A Practical Real-Time Model for Diffraction on Rough Surfaces

Olaf Clausen[1]
TH Köln

Martin Mišiak[1]
University of Würzburg

Arnulph Fuhrmann
TH Köln

Ricardo Marroquim
TU Delft

Marc Erich Latoschik
University of Würzburg

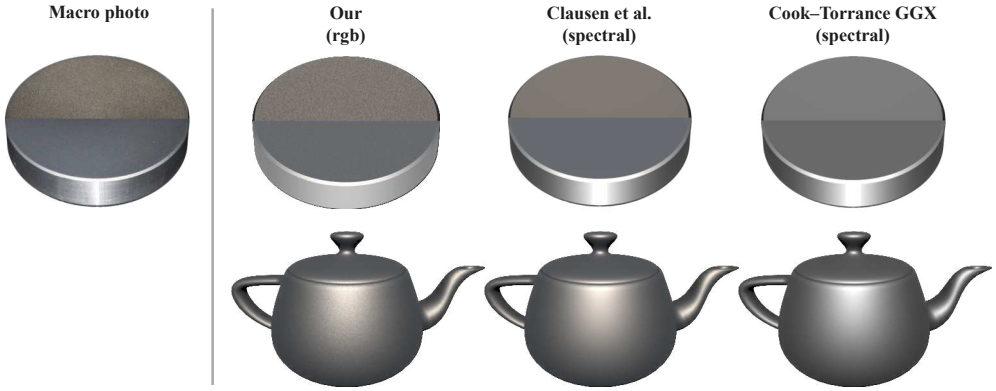


Figure 1. Upper row: The macro photo on the left shows a rough aluminum sample under a forward (upper half) and backscattering (lower half) case. On the right, renderings with our new RGB model, including speckle patterns, the original spectral approach by Clausen et al. [2023] and the Cook–Torrance GGX model are depicted. Lower row: The teapot renderings demonstrate the visual impact of diffraction-induced color shifts and speckle patterns.

Abstract

Wave optics phenomena have a significant impact on the visual appearance of rough conductive surfaces even when illuminated with partially coherent light. Recent models address these phenomena, but none is real-time capable due to the complexity of the underlying physics equations. We provide a practical real-time model, building on the measurements and model by Clausen et al. [2023], that approximates diffraction-induced wavelength shifts and speckle patterns with only a small computational overhead compared to the popular Cook–Torrance

[1] Authors contributed equally to this work.

 ISSN 2331-7418

GGX model. Our model is suitable for virtual reality applications, as it contains domain-specific improvements to address the issues of aliasing and highlight disparity.

1. Introduction

Light simulation defines the visual appearance of virtual objects. It remains a major challenge in computer graphics due to the complexity of the light-matter interaction, which is well described in physics only for specific cases under defined boundary conditions. Computer graphics scenes are usually too complex to simulate light propagation in a holistic physical way. Therefore, light simulation is heavily approximated and only obeys a few fundamental laws of physics. These approximations simplify light simulation by taking advantage of the limitations of human vision so that it can be computed efficiently and, ideally, without introducing perceptible errors. However, in practice, the latter is usually not the case.

The light reflection from homogeneous materials is often described by the bidirectional reflectance distribution function (BRDF). BRDF models are usually based on geometric optics, assuming light propagation along straight paths. This approach simplifies the computation and is an excellent approximation when the considered geometric scale is significantly above the wavelength scale of visible light. However, many real-world materials break this assumption, and some visually prominent phenomena can only be described by the wave nature of light. Diffraction is one such phenomenon that is visible on hair, fur, scratches, small repeating structures, and rough surfaces, even when illuminated with partially coherent light. Recently, several BRDF models based on wave optics were developed to address these observations. Most of them integrate wave optical theory from physics, which makes them computationally expensive.

A different approach was proposed by Clausen et al. [2023]. They acquired a multimodal dataset of aluminum samples with varying roughness, which they used to analyze diffraction phenomena on rough surfaces. Their analysis shows that single reflections, predominately in the forward scattering, lead to a linear wavelength shift toward long wavelengths, thus to a reddish appearance. Analogously, multiple scattering, predominately in backscattering, leads to a linear shift toward short wavelengths, hence a bluish appearance. Based on these new insights, they provided a shift function that closely approximates the observed diffraction phenomenon. The shift function extends existing microfacet models, such as the popular Cook–Torrance model, without impacting the performance.

Although their model improves the simulation of the wavelength shifts, it ignores the spatial characteristic of speckle patterns, which has a great impact on the visual appearance, as illustrated in Figure 1. Further, the provided model was developed in the spectral domain, but real-time applications are typically restricted to RGB space.

In this work, we address both issues. We first transfer the shift function from the spectral to the RGB domain, resulting in a linear function as simple as the original one. Second, we analyze the speckle patterns on rough surfaces with the macro photos provided by Clausen et al. [2023]. The analysis shows that the rough aluminum samples have a Gaussian color distribution that can be well approximated by a multivariate normal distribution defined by a covariance matrix. We present a model to approximate the covariance matrix as a function of the incidence and reflection directions, which, combined with a simplex noise function, closely resembles the speckle patterns observed in reality.

Finally, we integrate our diffraction model into a popular game engine and introduce two specific improvements for the usage in virtual reality (VR) applications. First, we handle unwanted aliasing arising from the high-frequency pattern by performing runtime filtering, while modulating the speckle intensity based on its projected screen size. In addition to this, we promote the specularity of a material by introducing a controlled, stereoscopic shift into the speckle pattern.

The model is still simple enough to avoid any performance issues while, at the same time, significantly improving the realism of rough metallic surfaces compared to the classic Cook–Torrance model.

2. Related Work

In real-time rendering, BRDF models based on microfacet theory are widely used, such as the popular Cook–Torrance model. This model assumes the microsurface as a collection of randomly distributed tiny mirrors, where their distribution and reflectance are described by the normal distribution function and the Fresnel equations, respectively. Moreover, the shadowing-masking function describes the visibility of microfacets given incidence and reflection directions. The model is parameterized by a roughness value and the refractive index. Further, it obeys the physical laws of positivity, Helmholtz reciprocity, and energy conservation. Overall, the model provides a good trade-off between being physically based and artist-friendly while allowing for high-performance rendering and is currently a standard in industry and research. However, when fitting the model against real-world measurements, some scatter phenomena are not well represented.

One explanation for the deviations between the model and measured data is diffraction, as stated by Holzschuch and Pacanowski [2017]. Yan et al. [2018] and Clausen et al. [2018; 2023] also noticed visible prominent diffraction phenomena on rough surfaces, dielectric and conductive, even when illuminated with partially coherent light. These phenomena can only be described by the wave nature of light. Thus, BRDF models based on geometric optics, such as the Cook–Torrance model, do not take them into account.

In physics, two different analytical approaches describe light-matter interaction: the Rayleigh–Rice (RR) vector perturbation theory [Rice 1951; Church et al. 1979] and the Beckmann–Kirchhoff (BK) scattering theory [Beckmann and Spizzichino 1987]. Both are complementary in their range of validity. The Rayleigh–Rice theory is valid for smooth surfaces for arbitrary incidence and reflection angles. The Beckmann–Kirchhoff theory is valid for rougher surfaces but only for small incidence and reflection angles. The generalized Harvey–Shack (GHS) theory [Krywonos et al. 2011] combines the advantages of the previous approaches without their limitations. Nevertheless, Schröder et al. [2011] noticed deviations between measured data and the GHS theory when dealing with rougher surfaces. The authors assumed that these deviations resulted from measurement errors. All presented theories only consider the first surface reflection, thus don't take into account multiple reflections.

A few BRDF models were developed based on these theories, which we group into those describing the average BRDF for a large area and those considering the reflectance from spatially varying surfaces, including anisotropic surface details on the microscale.

Large Area He et al. [1991] extended the Cook–Torrance model using the BK scattering theory. Their model matches well the measured data of a few materials. Nevertheless, Ngan et al. [2005] showed that when fitting against the MERL database [Matusik et al. 2003], He's model leads to similar results as the Cook–Torrance model. Löw et al. [2012] introduced a BRDF model based on the RR theory. Even though it is limited to smooth surfaces, it represents such cases from the MERL database very well.

A different approach was presented by Dong et al. [2016]. For the first time, the anisotropic appearance of metal samples was determined only by their microgeometry. They demonstrated that both geometric-based and Kirchhoff-based approaches can model the appearance properly. However, a systematical validation of the proposed model with measured BRDFs is missing.

