Efficient Rendering of Layered Materials using an Atomic Decomposition with Statistical Operators

Laurent Belcour

Efficient Rendering of Layered Materials using an Atomic Decomposition with Statistical Operators

LAURENT BELCOUR, Unity Technologies

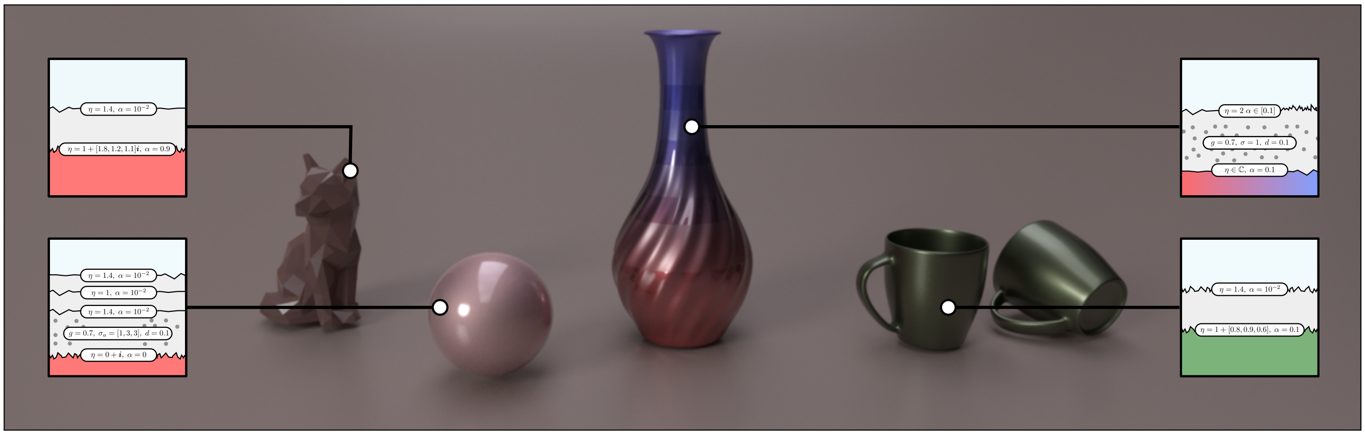

$\eta = 1.4,\ \alpha = 10^{-2}$
$\eta = 1 + [1.8, 1.2, 1.1]\boldsymbol{i},\ \alpha = 0.9$
$\eta = 1.4,\ \alpha = 10^{-2}$
$\eta = 1,\ \alpha = 10^{-2}$
$\eta = 1.4,\ \alpha = 10^{-2}$
$g = 0.7,\ \sigma_a = [1,3,3],\ d = 0.1$
$\eta = 0 + \boldsymbol{i},\ \alpha = 0$
$\eta = 2\ \alpha \in [0.1]$
$g = 0.7,\ \sigma = 1,\ d = 0.1$
$\eta \in \mathbb{C},\ \alpha = 0.1$
$\eta = 1.4,\ \alpha = 10^{-2}$
$\eta = 1 + [0.8, 0.9, 0.6],\ \alpha = 0.1$

Fig. 1. We introduce a BSDF model to render plane-parallel layered materials using an analysis of the directional statistics of light interaction with microfacet geometry and participating media. Our model closely matches the reference and supports an arbitrary number of textured layers while being energy conserving, free from heavy per-material precomputation, and compatible with real-time constraints.

We derive a novel framework for the efficient analysis and computation of light transport within layered materials. Our derivation consists of two steps. First, we decompose light transport into a set of atomic operators that act on its directional statistics. Specifically, our operators consist of reflection, refraction, scattering, and absorption, whose combinations are sufficient to describe the statistics of light scattering multiple times within layered structures. We show that the first three directional moments (energy, mean and variance) already provide an accurate summary. Second, we extend the adding-doubling method to support arbitrary combinations of such operators efficiently. During shading, we map the directional moments to BSDF lobes. We validate that the resulting BSDF closely matches the ground truth in a lightweight and efficient form. Unlike previous methods, we support an arbitrary number of textured layers, and demonstrate a practical and accurate rendering of layered materials with both an offline and real-time implementation that are free from per-material precomputation.

CCS Concepts: •**Computing methodologies** →**Reflectance modeling;**

Additional Key Words and Phrases: Layered Materials, Statistical Analysis, Real-Time Rendering

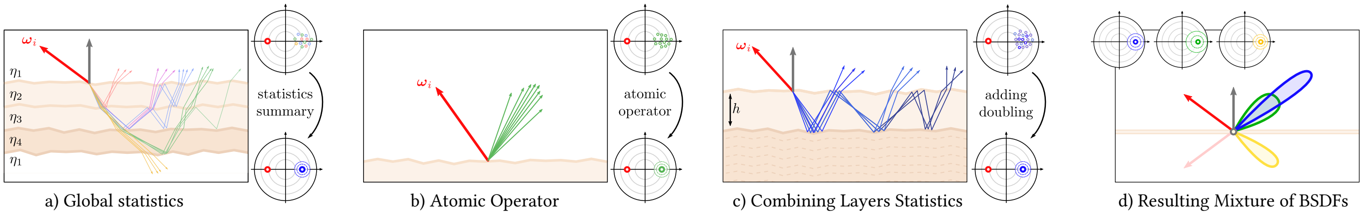
ACM Reference format:
Laurent Belcour. 2018. Efficient Rendering of Layered Materials
using an Atomic Decomposition with Statistical Operators. *ACM Trans. Graph.* 37, 4, Article 73 (August 2018), 15 pages.
DOI: 10.1145/3197517.3201289

1 INTRODUCTION

Elaborate shading models allow the realistic reproduction of the diversity and complexity of surface appearance found in nature or man-made objects. A simple way to enrich a set of materials is to coat them with one or many strata of glazing or clear coat. This is typically what happens when objects are wet or paints are glazed. For example, applying a clear coat on a rough metal will create a distinctive *hazy* look. Unfortunately, coatings increase the difficulty of modeling the resulting appearance and design efficient shading models for rendering. Up to now, no one has provided a comprehensive model of the *Bidirectional Scattering Distribution Function* (BSDF) of coated materials. In this work, we focus on a subset of coatings called *layered materials* [Jakob et al. 2014].

Layered Materials. Layered materials assume that the coating layers are plane parallel with respect to the shading normal and separated by rough interfaces. It is also assumed that geometrical optics fully describes the appearance. Thus, the thicknesses of layers is larger than the wavelengths of visible light (removing wave optics from our study). However, the number of layers can be arbitrary and thus light can bounce many times before exiting the material, which makes evaluation difficult. Offline productions [Hery et al. 2017; Kulla and Conty 2017; Langlands 2014] and the real-time community [Drobot 2017] have expressed interest in those appearance. However, they limit themselves to only two layers evaluated with a single reflection per layer (that is paths directly reflected by the top layer (denoted R) and paths transmitted by the top layer and reflected by the base layer (denoted TRT)). Thus, more general – yet efficient – models are needed by the community.

0730-0301/2018/8-ART73 $15.00
DOI: 10.1145/3197517.3201289



a) Global statistics

b) Atomic Operator on Statistics

c) Combining Layers Statistics

d) Resulting Mixture of BSDFs

Fig. 2. We express the directional statistics (energy, mean and variance) of a layered BSDF in the projected plane (a). Instead of computing the complete transport, we track statistical summary at each step. We first study the impact on directional statistics of different *atomic* operations, *i.e.* reflection, refraction, scattering, and absorption. For example, rough reflection increases the variance of the incident radiance (b). We evaluate multiple scattering between layers by combining those atomic operators with a new *adding-doubling* algorithm working on those statistics (c). Finally, we instantiate multiple BRDF lobes from those statistics to approximate the entire layered BSDF (d).

Technical Difficulties. A key difficulty is to provide a realistic model that works with an *arbitrary number of layers* (possibly textured), accounts for *multiple scattering*, is *energy conserving*, requires *little storage*, has a *short precomputation time*, supports good *importance sampling* and is *symmetric with respect to light transport* evaluation (to be compatible with bidirectional rendering techniques). In the pursuit of some of those goals, researchers often rely on heavy precomputation and/or tabulate much of the data [Jakob et al. 2014] to faithfully reproduce the target appearance or devise approximative light transport schemes [Weidlich and Wilkie 2007] that unfortunately miss some important effects. The main objective of our work is to reach a quality comparable to tabulated methods, while providing a lightweight solution that is efficient like the approximative methods. While we are inspired by how Jakob et al. [2014] compute multiple scattering between layers, we use a lighter representation. Our insight is that light interaction with individual layers is simple and often results in only stretches, compressions, and blurs of the directional distribution of light. It is the combination of interactions in the structure that creates complex appearance. Thus, our idea is to work at the level of individual layers where light-matter interaction is described with simple transport operators (reflection, refraction, absorption, and scattering) to keep derivation tractable and to use an efficient combination algorithm to build complex scattering functions.

We make the following contributions:

- We introduce a statistical framework to estimate the *energy, mean, and variance* of the BSDF for any layered material configuration (Fig. 2(a)). Our framework builds upon the statistics of projected directions (Section 3)
- We provide a set of atomic operators (reflection, refraction, absorption, and scattering) within our framework and derive close approximations of light interaction with interfaces or media (Section 4 and Fig. 2(b)).
- We derive a new method, similar to the one of Jakob et al. [2014], that works on statistics and allows us to efficiently evaluate the directional statistics due to multiple scattering in the layered structure (Section 5, and Fig. 2(c)).
- Using our framework, we implement a new BSDF models for offline and real-time rendering of layered materials that is energy conserving, requires little storage, is free from per-material precomputation, and has good importance sampling (Section 6, and Fig. 2(d)). Our offline BSDF model is also symmetric.

Assumptions & Limitations. In this work, we assume that all interfaces are composed of specular microfacets following the GGX distribution. Due to our approximations, our model accurately reproduces the ground truth for low and moderate roughness only (see Section 7.3). Our current formulation does not support other surface models, e.g. Lambertian diffuse layers.

2 PREVIOUS WORK

Specialized models. Many models were derived for a specific configuration of a layered structure. For example, Dorsey and Hanrahan [1996] introduced a model for aging of metals, Jensen et al. [1999] derived a model accounting for a water layer on top of objects, Stam [2001] derived a BSDF model for human skin by layering a dielectric surface on a participating medium, Ershov et al. [2001] derived a model of flakes in a coating, Dai et al. [2009] modeled the refraction of light by two parallel rough microfacets, *etc.* While efficient, those models can be difficult to extend beyond their original purpose. In contrast, we propose a general formulation, that is not restricted to a particular layered configuration.

Spatial Diffusion. It is also possible to model the diffusion of light in highly diffusing layered materials [Donner and Jensen 2005]. This usually results in a *Bidirectional Subsurface Scattering Reflectance Distribution Function* (BSSRDF) model as the spatial diffusion of light in the medium is accounted for. We do not try to reproduce this effect and restrict ourselves to the case in which the approximation that light enters and exits the layered structure at the same position is valid.

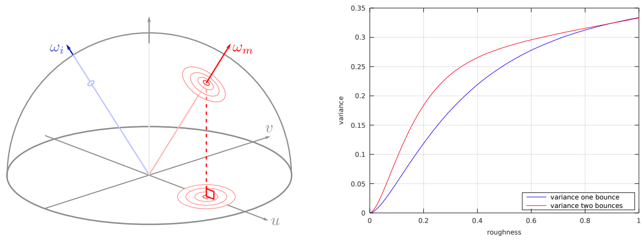
General models. Some models reproduce the appearance of stratified materials without the *explicit* evaluation of all light transport. Weidlich and Wilkie [2007] used stratified microfacets models [Walter et al. 2007] to model transmissive rough interfaces but simplified the transport in the layered structure by refracting query rays with respect to a single microfacet. This unfortunately fails to correctly *capture the blur due to a stack of layers* as shown in Fig. 13. Jakob et al. [2014] introduced a numerical scheme to compute and store a tabulated representation of layered BRDFs that accounts for multiple scattering between layers using the *adding-doubling* method [van de Hulst 1980]. However, it is impractical for production scenarios or real-time rendering. One of its drawbacks is that it is *not memory bound* as it relies on a Fourier decomposition of the BSDFs. Consequently, a single BSDF can weight more than a gigabyte in their framework.

