This is a post-peer-review, pre-copyedit version of an article published in Vis. Comput.
The final authenticated version is available online at: http://dx.doi.org/10.1007/s00371-018-1606-7

A microfacet based BRDF for the accurate and efficient rendering of high definition specular normal maps

Xavier Chermain[1] · **Frédéric Claux**[1] · **Stéphane Mérillou**[1]

Received: 30 Mar 2018 / Accepted: 12 Oct 2018

Abstract Complex specular microstructures found in glittery, scratched or brushed metal materials exhibit high frequency variations in reflected light intensity. These variations are important for the human eye and give materials their uniqueness and personality. To model such microsurfaces, high definition normal maps are very effective. The works of Yan et al. [21,22] enable the rendering of such material representations by evaluating a microfacet based BRDF related to a whole ray footprint. Still, in specific configurations and especially at grazing angles, their method does not fully capture the expected material appearance. We propose to build upon their work and tackle the problem of accuracy using a more physically based reflection model. To do so, the normal map is approximated with a mixture of anisotropic, noncentered Beckmann normal distribution functions from which a closed form for the masking-shadowing term can be derived. Based on our formal definition, we provide a fast approximation leading to a performance overhead varying from 5% to 20% compared to the method of Yan et al. [22]. Our results show that we more closely match ground truth renderings than their methods.

Keywords microfacet · BRDF · specular normal maps · microstructures · glints

Xavier Chermain
E-mail: xavier.chermain@unilim.fr
Frédéric Claux
E-mail: frederic.claux@unilim.fr
Stéphane Mérillou
E-mail: stephane.merillou@unilim.fr

[1] Univ. Limoges, CNRS, XLIM, UMR 7252, F-87000 Limoges, France.

1 Introduction

In the real world, materials are recognizable by their way of reflecting light: their microgeometry plays an important role in their appearance. In computer graphics, complex specular microstructures modeled by normal maps exhibit many microscopic details under sharp lighting, visible thanks to spatially varying high intensity reflections. Uniform pixel sampling is not efficient in this case, because the samples can fall next to the tiny portions of the normal map that reflect light. Normal map filtering using mipmapped textures is ineffective, as it averages the slopes and flattens out the real specular highlights. Glint rendering integrators are needed to take care of these complex lighting situations in a physically based renderer (Figure 1).

For this reason, several works deal with multi-scale specular microstructures rendering. Some methods enable the rendering of glittery [10] or scratched surfaces [15]. However, graphics artists tend to prefer normal map based representations, as they leave a lot of flexibility for material appearance. The works of Yan et al. [21, 22] enable the rendering of such highly detailed, versatile microsurfaces through the use of dedicated glint integrators. While demonstrating impressive results, these works do not completely address the problem of faithfully capturing the appearance of the normal mapped surfaces they propose themselves to replicate (Figure 1). Their methods also do not exhibit energy conservation, a property for BRDFs contributing to realistic light behavior at surface contact [11].

The model proposed by Yan et al. [21,22] uses a microfacet formulation. Heitz [8] has shown that a microfacet reflection model is physically based only when its projected area and masking function are correctly derived from its Normal Distribution Function (NDF).

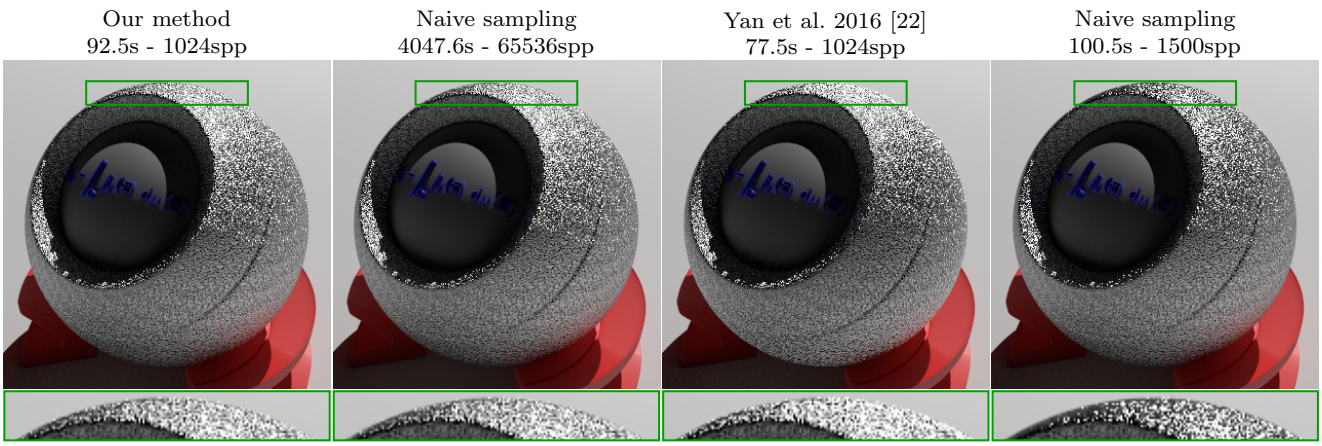

Fig. 1: Metallic orb with a high definition, perfectly specular normal map illuminated by a point light. Naive pixel sampling rendering with 1500 samples per pixel (spp) (right) using about the same time budget as our method (left) does not capture all the glare and needs much more samples, i.e. 65536spp for this scene (center left). Our method rapidly captures all the glints and more successfully matches the shininess of the reference material than the method of Yan et al. [22] (center right).

In this paper, we introduce a microsurface formulation from which an exact masking term and projected area can be correctly derived, enabling energy conservation. We also propose approximations for these terms exhibiting only 20% slower performance than the method of Yan et al. [22] in worst cases. Finally, we compare both our work and prior methods to renderings obtained through brute force, naive pixel sampling of reference, interpolated normal-mapped surfaces.

2 Previous work

Complex specular surfaces are difficult to reproduce in a renderer, especially if details are introduced by the use of a normal map. The accurate NDF contained in a ray footprint extracted from a high definition normal map results in a BRDF with many sharp lobes, sometimes more than one hundred. Several works try to approximate the reflection model with either one [18,13,5] or more [6,20] lobes using parametric functions, inevitably losing many subtle details of the full BRDF.

Glittery surfaces The method of Jakob et al. [10] and the improvements proposed by Atanasov and Koylazov [1] describe the microsurface with a collection of random, small discrete mirrors. Their microfacet based BRDF is evaluated by a procedural process calculating the number of specular flakes having a normal halfway between the lighting and viewing directions. For the masking-shadowing term, no information is given and they probably use the one of a smooth NDF. They do not compare their results with a reference because the flakes of the material do not have an explicit location per se, only their density is controllable. They cannot model scratched, brushed or bumpy surfaces, only glittery surfaces.

Scratched surfaces The rendering of scratched metal is another challenging task in computer graphics [12,4, 3]. There again, tiny scratches can influence the intensity of a pixel, despite of having a much smaller size. Raymond et al. [15] propose a dedicated model for this kind of material. They reconstruct a BRDF for a single mirror scratch using a 2D ray tracer, taking into account multiple scattering. At render time, they use a linear combination of the pre-computed scratch BRDFs weighted by their corresponding area in a pixel footprint, enabling the evaluation of the footprint's BRDF. Users can control the scratches' profile, orientation, density and micro-BRDF. Their method is very efficient and energy conserving but limited to the representation of scratches.

