


Importance Sampling of Glittering BSDFs based on Finite Mixture Distributions

X. Chermain¹ , B. Sauvage¹, J.-M. Dischler¹, C. Dachsbacher²

¹Université de Strasbourg, CNRS, ICube UMR 7357, France

²Institute for Visualization and Data Analysis, Karlsruhe Institute of Technology, Germany

Our sampling using multi-lobe component
128 spp, 70.5 s

Reference
1,024 spp, 567.9 s

Previous sampling using mono-lobe component
170 spp, 81.6 s



Figure 1: A glittering coloured glass sphere with a spatially varying microfacet density. A point light inside the sphere makes the surface sparkle; an environment map further illuminates the scene. Left: our sampling scheme uses the BSDF's multi-lobe term. Centre: reference with 1,024 samples per pixel (spp). Right: the Gaussian mono-lobe approximation of the BSDF (previous work) is used for sampling. With equal render time, our new sampling technique results in less variance compared to the previous technique. Timings are measured for 512^2 image resolution.

Abstract

We propose an importance sampling scheme for the procedural glittering BSDF of Chermain et al. [CSDD20]. Glittering BSDFs have multi-lobe visible normal distribution functions (VNDFs) which are difficult to sample. They are typically sampled using a mono-lobe Gaussian approximation, leading to high variance and fireflies in the rendering. Our method optimally samples the multi-lobe VNDF, leading to lower variance and removing firefly artefacts at equal render time. It allows, for example, the rendering of glittering glass which requires an efficient sampling of the BSDF. The procedural VNDF of Chermain et al. is a finite mixture of tensor products of two 1D tabulated distributions. We sample the visible normals from their VNDF by first drawing discrete variables according to the mixture weights and then sampling the corresponding 1D distributions using the technique of inverse cumulative distribution functions (CDFs). We achieve these goals by tabulating and storing the CDFs, which uses twice the memory as the original work. We prove the optimality of our VNDF sampling and validate our implementation with statistical tests.

CCS Concepts

• Computing methodologies → Reflectance modeling;

1. Introduction

Reflectance modelling is one core concept of rendering where bidirectional scattering distribution functions (BSDFs) describe how a surface scatters light. These functions are composed of a bidirectional reflection distribution function (BRDF) and a bidirectional transmittance distribution function (BTDF). Photo-realistic rendering of a 3D scene is tantamount to computing an integral equation in which BSDFs, modelling the surfaces' materials, are included. Nowadays, Monte Carlo methods are the standard approach to do so. To converge as quickly as possible, these require efficient BSDF sampling. In this paper, we focus on efficiently sampling a complex BSDF for modelling glittering materials.

BSDFs with a mono-lobe whose orientation is strongly influenced by the mirror reflection or refraction direction describe scattering from *glossy* materials [MDH*20] such as rough metal and glass (see left part of Figure 2). Some well-known *glossy analytic* BSDFs based on the microfacet theory [CT82; WMLT07] represent glossy appearance with (possibly spatially varying) parameters such as microfacet roughness. When it comes to modelling scattering from *glittering* materials, such as glittering metal or glass, as shown in the right part of Figure 2, microfacet-based BSDFs become much more complex and scale-dependent. At fine scales, the BSDFs exhibit multiple lobes, encoding narrow directional peaks of high reflection. At larger scales lobes aggregate, converging to a mono-lobe glossy BSDF.

Glittering BSDFs have been modelled with discrete distributions of flakes [JHY*14; AK16; WWH18] and with finite mixture distributions [CSDD20; CLS*21]. The evaluation when using a discrete distribution is performed by counting reflective microfacets within a ray footprint. These models are filterable [WWH18], can be importance sampled [AK16], and can be evaluated in real-time [WDH20]. Models using finite mixture distributions [CSDD20; CLS*21] achieve high-performance renderings and are filterable, energy-conserving, and memory efficient. However, these models lack good importance sampling, a crucial variance reduction technique for Monte Carlo integration. It consists in taking samples from a probability density function (PDF) which should be as similar to the integrand (here the BSDF) as possible.

In this work, we take advantage of the mixture formulation of Chermain et al.'s [CSDD20] model and derive an efficient and optimal importance sampling scheme to replace their mono-lobe Gaussian approximation as a sampling PDF. Their BSDF is based on microfacet theory and comprises a V-cavity masking-shadowing function and a zero-mean, symmetrical, and normalised normal distribution function (NDF). These properties allow us to leverage the optimal importance sampling for the V-cavity model of Heitz and d'Eon [Hd14] using the visible normal distribution function (VNDF). Chermain et al.'s multi-lobe NDF is a mixture of tensor products of transformed, tabulated, 1D marginal distributions. To sample it, we precompute and tabulate the cumulative marginal distributions functions (CDFs). Our sampling procedure then performs a binary search on the CDFs, conditioned by discrete random sampling of the ray footprint and the levels of detail.

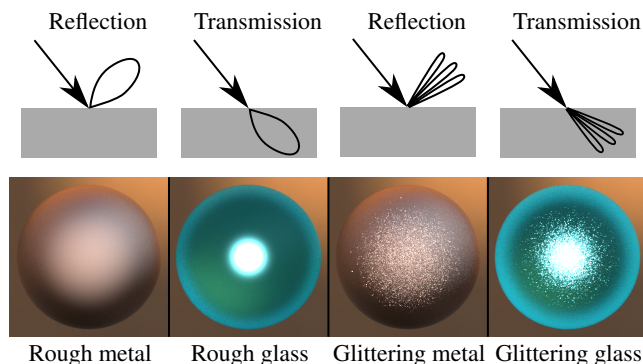


Figure 2: BSDF lobes (top) and their associated material (bottom). The glossy BSDF component is mono-lobe and models scattering from rough metals and glasses (left). The glittering BSDF component is multi-lobe and models scattering from glittering metals and glasses (right).

In this article, we make the following contributions:

- We introduce an importance sampling technique for the glittering NDF of Chermain et al. [CSDD20] (Section 4.2). We prove that the PDF of our sampling scheme equals the NDF and thus that our scheme is optimal.
- We combine it with the technique of Heitz and d'Eon [Hd14] to account for the incident direction in the sampler (Section 4.1).
- We measure the improvement of the convergence and show the removal of fireflies (Section 6).
- We provide an open source implementation of our technique in the `pbrrt-v3` renderer [PJH16]. We validate our implementation using the χ^2 statistical test [SA07] and the white furnace test (Section 5).

As demonstrated in Figure 1, our method can also render glittering transparent materials, such as glass with flakes, frosted glass, or ice. This is made possible by unifying three key properties. First, a physically based BSDF avoids energy leaks that make the surface appear too bright or too dark. Second, the BSDF is evaluated quickly. Third, an efficient importance sampling makes the rendering converge quickly. Wang et al. [WHY20] also address efficient rendering of glittering glass, but only when the rough microsurface is explicitly represented. In our case, the microsurface is represented by the procedural and multi-lobe NDF.

2. Related work

Reflectance modelling of glittering materials and importance sampling of the corresponding NDFs is closely related to microfacet theory and importance sampling of microfacet-based BSDFs. In this section, we discuss related work in these fields.