Real-time models. To gain efficiency, it is possible to restrict layered BSDFs to two rough interfaces and approximate the resulting transport by two BSDF lobes at the shading point (denoted bi-model here). A first BSDF lobe accounts for the first interface (or clear-coat) and the second lobe accounts for the remaining interfaces. For example, Elek [2010] uses Weidlich and Wilkie [2007] microfacet interfaces with further simplifications. Blur induced by the clear-coat is approximated by taking the max roughness. Guo et al. [2016] use two von Mises-Fisher (vMF) lobes at the surface of the object. They approximate the convolution of vMF lobes to model the impact of the clear-coat. vMF lobes are unlikely to be applied in production as they miss the heavy tails required to model metals. Furthermore, specific precomputations are required (e.g. precomputed environment maps and area lights) for real-time rendering. Similarly, De Rousiers et al. [2011] approximated the refraction of two successive rough interfaces using vMF lobes, sharing the same drawbacks.

Summary. In this work, we provide a new model for layered materials that permits real-time usage but without sacrificing much of the fidelity with respect to the reference light transport. We were inspired by the use of the adding-doubling algorithm [Jakob et al. 2014] but we forbade the use of any parameter-dependent or per-material time-consuming tabulation. Instead, we developed a statistical analysis of light interaction with the layered structure and tailored the adding-doubling method to track the energy, mean, and variance of the BSDF. To do so, we model the interactions of light in a layered structure as operators on statistics and apply the adding-doubling method to efficiently evaluate multiple scattering. Thanks to all this, we apply our solution to an arbitrary number of textured layers and show that, for low and moderate roughnesses, our solution is close to the reference appearance and energy conserving, but at the same time lightweight and efficient. We further show how to efficiently importance sample our model and how to make it symmetric. We demonstrate our model tailored to both offline and real-time scenarios.

3 DIRECTIONAL STATISTICS

Our insight is to estimate the directional statistics of a layered material for a given incidence and to inject those statistics into a BSDF for efficient shading. In this work, we restrict our statistical study to the *energy* e, the *mean* μ and the *variance* σ of a BSDF lobe[1]. We focus on the GGX microfacet model [Walter et al. 2007], but our ideas apply to other BSDF models. We build on the property that, for isotropic microfacet BSDF models like GGX or Beckmann, the bijective mapping from the parameters of the BSDF to its directional statistics enables us to instantiate a BSDF lobe from the knowledge of its energy, mean, and variance.



a) Projective space

b) Variance w/r roughness

Fig. 3. Our statistical analysis is done on the projective plane of directions $[u, v] = [\omega_x, \omega_y]$ (left inset). We study the mean and variance of the outgoing distribution in this space. We show in the right inset the variance of a GGX lobe at normal incidence with respect to roughness (in blue) as well as the variance of light bouncing twice on parallel surfaces with the same roughness (in red).

Statistics in the Projected Plane. To compute directional mean and variance, we need a parametrization. We use the orthographic projection of 3D coordinates of directions to the (x, y) plane (see Fig. 3). That is, we compute the mean and variance with respect to $[u, v\ ] = [\omega_x, \omega_y]$, where a_x is the x-component of vector $\mathbf{a}$. We experienced that in this space, an isotropic GGX peak is close to a circular shape. Löw et al. [2012, see Fig. 5] reported a similar behavior in the MERL database. We will use this property and assume that the projected BSDF is a radially symmetric function. Thus, we only need to track a 1D variance. Note that for grazing angles, GGX is notably skewed. However, since GGX lobes are parametrized by an energy, an incident direction and a roughness, we have no control over the higher order statistics of this distribution.

Equivalent Roughness. A BSDF lobe is not stable under convolution. Thus, the statistics specify an *equivalent* lobe that is an approximation since the shape of the spherical function is not preserved. In the following, we will display and use the *equivalent roughness* of a directional statistic using the mapping between roughness and variance and its inverse (see Fig. 3(b)).

4 ATOMIC OPERATORS ON STATISTICS

In this section, we describe statistical atomic operators approximating the effect of specific light transport interactions. Those operators will later be used to evaluate BSDF lobe statistics (Section 5) and instantiate an equivalent BSDF model (Section 6). The different operators we use are summarized in Table 1. We will discuss them in the following order: reflection by a rough surface (Section 4.1), refraction by a rough surface (Section 4.2), transmission and scattering in plane-parallel participating media (Section 4.3 and 4.4).

4.1 Reflection by a Rough Interface

A rough surface distributes the incoming light based on its *Bidirectional Reflectance Distribution Function* (BRDF). For a microfacet

[1] We deliberately omit the square on σ for better readability.

Table 1. We summarize the different atomic operators and how we approximate the outgoing energy e, mean μ, and variance σ given the incident energy e_i, mean μ_i, and variance σ_i. For each statistic, we indicate with an exponent whether it has to be used for a reflected or transmitted lobe (for example σ^R for the variance of a reflected lobe). This notation will be reused in the adding-doubling method. In this table, we use the convention that incident and outgoing vectors leave the surface. This explains why both reflection and refraction negate the mean.

	Rough Reflection	**Rough Refraction**	**Absorption**	**Forward Scattering**
energy	$e^R = e_i \times \mathrm{FGD}^\infty$	$e^T = e_i \times [1 - \mathrm{FGD}^\infty]$	$e^T = e_i \exp\left[-\frac{\sigma_t h}{\sqrt{1-\lvert\mu_i\rvert^2}}\right]$	$e^T = e_i\, [\frac{\sigma_s h}{\sqrt{1-\lvert\mu_i\rvert^2}}] \exp\left[-\frac{\sigma_t h}{\sqrt{1-\lvert\mu_i\rvert^2}}\right]$
mean	$\mu^R = -\mu_i$	$\mu^T = -\eta_{12}\,\mu_i$	$\mu^T = -\mu_i$	$\mu^T = -\mu_i$
variance	$\sigma^R = \sigma_i + \sigma_{12}^R$	$\sigma^T = \frac{\sigma_i}{\eta_{12}} + f(s \times \alpha_{12})$	$\sigma^T = \sigma_i$	$\sigma^T = \sigma_i + \sigma_g$

BRDF, it can be described by:

$$L_o(\boldsymbol{\omega}_o) = \int_\Omega F(\langle \boldsymbol{\omega}_i, \mathbf{h}\rangle)\, \frac{G(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o)\, D(\mathbf{h})}{4\langle \boldsymbol{\omega}_i, \mathbf{n}\rangle\langle \boldsymbol{\omega}_o, \mathbf{n}\rangle} L_i(\boldsymbol{\omega}_i)\langle \boldsymbol{\omega}_i, \mathbf{n}\rangle\, \mathrm{d}\boldsymbol{\omega}_i, \quad (1)$$

where L_o (resp. L_i) is the outgoing (resp. incoming) radiance, Ω is the upper hemisphere, $\mathbf{h} = (\omega_i+\omega_o)/||\omega_i+\omega_o||$ is the half-vector, $\langle \boldsymbol{\omega}_o, \mathbf{n}\rangle$ denotes the dot product, and F, G, and D describe the BRDF: F is the Fresnel term, D the *Normal Distribution Function* (NDF) and G the associated shadowing/masking.

Energy. The amount of energy reflected by the surface is called the *directional albedo*. For microfacet models, it is directly defined by the integrated Fresnel FGD:

$$\mathrm{FGD} = \int_\Omega F(\langle \boldsymbol{\omega}_i, \mathbf{h}\rangle)\, \frac{G(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o)\, D(\mathbf{h})}{4\langle \boldsymbol{\omega}_i, \mathbf{n}\rangle\langle \boldsymbol{\omega}_o, \mathbf{n}\rangle} \langle \boldsymbol{\omega}_i, \mathbf{n}\rangle\, \mathrm{d}\boldsymbol{\omega}_o \quad (2)$$

We will omit all parameters for $\mathrm{FGD} = \mathrm{FGD}(\boldsymbol{\omega}_i, \alpha, \eta + \mathrm{i}\kappa)$, where $\eta + \mathrm{i}\kappa$ is the complex *index of refraction* (IOR) at the interface. For isotropic BRDFs, the elevation is sufficient to parametrize the incoming direction. We approximate the energy of the outgoing radiance by decoupling FGD from incident radiance:

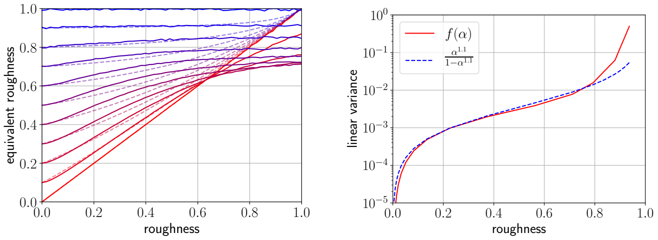
$$\int_\Omega L_o(\omega_o)\mathrm{d}\omega_o \;\simeq\; \mathrm{FGD} \times \int_\Omega L_i(\omega_i)\mathrm{d}\omega_i. \quad (3)$$

Since no closed form exists for FGD, we precompute it in a 4D table parametrized by an elevation, a roughness, and a complex index of refraction.

Mean. Largarde and De Rousiers showed that the mean of the reflected lobe is shifted towards the normal [Lagarde and De Rousiers 2014, Section 3.1.4]. This is typically used to fetch preintegrated environment maps in video-game engines. Note that this shift is directly integrated into the FGD texture. They showed that a good approximation of this shift is:

$$\boldsymbol{\omega}'_{r\,|xy} = \beta \mathbf{n}_{|xy} + (1-\beta)\boldsymbol{\omega}_{r\,|xy}, \quad (4)$$

where $\boldsymbol{\omega}_{|xy} = [\omega.x, \omega.y]$ is the vector in the tangential plane, and β depends on the shadowing-masking term (see Lagarde and De Rousiers [2014] for details). However, this *shift* is only important for rough configurations and we neglect it during the estimation of the BSDF. It can be applied later on, during shading, to correct misalignment of the preintegrated environment maps.



a) 2-bounce equivalent roughness

b) Linear space transformation

Fig. 4. We plot the equivalent roughness of two bounces of light on parallel layers with different micro-geometries using a normal incidence $|\mu_i| = 0$. The first layer is depicted with color, from red $\alpha = 0$ to blue $\alpha = 1$, and the second layer is specified by the abscissa. The equivalent roughness does not linearly depend on the input roughnesses. We found a transform (b) mapping roughness to a space of linear variance for reflection. We show our approximation in dashed.

