EUROGRAPHICS 2018 / D. Gutierrez and A. Sheffer
(Guest Editors)

Volume 37 (2018), Number 2

Real-Time Rendering of Wave-Optical Effects on Scratched Surfaces

Z. Velinov*, S. Werner* and M. B. Hullin

Rheinische Friedrich-Wilhelms-Universität Bonn, Germany
*Joint first authors

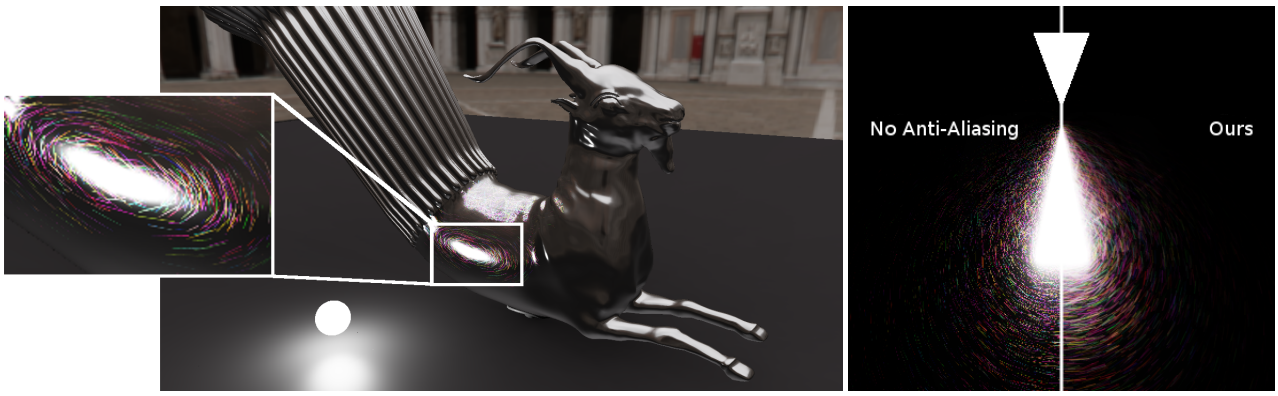

Figure 1: *We introduce a diffraction model for fine surface scratches which seamlessly integrates in modern real-time rendering pipelines. Left: Our approach reaches 30fps at 720p resolution on a notebook GPU (GTX 970M) with 10000 scratches and two light sources: a sphere and a rectangle approximating the highest intensity parts of the environment. Right: Our technique provides anti-aliased scratches for multi-scale rendering without multi-sampling. Shown here is the reflection of a triangular light source on a planar target.*

Abstract

The visual appearance of real-world materials is characterized by surface features across many scales and has received significant attention by the graphics community for decades. Yet, even the most advanced microfacet models have difficulties faithfully recreating materials like snow, sand, brushed metal or hair that feature scale-violating glints and speckles and defy any traditional notion of filtering and level of detail. In this work, we address an important subset of such materials, namely metal and dielectric surfaces that are covered with microscopic scratches, e.g., from polishing processes or surface wear. The appearance of such surfaces features fine-scale spatial detail and iridescent colors caused by diffraction, and has only recently been successfully recreated. We adopt the scratch iridescence model, which is known for plausible results in offline Monte Carlo settings but unsuitable for real-time applications where extensive illumination sampling is prohibitively expensive. In this paper, we introduce an efficient technique for incoherently integrating the contributions of individual scratches, as well as closed-form solutions for modeling spherical and polygonal area light sources, and for the first time bring scratch iridescence within reach of real-time applications.

1. Introduction

Real-world surfaces feature geometric detail across a wide range of scales, which results in a characteristic visual appearance that computer graphics has long attempted to faithfully model, capture and reproduce. The state of the art in digital material appearance is dominated by models that represent surface microgeometry by statistics like micro-facet or micro-flake distributions. This class of models is capable of closely approximating the far-field scattering properties of most real-world surfaces [GGG*16], and it has recently even been extended to include wave-optical effects [HP17, BB17]. Yet, significant effort is needed to enable the recreation of fine-scale reflectance detail such as isolated glints [YHJ*14, YHMR16, JHY*14]. In recent work, Werner et al. [WVJH17] introduced a spatially varying bidirectional re-

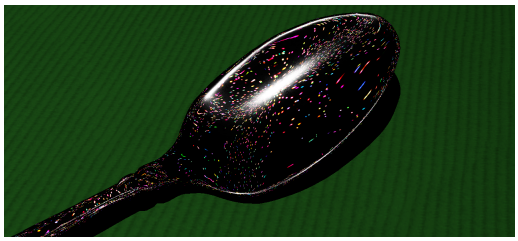

Figure 2: *Scratch iridescence model by Werner et al., rendered under point light source (1 light sample per pixel).*

flectance distribution function (SVBRDF) for a sub-class of microstructured materials, namely metallic and dielectric surfaces with fine enough scratches to produce iridescent reflections. Their method, to our knowledge, is the first to bridge the gap between non-paraxial wave-optical scattering theory and texture required for spatial detail in computer graphics applications. Neglected in prior wave-optical scattering models, their model correctly handles spatial variation by combining coherence theory with a vector-graphics representation for discrete scratches on a surface. In this paper, we address what we consider the most important drawback of the proposed technique: its need for extensive sampling of area light sources and pixel footprints. While the model can be evaluated with a single light sample for point sources, the resulting appearance is highly unrealistic (Fig. 2). Multi-sampling, on the other hand, produces well-filtered renderings in offline settings but is far too costly for real-time applications. Where precomputed radiance transfer delivers high-quality results for ordinary reflectance distributions under environment lighting [SKS02], highly specular glints as well as micrometer-level spatial detail prohibit the use of such approaches for the class of materials at hand. To mitigate this problem, we propose a set of analytical pre-integrations and approximations that build upon the model of Werner et al. and enable it to achieve real-time performance under area and environment lighting. In particular, our technical contributions include

- a closed-form solution of the integral over the pixel footprint visible from the camera,
- closed-form solutions for integrals over polygonal and spherical area light sources, and
- an optimized pipeline and data structure to enable efficient lookup of discrete scratch particles on the GPU.

With these improvements, our model achieves a speed-up of at least three orders of magnitude, enabling the rendering of scratch iridescence at HD resolution in real time. Our work is a promising step towards introducing these effects in current real-time rendering pipelines, and it is immediately applicable for visualization purposes in offline rendering production.

2. Related work

Our proposed method builds upon Werner et al.'s theory and model for wave-optical effects on worn surfaces [WVJH17], which heavily relies on Monte Carlo integration for the evaluation of the reflectance function in both spatial and angular domain. Multiple samples are cast on the footprint of each pixel in order to mitigate aliasing effects in the spatial domain; in angular domain, area and environment lights are evaluated through costly importance sampling. Here, our goal is to alleviate both these requirements. By deriving analytical pre-filtered versions of the reflectance function, we reduce the effort of evaluating the model from thousands of samples to one sample per pixel, effectively making it capable of real-time operation. In the following, we briefly review the state of the art in wave-optical shading models, and provide an overview of real-time rendering techniques for extended light sources. For a historic study of general BRDF modeling, we refer the interested reader to the survey by Guarnera et al. [GGG*16]. Further information about prefiltering different components of microfacet BRDFs can be found in the survey by Bruneton and Neyret [BN12].

2.1. Wave-optical shading models

A conclusive overview of wave-optical scattering models for different surface classes can be found in [Kry06], and fundamental principles are discussed in [Goo96]. Based on such models of light-surface interaction, the computer graphics community has developed a range of techniques accounting for diffraction and interference in ray-based frameworks. Stam [Sta99], as one of the first, proposed a diffraction model for rough surfaces. The concept of spatial coherence, which is critical for spatially resolved structures, was used by Levin et al. [LGX*13] and Dhillon et al. [DTS*14]. In contrast to Levin, who predicted BRDFs of lithographically structured surfaces, Dhillon used this concept to efficiently simulate diffraction effects of measured biological real-world surfaces. More recently, Belcour and Barla [BB17] proposed a modified microfacet model to recreate interference from thin layers. Dong et al. [DWMG15] derived far-field BRDF models from Kirchhoff scattering and microfacet theory with support for tilted surfaces. Holzschuch and Pacanowski [HP17] proposed a two-scale microfacet model based on generalized Harvey-Shack theory to combine reflection and diffraction. Toisoul et al. [TG17] introduced a model suitable for simulating the iridescent behavior of surfaces with periodic structure for small angles. Their technique is based on a tabulated pre-convolution lookup of the reflected radiance by periodic diffraction gratings. However, the memory requirements prohibit the use of their approach for real-time rendering of worn surfaces, which are composed of many scratches with randomized parameters. This approach relates to many previously studied techniques for precomputation described in the survey by Ramamoorthi [Ram09], and shares the same limitations.

2.2. Rendering scratch-like features

Rendering scratches and line-like features such as fibers [dMH13] or hair [dFH*11, YJR17] are closely related due to the common geometrical properties. Recent models for reflection from worn surfaces are limited to geometrical optics and represent scratches using scratch profiles made of aligned surfaces with associated procedural BRDFs, their positions described by curves on the surface [BPMG04]. More recently, Raymond et al. [RGB16] proposed a multi-scale SVBRDF model from a stack of scratch layers. Yan et al. [YHMR16] utilize optimized integration of the normal distribution function on normal-mapped surfaces to recreate glint-like reflectance behavior.