Holzschuch and Pacanowski [2017] assumed the surface reflectance as a superposition of the reflectance on the macro- and microgeometry. The first is characterized by the standard Cook–Torrance model, and the second by the modified Harvey–Shack theory. The model can reproduce some wavelength-dependent phenomena observed in the MERL database but still carries the limitations of the modified Harvey–Shack approach.

Spatially Varying The work by Werner et al. [2017] models the light reflection from surfaces covered with a collection of randomly oriented scratches using a Harvey–Shack-based wave optics model. The surface reflection is approximated by a smooth BRDF superimposed by one-dimensional scratches, enabling the calculation of re-

flection effectively and analytically. Velinov et al. [2018] furnished an extension of Werner's model, which provides real-time frame rates for a simple scene with 1000 scratches and two light sources. However, since the computational cost depends on the number of scratches, the frame rate rapidly drops as scene complexity increases.

Yan et al. [2018] derived a generalized BRDF representation where the three scalar diffraction theories, Harvey–Shack, generalized Harvey–Shack, and Kirchhoff, can be used alternatively. Their model allows the simulation of full diffraction effects of arbitrary microscale height-field geometries. Clausen et al. [2023] applied this model to measured microgeometries of rough aluminum samples to simulate the observed diffraction-based wavelength shifts. Indeed, Yan's method includes the linear wavelength shift toward longer wavelengths in the forward scattering; however, it does not represent the shift toward shorter wavelengths in the backscattering, which can be attributed to the fact that the model does not consider multiple reflections.

Recently, Xia et al. [2023] introduced a practical wave optics reflection model for hair and fur. They predicted the surface reflection and diffraction from rough fibres with a 3D wave simulator and characterized the highlight noise of the resulting simulations in terms of the statistics of a random process. They proposed a practical model suitable for rendering, which approximates the optical speckle in the reflection highlight with wavelet noise. The model was designed for simulating the reflection and diffraction from rough fibres, thus it is not suitable for rough surfaces. However, their work inspired us to the extent that we also approximate the speckle pattern with procedural noise.

To the best of our knowledge, none of the presented models fully represent the wavelength shifts observed by Clausen et al. [2023] because none of them calculates diffraction phenomena caused by multiple reflections, thus neglecting the blue shift toward backscattering. Further, the presented models are unsuitable for real-time applications due to the computational complexity of the underlying physics equations. This is particularly true for VR applications that demand high frame rates and image resolution. The model provided by Clausen et al. [2023], on the other hand, reasonably approximates the wavelength shifts in forward and backscattering while being interactive, as well as artist-friendly, since it is based on the Cook–Torrance model and two intuitive parameters controlling the shift function. However, their model is only available in the spectral domain, and the spatially varying characteristics of the speckle patterns are not considered.

3. Spectral to RGB

When illuminating a rough surface, conductive or dielectric, diffraction leads to a reddish and bluish appearance in forward– and backscattering. This diffraction-based phenomenon can be approximated by a shift function multiplied by a microfacet

model, such as the popular Cook–Torrance model [Cook and Torrance 1982]. The formula for a conductive surface is

$$f_{\text{diffraction}}(i, o, n, \lambda) = \mathit{shift} \cdot f_r, \tag{1}$$

where f_r describes the microfacet model and shift is the linear function. The shift function depends on the wavelength λ and the slope function m:

$$\mathit{shift}(i, o, \lambda) = m(i, o) \cdot (\lambda - 600) + 1. \tag{2}$$

This in turn depends on θ_m, the angle between the halfvector and surface normal n:

$$m(i, o) = h \cdot \cos(w \cdot \theta_m), \tag{3}$$

where i and o are the incidence and reflection directions, and w and h scale the cosine function in the x- and y-axes, respectively.

This diffraction model is defined in the spectral domain as indicated by the wavelength dependency. However, real-time applications, particularly VR applications, simulate light interaction in the RGB color space to reduce computational complexity. To make the model available for a wide range of applications, it has to be transferred to the RGB color space.

In RGB rendering, the multiplication of a reflectance with an illumination spectrum is usually approximated by multiplying the respective RGB values. Ensure that the reflectance spectra are multiplied by the relative spectral power distribution corresponding to the white point of the respective RGB color space before conversion. Otherwise, a uniform reflectance spectrum results in nonuniform RGB values; thus in the light simulation, the reflected light color differs from the incident light color. In the same way, we assume that the multiplication of the shift function with the microfacet model, as depicted in Equation (1), is closely approximated by the multiplication of their RGB values. Under this assumption, the conversion of the diffraction BRDF model from the spectral domain to the XYZ color space can be written as follows:

$$\int_\lambda \overline{m}(\lambda)\mathit{shift}(\lambda) f_r(\lambda) I(\lambda)\, d\lambda \approx \int_\lambda \overline{m}(\lambda)\mathit{shift}(\lambda) I(\lambda)\, d\lambda \cdot \int_\lambda \overline{m}(\lambda) f_r(\lambda) I(\lambda)\, d\lambda, \tag{4}$$

where $\overline{m} \in \{\bar{x}, \bar{y}, \bar{z}\}$ are the CIE standard observer functions, I is the spectral power distribution of a reference illuminant, and the integrals are computed over the visible light spectrum. Since microfacet models are already available in the RGB space, we derive only the conversion of the shift function.

First, we calculate the XYZ tristimulus values of the shift function multiplied by

the reference illuminant:

$$\begin{aligned} X &= \frac{1}{N}\int \bar{x}(\lambda)\mathit{shift}(\lambda)I(\lambda)\,d\lambda, \\ Y &= \frac{1}{N}\int \bar{y}(\lambda)\mathit{shift}(\lambda)I(\lambda)\,d\lambda, \\ Z &= \frac{1}{N}\int \bar{z}(\lambda)\mathit{shift}(\lambda)I(\lambda)\,d\lambda, \\ N &= \int \bar{y}(\lambda)I(\lambda)\,d\lambda. \end{aligned} \tag{5}$$

Next, we substitute the shift function from Equation (2) and rewrite the resulting integral as a sum of two integrals:

$$\begin{aligned} X &= m(i,o)\frac{1}{N}\int \bar{x}(\lambda)(\lambda-600)I(\lambda)\,d\lambda + \frac{1}{N}\int \bar{x}(\lambda)I(\lambda)\,d\lambda, \\ Y &= m(i,o)\frac{1}{N}\int \bar{y}(\lambda)(\lambda-600)I(\lambda)\,d\lambda + \frac{1}{N}\int \bar{y}(\lambda)I(\lambda)\,d\lambda, \\ Z &= m(i,o)\frac{1}{N}\int \bar{z}(\lambda)(\lambda-600)I(\lambda)\,d\lambda + \frac{1}{N}\int \bar{z}(\lambda)I(\lambda)\,d\lambda. \end{aligned} \tag{6}$$

Note that the slope function m is independent of the wavelength and can be removed from the first integral.

The evaluation of the two integrals returns the tristimulus values (X_s, Y_s, Z_s) and (X_I, Y_I, Z_I), which are then converted into the RGB color space by multiplying them by the inverted transformation matrix M, which is calculated from the RGB reference primaries:

$$\begin{aligned} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} &= m(i,o)\begin{bmatrix} X_s \\ Y_s \\ Z_s \end{bmatrix} + \begin{bmatrix} X_I \\ Y_I \\ Z_I \end{bmatrix}, \\ \begin{bmatrix} r \\ g \\ b \end{bmatrix} &= \Big[M\Big]^{-1}\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = m(i,o)\begin{bmatrix} r_s \\ g_s \\ b_s \end{bmatrix} + \begin{bmatrix} r_I \\ g_I \\ b_I \end{bmatrix}. \end{aligned} \tag{7}$$

In conclusion, in RGB space, the shift function is defined by the constant RGB values (r_s, g_s, b_s) and (r_I, g_I, b_I) and the slope function m. We computed the conversion for the widespread color spaces AdobeRGB and sRGB. The values for AdobeRGB are

$$\begin{aligned} (r_s, g_s, b_s) &= (14.12228819, -56.99255935, -155.01640388), \\ (r_I, g_I, b_I) &= (1.0, 1.0, 1.0), \end{aligned} \tag{8}$$

and for sRGB are

$$\begin{aligned} (r_s, g_s, b_s) &= (42.45318742, -56.98651893, -159.23105974), \\ (r_I, g_I, b_I) &= (1.0, 1.0, 1.0). \end{aligned} \tag{9}$$

4. Analysis of Speckle Patterns

The model provided by Clausen et al. [2023] closely resembles the color shifts observed in the macro photos. However, there are still clear visual differences. In Figure 1, the sample rendered with Clausen's model looks uniform and flat, whereas, in the macro photos (left), the sample appears textured and rough. When examined in detail, these differences become even more pronounced. The macro photos show, particularly in forward scattering, colorful speckle patterns, which Clausen's model does not consider. Our model, on the other hand, closely resembles both the color shifts and the speckle pattern, thus improving the rendering of rough surfaces.

