

Position-Free Monte Carlo Simulation for Arbitrary Layered BSDFs

YU GUO, University of California, Irvine

MILOŠ HAŠAN, Autodesk

SHUANG ZHAO, University of California, Irvine

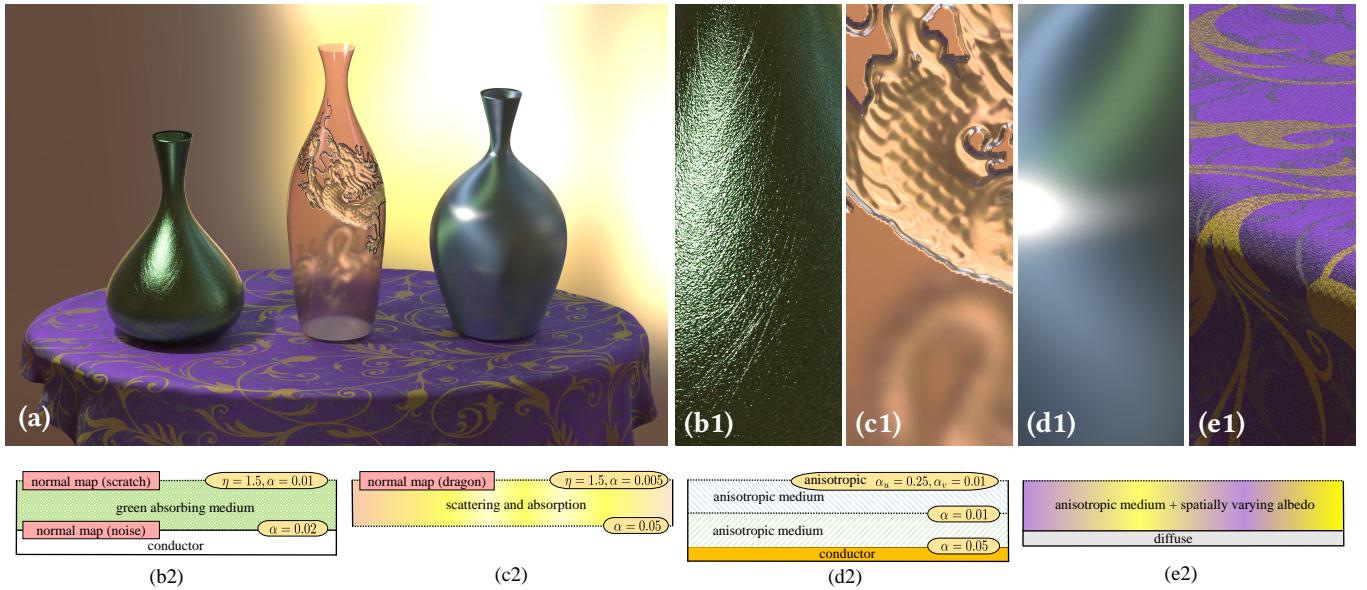


Fig. 1. We introduce a new BSDF model leveraging an efficient Monte Carlo simulation algorithm applied locally to layered geometries. Our model enjoys the flexibility of using arbitrary layer interfaces and internal media and is capable of reproducing a wide variety of appearances. This example contains three vases on a tablecloth, all described using our BSDF model (see the insets for layer configurations).

Real-world materials are often layered: metallic paints, biological tissues, and many more. Variation in the interface and volumetric scattering properties of the layers leads to a rich diversity of material appearances from anisotropic highlights to complex textures and relief patterns. However, simulating light-layer interactions is a challenging problem. Past analytical or numerical solutions either introduce several approximations and limitations, or rely on expensive operations on discretized BSDFs, preventing the ability to freely vary the layer properties spatially. We introduce a new unbiased layered BSDF model based on Monte Carlo simulation, whose only assumption is the layer assumption itself. Our novel position-free path formulation is fundamentally more powerful at constructing light transport paths than generic light transport algorithms applied to the special case of flat layers, since it is based on a product of solid angle instead of area measures, so does not contain the high-variance geometry terms needed in the standard

Authors' addresses: Yu Guo, University of California, Irvine, guo.yu@uci.edu; Miloš Hašan, Autodesk, miloshasan@gmail.com; Shuang Zhao, University of California, Irvine, shz@ics.uci.edu.

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 ACM 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 Association for Computing Machinery.

0730-0301/2018/11-ART279 \$15.00

<https://doi.org/10.1145/3272127.3275053>

formulation. We introduce two techniques for sampling the position-free path integral, a forward path tracer with next-event estimation and a full bidirectional estimator. We show a number of examples, featuring multiple layers with surface and volumetric scattering, surface and phase function anisotropy, and spatial variation in all parameters.