2.3. Extended light sources

The appearance of materials strongly depends on the distribution of light surrounding them. The graphics community therefore has invested major efforts to develope efficient approximate methods for shading with extended light sources. In opposition to standard brute-force path tracing where luminaires are sampled, many techniques have been studied to efficiently sample area sources of different geometry [SWZ96]; in particular spherical triangles [Arv95, 5.2], spherical rectangles [UnFK13], and spherical ellipses [GUnK*17]. When applied to strongly specular material models with high angular bandwidth, these integration strategies require a high number of samples for converged, noise-free images, making them unsuitable for most real-time rendering purposes.

In concurrent work, researchers have attempted to reduce area lights to representative point sources [Pic92], causing a severe loss in quality unless sets of many virtual point lights (VPLs) are used [Kel97, DKH*14]. The number of VPLs needed to obtain high-quality renderings increases with the bandwidth of the material, so that for specular materials real-time evaluation becomes more and more challenging. Techniques that project the illumination into spherical basis functions for efficient precomputation [SKS02] suffer from the same problem: high-frequency illumination in combination with high-frequency reflectance requires impractically high orders of spherical harmonics to achieve output of sufficient quality.

A third strand of research aims to approximate area light sources using a single sample, ranging from the well-known computation of form factors in the radiosity method for diffuse surfaces [GTGB84, BRW89] to extensions of the representative point method that approximate the combined effect of specular lobe and area sources [Dro14, Kar13, dCI17]. In the context of our work, the most relevant class of methods are those that derive analytical solutions for the illumination integral over the solid angle subtended by area light sources. For glossy materials, for instance, the problem has been addressed by several methods based on polynomial expansions of the two-dimensional (2D) illumination integral [BP93], on multiple one-dimensional (1D) integrals [PA91, TT97], or a single contour integral using irradiance tensors for polygonal area sources [Arv95, 4] [Sny96, LDSM16]. More recently, Heitz et al. proposed to transform microfacet distributions into linearly transformed cosine distributions (LTCs) to approximate the reflection of polygonal [HDHN16] and linear [HH17] light sources. Finally, Dupuy et al. [DHB17] used spherical pivot transformed distributions (SPTDs) to efficiently approximate the integral over the solid angle subtended by spherical light sources for microfacet models.

The appearance model that serves as the basis for the proposed technique [WVJH17] specializes on microscopic iridescent scratches on worn surfaces, which gives rise to extremely detailed and specular reflections. Applied naïvely to real-time rendering, their model misses important visual properties such as elongated scratches, but instead produces point-like features (cf. Fig. 2). We aim to mitigate these drawbacks while using only a single sample per pixel to achieve real-time capability.

Parameter	Definition
$\mathbf{s}^{(m)}$	Position of a scratch m in world coordinates
$\mathbf{x}$	Shaded position in world coordinates
$[\mathbf{t},\mathbf{b},\mathbf{n}]$	Surface tangent space vectors
(ϕ_i,θ_i)	Azimuth and zenith incident light angle
ω_o	Outgoing light vector
ξ	Vector of direction cosines
$\xi^{(m)}$	Direction cosines w.r.t. the basis vector of scratch m
$\sigma = 60\,\mu\text{m}$	Coherence area radius
$A_C = \pi\sigma^2$	Coherence area
λ	Wavelength
k	Wave number
$\mathcal{B}(\xi)$	Base response function
$\mathcal{S}(\xi)$	Scratch response function
$\mathcal{S}^{(m)}(\xi)$	Response function of the m^{th} scratch
$W^{(m)},D^{(m)}$	Width and depth of the m^{th} scratch
$\eta^{(m)}$	Spatial phases of the m^{th} scratch
$A_\mathcal{P}$	Area of the projected camera pixel footprint
a_1, a_2	Extent of the projected footprint ellipse
$l^{(m)}_{\text{contained}}$	Scratch length contained within pixel footprint
l_1, l_2	Scratch boundaries
$\rho_{c,\mathcal{P}}$	Scratch area density
$\mathbf{V}^{(m)}$	Vertex m of polygonal light source
$q^{(m)}$	Intersection with projected light source

Table 1: *Overview of the parameters used throughout this paper*

3. Theory

In this work, we seek to extend the scratch iridescence model by Werner et al. [WVJH17] for use in real-time applications, and in particular for rendering well-filtered scratches under extended light sources with a single sample per pixel. To this end, we provide approximations and techniques that are required to efficiently evaluate core elements of the model. In particular, we consider the missing pieces necessary to achieve real-time capability to be (1) antialiasing and (2) lighting by area light sources, which need to be realized using only one sample per pixel. To better understand the properties of the original model, we will in the following discuss its main elements and how to exploit them for our purpose.

3.1. Model

In its original form, the scratch iridescence model is given by the SVBRDF

$$f_r(\xi,\mathbf{x}) = \frac{1}{\pi\sigma^2\lambda^2}\left|\mathcal{B}(\xi) - \mathcal{S}(\xi,\mathbf{x})\right|^2, \tag{1}$$

where ξ is the vector of direction cosines [Kry06] and $\mathbf{x}$ the shaded surface position. In essence, a base response function $\mathcal{B}$ and a scratch response function $\mathcal{S}$ model the respective contributions of the unscratched surface and the iridescent scratches. We base our approximation of the scratch response $\mathcal{S}$ on a vector graphics representation (discrete lines) and sum over the contributions $\mathcal{S}^{(m)}$ of individual scratch primitives:

$$\begin{aligned}\mathcal{S}(\xi,\mathbf{x}) &= \sum_m \mathcal{S}^{(m)}\left(\xi^{(m)},\mathbf{x}\right)\\ &= \sum_m \mathcal{W}^{(m)}\mathcal{D}^{(m)}\eta^{(m)}\left(\xi^{(m)},\mathbf{x}\right).\end{aligned} \tag{2}$$

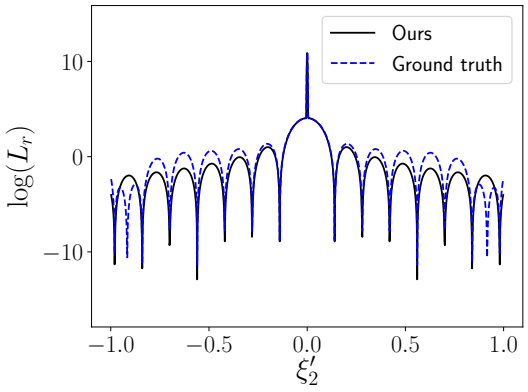

Figure 3: *Comparison of diffracted radiance for a single scratch with rectangular profile. The result obtained with our model (black line; no angle dependence, $\mathcal{D} \propto \exp(i4\pi D/\lambda)$) underestimates the diffraction of side lobes and neglects re-distribution of energy due to the angular dependence of the complex phase in the original model (blue dashed line; $\mathcal{D} \propto \exp(i2\pi(\gamma+\gamma_i)D/\lambda)$. However, the overall appearance is conserved as we closely preserve the envelope shape of the scattering function ($\lambda = 700\,nm$).*

Assuming a rectangular scratch cross-section, the individual factors for the m^{th} scratch are:

$$\mathcal{W}^{(m)}\left(\xi^{(m)}\right) = W^{(m)}\operatorname{sinc}\left(\pi W^{(m)}\xi_2/\lambda\right), \tag{3}$$

$$\mathcal{D}^{(m)} = 1 - e^{i4\pi D^{(m)}/\lambda}, \qquad \text{and} \tag{4}$$

$$\eta^{(m)}\left(\xi^{(m)}, \mathbf{x} | l_1, l_2\right) = d\,\operatorname{erf}\left(\left((s_t^{(m)} + h)/\sigma + i\sigma k\xi_1\right)/\sqrt{2}\right)\Big|_{h=l_1}^{l_2}. \tag{5}$$

$\mathcal{W}^{(m)}$ denotes the spectral term which, in turn, depends on the *scratch width* $W^{(m)}$. $\mathcal{D}^{(m)}$ denotes the *depth*-related term, and $\eta^{(m)}$ encodes the spatial phases with respect to $\mathbf{x}$. $\mathcal{D}^{(m)}$ is approximated for small angles to enable analytic integration for area light sources in the following. To evaluate this deviation from the original work by Werner et al., we provide a qualitative comparison of this approximation in Fig. 3 which shows that the overall shape of the scattering function is well preserved when only separate scratches are considered. This approximation is provided in the public implementation of the work [WVJH17] without being discussed in the main text.

We further provide the main elements of the model for completeness and easier implementation:

$$\xi^{(m)} = (\xi_1, \xi_2)^{(m)} = (\omega_i + \omega_o) \cdot (\mathbf{t}, \mathbf{b})^{(m)}$$

$$\mathbf{s}'^{(m)} = \mathbf{s}^{(m)} - \mathbf{x} = s_t^{(m)}\mathbf{t}^{(m)} + s_b^{(m)}\mathbf{b}^{(m)}$$

$$k = \frac{2\pi}{\lambda}$$

$$d = \sqrt{\pi/2}\,\sigma \exp\left(-\frac{1}{2\sigma^2}{s_b^{(m)}}^2 - \frac{\sigma^2}{2}k^2{\xi_1^{(m)}}^2 - iks_b^{(m)}\xi_2^{(m)}\right),$$

where $\xi^{(m)}$ is the vector of direction cosines, $(\mathbf{t}, \mathbf{b})^{(m)}$ the tangent and bitangent of a scratch, $\mathbf{s}'^{(m)}$ its position with respect to $\mathbf{x}$ which is further decomposed in tangential ($s_t^{(m)}$) and bitangential ($s_b^{(m)}$) components, $\sigma = 60\,\mu m$ the radius of the coherence area and $k = \frac{2\pi}{\lambda}$ the wavenumber with respect to wavelength λ.