Speckle patterns can be accurately simulated by physics models, and their behavior is well described in literature [Goodman 2020]. However, to describe the complex behavior of the speckle pattern in a more practical way, suitable for real-time rendering, it has to be simplified. Therefore, we analyzed the macro photos of six aluminum samples with varying roughness provided by Clausen et al. [2023]. The macro photos were captured under a fixed camera with several illumination angles, and their resolution is sufficient to analyze the speckle patterns manifested on the sample surfaces. The thorough analysis gave us a better understanding of the speckle color distribution, spatial correlation, and dependency on the incidence and reflection directions. In the following, we present the results of the roughest sample. Nevertheless, we observed similar results for the remaining smoother samples.

4.1. Speckle Distribution

In the first step, we analyzed the color distributions of the speckle patterns. To avoid artifacts due to surface and illumination irregularities, we isolated a patch of 500×500 pixels from the macro photos as illustrated in Figure 2. The corresponding distribu-

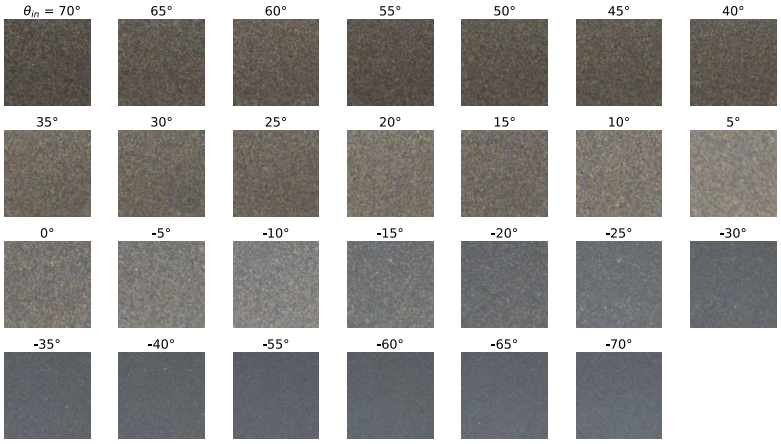
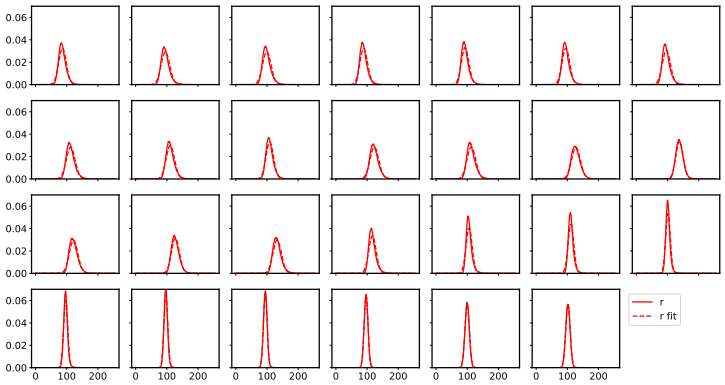
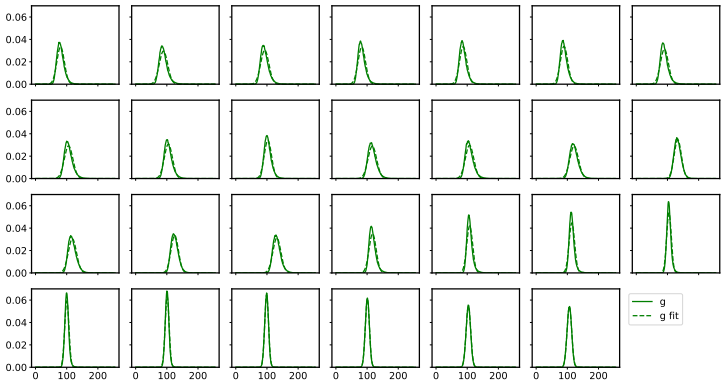


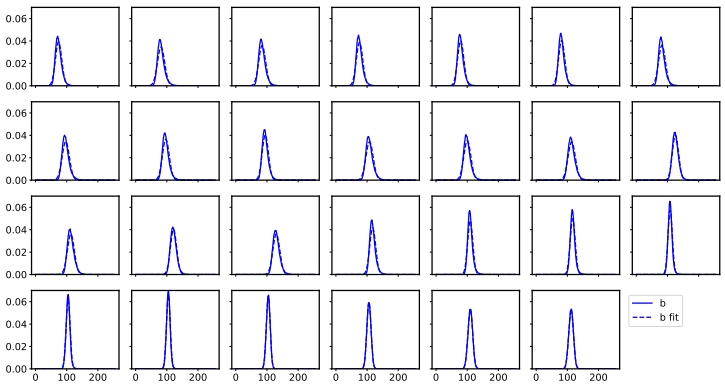
Figure 2. Macro photos (500×500 px) of a rough aluminum sample captured under varying illumination angles.



(a) *Red distribution*



(b) *Green distribution*



(c) *Blue distribution*

Figure 3. The macro photos' RGB distributions and their fitting with a Gaussian function.

tions of the red, green, and blue channels, as well as the fitting with a Gaussian function, are shown in Figure 3. The Gaussian function describes the probability density function of a normally distributed random variable and is given by

$$p(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right). \tag{10}$$

The close fit of the RGB distributions with the Gaussian function, where the mean μ and variance σ^2 are calculated from the macro photos, demonstrates that the RGB values are normally distributed for all incidence angles. This observation heavily simplifies the simulation of the speckle patterns, as it allows us to describe the distributions statistically. From the plots, we further observe that an increase in reflection angle is accompanied by a decrease in variance. This trend is of great importance when modelling the speckle distribution as a function of incidence and reflection angles, as detailed in Section 4.3.

In Figure 4, we plot the RGB values resulting from the macro photos and two statistical approaches in 3D scatter plots to investigate their distribution in RGB space. The naive approach (middle column) assumes that the RGB channels are uncorrelated, thus approximating the speckle distribution by three independent Gaussian normal distributions. The required mean and variance values for each channel are again calculated from the macro photos. This approach does not represent the RGB dis-

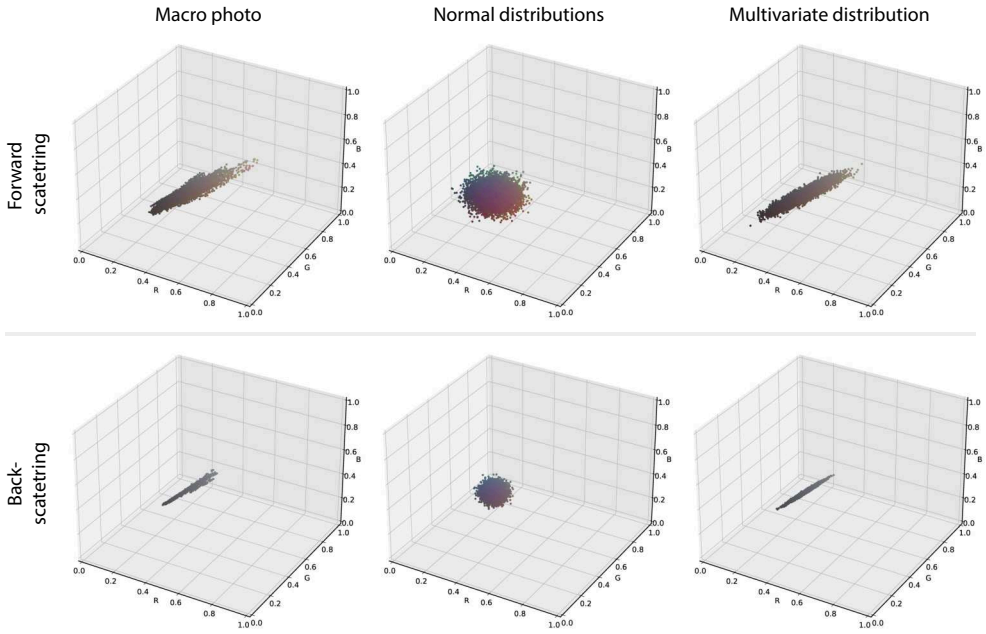


Figure 4. Three-dimensional plots of the RGB distributions of the macro photos (left column), the simulation with three independent Gaussian normal distributions (middle column), and the simulation with a multivariate normal distribution (right column).