CCS Concepts: • Computing methodologies → Rendering; Ray tracing;

Additional Key Words and Phrases: BSDF, layered materials, Monte Carlo integration

ACM Reference Format:

Yu Guo, Miloš Hašan, and Shuang Zhao. 2018. Position-Free Monte Carlo Simulation for Arbitrary Layered BSDFs. *ACM Trans. Graph.* 37, 6, Article 279 (November 2018), 17 pages. <https://doi.org/10.1145/3272127.3275053>

1 INTRODUCTION

Physically-based shading models have become mature and commonplace in recent years across a number of rendering applications, within entertainment, architecture, and industrial design. However, we are seeing constant progress in the area of material reflection and scattering models, aiming to achieve higher physical realism and to enable more effective material content creation.

Many real world materials are comprised of thin layers with varying compositions. For example, metallic paint is a dielectric

coating covering a metallic substrate composed of randomly oriented aluminum flakes; the absorption and scattering properties of the dielectric layer give the material its color and modify its directional scattering properties as well. Different characteristics of such interfaces and volumetric scattering properties can produce richly diverse material appearances from anisotropic highlights to complex textures. Furthermore, detailed layer thickness variations, scratches and bumps on the layer interfaces give these materials additional richness. Accurately understanding and simulating these interactions is therefore key to further progress in the rendering of materials.

However, explicitly simulating light-layer interactions by modeling the full geometry of these layers would be very expensive and cumbersome. The complex and spatially varying interface and internal microgeometries are much too costly to describe and simulate using standard 3D scene modeling tools such as triangle meshes and volumetric grids.

A few techniques have been developed to address this problem. Weidlich and Wilkie [2007] construct a simple and flexible analytical model. However, significant approximations are necessary; interface roughness is not fully handled for transmission, and no volumetric scattering is supported. The work of Belcour [2018] recently introduced a more advanced approach based on tracking low-order moments of the BSDF lobes; however, it still introduces some approximations and limitations. On the other hand, Jakob et al. [2014] (with a recent follow-up [Zeltner and Jakob 2018]) introduce a solution that is very accurate, but expensive: it represents BSDFs as discretized datasets and relies on expensive Fourier-domain operations on these to implement layer composition and thickness adjustment. This makes free spatial variation of the layer properties prohibitively expensive: a significant limitation in practice.

In this paper, we introduce a new layered BSDF model without the above limitations. Our model provides an accurate, unbiased solution; to our knowledge, it is the only such model. Unlike previous work, we do not attempt to derive an analytic model for the BSDF lobe shapes. Instead, inside the evaluation and sampling routines of the layered BSDF, we run a Monte Carlo simulation of light transport within flat slabs. This requires no precomputation and thus can efficiently handle spatially varying appearances. It also supports the full range of editability of the layer properties, both interface and volumetric, and allows anisotropy in both interface BSDFs and phase functions. In fact, the only limiting assumption of our model is the layer assumption itself.

Our solution is fundamentally more powerful at constructing light transport paths than generic transport algorithms (e.g. standard path tracing, bidirectional or Metropolis transport); see Figure 2. We introduce a modified path integral framework for light transport in flat slabs, superior to the standard path formulation in this setting. Because it is based on a product of solid angle instead of area measures, it does not contain the high-variance geometry terms needed in standard algorithms. We introduce two simulation techniques within this formulation: the first is analogous to a forward path tracer with next event estimation through layer boundaries and multiple importance sampling; the second is a fully bidirectional estimator. We show the capabilities of this solution on a number of examples, featuring multiple layers with surface and volumetric

scattering. Our examples show spatial variation in all parameters: surface BSDF, volume and phase function parameters, layer thickness and surface normal.

2 RELATED WORK

2.1 Discretized layered BSDFs

Previously, a number of BSDF models have been proposed to describe layers with various assumptions on the interface and subsurface scattering.

An early analytical model by Hanrahan and Krueger [1993] already supported multiple layers, but only single scattering, and without supporting arbitrary BSDFs at interfaces. They also proposed to add multiple scattering by Monte Carlo simulation, but their simulation approach only considers volume scattering events (as opposed to a combination of volume and rough interface events). Furthermore, it uses binning on the outgoing direction, as opposed to an efficient BSDF evaluation method for a given outgoing direction, which is provided by our approach.