The base material $\mathcal{B}$ can be any physically based material model under the assumption of negligible scratch-surface interference. More formally,

$$\begin{aligned} f_r(\xi, \mathbf{x}) &= \frac{1}{\pi\sigma^2\lambda^2}\left(|\mathcal{B}|^2 + |\mathcal{S}|^2 - 2\,\mathcal{B} * \mathcal{S}\right) \\ &\approx \frac{1}{A_{\mathcal{C}}\lambda^2}\left((1-\rho_c)|\mathcal{B}|^2 + |\mathcal{S}|^2\right), \end{aligned} \tag{6}$$

where the cross-term related to scratch-surface interaction is neglected and $A_{\mathcal{C}} = \pi\sigma^2$ is the coherence area covered by the spatial filter. To ensure energy conservation, a simple alpha blending term based on the *scratch area density* ρ_c is introduced. We focus on using microfacet materials to define the base contribution as they are prevalent in the current rendering systems and provide overall easier control over the appearance of the surface through intuitive parameters. For efficient rendering of microfacet surfaces lit by extended polygonal, tube or spherical light sources we refer to the recent works of Heitz et al. [HDHN16, HH17] and Dupuy et al. [DHB17]. We use these distributions fit to the GGX model and directly superimpose the scratches as illustrated by Eq. (6).

The solution for incoherent superposition additionally assumes that mutual interference of scratch ensembles, expressed by cross-terms appearing in $|\mathcal{S}(\xi)|^2$, can be neglected such that

$$|\mathcal{S}(\xi)|^2 = \left|\sum_m \mathcal{S}^{(m)}\left(\xi^{(m)}\right)\right|^2 \approx \sum_m \left|\mathcal{S}^{(m)}\left(\xi^{(m)}\right)\right|^2. \tag{7}$$

All previously expressed formulations within this section follow directly from the work by Werner et al. and its publicly available implementation. Based on them, we are now able to separately compute the radiance scattered by each single scratch and to derive more efficient formulas for certain individual parts.

It is generally assumed that the scratches fall completely within the coherence area. Consequently $\eta^{(m)}$ can be simplified by taking the limit of an infinite scratch:

$$\eta^{(m)} = \sqrt{2\pi}\sigma \exp\left(-\frac{1}{2\sigma^2}{s_b^{(m)}}^2 - \frac{\sigma^2}{2}k^2\xi_1^2 - iks_b^{(m)}\xi_2\right). \tag{8}$$

The complex part of the exponential is completely canceled when we take the absolute value squared, leading to a very compact form of the term for the reflectance along the scratch:

$$|\eta^{(m)}|^2 = 2A_{\mathcal{C}} \exp\left(-\frac{1}{\sigma^2}{s_b^{(m)}}^2 - \sigma^2k^2\xi_1^2\right), \tag{9}$$

which separates into an angular and spatial part, respectively

$$\begin{aligned} \left|\eta_a^{(m)}\right|^2 &= \exp\left(-\sigma^2k^2\xi_1^2\right); \\ \left|\eta_s^{(m)}\right|^2 &= 2A_{\mathcal{C}} \exp\left(-\frac{{s_b^{(m)}}^2}{\sigma^2}\right). \end{aligned} \tag{10}$$

The separability enables us to derive the integral over a pixel footprint required for correct antialiasing in spatial domain and the integral over solid angle subtended by area light sources. Evaluation of the absolute value squared of $\mathcal{D}^{(m)}$ additionally yields

$$\left|\mathcal{D}^{(m)}\right|^2 = \left(2 - 2cos(4\pi D^{(m)}/\lambda)\right), \tag{11}$$

which is a well known result that can be traced back to the derivation of the equation proposed by Stam [Sta99] for modeling the

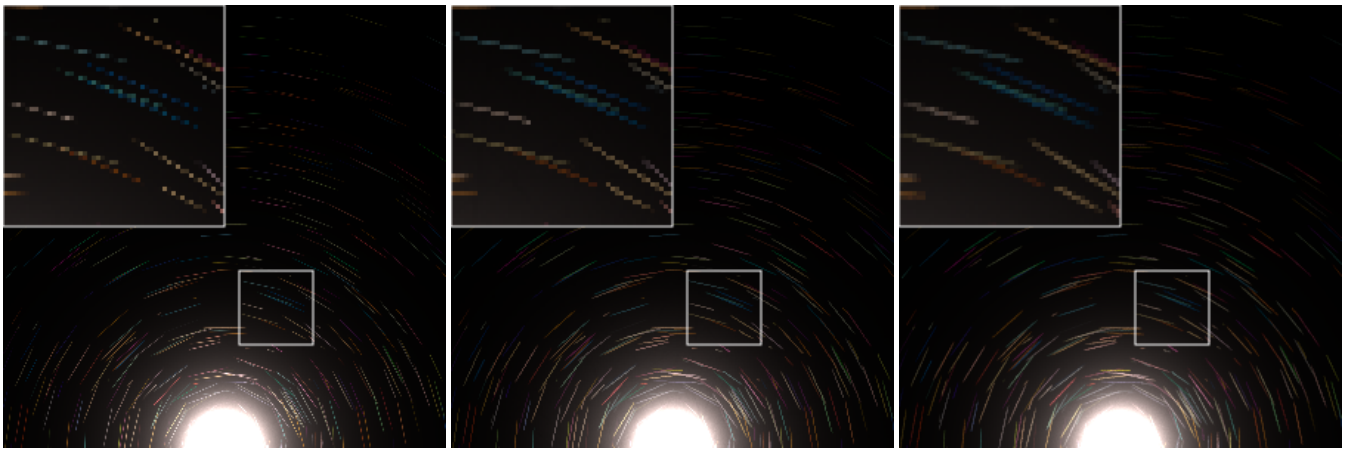

Figure 4: *Footprint antialiasing evaluation. Without spatial integration over the footprint (Left), scratches appear segmented. Footprint integration (Middle) generates connected scratches. Antialiasing is only performed along scratches which reflects the assumption that the Gaussian filter extent is much smaller than the scratch width. Monte Carlo integration (Right) additionally is able to anti-alias across scratches.*

appearance of CDs with predefined periodic structure. These results already improve the performance of the model for the specific case of incoherent surfaces.

3.2. Integration of the spatially varying reflectance over the camera pixel footprint

The full flux received by a pixel is determined by the irradiance impinging on its spatial footprint, which we approximate with an ellipse in the tangent plane with primary and secondary axes a_1 and a_2 respectively. In a wave-optical context with spatial coherence taken into account, we can think of the pixel footprint as being subsampled with coherent samples (i.e. Fourier transform of spatially weighted surface features). The coherent subsamples are converted to radiance (via BRDF) and then averaged over the surface area visible in the pixel, which corresponds to an incoherent superposition. Following Eq. (6) and (7) we obtain

$$\int_{\mathbf{x}\in\mathcal{P}} f_r(\xi,\mathbf{x}) \approx \frac{1}{A_\mathcal{P} A_\mathcal{C} \lambda^2} \int_{\mathbf{x}\in\mathcal{P}} \left[(1-\rho_c)\,|\mathcal{B}|^2 + \sum_m \left|\mathcal{S}^{(m)}\right|^2 \right] d\mathbf{x}, \tag{12}$$

where $\mathbf{x} = (x,y)^T$ is the point of intersection, $\mathcal{P}$ is the pixel footprint and $A_\mathcal{P} = a_1 a_2 \pi$ is a factor which normalizes the integral to the covered area. In the case of incoherent scratches (and scratches extending over the coherence area) the only term dependent on $\mathbf{x}$ is $\left|\eta_s^{(m)}\right|^2$ (cf. Eq. (10)). What is left to solve is then

$$\frac{1}{A_\mathcal{P}} \int_{\mathbf{x}\in\mathcal{P}} \left|\eta_s^{(m)}\right|^2 d\mathbf{x} = \frac{2A_\mathcal{C}}{A_\mathcal{P}} \int_{\mathbf{x}\in\mathcal{P}} \exp\left(-\frac{{s_b^{(m)}}^2}{\sigma^2} \right) d\mathbf{x}. \tag{13}$$

If this integral needs to be performed over an arbitrary region, interdependencies between the integration bounds for tangential and bitangential direction might occur. However, for our case it should be safe to assume that the footprint is much larger than the coherence area so that we can extend the integration bounds of the bitangential direction to

$$\begin{aligned} \int_{\mathbf{x}\in\mathcal{P}} \exp\left(-\frac{{s_b^{(m)}}^2}{\sigma^2} \right) d\mathbf{x} &\approx \int_{-\infty}^{\infty} \exp\left(-\frac{(s_y^{(m)} - y)^2}{\sigma^2} \right) dy \int_{s_1}^{s_2} dx \\ &= \sigma\sqrt{\pi} \cdot l_{\text{contained}}^{(m)}, \end{aligned} \tag{14}$$

where s_1 and s_2 are the intersection points of the scratch with the pixel footprint and thus the second integral yields the length fraction $l_{\text{contained}}^{(m)}$ of the scratch that is contained in the footprint. To ensure that we do not overestimate $l_{\text{contained}}^{(m)}$, we perform a line-ellipse intersection test and retrieve the accurate contained geometric length. We can then approximate the integral over spatial phases by

$$\left|\eta_\mathcal{P}^{(m)}\right|^2 := \int_{\mathbf{x}\in\mathcal{P}} \left|\eta_s^{(m)}\right|^2 d\mathbf{x} \approx 2\frac{\sigma^3}{a_1 a_2} \sqrt{\pi} \cdot l_{\text{contained}}^{(m)}. \tag{15}$$

We expect this approximation to be valid for larger footprints and to overestimate the integral for smaller ones. We therefore propose a correction term for small footprint-to-coherence area ratios which linearly interpolates between two limit cases of the integral. For the lower bound we assume the footprint size to be negligible, the upper bound is given by the previous formulation in Eq. (15). We thus obtain a corrected estimate of

$$\left|\eta_\mathcal{P}^{(m)}\right|^2 \approx \frac{2\sigma^2}{A_\mathcal{P}} \left[\alpha\sigma\sqrt{\pi^3} \cdot l_{\text{contained}}^{(m)} + (1-\alpha) A_\mathcal{P} e^{-{s_b^{(m)}}^2/\sigma^2} \right] \tag{16}$$

with the blending factor $\alpha = \min(\frac{1}{2}\frac{A_\mathcal{P}}{A_\mathcal{C}}, 1.0)$. With this correction term, we obtain low errors for the standard case of footprint areas about 7 times larger than the coherence area (cf. Fig. 5, middle). Although we overestimate the scratch area density with our approximation which lets scratches appear wider than expected, our method still yields visually close results in comparison to reference Monte Carlo renderings visible in Fig. 4.