tributions from the macro photos (left column) well since it generates, as expected, uncorrelated, spherically distributed RGB values for both forward and backscattering. In reality, the RGB values are correlated, resulting in less saturated colors than with the naive approach. Such a multidimensional distribution where the features are correlated and each feature is normally distributed can be described by a multivariate normal distribution. The probability density function of a multivariate normal distribution is given by

$$p(x \mid \mu, \Sigma) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right), \tag{11}$$

where μ is a vector of the feature's mean values, and Σ is the covariance matrix describing the variances of the individual feature on the diagonal and the pairwise covariance between features on the off-diagonal elements. In the following, we use for Equation (11) the expression $X \sim \mathcal{N}(\mu, \Sigma)$ with X being a vector-valued random variable $X = [X_1 \dots X_n]^T$.

The multivariate normal distribution closely approximates the captured speckle distributions in both forward and backscattering and outperforms the naive approach with three independent normal distributions.

4.2. Spatial Correlation

When taking a closer look at the macro photos in Figure 5, we see that each colored speckle spans across multiple pixels, making its RGB values spatially correlated. Hence, simulating speckle patterns requires an approach providing spatial correlation while at the same time obeying a certain color distribution given by a covariance matrix and a mean vector. For static images, this can be achieved by creating random values with a multivariate normal distribution and mean value of zero, convolving them with a Gaussian filter, and finally adding the target mean value. The resulting speckle patterns are shown in the middle column of Figure 5. Although this approach visually agrees well with the measured speckle patterns, it is impractical for real-time rendering applications.

Instead of convolving spatially uncorrelated samples, we use a simplex noise function to generate our samples. Simplex [Perlin 2002] is a gradient-based noise approach commonly used in real-time computer graphics that considers the spatial correlation of its samples. In the case of the OpenSimplex Noise algorithm,[1] the generated values are almost normally distributed with a mean value close to zero. Upon normalization using the function's standard deviation, we obtain a multivariate normal distribution $x \sim \mathcal{N}(0,\, I_d)$ with a unit covariance matrix I_d. We can compute the target distribution $y \sim \mathcal{N}(\mu,\, \Sigma\,)$ using the transformation $y = Lx + \mu$, where L is the Cholesky decomposition of $\Sigma : \Sigma = LL^T$. The simplex noise approach leads to

[1] https://pypi.org/project/opensimplex/

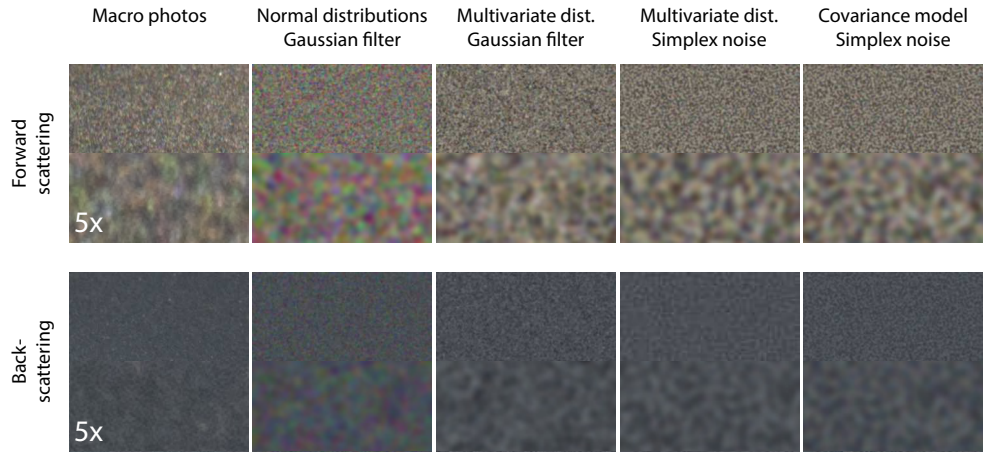


Figure 5. Simulation of the spatial characteristic of speckle patterns with a Gaussian filter (columns 2–3) and the simplex noise approach (columns 4–5).

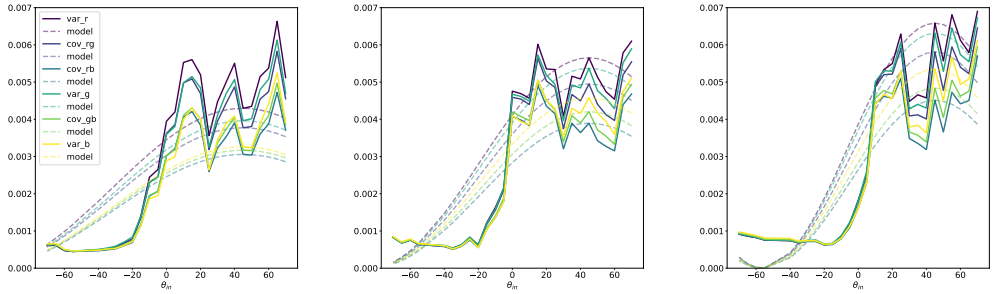
visually similar results as the Gaussian convolution approach, as shown in the fourth column of Figure 5. Since the OpenSimplex noise and Gaussian approaches have the same multivariate normal distribution, they only differ in the spatial distribution. The OpenSimplex noise tends to look more structured than the Gaussian approach.

So far, we calculated the required mean μ and covariance matrix Σ from the macro photos. In a reflection model, both parameters have to be predicted as a function of the incidence and reflection directions. The model by Clausen et al. [2023] provides the required mean value, but the covariance matrix is unknown. In the following section, we present a model to approximate the covariance matrix as a function of incidence and reflection directions.

4.3. Dependency on Incidence and Reflection Directions

In Section 4.1, we claimed that the RGB distribution of the speckle patterns depends on the incidence and reflection angles. We observed that the speckle patterns appear colorful in forward scattering and homogeneous in backscattering, which is confirmed by the decreasing trend of the variance toward backscattering. To further investigate this dependency, we calculate the covariance matrices of the macro photos and plot them as a function of the incidence angle. Figure 6 illustrates the resulting distributions of three aluminum samples with varying roughness. We conclude that the covariance distribution is similar for different sample roughness, where the covariance values are spread and form a plateau in the forward scattering while converging and falling off quickly toward backscattering. The size and height of the plateau depend on the sample's roughness, where rougher samples have a lower but broader plateau.

The distribution of the covariance values reminds us of the slope distribution of



(a) *Sample 1: rms 2.39* **(b)** *Sample 2: rms 1.76* **(c)** *Sample 3: rms 1.20*

Figure 6. Variance and covariance values of the covariance matrix at the reflection angle of $-45°$ as a function of the incidence angle θ_{in} and the respective approximation with the covariance model for three samples with varying roughness. The root mean square (rms) defines the surface's roughness, where rough surfaces have large rms. Thus, Sample 1 is the roughest and Sample 3 the smoothest surface.

the linear wavelength shift, which Clausen et al. [2023] approximated in Equation (3) by a cosine function. The cosine function depends on θ_m and is scaled in the x- and y-axes by the parameters w and h, respectively.

Clausen et al. [2023] have clear evidence that diffraction caused by the first and multiple surface reflections leads to positive and negative slopes, thus to shifts toward longer and shorter wavelengths, respectively. Since speckle patterns are caused by diffraction, we conclude that the speckle statistic depends on the surface scattering in a similar way as the slope. Therefore, we developed the covariance model in Equation (12) based on the shift function (Equation (2)), where we replace the parameter h by cov_{init} and place the cosine function in the range $[0, 1]$.

$$cov(i, o) = cov_{\text{init}} \cdot 0.5 \cdot (\cos(w \cdot \theta_m) + 1). \tag{12}$$

The cov_{init} parameter is defined by the covariance matrix at the mirror direction. Hence, the covariance model assumes that the covariance matrix values are maximum at the mirror direction and decrease according to the cosine function away from the mirror direction.