2.1. Microfacet-based glossy BSDF

Torrance and Sparrow [TS67] introduced the microfacet theory for specular reflections from rough surfaces. Later, Cook and Torrance [CT82] applied this theory to computer graphics.

Stam [Sta01] proposed an extension to transmittance, and Walter et al. [WMLT07] further explored this topic by comparing different normal distribution functions (NDFs) and describing efficient schemes for sampling the microfacet models. Heitz [Hei14] investigated the properties of the masking-shadowing functions and introduced the visible normal distribution function (VNDF). Heitz and d’Eon [Hd14] reduced the variance of Monte Carlo estimators by sampling the VNDF instead of the NDF. In our work, we use their technique to take into account the view direction.

2.2. Microfacet-based glittering BSDFs

The aforementioned works consider mono-lobe NDFs, which can only model reflection or transmission from *glossy* materials [MDH*20]. To model *glittering* materials, specialised methods using spatially varying, multi-scale, and multi-lobe NDFs are required. There are two main strategies for modelling such NDFs: They can be computed from a high-resolution normal map during rendering or they can be generated procedurally.

Normal map-based multi-lobe NDFs The methods based on normal maps are typically used to represent specular surfaces with certain details, such as scratches, waves, or glitters, and are memory intensive –in the order of at least 30 MiB per normal map for the most memory-efficient methods [WHHY20; AWKK21]. Drawing a normal map by hand is tedious because it is difficult to predict how the overall surface will look later. Therefore, the textures are usually generated procedurally before rendering, for example, by sampling a NDF to set flake orientations [AWKK21]. Moreover, these techniques are not suited for spatial variations such as in Figure 1, because it requires enormous normal maps, much larger than standard textures.

Importance sampling of the NDF contained in a ray footprint \mathcal{P} (called the \mathcal{P} -NDF) is possible. First, a position within the ray footprint \mathcal{P} is sampled, then the sampled direction is computed by reflecting the view direction with the normal at the sampled position [YHJ*14; AWKK21]. Sampling of the *visible* normal distribution function (VNDF) is also possible [CCM19].

Procedural multi-lobe NDFs Jakob et al. [JHY*14] introduced a stochastic discrete microfacet \mathcal{P} -NDF which models glittering surfaces and converges to a mono-lobe NDF as the microfacet density grows. The model is physically based, it can render highly realistic images, and the memory cost is meagre. However, the render times are high compared to materials using glossy BRDFs. This approach models a discrete set of microfacets randomly placed and oriented. The evaluation is based on a costly counting process that traverses a 4-dimensional hierarchy. Wang et al. [WWH18] approximate the 4-dimensional search by two separate 2-dimensional searches. They also store and pre-filter directional densities to increase performance when seen from a distance, while the memory load increases to 5 MiB per roughness parameter. In both papers, they use importance sampling for the converged mono-lobe NDF and not the multi-lobe \mathcal{P} -NDF; for closeups, this leads to higher variance in the Monte Carlo estimator. Later, Wang et al. extended the model [WDH20] and introduced approximations to achieve real-time performance. This method is also compatible

with real-time lighting based on environment mapping. However, it consumes a considerable amount of memory with more than 256 MiB per roughness parameter. Atanasov and Koylazov [AK16] approximate the \mathcal{P} -NDF of Jakob et al. [JHY*14] and introduce an optimal importance sampling scheme by constructing a cumulative table of weights. However, their NDF is no longer normalised. Note that the aforementioned techniques do not offer spatial variations of the model’s parameters in practice. Zirr and Kaplanyan [ZK16] proposed a fast to evaluate procedural \mathcal{P} -NDF, which is also based on a discrete counting process. However, the model is not physically based, and no importance sampling scheme is available. Hence, it is not available for physically based renderers using multiple importance sampling.

Chermain et al. [CSDD20; CLS*21] target high-performance rendering. Unlike the previous discrete processes, they define a \mathcal{P} -NDF as a mixture of procedural, multi-scale, and normalised multi-lobe NDFs. Their model is physically based (the \mathcal{P} -NDF is normalised), requires little memory (less than 1 MiB), and the model’s parameters can be spatially varying. A recent extension [CLS*21] demonstrates that their \mathcal{P} -NDF is filterable and their BRDF compatible with LEAN normal map filtering [OB10]. However, it is not suitable for offline renderers using multiple importance sampling because no efficient sampling is available. In our work, we propose an importance sampling scheme for the \mathcal{P} -VNDF of this method.

3. Evaluation of the glittering BSDF

This section first reviews microfacet-based BSDFs [WMLT07; Hei14; Hd14; HHdD16] (Section 3.1) and then the procedural glittering \mathcal{P} -NDF of Chermain et al. [CSDD20] (Section 3.2).

3.1. Evaluation of microfacet-based BSDFs

Scattering from rough surfaces is described with a microfacet-based BSDF $f(\omega_o, \omega_i)$, where ω_o and ω_i denote the observation and incident direction, respectively. The BSDF is defined as the sum of the BRDF $f_r(\omega_o, \omega_i)$ modelling reflectance and the BTDF $f_t(\omega_o, \omega_i)$ modelling transmittance. It is more convenient to evaluate and sample the BSDF weighted by the dot product term $|\omega_i \cdot \omega_g|$, where ω_g is the geometric normal of the surface. The weighted BSDF is

$$\begin{aligned} f(\omega_o, \omega_i) |\omega_i \cdot \omega_g| &= (f_r(\omega_o, \omega_i) + f_t(\omega_o, \omega_i)) |\omega_i \cdot \omega_g| \quad (1) \\ &= \frac{D_{\mathcal{P}}(\omega_{h_r}, \omega_o) G_1(\omega_{h_r}, \omega_i) F(\omega_o, \omega_{h_r})}{4 \langle \omega_o, \omega_{h_r} \rangle} + \\ &\quad \frac{D_{\mathcal{P}}(\omega_{h_t}, \omega_o) G_1(\omega_{h_t}, \omega_i) \eta_o^2 (1 - F(\omega_o, \omega_{h_t}))}{(\eta_o(\omega_o \cdot \omega_{h_t}) + \eta_i(\omega_i \cdot \omega_{h_t}))^2 \langle \omega_o, \omega_{h_t} \rangle}. \end{aligned}$$

The terms ω_{h_r} and ω_{h_t} are the half vectors of reflection and transmission, respectively, and the terms η_o and η_i are the indices of refraction of the observed and transmitted sides. The shadowing function is denoted as G_1 , and F is the Fresnel term. The \mathcal{P} -VNDF, designated as $D_{\mathcal{P}}(\omega_m, \omega_o)$, mainly defines the shape of the BSDF. In the following, we elaborate on this expression.