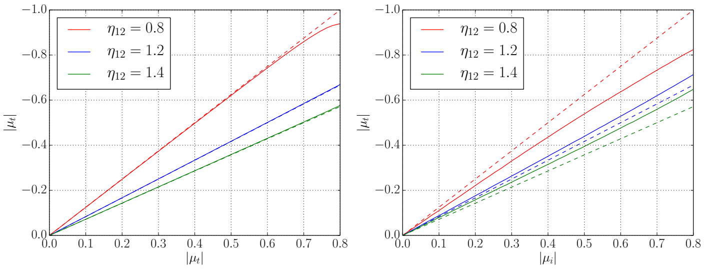
Variance. A rough surface increases the variance of the incident distribution [Durand et al. 2005]. However, this increase is not linear with respect to roughness. It is approximately linear with respect to variance for small roughnesses ($\alpha < 0.2$). For higher roughnesses, we can find a space where rough reflections have a linear behavior (see Appendix A). That is, the variance of the light bouncing twice on parallel surfaces with roughnesses α_1 and α_2 is:

$$\sigma_{12} = f(\alpha_1) + f(\alpha_2), \quad (5)$$

where $f(\alpha)$ is the transform of roughness to variance. We display this transformation in Fig. 4(b). We convert roughness to linear variance using:

$$\sigma = f(a) \approx \frac{a^{1.1}}{(1-a^{1.1})}, \quad (6)$$

and from linear variance to roughness using the inverse. This transform does not perfectly fit the linear-space mapping, but it approximates the outgoing directional statistics of two light bounces on



a) $\alpha = 0.1$

b) $\alpha = 0.6$

Fig. 5. We display the mean of the transmitted lobe $\mu_t = \sin(\theta_t)$ with respect to the incident direction (specified by $|\mu_i| = |\sin(\theta_i)|$) for different refractive index. We also display (in dashed) the refracted direction $\sin(\theta_r)$. We can see very little deviation of the mean transmitted direction from the refracted direction for small roughnesses (a), but a noticeable one for higher roughnesses (b).

microfacet interfaces of moderate roughnesses (below 0.4) well, as shown in Fig. 4(a). An interactive demo of this approximation using a two bounce setup is available in our supplemental material[2].

Energy Conservation. The classical microfacet theory only accounts for single scattering in the microgeometry and thus omits the energy due to multiple bounces in the microgeometry. According to Heitz et al. [2016], a scaling factor of the directional albedo approximates multiple scattering well. Thus, it should have no effect on the mean and variance and only affect the energy term. We incorporate it directly into the FGD term during precomputation. For a rough interface, we replace the previously defined FGD term by FGD^∞ which is the result of the stochastic evaluation of Heitz et al.'s model. This new table has the same 4D parametrization.

Summary. Given an incident directional function with energy e_i, mean μ_i and variance σ_i, the reflected energy e_R, mean μ_R, and variance σ_R are approximated by:

$$e^R = e_i \times \text{FGD}^\infty \tag{7}$$

$$\mu^R = -\mu_i \tag{8}$$

$$\sigma^R = \sigma_i + f(\alpha) \tag{9}$$

4.2 Refraction by a Rough Interface

A rough dielectric interface distributes light in both the upper and lower hemisphere of local directions. The later is described by the BTDF and, in the context of microfacet theory, its definition is similar to the BRDF [Walter et al. 2007]. In the following, we will note $\eta_{12} = \frac{\eta_1}{\eta_2}$ the ratio of refractive indices of the two media.

Energy. The energy scaling of an incident angular field by a rough dielectric interface is given by $1 - \text{FGD}^\infty$. This also ensures energy conservation between the reflected and refracted directional fields.

Mean. For rough interfaces, the transmitted lobe peak is not the refracted incident direction. We found that, as with off-specular reflection, the mean refracted direction is roughness dependent (see Fig. 5). Its direction also depends on the refractive index. At grazing angles, and for higher refractive index ratio, the peak of the lobe differs more from the refracted incident direction. However, we empirically found that using the purely refracted direction was sufficient (see our supplemental material).

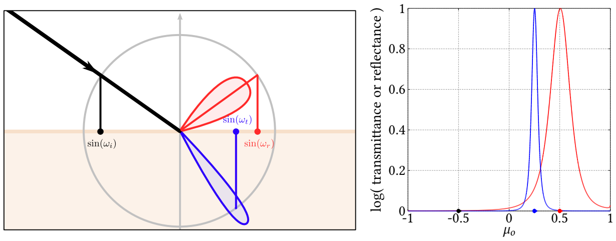
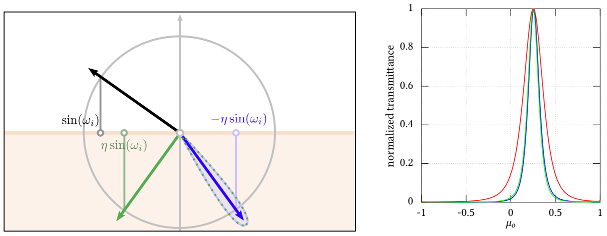


Fig. 6. The transmitted lobe (in blue) has, for the same roughness, a different angular width compared to the reflected lobe (in red). We analyze them in the scattering plane, parametrized by the sine of the outgoing angle for an incident angle of 30 degrees (c). We scaled the reflectance and transmittance to one to compare the width of both lobes.



a) Fake transmission

b) Comparison, $\mu_i = \frac{1}{2}$

Fig. 7. To compute the variance we fake rough refraction using a rough reflection with a new incidence direction $\mu_r = \eta_{12}\,\mu_i$ (left, in green). The *transmitted roughness* we use is $\alpha_r = s \times \alpha$, where α is the roughness of the microgeometry. We show that our fake refraction (left, in green) closely matches the reference curve (left, in blue), while the one of Kulla and Conty [2017] (left, in red) overestimates it.

Variance. Due to Snell's law, light rays are bent when passing through a dielectric interface. Likewise, the incident variance is scaled by the ratio of the index of refraction when transmitted. As in the reflection case, this scaled incident variance is increased by the roughness of the surface. We derive this property by considering transmission like a reflection. For the same roughness and incident direction, the refracted and transmitted lobes have different widths (see Fig. 6). We use the notion of *transmitted roughness* to analyze the variance of the transmitted lobe. The idea is to fake the transmitted lobe using a reflection from underneath the interface (see Fig. 7 (a) in green). To match the reference transmitted lobe, we scale the roughness of our fake transmission. For stretch-invariant microfacet distributions, the scale s is the ratio of the derivatives of the transmitted and reflected half vector's tangents:

$$s = \frac{1}{2}\left[1 + \eta_{12}\frac{\langle \boldsymbol{\omega}_i \cdot \mathbf{n}\rangle}{\langle \boldsymbol{\omega}_t \cdot \mathbf{n}\rangle}\right]. \tag{10}$$

We detail its derivation in Appendix B.

Kulla and Conty [2017] derived a similar scaling factor by equating the fake reflected and transmitted lobe peaks but it overestimates

[2] See webgl/validate_reflection.html

the transmitted roughness. Guo et al. [2016] approximated a von Mises-Fisher exponent similar to our equivalent roughness. However, neither of them provide a close match to the reference BSDF. We compare our scaling factor to the one of Kulla and Conty [2017] in Fig. 7 (b). An interactive demo of the off-specular transmitted direction and the different scaling factors is available in our supplemental material[3].

Summary. Given an incident directional function with energy e_i, mean μ_i and variance σ_i, the transmitted energy e_T, mean μ_T, and variance σ_T are approximated by:

$$e^T = e_i \times (1 - \mathrm{FGD}^{\infty}) \tag{11}$$

$$\mu^T = -\eta_{12}\, \mu_i \tag{12}$$

$$\sigma^T = \frac{\sigma_i}{\eta_{12}} + f(s \times \alpha) \tag{13}$$

4.3 Volume Absorption

Assuming that the interface between the medium and the outside (be it air or another medium) is already resolved (as discussed in the previous sections), we can study the changes in energy, mean and variance due to a participating media. In this Section, we cover how *absorption* affects statistics. The impact of *scattering* on statistics will be covered in Section 4.4. As in Monte-Carlo rendering, we treat them separately and combine them together afterwards.

Energy. The attenuation of the incident light-field by a random medium is described by Beer-Lambert's law. It states that the incident radiance is scaled down depending on the optical depth:

$$L_o(\boldsymbol{\omega}_o) \;=\; \exp\left[-\sigma_t \frac{h}{\langle \boldsymbol{\omega}_o, \mathbf{n}\rangle}\right] L_i(\boldsymbol{\omega}_o), \tag{14}$$

where σ_t is the transmittance cross-section and h is the depth of the layer. We approximate the average energy using the incident mean attenuation: $e_o = e_i \exp\left(-\frac{\sigma_t h}{\langle \boldsymbol{\omega}_o, \mathbf{n}\rangle}\right)$.

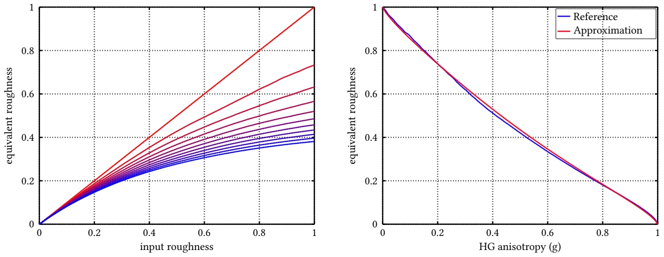
Mean and Variance. Since the optical depth between the bottom and the top layers depends on the incident direction, the mean and variance are also affected (see Fig. 8(a) for the variance). We empirically found out that the impact of absorption on the mean and variance is negligible. An interactive demo of this impact and of our approximation is available in our supplemental material[4].

Summary. Given an incident directional function with energy e_i, mean μ_i and variance σ_i, we approximate the energy e_T, mean μ_T, variance σ_T transmitted by volumetric absorption by:

$$e^T = e_i \exp\left(-\frac{\sigma_t h}{\langle \boldsymbol{\omega}_o, \mathbf{n}\rangle}\right) \tag{15}$$

$$\mu^T = -\mu_i \tag{16}$$

$$\sigma^R = \sigma_i \tag{17}$$



a) Absorption

b) Scattering with HG

Fig. 8. We display the equivalent roughness of both absorption (a) and scattering (b) of a participating layer with the Henyey-Greenstein (HG) phase function. For the absorption, we used varying optical depths (from $h = 0$ in red to $h = 1$ in blue). For both plots, we vary the anisotropy parameter $g \in [0, 1]$. Our approximation of the HG roughness (in red) closely matches the reference curve (in blue).

4.4 Volume Scattering

In this section, we restrict ourselves to the study of an optically thin homogeneous slab of height h that does not emit light. In such a case, single scattering is predominant and we can neglect multiple scattering. Given the light incident to the slab $L_i(\boldsymbol{\omega})$, the amount of light scattered by the medium is [Pharr and Humphreys 2010]:

$$L_o(\boldsymbol{\omega}) \;=\; \sigma_s \int_0^h \exp\left[-\frac{\sigma_t\; t}{\langle \boldsymbol{\omega}, \mathbf{n}\rangle}\right] L_s(t, \boldsymbol{\omega})\mathrm{d}t, \tag{18}$$

$$\text{with } L_s(t, \boldsymbol{\omega}) = \int_{S^2} p(\boldsymbol{\omega}_i, \boldsymbol{\omega}) \exp\left[-\frac{\sigma_t\,(h-t)}{\langle \boldsymbol{\omega}_i, \mathbf{n}\rangle}\right] L_i(\boldsymbol{\omega}_i)\mathrm{d}\boldsymbol{\omega}_i, \tag{19}$$

where σ_s is the scattering cross-section.

Energy and Mean. Assuming that the phase function is strongly forward scattering (most of its energy is scattered forward), we approximate the outgoing energy using the attenuation evaluated in the mean direction of the incoming light:

$$\int_\Omega L_o(\boldsymbol{\omega})\mathrm{d}\boldsymbol{\omega} \;\approx\; \sigma_s\; h \exp\left[-\frac{\sigma_t h}{\langle \boldsymbol{\omega}_o, \mathbf{n}\rangle}\right] \int_\Omega L_i(\boldsymbol{\omega})\mathrm{d}\boldsymbol{\omega}. \tag{20}$$

This approximation assumes that the phase function does not "lose" energy in the backward directions. However, it is possible to account for backscattering by modulating this energy by the amount of light the phase function scatters forward. For $g \geq 0.7$, this scaling is unnecessary. As for the absorption, we empirically found that scattering did not alter the mean significantly for forward phase functions.