The approximated covariance values are plotted in Figure 6 with a dashed line. Though the covariance model offers only a rough approximation of the measured values, it effectively generates visually pleasing reproductions of the speckle patterns, as demonstrated in the final column of Figure 5. It is worth noting that the top row depicts a section spanning approximately 7×3.5 mm. Consequently, when we observe materials with such surface characteristics, the speckle patterns appear smaller, thereby making deviations from the macro photos even less noticeable. The covariance model only works properly for rough surfaces due to the periodic nature of the cosine function. As discussed in Section 5.5, for smooth surfaces the covariance

values increase again toward backscattering, leading to the reappearance of speckle patterns.

Until now, we have shown how to create a spatially correlated speckle pattern that obeys a given statistical color distribution. However, it is well known from speckle patterns of lasers that the incidence and reflection directions also influence the spatial distribution of the pattern. As the observer or the light source moves parallel to the surface, the pattern changes while maintaining the same color distribution. Due to limited angular resolution of the provided macro photos, we studied the speckle pattern's spatial behavior in the laboratory. We observed that the speckle pattern changes spatially when manipulating the light or viewing position—except when moving along the incidence or reflection direction.

To simulate this behavior, we sample our speckle pattern from a 4D simplex noise function $sn(u, v, \phi, \theta)$. The first two dimensions (u, v) describe the spatial dependency of the reflecting surface and the last two (ϕ, θ) the dependency on the camera and light source directions. To reduce the degrees of freedom, we consider only the halfway vector, as it captures changes in both the incidence and the reflection direction. For ϕ and θ we use the polar and azimuthal angles between the halfway vector and the surface normal: ϕ_m and θ_m. This parametrization, in combination with the 4D simplex function, allows us to approximate the smooth spatial pattern variations as a result of light, viewer, and object movement.

5. Real-Time Implementation

To show the practicability of our described model for rendering diffraction-based speckle patterns, we integrated it into the Unity game engine. The approach is computationally inexpensive, which we demonstrate by evaluating it in the context of a VR application. In the following, we describe the implementation details of our used material model, the relevant shader computations, and how we handle minification and highlight disparity.

5.1. Material Model

The proposed model is based on the work of Clausen et al. [2023]. Therefore, we inherit the parameters of the Cook–Torrance BRDF and the spectral shift function (h, w). Our model introduces additional parameters, namely cov_{init}, S_x, S_y, M, T, and H, which offer control over the statistics and behavior of the speckle pattern, as well as its binocular appearance. In the following, we provide a detailed explanation of each model parameter and elaborate on methods for determining them through measurements or manual configuration.

α Surface roughness (for a definition, see Walter et al. [2007]).

η Material refractive index (for a definition, see Walter et al. [2007]).

w Scaling factor of the shift's cosine function along the x-axis. The parameter dictates the rate at which the wavelength shift of the reflectance spectrum transitions from a positive slope in forward scattering to a negative slope in backscattering. As smoother surfaces concentrate forward scattering into a narrower cone, the w parameter tends to be larger for smoother surfaces. Clausen et al. determined this parameter through fitting against measured data, obtaining $w = 2.3394$ for a rough surface ($\alpha = 0.3922$) and $w = 6.4455$ for a smoother surface ($\alpha = 0.14033$). In the absence of measured data, the w parameter for surfaces with roughness values within this range can be estimated using linear interpolation.

h Scaling factor of the shift's cosine function along the y-axis. The parameter governs the degree of tilt of the reflectance spectrum, thus influencing the strength of the red and blue tinge observed in forward and backscattering, respectively. As diffraction-induced wavelength shifts are more pronounced on rougher surfaces, the h parameter tends to be larger for such surfaces. Clausen et al. determined this parameter through fitting against measured data, obtaining $h = 0.001025$ for a very rough surface ($\alpha = 0.3922$) and $h = 0.00077$ for a smoother surface ($\alpha = 0.14033$). In the absence of measured data, the h parameter for surfaces with roughness values within this range can be estimated using linear interpolation.

cov_{init} Initial covariance matrix. This is a material-specific property obtained from measured data. It can be determined by capturing a macro photo (≈ 1800 dpi) in the mirror direction of the sample surface and computing its covariance matrix. Alternatively, the initial covariance matrix can be computed through simulation using the physics model by Yan et al. [2018] and a synthetic microgeometry with the corresponding roughness. A Cholesky decomposition is applied to this matrix before passing it to the shader.

S_x, S_y Scaling factors for controlling the spatial frequency of the speckle pattern. The size of a single speckle depends on the used surface parametrization of a given object. Since this parametrization can vary across objects, these factors can be used to increase further or decrease the size of the final pattern. When rendering for current generation VR headsets, we recommend targeting speckle sizes of 1 mm and larger.

M Scaling factor for the incidence and reflection direction-based pattern variations. As the viewer and the light source move, the diffraction pattern changes as well. We approximate this effect using a heuristic based on the halfway vector. The factor M scales the polar and azimuthal angle differences between the

halfway vector and the surface normal, which are input into the simplex noise function. In our examples, we use a value of $M = 7$.

T Convergence threshold for minification. If this many speckles fall onto the footprint of a display pixel, the pattern will no longer be rendered. Instead, the singular shift value is used, which corresponds to an average of the speckle pattern. The threshold value should be set to the multisampling rate used during the minification process. Higher threshold values allow the diffraction pattern to be seen from farther distances, however, at the cost of introducing potential aliasing.

H Scaling factor for highlight disparity. When viewing a surface point, the speckle pattern is slightly different for the left and right eye. This results in highlight disparity. The scaling factor H is used to control the offset of the speckle pattern between both eyes, which influences the amount of perceived highlight disparity. A value of 0 disables this effect, while larger values offset the pattern by a fraction of a speckle (up to a maximum of 1 speckle offset). In our testing, a good mixture of viewing comfort and highlight disparity is achieved using a value of $H = 0.5$.

In addition, we also employ the RGB shift factors $(r_s, g_s, b_s), (r_I, g_I, b_I)$ computed in Section 3, which are constant per color space.

5.2. GPU Implementation

The presented approach does not require any lengthy preprocessing and simply extends an existing Cook–Torrance BRDF evaluation term for a single light source. The code can be found in Listing 1. We omit the code for the shift function and the covariance model, as they are relatively straightforward implementations of Equations (7) and (12), respectively.

As described in Section 4.2, the diffraction pattern is the additive result of a multivariate simplex noise distribution and the computed shift value. Due to the high storage requirements of a precomputed 4D texture, the simplex noise function is evaluated at runtime in the shader. We use an implementation by Stefan Gustavson[2]. In comparison to the CPU-based OpenSimplex shown in Figure 5, Gustavson's GPU implementation tends to create more repeating patterns with a somewhat blocky appearance. We alleviate this to an extent by additionally perturbing the noise input vector.[3] The resulting noise does not have an ideal normal distribution; however, in natural viewing conditions, the visual differences to the CPU implementation are rather small.

[2] https://github.com/stegu/webgl-noise
[3] https://github.com/KdotJPG/OpenSimplex2

```
float3 EvaluateBRDF( ... , float2 uv, float3 halfVec, float3 n )
{
    // ...Compute F, Ds, Gs terms for Cook-Torrance BRDF
    //

    float3 shift             = Shift_function(θ_m, w, h);
    float cov_factor         = Cov_model(θ_m, w);
    float3x3 covar_decomp    = sqrt(cov_factor) * cov_init;

    float2 d_uv          = float2(uv.x*S_x, uv.y*S_y);
    float3 rnd_number    = SampleNoise(d_uv, halfVec, normal);
    rnd_number           = mul(covar_decomp, rnd_number);
    float3 BRDFspecular = (shift + rnd_number) * (F * Ds * Gs);

    return BRDFspecular;
}
```

Listing 1. BRDF evaluation code in HLSL.

The first two input values for the simplex function are derived from an existing surface parametrization (uv map). To adjust the spatial frequency of the final speckle pattern, additional scaling factors S_x, S_y are applied to the uv coordinates. To obtain the required multivariate noise distribution, the sampled noise value is multiplied by the Cholesky-decomposed covariance matrix. Here, we use a straightforward implementation of the Cholesky–Banachiewicz 3×3 matrix decomposition, which is performed on the cov_{init} matrix during a preprocessing step. The adjusted noise value and the result of the shift function are then used to multiply the remaining BRDF terms.

