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

To cite this version:

HAL Id: hal-01785457
https://hal.science/hal-01785457v3
Submitted on 24 Aug 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Efficient Rendering of Layered Materials using an Atomic Decomposition with Statistical Operators

LAURENT BELCOUR, Unity Technologies

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:
DOI: 10.1145/3197517.3201289

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2018 Copyright held by the owner/author(s). Publication rights licensed to ACM.
0730-0301/2018/8-ART73 $15.00
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.
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 approximate 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 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 conservation, 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 $\mu$ and the variance $\sigma$ 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

\(^1\)We deliberately omit the square on $\sigma$ for better readability.
Table 1. We summarize the different atomic operators and how we approximate the outgoing energy $e$, mean $\mu$, and variance $\sigma$ given the incident energy $e_i$, mean $\mu_i$, and variance $\sigma_i$. For each statistic, we indicate with an exponent whether it has to be used for a reflected or transmitted lobe (for example $\sigma^R$ 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.

<table>
<thead>
<tr>
<th>Rough Reflection</th>
<th>Rough Refraction</th>
<th>Absorption</th>
<th>Forward Scattering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Energy</td>
<td>$e^R = e_i \times \text{FGD}^{\sigma_0}$</td>
<td>$e^T = e_i \times [1 - \text{FGD}^{\sigma_0}]$</td>
<td>$e^T = e_i \exp \left[-\frac{\sigma_i h}{\sqrt{1 -</td>
</tr>
<tr>
<td>Mean</td>
<td>$\mu^R = -\mu_i$</td>
<td>$\mu^T = -\eta_{12} \mu_i$</td>
<td>$\mu^T = -\mu_i$</td>
</tr>
<tr>
<td>Variance</td>
<td>$\sigma^R = \sigma_i + \sigma^{R}_{12}$</td>
<td>$\sigma^T = \sigma_{1} + f(s \times \sigma_{12})$</td>
<td>$\sigma^T = \sigma_{1}$</td>
</tr>
</tbody>
</table>

BRDF, it can be described by:

$$L_o(\omega_o) = \int_{\Omega} F(\langle \omega_i, h \rangle) \frac{G(\omega_i, \omega_o) D(h)}{4(\omega_i, n)(\omega_o, n)} L_i(\omega_i) \langle \omega_i, n \rangle d\omega_i,$$

(1)

where $L_o$ (resp. $L_i$) is the outgoing (resp. incoming) radiance, $\Omega$ is the upper hemisphere, $h = (\omega_i + \omega_o)/|\omega_i + \omega_o|$ is the half-vector, $\langle \omega_i, n \rangle$ 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:

$$\text{FGD} = \int_{\Omega} F(\langle \omega_i, h \rangle) \frac{G(\omega_i, \omega_o) D(h)}{4(\omega_i, n)(\omega_o, n)} \langle \omega_i, n \rangle d\omega_o$$

(2)

We will omit all parameters for $\text{FGD} = \text{FGD}(\omega_i, \alpha, \eta + ix)$, where $\eta + ix$ is the complex index of refraction (IOR) at the interface. For isotropic BRDFs, the elevation is sufficient to parameterize the incoming direction. We approximate the energy of the outgoing radiance by decoupling FGD from incident radiance:

$$\int L_o(\omega_o) d\omega_o \approx \text{FGD} \times \int L_i(\omega_i) d\omega_i.$$

(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:

$$\omega'_{\|xy} = \beta n_{\|xy} + (1 - \beta)\omega_{\|xy},$$

(4)

where $\omega_{\|xy} = [\omega_x, \omega_y]$ is the vector in the tangential plane, and $\beta$ 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.

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 $\alpha_1$ and $\alpha_2$ is:

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

(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:

$$\sigma = f(\alpha) \approx \frac{\alpha^{4.1}}{(1 - \alpha^{1.1})},$$

(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
We can see very little deviation of the mean transmitted direction from the reflection, the mean refracted direction is roughness dependent conservation between the reflected and refracted directional fields. A rough dielectric interface distributes light in both the upper and lower half spaces. The mean transmitted direction is given by 1 − FGD∞. We can see very little deviation of the mean transmitted direction from the reflected direction for small roughnesses (a), but a noticeable one for higher roughnesses (b).