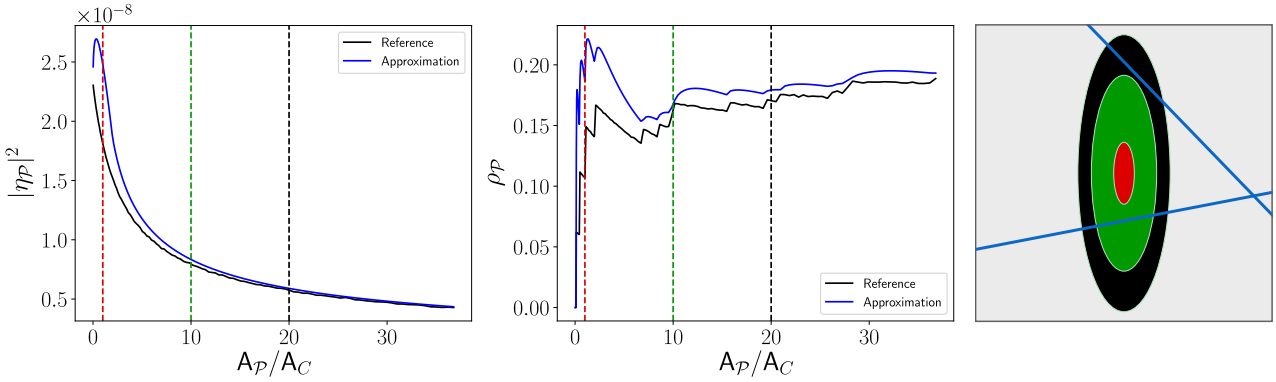

Figure 5: *Comparison of numeric (reference) vs analytic (approximation) results of the integration of scratch area density $\rho_{\mathcal{P}}$ and spatial phases $|\eta_{\mathcal{P}}|^2$, respectively given by the integral of ρ_c and $|\eta_s^{(m)}|^2$ over the pixel footprint. For footprint sizes smaller than the coherence area ($A_{\mathcal{P}}/A_{\mathcal{C}} < 1$, depicted by the dashed red line) our solution for $|\eta_{\mathcal{P}}|^2$ (Left) fails as a single pixel only observes a small fraction of the coherence area, an unlikely case which is not covered by our approximation. On the other hand, our solution for $\rho_{\mathcal{P}}$ (Middle, $\xi_1 = 0$) overestimates the area covered by scratches for smaller ratios $A_{\mathcal{P}}/A_{\mathcal{C}}$ but closely resembles the complex dependency. For increasing area ratios our assumptions are matched and the error becomes smaller. The dashed vertical lines correspond to ratios of footprint area to coherence area $A_{\mathcal{P}}/A_{\mathcal{C}} \in [1, 10, 20]$. The corresponding pixel footprints are depicted as color matching ellipses on the surface used for our simulations (Right).*

3.3. Scratch area density

To allow combination between the incoherent model and arbitrary base BRDFs, Werner et al. use a modified alpha-blending step that takes into account the scratch area contained within the coherence area. More formally, the resulting area density ρ_c is given by the Gaussian-weighted scratch length and its width. For scratches extending over the coherence area this yields

$$\begin{aligned}\rho_c &= \frac{1}{A_{\mathcal{C}}} \sum_m W^{(m)} \int_{s_1}^{s_2} ds \exp\left(-\frac{|\mathbf{s}''^{(m)}(s)|^2}{2\sigma^2}\right) \\ &\approx \frac{1}{A_{\mathcal{C}}} \sum_m W^{(m)} \sqrt{2\pi}\sigma \exp\left(-\frac{{s_b^{(m)}}^2}{2\sigma^2}\right) \end{aligned} \tag{17}$$

with $\mathbf{s}''^{(m)}(s) = \mathbf{s}'^{(m)} + s\mathbf{t}^{(m)}$, m denoting the m^{th} scratch. For integration over the pixel footprint, we have to modify the area density to account for *subsampling* of the footprint with samples of the size of the coherence area:

$$\begin{aligned}\rho_{\mathcal{P}} &= \frac{1}{A_{\mathcal{P}}} \int_{\mathbf{x}\in\mathcal{P}} \rho_c(\mathbf{x}) \\ &= \frac{1}{A_{\mathcal{P}} A_{\mathcal{C}}} \sqrt{2\pi}\sigma \sum_m W^{(m)} \int_{\mathbf{x}\in\mathcal{P}} \exp\left(\frac{-{s_b^{(m)}}^2}{2\sigma^2}\right) d\mathbf{x} \\ &\approx \frac{2}{A_{\mathcal{P}}} \sum_m W^{(m)} l^{(m)}_{\text{contained}}, \end{aligned} \tag{18}$$

where we used the solution of the integral found in Eq. (15). Eq. (18) now yields the mixing factor necessary for the alpha-blending between our scratch response and (arbitrary) microfacet base BRDFs which replaces ρ_c within the context of Eq. (6). Fig. 5 (top) reveals that for small footprint size, a high deviation between reference and approximation is obtained, which corresponds to unlikely cases of pixels covering only a tiny fraction of the coherence area. For increasing size however, the results agree well and only small errors are reported. Our footprint integration scheme relies on the extension of integration limits to $(-\infty, \infty)$ (cf. Eq. (14)). This enables us to develope antialiasing along scratches, but disables smoothing across scratches which becomes visible in Fig. 4, a result comparable to standard antialiasing techniques based on weighted area sampling (e.g [GS81]), albeit with an anisotropic filter kernel. With this we are able to significantly improve performance and recreate connected scratches with the minor disadvantage of non-perfect antialiasing.

3.4. Integration of the reflectance in angular domain over the solid angle subtended by area light sources

The angular dependence of the shading model can be fully expressed by Eq. (3) and (10). Both depend on the direction cosine along a specific direction. To derive an approximation of the integral over the solid angle subtended we must first rewrite the rendering equation, so that it is in direction cosine space. We start from the classical rendering equation first introduced to graphics community by Kajiya [Kaj86]:

$$L = \iint f(\phi_i, \phi_o, \theta_i, \theta_o) L_i \cos\theta_i \sin\theta_i \, d\theta_i \, d\phi_i. \tag{19}$$

The direction cosines irrespective of features in the tangent plane can be expressed as:

$$\begin{aligned}\xi_1 &= \cos\phi_i \sin\theta_i + \omega_{o,x} \\ \xi_2 &= \sin\phi_i \sin\theta_i + \omega_{o,y}. \end{aligned} \tag{20}$$

Rewriting these equations according to the azimuth and altitude angles yields

$$\omega_{o,t} = (\omega_{o,x}, \omega_{o,y})$$

$$\theta_i = \operatorname{asin}|\xi - \omega_{o,t}| \tag{21}$$

$$\phi_i = \operatorname{atan}\frac{\xi_2 - \omega_{o,y}}{\xi_1 - \omega_{o,x}}, \tag{22}$$

which we use to derive the Jacobian determinant for a transformation of the integral into direction cosine space:

$$|J| = \left| \frac{\delta(\theta_i, \phi_i)}{\delta(\xi_1, \xi_2)} \right| = \begin{vmatrix} \frac{\xi_1 - \omega_{o,x}}{\cos\theta_i \sin\theta_i} & \frac{\xi_2 - \omega_{o,y}}{\cos\theta_i \sin\theta_i} \\ -\frac{\xi_2 - \omega_{o,y}}{|\xi - \omega_{o,t}|^2} & \frac{\xi_1 - \omega_{o,x}}{|\xi - \omega_{o,t}|^2} \end{vmatrix} = \frac{1}{\cos\theta_i \sin\theta_i}, \quad (23)$$

with which the rendering equation can be expressed as:

$$L = \iint f(\xi_1, \xi_2) L_i \, d\xi_1 \, d\xi_2. \quad (24)$$

The derived relationship signifies that the cosine between the outgoing direction and the surface normals acts only as an offset of the projected light source and thus simplifies further derivations.