5.3. Minification

An important aspect of rendering small-scale effects, such as the proposed speckle pattern, is minification. Though the speckle pattern is clearly visible in close interactions, it can be problematic for further viewing distances. As multiple speckles fall onto the footprint of a single screen pixel, the diffraction pattern becomes a source of aliasing.

To address this problem, we leverage the fact that the used simplex distribution has a mean of zero and is "overlaid" on top of the shift value. Hence, as the number of speckles per pixel increases, the noise amplitude is reduced toward zero—making the diffraction pattern converge to the shift value. To avoid aliasing during the convergence time, we additionally employ multisampling on the noise distribution. Given a number of speckles N_s that are currently projected onto the footprint of a display pixel, the multisampling rate N_m and the convergence threshold T, the noise distri-

```
float3 SampleNoise( float2 uv, float3 h, float3 n )
{
    const float UV_TO_SPECKLE_FACTOR        = 0.5;
    // Amplitude Reduction Falloff
    const float A_R_F                       = 1.0; // T = (ARF + 1)^2

    // Compute screen-space derivative of uvs
    float delta_uv      = max_derivative(ddx(uv),ddy(uv));

    // How many speckles fit along one dimension of a screen pixel
    // is equivalent to sqrt(N_s)
    float sqrt_speckles_per_pixel = delta_uv / UV_TO_SPECKLE_FACTOR;
    // 0 to 1 sqrt_speckles_per_pixel -> no modulation
    // 1 to (1 + A_R_F) -> linear reduction
    float amplitude_modulation    =
        1.0 - min((max(speckles_per_pixel - 1, 0) / A_R_F), 1.0);

    // Determine the last two dimensions of the 4D lookup
    float3 h_polar = cartesian2Polar(h);
    float3 n_polar = cartesian2Polar(n);
    // M: Pattern shift factor
    float h_a = (h_polar.x - n_polar.x) * M;
    float h_p = (h_polar.y - n_polar.y) * M;

    float3 filtered_noise = float3(0,0,0);
    if ( (sqrt_speckles_per_pixel > 1.0) &&
       (  sqrt_speckles_per_pixel <= (1 + A_R_F)))
    {
        // Sample location and sampling radius
        filtered_noise = multisample(uv, delta_uv);
    } else if (sqrt_speckles_per_pixel <= 1.0 )
    {
        filtered_noise = snoise(uv, h_a, h_p);
    }
    // Modulate noise amplitude toward 0, to converge against shift
    return amplitude_modulation * filtered_noise;
}
```

Listing 2. Noise distribution sampling code in HLSL.

bution is sampled as follows:

$$\texttt{sampleNoise}(N_s) = \begin{cases} sn(u,v,\phi,\theta), & \text{if } N_s \leq 1, \\ A(N_s)\frac{1}{N_m}\sum_{i=1}^{N_m} sn(u_i,v_i,\phi,\theta), & \text{if } 1 < N_s \leq T, \\ 0, & \text{if } N_s > T, \end{cases} \tag{13}$$

with $A(N_s)$ being the amplitude reduction function:

$$A(N_s) = 1 - \min\left(\frac{N_s}{T}, 1\right). \tag{14}$$

The current number of speckles per pixel N_s is computed using the screen-space derivatives of the surface parametrization used to index the noise function. This value is then doubled, as one noise period contains two speckles (positive and negative peak). The code for the noise distribution sampling can be found in Listing 2.

Setting $T = N_m$ avoids any aliasing, as the effective number of speckles per pixel, N_s, is never greater than the sampling frequency. This is ensured via the amplitude reduction term in Equation (14), by which the noise amplitude is reduced to zero once N_s exceeds the convergence threshold T. Setting $T > N_m$ will result in a diffraction pattern that is visible from a greater distance at the cost of potential aliasing after N_s exceeds N_m. In our implementation, we set $N_m = 4$ samples and the convergence threshold to $T = 6$. We did not notice any disturbing aliasing for this parametrization when used in VR.

5.4. Highlight Disparity

An important material quality for stereoscopically viewed surfaces, such as those in VR, is highlight disparity [Blake and Bülthoff 1990]. It describes the difference in binocular disparities between a specular reflection and the underlying reflector surface. It improves the realism of a given material by strengthening its specular appearance. While the diffraction pattern is not a specular reflection as defined by a geometric optics approach, it clearly exhibits highlight disparity. This can be observed on surfaces with a diffraction pattern, where the pattern requires a different eye convergence angle than the underlying surface.

In Section 4.3, we described how the diffraction pattern depends on the camera and light source angles. A naive application of our technique should result in highlight disparity as each eye position corresponds to a different halfway vector. In practice, however, the viewer is more likely to experience binocular rivalry as the diffraction pattern changes entirely between the two eyes. As this can be very distracting and uncomfortable for the users, the presented technique requires special handling in stereoscopic rendering contexts.

Instead of computing the halfway vector between the current eye and the light source, we use the cyclopean eye (located between the left and right eye). Before sampling the noise distribution, we offset the azimuth angle ϕ_m by a fraction of a speckle size:

$$\phi = \phi_m + \frac{1}{2} HE, \tag{15}$$

where $H \in [0, 1]$ is a user-controllable variable for the strength of the highlight disparity, and E is the index of the currently rendered eye (0 or 1). This guarantees that