Variance. As for rough interfaces, the incident variance increases by the width of the phase function in the forward direction Belcour et al. [2014]. We empirically found a fit of the equivalent GGX roughness for the Henyey-Greenstein phase function (HG) in the forward direction (see Fig. 8). It is given by:

$$\sigma_g = \left[\frac{1-g}{g}\right]^{0.8} \frac{1}{1+g}, \tag{21}$$

where g is HG's anisotropy factor. However, the Henyey-Greenstein phase function can have a non-negligible backscattering when $g \ll 0.7$. This creates two modes in directional statistics that need to be

[3] See webgl/validate_transmission.html

[4] See webgl/validate_absorption.html

resolved separately in our methodology. We focused on forward scattering only media and left the derivation of the backscattering variance to future work. An interactive demo of the impact of HG's g factor on the angular statistics and of our different approximations is available in our supplemental material[5].

Summary. Given an incident directional function with energy e_i, mean μ_i and variance σ_i, the transmitted energy e_T, mean μ_T, variance σ_T from volumetric scattering are approximated by:

$$e^T = e_i \frac{\sigma_s h}{\langle \boldsymbol{\omega}_o, \mathbf{n} \rangle} \exp\left(-\frac{\sigma_t h}{\langle \boldsymbol{\omega}_o, \mathbf{n} \rangle}\right) \tag{22}$$

$$\mu^T = -\mu_i \tag{23}$$

$$\sigma^R = \sigma_i + \sigma_g \tag{24}$$

5 STATISTICS WITH MULTIPLE LAYERS

The outgoing light distribution aggregates many light paths (e.g. TRT and $TRRRT$ paths). So far, our statistical analysis models groups of paths undergoing similar transport (e.g. all TRT paths). Thus, we need to combine the statistics of those path groups in an efficient way. For that, we rely on the adding-doubling algorithm. We first apply it to the case of energy (Section 5.1). Then, we apply it to compute the variance (Section 5.2). Since the mean approximately aligns with the reflected or refracted direction, it is preserved after multiple bounces and we do not treat it.

5.1 The Adding-Doubling Method

The Adding-Doubling method (see Grant and Hunt [1969] for a complete overview) allows up to express radiative transfer in plane parallel media using a discrete form. The idea is to model transmission and reflection due to a thin homogeneous slab with linear operators and to combine them to estimate the transmission and reflection of a thick heterogeneous slab. We recall here its mathematical principle in a simple form. For that, we will assume that the interacting interfaces are purely smooth.

Adding-doubling requires the definition of the reflection $\mathbf{r}$ and transmission $\mathbf{t}$ coefficients of radiance with respect to the light's direction of propagation: $\mathbf{r}_{k\,k+1}$ is the reflection coefficient of the slab for light propagating downward and $\mathbf{r}_{k+1\,k}$ is the reflection coefficient of the slab for light propagating upward. With no light source in the interface, the light exiting an interface upward can be expressed as (see Fig. 9):

$$l_o^+ = \mathbf{r}_{12} l_i^- + \mathbf{t}_{21} l_i^+, \tag{25}$$

reads: *light propagating upward from the slab is the incident light propagating downward reflected by the slab, plus the incident light propagating upward transmitted by the slab.* In the same fashion, we can express the light propagating downward from the slab as:

$$l_o^- = \mathbf{r}_{21} l_i^+ + \mathbf{t}_{12} l_i^- . \tag{26}$$

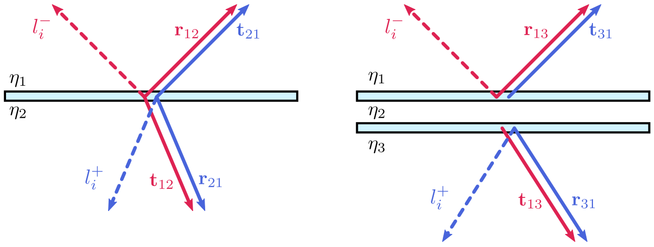


Fig. 9. Given the transmittance and reflectance $\mathbf{r}_{12}$, $\mathbf{t}_{12}$, $\mathbf{r}_{21}$, and $\mathbf{t}_{21}$ of a layer (left, between two media, η_1 and η_2). The adding-doubling method permits to calculate the scattering properties of a stack of layers (right), accounting for the inter-reflection between them (that is all TR^+T paths).

We can express the global upward reflectance summarizing light transport between two layers $\mathbf{r}_{12}$ and $\mathbf{r}_{23}$ as:

$$\begin{aligned} \mathbf{r}_{13} &= \mathbf{r}_{12} + \mathbf{t}_{12}\mathbf{r}_{23}\mathbf{t}_{21} + \mathbf{t}_{12}\mathbf{r}_{23}^2\mathbf{r}_{21}\mathbf{t}_{21} + \cdots \\ &= \mathbf{r}_{12} + \sum_{k=0}^{\infty} \mathbf{t}_{12}\mathbf{r}_{23}^{k+1}\mathbf{r}_{21}^{k}\mathbf{t}_{21}. \end{aligned} \tag{27}$$

This form accounts for all bounces between the two layers. It can be summarized using the analytic form of this arithmetic series:

$$\mathbf{r}_{13} = \mathbf{r}_{12} + \frac{\mathbf{t}_{12}\mathbf{r}_{23}\mathbf{t}_{21}}{1 - \mathbf{r}_{23}\mathbf{r}_{21}}. \tag{28}$$

Similarly, we can express the global upward transmittance as:

$$\mathbf{t}_{31} = \frac{\mathbf{t}_{32}\mathbf{t}_{21}}{1 - \mathbf{r}_{23}\mathbf{r}_{21}}. \tag{29}$$

We can as well express the downward reflectance and transmittance as:

$$\mathbf{r}_{31} = \mathbf{r}_{32} + \frac{\mathbf{t}_{32}\mathbf{r}_{21}\mathbf{t}_{23}}{1 - \mathbf{r}_{23}\mathbf{r}_{21}} \tag{30}$$

$$\mathbf{t}_{13} = \frac{\mathbf{t}_{12}\mathbf{t}_{23}}{1 - \mathbf{r}_{23}\mathbf{r}_{21}}. \tag{31}$$

The *adding* algorithm iteratively expresses the reflectance and transmittance of an increasing number of interfaces by using $\mathbf{r}_{13}$, $\mathbf{t}_{31}$, $\mathbf{r}_{31}$, and $\mathbf{t}_{31}$ as the reflectance and transmittance of a virtual interface in Equations 28 to 31.

This method can also evaluate multiple scattering in a homogeneous medium of any depth with the *doubling* algorithm. Here, we stack together the same layer to generate a virtual layer of twice the depth. That is, we apply Equations 28 to 31 with $\mathbf{r}_{12} = \mathbf{r}_{23}$, $\mathbf{t}_{12} = \mathbf{t}_{23}$, $\mathbf{r}_{21} = \mathbf{r}_{32}$, and $\mathbf{t}_{21} = \mathbf{t}_{32}$. By iterating this operation on an input layer of very small size (typically $h = 10^{-8}$), we can increase the depth of a layer to any size.

Using adding-doubling, we can now evaluate the directional albedo of the layered BSDF. This is done by replacing the different $\mathbf{r}$ and $\mathbf{t}$ terms with the reflectance and transmittance terms studied in Section 4. For a rough surface, we have: $\mathbf{r}_{12} = \mathrm{FGD}^{\infty}$, $\mathbf{t}_{12} = [1 - \mathrm{FGD}^{\infty}]$, $\mathbf{r}_{21} = \mathrm{FGD}^{\infty}$, and $\mathbf{t}_{21} = [1 - \mathrm{FGD}^{\infty}]$.

[5] See `webgl/validate_scattering.html`

Total Internal Reflection. While the defined adding-doubling algorithm permits approximating multiple scattering between two layers, it partially misses the impact of the totally reflected light by the upper Fresnel interfaces. In the case of a smooth interface, there are angular configurations where no light is refracted and the dielectric interface behaves like a pure mirror. This behavior is called *Total Internal Reflection* (TIR). Transmission $\mathbf{t}_{21}$ does not account for the angular spread of light reflected on interface 2 ⟶ 3 and we need to bring this information back:

$$L_t(\boldsymbol{\omega}_o) = \int_\Omega L_i(\boldsymbol{\omega}_i)\, T(\boldsymbol{\omega}_t)\, \rho(\boldsymbol{\omega}_i, \boldsymbol{\omega}_t, \alpha_{23})\, \mathrm{d}\boldsymbol{\omega}_t, \tag{32}$$

where $\boldsymbol{\omega}_t = \mathrm{refract}(\boldsymbol{\omega}_o, \eta_{12})$, $T(\boldsymbol{\omega}_t) = 1 - F(\boldsymbol{\omega}_t)$ is the Fresnel transmittance, and $\rho(\boldsymbol{\omega}_i, \boldsymbol{\omega}_t, \alpha_{23})$ is the microfacet BRDF of layer interface 2 ⟶ 3. When TIR occurs, $T(\boldsymbol{\omega}_i) = 0$ for $\boldsymbol{\omega}_i$ outside of the extinction cone. A solution to compute the amount of energy lost due to TIR is to decouple its computation from the transport integral:

$$L_t(\boldsymbol{\omega}_o) = \int_\Omega L_i(\boldsymbol{\omega}_i)\, \rho(\boldsymbol{\omega}_i, \boldsymbol{\omega}_t, \alpha_{23})\, \mathrm{d}\boldsymbol{\omega}_t \times \underbrace{\int_\Omega (1 - F(\boldsymbol{\omega}_t))\, D(\boldsymbol{\omega}_t, \alpha_{23})\, \mathrm{d}\boldsymbol{\omega}_i}_{=\mathrm{TIR}}, \tag{33}$$

where $D(\boldsymbol{\omega}_i, \alpha_{23})$ is the NDF. The second integral of this product is similar to the FGD table. We thus precompute this term in a 3D table. We show in our supplemental material[6] that this decoupling gives good results in general. During the adding-doubling, we replace $\mathbf{r}_{21}$, and $\mathbf{t}_{21}$ by:

$$\mathbf{r}_{21} \longleftarrow \mathbf{r}_{21} + (1 - \mathrm{TIR}) \times \mathbf{t}_{21} \tag{34}$$

$$\mathbf{t}_{21} \longleftarrow \mathrm{TIR} \times \mathbf{t}_{21} \tag{35}$$

5.2 Adding-Doubling for Variance

We reuse the idea of adding-doubling for the case of variance. As seen in Sections 4.1 and 4.4, the variance of the interaction with an interface or medium has an affine form. Also, the variance of a weighted sum of distributions sharing the same mean is the weighted arithmetic mean of the individual variances. Those properties permit the use of the adding-doubling methodology to compute variance due to multiple scattering between layers.