Normal map based surfaces One of the most flexible representation for graphics artists is a high definition normal map. The first work which uses an explicit, arbitrary normal map to model the microsurface is the one of Yan et al. [21]. For a surface patch $\mathcal{P}$ associated to a ray sample on the surface – either referred to as patch or *ray footprint* later on – they evaluate the associated NDF, called the $\mathcal{P}$-NDF. To do so, they integrate a triangulated approximation of an interpolated normal map over the patch $\mathcal{P}$. This calculation is expensive because no closed form is available for the integral. The method of Yan et al. [22] improves the $\mathcal{P}$-NDF evaluation and is

$\boldsymbol{\omega}_1 \cdot \boldsymbol{\omega}_2$	dot product
$\langle \boldsymbol{\omega}_1 \cdot \boldsymbol{\omega}_2 \rangle$	dot product clamped to 0
$\chi^+(a)$	Heaviside function: 1 if $a > 0$ and 0 if $a \leq 0$
$\boldsymbol{\omega}_g$	geometric normal
$\boldsymbol{\omega}_o$	viewing/outgoing direction
$\boldsymbol{\omega}_i$	lighting/incoming direction
$\boldsymbol{\omega}_n$	element of the normal map
$\tilde{\mathbf{n}}$	normal slope
$\boldsymbol{\omega}_h$	half vector $\frac{\boldsymbol{\omega}_o+\boldsymbol{\omega}_i}{\|\|\boldsymbol{\omega}_o+\boldsymbol{\omega}_i\|\|}$
$\mathbf{u}$	surface position
$k_{\mathcal{P}}$	ray footprint
$D_{\mathcal{P}}$	patch NDF ($\mathcal{P}$-NDF)
$G_2(\mathcal{P})$	patch masking/shadowing function
$\Lambda_{\mathcal{P}}$	patch Λ Smith function
$A_{\mathcal{P}}$	patch projected area
D_i	i^{th} local Beckmann NDF
k_i	i^{th} 2D Gaussian weighting local NDF D_i
W_i	i^{th} weight: $\int_{\mathbb{R}^2} k_i(\mathbf{u}) k_{\mathcal{P}}(\mathbf{u}) \mathrm{d}\mathbf{u}$
Λ_i	i^{th} Λ Smith function
$\boldsymbol{\omega}_{\bar{n}_i}$	mean of the i^{th} NDF

Table 1: Important notations

about 100× faster. They approximate the normal map function by a mixture of millions of 4D Gaussians, inducing an analytic solution during patch integration. Their model is not physically based because approximations are made for the normalization factor, not taking into account the ray footprint like their NDF does. Furthermore, microfacet reflection models should be normalized by the projected area of the microsurface, which should itself also be derived from the NDF mean [8]. Their BRDF formulation only relies on the geometric normal, which further leads to inaccurate light scattering. They do not compare themselves to brute force, ground truth normal map rendering obtained through massive pixel sampling. Yan et al. have very recently [23] proposed a reflection model simulating iridescent height fields based on wave optics. It is mainly tailored to reproduce spectral effects, is computationally more expensive, and beyond the scope of our work..

Contrary to Yan et al. 2016 [22], our microsurface definition enables us to correctly take into account patch related data (projected area and masking term), and can be used in a microfacet formulation to both match reference, normal-mapped surface renderings as well as guarantee energy conservation.

3 Microsurface definition

Using a normal map to define the microsurface of a material is a simple and intuitive approach. By interpolating the normals, we obtain a continuous set that can be accurately approximated by a Gaussian-weighted NDF sum. Our surface representation is related but not completely similar to the one of Yan et al. [22]. The differences are important and ensure we have a well-defined model. We list the important mathematical notations used in the paper in Table 1.

Slope representation The normal map can be seen as a function giving a slope $\tilde{\mathbf{n}}(\mathbf{u}) \in \mathbb{R}^2$ for each surface coordinate $\mathbf{u}$. This representation affords to better fit into the microfacet theory [8]. To compute the slope $\tilde{\mathbf{n}} = (x_{\tilde{n}}, y_{\tilde{n}})^T$ associated to a normal (or direction) $\boldsymbol{\omega}_n = (x_n, y_n, z_n)^T$, we use the mathematical relations $x_{\tilde{n}} = -x_n/z_n$ and $y_{\tilde{n}} = -y_n/z_n$. All normals and directions are defined over the hemisphere centered around the geometric surface normal $\boldsymbol{\omega}_g = (0, 0, 1)^T$. We refer to this hemisphere as $\mathcal{H}^2$ onwards.

3.1 The mixture of Beckmann NDF

To evaluate a microfacet based BRDF, we need to know the density of a micronormal $\boldsymbol{\omega}_m$ at any surface point $\mathbf{u}$. Combining a pre-computed noncentered Beckmann NDFs $D_i(\boldsymbol{\omega}_m)$ yields an accurate approximation of the position normal distribution function $D_n(\boldsymbol{\omega}_m, \mathbf{u})$:

$$\begin{aligned} D_n(\boldsymbol{\omega}_m, \mathbf{u}) &= \delta(\boldsymbol{\omega}_n(\mathbf{u}) - \boldsymbol{\omega}_m) \\ &\approx \sum_i^a D_i(\boldsymbol{\omega}_m) k_i(\mathbf{u}). \end{aligned} \tag{1}$$

This sum is our definition of the microsurface where k_i is a Gaussian weight associated to the Beckmann NDF $D_i(\boldsymbol{\omega}_m)$. For a large enough, the position normal distribution function $D_n(\boldsymbol{\omega}_m, \mathbf{u})$ can be well approximated, especially if the Beckmann NDFs $D_i(\boldsymbol{\omega}_m)$ closely match local distributions from the normal map.

Gaussian weight definition The normal map is uniformly sampled by a seed points $\mathbf{u}_i$ spaced out by distance h between two adjacent samples. To have a smooth transition between the discrete samples, they are weighted by 2D Gaussians $k_i(\mathbf{u})$ centered around $\mathbf{u}_i$ (Figure 2, top). The standard deviations for $k_i(\mathbf{u})$ are defined so that the half of their peak value exactly lies at the midpoint between two adjacent seed points. They also integrate to the area they represent, e.g. $\int_{\mathbb{R}^2} k_i(\mathbf{u}) \mathrm{d}\mathbf{u} = h^2$.

3.2 Local NDF reconstruction

The Beckmann distribution is built using a 2D Gaussian of slope P_i^{22}

$$D_i(\boldsymbol{\omega}_m) = \frac{P_i^{22}(\tilde{\mathbf{m}})}{(\boldsymbol{\omega}_m \cdot \boldsymbol{\omega}_g)^4} \tag{2}$$

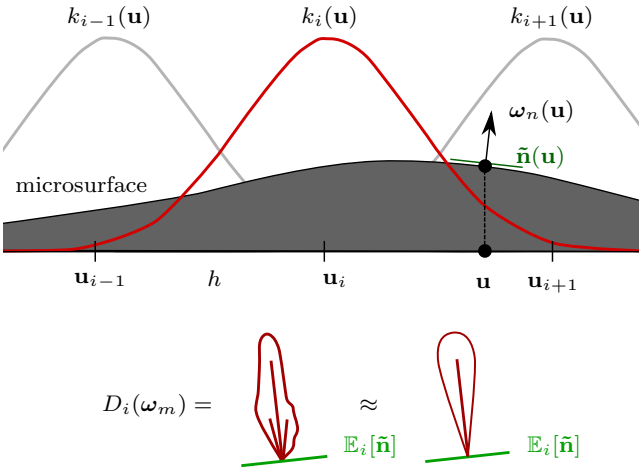

Fig. 2: Top: illustration of the 2D Gaussian k_i. Bottom: we approximate the NDF D_i associated to k_i using a noncentered Beckmann distribution.

where the denominator is the Jacobian of the normal to slope operation $1/(\boldsymbol{\omega}_m \cdot \boldsymbol{\omega}_g)^3$ multiplied by the inverse projection $1/(\boldsymbol{\omega}_m \cdot \boldsymbol{\omega}_g)$.