The BRDF that we are studying is separable in angular domain and changes according to the following relation:

$$\left| \mathcal{S}^{(m)}\left(\xi^{(m)}\right) \right|^2 \propto \mathrm{sinc}^2\left(\pi \frac{W^{(m)}}{\lambda} \xi_2^{(m)}\right) \exp\left(-\sigma^2 k^2 {\xi_1^{(m)}}^2\right). \quad (25)$$

The key observation is that the standard deviation of the Gaussian function is relatively low. We can establish an upper bound of the standard deviation by taking the highest wavelength in the visible spectrum $\lambda_{\max} = 750\,\mu\mathrm{m}$:

$$s_{\max} = \frac{\lambda_{\max}}{2\sqrt{2}\pi\sigma} = 1.40674 \cdot 10^{-3}. \quad (26)$$

A common empirical rule for upper bounds is to use three times the standard deviation; from this it becomes clear that for most practical purposes the area with the highest contribution would be significantly smaller than the projection of most light sources. The main benefit of this observation is that we can limit the integration to a small band around the origin in direction cosine space as shown in Fig. 6. Furthermore, we can ignore the curvature around the light source's edges such that integration can be performed separately for each axis. The integral of a Gaussian distribution at infinity is well known while the integral of $\mathrm{sinc}^2(x)$ can be derived through integration by parts. The final equation has the following form:

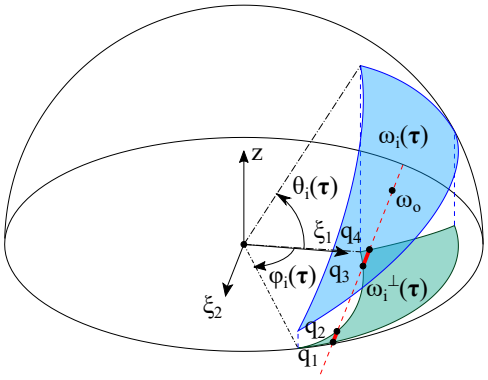

Figure 6: *Approximation of the integral over the solid subtended by triangular light source in direction cosine domain. We limit the integration over a small band along the bitangential direction cosine, starting at the origin* ω_o.

$$S(\xi|q_0, q_1)|^2 = {W^{(m)}}^2 {\mathcal{D}^{(m)}}^2 {\eta_s^{(m)}}^2 \left(\mathcal{N}^{(m)}(\xi, q_1) - \mathcal{N}^{(m)}(\xi, q_0)\right)$$
$$\mathcal{N}^{(m)}(\xi, q) = \frac{\sqrt{\pi}}{\sigma k}\left(2\frac{\mathrm{Si}(kW^{(m)}q)}{kW^{(m)}} - 4\frac{\sin^2(kW^{(m)}q/2)}{k^2 {W^{(m)}}^2 q}\right) \quad (27)$$

where q_0 are q_1 are the integration bounds according to ξ_2 which represents the azimuthal reflectance, and $\mathrm{Si}(x)$ is the trigonometric integral

$$\mathrm{Si}(x) = \int_0^x \mathrm{sinc}(t)\, dt = \int_0^x \frac{\sin(t)}{t}\, dt. \quad (28)$$

As $\mathrm{Si}(x)$ is a nonelementary integral and not available on the GPU, we approximate it as the superposition of a sigmoid and a hyperbolically decaying oscillation. The following equation is used by our code:

$$\mathrm{Si}(x) \approx \frac{\pi}{2}\tanh(x) - \frac{a_0 x}{1 + (x - \mathrm{sgn}(x) b_0)^2} + \mathrm{sgn}(x)\frac{1 - \cos(x)}{1 + |x|} \quad (29)$$

with $a_0 = 0.66142739$ and $b_0 = 0.38272292$ obtained via least-squares fit of the middle term to the residual in range $x \in [-10, 10]$ to prevent overestimation of the oscillations. We provide approximation error analysis in Figure 7.

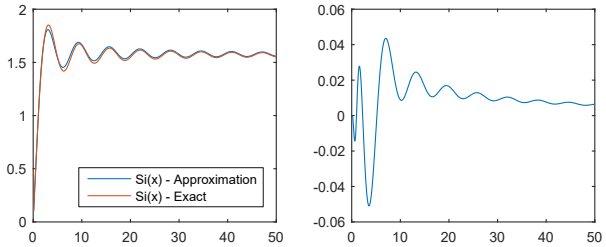

Figure 7: *Comparison between the approximation of* Si *and the exact function (left), as well as the corresponding residual (right).*

3.4.1. Spherical light sources

We compute the boundaries in direction cosine domain by using the normalized equation of a disk that can be expressed as:

$$\mathbf{G} = \frac{\mathbf{p} + r_t \cos\theta \mathbf{t} + r_b \sin\theta \mathbf{b}}{|\mathbf{p} + r_t \cos\theta \mathbf{t} + r_b \sin\theta \mathbf{b}|} \cdot p \quad (30)$$

The sphere is just a special case of this equation where the radius of the disk is expressed as $r = r_t = r_b$ and the major axes are perpendicular to the relative position $\mathbf{p}$ ($(\mathbf{t} \cdot \mathbf{p}) = (\mathbf{b} \cdot \mathbf{p}) = 0$), leading to

$$\mathbf{G}_S = \frac{\mathbf{p} + r\cos\theta \mathbf{t} + r\sin\theta \mathbf{b}}{\sqrt{r^2 + |\mathbf{p}|^2}}. \quad (31)$$

Constructing the orthogonal basis around $\mathbf{p}$ is achieved by using the branchless version of the code proposed by Duff et al. [DBC*17]. To find the intersection points with the line passing through the origin we must solve the following equation:

$$t_{\xi_1} r\cos\theta + b_{\xi_1} r\sin\theta + p_{\xi_1} + \omega_{o,\xi_1}\sqrt{r^2 + |\mathbf{p}|^2} = 0 \quad (32)$$

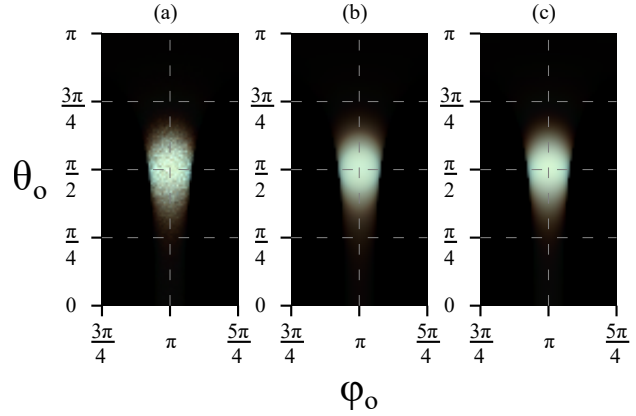

Figure 8: *Comparison of the specular lobe of (a) the ground truth Monte Carlo sampling of light source at 16k spp, (b) our spherical light source approximation and (c) the same approximation with exact* $\mathbf{Si}(x)$*. Equirectangular mapping of the viewing directions is used with radiance boosted ten times. The light source is placed close to the origin of the polar coordinate system and it is mirror reflected* π *radians around the axis while the scratch is perpendicular to the main axis.*

Using the trigonometric identities turns the equation into quadratic form:

$$\nu = \tan\frac{\gamma}{2}\,;\, \sin\gamma = \frac{2\nu}{1+\nu^2}\,;\, \cos\gamma = \frac{1-\nu^2}{1+\nu^2} \tag{33}$$

We use the method stated in Numerical Recipes [PTVF92] for solving quadratic equations in a numerically stable fashion:

$$D = {b_{\xi_1}}^2 r^2 - \left(\left(p_{\xi_1} + \omega_{o,\xi_1}\sqrt{r^2+|\mathbf{p}|^2}\right)^2 - {t_{\xi_1}}^2 r^2\right) \tag{34}$$

$$\nu = \left(\frac{-(b_{\xi_1} r + \mathrm{sgn}(b_{\xi_1})\sqrt{D})}{p_{\xi_1} + \omega_{o,\xi_1}\sqrt{r^2+|\mathbf{p}|^2} - t_{\xi_1} r}\right)^{\pm 1}, \tag{35}$$

and find $\sin\gamma$ and $\cos\gamma$ by back-substituting ν into the respective trigonometric identities (cf. Eq. (33)).

The complete solution requires clipping of the spherical disk to limit the contribution to the upper hemisphere. We perform clipping by making a check whether the intersection point is below the surface by substituting only the normal direction cosine component z in the numerator of Eq. (31). In case it is below the surface, we snap the point to the arc defined by the intersection of the disk with the plane that splits the circle into two hemispheres. The equation of an arc $\mathbf{C}^{(m)}$ can be defined as the circular interpolation in the plane formed by the projected vertex on the hemisphere $\mathbf{U}^{(m)}$ and the orthogonal vector $\hat{\Gamma}^{(m)}$:

$$\begin{aligned}
\mathbf{U}^{(m)} &= \frac{\mathbf{V}^{(m)} - \mathbf{x}}{|\mathbf{V}^{(m)} - \mathbf{x}|} \\
\mathbf{U}^{(m)} &= \left(\xi^{(m)} - \omega_{o,t}, \sqrt{1 - \left|\xi^{(m)} - \omega_{o,t}\right|^2}\right) \\
\Gamma^{(m)} &= \mathbf{U}^{(m+1)} - \mathbf{U}^{(m)}(\mathbf{U}^{(m)} \cdot \mathbf{U}^{(m+1)}) \\
\hat{\Gamma}^{(m)} &= \frac{\Gamma^{(m)}}{|\Gamma^{(m)}|} \\
\mathbf{C}^{(m)} &= \mathbf{U}^{(m)}\cos\gamma + \hat{\Gamma}^{(m)}\sin\gamma
\end{aligned} \tag{36}$$

where the angle on the circle is defined in the range $\gamma \in [0, \mathrm{acos}(\mathbf{U}^{(m)} \cdot \mathbf{U}^{(m+1)})]$. To find the intersection points with the plane between the two hemispheres we must solve the equation:

$$t_z r\cos\theta + b_z r\sin\theta + p_z = 0 \tag{37}$$

It is solved in analogy to Eq. (32), however, two corner cases have to be considered: The first is caused as the sphere approaches grazing angles and the projected ellipse of decreasing size eventually turns into a line. We detect this case and perform a simple intersection between a line and the arc to find the bounds of the integral. The second corresponds to a sphere fully below the horizon. We test whether the sphere center is further than one radius under the horizon and discard shading of the respective point upon validation.