**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 empirically found that using the purely refracted direction was better, as adding the fake reflected and transmitted lobe peaks but it overestimates 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:

\[
s = \frac{1}{2} \left[ 1 + \eta_{12} \left( \frac{\omega_i \cdot n}{\alpha_t \cdot n} \right) \right].
\]

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

---

2See 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.

Summary. Given an incident directional function with energy \( e_i \), mean \( \mu_i \) and variance \( \sigma_i \), the transmitted energy \( e_T \), mean \( \mu_T \), and variance \( \sigma_T \) are approximated by:

\[
e_T = e_i \times (1 - \text{FGD}^{\infty})
\]

\[
\mu_T = -\eta_{12} \mu_i
\]

\[
\sigma_T = \frac{\sigma_i}{\eta_{12}} + f(s \times \alpha)
\]

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:

\[
L_\omega(\omega_0) = \exp\left[-\sigma_L \frac{h}{\langle \omega_0, \omega \rangle} \right] L_i(\omega_0),
\]

where \( \sigma_L \) is the transmittance cross-section and \( h \) is the depth of the layer. We approximate the average energy using the incident mean attenuation: \( e_\omega = e_i \exp\left(\frac{-\sigma_L h}{\langle \omega_0, \omega \rangle}\right) \).

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.

Summary. Given an incident directional function with energy \( e_i \), mean \( \mu_i \) and variance \( \sigma_i \), we approximate the outgoing energy \( e_T \), mean \( \mu_T \), and variance \( \sigma_T \) transmitted by volumetric absorption by:

\[
e_T = e_i \exp\left(\frac{-\sigma_L h}{\langle \omega_0, \omega \rangle}\right)
\]

\[
\mu_T = -\mu_i
\]

\[
\sigma_T = \sigma_i
\]

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 \( L_i(\omega) \), the amount of light scattered by the medium is [Pharr and Humphreys 2010]:

\[
L_\omega(\omega) = \sigma_s \int_0^h \exp\left[-\sigma_t t \right] L_s(t, \omega) dt,
\]

with \( L_s(t, \omega) = \int_{S^2} p(t, \omega, \omega_i) \exp\left[-\sigma_t (\omega, n) \right] L_i(\omega_i) d\omega_i \).

\[
\end{equation}

where \( \sigma_s \) 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:

\[
\int_\Omega L_\omega(\omega) d\omega \approx \sigma_s h \exp\left[-\frac{\sigma_t h}{\langle \omega_0, \omega \rangle}\right] \int_\Omega L_i(\omega) d\omega.
\]

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 \( \sigma_t \geq 0.7 \), 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:

\[
\sigma_\omega = \left[1 - \frac{g}{g + \frac{1}{1 + g}}\right]^{0.8}
\]

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
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.

Summary. Given an incident directional function with energy \( e_i \), mean \( \mu_i \) and variance \( \sigma_i \), the transmitted energy \( e_T \), mean \( \mu_T \), variance \( \sigma_T \) from volumetric scattering are approximated by:

\[
e_T = e_i \left( \frac{\sigma_i h}{\omega_{\infty}, n} \right) \exp \left( - \frac{\sigma_i h}{\omega_{\infty}, n} \right)
\]

\[
\mu_T = -\mu_i
\]

\[
\sigma_T = \sigma_i + \sigma_y
\]

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 us 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 \( r \) and transmission \( t \) coefficients of radiance with respect to the light’s direction of propagation: \( r_{k,k+1} \) is the reflection coefficient of the slab for light propagating downward and \( r_{k+1,k} \) 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):

\[
l_r^+ = r_{12}l_i^- + t_{21}l_r^+.
\]

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:

\[
l_r^- = r_{21}l_r^+ + t_{12}l_i^-.
\]

We can express the global upward reflectance summarizing light transport between two layers \( r_{12} \) and \( r_{23} \) as:

\[
r_{13} = r_{12} + t_{12}r_{23}t_{21} + t_{12}^2r_{23}t_{21} + \cdots
\]

\[
= r_{12} + \sum_{k=0}^{\infty} t_{12}r_{23}^k t_{21}.
\]

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

\[
r_{13} = r_{12} + \frac{t_{12}r_{23}t_{21}}{1 - r_{23}t_{21}}.
\]

Similarly, we can express the global upward transmittance as:

\[
t_{31} = \frac{t_{32}t_{21}}{1 - r_{23}t_{21}}.
\]