A model by Stam [2001] introduces a solution for rendering skin as a layered material consisting of rough dielectric interfaces bounding a volumetric scattering slab. The solution is based on discretization of the BSDF into a directional basis, on which the light transport problem is solved. The model introduced by Jakob et al. [2014] can be seen as a significant extension of Stam’s discretization approach, working in the Fourier domain. It handles arbitrary layer stacks, supporting subsurface scattering within thin layers using the adding-doubling method, in addition to microfacet rough interfaces. The work of Zeltner extends this approach to anisotropic surface reflectance [2018]. These models are highly accurate and efficient to render with, once the discretized BSDF has been constructed. However, as the BSDF construction in the discretized basis is relatively expensive, they are best suited for homogeneous BSDFs. A small number of such BSDFs can be spatially blended with varying weights, but this has strict limitations, compared to our support for arbitrary spatial texturing of all parameters.

2.2 Analytic layered BSDFs

The model by Weidlich and Wilkie [2007] takes a different approach. They focus on layers where subsurface scattering is absent (though absorption is allowed), by analytically combining microfacet BSDFs from the interfaces into a single, potentially multi-lobe, microfacet-like BSDF. There are significant approximations in this approach, carefully chosen so that integration (Monte Carlo or otherwise) is never required within a single BSDF query. This makes the model fast and flexible. Another recent model [Guo et al. 2017] also takes the approach of avoiding Monte Carlo integration during queries, by introducing extended normal distribution functions (ENDFs), analogous to microfacet NDFs but capturing multiple reflection or scattering events. In the most recent work, Belcour [2018] introduced an approach based on tracking low-order moments of the BSDF lobes. This is a very fast and practical solution, but still introduces some approximations and limitations (e.g. no surface or volume anisotropy). In contrast, our method offers unbiased accuracy and even more flexibility, at the cost of some additional computation and variance. Several previous techniques model light

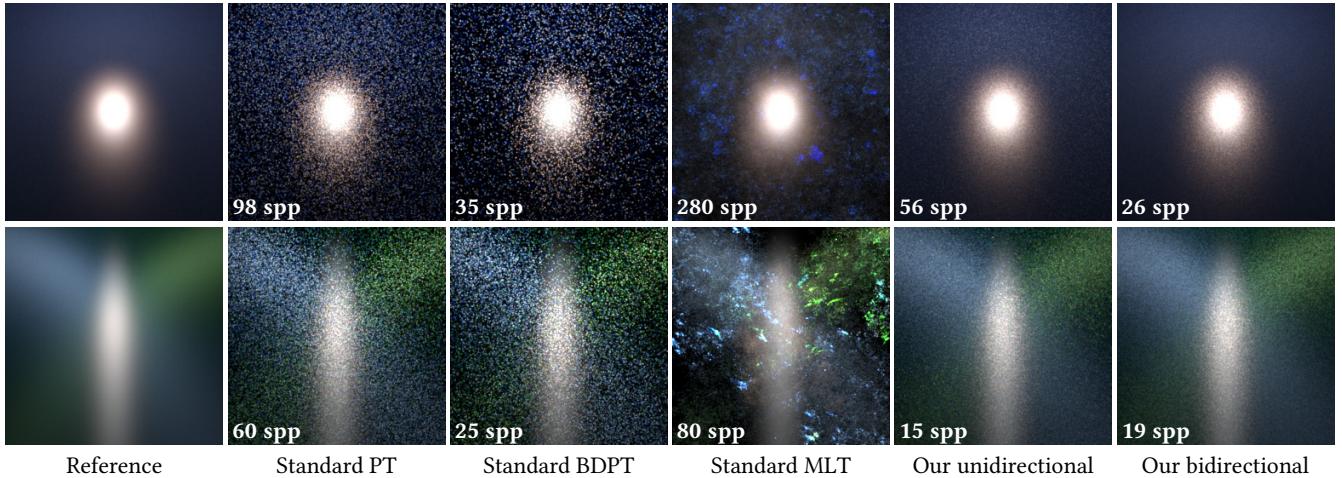


Fig. 2. Equal-time comparisons of our unidirectional and bidirectional approach to standard transport algorithms, on a simple flat layered configuration lit by a small area light. For standard PT, BDPT and MLT, results are all generated using 3D tracing by applying these algorithms in a simple 3D scene containing a very large slab with flat interfaces. **Top:** A single slab with Henyey-Greenstein scattering between two interfaces, where our estimators perform similarly, but both significantly outperform path tracing, bidirectional and Metropolis transport. **Bottom:** A more complex configuration with two slabs and three interfaces; both media are using an anisotropic microflake phase function [Jakob et al. 2010]. Our bidirectional estimator is a clear winner in this case. The references are generated using standard PT with 100K spp, and all the other images are rendered in 10 seconds.

scattering in layered materials like human skin [Donner et al. 2008], but these are focused on lateral light spreading in BSSRDFs, and are orthogonal to our focus on the directional properties of BSDF models.