Representation of the microsurface In microfacet theory, a microsurface is represented by a \mathcal{P} -NDF $D_{\mathcal{P}}(\omega_m)$. It gives the den-

sity of a micronormal ω_m within an area \mathcal{P}^\dagger . The micronormals $\omega_m = (x_m, y_m, z_m)^T$ are defined in the unit hemispherical domain centred around the surface normal ω_g . If the surface is transmissive, we always define the geometric normal ω_g on the observed surface's side, i.e., $\omega_o \cdot \omega_g > 0$ and $\omega_m \cdot \omega_g > 0$

Evaluation of the patch VNDF The \mathcal{P} -VNDF $D_{\mathcal{P}}(\omega_m, \omega_o)$ is the \mathcal{P} -NDF weighted by the projected area of each micronormal $\langle \omega_o \cdot \omega_m \rangle$ and by the masking function $G_1(\omega_o, \omega_m)$. The observation direction is ω_o , and $\langle \omega_o \cdot \omega_m \rangle$ denotes the clamped dot product. In their work, Chermain et al. [CSDD20] use the V-cavity masking function [CT82]. The definition of the \mathcal{P} -VNDF is [Hei14]:

$$D_{\mathcal{P}}(\omega_m, \omega_o) = \frac{G_1(\omega_o, \omega_m) \langle \omega_o \cdot \omega_m \rangle D_{\mathcal{P}}(\omega_m)}{\omega_o \cdot \omega_g}. \quad (2)$$

Evaluation of the patch NDF The NDF is often expressed as a transformation of a patch slope distribution function (\mathcal{P} -SDF) $P_{\mathcal{P}}^{22}(\tilde{m})$:

$$(\omega_m \cdot \omega_g) D_{\mathcal{P}}(\omega_m) = \frac{P_{\mathcal{P}}^{22}(\tilde{m})}{(\omega_m \cdot \omega_g)^3}. \quad (3)$$

Here, the term $(\omega_m \cdot \omega_g) D_{\mathcal{P}}(\omega_m)$ is the NDF projected onto the tangent plane of the surface. The normal-to-slope transformation is $\tilde{m} = (\frac{-x_m}{z_m}, \frac{-y_m}{z_m})^T$, where the microslopes $\tilde{m} = (x_{\tilde{m}}, y_{\tilde{m}})^T$ are defined in \mathbb{R}^2 .

3.2. Evaluation of the glittering patch SDF

This section reminds the definition and evaluation of the patch SDF ($P_{\mathcal{P}}^{22}$ in Equation 3) of Chermain et al. [CSDD20]. They define the \mathcal{P} -SDF as a finite mixture of transformed slope distribution functions (SDFs) P_M^{22} for a discrete level of detail (LOD) l and a cell s :

$$P_{\mathcal{P}}^{22}(\tilde{m}) = \sum_{l=\lfloor l_{\mathcal{P}} \rfloor}^{\lfloor l_{\mathcal{P}} \rfloor + 1} w(l) \sum_{s \in \mathcal{P}} W_{\mathcal{P}}(l, s) P_M^{22}(\tilde{m}, l, s). \quad (4)$$

The continuous level of detail $l_{\mathcal{P}}$ is computed with the minor length of the ray footprint \mathcal{P} as for anisotropic MIP map filtering [Hec89]. The function $w(l)$ weights the adjacent discrete LODs $\lfloor l_{\mathcal{P}} \rfloor$ and $\lfloor l_{\mathcal{P}} \rfloor + 1$; this weighting is normalised, i.e., $\sum_{l=\lfloor l_{\mathcal{P}} \rfloor}^{\lfloor l_{\mathcal{P}} \rfloor + 1} w(l) = 1$. For each LOD l , the surface is subdivided into cells of integer coordinates s (the authors use a MIP grid hierarchy). The function $W_{\mathcal{P}}(l, s)$ weights each cell s within the ray footprint \mathcal{P} , and this function is also normalised, i.e., $\sum_{s \in \mathcal{P}} W_{\mathcal{P}}(l, s) = 1$. Please refer to the original article for more details concerning this glint model.

Evaluation of the transformed SDF The transformed SDF $P_M^{22}(\tilde{m}, l, s)$ (Equation 4) gives the density of the microslope \tilde{m} at LOD l within a cell s ; it is normalised, i.e., $\int_{\mathbb{R}^2} P_M^{22}(\tilde{m}, l, s) d\tilde{m} = 1$. The SDF P_M^{22} is generated by applying a linear transformation, the 2×2 matrix M , to the original slope \tilde{m}_o associated with the original

distribution $P_o^{22}(\tilde{m}_o, l, s)$. The transformation is $\tilde{m} = M\tilde{m}_o$ and the definition of the transformed SDF is:

$$P_M^{22}(\tilde{m}, l, s) = \frac{1}{\det(M)} P_o^{22}(M^{-1}\tilde{m}, l, s). \quad (5)$$

Evaluation of the original SDF The original slope $\tilde{m}_o = (x_{\tilde{m}_o}, y_{\tilde{m}_o})$ is obtained from the inverse transformation $\tilde{m}_o = M^{-1}\tilde{m}$. Each component of the original slope is used independently to evaluate the original distribution, because P_o^{22} is defined as the tensor product of its marginal distributions P_o^{2-} and P_o^{-2} :

$$P_o^{22}(\tilde{m}_o, l, s) = P_o^{2-}(x_{\tilde{m}_o}, l, s) P_o^{-2}(y_{\tilde{m}_o}, l, s). \quad (6)$$

The marginal distributions are multi-scale, multi-lobe, tabulated 1D functions and stored in a set of distributions called the dictionary in the method of Chermain et al. [CSDD20].

4. Sampling of the glittering BSDF

This section explains how to sample the distributions defined in the previous section. We begin by reviewing sampling a microfacet-based BSDF (Section 4.1) and its VNDF using the V-cavity model. Then we detail our importance sampling approach for the glittering \mathcal{P} -SDF (Section 4.2).

4.1. Sampling of the microfacet-based BSDF

Heitz and d'Eon [Hd14] detail importance sampling microfacet-based BSDFs using VNDFs. In Algorithm 1, we summarise their method using the V-cavity model. The probability density functions (PDFs) resulting from the sampling procedures are shown in the algorithm. The terms U_1 and U_2 are random numbers with uniform probability $\mathcal{U}_{[0,1]}$.

Algorithm 1: Sampling of $f(\omega_o, \omega_i) | \omega_i \cdot \omega_g$ (Eq. 1).

$\tilde{m} \leftarrow$ Sample PDF(\tilde{m}) = $P_{\mathcal{P}}^{22}(\tilde{m})$ (Eq. 4) \triangleright Alg. 2.