Multiple Scattering. Given two interfaces, the unnormalized average variance $\tilde{\sigma}^R_{13}$ accounting for multiple scattering is:

$$\tilde{\sigma}^R_{13} = \mathbf{r}_{12}\, \sigma^R_{12} + \sum_{k=0}^{+\infty} \left[\mathbf{t}_{12}\, \mathbf{r}_{23}^{k+1}\, \mathbf{r}_{21}^{k}\, \mathbf{t}_{21}\right] \left[\sigma^T_{12} + \Big((k+1)\, s^R_{23} + k\, \sigma^R_{21}\Big) J_{21} + \sigma^T_{21}\right], \tag{36}$$

where s^R_{ij} (resp. s^T_{ij}) is the additional variance when reflecting (resp. transmitting) on a rough interface between indices of refraction i and j, and J_{ij} is the transmission scaling factor. This formula can be separated into a geometric series and an arithmetico-geometric series. The geometric series has the analytic form we saw for the classical adding-doubling. The arithmetico-geometric series has the following analytic form [Riley et al. 2010]:

$$\sum_{k=0}^{+\infty} k r^k = \frac{r}{(1-r)^2} \qquad \text{with } r \in [0, 1[. \tag{37}$$

Since $\mathbf{r}_{23}\mathbf{r}_{21} < 1$ (or no light can enter the layer), Equation 36 simplifies to:

$$\tilde{\sigma}^R_{13} = \mathbf{r}_{12}\, \sigma^R_{12} + \left[\frac{\mathbf{t}_{12}\, \mathbf{r}_{23}\, \mathbf{t}_{21}}{1 - \mathbf{r}_{23}\mathbf{r}_{21}}\right] \left[\sigma^T_{12} + \sigma^T_{21} + J_{21}\left(\sigma^R_{23} + \frac{\mathbf{r}_{23}\mathbf{r}_{21}}{1 - \mathbf{r}_{23}\mathbf{r}_{21}} \sigma^R_{21}\right)\right]. \tag{38}$$

Similar forms can be derived for the transmitted average variance. We detail them in Appendix C. They will be used in the next iteration of the adding algorithm in place of σ^R_{12} and σ^T_{12}. For the case of participating media, we use $\sigma^R_{12} = 0$, $\sigma^R_{21} = 0$, $\sigma^T_{12} = \sigma_g$, $\sigma^T_{21} = \sigma_g$, $J_{12} = 1$, and $J_{21} = 1$ (see Equation 21 for the definition of σ_g).

Adding-Doubling. Using the multiple scattering equations, we can build an adding-doubling algorithm. We start with an empty layer, that is $e^R_{ij} = e^R_{ji} = 0$, $e^T_{ij} = e^T_{ji} = 1$, and $\sigma^R_{ij} = \sigma^R_{ji} = \sigma^T_{ij} = \sigma^T_{ji} = 0$. For each interface in the stack (starting from the upper one to the lower one) we apply the multiple scattering equations on energies and variances (e.g. Equation 38 and 28 for the upper reflected distribution). We also need to track the refraction scaling factors J_{ij}, and J_{ji} during the process.

Limitations. Note that those derivations assume that all lobes share approximately the same mean direction. Consequently, the described adding-doubling method cannot work with multimodal BSDFs without special-case handling. For example, to handle retro-reflective phase functions (e.g. HG with $g < 0$) we need to track another set of statistics in the retro-reflective direction.

6 IMPLEMENTATION

Based on our findings, we developed two BSDF models: one tailored for forward path tracing and real-time rendering, denoted the Forward model (Section 6.1); and one tailored for bidirectional light transport algorithms, denoted the Symmetric model (Section 6.2).

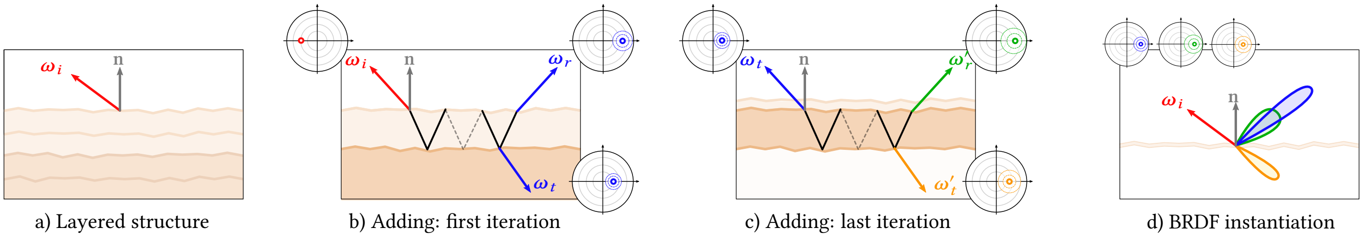
6.1 Forward Model

As illustrated in Fig. 10, to evaluate the BSDF, we start from a unit energy $e_i = 1$, the incident mean $\mu_i = [\,\boldsymbol{\omega}_{i\,|x}, \boldsymbol{\omega}_{i\,|y}\,]$, a zero variance $\sigma_i = 0$, and evaluate the adding-doubling method to gather intermediate variance (See Appendix C) and energy in a vector of BRDF lobes[7]. For the transmitted lobe, we record the transmitted statistics at the end of the adding-doubling algorithm[8] (Equation 49). Each entry in the vector corresponds to a BSDF lobe and contains its energy e_k, mean μ_k, and variance σ_k. From this vector of statistics, we instantiate an approximate BSDF model consisting of a weighted sum of microfacet GGX lobes with Fresnel e_k, incident direction reflect(μ_k), and roughness $f^{-1}(\sigma_k)$. We use our fake refraction

[6] See `webgl/compute_TIR.html`

[7] see our code `mitsuba/plugins/layered_forward.cpp`, line 105

[8] see our code `mitsuba/plugins/layered_dielectric.cpp`, line 258



a) Layered structure b) Adding: first iteration c) Adding: last iteration d) BRDF instantiation

Fig. 10. For shading, we track the BRDF lobe statistics starting from the incident directions (a). We estimate iteratively a vector of outgoing lobes using the adding-doubling algorithm (b-c). Finally, we instantiate a set of BRDF lobes using the vector of statistics and the inverse mapping. This process outputs as many lobes as there are layers. It is possible to merge statistics together to reduce the number of BRDF instances.

model for the transmitted lobe. Thus the reflection and transmission models are:

$$\rho(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o) = \sum_{k=0}^{N} e_k \rho_k(\boldsymbol{\omega}_k, \boldsymbol{\omega}_o, \alpha_k), \tag{39}$$

$$\text{with} \begin{cases} \alpha_k = f^{-1}(\sigma_k) \\ \boldsymbol{\omega}_k = \text{reflect}(\mu_k) \\ \rho_k(\boldsymbol{\omega}_k, \boldsymbol{\omega}_o, \alpha_k) = \dfrac{D(\mathbf{h})G(\boldsymbol{\omega}_k, \boldsymbol{\omega}_o)}{4\langle \boldsymbol{\omega}_k, \mathbf{n}\rangle \langle \boldsymbol{\omega}_o, \mathbf{n}\rangle} \end{cases}$$

where e_k, $\boldsymbol{\omega}_k$ and σ_k are the energy, mean, and variance for the k^{th} lobe respectively. For better performance, it is possible to merge some of the variances (scaled by energies and then normalized by the total energy). In our implementation, we kept all lobes as-is.

Importance Sampling & MIS. Since our models are a weighted average of multiple lobes, we randomly select one based on the energy and importance sample the visible normals [Heitz and d'Eon 2014] based on the fake incident direction and roughness. However, this strategy is not optimal and creates fireflies since the different lobes overlap. To avoid this we use multiple importance sampling on the different lobes with the balance heuristic [Veach 1998]. The contribution of a BRDF lobe sample becomes[9]:

$$p = \frac{e_{all}}{\sum_{i=0}^{N} e_i p_i} \sum_{i=0}^{N} e_i \rho_i(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o, \alpha_i), \tag{40}$$

where $\boldsymbol{\omega}_o$ is the outgoing direction selected by one of the strategies, e_i is the energy for the i^{th} lobe, e_{all} is the total energy, p_i is the *probability density function* (pdf) of sampling the vNDF for roughness α_i, and ρ_i is the pdf of the i^{th} microfacet model.

Real-Time Model. For real-time scenarios, we reuse the Forward model, restricted to two or three lobes. Since, in all real-time engines, lights such as area lights and environment maps are preintegrated with respect to a GGX model, we evaluate our model in this context.

6.2 Symmetric Model

For offline scenarios where symmetry of light transport matters (such as BDPT or MLT), we build an ad-hoc symmetrization from the Forward model. As Weidlich and Wilkie [2007], we assume that the incoming and outgoing directions share the same microfacet. We further use the property that the refracted rays will share the same half-vector (see Appendix E for proof). It is thus possible to use our method with respect to the difference vector θ_d. This virtually injects the adding-doubling roughness on the half-vector normal. The resulting statistics might not match the Forward model, since the FGD term is evaluated using $\cos(\theta_d)$ and not $\cos(\theta_i)$ and the transmitted roughnesses will slightly differ. This doesn't however introduce an over-blur of the resulting lobe, and we found that the results of the Symmetric model were often closer to the reference (see our supplemental material). The Symmetric model uses the same importance sampling strategy as the Forward model.

7 RESULTS

In this section, we demonstrate the use of our different models in both offline (Section 7.1) and real-time (Section 7.2) contexts. All results are computed on a 16 core i7 processor with 32GB of RAM and an Nvidia GTX 980 graphics card. All computations are using parallel computing as much as possible. For our offline method, we precomputed the FGD using the complex index of refraction, resulting in a 4D table, precomputed at a resolution of 64^4 (making it 64MB in size). We used linear interpolation to evaluate it. For the real-time method, we used the available FGD 2D texture that performs a split sum of Schlick's Fresnel [Karis 2013]. We also precomputed the 3D TIR table using a 64^3 resolution, making it 1MB in size. Unless noted, we use the Symmetric model for our offline results.

7.1 Offline Rendering

Validation. We validated our method against a stochastic ground truth. See Appendix D and our supplemental material for more details. We show that our model captures the color saturation due to multiple scattering (Fig. 11) and is energy conserving (Fig. 12). Thanks to the adding method, we do not lose energy due to multiple scattering between layers. Even though the reflectance and transmittance terms are approximate, our model is close to the saturated color of the reference. We also show in the Frosted Metal scene (Fig. 13) that we correctly predict the rough appearance of frosted paints. Here, we replicate a figure of Weidlich and Wilkie [2007]: a rough dielectric interface on top of a smooth conducting one. The method of Weidlich and Wilkie [2007] fails to propagate

[9] see our code `mitsuba/plugins/layered_forward.cpp`, line 453

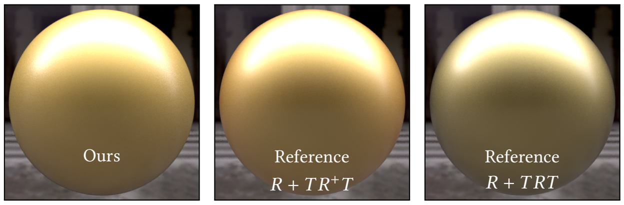


Fig. 11. Thanks to the adding algorithm, we estimate the color saturation due to multiple scattering between interfaces. Here we show our method (left) compared to two references: one computing all bounces between interfaces (middle) and one computing only one reflection per layer (right).

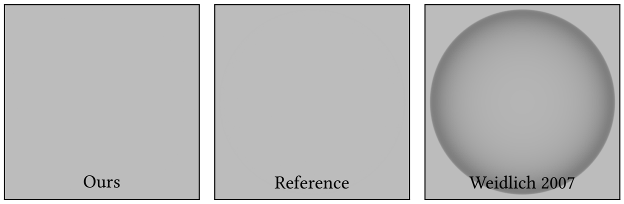


Fig. 12. Our model is energy conserving (as long as the FGD texture is). In this scene, we show that for a complex layered structure with five dielectric layers (alternating $\eta = 1.3$ and $\eta = 1$) on top of a pure mirror, our model does not lose energy as opposed to the one of Weidlich and Wilkie [2007].

roughness between layers: the base layer still behaving as a mirror on the global reflectance. Note that Elek [2010] mitigates this issue by using the maximum of roughness of the top and bottom layers. Please refer to our supplemental document for more comparisons with ground truth and the model of Weidlich and Wilkie [2007]

In the Dragon scene (Fig. 14), we replicated a figure from Jakob et al. [2014] to show how our method handles participating media. The dragon figure consists of a golden base on top of which we apply a diffusing medium with increasing optical depth. We used a forward phase function $g = 0.7$, and used $\sigma_a = 0$ and $\sigma_s = 1$. Thanks to our statistical analysis, we easily predict the diffusion and attenuation of light rays inside this structure. However, to simplify our implementation, we only account for single scattering within media layers (see Section 7.3).