2.3 Microfacet models for interfaces

BSDF models based on the microfacet theory are commonly used in computer graphics to capture how light reflects and refracts when interacting with specular surfaces with rough microstructure. The model by Walter et al. [2007] extends the microfacet model of Cook and Torrance [1982] to handle light reflection and transmittance through rough dielectric interfaces, and is currently seen as standard in physically-based rendering. We use this model to describe our layer interfaces.

The microfacet model recently developed by Heitz et al. [2016] is capable of capturing interreflections between the facets and better conserves energy. Schüssler [2017] introduced a solution to the energy loss common in normal mapping techniques, caused by a mismatch between the shading and geometric normal. These models (or any future improved microfacet models) could be combined with our approach.

2.4 Capability comparison

In Figure 3, we compare the capabilities of our approach to recent work [Belcour 2018; Zeltner and Jakob 2018]. We consider three features supported by our approach: surface anisotropy, spatial variation, and volumetric medium anisotropy. Only one of these is supported in the compared systems: spatial variation in Belcour’s approach and surface anisotropy in Zeltner’s.

3 BACKGROUND AND OVERVIEW

In this section, we explicitly state the assumptions of our method, provide background on the standard path formulation of light transport, and provide a quick overview of the rest of the paper.

3.1 Assumptions

Although light generally enters and leaves the layer from different locations, we note that when the layers are thin and the lighting is comparably distant, the entrance and departure locations will be close enough to each other. We assume it is acceptable to ignore this displacement, allowing us to describe the light transport in the layers using BSDFs, rather than BSSRDFs (Figure 5).

Furthermore, we assume that the spatial variation of layer properties is slow enough that a BSDF evaluation at a single surface point can locally approximate them as spatially uniform. This is related to the above in assuming that the horizontal spreading of light is small enough to be negligible.

In fact, these are the *only* approximating assumptions of our approach, which otherwise offers unbiased accuracy and full flexibility in setting the layer properties and varying them spatially.

3.2 Review of Veach’s path integral formulation

In the Veach formulation of light transport [1997], light paths are defined as sequences of vertices connected by segments. The value of a light transport integral (for example, but not necessarily limited to, a pixel value) is written as

$$I = \int_{\Omega} f(\bar{x}) d\mu(\bar{x}), \quad (1)$$

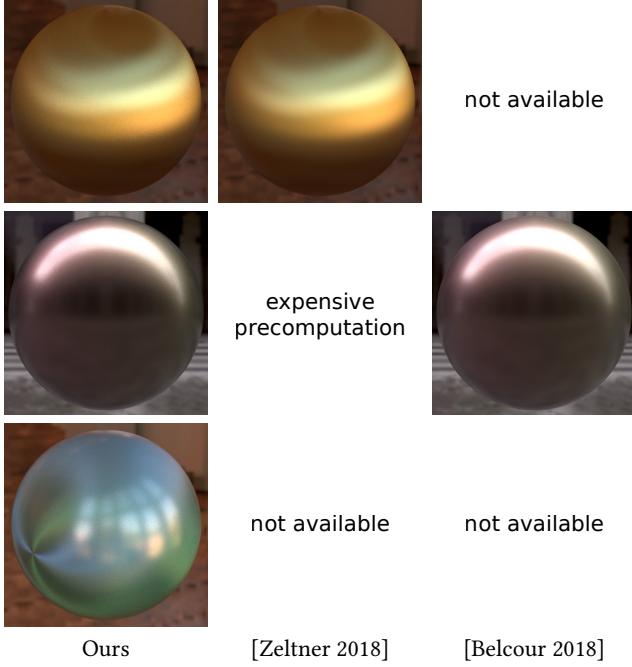


Fig. 3. Comparison to previous work. The **top row** shows an example with anisotropic surface reflectance, where our solution closely matches Zeltner’s, but Belcour’s approach does not support anisotropy. The **middle row** shows an example with spatial variation in the parameters; here our method closely matches Belcour’s, but Zeltner’s approach does not naturally support spatial variation. The **bottom row** shows a two-layer configuration with anisotropic microflake phase functions, which is only supported by our method.

where $\bar{x} = (x_0, \dots, x_k)$ is a path with k segments and $k + 1$ vertices on the surfaces or within the participating media of a scene. Ω is the space of all paths and is defined as the union of Ω_k for $k \geq 0$, where Ω_k indicates the set of paths of length k . Furthermore, $f(\bar{x})$ is the path contribution to the integral, and $\mu(\bar{x})$ is a special measure on the path space, defined as the product of area measures on the vertices x_i . The contribution $f(\bar{x})$ is a product of vertex terms (normally BSDFs and phase functions) and geometry terms corresponding to path segments. The geometry terms contain the squared distance between the two vertices in the denominator; this is a significant source of variance when trying to connect independently sampled vertices on thin layer configurations.