For a 2D Gaussian $k_i(\mathbf{u})$, the slope distribution function $P_i^{22}(\tilde{\mathbf{m}})$ is defined, for a microslope $\tilde{\mathbf{m}}$, as

$$P_i^{22}(\tilde{\mathbf{m}}) = \int_{\mathbb{R}^2} k_i(\mathbf{u})\delta(\tilde{\mathbf{n}}(\mathbf{u}) - \tilde{\mathbf{m}})\mathrm{d}\mathbf{u}. \tag{3}$$

This distribution can be approximated with a 2D Gaussian (Figure 2, bottom) by numerically computing the slope mean, standard deviation and correlation factor corresponding to the Gaussian weight $k_i(\mathbf{u})$:

$$P_i^{22}(\tilde{\mathbf{m}}) \approx \frac{\exp\left(-\frac{1}{2}(\tilde{\mathbf{m}} - \mathbb{E}_i[\tilde{\mathbf{n}}])^T \Sigma_i^{-1}(\tilde{\mathbf{m}} - \mathbb{E}_i[\tilde{\mathbf{n}}])\right)}{2\pi\sqrt{|\Sigma_i|}}. \tag{4}$$

The covariance matrix is $\Sigma_i = \begin{bmatrix} \sigma_{x,i}^2 & c_{xy,i} \\ c_{xy,i} & \sigma_{y,i}^2 \end{bmatrix}$ and its coefficients are

$$\begin{aligned} \sigma_{x,i}^2 &= \mathbb{E}_i[x_{\tilde{n}}^2] - \mathbb{E}_i^2[x_{\tilde{n}}], \\ \sigma_{y,i}^2 &= \mathbb{E}_i[y_{\tilde{n}}^2] - \mathbb{E}_i^2[y_{\tilde{n}}], \\ c_{xy,i}^2 &= \mathbb{E}_i[x_{\tilde{n}}y_{\tilde{n}}] - \mathbb{E}_i[x_{\tilde{n}}]\mathbb{E}_i[y_{\tilde{n}}], \end{aligned} \tag{5}$$

where the i^{th} expected values are computed with regards to the i^{th} 2D Gaussian k_i. For example, the mean slope $\mathbb{E}_i[\tilde{\mathbf{n}}]$ is

$$\mathbb{E}_i[\tilde{\mathbf{n}}] = \int_{\mathbb{R}^2} \tilde{\mathbf{n}}(\mathbf{u})k_i(\mathbf{u})\mathrm{d}\mathbf{u} \tag{6}$$

and definitions of the other expected values are similar.

The parameters of the NDFs are computed numerically using a Monte-Carlo integrator. Each 2D Gaussian k_i is sampled giving a set of positions $\mathbf{u}$. These positions are used to calculate $x_{\tilde{n}}$, $y_{\tilde{n}}$, $x_{\tilde{n}}^2$, $y_{\tilde{n}}^2$ and $x_{\tilde{n}}y_{\tilde{n}}$ and their respective averages $\mathbb{E}_i[x_{\tilde{n}}]$, $\mathbb{E}_i[y_{\tilde{n}}]$, $\mathbb{E}_i[x_{\tilde{n}}^2]$, $\mathbb{E}_i[y_{\tilde{n}}^2]$ and $\mathbb{E}_i[x_{\tilde{n}}y_{\tilde{n}}]$, i.e. the five i^{th} NDF parameters. When the surface is perfectly flat locally, the standard deviations of slopes $\sigma_{x,i}$ and $\sigma_{y,i}$ are 0, leading to a singularity. In this case, these terms are set to a small value $\sigma_\epsilon = 0.01$, implying that the microsurface has a negligible, tiny bit of roughness.

4 $\mathcal{P}$-BRDF

Rendering highly-specular subpixel normal-mapped details starts with determing the $\mathcal{P}$-NDF, i.e. the patch normal distribution function $D_\mathcal{P}$ (section 4.1). The $\mathcal{P}$-NDF can then be injected into a microfacet based BRDF $f_\mathcal{P}$ [19,8], along with its associated patch masking-shadowing term G_2 (section 4.2) and its patch projected area $A_\mathcal{P}$ (section 4.2.1). Its mathematical definition is

$$f_\mathcal{P}(\boldsymbol{\omega}_o, \boldsymbol{\omega}_i) = \frac{F(\boldsymbol{\omega}_o, \boldsymbol{\omega}_h)G_2(\mathcal{P}, \boldsymbol{\omega}_h, \boldsymbol{\omega}_o, \boldsymbol{\omega}_i)D_\mathcal{P}(\boldsymbol{\omega}_h)}{4A_\mathcal{P}(\boldsymbol{\omega}_o)A_\mathcal{P}(\boldsymbol{\omega}_i)} \tag{7}$$

where $\boldsymbol{\omega}_h$ is the half vector of reflection, F the Fresnel factor and $\mathcal{P}$ the ray footprint defined in the next section. Normally, the projected areas $A_\mathcal{P}(\boldsymbol{\omega}_o)$ and $A_\mathcal{P}(\boldsymbol{\omega}_i)$ are cosines of the polar angle of respectively the viewing and lighting direction, except when a normal map is used to enhance the surface [5], which is precisely our case.

4.1 $\mathcal{P}$-NDF

Determining all the outgoing radiance leaving a given ray footprint $\mathcal{P}$ over a normal map amounts to first evaluating the $\mathcal{P}$-NDF. Common methods for surface filtering give the orientation and size of a 2D Gaussian $k_\mathcal{P}$, using either ray differentials [9], path differentials [17] or covariance filtering [2]. This filter is defined over the uv-parameterization domain $\mathbf{u}$ of the normal map and is considered as the ray footprint $\mathcal{P}$. The 2D Gaussian $k_\mathcal{P}$ is normalized over $\mathbf{u}$, satisfying $\int_{\mathbb{R}^2} k_\mathcal{P}(\mathbf{u})\mathrm{d}\mathbf{u} = 1$.

Using the definition of the above ray footprint, we can now define the $\mathcal{P}$-NDF for an half vector $\boldsymbol{\omega}_h$ as

$$\begin{aligned} D_\mathcal{P}(\boldsymbol{\omega}_h) &= \int_{\mathbb{R}^2} D_n(\boldsymbol{\omega}_h, \boldsymbol{\omega}_n(\mathbf{u}))k_\mathcal{P}(\mathbf{u})\mathrm{d}\mathbf{u} \\ &= \int_{\mathbb{R}^2} \delta(\boldsymbol{\omega}_n(\mathbf{u}) - \boldsymbol{\omega}_h)k_\mathcal{P}(\mathbf{u})\mathrm{d}\mathbf{u}. \end{aligned} \tag{8}$$

Using our Beckmann NDF sum formulation (equation 1), we infer the more convenient expression of the $\mathcal{P}$-NDF

$$\begin{aligned} D_{\mathcal{P}}(\boldsymbol{\omega}_h) &\approx \int_{\mathbb{R}^2} \sum_i^a D_i(\boldsymbol{\omega}_h) k_i(\mathbf{u}) k_{\mathcal{P}}(\mathbf{u}) \mathrm{d}\mathbf{u} \\ &\approx \sum_i^a D_i(\boldsymbol{\omega}_h) \int_{\mathbb{R}^2} k_i(\mathbf{u}) k_{\mathcal{P}}(\mathbf{u}) \mathrm{d}\mathbf{u} \\ &\approx \sum_i^a D_i(\boldsymbol{\omega}_h) W_i \end{aligned} \tag{9}$$

where the NDF $D_i(\boldsymbol{\omega}_h)$ is moved out of the integral because it does not depend on the position $\mathbf{u}$. Each W_i term has an analytic solution as it is the integral of the product of two Gaussians. The sum of all weights is normalized, i.e. $\sum_{i=1}^a W_i = 1$.