\triangleright PDF(ω_m) = $(\omega_m \cdot \omega_g) D_{\mathcal{P}}(\omega_m)$ (Eq. 3)

$\omega_m \leftarrow \frac{(-x_{\tilde{m}}, -y_{\tilde{m}}, 1)^T}{\sqrt{x_{\tilde{m}}^2 + y_{\tilde{m}}^2 + 1}}$

\triangleright PDF(ω_m, ω_o) = $D_{\mathcal{P}}(\omega_m, \omega_o)$ (Eq. 2)

$\omega'_m \leftarrow (-x_m, -y_m, z_m)^T$ \triangleright Symmetrical micronormal

projArea $\leftarrow \frac{\langle \omega_o, \omega'_m \rangle}{\langle \omega_o, \omega_m \rangle + \langle \omega_o, \omega'_m \rangle}$ \triangleright Projected area of ω'_m

if $U_1 < \text{projArea}$ **then**

$\omega_m \leftarrow \omega'_m$

end

1 if $U_2 < F(\omega_o, \omega_m)$ **then**

\triangleright PDF(ω_o, ω_i) = $\frac{D_{\mathcal{P}}(\omega_{h_r}, \omega_o) F(\omega_o, \omega_{h_r})}{4 \langle \omega_o, \omega_{h_r} \rangle}$

$\omega_i \leftarrow \text{reflect}(\omega_o, \omega_m)$

else

\triangleright PDF(ω_o, ω_i) = $\frac{D_{\mathcal{P}}(\omega_{h_t}, \omega_o) \eta_o^2 (1 - F(\omega_o, \omega_{h_t}))}{(\eta_o \langle \omega_o, \omega_{h_t} \rangle + \eta_i \langle \omega_i, \omega_{h_t} \rangle)^2 \langle \omega_o, \omega_{h_t} \rangle}$

$\omega_i \leftarrow \text{transmit}(\omega_o, \omega_m)$

end

[†] In this article, the patch \mathcal{P} is the ray footprint, but usually, the patch \mathcal{P} is not mentioned and is set to the unit area for convenience

4.2. Sampling of the glittering patch SDF

In the following, we detail each part of our sampling technique which is summarised in Algorithm 2. In Appendix A, we prove its correctness, i.e., that it optimally samples $P_{\mathcal{P}}^{22}$.

Algorithm 2: Sampling of $P_{\mathcal{P}}^{22}(\tilde{m})$ (Eq. 4).

- ▷ Sample a LOD.
 - 1 $l \leftarrow$ Sample PMF(l) = $w(l)$

 - ▷ Sample a cell given a LOD.
 - 2 $s \leftarrow$ Sample PMF(s) = $W_{\mathcal{P}}(l, s)$

 - ▷ Sample a microslopes given a LOD and a cell
 - 3 ▷ PDF(\tilde{m}_o) = $P_o^{22}(\tilde{m}_o, l, s)$ (Eq. 6)
 - $x_{\tilde{m}_o} \leftarrow$ Sample PDF($x_{\tilde{m}_o}$) = $P_o^{2-}(x_{\tilde{m}_o}, l, s)$
 - $y_{\tilde{m}_o} \leftarrow$ Sample PDF($y_{\tilde{m}_o}$) = $P_o^{-2}(y_{\tilde{m}_o}, l, s)$
 - $\tilde{m}_o = (x_{\tilde{m}_o}, y_{\tilde{m}_o})^T$

 - ▷ PDF(\tilde{m}) = $P_M^{22}(\tilde{m}, l, s)$ (Eq. 5)
 - 4 $\tilde{m} \leftarrow M\tilde{m}_o$ ▷ Transformation of the original sample
-

Sampling the finite mixture distribution The glittering patch SDF $P_{\mathcal{P}}^{22}(\tilde{m})$ (Equation 4) is called a *finite mixture distribution*: it is a mixture over LODs of a mixture over cells of distributions. To sample this mixture, a LOD and a cell are first drawn as *discrete* random variables according to a probability mass function (PMF) defined by the mixture weights. Then, a microslopes is drawn as a *continuous* random variable according to the PDF P_M^{22} .

Sampling the LOD l (Alg. 2, line 1). The PMF for l has two events: $\lfloor l_{\mathcal{P}} \rfloor$ and $\lfloor l_{\mathcal{P}} \rfloor + 1$, with probabilities $w(\lfloor l_{\mathcal{P}} \rfloor)$ and $1 - w(\lfloor l_{\mathcal{P}} \rfloor)$. As for the Fresnel term (Alg. 1, line 1), its sampling is performed by testing $w(\lfloor l_{\mathcal{P}} \rfloor)$ against a uniform random variable.

Sampling the cell s (Alg. 2, line 2). For any given ray footprint \mathcal{P} and LOD l , the PMF for s equals $W_{\mathcal{P}}(l, s)$ and has a low number of events equal to the number of cells $s \in \mathcal{P}$. Thus s can be drawn on the fly by a) computing the PMF, b) computing the discrete cumulative distribution function (CDF), c) drawing a uniform random number $U \in [0, 1]$, and d) performing a binary search of the larger cell index s such that $\text{CDF}(s) \leq U$. This index gives the sampled cell.

Sampling of the original SDF (Alg. 2, line 3). The 2D joint density function $P_o^{22}(\tilde{m}_o, l, s)$ is separable (see Equation 6) and can be expressed as the product of its 1D marginal densities. In this case, random variables can be found by independently sampling from $P_o^{2-}(x_{\tilde{m}_o}, l, s)$ and $P_o^{-2}(y_{\tilde{m}_o}, l, s)$ using inverse CDF sampling. Since the marginal distributions are tabulated in a precomputed dictionary, we precompute and tabulate the CDF of all of them. Using a precomputed CDF is standard practice for sampling piecewise linear 1D distributions [PJH16]. Then we sample from this CDF on the fly using a binary search, in the same way as we sample the

cells s . Currently, the tabulated PDF and CDF size is 64, requiring six fetches to sample a slope.

An alternative with a single fetch could consist in storing the inverse CDF directly. However, it would cause memory and precision issues. Indeed, the input argument of the CDF is a slope $\tilde{m} \in \mathbb{R}$. The domain of the slopes is uniformly sampled, but the domain of the CDF values may not be. Thus, it would require resampling the CDF values uniformly. Unless using a memory consuming high-frequency resampling, this would introduce numerical errors that lead to a failure of the χ^2 validation tests. For future research, it would be interesting to investigate such a precision versus speedup trade-off.

Sampling of the transformed SDF (Alg. 2, line 4). Transforming samples \tilde{m}_o from the distribution $P_o^{22}(\tilde{m}_o, l, s)$ with the transformation $\tilde{m} = M\tilde{m}_o$ yields samples proportional to the PDF $\frac{1}{\det(M)} P_o^{22}(M^{-1}\tilde{m}, l, s)$. The density of the sample \tilde{m} is exactly $P_M^{22}(\tilde{m}, l, s)$ (Equation 5).

5. Implementation and validation

We have implemented our importance sampling in the `pbrt-v3` renderer [PJH16]. We also provide the code and scripts to reproduce the results and the validation tests of the implementation. Please see the supplemental document 2/2.

Timings and performance comparisons are performed on an Intel(R) Core(TM) i9-9900 CPU (16 threads, 3.10 GHz). Our memory usage is 384 KiB (CDFs included): the dictionary used in all the renderings comprises 786 1D distributions with eight levels of detail for each. For rendering, we use the forward path tracer because `pbrt-v3` only supports ray differentials [Ige99], and therefore the pixel footprint is only available for the first ray intersection. For intersections with a depth greater than one, we use the mono-lobe glossy BSDF (Figure 2, left) instead of the multi-lobe glittering BSDF (Figure 2, right) as in the method of Jakob et al. [JHY*14].