3.3 Paper overview

In Section 4, we describe our path formulation of layered light transport. Our path integral differs from Veach’s formulation in that it is *position-free*. The key idea is that on an infinite flat slab, the horizontal positions of vertices do not matter: it is only the vertical position (depth) of a vertex, and the *directions* between vertices, that are relevant to a light transport integral. The vertices are defined by their depth in the layer, as opposed to a full 3D position, and the segments have variable unit directions.

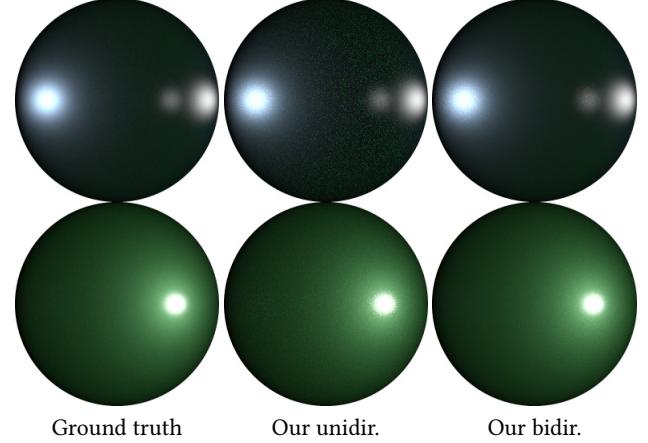


Fig. 4. Outgoing lobes of a layered BSDF (reflection and transmission) visualized as projected hemispheres. **Left:** ground truth computed by sampling and binning the light paths. **Middle:** Our unidirectional estimator. **Right:** Our bidirectional estimator (same time).

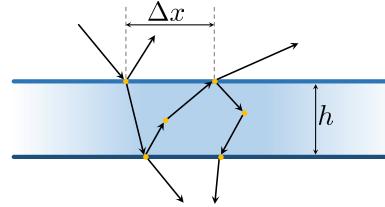


Fig. 5. Small displacement assumption: when light hits a thin layer, it gets reflected and refracted by the interfaces and scattered and absorbed internally. Since the geometric thickness h of the layer is small, we assume the displacements (e.g., Δx) of light’s entrance and departure locations can be neglected.

It is important to note that our position-free formulation is not just a simplified specialization of the standard formulation to the flat slab setting, but in fact a new approach that achieves much superior variance to the standard formulation. The key benefit of this new formulation is that it does not contain the inverse square distance falloff terms that are required between any two vertices with full positional information. This leads to high variance, even in advanced estimators such as bidirectional and Metropolis transport, which in fact perform even worse in this setting than unidirectional; see Figure 2 for examples.

In contrast, our approach leads to an efficient estimator based on unidirectional sampling with next event estimation, and an even more efficient bidirectional estimator. The unidirectional performs similarly (though usually not better) in simpler cases, but in challenging cases with sharp and/or anisotropic BSDFs and phase functions, the bidirectional version is clearly more efficient (Figure 2, bottom). Figure 4 demonstrates the performance of the estimators through BSDF lobe visualization, also showing a close match to ground truth.

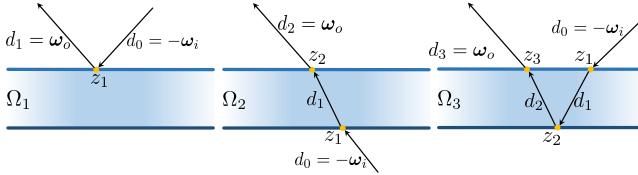


Fig. 6. Example paths of lengths 1, 2 and 3. In our formulation, the exact positions of the vertices do not matter: the z_i only carry information about which interface the vertex occurs on. The first and last in the sequence of directions d_i map to the incoming and outgoing directions of the underlying BSDF query.

In Section 5, we describe these two estimators in detail, and also focus on the two additional operations critical for integrating a BSDF into a practical renderer: importance sampling and pdf evaluation.

Finally, we present results in Section 6, and summarize in Section 7.

4 POSITION-FREE PATH FORMULATION: DEFINITION AND DERIVATION

In this section, we theoretically define the value of a layered BSDF due to a given layer stacking, for given query directions ω_i and ω_o , as a path integral. Given such a definition, any Monte Carlo method can be used to evaluate the BSDF by randomly sampling paths, evaluating their contributions and dividing by the corresponding probability density values.