In the Plates (Fig. 15) and Robot (Fig. 16) scenes, we display textured assets. In Plates, we use a single texture for the roughness of some of the layers and in Robot we combined three textures together. Compared to Jakob et al. [2014], our method is not limited by precomputation of discrete values of the parameters and resolves a BSDF on the fly. Consequently, our method handles efficiently any textured combination of inputs for both surface and media layers. For example, in the Robot scene, three parameters are textured. To illustrate, the method of Jakob et al. [2014] requires 1.7GB of storage for the Metallic Paint example alone and its computation takes 21min and requires up to 15.5GB of RAM. Texturing this asset would require an enormous amount of disk storage for the different BSDF files and they might not even fit in memory for rendering.

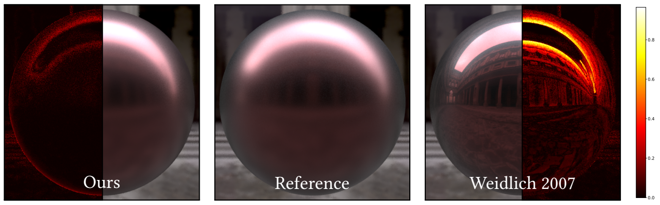


Fig. 13. We reproduce the Frosted Metal example from Weidlich and Wilkie [2007]. While Weidlich and Wilkie [2007] misses the correlation between layers, we stay close to the reference image computed using a stochastic evaluation of the layered structure.

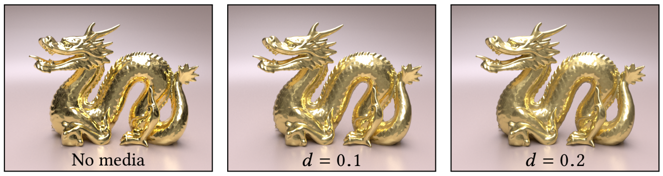


Fig. 14. The Dragon scene illustrates the ability of our method to add absorption and single scattering. Here, we start from a specular gold dragon and add a layer of scattering particles on top (with $g = 0.7$). As we increase the optical depth of this layer, the dragon appears rougher and darker due to scattering and absorption.

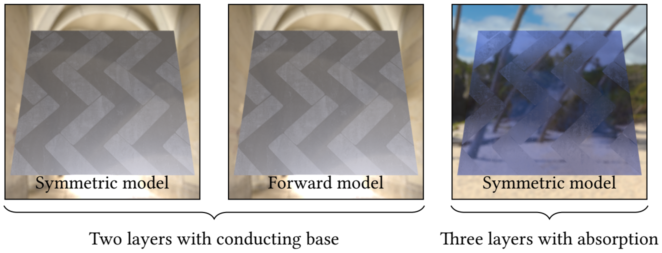


Fig. 15. The Plates scene illustrates our ability to work with textured assets. In the first example, the plate is composed of a two layers material with $\eta_0 = 2, \alpha_0 = 0$ for the first layer and $\eta_1 = 1 + 0.5\mathbf{i}, \alpha_1 \in [0, 1]$ for the second layer. We display side-by-side our symmetric and forward model. In the second example, the plate is composed of three layers with $\eta_0 = 1.8, \alpha_0 \in [0, 1], d_1 = 1, \sigma_a = [1, 0.7, 0.2]$, and $\eta_2 = 1, \alpha_0 =\in [0, 1]$. We used a specialized plugin for dielectrics since Mitsuba separates the two cases.

Peformance. We evaluated the performance of our code base. We report the timings of our offline method compared to the method of Weidlich and Wilkie [2007] in Table 2. Our method's costs are primarily due to the adding-doubling calculation and to the linear interpolation of tables. The latter cost is alleviated on graphics hardware.

7.2 Real-Time Rendering

We implemented our real-time BRDF model inside a commercial 3D engine using forward rendering to validate the practicability of our implementation. Note that our model applies to deferred rendering as well. In our implementation, we restricted the number

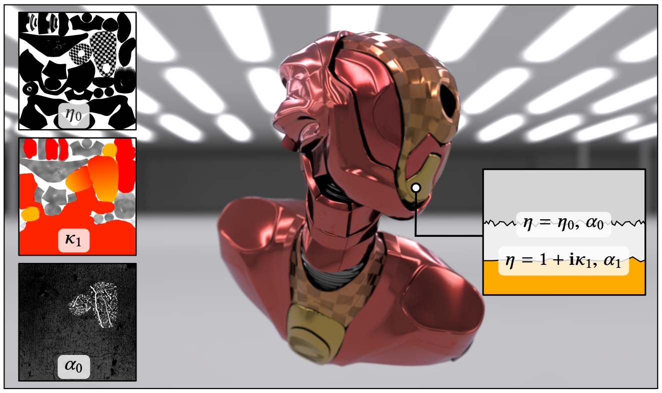


Fig. 16. The ROBOT scene illustrates the ability of our method to efficiently work with textured assets. In this scene, we used a 2-layer structure where η_0, α_0, and κ_1 are texture mapped. We set $\eta_1 = 1$ and $\alpha_1 = 0$.

of layers to three, and outgoing lobes to two. We also fixed the second interface to be a participating medium. We used the same adding code for the Mitsuba implementation and for the real-time rendering one. We show a comparison between our real-time implementation, our Forward model, and the reference in Fig. 17. Our real-time implementation matches the reference appearance albeit with the technical limitations of using a simple Fresnel form, and environment map pre-integration. We measure the running time of our shader using a full-screen quad at a resolution of 1920 × 1080 pixels. Here, the full frame takes between 1.9ms to 2.1ms to render. We compare this running time to the standard shader of the engine. In this case, the full frame takes between 1.7ms to 2.0ms to render.

The BEETLE scene (Fig.18) demonstrates our real-time implementation. It depicts a car fully covered with our layered BRDF model. We rendered it at a resolution of 1920×1080 during a material design session that we show in our supplemental video.

7.3 Limitations & Failure Cases

Accuracy with respect to roughness. Our model approximates multiple scattering lobes using equivalent GGX models. While this faithfully predicts the behavior of layered materials for relatively smooth surfaces, it becomes increasingly inaccurate when most of the layers are very rough. We illustrate the lost of accuracy of our model in Fig. 19. Here we can see that for multiple layers with roughnesses close to one, our model significantly diverges from the ground truth.

Table 2. We report the cost of our method compared to the one of Weidlich and Wilkie [2007] for three scene: FROSTED METAL (Fig. 13, at 1024 spp), WHITE FURNACE (Fig. 12, at 1024 spp), and DRAGON (Fig. 14, at 512 spp). FROSTED METAL and WHITE FURNACE use the direct integrator while DRAGON uses the path integrator. We report the timings of our method without linear interpolation of the FGD and TIR texture in parenthesis.

	Fig. 13	Fig. 12	Fig. 14
Ours	46s (24s)	1.8m (35s)	1.84m (1.64m)
Weidlich [2007]	17s	20s	1.60m

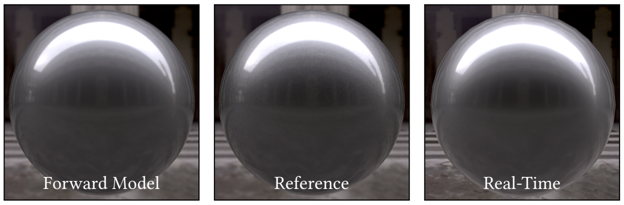


Fig. 17. Our real-time implementation matches the offline appearance for this coated metallic sphere with $\eta_0 = 1.4$, $\alpha = 0.01$, and $\eta_1 = 1 + \boldsymbol{i}$, $\alpha = 0.1$. For example, we can observe brighter grazing angles for the real-time implementation due to the split-sum FGD texture. Also, we used a thin-lens for the Mitsuba renderer that we could not exactly match with the real-time camera.

Fig. 18. The BEETLE showcase our real-time implementation of the Forward model inside a commercial 3D engine.

Color shift. Because of total internal reflection, the color of the multiple scattering between layers might not be the same as the transmitted color. This happens for metals with a strong color shift at grazing angles. In such a case, our method will not be able to reproduce the exact tint of the material. This is noticeable in the FROSTED METAL example (Fig. 22) where at grazing angles the base layer takes on a green tint that our model misses.

Multiple scattering in media. Our prototypes are currently limited to single scattering from participating media. This is a design choice to keep the implementation simple to read. Accounting for multiple scattering would require to add another direction in the adding-doubling algorithm, *i.e.* the retro-reflective direction. This would allow us to track scattering between media interfaces and to add the missing retro-reflective look of dust.

Interfaces. Due to the derivation of the linear space for GGX, our model only accounts for a single type of NDF for interfaces. We cannot accurately predict the case of multiple interfaces with mixed NDF geometries (such as Beckmann [Walter et al. 2007], GTR [Burley 2012], Student-t [Ribardière et al. 2017], *etc.*). Note, however, that a linear space can be derived for any NDF as long as all interfaces are described with such an NDF. Similarly, we cannot accurately reproduce the effect of a Lambertian layer. This is due to

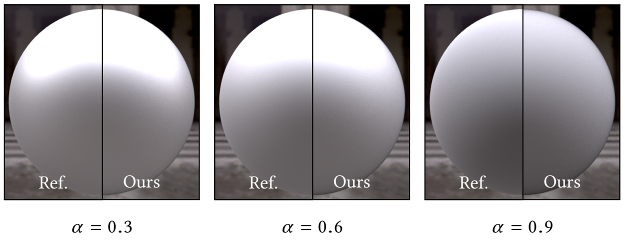


Fig. 19. For higher roughnesses, our model lose its accuracy compared to a stochastic ground truth. In this example, we use a rough dielectric layer, $\eta = 2$, on top of a rough conducting base, $\eta = 0.01 + \mathbf{i}$, with equal roughnesses $\alpha_1 = \alpha_2 = \alpha$. Here, we did not account for multiple scattering in the microfacet geometry and isolate the error due to approximating multiple scattering between layers.

the fact that the resulting distribution will always be a GGX lobe. Another strong limitation that is shared by all the works on layered materials is that we cannot work with non-parallel interfaces. While using a different normal to evaluate Fresnel for the base and coat layers is typically done in production and works with our method, it is incorrect with respect to the resolution of multiple scattering.

Skewness at grazing angles. Due to the extinction cutoff of the Fresnel transmission, transmitted lobes at grazing angles are anisotropic and skewed. Since we used radially symmetric NDFs for our analysis, we cannot model this behavior correctly.

8 DISCUSSION & FUTURE WORK

Complex Appearance. While we restricted ourselves to geometrical optics for our study by enforcing the thickness of layers to be greater than the visible light wavelengths, it is possible to incorporate iridescence effects into our model. Since we rely on the evaluation of the integrated Fresnel term during the adding-doubling, we can use any model that replaces the Fresnel term. For example, it is possible to replace one of the evaluations of the FGD term by the model of Belcour and Barla [2017]. Similarly, we could incorporate glittery appearance into our framework by using one of the available glint models [Jakob et al. 2014]. Note that in such a case, the correlation between the discrete evaluation of glints and the upper layers' Fresnel will be missed.