In Appendix A, we prove that our importance sampling scheme is optimal for $P_{\mathcal{P}}^{22}$. To validate our implementation, we carried out a χ^2 test and a white furnace test. The χ^2 test compares the PDF with the histogram built by sampling this PDF. Passing the test means that the sampling does not deviate from the distribution (i.e. statistically no significant difference). Our implementation passed the test for the VNDF (Equation 2) as well as the BRDF and BSDF (Equation 1).

The white furnace test places the virtual material in a uniformly white environment [Hei14]. Thus the outgoing radiance is the integral of Equation 1, which is equal to the average of the shadowing term G_1 . This integral is computed with a Monte-Carlo estimator using importance sampling (formulas are provided in the supplemental document 1/2). Figure 3 shows the white furnace tests for the dielectric and conducting spheres in Figures 1 and 4. Due to the white environment, the glints should not be visible, and thus the bright pixels are fireflies. Conversely, with a sampler using the mono-lobe approximation of the BSDF, many fireflies are visible, even above 1,000 samples per pixel for the dielectric material. Furthermore, the estimator converges slowly compared to our estimator using importance sampling of the multi-lobe component.

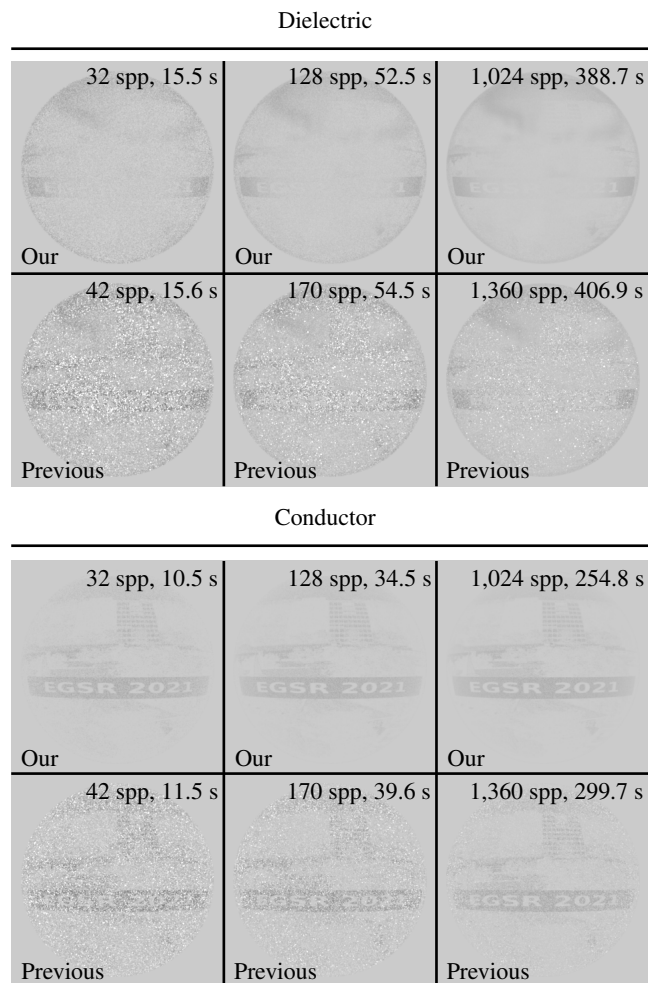


Figure 3: White furnace tests of conductors and dielectrics with spatially varying microfacet density. Our sampling provides faster convergence. Timings are measured for 512^2 image resolution.

6. Results

Figure 1 shows spheres rendered under environment illumination. The material is a glittering coloured glass modelled with the BSDF of Equation (1). The inscriptions on the spheres are produced by spatial variations of the glint density with a texture. The base material (where there is no glitter) is modelled with a glossy BSDF with very low roughness. The same scene is shown in Figure 4, where the material is a glittering copper modelled with the BRDF only. As shown in the close-ups, with similar render time, our sampling scheme achieves results closer to the reference than the previous sampling technique using the mono-lobe approximation. In particular, it reduces firefly artefacts that are due to very substantial weights in the Monte Carlo estimator. These artefacts are even more visible in the accompanying video.

In Figure 5, a more complex scene exhibits shadows, indirect lighting (reflected by the dog), spatial variations of the microfacet

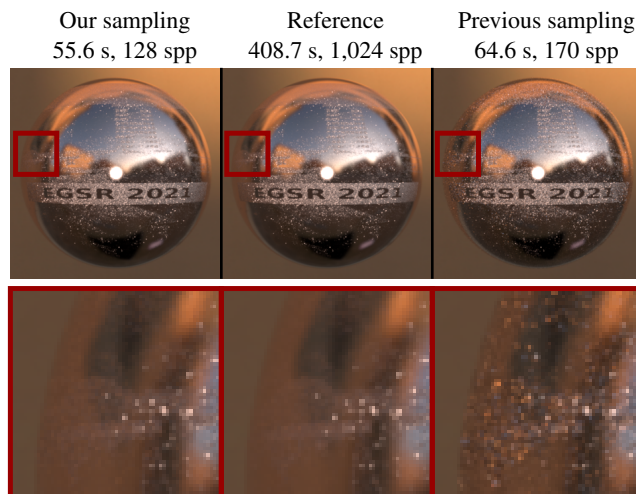


Figure 4: Comparison of our sampling scheme (left) with the previous sampling scheme (right) for similar rendering times. The reference (middle) is obtained with 1,024 samples per pixels (spp). A glittering copper sphere with spatially varying microfacet density is illuminated by an environment map and a point light. Our sampler converges faster, and there are fewer firefly artefacts. Timings are measured for 512^2 image resolution.

density (chessboard, cat), and transparency (Suzanne). We compare our sampling method (top) to the previous sampling technique using the mono-lobe approximation (bottom). The top image has less variance for a similar rendering time (please also see the insets and the accompanying video).

Due to the extra cost for sampling from the multi-lobe component of the BSDF, we observe that our sampling scheme draws roughly 25% fewer samples for a similar rendering time, yet it still converges faster.

To measure the performance of our importance sampling, we monitor the convergence of the integral of the glittering BSDF as a function of the number N of samples with different sets of parameters (width of the patch \mathcal{P} , incidence angle of the view direction, roughness). For each set of parameters, we run 1,000 realisations of the estimator and store the 1,000 resulting curves. To make the graphs more readable and statistically representative, we plot the curves of pointwise quartiles $F_0\%$, $F_{25\%}$, $F_{50\%}$, $F_{75\%}$ and $F_{100\%}$ defined as pointwise boxplots. We focus here on salient observations and provide mathematical details and extensive results in the supplemental document 1/2.

Figure 6 compares our BSDF sampler (green) to the mono-lobe approximation (red) in the most challenging sampling situations: grazing angle observation, rough surfaces, and dielectric materials. We make the following observations:

1. Both samplers converge to the same value.
2. Our BSDF sampler achieves faster convergence in the worst case, as shown by the extremum curves $F_0\%$ and $F_{100\%}$ (dotted lines).