To evaluate our derivation we rendered the response in angular domain according to different viewing directions by using our approximation and Monte Carlo sampling of the light source (Fig. 8). Apart from minor discrepancies due to our approximation of the $\mathrm{Si}(x)$-function, we closely match the ground truth. Thus our approach can be used without performing extraneous normalization.

3.4.2. Polygonal light sources

Polygonal lights can be represented by a collection of spherical triangles whose edges are defined as parts of three great circles. Each edge can be expressed as an arc via Eq. (36). To find the intersections with each edge the following equation must be solved:

$$\omega_{o,\xi_1} + U_{\xi_1}^{(m)}\cos\gamma + \hat{\Gamma}_{\xi_1}^{(m)}\sin\gamma = 0 \tag{38}$$

Analogously to Eq. (31) we must solve a quadratic equation that has the following roots:

$$D = \hat{\Gamma}_{\xi_1}^{(m)2} + {U_{\xi_1}^{(m)}}^2 - {\omega_{o,\xi_1}}^2 \tag{39}$$

$$\nu = \left(\frac{-\left(\hat{\Gamma}_{\xi_1}^{(m)} + \mathrm{sgn}\left(b_{\xi_1}\right)\sqrt{D}\right)}{\omega_{o,\xi_1} - U_{\xi_1}^{(m)}}\right)^{\pm 1} \tag{40}$$

In most cases, the line intersects exactly one edge. For the corner case when a point lies on two arcs we determine whether to keep if there is a difference of the derivatives according to x:

$$\frac{\delta y}{\delta x} = \frac{-\mathrm{U}_{\xi_2}^{(m)}\sin\gamma + \Gamma_{\xi_2}^{(m)}\cos\gamma}{-\mathrm{U}_{\xi_1}^{(m)}\sin\gamma + \Gamma_{\xi_1}^{(m)}\cos\gamma}. \tag{41}$$

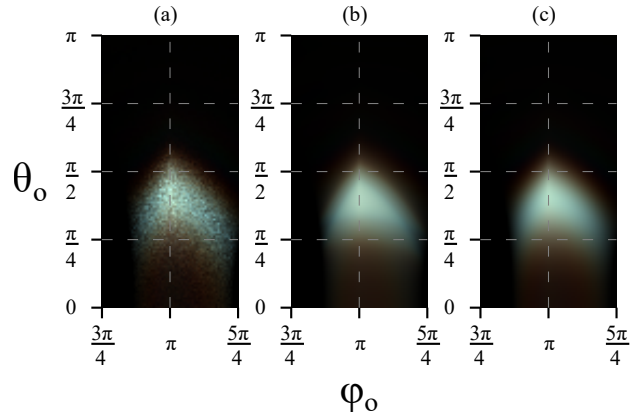

Figure 9: *Comparison of the specular lobe of (a) the ground truth Monte Carlo sampling, (b) our triangular light source approximation and (c) the same approximation with exact* $\mathrm{Si}(x)$ *computation. Equirectangular mapping of the viewing directions is used with radiance boosted ten times. The coordinate system follows the same principles that we outlined for the spherical light sources.*

After we have determined all intersection points we must form line segments. For this purpose we sort the roots by using bitonic sort [Bat68] by exploiting swizzling and the min and max operations supported by the GPU. We form a line segment out of each pair and compute Eq. (27). We clip the polygons by the plane that splits the hemisphere and form a new triangle when necessary.

Similar to the sphere implementation we validate the response in angular domain based on different viewing directions in Fig. 9. Analogously, minor discrepancies become visible that are related to the $\mathrm{Si}(x)$ approximation.

4. Implementation

We implemented our technique inside a real-time rendering framework following the principles expressed by Burns et al. [BH13]; we use rasterization to cast primary rays in a scene and directly sample the geometry. The main benefit of this technique is that our approach is evaluated only for the visible pixels. Another similar approach that was widely used in the real-time rendering community was G-buffer based deferred shading, which traces its origin to the technique proposed by Saito et al. [ST90]. This technique is not directly applicable to our approach as it would require to store the intersected scratches. Also, storing high quality normals requires additional bandwidth; more traditional ways of encoding the normal in 8-bit or 16-bit variables would lead to severe artifacts caused by the highly specular phenomenon that we are modeling. The shading pipeline used by our renderer follows the widely adopted principles within production real-time renderers [HMD*14].

We adopt two basic data structures to store the scratches within our pipeline. We use the basic structure covered by Werner et al. for the case when the footprint integration is enabled, as it requires complete knowledge about the scratches covered by a single pixel. For near-field rendering the visibility buffer structure can be exploited to optimize the intersection checks. We store the scratches associated with a single triangle that fall within the coherence area, which enables faster linear time lookup. It works very efficiently on the GPU for high polygon meshes at the expense of bigger storage requirements. Each node associated with a polygon stores a 32-bit index into a scratch reference buffer and a 32-bit number of scratches within a polygon. To obtain the scratches within the polygon we perform cylinder-line intersection tests where the radius is proportional to three times the coherence area. Additional experiments with a hierarchical version of this technique did not show significant improvement in any of the scenes that we tested, which we attribute to a more divergent execution profile.

We compute the projected camera pixel footprint by first applying the partial derivative functions (dFdx, dFdy) on the world coordinates of the intersection points, which is supported by the GPU. Then we construct an orthogonal basis on the surface and pass only the tangent and the extent of the pixel footprint ellipse to the functions that evaluate the spatial response.

We perform the computation individually for each light source. Clustering is also applicable to our rendering algorithms; however, we left the exploration of these techniques for future work. For the specific case of polygonal lights we evaluate the contribution of each triangle within the inner loop of the scratch intersection code.

5. Results

We performed a validation of our technique against classic Monte Carlo integration according to the publication by Werner et al. which is shown in Fig. 10. As expected from the angular domain plots (cf. Fig. 8, 9) our results match well. Main discrepancies are color differences of some of the scratches, most likely caused by our $\mathrm{Si}(x)$ approximation and the shape of the reflection by the base material. Note that we use the techniques described by Heitz and Dupuy et al. [HDHN16, DHB17] for computing the integral over the solid angle subtended of the base material by spherical and polygonal light source. These are known to fail to match the shape of the highlight at grazing angles. The spherical light source is situated around a meter away from the spoon and has a radius of 5 mm. We use a star shaped light composed of eight individual polygons to illustrate that our technique performs well with polygonal light sources of any shape. In both cases we use scratch parameters (width and depth) uniformly distributed between 0.8 and 6 microns. We show a faithful recreation (Fig. 11) of one of the high quality scenes from the original work by Werner et al. Furthermore, we include validation of our technique against one of the ground truth scenes in the supplemental material.

A benchmark of our technique on a notebook GPU (NVIDIA GTX 970M) is provided in Table 2; our technique works at real-time performance for spherical light sources. Computation cost is mostly dominated by intersection tests. Except for extreme close-ups, we achieve close to real-time performance which is most likely related to the more complex code for intersecting the projection of a triangle light source. Our technique that reuses the triangle intersections by building per triangle arrays noticeably improves performance. We attribute the low performance on the GPU on divergence caused by branching in the code. Applying our antialiasing technique increases the shading cost as we must intersect and evaluate shading for noticeably more scratches. We tested our

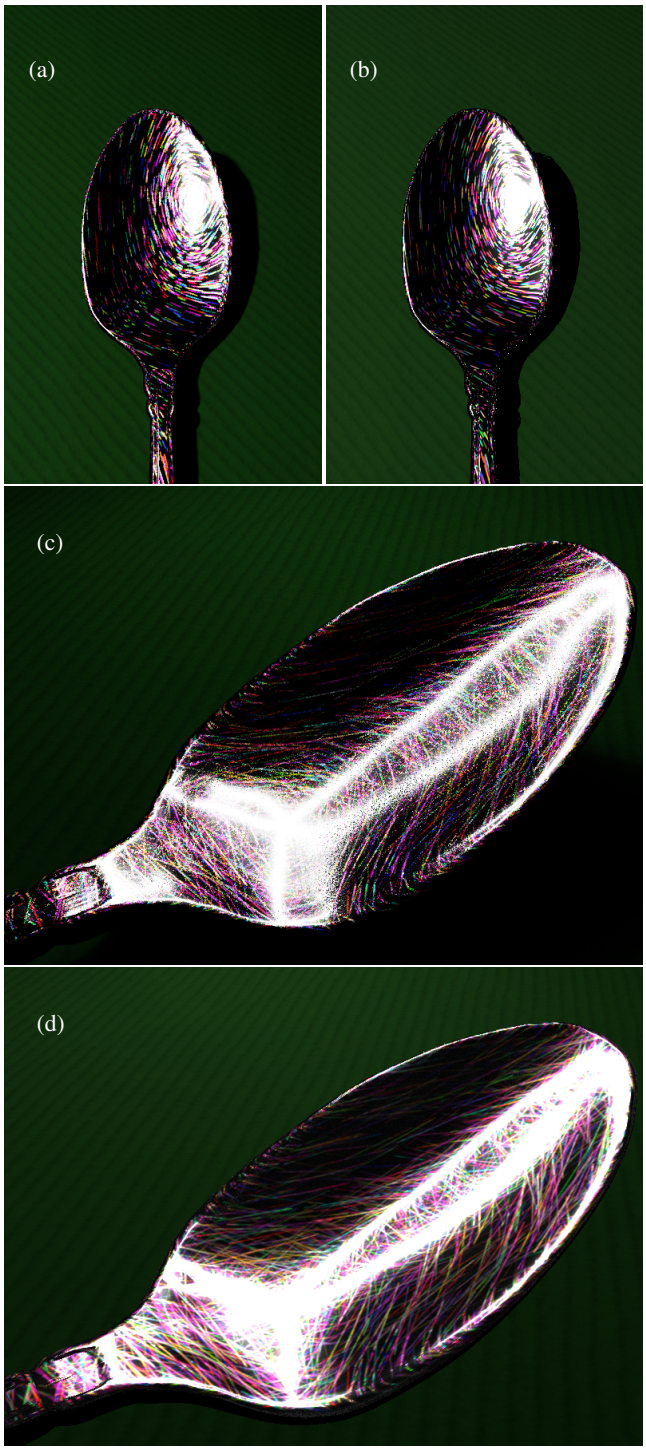

Figure 10: *Comparison between Monte Carlo integration over the solid angle subtended by (a) spherical and (c) polygonal light source against our approximation of the same integral for (b) spherical and (d) polygonal light source. The difference in the base material highlight is due to the properties of the LTC and SPTD distributions.*