4.1 Notation

We will use the notation $\cos \omega$ to denote the z -component of the unit vector ω . We will also use $\mathbb{I}(x)$ to denote an indicator function, returning 1 if the boolean condition x is true and 0 if false. A bold font is used to denote unit vectors (directions) on \mathcal{S}^2 . Please refer to Table 1 for the notation used in this section.

4.2 Position-free path integral

To develop the theory, we will first assume a single infinite flat slab with a BSDF f_{\uparrow} on the top interface and a BSDF f_{\downarrow} on the bottom interface, combined with a homogeneous scattering volume inside the slab to produce a resulting layered BSDF. The volumetric medium is defined by a phase function f_p , scattering coefficient σ_s and extinction coefficient σ_t ; we will use the notation $\hat{f}_p = \sigma_s f_p$.

For simplicity, we will drop the depth dependence of the volume parameters (though they could vary) and we will assume constant scattering / extinction coefficients, though they can vary with direction for fully anisotropic phase functions, which we also support. We will further assume that the slab has unit thickness; the formulation can be easily adjusted for any thickness.

A **vertex** $z_i \in [0, 1]$ is a single real number indicating the depth within the layer. A value of 0 or 1 indicates a surface reflection or refraction event on the bottom or top interface, respectively. Fractional values indicate volume scattering events at the specified depth. Note again that the horizontal positions of vertices on the infinite flat interfaces are not needed.

A **direction** d_i is a unit vector on \mathcal{S}^2 denoting the light flow between vertices. In our convention (inherited from Veach), the vectors point in the direction of light flow (i.e. from light source to camera), and the vertex/direction indexing follows this as well.

A **light path** \bar{x} is a sequence of directions and vertices: $\bar{x} = (\mathbf{d}_0, z_1, \mathbf{d}_1, \dots, z_k, \mathbf{d}_k)$. The first and last directions are aligned with the input and output directions of the layered BSDF query, i.e. $\mathbf{d}_0 = -\omega_i$ and $\mathbf{d}_k = \omega_o$. In contrast to Veach's formulation, the path interleaves directions with vertices, and the two ends of the path are defined by directions (not vertices). See Figure 6 for some example paths.

The **path contribution** $f(\bar{x})$ of a light path is the product of vertex terms v_i (on each vertex) and segment terms s_i (on all internal segments):

$$f(\bar{x}) = v_1 s_1 v_2 s_2 \dots s_{k-1} v_k. \quad (2)$$

The vertex term consists of the BSDF or phase function value:

$$v_i = v(z_i, -\mathbf{d}_{i-1}, \mathbf{d}_i) = \begin{cases} f_{\uparrow}(-\mathbf{d}_{i-1}, \mathbf{d}_i) & \text{if } z_i = 0, \\ f_{\downarrow}(-\mathbf{d}_{i-1}, \mathbf{d}_i) & \text{if } z_i = 1, \\ \hat{f}_p(-\mathbf{d}_{i-1}, \mathbf{d}_i) & \text{if } 0 < z_i < 1. \end{cases} \quad (3)$$

Define the transfer term $\tau(z_1, z_2, \omega)$ as follows:

$$\tau(z, z', \omega) := \exp\left(\frac{-\sigma_t |z' - z|}{|\cos \omega|}\right) \cdot \mathbb{I}\left(\frac{z' - z}{\cos \omega} > 0\right). \quad (4)$$

The purpose of the exponential term is to compute the transmittance when going from depth z to z' following direction ω . The indicator term checks the validity of the configuration (i.e. if the direction points up, then z' should be greater than z , and vice versa). The segment term for internal segments can now be defined as:

$$s_i = s(z_i, z_{i+1}, \mathbf{d}_i) := \tau(z_i, z_{i+1}, \mathbf{d}_i) \cdot |\cos \mathbf{d}_i|^{\alpha_i}, \quad (5)$$

where

$$\alpha_i = \mathbb{I}(z_i \in \{0, 1\}) + \mathbb{I}(z_{i+1} \in \{0, 1\}) - 1. \quad (6)$$

This definition encapsulates the subtle behavior of cosine terms along the path segments. For a detailed derivation, please refer to Appendix A.

The **path space** $\Omega(\omega_i, \omega_o)$ is the set of all paths of one or more vertices, such that the first direction of the path is equal to $-\omega_i$ and the last to ω_o . It can be seen as the union of the spaces of such paths of all lengths $k \geq 1$, that is, $\Omega = \bigcup_{k \geq 1} \Omega_k$.