4.2 $\mathcal{P}$-Smith term

A very important part of the microfacet theory is the masking-shadowing term associated with the NDF. This term serves two purposes. First, it has an impact on the final image quality (Figure 3). Second, the masking-shadowing term is the normalization factor of the $\mathcal{P}$-BRDF that guarantees energy conservation. In this section, we derive the exact $\mathcal{P}$-Smith term for our reflection model, making it well-defined.

Patch masking function The Smith microsurface profile [16,8] assumes that the masking function G_1 does not depend on micronormals $\boldsymbol{\omega}_m$ when they are not backfacing ($\boldsymbol{\omega}_o \cdot \boldsymbol{\omega}_m > 0$). This leads to a formulation implying the existence of both *local* and *distant* masking. The former is related to the micronormal $\boldsymbol{\omega}_m$ and is defined as $\chi^+(\boldsymbol{\omega}_o \cdot \boldsymbol{\omega}_m)$, while the latter just depends on the viewing direction $\boldsymbol{\omega}_o$. In our case, distant masking also depends on the patch $\mathcal{P}$ – we therefore denote it $G_1^{\text{dist}}(\mathcal{P}, \boldsymbol{\omega}_o)$. Our patch masking function is the combination of these two terms, where the micronormal is the half vector $\boldsymbol{\omega}_h$:

$$G_1(\mathcal{P}, \boldsymbol{\omega}_h, \boldsymbol{\omega}_o) = \chi^+(\boldsymbol{\omega}_h \cdot \boldsymbol{\omega}_o) G_1^{\text{dist}}(\mathcal{P}, \boldsymbol{\omega}_o). \tag{10}$$

Smith distant patch masking function The Smith distant patch masking function normalizes the BRDF, and brings in an integral over the hemispherical domain $\mathcal{H}^2$:

$$G_1^{\text{dist}}(\mathcal{P}, \boldsymbol{\omega}_o) = \frac{A_{\mathcal{P}}(\boldsymbol{\omega}_o)}{\int_{\mathcal{H}^2} \langle \boldsymbol{\omega}_o, \boldsymbol{\omega}_h \rangle D_{\mathcal{P}}(\boldsymbol{\omega}_h) \mathrm{d}\boldsymbol{\omega}_h}. \tag{11}$$

The patch projected area $A_{\mathcal{P}}(\boldsymbol{\omega}_o)$ is studied in section 4.2.1. In the denominator, we inject our $\mathcal{P}$-NDF formulation (equation 9) and move the weight W_i out of the integral, as it only depends on the surface position $\mathbf{u}$:

$$\begin{aligned} G_1^{\text{dist}}(\mathcal{P}, \boldsymbol{\omega}_o) &= \frac{A_{\mathcal{P}}(\boldsymbol{\omega}_o)}{\int_{\mathcal{H}^2} \langle \boldsymbol{\omega}_o, \boldsymbol{\omega}_h \rangle \sum_i^a D_i(\boldsymbol{\omega}_h) W_i \mathrm{d}\boldsymbol{\omega}_h} \\ &= \frac{A_{\mathcal{P}}(\boldsymbol{\omega}_o)}{\sum_i^a \int_{\mathcal{H}^2} \langle \boldsymbol{\omega}_o, \boldsymbol{\omega}_h \rangle D_i(\boldsymbol{\omega}_h) \mathrm{d}\boldsymbol{\omega}_h W_i} \end{aligned} \tag{12}$$

The integral over the hemisphere $\mathcal{H}^2$ can be expressed in terms of local projected area and Smith Λ_i function [8] corresponding to the i^{th} NDF D_i:

$$G_1^{\text{dist}}(\mathcal{P}, \boldsymbol{\omega}_o) = \frac{A_{\mathcal{P}}(\boldsymbol{\omega}_o)}{\sum_i^a \frac{\langle \boldsymbol{\omega}_{\bar{n}_i}, \boldsymbol{\omega}_o \rangle}{\boldsymbol{\omega}_{\bar{n}_i} \cdot \boldsymbol{\omega}_g} (1 + \Lambda_{\text{i}}(\boldsymbol{\omega}_o)) W_i} \tag{13}$$

where $\boldsymbol{\omega}_{\bar{n}_i}$ is the mean normal of D_i (computed from the mean slope $\mathbb{E}_i[\tilde{\mathbf{n}}]$). Because we formulate our NDF as a sum of noncentered, non axis-aligned and anisotropic Beckmann distributions, an analytic solution exists for $\Lambda_{\text{i}}(\boldsymbol{\omega}_o)$ [5].

Generalized form of the Smith masking function By developping the sum of equation 13 and dividing both the numerator and denominator by the patch projected area

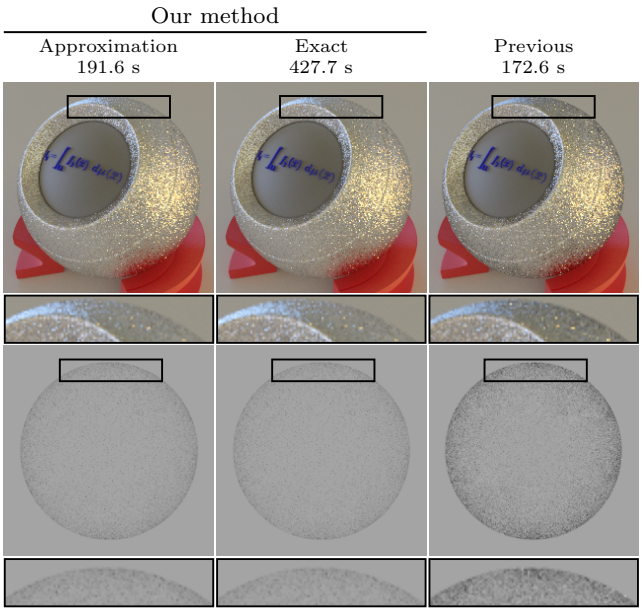

Fig. 3: Comparison of the masking-shadowing and projected area terms for the same $\mathcal{P}$-NDF. Left: approximated terms using our method. Center: exact terms. Right: Yan et al.'s terms. Top row: rough metallic orb illuminated by an environment map. Bottom row: white furnace tests [8]. We achieve better energy conservation than Yan et al. 2016 [22]. Their method overestimates the masking-shadowing as the normal map is extremely rough in this scene. Also notice the presence of bright pixels (i.e. brighter than the grey environment map) for their method, indicating the creation of energy.

$A_{\mathcal{P}}$ (defined in section 4.2.1), we obtain the generalized form of the Smith masking function:

$$G_1^{\text{dist}}(\mathcal{P}, \boldsymbol{\omega}_o) = \frac{1}{1 + \Lambda_{\mathcal{P}}(\boldsymbol{\omega}_o)} \tag{14}$$

where the patch $\Lambda_{\mathcal{P}}$ Smith term is

$$\Lambda_{\mathcal{P}}(\boldsymbol{\omega}_o) = \frac{\sum_i^a \frac{\langle \boldsymbol{\omega}_{\bar{n}_i}, \boldsymbol{\omega}_o \rangle}{\boldsymbol{\omega}_{\bar{n}_i} \cdot \boldsymbol{\omega}_g} \Lambda_i(\boldsymbol{\omega}_o) W_i}{A_{\mathcal{P}}(\boldsymbol{\omega}_o)}. \tag{15}$$

We can now inject equation 14 into equation 10 and use it to define the exact patch masking-shadowing function

$$G_2(\mathcal{P}, \boldsymbol{\omega}_h, \boldsymbol{\omega}_o, \boldsymbol{\omega}_i) = G_1(\mathcal{P}, \boldsymbol{\omega}_h, \boldsymbol{\omega}_o) G_1(\mathcal{P}, \boldsymbol{\omega}_h, \boldsymbol{\omega}_i). \tag{16}$$

This term is used to evaluate the $\mathcal{P}$-BRDF (equation 7). At this point, we still have to define the patch projected area $A_{\mathcal{P}}$.