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

\[
r_{31} = r_{32} + \frac{t_{32}t_{21}r_{23}}{1 - r_{23}t_{21}}
\]

\[
t_{13} = t_{12}r_{23} + \frac{t_{12}t_{21}}{1 - r_{23}t_{21}}.
\]

The adding algorithm iteratively expresses the reflectance and transmittance of an increasing number of interfaces by using \( r_{13} \), \( t_{31} \), \( r_{31} \), and \( t_{13} \) 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 \( r_{12} = r_{23} \), \( t_{12} = t_{23} \), \( r_{21} = r_{32} \), and \( t_{21} = t_{32} \). 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 \( r \) and \( t \) terms with the reflectance and transmittance terms studied in Section 4. For a rough surface, we have: \( r_{12} = \text{F\hspace{1pt}G\hspace{1pt}D}^{\infty} \), \( t_{12} = [1 - \text{F\hspace{1pt}G\hspace{1pt}D}^{\infty}] \), \( r_{21} = \text{F\hspace{1pt}G\hspace{1pt}D}^{\infty} \), and \( t_{21} = [1 - \text{F\hspace{1pt}G\hspace{1pt}D}^{\infty}] \).

---

5See 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 \( t_{21} \) does not account for the angular spread of light reflected on interface 2 \( \rightarrow 3 \) and we need to bring this information back:

\[
L_t(\omega_o) = \int_{\Omega} L_i(\omega_i) \rho(\omega_i, \omega_t, \sigma_{23}) d\omega_t,
\]

where \( \omega_i = \text{refract}(\omega_o, \eta_{12}) \), \( T(\omega_i) = 1 - F(\omega_i) \) is the Fresnel transmittance, and \( \rho(\omega_i, \omega_t, \sigma_{23}) \) is the microfacet BRDF of layer interface 2 \( \rightarrow 3 \). When TIR occurs, \( T(\omega_i) = 0 \) for \( \omega_i \) outside of the extinction cone. A solution to compute the amount of energy lost due to TIR is to decouple its contribution from the transport integral:

\[
L_t(\omega_o) = \int_{\Omega} L_i(\omega_i) \rho(\omega_i, \omega_t, \sigma_{23}) d\omega_t \times \left( \int_{\Omega} (1 - F(\omega_i)) D(\omega_i, \sigma_{23}) d\omega_i \right)^{\Omega \rightarrow \text{TIR}},
\]

where \( D(\omega_i, \sigma_{23}) \) 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 \( r_{21} \), and \( t_{21} \) by:

\[
r_{21} \leftarrow r_{21} + (1 - \text{TIR}) \times t_{21},
\]

\[
t_{21} \leftarrow \text{TIR} \times t_{21}.
\]

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 arithmetico-geometric 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 \( \sigma_{13}^R \) accounting for multiple scattering is:

\[
\sigma_{13}^R = r_{12} \sigma_{12}^R + \sum_{k=0}^{\infty} \left[ t_{12} \frac{k+1}{r_{23}} \right] \left[ \sigma_{12}^T + \left( (k+1) \frac{r_{23}}{r_{21}} + k \sigma_{21}^R \right) \right] J_{21} + \sigma_{21}^T,
\]

where \( \sigma_{ij}^R \) (resp. \( \sigma_{ij}^T \)) 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]:

\[
\sum_{k=0}^{\infty} kr^k = \frac{r}{(1-r)^2} \quad \text{with } r \in [0, 1].
\]

Since \( r_{23}f_{21} < 1 \) (or no light can enter the layer), Equation 36 simplifies to:

\[
\sigma_{13}^R = r_{12} \sigma_{12}^R + \left[ t_{12} r_{23} t_{21} \frac{\sigma_{12}^T + \sigma_{21}^T}{1 - r_{23}f_{21}} \right] J_{21} + \sigma_{21}^T + J_{21} \left[ \sigma_{23}^R + \frac{r_{23}f_{21}}{1 - r_{23}f_{21}} \right].
\]

Similar forms can be derived for the transmitted average variance. We detail them in Appendix C. They will be used in the next iteration toward model (Section 6.1); and one tailored for bidirectional light transport (Section 6.2). 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 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 \( \mu_i = [\omega_{i1}, \omega_{i2}, \omega_{i3}] \), 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 \( \mu_k \) and variance \( \sigma_k \). 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 \( \mu_k \), and roughness \( f^{-1} \). We use our fake refraction