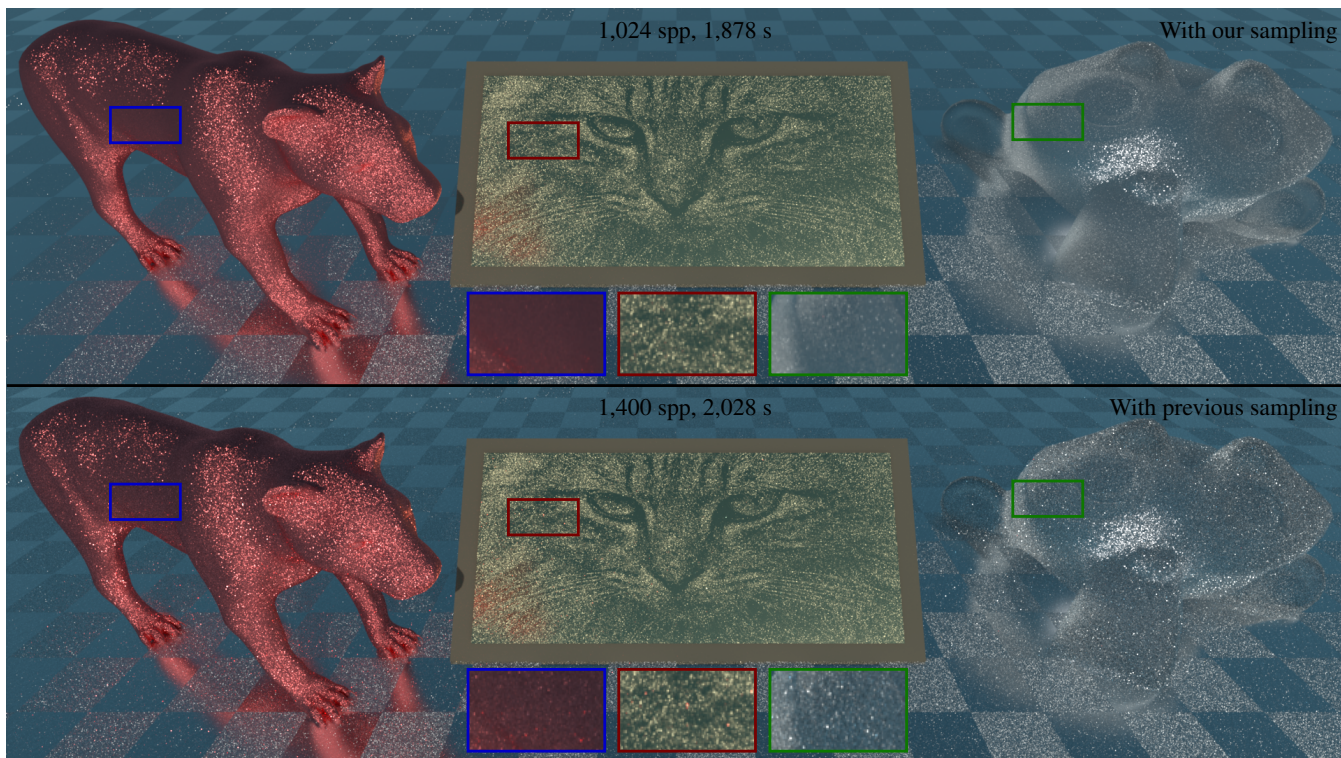


Figure 5: Comparison of our sampling (top) to the sampling using the mono-lobe approximation (bottom). Even with 1,400 samples per pixel, the previous sampling scheme exhibits firefly artefacts. Timings are measured for 1920×540 image resolution.

3. It has less variance: the inter-quartile $F_{75\%} - F_{25\%}$ is lower.
4. The Monte Carlo weights never exceed one because, in our case, the integrand divided by the PDF is the term G_1 .
5. Our BSDF sampler is insensitive to the number of microfacets within the patch \mathcal{P} . Conversely, the mono-lobe sampling scheme converges quicker as the number of microfacets increases because $P_{\mathcal{P}}^{22}$ converges towards the BSDF mono-lobe. Note that in practice, it only becomes competitive with our scheme when the glints become invisible.
6. The incidence angle has an impact on the convergence (see the supplemental document 1/2). The lower the angle, the better the convergence: this is due to the shadowing term in Equation 1, which makes the dielectric BSDF sampling sub-optimal when the incidence increases.
7. The roughness, for the same reason, has the same effect as the incidence angle.

Measuring the rendering speed-up is a delicate task. It depends on the complexity of the scene, the type of materials, and the lighting environment. It would also require a quantitative measure. Figure 3 is meant to give an idea of the speed-up. For the conductor and the dielectric, our scheme with 32 spp has less observable variance than the previous method with 1,360 spp. For this scene, we estimate our technique to converge 40 times faster.

7. Limitations and future work

In this paper, we discussed importance sampling of glittering BSDFs at three levels. (1) The SDF (Equation 4) is optimally sampled, as well as (2) the VNDF (Equation 2); the view direction and the masking effects (G_1 term in Equation 2) are taken into account using the V-cavity model. (3) Sampling the BSDF as a whole is still sub-optimal in our method because of the shadowing term. The BSDF (Equation 1) assumes single scattering of light within the microsurface. Therefore, samples that are shadowed are lost. The BSDF should benefit from methods simulating multiple scattering inside the V-Cavity [LJJ*18; XH18], and, in this case, the sampling would be optimal.

Our current implementation is based on ray differentials [Ige99]; thus, the ray footprint is only available for the first ray intersection. Therefore, we use the mono-lobe approximation of the glittering BSDF (glossy case) for the following ray intersections, as it does not require a ray footprint to be evaluated. With this approximation, glints cannot be visible by reflection from rough surfaces. This effect would require covariance tracing [BYRN17] or the ray-cones method [ACB*21].

As for the glossy BSDF model, the microgeometry cannot be observed with the glittering BSDF, only the reflection or transmittance from it. We suggest using normal map based methods to this end. An interesting future work could be converting the explicit mi-

