf model for the rendering of porous layers. In *SIGGRAPH Asia 2023 Conference Papers* (New York, NY, USA, 2023), SA '23, Association for Computing Machinery. 1, 2, 4

[LRPB24] Lucas S., Ribardière M., Pacanowski R., Barla P.: A Fully-correlated Anisotropic Micrograin BSDF Model. *ACM Transactions on Graphics 43*, 4 (July 2024), 111. 1, 2, 3, 4

[LRR04] Lawrence J., Rusinkiewicz S., Ramamoorthi R.: Efficient BRDF importance sampling using a factored representation. *ACM Transactions on Graphics (Proc. SIGGRAPH)* (Aug. 2004). 2

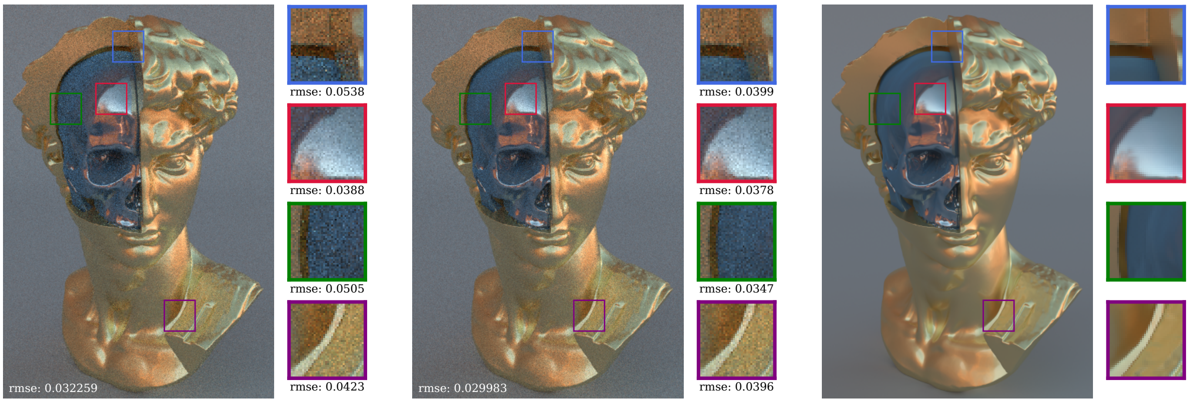
[SFA*24] Sakai H., Freude C., Auzinger T., Hahn D., Wimmer M.: A statistical approach to monte carlo denoising. In *SIGGRAPH Asia 2024 Conference Papers* (New York, NY, USA, 2024), SA '24, Association for Computing Machinery. 1

[TE24] Tokuyoshi Y., Eto K.: Bounded vndf sampling for the smith-ggx brdf. *Proc. ACM Comput. Graph. Interact. Tech. 7*, 1 (may 2024). 2

[TR75] Trowbridge T. S., Reitz K. P.: Average irregularity representation of a rough surface for ray reflection. *J. Opt. Soc. Am. 65*, 5 (May 1975), 531–536. 1

[Vea98] Veach E.: *Robust monte carlo methods for light transport simulation*. PhD thesis, Stanford, CA, USA, 1998. AAI9837162. 7

[WMLT07] Walter B., Marschner S. R., Li H., Torrance K. E.: Microfacet models for refraction through rough surfaces. In *Computer Graphics Forum, EGSR proceedings* (2007). 1, 7