\( ^7 \)see our code mitsuba/plugins/layered_forward.cpp, line 105
\( ^8 \)see our code mitsuba/plugins/layered_dielectric.cpp, line 258

\(^6\)See webgl/compute_TIR.html
model for the transmitted lobe. Thus the reflection and transmission models are:

\[
\rho(\omega_i, \omega_o) = \sum_{k=0}^{N} e_k \rho_k(\omega_k, \omega_o, \sigma_k),
\]

\[\text{with} \quad \begin{cases} 
\sigma_k &= \theta^{-1}(\sigma_k) \\
\omega_k &= \text{reflect}(\rho_k) \\
\rho_k(\omega_k, \omega_o, \sigma_k) &= \frac{D(h)G(\omega_k, \omega_o)}{4(\omega_k, n) (\omega_o, n)}
\end{cases}
\]

where \(e_k\), \(\omega_k\), and \(\sigma_k\) 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 artifacts, 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:

\[
p = \frac{e_{all}}{\sum_{i=0}^{N} e_{p_i}} \sum_{i=0}^{N} e_{p_i}(\omega_i, \omega_o, \alpha_i),
\]

where \(\omega_o\) 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 \(\alpha_i\), and \(p_i\) 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 \(\theta_i\). 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 \(\cos(\theta_i)\) and not \(\cos(\theta_i)\) and the transmitted roughness 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...
would require an enormous amount of disk storage for the different references: one computing all bounces between interfaces (middle) and one computing only one reflection per layer (right).

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 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.

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 efficiently uses 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.

**Performance.** 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 practicality of our implementation. Note that our model applies to deferred rendering as well. In our implementation, we restricted the number...
We rendered it at a resolution of 1920x1080 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.

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
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 glint 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 LSFM1 from www.blendswap.com respectively. The Fox model is designed by Slavik from www.thingsiverse.com.
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 \( \alpha \):

\[
\sigma_{12} = f^{-1}(f(\alpha) + f(\alpha)) = f^{-1}(2f(\alpha)).
\]

It follows that for this configuration:

\[
f(\sigma_{12}) = 2f(\alpha).
\]

Using our numerical evaluation of the variance w/r roughness for the 1-bounce and 2-bounces case, we can extract \( \sigma_{12} \) as a function of \( \alpha \). 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 \( \alpha \) and \( f(\alpha) \) (we used \( 10^{-3} \)) and iteratively evaluate \( \sigma_{12} \) from the graph and \( f(\sigma_{12}) \) using the current function value.

B ROUGHNESS SCALING FOR TRANSMISSION

Here we derive the following Jacobians:

\[
J_f = \frac{d\tan(h_r)}{d\theta} = \frac{\eta_2(\alpha_1 \cdot \mathbf{n})}{\eta_1(\alpha_1 \cdot \mathbf{n}) + \langle \alpha_1 \cdot \mathbf{n} \rangle} \tag{44}
\]

\[
J_r = \frac{d\tan(h_r)}{d\theta} = \frac{1}{2} \tag{45}
\]

Remember that:

\[
\tan(h_r) = \frac{\eta_1 \sin(\theta_1) + \eta_2 \sin(\theta_1)}{\eta_1 \cos(\theta_1) + \eta_2 \cos(\theta_1)}
\]

It follows that:

\[
J_f = \frac{d\tan(h_r)}{d\theta} = \frac{\eta_2 \cos(\theta_1) \sin(\theta_1) - \eta_1 \cos(\theta_1) \sin(\theta_1)}{\eta_1 \cos(\theta_1) + \eta_2 \cos(\theta_1)}
\]

and using the equality \( \sin(\theta_1) \sin(\theta_1) = -\frac{1}{\eta_1^2} \sin^2(\theta_1) \), we obtain:

\[
J_f = \frac{\eta_2 \cos(\theta_1) \sin(\theta_1) + 1 - \sin^2(\theta_1)}{\eta_1 \cos(\theta_1) + \cos(\theta_1)}
\]

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

\[
\tan(h_r) = -\sin(\theta_1) + \sin(\theta_o) \cos(\theta_1) + \cos(\theta_o)
\]

we derive:

\[
J_r = \frac{d\tan(h_r)}{d\theta} = \frac{\cos(\theta_o) \sin(\theta_1) + \sin(\theta_o) \cos(\theta_1)}{\cos(\theta_1) + \cos(\theta_o)}
\]