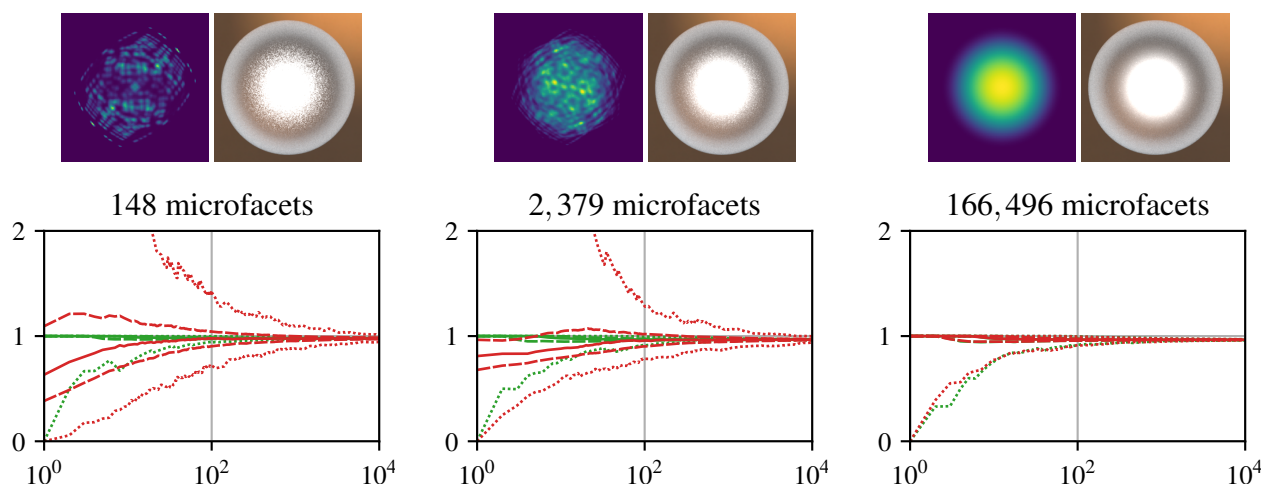


Figure 6: Convergence of our sampling scheme (green curves) compared to the sampling of the mono-lobe (red curves) as functions of the number of samples. Top row: plot of the projected \mathcal{P} -NDF (Equation 3) and rendering of the dielectric material. The curves of pointwise quartiles are plotted: the median $F_{50\%}$ in solid line; $F_{25\%}$ and $F_{75\%}$ in dashed lines; $F_{0\%}$ and $F_{100\%}$ in dotted lines. The material is dielectric, the incidence angle is grazing (1.5 rad), and the roughness is high ($\alpha = 0.6$). From left to right, there are $K = 148, 2,379,$ and $166,496$ microfacets within the pixel footprint. With our scheme, the convergence is faster, and the variance is lower.

crogeometry in the ray footprint into glittering BSDF parameters, like LEAN [OB10] and LEADR [DHI*13] mapping do with glossy BRDFs.

8. Conclusion

This paper proposed a scheme to importance sample the multi-lobe glittering BSDF of Chermain et al. [CSDD20]. This BSDF was used for real-time rendering, and it is now suitable for offline photo-realistic physically based rendering. The core algorithm optimally samples the slope distribution function, which is a finite mixture distribution. Then the visible normal distribution function is optimally sampled, based on the work of Heitz and d’Eon [Hd14]. Compared to the previous mono-lobe approximation, it results in a much faster convergence of the Monte-Carlo integration, allowing for high-quality rendering of glittering materials, including transparent materials such as glass. We provide an open source implementation, which is statistically validated.

Acknowledgements

This work has been published under the framework of the IdEx Unistra supported by the Investments for the future program of the french government. This work has been partially funded by the project ReProcTex from the Agence Nationale de la Recherche (ANR-19-CE33-0011-01) and the Deutsche Forschungsgemeinschaft (project 431478017).

References

[ACB*21] AKENINE-MÖLLER, TOMAS, CRASSIN, CYRIL, BOKSAN-SKY, JAKUB, et al. “Improved Shader and Texture Level of Detail Using Ray Cones”. *Journal of Computer Graphics Techniques* 10.1 (2021), 1–24 7.

[AK16] ATANASOV, ASEN and KOYLAZOV, VLADIMIR. “A Practical Stochastic Algorithm for Rendering Mirror-like Flakes”. *ACM SIGGRAPH Talks*. 2016 2, 3.

[AWKK21] ATANASOV, ASEN, WILKIE, ALEXANDER, KOYLAZOV, VLADIMIR, and KRIVÁNEK, JAROSLAV. “A Multiscale Microfacet Model Based on Inverse Bin Mapping”. *Comput. Graph. Forum (Proc. of Eurographics)* 40.2 (2021) 3.

[BYRN17] BELCOUR, LAURENT, YAN, LING-QI, RAMAMOORTHY, RAVI, and NOWROUZEZAHRAI, DEREK. “Antialiasing Complex Global Illumination Effects in Path-space”. *ACM Trans. Graph.* 36.1 (2017) 7.

[CCM19] CHERMAIN, XAVIER, CLAUX, FRÉDÉRIC, and MÉRILLOU, STÉPHANE. “Glint Rendering based on a Multiple-Scattering Patch BRDF”. *Comput. Graph. Forum (Proc. Eurographics Symposium on Rendering)* 38.4 (2019), 27–37 3.

[CLS*21] CHERMAIN, XAVIER, LUCAS, SIMON, SAUVAGE, BASILE, et al. “Real-Time Geometric Glint Anti-Aliasing with Normal Map Filtering”. *Proc. ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games* 4.1 (2021) 2, 3.

[CSDD20] CHERMAIN, XAVIER, SAUVAGE, BASILE, DISCHLER, JEAN-MICHEL, and DACHSBACHER, CARSTEN. “Procedural Physically-based BRDF for Real-Time Rendering of Glints”. *Comput. Graph. Forum (Proc. Pacific Graphics)* 39.7 (2020), 243–253 1–4, 8.

[CT82] COOK, R. L. and TORRANCE, K. E. “A Reflectance Model for Computer Graphics”. *Comput. Graph. (Proc. SIGGRAPH)* 1.1 (1982), 7–24 2, 4.

[DHI*13] DUPUY, JONATHAN, HEITZ, ERIC, IEHL, JEAN-CLAUDE, et al. “Linear Efficient Antialiased Displacement and Reflectance Mapping”. *ACM Trans. Graph.* 32.6 (2013), 211:1–211:11 8.

[Hd14] HEITZ, E. and D’EON, E. “Importance Sampling Microfacet-Based BSDFs using the Distribution of Visible Normals”. *Comput. Graph. Forum (Proc. Eurographics Symposium on Rendering)* 33.4 (2014), 103–112 2–4, 8.

[Hec89] HECKBERT, PAUL S. “Fundamentals of Texture Mapping and Image Warping”. MA thesis. University of California, Berkeley, 1989 4.