4.2.1 $\mathcal{P}$-projected area

The masking function is an important part of the microfacet theory. To be well-defined, it needs to use the accurate projected area $A_{\mathcal{P}}$ of the surface. Note that $A_{\mathcal{P}}$ is used in both equations 7 and 11.

Following our $\mathcal{P}$-NDF formulation, the patch projected area $A_{\mathcal{P}}(\boldsymbol{\omega}_o)$ is the sum of the i^{th} projected area $\frac{\langle \boldsymbol{\omega}_{\bar{n}_i}, \boldsymbol{\omega}_o \rangle}{\boldsymbol{\omega}_{\bar{n}_i} \cdot \boldsymbol{\omega}_g}$ corresponding to each Beckmann NDF D_i, weighted by W_i:

$$A_{\mathcal{P}}(\boldsymbol{\omega}_o) = \sum_{i=1}^{a} \frac{\langle \boldsymbol{\omega}_{\bar{n}_i}, \boldsymbol{\omega}_o \rangle}{\boldsymbol{\omega}_{\bar{n}_i} \cdot \boldsymbol{\omega}_g} W_i. \tag{17}$$

This term replaces the more usual cosine of the viewing direction polar angle (used in Yan et al. [22]) which should not be used in our BRDF as the surface is enhanced by a normal map [5].

4.2.2 Approximation

For performance purposes and when energy conservation is not a critical need, we propose to use an approximated patch masking-shadowing function (equation 16) and projected area $A_{\mathcal{P}}$ (equation 17).

Similarly to Dupuy et al. [5], we use mipmapped filtering for the mean, standard deviation and correlation factor to compute an average Beckmann NDF relative to the patch $\mathcal{P}$. The associated Smith term of the filtered, single lobe NDF is then used to evaluate a good approximation of G_2 (Figure 3). The mipmap data structure gives a good estimate of the mean slope of a ray footprint $\mathcal{P}$ from which a projected area $A_{\mathcal{P}}$ can be derived. The Gaussian ray footprint $k_{\mathcal{P}}$ determines which levels of the mipmap pyramid [7] should be used for NDF lookups. Mipmap texels are weighted by W_i to maintain coherence with the reference mixture of NDF. Each local Gaussian k_i is fitted for each level of the mipmap pyramid accordingly (Figure 4).

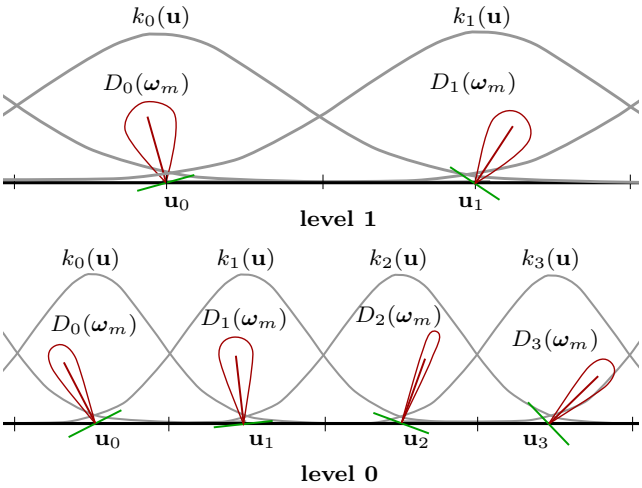

Fig. 4: NDF mipmap. The local Gaussians k_i are sized using their corresponding normal map texel area. Local NDFs D_i (red) and their mean slopes $\mathbb{E}_i[\tilde{\mathbf{n}}]$ (green) are averaged to construct the upper level. The former term is used for the masking-shadowing function and the latter for the projected area approximation.

5 Results and discussion

Our $\mathcal{P}$-BRDF is implemented in the physically based renderer PBRT-v3 [14]. Importance sampling and NDF evaluation is performed as in Yan et al. [22]. Importance sampling our reflection model amounts to sampling the reference $\mathcal{P}$-NDF $D_{\mathcal{P}}$. To do it, the ray footprint $k_{\mathcal{P}}$ is sampled giving a position $\mathbf{u}$. The associated normal $\boldsymbol{\omega}_n(\mathbf{u})$ is then used to reflect the viewing direction, giving the reflected direction. The associated PDF is $D_{\mathcal{P}}(\boldsymbol{\omega}_n)(\boldsymbol{\omega}_n \cdot \boldsymbol{\omega}_g)$ multiplied by the Jacobian of the reflect operator. The evaluation of $D_{\mathcal{P}}(\boldsymbol{\omega}_h)$ is accelerated through a hierarchy of NDF bounding boxes in the $(\mathbf{u}, \tilde{\mathbf{n}})$ space. For a given half vector $\boldsymbol{\omega}_h$ and ray footprint $k_{\mathcal{P}}$, only a small number of NDFs $D_i(\boldsymbol{\omega}_h)$ have a significant contribution in the $\mathcal{P}$-NDF sum of equation 9. We consider that the isotropic Gaussian filter k_i has nonzero values for positions $\mathbf{u}$ within the 2D bounding box $[\mathbf{u}_i - 3\boldsymbol{\sigma}; \mathbf{u}_i + 3\boldsymbol{\sigma}], \boldsymbol{\sigma} = (\sigma, \sigma)$ where σ is the standard deviation. For the NDF D_i, we consider that it returns nonzero values for normal slopes $\tilde{\mathbf{n}}$ within the

2D bounding box of the region containing 99% of the P_i^{22} probability mass.

5.1 $\mathcal{P}$-NDF

In this section, we evaluate our $\mathcal{P}$-NDF by comparing it to the ground truth and to the one of Yan et al. 2016 [22]. We also measure its runtime performance and memory occupation.

5.1.1 Accuracy

Our $\mathcal{P}$-NDF accurately approximates the ground truth. This is shown in Figure 5. Obviously, the number of seed points $\mathbf{u}_i$ needed to have a good approximation depends on the frequency content of the normal map. In all our results, we set $h = 1$, i.e. 1 NDF per normal map texel. Notice here that the $\mathcal{P}$-NDF of Yan et al. 2016 [22] is slightly more accurate than ours, as a 2D gaussian can be seen as a subset of a 4D gaussian with two fixed parameters, but the differences do not impact the renderings (Figure 6). A bullet-proof NDF match is not required to copy the appearance of the reference material (Section 5.2.2). The masking-shadowing term has much more impact on final renderings.

Evaluation times and memory consumption (Section 5.1.2) are comparatively higher for the method of Yan et al. [22].

5.1.2 Performance and memory cost

Yan et al. use a sum similar to equation 9 for the $\mathcal{P}$-NDF, with position-normal distributions in the form of a 4D Gaussian sum. To evaluate the NDF, they need to extract a 2D slice from their representation. No slicing is needed with ours, as we sum 2D Gaussians of slopes. This leads to better performance, as shown in Figures 5 and 6.

Our microsurface representation requires 5 scalar values per NDF ($\mathbb{E}_i[x_{\tilde{n}}]$,$\mathbb{E}_i[y_{\tilde{n}}]$, $\mathbb{E}_i[x_{\tilde{n}}^2]$, $\mathbb{E}_i[y_{\tilde{n}}^2]$ and $\mathbb{E}_i[x_{\tilde{n}}y_{\tilde{n}}]$), compared to 12 for the 4D Gaussian of Yan et al. 2016 [22] (they are represented by a 4×4 symmetric-positive matrix: the upper or lower triangular matrix need to be stored, plus 2 floats for their normal means).

5.2 Renderings

Renderings are performed on a 16-thread computer with two Intel Xeon E5-2650 v2 processors (8 cores). We use 2048×2048 mipmapped textures (Figure 7) and one NDF per normal map texel, requiring 112Mib of memory for the Gaussians and 68Mib for the acceleration hierarchy.