Anisotropy & Spatial Diffusion. While we restricted our analysis to isotropic GGX BSDFs, our analysis could be applied to track anisotropic lobes. Here, the tracking of variance in the adding-doubling algorithm would require tracking eight scalars instead of four. It would be interesting to see if the statistical analysis can be performed on the spatial components as well, in order to track a subsurface scattering profile.

Other Applications. Since our model is fast to evaluate and doesn't require a per-material pre-computation, it could be used to perform nonlinear fitting of data. Also, it would be interesting to see if our layered materials model can be used for inverse design. There, artists could design the desired look using many lobes and extract a set of layers with specified index of refraction and roughnesses. This could be interesting for the design and manufacture of real life objects.

9 CONCLUSION

We introduced a novel statistical analysis of layered materials that builds on an atomic decomposition of light transport to track the energy, mean and variance of the layered BSDF. Our analysis is versatile and can account for both surface and volumetric light-matter interaction. Furthermore, we leveraged the power of the adding-doubling method on this statistical representation to compute the layered BSDF accounting for multiple scattering in the layered structure. We demonstrated the applicability and accuracy of our representation with offline and real-time implementations showing an *efficient*, *energy conserving* and *symmetric* BSDF model for layered materials. Our model still requires precomputing a 4D and a 3D table. But, given the regularity of this data, it seems possible to find good and efficient approximations that would make our model much faster. Although expanding our model to account for anisotropy or spatial diffusion is still open, we believe that our method is a first step towards efficient and lightweight BSDF/BSSRDF models for layered materials.

ACKNOWLEDGEMENTS

The author thank Pascal Barla, Emmanuel Turquin, Simon Premože, and Romain Pacanowski for their comments during the early stage of writing, Kenneth Vanhoey for the video voice-over, and Stephen Hill and Derek Nowrouzezahrai for their English proof-reading. The Robot, Teacup and Bottle models were designed by MasterXeon1001, Optic, and LSFMT from www.blendswap.com respectively. The Fox model is designed by Slavik from www.thingiverse.com.

REFERENCES

Laurent Belcour, Kavita Bala, and Cyril Soler. 2014. A local frequency analysis of light scattering and absorption. *ACM Transactions on Graphics* 33, 5 (2014), 163.

Laurent Belcour and Pascal Barla. 2017. A Practical Extension to Microfacet Theory for the Modeling of Varying Iridescence. *ACM Transactions on Graphics* 36, 4 (July 2017), 65.

Brent Burley. 2012. Physically-based shading at disney. In *Physically Based Shading in Theory and Practice - SIGGRAPH Courses*, Vol. 2012. 1–7.

Qiang Dai, Jiaping Wang, Yiming Liu, John Snyder, Enhua Wu, and Baining Guo. 2009. The Dual-microfacet Model for Capturing Thin Transparent Slabs. In *Computer Graphics Forum*, Vol. 28. Wiley Online Library, 1917–1925.

Charles De Rousiers, Adrien Bousseau, Kartic Subr, Nicolas Holzschuch, and Ravi Ramamoorthi. 2011. Real-time rough refraction. In *Symposium on Interactive 3D Graphics and Games*. ACM, 1–7.

Craig Donner and Henrik Wann Jensen. 2005. Light diffusion in multi-layered translucent materials. *ACM Transactions on Graphics* 24, 3 (2005), 1032–1039.

Julie Dorsey and Pat Hanrahan. 1996. Modeling and Rendering of Metallic Patinas. In *SIGGRAPH '96*. ACM, 387–396.

Michal Drobot. 2017. Practical Multilayered Materials in Call of Duty: Infinite Warfare. In *Physically Based Shading Theory Practice - SIGGRAPH Courses*.

Frédo Durand, Nicolas Holzschuch, Cyril Soler, Eric Chan, and François X. Sillion. 2005. A frequency analysis of light transport. In *ACM SIGGRAPH*, Vol. 24. 1115.

Oskar Elek. 2010. Layered Materials in Real-Time Rendering. In *14th Central European Seminar on Computer Graphics*. 27.

Sergey Ershov, Konstantin Kolchin, and Karol Myszkowski. 2001. Rendering Pearlescent Appearance Based On Paint-Composition Modelling. *Computer Graphics Forum* 20, 3 (sep 2001), 227–238.

Ian Grant and G Hunt. 1969. Discrete space theory of radiative transfer. I. Fundamentals. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, Vol. 313. The Royal Society, 183–197.

Jie Guo, Qian Jinghui, Guo Yanwen, and Pan Jingui. 2016. Rendering Thin Transparent Layers with Extended Normal Distribution Functions. *IEEE Transactions on Visualization and Computer Graphics* PP, 99 (2016), 1–1.

Eric Heitz and Eugene d'Eon. 2014. Importance Sampling Microfacet-Based BSDFs using the Distribution of Visible Normals. In *Computer Graphics Forum*, Vol. 33. Wiley Online Library, 103–112.

Eric Heitz, Johannes Hanika, Eugene d'Eon, and Carsten Dachsbacher. 2016. Multiple-scattering microfacet BSDFs with the Smith model. *ACM Transactions on Graphics* 35, 4 (2016), 58.

Christophe Hery, Ryusuke Villemin, and Junyi Ling. 2017. Pixarfis Foundation for Materials. In *Physically Based Shading in Theory and Practice - SIGGRAPH Courses.*

Wenzel Jakob, Eugene D'Eon, Otto Jakob, and Steve Marschner. 2014. A comprehensive framework for rendering layered materials. *ACM Transactions on Graphics* 33, 4 (jul 2014), 1–14.

Henrik Wann Jensen, Justin Legakis, and Julie Dorsey. 1999. Rendering of wet materials. In *Rendering Techniquesfi 99.* Springer, 273–281.

Brian Karis. 2013. Real Shading in Unreal Engine 4. In *Physically Based Shading in Theory and Practice - SIGGRAPH Courses.*

Christopher Kulla and Alejandro Conty. 2017. Revisiting Physically Based Shading at Imageworks. In *SIGGRAPH Course, Physically Based Shading.*

Sebastien Lagarde and Charles De Rousiers. 2014. Moving Frostbite to PBR. In *Physically Based Shading Theory Practice - SIGGRAPH Courses.*

Anders Langlands. 2014. Physically Based Shader Design in Arnold. In *Physically Based Shading in Theory and Practice - SIGGRAPH Courses.*

Joakim Löw, Joel Kronander, Anders Ynnerman, and Jonas Unger. 2012. BRDF models for accurate and efficient rendering of glossy surfaces. *ACM Transactions on Graphics* 31, 1 (jan 2012), 1–14.

Matt Pharr and Greg Humphreys. 2010. *Physically Based Rendering, Second Edition: From Theory To Implementation* (2nd ed.). Morgan Kaufmann Publishers Inc.

Mickael Ribardière, Benjamin Bringier, Daniel Meneveaux, and Lionel Simonot. 2017. STD: Student's t-Distribution of Slopes for Microfacet Based BSDFs. In *Computer Graphics Forum*, Vol. 36. Wiley Online Library, 421–429.

K. F. Riley, M. P. Hobson, and S. J. Bence. 2010. *Mathematical methods in physics and engineering (3rd ed.).* Cambridge University Press.

Jos Stam. 2001. An illumination model for a skin layer bounded by rough surfaces. In *Rendering Techniques 2001.* Springer, 39–52.

H. C. van de Hulst. 1980. *Multiple Light Scattering.* Vol. 1. Academic Press, New York.

Eric Veach. 1998. *Robust monte carlo methods for light transport simulation.* Stanford University Stanford.

Bruce Walter, Stephen Marschner, Hongsong Li, and Kenneth Torrance. 2007. Microfacet Models for Refraction through Rough Surfaces. *Rendering Techniques* (2007), 195–206.

Andrea Weidlich and Alexander Wilkie. 2007. Arbitrarily Layered Micro-facet Surfaces. In *ACM GRAPHITE.* 171–178.

A FINDING GGX'S LINEAR SPACE

To find the function $f(\alpha)$ that transform the roughness of the GGX lobe into a space where multiple bounces behave linearly, we start from the 2-bounces case where both surface have the same roughness α:

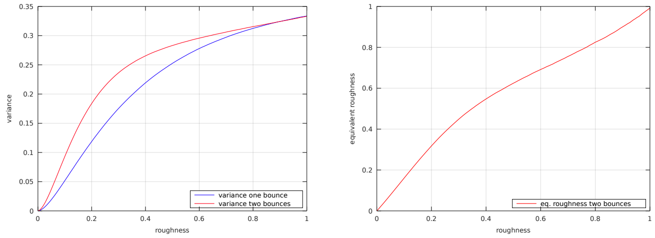
$$\begin{aligned} \alpha_{12} &= f^{-1}\left[f(\alpha) + f(\alpha)\right] && (41) \\ &= f^{-1}\left[2 f(\alpha)\right], && (42) \end{aligned}$$

It follows that for this configuration:

$$f(\alpha_{12}) = 2 f(\alpha). \qquad (43)$$

Using our numerical evaluation of the variance w/r roughness for the 1-bounce and 2-bounces case, we can extract α_{12} as a function of α. For that, we invert numerically the function variance w/r roughness for the 1-bounce case and evaluate the roughness corresponding to the 2-bounces variance. Fig. 20, illustrate this step.

Using this knowledge, we can extract the linear space transformation (see Fig. 4 left inset) up to a scaling factor. We iterate Equation 43 from a small value for both α and $f(\alpha)$ (we used 10^{-3}) and iteratively evaluate α_{12} from the graph and $f(\alpha_{12})$ using the current function value.



a) Variance w/r roughness

b) 2-bounces roughness w/r roughness

Fig. 20. Using the mapping between roughness and variance for the 1-bounce and 2-bounces cases (where both roughnesses are equal (a)), we can extract the apparent roughness of the 2-bounces configuration (b).

B ROUGHNESS SCALING FOR TRANSMISSION

Here we derive the following Jacobians:

$$\begin{aligned} J_t &= \frac{\mathrm{d}\tan(\mathbf{h}_t)}{\mathrm{d}\theta_t} = \frac{\eta_{12}\langle \boldsymbol{\omega}_t \cdot \mathbf{n}\rangle}{\eta_{12}\langle \boldsymbol{\omega}_i \cdot \mathbf{n}\rangle + \langle \boldsymbol{\omega}_t \cdot \mathbf{n}\rangle} && (44) \\ J_r &= \frac{\mathrm{d}\tan(\mathbf{h}_r)}{\mathrm{d}\theta_o} = \frac{1}{2} && (45) \end{aligned}$$

Remember that:

$$\tan(\mathbf{h}_t) = \frac{\eta_1 \sin(\theta_i) + \eta_2 \sin(\theta_t)}{\eta_1 \cos(\theta_i) + \eta_2 \cos(\theta_t)}.$$