- [Hei14] HEITZ, ERIC. “Understanding the Masking-Shadowing Function in Microfacet-Based BRDFs”. *Journal of Computer Graphics Techniques* 3.2 (2014), 48–107 3–5.
- [HHdD16] HEITZ, ERIC, HANIKA, JOHANNES, D’EON, EUGENE, and DACHSBACHER, CARSTEN. “Multiple-Scattering Microfacet BSDFs with the Smith Model”. *ACM Trans. Graph. (Proc. SIGGRAPH)* 35.4 (2016) 3.
- [Ige99] IGEHY, HOMAN. “Tracing Ray Differentials”. *Proc. ACM SIGGRAPH*. 1999, 179–186 5, 7.
- [JHY*14] JAKOB, WENZEL, HAŠAN, MILOŠ, YAN, LING-QI, et al. “Discrete Stochastic Microfacet Models”. *ACM Trans. Graph. (Proc. SIGGRAPH)* 33.4 (2014) 2, 3, 5.
- [LJJ*18] LEE, JOO HO, JARABO, ADRIAN, JEON, DANIEL S., et al. “Practical Multiple Scattering for Rough Surfaces”. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* 37.6 (2018) 7.
- [MDH*20] MCGUIRE, MORGAN, DORSEY, JULIE, HAINES, ERIC, et al. “A Taxonomy of Bidirectional Scattering Distribution Function Lobes for Rendering Engineers”. *8th annual Workshop on Material Appearance Modeling*. June 2020, 4 2, 3.
- [OB10] OLANO, MARC and BAKER, DAN. “LEAN Mapping”. *Proc. ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*. 2010, 181–188 3, 8.
- [PJH16] PHARR, MATT, JAKOB, WENZEL, and HUMPHREYS, GREG. *Physically Based Rendering: From Theory to Implementation*. 3rd. Morgan Kaufmann Publishers Inc., 2016. ISBN: 0128006455, 9780128006450 2, 5.
- [SA07] SUBR, K. and ARVO, J. “Statistical Hypothesis Testing for Assessing Monte Carlo Estimators: Applications to Image Synthesis”. *Proc. Pacific Graphics*. 2007, 106–115 2.
- [Sta01] STAM, JOS. “An Illumination Model for a Skin Layer Bounded by Rough Surfaces”. *Proc. Eurographics Workshop on Rendering*. 2001, 39–52 3.
- [TS67] TORRANCE, K. E. and SPARROW, E. M. “Theory for Off-Specular Reflection From Roughened Surfaces”. *Journal of the Optical Society of America* 57.9 (1967), 1105–1114 2.
- [WDH20] WANG, BEIBEI, DENG, HONG, and HOLZSCHUCH, NICOLAS. “Real-Time Glints Rendering With Pre-Filtered Discrete Stochastic Microfacets”. *Comput. Graph. Forum* (2020) 2, 3.
- [WHHY20] WANG, BEIBEI, HAŠAN, MILOŠ, HOLZSCHUCH, NICOLAS, and YAN, LING-QI. “Example-Based Microstructure Rendering with Constant Storage”. *ACM Trans. Graph.* 39.5 (2020) 3.
- [WHY20] WANG, BEIBEI, HAŠAN, MILOŠ, and YAN, LING-QI. “Path Cuts: Efficient Rendering of Pure Specular Light Transport”. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* 39.6 (2020) 2.
- [WMLT07] WALTER, BRUCE, MARSCHNER, STEPHEN R., LI, HONGSONG, and TORRANCE, KENNETH E. “Microfacet Models for Refraction Through Rough Surfaces”. *Proc. Eurographics Symposium on Rendering*. 2007, 195–206 2, 3.
- [WWH18] WANG, BEIBEI, WANG, LU, and HOLZSCHUCH, NICOLAS. “Fast Global Illumination with Discrete Stochastic Microfacets Using a Filterable Model”. *Comput. Graph. Forum (Proc. Pacific Graphics)* 37.7 (2018), 55–64 2, 3.
- [XH18] XIE, FENG and HANRAHAN, PAT. “Multiple Scattering from Distributions of Specular V-Grooves”. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* 37.6 (2018) 7.
- [YHJ*14] YAN, LING-QI, HAŠAN, MILOŠ, JAKOB, WENZEL, et al. “Rendering Glints on High-Resolution Normal-Mapped Specular Surfaces”. *ACM Trans. Graph. (Proc. SIGGRAPH)* 33.4 (2014) 3.
- [ZK16] ZIRR, TOBIAS and KAPLANYAN, ANTON S. “Real-time Rendering of Procedural Multiscale Materials”. *Proc. ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*. 2016, 139–148 3.

Appendix A: Demonstration of our algorithm’s correctness.

Here we prove that Algorithm 2 provides an optimal sampling of $P_{\mathcal{P}}^{22}$, by showing that the joint probability of $X_{\tilde{m}}$ and $Y_{\tilde{m}}$ is equal to Equation (4). To do so, we derive the probability distributions for the random variables $L, S, X_{\tilde{m}_o}, Y_{\tilde{m}_o}, X_{\tilde{m}}$ and $Y_{\tilde{m}}$.

From the algorithm, we have the probabilities

$$P(L = l) = w(l) \quad (7)$$

$$P(S = s | L = l) = W(l, s), \quad (8)$$

and the densities

$$p(X_{\tilde{m}_o} = x_{\tilde{m}_o} | S = s, L = l) = P_o^{2-}(x_{\tilde{m}_o}, l, s) \quad (9)$$

$$p(Y_{\tilde{m}_o} = y_{\tilde{m}_o} | S = s, L = l) = P_o^{-2}(y_{\tilde{m}_o}, l, s). \quad (10)$$

For any fixed values l and s , $X_{\tilde{m}_o}$ and $Y_{\tilde{m}_o}$ are independent, so their joint probability is the product (Equation 6) of the marginal distributions (Equation 9 and (10)):

$$p\left(\begin{bmatrix} X_{\tilde{m}_o} \\ Y_{\tilde{m}_o} \end{bmatrix} = \begin{bmatrix} x_{\tilde{m}_o} \\ y_{\tilde{m}_o} \end{bmatrix} | S = s, L = l\right) = P_o^{22}\left(\begin{bmatrix} x_{\tilde{m}_o} \\ y_{\tilde{m}_o} \end{bmatrix}, l, s\right). \quad (11)$$

Using a change of variable and equation (5), we have

$$p\left(\begin{bmatrix} X_{\tilde{m}} \\ Y_{\tilde{m}} \end{bmatrix} = \begin{bmatrix} x_{\tilde{m}} \\ y_{\tilde{m}} \end{bmatrix} | S = s, L = l\right) = P_M^{22}\left(\begin{bmatrix} x_{\tilde{m}} \\ y_{\tilde{m}} \end{bmatrix}, l, s\right). \quad (12)$$

From the law of total probability

$$p\left(\begin{bmatrix} X_{\tilde{m}} \\ Y_{\tilde{m}} \end{bmatrix} | L = l\right) = \sum_s P(S = s | L = l) p\left(\begin{bmatrix} X_{\tilde{m}} \\ Y_{\tilde{m}} \end{bmatrix} | S = s, L = l\right) \quad (13)$$

and thus, from (8) and (12)

$$p\left(\begin{bmatrix} X_{\tilde{m}} \\ Y_{\tilde{m}} \end{bmatrix} = \begin{bmatrix} x_{\tilde{m}} \\ y_{\tilde{m}} \end{bmatrix} | L = l\right) = \sum_s W(l, s) P_M^{22}\left(\begin{bmatrix} x_{\tilde{m}} \\ y_{\tilde{m}} \end{bmatrix}, l, s\right). \quad (14)$$

From the law of total probability, again,

$$p\left(\begin{bmatrix} X_{\tilde{m}} \\ Y_{\tilde{m}} \end{bmatrix}\right) = \sum_l P(L = l) p\left(\begin{bmatrix} X_{\tilde{m}} \\ Y_{\tilde{m}} \end{bmatrix} | L = l\right) \quad (15)$$

and thus from (7), (14) and (4)

$$p\left(\begin{bmatrix} X_{\tilde{m}} \\ Y_{\tilde{m}} \end{bmatrix} = \begin{bmatrix} x_{\tilde{m}} \\ y_{\tilde{m}} \end{bmatrix}\right) = P_{\mathcal{P}}^{22}\left(\begin{bmatrix} x_{\tilde{m}} \\ y_{\tilde{m}} \end{bmatrix}\right). \quad (16)$$