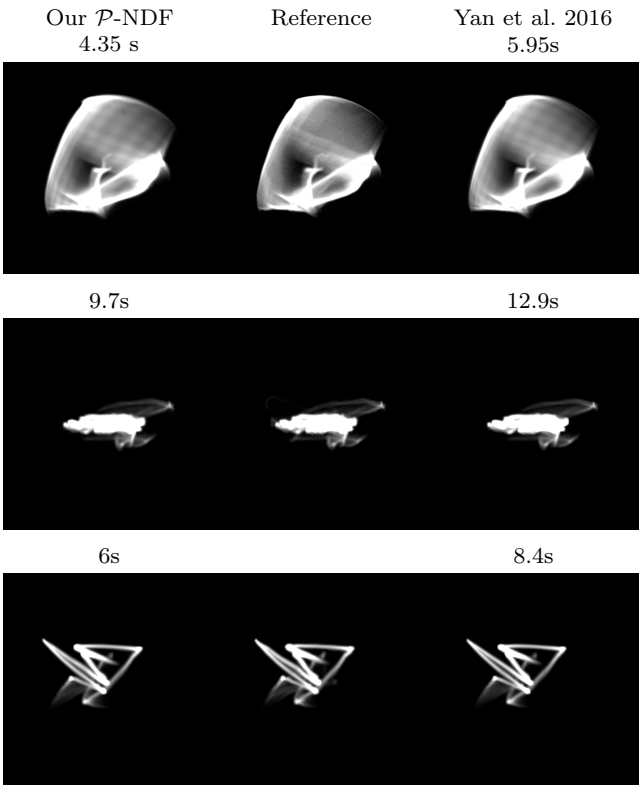

Fig. 5: Comparison of our $\mathcal{P}$-NDF, its interpolated normal map counterpart (reference) built using normal map binning and the $\mathcal{P}$-NDF of Yan et al. [22]. Construction times for the plots are included.

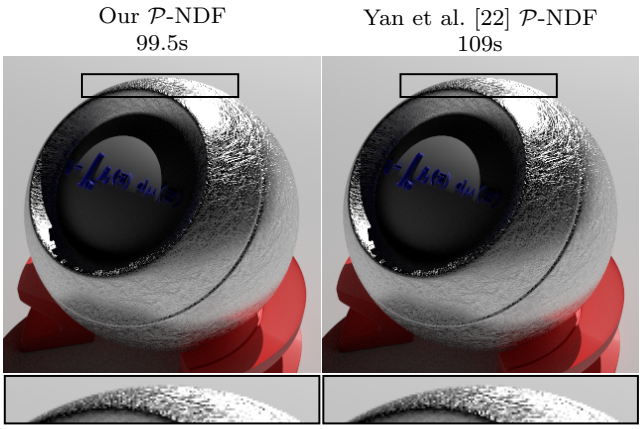

Fig. 6: Scratched metallic orb renderings using either our $\mathcal{P}$-NDF (left) or the one of Yan et al. 2016 [22] (right) using distance $h = 1$ between normal map samples. The other parameters of the reflection model are exactly the same. We achieve a very close match, demonstrating that our microsurface representation is accurate enough to reproduce the material appearance obtained with the method of Yan et al. 2016 [22].

For the same configuration, the method of Yan et al. 2016 [22] needs 192Mib for the Gaussians.

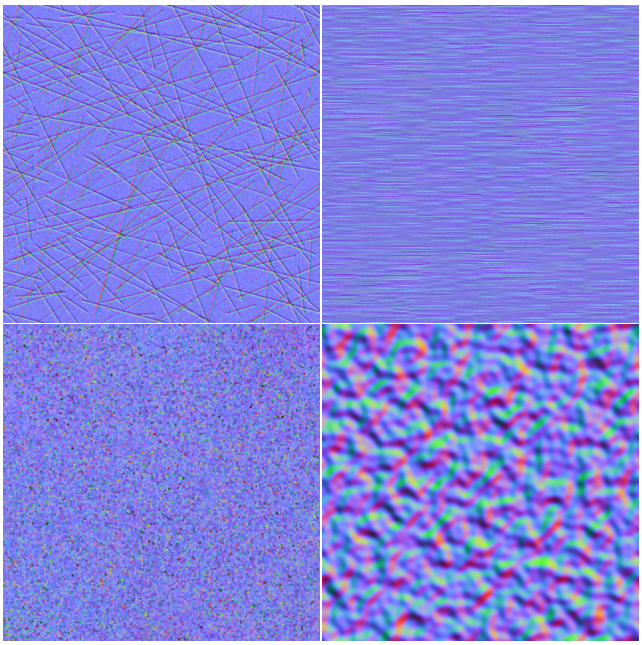

Fig. 7: Normal maps used for the results. Top left: scratched surface used in Figures 6, 10 and 11. Top right: brushed surface used in Figures 8 and 9. Bottom left: very rough surface used for Figures 8 and 9. We use a rougher version in Figure 3. Bottom right: rough surface used in Figure 1.

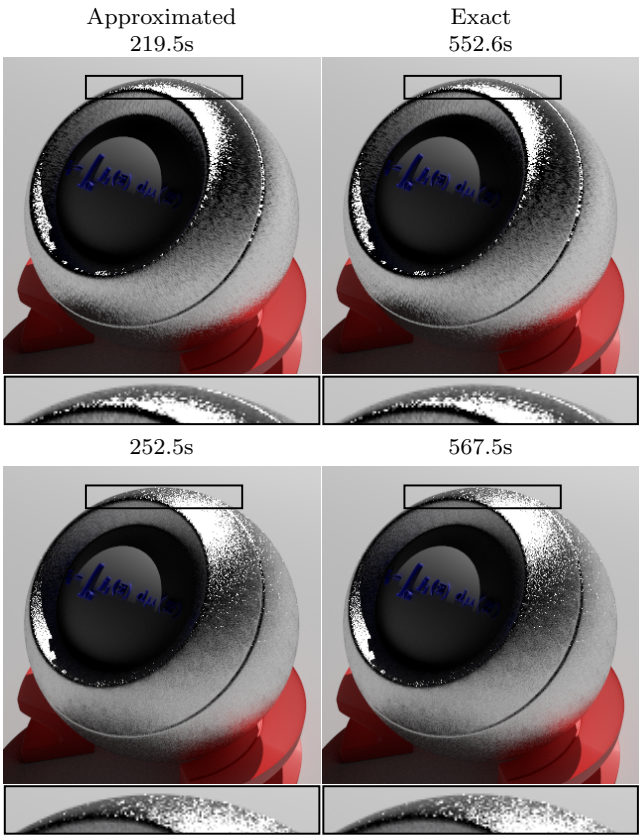

Fig. 8: Renderings obtained using the approximated (left) and exact (right) masking-shadowing function and projected area. The material is either anisotropic (top) or isotropic (bottom). The approximation makes the edges of the orb slightly darker but more than halves the rendering times.

5.2.1 Masking-shadowing and projected area

We first demonstrate the ability of our mipmap-based, approximated masking-shadowing function and projected area to match renderings using the exact terms, those for which the definition is respectively given in equations 16 and 17.

Figures 3 and 8 show that we almost have the same appearance using the exact or approximated masking-shadowing term. There are differences at grazing angles, where the masking function plays a predominant role. The approximation tends to overestimate masking because averaging Beckmann NDFs causes roughness to increase in the ray footprint. Rendering times are of course much lower when approximations are used. In Figure 3, for the same $\mathcal{P}$-NDF, we see that the method of Yan et al. 2016 [22] may create or loose energy locally because of its inaccurate, footprint-independent normalization factor.

5.2.2 Comparison with ground truth

To validate our method, we compare ourselves with naive renderings of the interpolated normal map, our reference material used as input data to build the NDF mixture. We compare the results obtained with renderings performed with the method of Yan et al. 2016 [22].