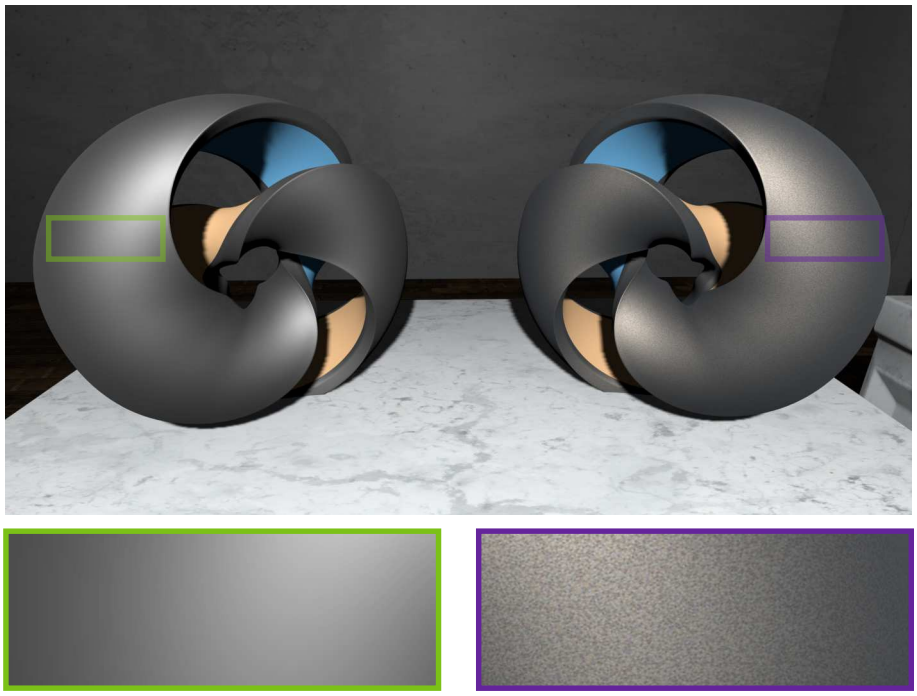


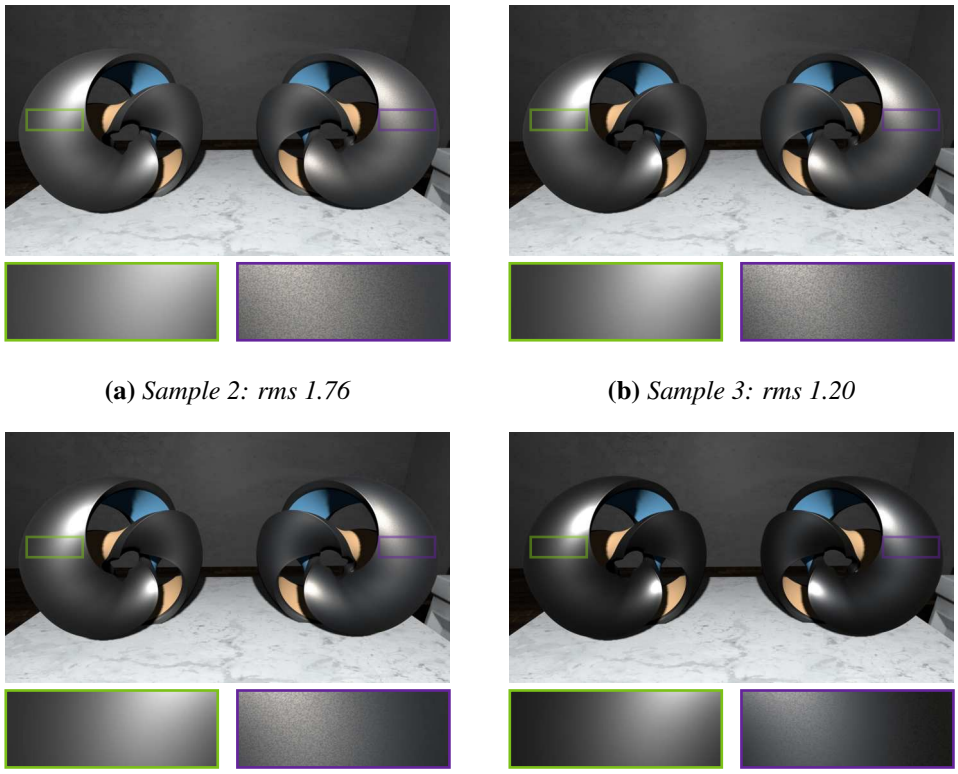
Figure 7. Rendering of rough metallic objects with the Cook–Torrance GGX model (left) and our model (right). The material parameters are determined using Sample 1 (rms 2.39) and are listed in Table 1. The close-ups demonstrate the impact of wave optics phenomena on the appearance of metallic objects, which are not considered by the Cook–Torrance model.

the pattern remains very similar between both eyes as it can change at most by a half noise period.

5.5. Evaluation and Results

In the following, we shortly present rendering results and evaluate the computational costs. Figure 1 depicts on the left macro photos of a rough aluminum sample and on the right respective renderings computed with our RGB model (left), the original spectral approach [Clausen et al. 2023] (middle), and the popular Cook–Torrance GGX model (right). The renderings of the aluminum samples demonstrate that our RGB model resembles the red and blue shifts in the forward and backscattering in the same way as the spectral approach. Since our model also takes into account the spatially varying speckle patterns, it better resembles the macro photos than the original model, thus improving the level of realism. In Figure 7, we further compare our model with the Cook–Torrance GGX model in a more realistic museum environment. With

our model (right), the object appears rougher and more metallic than with the Cook–Torrance GGX model (left), leading to an overall more realistic look. The close-up on the right shows the transition from strong speckle patterns with an on-average reddish appearance in the forward scattering to barely visible speckle patterns with an on-average bluish appearance in the backscattering. These diffraction-based phenomena are not considered by the Cook–Torrance model, resulting in a much more synthetic appearance.



(a) *Sample 2: rms 1.76*

(b) *Sample 3: rms 1.20*

(c) *Sample 4: rms 0.98*

(d) *Sample 5: rms 0.52*

Figure 8. Renderings of the aluminum Samples 2–5 using the parameters listed in Table 1.

Our model works also for smoother surfaces, as shown in Figure 8. However, when the surface becomes smooth (see Sample 5), the covariance model struggles to properly approximate the speckle pattern. Since smooth surfaces reflect light in a narrow cone, small wavelengths are required to represent the rapid fading of speckle patterns away from the mirror direction with a cosine function, which in turn leads to a reappearance of speckles in backscattering.

To test the performance of our implementation, we created a simple exhibit scene, showcasing four different objects using the presented diffraction model. The scene is rendered to a Meta Quest 3 VR headset with a per-eye resolution of 2432×2592,



Figure 9. Test scene used in our performance evaluation. Left: An overview of all objects. Right: Close-up inspection of a single model.

using an NVIDIA 4070Ti GPU. All presented frame times are for both eyes combined. The main computational cost of our technique is the evaluation of the 4D noise function. We compared the performance overhead of the presented diffraction model against a reference implementation of the Cook–Torrance BRDF in three scenarios. In all tested scenarios, the Cook–Torrance GGX model exhibited an approximately constant GPU frame time of 1.6 ms.

For the first scenario (Figure 9, left), the user has an overall view of all showcased objects. At this distance, the diffraction pattern has already converged to the mean value, which results in only one evaluation of the noise function. Here, the GPU frame time overhead is minimal (+0.2 ms). In the second scenario (Figure 9, right), the user performs a close-up inspection of a single object. The diffraction pattern is

	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5
α	0.392237	0.303354	0.242714	0.220624	0.140337
η	Al	Al	Al	Al	Al
w	2.339413	2.792546	3.554197	4.059328	6.4455
h	0.001025	0.000713	0.000767	0.000973	0.000778
var_R	0.004296	0.005652	0.006584	0.005495	0.004950
var_G	0.003961	0.005369	0.006312	0.005117	0.004861
var_B	0.003261	0.004574	0.005349	0.003978	0.003840
$covar_{RG}$	0.003773	0.004949	0.00581	0.004817	0.004593
$covar_{RB}$	0.003057	0.003895	0.004511	0.003723	0.003655
$covar_{GB}$	0.003170	0.004203	0.004842	0.003922	0.003861

Table 1. Used model parameters for the aluminum Samples 1–5. The covariance matrices are calculated from the macro photos, and the remaining values are provided by Clausen et al. [2023].

now fully visible. This leads to a significant number of fragments where the noise function is evaluated either once (magnification), multiple times (minification during convergence), or not at all (minification after convergence). The GPU frame time overhead increases moderately (+1.6 ms). While the first two scenarios reflect practical use cases, we also constructed a theoretical worst-case scenario.[4] We rendered a full-screen quad with the diffraction pattern, where each fragment is in the convergence phase of the minification process. Here, our model introduced a significant GPU frame time overhead compared to the Cook–Torrance GGX model (+6.0 ms).

6. Conclusion and Future Work

When illuminating a rough surface with sharp, direct light, diffraction causes spatially varying speckle patterns and an on-average reddish and bluish appearance in the forward and backscattering, respectively. We introduced a practical real-time BRDF model based on the model by Clausen et al. [2023] addressing these two diffraction phenomena. We first transferred their model from the spectral to the RGB domain without introducing perceivable errors. We further extended it by introducing spatially varying speckle patterns. We approximate the speckle pattern's color distribution with a multivariate normal distribution, its spatial correlation with a simplex noise function, and its dependency on the incidence and reflection directions with a covariance model, which is fully defined with a single covariance matrix acquired in the mirror direction. To evaluate our model in virtual reality, we integrated it into the Unity game engine. We considered the speckle patterns' highlight disparity and removed aliasing artifacts.

The color shifts in the forward and backscattering and the dynamic speckle patterns lead to a lively appearance. Further, rough conductive surfaces appear with speckle patterns and highlight disparity, seeming more metallic and rougher. Overall, our model results in a more realistic appearance than the popular Cook–Torrance model with a marginal increase in computational costs.

For future work, it would be beneficial to improve the shift function provided by Clausen et al. [2023]. As mentioned by the authors, the cosine function struggles to approximate the steep increase of the slope distribution and is limited to a certain roughness range. Given that our covariance model relies on the shift function, it encounters similar challenges in approximating the steep increase of the covariance values and in accurately representing covariance values for smooth surfaces. Furthermore, it would be interesting to improve Gustavson's GPU implementation of 4D simplex noise. In comparison to the Python OpenSimplex noise function, it is not ideally normally distributed and appears blocky under very close inspection.

[4] Under the assumption of a pipeline with an early-z test, where only visible fragments are shaded.

Acknowledgements

We thank the anonymous reviewers for their detailed reviews. This work was partially funded by the Ministry of Culture and Science of the State of North Rhine-Westphalia in the *Zukunftsfonds* program as part of the project *Games Technology Network*.

Index of Supplemental Materials

A virtual reality demo and the source code of our presented technique can be found at https://jcgt.org/published/0013/01/01/Realtime-Diffraction-PBR-1_0.zip.

A video can be found at https://jcgt.org/published/0013/01/01/diffraction_speckle_JCGT.mp4.