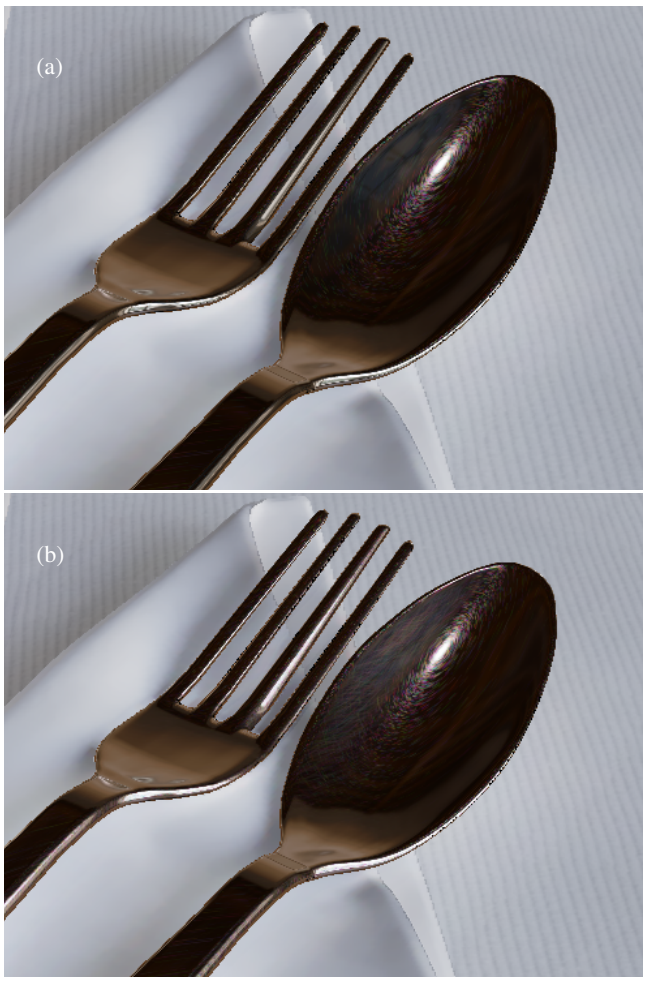

Figure 11: *Recreation of the dining table scene from the original work on iridescent scratches with finer scratches* $\sigma = 10nm$ *and 200000 individual scratches split between the spoon and the fork. Recreating one of the bright windows in the scene enhances the realism (b) compared to (a) a single light. We provide additional analysis of this many-lights approach in the supplemental material.*

technique on a previous generation desktop GPU (NVIDIA GTX 980) and it performs at least a factor of two faster in comparison to the notebook GPU. Further benchmarks against high quality scenes are provided in the supplemental material. We expect that our technique would perform much better on current and next generation GPUs making it a useful tool for visualization of iridescent scratches in real-time.

6. Discussion

We developed a set of techniques which allow the direct application of iridescent scratches within real-time rendering frameworks. Our techniques achieve several orders of magnitude of performance improvement over previous Monte Carlo sampling based techniques. However, to reach truly practical implementation quality for production code bases, several avenues for improvement remain, which we will discuss in the following.

	AA disabled						AA enabled		
Type	BVH			Per Triangle Array			BVH		
Zoom-In	0.5x	1x	2x	0.5x	1x	2x	0.5x	1x	2x
Intersection Only	4.1ms	4.4ms	5.5ms	2.9ms	3.5ms	3.8ms	18.1ms	17.5ms	15.5ms
Sphere	4.5ms	6.3ms	12.8ms	3.9ms	4.7ms	8.8ms	27.3ms	31.7ms	36.3ms
Triangle	13.9ms	26.5ms	60.6ms	12.5ms	21.1ms	43.4ms	48.5ms	60.3ms	94.1ms

Table 2: *Benchmark of the scene presented in Fig. 10 (b) with 10000 scratches split into 411810 segments with and without antialiasing enabled. AA is performed using camera footprint integration. The values were capture on a notebook GTX 970M GPU at 720p image resolution. We provide the respective images under different zoom levels in the supplemental material.*

Improved storage and lookup. We used direct segmentation of scratches according to the polygon boundaries of the shaded mesh. Another possibility that would improve storage efficiency is to store the extent of scratch particles as box-like structures. This technique is known as *deferred decals* in the real-time rendering community. It would greatly reduce the memory footprint of the data structures and reduce lookup times as more shallow data structures are traversed. Another possibility is to apply procedural models to greatly reduce storage. We aim to pursue these and other ideas that improve the memory usage.

More GPU friendly implementation. Currently, the intersection code is heavily reliant on branching to handle corner cases. We are interested in exploring other possible techniques that can handle this problem in a more GPU friendly branchless fashion.

Modeling improvements. We employ the simple Harvey model for our computations as doing an analytic solution of this model is tractable. We would like to explore possible solutions to handle the Generalized Harvey-Shack model [Kry06]. Another open problem is the handling of more complex scratch profiles as well as their variation along the scratch, as they cannot be integrated in closed form and necessitate more advanced antialiasing techniques. Although many possibilities exist for decomposing the 1D profile functions into terms that are easier to integrate, most of them are not particularly efficient memory-wise or performance-wise. We are also interested in deriving efficient techniques for approximating the contribution of many wavelengths as we limited the results shown in this publication to RGB. Currently, it can be approximated by evaluating the functions for many wavelengths in spectral rendering fashion and converting them to RGB. We provide performance evaluation of this approach in the supplemental material. Another limitation that we share with other area lighting techniques is handling of the Fresnel term which is only approximately supported for dielectrics by the distribution based approaches that fit the Schlick approximation. We are interested in exploring this problem in the future.

Handling of more complex light sources. We currently handle only spherical and polygonal light sources. However, the approach of handling the intersection in direction cosine space should fairly easily generalize to other geometry shapes. An often neglected but still very important open problem is the handling of shading by complex concave meshes. From a theoretic standpoint it is possible to clip polygons so that only visible faces are considered in the computation of the solid angle subtended, which, however, would be a very computation intensive task. We are interested in looking for solutions of this problem, especially with respect to the special case of rendering scratched surfaces and in the more generic case of microfacet theory based modeling. Rendering of environment light probes is also another possible direction for future research.

Shadow mapping. The closed form solution of shadow mapping by area light sources is considered an open problem in the real-time rendering community. Some special case solutions exist as the ones shown by Dupuy et al. [DHB17] for shadows cast by spheres. However, the generic case of arbitrary geometry of light sources and shadow casters is unsolved.

Acknowledgements

This work was supported by X-Rite Chair and Graduate School for Digital Material Appearance. We thank Wenzel Jakob for the many discussions regarding the scratch iridescence rendering project, as well as Rodrigo Martín and Heinz-Christian Steinhausen for their feedback on the manuscript.

References

[Arv95] ARVO J. R.: *Analytic Methods for Simulated Light Transport*. PhD thesis, Yale University, New Haven, CT, USA, 1995. 3

[Bat68] BATCHER K. E.: Sorting networks and their applications. In *Proc. April 30–May 2, 1968, Spring Joint Computer Conference* (1968), AFIPS '68 (Spring), ACM, pp. 307–314. 9

[BB17] BELCOUR L., BARLA P.: A Practical Extension to Microfacet Theory for the Modeling of Varying Iridescence. *ACM Trans. Graph. (Proc. SIGGRAPH) 36*, 4 (2017), 65. 1, 2

[BH13] BURNS C. A., HUNT W. A.: The visibility buffer: A cache-friendly approach to deferred shading. *J. Computer Graphics Techniques (JCGT) 2*, 2 (2013), 55–69. 9

[BN12] BRUNETON E., NEYRET F.: A survey of nonlinear prefiltering methods for efficient and accurate surface shading. *IEEE Transactions on Visualization and Computer Graphics 18*, 2 (2012), 242–260. 2

[BP93] BAO H., PENG Q.: Shading models for linear and area light sources. *Computers & Graphics 17*, 2 (1993), 137 – 145. 3

[BPMG04] BOSCH C., PUEYO X., MÉRILLOU S., GHAZANFARPOUR D.: A physically-based model for rendering realistic scratches. *Computer Graphics Forum 23*, 3 (2004), 361–370. 2

[BRW89] BAUM D. R., RUSHMEIER H. E., WINGET J. M.: Improving radiosity solutions through the use of analytically determined form-factors. In *Proc. 16th Ann. Conf. on Computer Graphics and Interactive Techniques* (1989), SIGGRAPH '89, ACM, pp. 325–334. 3

[DBC*17] DUFF T., BURGESS J., CHRISTENSEN P., HERY C., KENSLER A., LIANI M., VILLEMIN R.: Building an orthonormal basis, revisited. *J. Computer Graphics Techniques (JCGT) 6*, 1 (2017), 1–8. 7