The **path space measure** $\mu(\bar{x})$ is a product of solid angle measures σ on the internal directions of the path, times the product of line measures λ on volumetric scattering vertices. That is, for a k -vertex path,

$$\mu(\bar{x}) = \prod_{i=1}^{k-1} \sigma(\mathbf{d}_i) \cdot \prod_{i \in V(\bar{x})} \lambda(z_i). \quad (7)$$

Here $V(\bar{x})$ is the set of indices of volumetric vertices on \bar{x} , and λ is the line measure (i.e. standard Lebesgue measure on the real numbers).

Finally, we can define the **layered BSDF value** $f_l(\omega_i, \omega_o)$ as an integral over the set of paths $\Omega(\omega_i, \omega_o)$:

$$f_l(\omega_i, \omega_o) = \int_{\Omega(\omega_i, \omega_o)} f(\bar{x}) d\mu(\bar{x}). \quad (8)$$

Table 1. Notation used in the path formulation (§4).

ω_i	light direction
ω_o	camera direction
$\cos \omega$	z -component of the unit vector ω
$\mathbb{I}(x)$	binary indicator function
$f_l(\omega_i, \omega_o)$	layered BSDF (our goal)
$f_s(z, \omega_i, \omega_o)$	interface BSDF at depth z
f_t, f_{\downarrow}	BSDFs f_s at top and bottom interface
$f_p(\omega_i, \omega_o)$	phase function (normalized as a pdf)
σ_s, σ_t	scattering and absorption coefficient
\hat{f}_p	reduced phase function, $\hat{f}_p = \sigma_s f_p$
z_i	depth of i -th path vertex
d_i	direction of i -th path segment
\bar{x}	light path ($d_0, z_1, d_1, \dots, z_k, d_k$)
v_i	i -th vertex contribution
s_i	i -th segment contribution
$\tau(z, z', \omega)$	transfer through segment
α_i	i -th segment cosine term exponent
$\mu(\bar{x})$	path space measure
$\sigma(\omega)$	solid angle measure on unit directions
$\lambda(z)$	line (Lebesgue) measure on real numbers
$p(\bar{x})$	pdf of path \bar{x} in measure $\mu(\bar{x})$
$L_v(z, \omega_o)$	volume radiance
$L_s(z, \omega_o)$	outgoing surface radiance
$L_s^i(z, \omega_i)$	incoming surface radiance
$S(z, \omega)$	source term in radiative transfer eq.

As usual, any Monte Carlo method can be used to compute this integral. As long as the probability density $p(\bar{x})$ with respect to measure $\mu(\bar{x})$ of generated sample paths is known, we simply average a number of samples of the form $f(\bar{x})/p(\bar{x})$.

4.3 Derivation

Here we sketch the derivation of the path formulation. Like in Veach's version (and its volumetric extension), the derivation proceeds by recursively expanding the surface and volume rendering equation (the latter also commonly known as the radiative transfer equation). Denote the surface radiance by $L_s(z, \omega)$ (for $z \in \{0, 1\}$) and the volume radiance $L_v(z, \omega)$ (for $z \in [0, 1]$).

The volume radiance will satisfy the standard radiative transfer equation, specialized to our position-free setting:

$$L_v(z, \omega) = S(z, \omega) + \int_0^1 \frac{\tau(z', z, \omega)}{|\cos \omega|} \int_{S^2} \hat{f}_p(\omega', \omega) L_v(z', \omega') d\omega' dz', \quad (9)$$

where the source term $S(z, \omega)$ gives illumination from the boundary of the slab:

$$S(z, \omega) = \tau(0, z, \omega) L_s(0, \omega) + \tau(1, z, \omega) L_s(1, \omega). \quad (10)$$

Notice that, although the source term has two components, only one of them will be non-zero for any given query. This formulation is valid even with no scattering within the layer, in which case $\hat{f}_p = 0$ and the second term of Eq. (9) vanishes. Further, the $1/|\cos \omega|$ factor

is due to a change of variable (from free-flight distance to depth). For more details, please refer to Appendix A.

The surface radiance $L_s(z, \omega)$ satisfies the standard rendering equation:

$$L_s(z, \omega) = \int_{S^2} f_s(z, \omega, \omega') |\cos \omega'| L_s^i(z, \omega') d\sigma(\omega'), \quad (11)$$

where $L_s^i(z, \omega')$ is the incoming surface radiance. In case the incoming radiance query points back into the layer, we have

$$L_s^i(z, \omega) = L_v(z, -\omega). \quad (12)$$

The BSDF value is defined as the radiance leaving the surface in direction ω_o , under unit irradiance from a directional light in direction ω_i . This is equivalent to evaluating $L_s(1, \omega_o)$ under the boundary condition

$$L_s^i(z, \omega) = \frac{\delta(\omega - \omega_i)}{|\cos \omega_i|}. \quad (13)$$

One can easily check that the irradiance under this illumination is unit. Thus the incoming surface radiance L_s^i is given by Eq. (13) when ω points out of the layer and Eq. (12) when it points back into the layer.