A classic normal map rendering requires at least 65536 samples per pixel to capture the majority of the glints in our scenes (except for Figure 11). The material is implemented as a near-perfect specular surface. We use a microfacet based BRDF with near-zero roughness $\sigma_\epsilon = 0.01$, centered around the normal at the location on the surface hit by the path tracer. The same roughness parameter is used to build the NDF mixture using our method (Section 3) and the one of Yan et al. [22]. Figure 9 demonstrates our ability to model brushed metal and metallic paint more faithfully to the ground

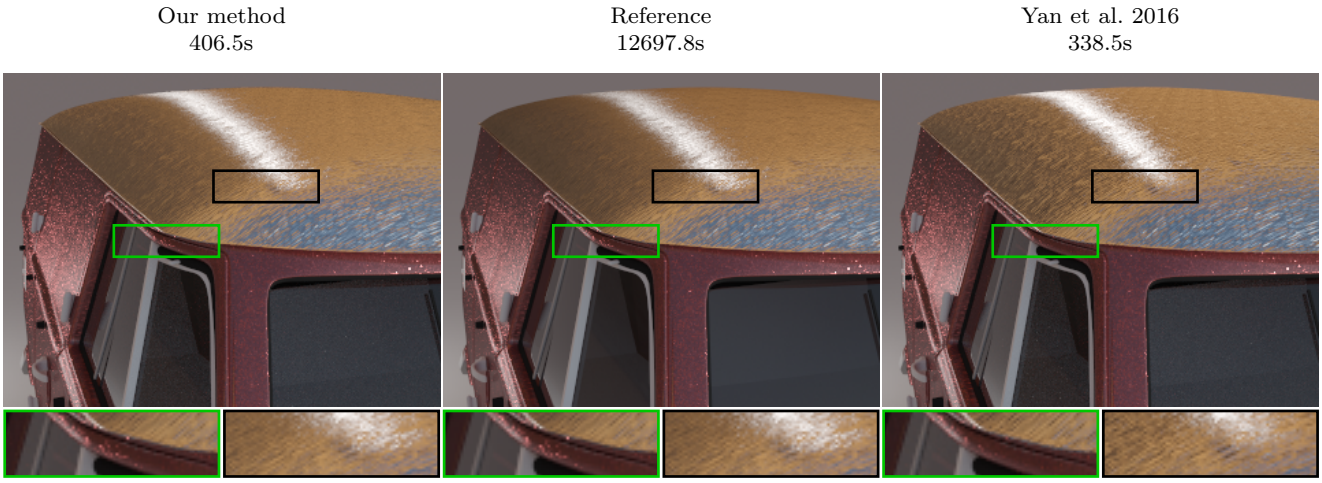

Fig. 9: A van with a metallic paint and a brushed metallic roof, using our approximated method (left, 1024 spp), classic normal map (middle, 65536 spp) and the method of Yan et al. 2016 [22] (right, 1024 spp) to model the materials. The van is illuminated by a environment map and point lights. Some glints are not rendered with the method of Yan et al. as shown in the green insets. In the black insets, the Yan et al. specular highlight shape differs from the reference while our not.

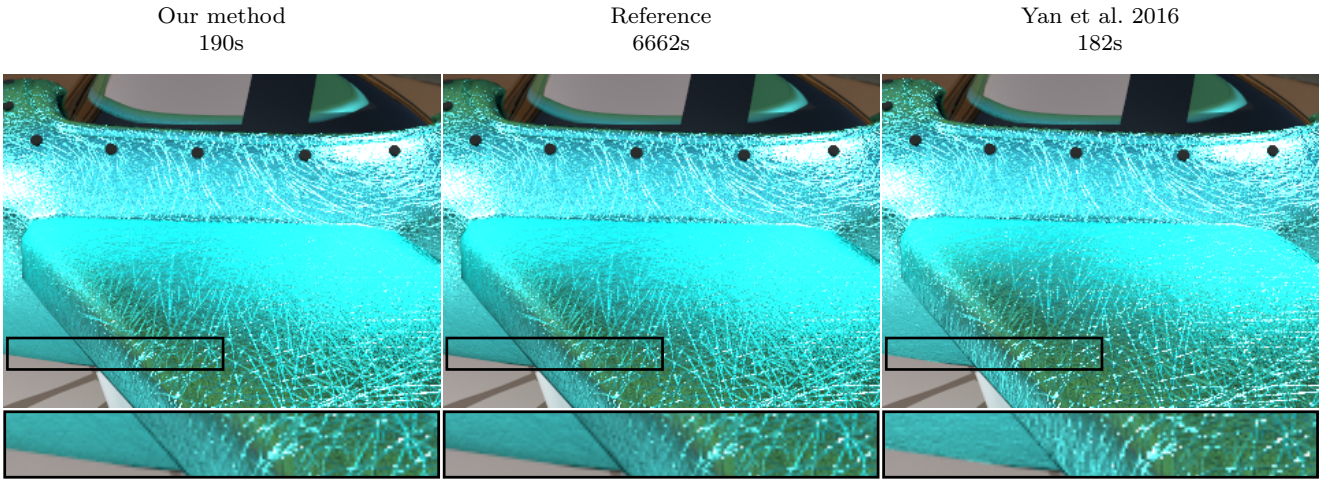

Fig. 10: Renderings of a very rough scratched metallic toy plane, obtained with our method using 1024 spp (left), an interpolated normal map using 65536 spp (middle) and the method of Yan et al. [22] using 1024 spp (right). With the method of Yan et al. 2016 [22], many scratches reflect light while the ground truth does not and the other way around (see inset). Our BRDF reproduces the normal map's appearance correctly, at the price of a mere 5% additional calculation time in this scene.

truth than the method of Yan et al. 2016 [22]. This is also the case for scratched metallic objects, as shown in Figure 10 and Figure 11 (a million of pixel samples are used for the reference in the latter case). Temporal versions of Figure 9 and Figure 10 can be seen in our video results.

Subtle differences between renderings can be observed when comparing our method with the reference – for example, at grazing angles, naive sampling exhibits darker shading than our method. These divergences are due to the combination of three factors. First, just like the method of Yan et al. [22], we rely on an approximation of the normal map and the ray footprint. The real footprint may have a different shape than our planar elliptical estimate, especially where there is high local surface curvature. Second, naive sampling cannot capture all glints because their size is infinitesimal compared to the size of a pixel. For example, the pixel footprint at the orb's silhouettes of Figure 11 is about 320000^2 texels, which would require $2^{19} \times 2^{19}$spp to have at least