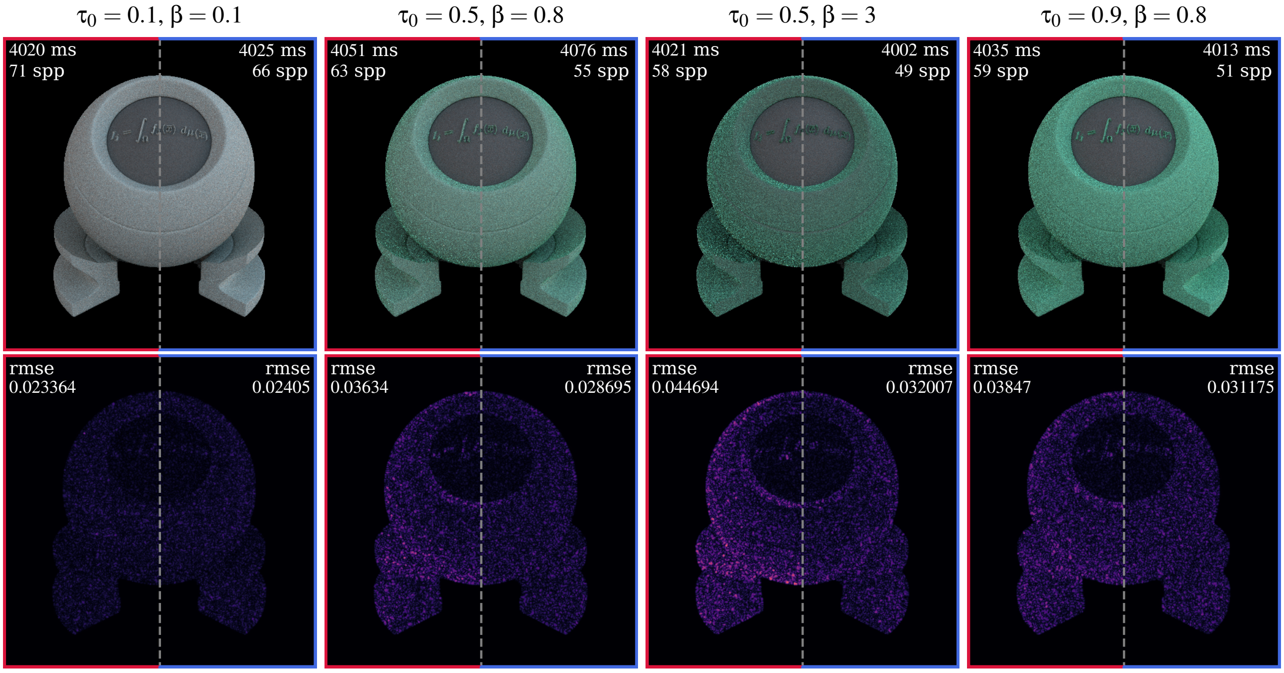


(a) NDF: 41 spp, 6015 ms

(b) vNDF: 38 spp, 6084 ms

(c) Reference: 27k spp, 68 min

Figure 7: *Multiple renderings of a statue covered by micrograins using (a) NDF sampling and (b) vNDF sampling, with (c) the reference. Insets highlights of the impact of the vNDF sampling with a noticeable noise reduction mainly and lower RMSE at grazing angles for constant time renderings. The skull has a conductor base material covered by blue diffuse micrograins (*$\tau_0 = 0.4$*,* $\beta = 3, \eta = [0.18299, 0.42108, 1.37340]$ *and* $\kappa = [3.4242, 2.34590, 1.77040]$*). The face is made of the same base material but is covered by densely packed gold micrograins (*$\tau_0 = 0.9$*,* $\beta = 0.3$*,* $K_d = [0.18299, 0.42108, 0.67340]$*).*


$\tau_0 = 0.1, \beta = 0.1$

$\tau_0 = 0.5, \beta = 0.8$

$\tau_0 = 0.5, \beta = 3$

$\tau_0 = 0.9, \beta = 0.8$

Figure 8: *Renderings with NDF sampling (red) and vNDF sampling (blue) at near constant time with green diffuse micrograins (*$K_d = [0.18299, 0.67340, 0.42108]$*) on top of a gray Lambertian base (*$K_d = 0.5$*), lit by an environment map for different filling factors* τ *and anfractuosities* β*. Both the FLIP error images and RMSE metrics, computed with respect to converged renders, illustrate a reduction in noise achieved by our method. This improvement is especially noticeable at grazing angles when compared to NDF sampling.*

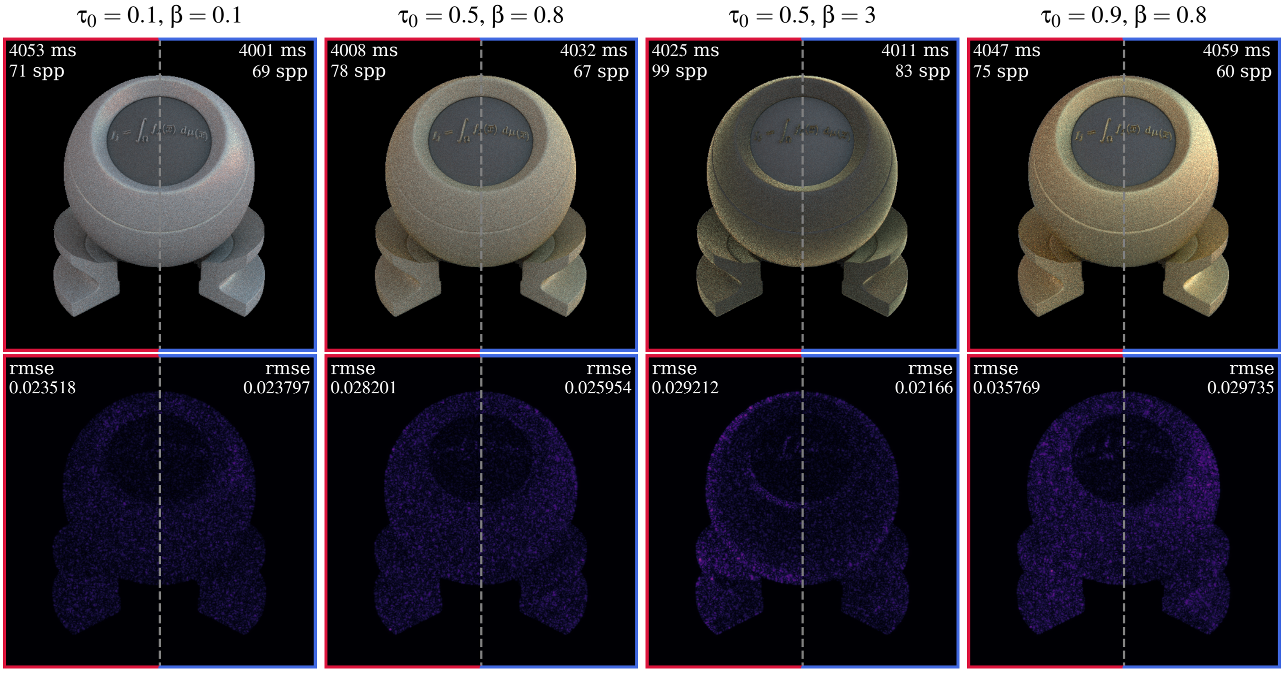


Figure 9: *Renderings with NDF sampling (red) and vNDF sampling (blue) at near constant time, with gold micrograins (*$\eta = [0.18299, 0.42108, 1.37340], \kappa = [3.4242, 2.34590, 1.77040]$*) on top of a gray Lambertian base (*$K_d = 0.5$*), lit by an environment map for different filling factors* τ *and anfractuosities* β*. Both the FLIP error images and RMSE metrics, computed with respect to converged renders, illustrate a reduction in noise with our method. This is especially noticeable at grazing angles when compared to NDF sampling.*