References

BECKMANN, P., AND SPIZZICHINO, A. 1987. *The Scattering of Electromagnetic Waves from Rough Surfaces*. Norwood. URL: https://www.google.com/books/edition/_/AzO7QgAACAAJ. 4

BLAKE, A., AND BÜLTHOFF, H. 1990. Does the brain know the physics of specular reflection? *Nature 343*, 6254, 165–168. URL: https://doi.org/10.1038/343165a0, doi:10.1038/343165a0. 19

CHURCH, E. L., JENKINSON, H. A., AND ZAVADA, J. M. 1979. Relationship between surface scattering and microtopographic features. *Optical Engineering 18*, 2, 182125. URL: https://doi.org/10.1117/12.7972337, doi:10.1117/12.7972337. 4

CLAUSEN, O., MARROQUIM, R., AND FUHRMANN, A. 2018. Acquisition and validation of spectral ground truth data for predictive rendering of rough surfaces. *Computer Graphics Forum 37*, 4, 1–12. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13470, doi:10.1111/cgf.13470. 3

CLAUSEN, O., CHEN, Y., FUHRMANN, A., AND MARROQUIM, R. 2023. Investigation and simulation of diffraction on rough surfaces. *Computer Graphics Forum 42*, 1, 245–260. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.14717, doi:10.1111/cgf.14717. 1, 2, 3, 5, 8, 12, 13, 14, 20, 22, 23

COOK, R. L., AND TORRANCE, K. E. 1982. A reflectance model for computer graphics. *ACM Transactions on Graphics 1*, 1 (Jan), 7–24. URL: https://doi.org/10.1145/357290.357293, doi:10.1145/357290.357293. 6

DONG, Z., WALTER, B., MARSCHNER, S., AND GREENBERG, D. P. 2016. Predicting appearance from measured microgeometry of metal surfaces. *ACM Transactions on Graphics 35*, 1 (Dec), 9:1–9:13. URL: https://doi.org/10.1145/2815618, doi:10.1145/2815618. 4

GOODMAN, J. 2020. *Speckle Phenomena in Optics: Theory and Applications*. Press Monographs. SPIE Press. URL: https://books.google.com.pe/books?id=3FvmyAEACAAJ. 8

HE, X. D., TORRANCE, K. E., SILLION, F. X., AND GREENBERG, D. P. 1991. A comprehensive physical model for light reflection. *ACM SIGGRAPH Computer Graphics 25*, 4 (Jul), 175–186. URL: https://doi.org/10.1145/127719.122738, doi:10.1145/127719.122738. 4

HOLZSCHUCH, N., AND PACANOWSKI, R. 2017. A two-scale microfacet reflectance model combining reflection and diffraction. *ACM Transactions on Graphics 36*, 4 (Jul), 66:1–66:12. URL: https://doi.org/10.1145/3072959.3073621, doi:10.1145/3072959.3073621. 3, 4

KRYWONOS, A., HARVEY, J. E., AND CHOI, N. 2011. Linear systems formulation of scattering theory for rough surfaces with arbitrary incident and scattering angles. *Journal of the Optical Society of America A 28*, 6 (Jun), 1121–1138. URL: https://opg.optica.org/josaa/abstract.cfm?URI=josaa-28-6-1121, doi:10.1364/JOSAA.28.001121. 4

LÖW, J., KRONANDER, J., YNNERMAN, A., AND UNGER, J. 2012. BRDF models for accurate and efficient rendering of glossy surfaces. *ACM Transactions on Graphics 31*, 1 (Feb), 9:1–9:14. URL: https://doi.org/10.1145/2077341.2077350, doi:10.1145/2077341.2077350. 4

MATUSIK, W., PFISTER, H., BRAND, M., AND MCMILLAN, L. 2003. A data-driven reflectance model. *ACM Transactions on Graphics 22*, 3 (Jul), 759–769. URL: https://doi.org/10.1145/882262.882343, doi:10.1145/882262.882343. 4

NGAN, A., DURAND, F., AND MATUSIK, W. 2005. Experimental analysis of BRDF models. In *Eurographics Symposium on Rendering (2005)*, The Eurographics Association, K. Bala and P. Dutre, Eds., 117–126˙ URL: https://dx.doi.org/10.2312/EGWR/EGSR05/117-126, doi:10.2312/EGWR/EGSR05/117-126. 4

PERLIN, K. 2002. Improving noise. In *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques*, ACM Press, SIGGRAPH '02, 681–682. URL: https://doi.org/10.1145/566570.566636, doi:10.1145/566570.566636. 11

RICE, S. O. 1951. Reflection of electromagnetic waves from slightly rough surfaces. *Communications on Pure and Applied Mathematics 4*, 2–3, 351–378. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/cpa.3160040206, doi:10.1002/cpa.3160040206. 4

SCHRÖDER, S., DUPARRÉ, A., CORIAND, L., TÜNNERMANN, A., PENALVER, D. H., AND HARVEY, J. E. 2011. Modeling of light scattering in different regimes of surface roughness. *Optics Express 19*, 10 (May), 9820–9835. URL: http://opg.optica.org/oe/abstract.cfm?URI=oe-19-10-9820, doi:10.1364/OE.19.009820. 4

VELINOV, Z., WERNER, S., AND HULLIN, M. B. 2018. Real-time rendering of wave-optical effects on scratched surfaces. *Computer Graphics Forum 37*, 2, 123–134. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13347, doi:10.1111/cgf.13347. 5

WALTER, B., MARSCHNER, S. R., LI, H., AND TORRANCE, K. E. 2007. Microfacet models for refraction through rough surfaces. In *Proceedings of the 18th Eurographics Conference on Rendering Techniques*, Eurographics Association, EGSR'07, 195–206. URL: http://dx.doi.org/10.2312/EGWR/EGSR07/195-206. 14

WERNER, S., VELINOV, Z., JAKOB, W., AND HULLIN, M. B. 2017. Scratch iridescence: Wave-optical rendering of diffractive surface structure. *ACM Transactions on Graphics 36*, 6 (Nov), 207:1–207:14. URL: https://doi.org/10.1145/3130800.3130840, doi:10.1145/3130800.3130840. 4

XIA, M., WALTER, B., HERY, C., MAURY, O., MICHIELSSEN, E., AND MARSCHNER, S. 2023. A practical wave optics reflection model for hair and fur. *ACM Transactions on Graphics 42*, 4 (Jul), 39:1–39:15. URL: https://doi.org/10.1145/3592446, doi:10.1145/3592446. 5

YAN, L.-Q., HAŠAN, M., WALTER, B., MARSCHNER, S., AND RAMAMOORTHI, R. 2018. Rendering specular microgeometry with wave optics. *ACM Transactions on Graphics 37*, 4 (Jul), 75:1–75:10. URL: https://doi.org/10.1145/3197517.3201351, doi:10.1145/3197517.3201351. 3, 5, 15

Author Contact Information

Olaf Clausen
Computer Graphics Group
TH Köln
Betzdorfer Str. 2
Köln, 50679 Germany
olafclausen@posteo.de

Martin Mišiak
Lehrstuhl für Mensch-Computer-Interaktion
University of Würzburg
Emil-Fischer-Straße 50
Würzburg, 97074 Germany
martin.misiak.89@gmail.com

Arnulph Fuhrmann
Computer Graphics Group
TH Köln
Betzdorfer Str. 2
Köln, 50679 Germany
arnulph.fuhrmann@th-koeln.de

Ricardo Marroquim
Computer Graphics and Visualization
Delft University of Technology
Van Mourik Broekmanweg 6
Delft, 2628XE The Netherlands
r.marroquim@tudelft.nl

Marc Erich Latoschik
Lehrstuhl für Mensch-Computer-Interaktion
University of Würzburg
Emil-Fischer-Straße 50
Würzburg, 97074 Germany
marc.latoschik@uni-wuerzburg.de

Olaf Clausen, Martin Mišiak, Arnulph Fuhrmann, Ricardo Marroquim, and Marc Erich Latoschik, A Practical Real-Time Model for Diffraction on Rough Surfaces, *Journal of Computer Graphics Techniques (JCGT)*, vol. 13, no. 1, 1–27, 2024
http://jcgt.org/published/0013/01/01/

Received:	2023-11-09		
Recommended:	2024-01-17	Corresponding Editor:	Eric Haines
Published:	2024-06-26	Editor-in-Chief:	Marc Olano

© 2024 Olaf Clausen, Martin Mišiak, Arnulph Fuhrmann, Ricardo Marroquim, and Marc Erich Latoschik (the Authors).
The Authors provide this document (the Work) under the Creative Commons CC BY-ND 3.0 license available online at http://creativecommons.org/licenses/by-nd/3.0/. The Authors further grant permission for reuse of images and text from the first page of the Work, provided that the reuse is for the purpose of promoting and/or summarizing the Work in scholarly venues and that any reuse is accompanied by a scientific citation to the Work.