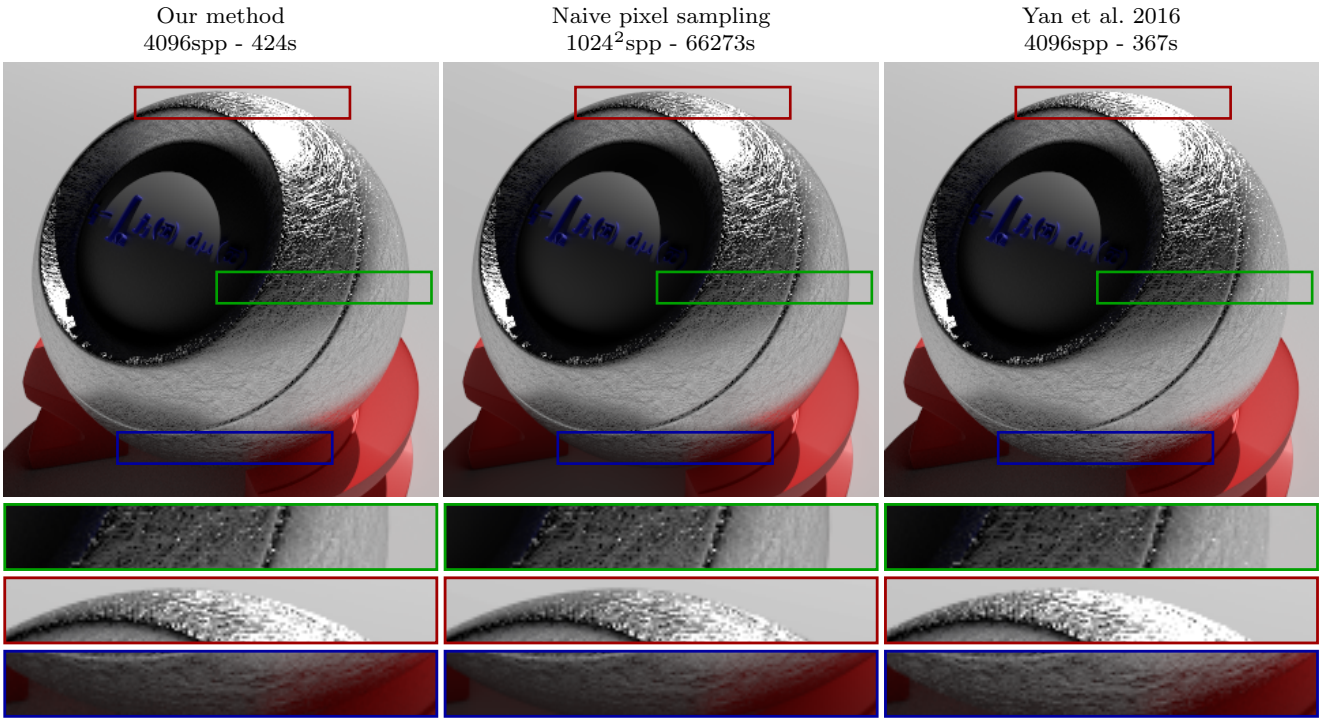

Fig. 11: A scratched, metallic orb illuminated by a point light. Our method (left) better captures the appearance of the reference material (center) than the method of Yan et al. [22] (right).

1 sample per texel. In practice, we cannot use as many samples, resulting in darker borders for the naive pixel sampling method. Finally, our masking-shadowing approximation tends to darken the shading. The larger the ray footprint, the more NDFs are mipmapped together, producing a single NDF with large standard deviation, incurring strong masking-shadowing. These behaviors can be observed in most renderings.

6 Conclusion and future works

In this paper, we proposed a physically based reflection model where the microsurface is defined with a mixture of Beckmann NDFs. These NDFs accurately approximate a high definition, specular normal map. Our representation can be reliably integrated into a microfacet based BRDF, where the masking shadowing term guarantees energy conservation and has a closed-form expression. We proposed an accurate approximation for the latter enabling practical, efficient use of our BRDF in readily available physically based renderers. We achieve the rendering of a wide range of materials like scratched and brushed surfaces, very rough metal but also metallic paint, where the surface produces a lot of glints under intense and sharp lighting. We compared ourselves with renderings based on naive sampling of reference, interpolated normal maps and achieve better matching than the method of Yan et al. 2016 [22].

The mixture of NDFs used to model the microsurface could be improved. In particular, the formulation could be more adaptive to local surface curvature, saving up memory and providing faster evaluation. Handling multiple scattering within the microstructure would bring even more realistic results with no loss of energy, which is inherent to formulations using shadowing.

7 Compliance with Ethical Standards

Conflict of Interest: The authors declare that they have no conflict of interest.

References

1. Atanasov, A., Koylazov, V.: A practical stochastic algorithm for rendering mirror-like flakes. In: ACM SIGGRAPH 2016 Talks, SIGGRAPH '16, pp. 67:1–67:2. ACM, New York, NY, USA (2016)
2. Belcour, L., Yan, L.Q., Ramamoorthi, R., Nowrouzezahrai, D.: Antialiasing complex global illumination effects in path-space. ACM Trans. Graph. **36**(1), 9:1–9:13 (2017)
3. Bosch, C., Patow, G.: Real-time path-based surface detail. Computers and Graphics **34**(4), 430 – 440 (2010). Procedural Methods in Computer Graphics Illustrative Visualization

4. Bosch, C., Pueyo, X., Mérillou, S., Ghazanfarpour, D.: A physically based model for rendering realistic scratches. Computer Graphics Forum **23**(3), 361–370 (2004)
5. Dupuy, J., Heitz, E., Iehl, J.C., Poulin, P., Neyret, F., Ostromoukhov, V.: Linear Efficient Antialiased Displacement and Reflectance Mapping. ACM Trans. Graph. **32**(6), 211:1–211:11 (2013)
6. Han, C., Sun, B., Ramamoorthi, R., Grinspun, E.: Frequency domain normal map filtering. ACM Trans. Graph. **26**(3) (2007)
7. Heckbert, P.S.: Fundamentals of texture mapping and image warping. Master's thesis (1989)
8. Heitz, E.: Understanding the Masking-Shadowing Function in Microfacet-Based BRDFs. Journal of Computer Graphics Techniques (JCGT) **3**(2), 48–107 (2014)
9. Igehy, H.: Tracing ray differentials. In: Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '99, pp. 179–186. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA (1999)
10. Jakob, W., Hašan, M., Yan, L.Q., Lawrence, J., Ramamoorthi, R., Marschner, S.: Discrete stochastic microfacet models. ACM Trans. Graph. **33**(4), 115:1–115:10 (2014)
11. Lewis, R.R.: Making shaders more physically plausible. Computer Graphics Forum **13**(2), 109–120 (1994)
12. Merillou, S., Dischler, J., Ghazanfarpour, D.: Surface scratches: measuring, modeling and rendering. The Visual Computer **17**(1), 30–45 (2001)
13. Olano, M., Baker, D.: Lean mapping. In: Proceedings of the 2010 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games, I3D '10, pp. 181–188. ACM, New York, NY, USA (2010)
14. Pharr, M., Jakob, W., Humphreys, G.: Physically Based Rendering: From Theory to Implementation, 3rd edn. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2016)
15. Raymond, B., Guennebaud, G., Barla, P.: Multi-scale Rendering of Scratched Materials Using a Structured SV-BRDF Model. ACM Trans. Graph. **35**(4), 57:1–57:11 (2016)
16. Smith, B.: Geometrical shadowing of a random rough surface. IEEE Transactions on Antennas and Propagation **15**(5), 668–671 (1967)
17. Suykens, F., Willems, Y.D.: Path differentials and applications. In: S.J. Gortler, K. Myszkowski (eds.) Rendering Techniques 2001, pp. 257–268. Springer Vienna, Vienna (2001)
18. Toksvig, M.: Mipmapping normal maps. Journal of Graphics Tools **10**(3), 65–71 (2005)
19. Walter, B., Marschner, S.R., Li, H., Torrance, K.E.: Microfacet models for refraction through rough surfaces. In: Proceedings of the 18th Eurographics Conference on Rendering Techniques, EGSR'07, pp. 195–206. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland (2007)
20. Xu, C., Wang, R., Zhao, S., Bao, H.: Real-Time Linear BRDF MIP-Mapping. Computer Graphics Forum **36**(4), 27–34 (2017)
21. Yan, L.Q., Hašan, M., Jakob, W., Lawrence, J., Marschner, S., Ramamoorthi, R.: Rendering glints on high-resolution normal-mapped specular surfaces. ACM Trans. Graph. **33**(4), 116:1–116:9 (2014)
22. Yan, L.Q., Hašan, M., Marschner, S., Ramamoorthi, R.: Position-normal distributions for efficient rendering of specular microstructure. ACM Trans. Graph. **35**(4), 56:1–56:9 (2016)
23. Yan, L.Q., Hašan, M., Walter, B., Marschner, S., Ramamoorthi, R.: Rendering specular microgeometry with wave optics. ACM Trans. Graph. **37**(4), 75:1–75:10 (2018)