It follows that:

$$\begin{aligned} J_t &= \frac{\mathrm{d}\tan(\mathbf{h}_t)}{\mathrm{d}\theta_t} \\ &= \frac{\eta_2 \cos(\theta_t)}{\eta_1 \cos(\theta_i) + \eta_2 \cos(\theta_t)} + \frac{\eta_2 \sin(\theta_t)\left[\eta_1 \sin(\theta_i) + \eta_2 \sin(\theta_t)\right]}{\left[\eta_1 \cos(\theta_i) + \eta_2 \cos(\theta_t)\right]^2} \\ &= \frac{\eta_{12}\left[\cos(\theta_i)\cos(\theta_t) + \sin(\theta_i)\sin(\theta_t)\right] + 1}{\left[\eta_{12}\cos(\theta_i) + \cos(\theta_t)\right]^2} \end{aligned}$$

and using the equality $\sin(\theta_i)\sin(\theta_t) = -\frac{1}{\eta_{12}}\sin^2(\theta_t)$, we obtain:

$$\begin{aligned} J_t &= \frac{\eta_{12}\left[\cos(\theta_i)\cos(\theta_t)\right] + \left[1 - \sin^2(\theta_t)\right]}{\left[\eta_{12}\cos(\theta_i) + \cos(\theta_t)\right]^2} \\ &= \frac{\cos(\theta_t)}{\eta_{12}\cos(\theta_i) + \cos(\theta_t)} \end{aligned}$$

which is Equation 44. For the second Jacobian, we use the definition of the reflected tangent:

$$\tan(\mathbf{h}_r) = \frac{-\sin(\theta_t) + \sin(\theta_o)}{\cos(\theta_t) + \cos(\theta_o)},$$

we derive:

$$\begin{aligned} J_r &= \frac{\mathrm{d}\tan(\mathbf{h}_r)}{\mathrm{d}\theta_o} \\ &= \frac{\cos(\theta_o)\left[\cos(\theta_t) + \cos(\theta_o)\right] + \sin(\theta_o)\left[-\sin(\theta_t) + \sin(\theta_o)\right]}{\left[\cos(\theta_t) + \cos(\theta_o)\right]^2}. \end{aligned}$$

Assuming that we are in the fake reflection configuration, we equate $\theta_o = \theta_t$ and obtain:

$$\begin{aligned} J_r &= \frac{2\cos(\theta_t)^2}{\left[2\cos(\theta_t)\right]^2} \\ &= \frac{1}{2}. \end{aligned}$$

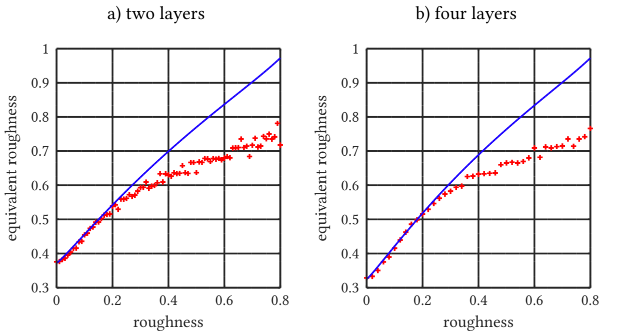


Fig. 21. We validated our adding method (in blue) with respect to a Monte-Carlo simulation of the layered structure (in red). We can predict quite closely the mean variance curve of a 2-layers structure when varying one of the roughness. Due the limitation of our fitting of the linear space transformation, our method is unable to accurately predict the correct mean roughness for higher input roughnesses. In this figure, we used $\eta_0 = 1.4$, $\alpha_0 \in [0, 1]$, $\eta_1 = 1 + i$, $\alpha_1 = 0.3$ for the two layers configuration and $\eta_0 = 1.4$, $\alpha_0 \in [0, 1]$, $\eta_1 = 1.2$, $\alpha_1 = 0.1$, $\eta_2 = 1.4$, $\alpha_2 = 0.1$, $eta_3 = 1 + i$, $\alpha_3 = 0.01$ for the four layers configuration. Both configurations use $\boldsymbol{\omega}_i = [0, 0, 1]$.

C ADDING-DOUBLING

We express here the transmitted average variance in its unnormalized form $\tilde{\sigma}^T_{13}$:

$$\tilde{\sigma}^T_{13} = \mathbf{t}_{12}\mathbf{t}_{23}\left[J_{23}\sigma^T_{12} + s^T_{23}\right] + \mathbf{t}_{12}\mathbf{r}_{23}\mathbf{r}_{21}\mathbf{t}_{23}\left[J_{23}\left(\sigma^T_{12} + s^R_{23} + s^R_{21}\right) + s^T_{23}\right] + \cdots \quad (46)$$

which simplifies to:

$$\tilde{\sigma}^T_{13} = \frac{\mathbf{t}_{12}\mathbf{t}_{23}}{1 - \mathbf{r}_{23}\mathbf{r}_{21}}\left[J_{23}\sigma^T_{12} + s^T_{23} + J_{23}\left(s^R_{23} + \sigma^R_{21}\right)\frac{\mathbf{r}_{23}\mathbf{r}_{21}}{1 - \mathbf{r}_{23}\mathbf{r}_{21}}\right]. \quad (47)$$

The normalized average variance, that will later be used in the remaining adding-doubling algorithm, is the unnormalized variance divided by the energy of light transmitted through both interfaces. It can be expressed as:

$$\sigma^T_{13} = \tilde{\sigma}^T_{13}\left[\frac{1 - \mathbf{r}_{23}\mathbf{r}_{21}}{\mathbf{t}_{12}\mathbf{t}_{23}}\right]. \quad (48)$$

We can further simplify this later equation to get:

$$\sigma^T_{13} = J_{23}\sigma^T_{12} + s^T_{23} + J_{23}\left(s^R_{23} + \sigma^R_{21}\right)\frac{\mathbf{r}_{23}\mathbf{r}_{21}}{1 - \mathbf{r}_{23}\mathbf{r}_{21}}. \quad (49)$$

This later equation contains two elements. First the variance of the purely transmitted lobe $J_{23}\sigma^T_{12} + s^T_{23}$ (e.g. the TT paths) to which adds an additional variance due to multiple scattering $J_{23}\left(s^R_{23} + \sigma^R_{21}\right)$ (e.g. the $T(RR)^+T$ paths). Similarly, we find the following expression for $\tilde{\sigma}^R_{31}$ and σ^T_{31}:

$$\sigma^T_{31} = J_{21}s^T_{32} + \sigma^T_{21} + J_{21}\left(s^R_{23} + \sigma^R_{21}\right)\frac{\mathbf{r}_{23}\mathbf{r}_{21}}{1 - \mathbf{r}_{23}\mathbf{r}_{21}} \quad (50)$$

$$\tilde{\sigma}^R_{31} = \mathbf{r}_{32}s^R_{32} + \left[\frac{\mathbf{t}_{32}\mathbf{t}_{23}\mathbf{r}_{21}}{1 - \mathbf{r}_{21}\mathbf{r}_{23}}\right] \times \left[s^T_{23} + J_{23}\left(s^T_{32} + \sigma^R_{21} + \left(s^R_{23} + \sigma^R_{21}\right)\left(\frac{\mathbf{r}_{21}\mathbf{r}_{23}}{1 - \mathbf{r}_{21}\mathbf{r}_{23}}\right)\right)\right]. \quad (51)$$

Note that when computing those intermediate variances, we will have to keep track of the global scaling factor J_{1i} and J_{i1} due to all interfaces from the top layer to the current bottom layer.

We validated our computation of the mean variance for various layer configuration (see Fig. 21. We found that our approximation of the mean variance is quite close to the ground truth for interfaces with low and moderate roughnesses (from 0 to 0.5). For higher roughnesses, we experience higher deviation from the ground truth. This is primarily due to our approximation of GGX's linear space transformation, we only fitted well the low to moderate part of the curve.

D COMPARING OFFLINE RESULTS

In Fig. 22, we provide the comparison of our method to a ground truth. To generate this later model, we stochastically evaluated the layer material by explicitly tracing the rays in the structure.

E HALF VECTOR OF TRANSMITTED RAYS

We derive the half vector of two rays wi and $\boldsymbol{\omega}_o$ refracted by a planar interface of normal the half vector $\mathbf{h} = \frac{\boldsymbol{\omega}_i + \boldsymbol{\omega}_o}{|\boldsymbol{\omega}_i + \boldsymbol{\omega}_o|}$. The expression of both refracted rays is [Walter et al. 2007]:

$$\mathbf{i}_t = \left[\eta\cos(\theta_d) - \sqrt{1 + \eta\left(\cos(\theta_d) - 1\right)}\right]\mathbf{h} - \eta\boldsymbol{\omega}_i \quad (52)$$

$$\mathbf{o}_t = \left[\eta\cos(\theta_d) - \sqrt{1 + \eta\left(\cos(\theta_d) - 1\right)}\right]\mathbf{h} - \eta\boldsymbol{\omega}_o, \quad (53)$$

where η is the ratio of index of refraction, and θ_d is the difference angle in the half-vector parametrization. If we express the unnormalized half vector of $\mathbf{i}_t$, and $\mathbf{o}_t$, we obtain:

$$\mathbf{i}_t + \mathbf{o}_t = 2\left[\eta\cos(\theta_d) - \sqrt{1 + \eta\left(\cos(\theta_d) - 1\right)}\right]\mathbf{h} - \eta\left[\boldsymbol{\omega}_i + \boldsymbol{\omega}_o\right] \quad (54)$$

$$= \left[2\left[\eta\cos(\theta_d) - \sqrt{1 + \eta\left(\cos(\theta_d) - 1\right)}\right] - \eta\right]\mathbf{h}. \quad (55)$$

The normalized half vector is thus $\mathbf{h}$.

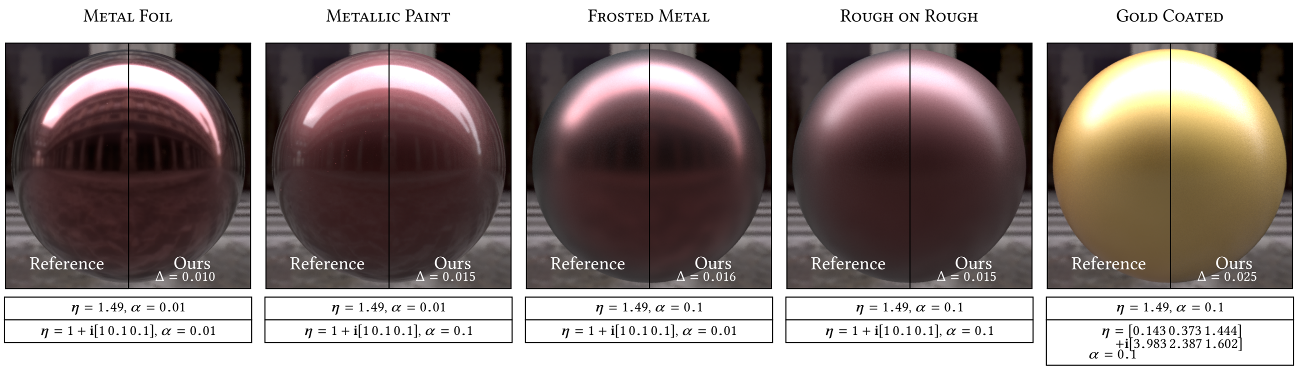

Metal Foil
Metallic Paint
Frosted Metal
Rough on Rough
Gold Coated
Reference
Ours
$\Delta = 0.010$
$\Delta = 0.015$
$\Delta = 0.016$
$\Delta = 0.015$
$\Delta = 0.025$
$\eta = 1.49, \alpha = 0.01$
$\eta = 1 + \mathbf{i}[1\,0.1\,0.1], \alpha = 0.01$
$\eta = 1.49, \alpha = 0.01$
$\eta = 1 + \mathbf{i}[1\,0.1\,0.1], \alpha = 0.1$
$\eta = 1.49, \alpha = 0.1$
$\eta = 1 + \mathbf{i}[1\,0.1\,0.1], \alpha = 0.01$
$\eta = 1.49, \alpha = 0.1$
$\eta = 1 + \mathbf{i}[1\,0.1\,0.1], \alpha = 0.1$
$\eta = 1.49, \alpha = 0.1$
$\eta = [0.143\,0.373\,1.444] + \mathbf{i}[3.983\,2.387\,1.602]$
$\alpha = 0.1$

Fig. 22. We produce various appearance using two layers (one dielectric and one metallic). We show that our method faithfully reproduces the apparent roughness of those materials and, to a certain degree, the color saturation due to multiple scattering. For each example, we report the RMSE Δ between our symmetric model and the reference.