Assuming that we are in the fake reflection configuration, we equate \( \theta_o = \theta_1 \) and obtain:

\[
J_r = \frac{2 \cos(\theta_1)^2}{2 \cos(\theta_1)^2} = \frac{1}{2}
\]
We can further simplify this later equation to get:

\[ \sigma_{51}^R = r_{32} s_{32}^T + \frac{t_{32} t_{23} F_{21}}{1 - r_{21} F_{23}} \times \left[ s_{23}^T + J_{23} \left( s_{23}^T \odot \sigma_{21}^R + \sigma_{21}^R \right) \right] \quad (51) \]

Note that when computing those intermediate variances, we will have to keep track of the global scaling factor \( J_{21} \) and \( J_{11} \) 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 \( w_i \) and \( \omega_o \) refracted by a planar interface of normal the half vector \( h = \frac{\omega_i + \omega_o}{\|\omega_i + \omega_o\|} \). The expression of both refracted rays is [Walter et al. 2007]:

\[ i_r = \left[ \eta \cos(\theta_d) - \sqrt{1 + \eta \left( \cos(\theta_d) - 1 \right)} \right] h - \eta \omega_i \]

\[ o_r = \left[ \eta \cos(\theta_d) - \sqrt{1 + \eta \left( \cos(\theta_d) - 1 \right)} \right] h - \eta \omega_o \]

where \( \eta \) is the ratio of index of refraction, and \( \theta_d \) is the difference angle in the half-vector parametrization. If we express the unnormalized half vector of \( i_r \) and \( o_r \), we obtain:

\[ i_r + o_r = 2 \left[ \eta \cos(\theta_d) - \sqrt{1 + \eta \left( \cos(\theta_d) - 1 \right)} \right] h - \eta \left[ \omega_i + \omega_o \right] \]

\[ = 2 \left[ \eta \cos(\theta_d) - \sqrt{1 + \eta \left( \cos(\theta_d) - 1 \right)} \right] - \eta \ h. \quad (54) \]

The normalized half vector is thus \( h \).

We express here the transmitted average variance in its unnormalized form \( \tilde{\sigma}_{13}^T \):

\[
\tilde{\sigma}_{13}^T = t_{12} t_{23} \left[ J_{23} s_{12}^T + s_{23}^T \right] + t_{12} r_{23} r_{21} t_{23} \left[ J_{23} \left( s_{12}^T + r_{23}^T + \sigma_{21}^R \right) + s_{23}^T \right] + \cdots \quad (46)
\]

which simplifies to:

\[
\tilde{\sigma}_{13}^T = \frac{t_{12} t_{23}}{1 - r_{23} r_{21}} \left[ J_{23} s_{12}^T + s_{23}^T \right] + J_{23} \left( s_{23}^T + r_{23}^T + \sigma_{21}^R \right) \frac{r_{23} r_{21}}{1 - r_{23} r_{21}} \quad (47)
\]

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:

\[ \sigma_{13}^T = \tilde{\sigma}_{13}^T \left[ 1 - r_{23} r_{21} \right], \quad (48) \]

We can further simplify this later equation to get:

\[ \sigma_{13}^T = J_{23} s_{12}^T + s_{23}^T + J_{23} \left( s_{23}^T + r_{23}^T + \sigma_{21}^R \right) \frac{r_{23} r_{21}}{1 - r_{23} r_{21}} \quad (49) \]

This later equation contains two elements. First the variance of the purely transmitted lobe \( J_{23} s_{12}^T + s_{23}^T \) (e.g. the TT paths) to which adds an additional variance due to multiple scattering \( J_{23} \left( s_{23}^T + r_{23}^T \right) \) (e.g. the TR+RT paths). Similarly, we find the following expression for \( \tilde{\sigma}_{31}^R \) and \( \sigma_{31}^R \):

\[ \tilde{\sigma}_{31}^R = J_{21} s_{32}^R + s_{21}^R + J_{21} \left( s_{21}^R + r_{21}^R \right) \frac{r_{21} r_{23}}{1 - r_{21} r_{23}} \quad (50) \]
Fig. 22. We produce various appearances 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 $\Delta$ between our symmetric model and the reference.