[dCI17] DE CARPENTIER G., ISHIYAMA K.: Decima: Advances in lighting and AA. In *Advances in Realtime Rendering, Part II (SIGGRAPH 2017 Course)*, Tatarchuk N. et al., (Eds.). 2017. http://advances.realtimerendering.com/s2017/DecimaSiggraph2017.pdf. 3

[dFH*11] D'EON E., FRANCOIS G., HILL M., LETTERI J., AUBRY J.-M.: An energy-conserving hair reflectance model. *Computer Graphics Forum 30*, 4 (2011). 2

[DHB17] DUPUY J., HEITZ E., BELCOUR L.: A spherical cap preserving parameterization for spherical distributions. *ACM Trans. Graph.* (2017). 3, 4, 9, 11

[DKH*14] DACHSBACHER C., KŘIVÁNEK J., HAŠAN M., ARBREE A., WALTER B., NOVÁK J.: Scalable realistic rendering with many-light methods. *Computer Graphics Forum 33*, 1 (2014), 88–104. 3

[dMH13] D'EON E., MARSCHNER S., HANIKA J.: Importance sampling for physically-based hair fiber models. In *SIGGRAPH Asia 2013 Technical Briefs* (2013), SA '13, ACM, pp. 25:1–25:4. 2

[Dro14] DROBOT M.: Physically based area lights. In *GPU Pro 5*, Engel W., (Ed.). CRC Press, 2014, pp. 67–100. 3

[DTS*14] DHILLON D., TEYSSIER J., SINGLE M., GAPONENKO I., MILINKOVITCH M., ZWICKER M.: Interactive diffraction from biological nanostructures. *Comput. Graph. Forum 33*, 8 (2014), 177–188. 2

[DWMG15] DONG Z., WALTER B., MARSCHNER S., GREENBERG D. P.: Predicting appearance from measured microgeometry of metal surfaces. *ACM Trans. Graph. 35*, 1 (2015), 9:1–9:13. 2

[GGG*16] GUARNERA D., GUARNERA G., GHOSH A., DENK C., GLENCROSS M.: Brdf representation and acquisition. *Comput. Graph. Forum 35*, 2 (2016), 625–650. 1, 2

[Goo96] GOODMAN J.: *Introduction to Fourier Optics*. McGraw-Hill Series in Electrical and Computer Engineering: Communications and Signal Processing. McGraw-Hill, 1996. 2

[GS81] GUPTA S., SPROULL R. F.: Filtering edges for gray-scale displays. In *Proc. 8th Ann. Conf. on Computer Graphics and Interactive Techniques* (1981), SIGGRAPH '81, ACM, pp. 1–5. 6

[GTGB84] GORAL C., TORRANCE K., GREENBERG D., BATTAILE B.: Modeling the interaction of light between diffuse surfaces. In *Proc. 11th Ann. Conf. on Computer Graphics and Interactive Techniques* (1984), SIGGRAPH '84, ACM, pp. 213–222. 3

[GUnK*17] GUILLÉN I., UREÑA C., KING A., FAJARDO M., GEORGIEV I., LÓPEZ-MORENO J., JARABO A.: Area-preserving parameterizations for spherical ellipses. *Computer Graphics Forum (Proceedings of EGSR) 36*, 4 (2017). 3

[HDHN16] HEITZ E., DUPUY J., HILL S., NEUBELT D.: Real-time polygonal-light shading with linearly transformed cosines. *ACM Trans. Graph. 35*, 4 (2016), 41:1–41:8. 3, 4, 9

[HH17] HEITZ E., HILL S.: Linear-light shading with linearly transformed cosines. In *GPU Zen: Advanced Rendering Techniques*, Engel W., (Ed.). Bowker Identifier Services, 2017. 3, 4

[HMD*14] HILL S., MCAULEY S., DUPUY J., GOTANDA Y., HEITZ E., HOFFMAN N., LAGARDE S., LANGLANDS A., MEGIBBEN I., RAYANI F., DE ROUSIERS C.: Physically based shading in theory and practice. In *ACM SIGGRAPH 2014 Courses* (2014), SIGGRAPH '14, ACM, pp. 23:1–23:8. 9

[HP17] HOLZSCHUCH N., PACANOWSKI R.: A two-scale microfacet reflectance model combining reflection and diffraction. *ACM Trans. Graph. (Proc. SIGGRAPH) 36*, 4 (2017), 66:1–66:12. 1, 2

[JHY*14] JAKOB W., HAŠAN M., YAN L.-Q., LAWRENCE J., RAMAMOORTHI R., MARSCHNER S.: Discrete stochastic microfacet models. *ACM Trans. Graph. 33*, 4 (2014), 115:1–115:10. 1

[Kaj86] KAJIYA J. T.: The rendering equation. In *Proc. 13th Ann. Conf. on Computer Graphics and Interactive Techniques* (1986), SIGGRAPH '86, ACM, pp. 143–150. 6

[Kar13] KARIS B.: Real shading in unreal engine 4. In *SIGGRAPH '13: ACM SIGGRAPH 2013 Courses* (2013), ACM. 3

[Kel97] KELLER A.: Instant radiosity. In *Proc. 24th Ann. Conf. on Computer Graphics and Interactive Techniques* (1997), SIGGRAPH '97, ACM Press/Addison-Wesley Publishing Co., pp. 49–56. 3

[Kry06] KRYWONOS A.: *Predicting Surface Scatter Using a Linear Systems Formulation of Non-paraxial Scalar Diffraction*. University of Central Florida, 2006. 2, 3, 11

[LDSM16] LECOCQ P., DUFAY A., SOURIMANT G., MARVIE J.-E.: Accurate analytic approximations for real-time specular area lighting. In *Proc. 20th ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games* (2016), I3D '16, ACM, pp. 113–120. 3

[LGX*13] LEVIN A., GLASNER D., XIONG Y., DURAND F., FREEMAN W., MATUSIK W., ZICKLER T.: Fabricating brdfs at high spatial resolution using wave optics. *ACM Trans. Graph. 32*, 4 (2013), 144:1–144:14. 2

[PA91] POULIN P., AMANATIDES J.: Shading and shadowing with linear light sources. *Computers & Graphics 15*, 2 (1991), 259–265. 3

[Pic92] PICOTT K. P.: Extensions of the linear and area lighting models. *IEEE Comput. Graph. Appl. 12*, 2 (1992), 31–38. 3

[PTVF92] PRESS W. H., TEUKOLSKY S. A., VETTERLING W. T., FLANNERY B. P.: *Numerical Recipes in Fortran 77: The Art of Scientific Computing*, 2nd ed. Cambridge University Press, 1992. 8

[Ram09] RAMAMOORTHI R.: *Precomputation-Based Rendering*. Now Publishers Inc., Hanover, MA, USA, 2009. 2

[RGB16] RAYMOND B., GUENNEBAUD G., BARLA P.: Multi-scale rendering of scratched materials using a structured sv-brdf model. *ACM Trans. Graph. 35*, 4 (2016), 57:1–57:11. 2

[SKS02] SLOAN P.-P., KAUTZ J., SNYDER J.: Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. In *Proc. 29th Ann. Conf. on Computer Graphics and Interactive Techniques* (2002), SIGGRAPH '02, ACM, pp. 527–536. 2, 3

[Sny96] SNYDER J.: *Area light sources for real-time graphics*. Tech. Rep. MSR-TR-96-11, Microsoft Research, 1996. 3

[ST90] SAITO T., TAKAHASHI T.: Comprehensible rendering of 3-d shapes. In *Proc. 17th Ann. Conf. on Computer Graphics and Interactive Techniques* (1990), SIGGRAPH '90, ACM, pp. 197–206. 9

[Sta99] STAM J.: Diffraction shaders. In *Proc. 26th Ann. Conf. on Computer Graphics and Interactive Techniques* (1999), SIGGRAPH '99, ACM Press/Addison-Wesley Publishing Co., pp. 101–110. 2, 4

[SWZ96] SHIRLEY P., WANG C., ZIMMERMAN K.: Monte carlo techniques for direct lighting calculations. *ACM Trans. Graph. 15*, 1 (1996), 1–36. 3

[TG17] TOISOUL A., GHOSH A.: Practical acquisition and rendering of diffraction effects in surface reflectance. *ACM Trans. Graph. 36*, 5 (2017). 2

[TT97] TANAKA T., TAKAHASHI T.: Fast analytic shading and shadowing for area light sources. *Computer Graphics Forum 16* (1997), C231–C240. 3

[UnFK13] UREÑA C., FAJARDO M., KING A.: An area-preserving parametrization for spherical rectangles. In *Proc. Eurographics Symposium on Rendering* (2013), EGSR '13, Eurographics Association, pp. 59–66. 3

[WVJH17] WERNER S., VELINOV Z., JAKOB W., HULLIN M. B.: Scratch iridescence: Wave-optical rendering of diffractive surface structure. *ACM Trans. Graph. (Proc. SIGGRAPH Asia) 36*, 6 (2017), 220:1–220:11. 1, 2, 3, 4

[YHJ*14] 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 (2014), 116:1–116:9. 1

[YHMR16] YAN L.-Q., HAŠAN M., MARSCHNER S., RAMAMOORTHI R.: Position-normal distributions for efficient rendering of specular microstructure. *ACM Trans. Graph. 35*, 4 (2016), 56:1–56:9. 1, 2

[YJR17] YAN L.-Q., JENSEN H. W., RAMAMOORTHI R.: An efficient and practical near and far field fur reflectance model. *ACM Trans. Graph. (Proc. SIGGRAPH) 36*, 4 (2017). 2