The path formulation can now be obtained by recursively expanding the desired value $L_s(1, \omega_o)$ using the above equations for L_s and L_v , terminating the paths using the boundary condition. Note that:

- Each recursive expansion of Eqs. (9) or (11) will contribute an \hat{f}_p or f_s term, respectively, to the path vertex.
- Each volumetric segment will introduce a $\tau(z, z', \omega)$ term, whether the first or second term in Eq. (9) is taken.
- Expanding the rendering equation contributes a cosine term to the *next* segment, while expanding the radiative transfer equation contributes a $1/\cos$ term to the *previous* segment. A combination of these contributions explains the α_i term above.
- The last surface cosine is canceled out when using the boundary condition, due to the denominator cosine in Eq. (13).

4.4 Normal mapping

An important feature of our method is the mapping of normals of the layer interfaces, introducing mismatches between geometric (flat) normals and shading (mapped) normals. The definition of the segment term (Eq. (5)) changes with the presence of shading normals. Precisely, it becomes

$$s_i = \tau(z_i, z_{i+1}, \mathbf{d}_i) \frac{|\langle \mathbf{n}(z_i), \mathbf{d}_i \rangle| |\langle \mathbf{n}(z_{i+1}), \mathbf{d}_i \rangle|}{|\cos \mathbf{d}_i|}, \quad (14)$$

where $\mathbf{n}(z)$ denotes the local shading normal at z (for $z \in \{0, 1\}$). This term is no longer symmetric, which implies that BSDFs with mapped normals will in general not be reciprocal. When sampling paths from the light, it is important to handle such BSDF using the correction term introduced by Veach [1997] (Eq. 5.19).

4.5 Note about reciprocity

Our layered BSDF will be reciprocal whenever the path contribution $f(\bar{x})$ is symmetric with respect to the reversal of the path. Assuming normal mapping is not used, the segment term s_i will be symmetric, so the reciprocity boils down to the symmetry of the vertex terms

ω_i . This will certainly hold if all phase functions and BSDFs are reciprocal.

Note, however, that crossing an interface between regions of different index of refraction (whether smooth or rough) is not reciprocal in the usual sense. Instead, a physical refractive BSDF should obey a modified reciprocity relation $f_s(\omega_i, \omega_o) = \eta_o^2 / \eta_i^2 \cdot f_s(\omega_o, \omega_i)$ [Walter et al. 2007], where η_i and η_o are the refractive indices of the corresponding media. In the common case where the layered BSDF's incoming and outgoing directions are both assumed to be in air, the final layered BSDF will still be reciprocal, because there will be an equal number of η^2 and $1/\eta^2$ terms along the path for each medium with index η .

4.6 Multiple slabs

Finally, we support extending the framework to multiple slabs. This is relatively straightforward theoretically, and simply requires explicitly keeping track of the interface or volume that a vertex/segment belongs to. We also need to modify the transfer term $\tau(z, z' \omega)$ to return zero in cases when the segment crosses an internal interface.

Another option to obtain a multi-layer BSDF is by recursively nesting the BSDFs. To construct the layered BSDF due to a layer stacking of n slabs, we define the layered BSDF due to the stacking of the bottom $n-1$ slabs, and use this BSDF as the bottom interface's BSDF in adding the top layer according to the above theory. We have found that this approach works in practice, but its performance is worse than the explicit implementation above.

5 OUR ESTIMATORS

We now describe our specific layered BSDF method, by presenting our Monte-Carlo solutions to enable the three key operations needed to fully define a BSDF model: sampling (§5.1), evaluation (§5.2) and pdf computation (§5.3). Sampling produces the outgoing direction ω_o given the incoming one ω_i (or the reverse), while evaluation answers the BSDF query for given ω_i and ω_o . Note that the values returned from sampling, evaluation and pdf procedures are themselves stochastic, and are equal to the true BSDF value, pdf value or sampling weight only in expectation. Stochastic evaluation was also used in some recent BSDF models [Heitz et al. 2016].

Multiple importance sampling (MIS) is commonly used to combine multiple techniques to produce a given path, and key to obtaining low-noise results under complex lighting conditions. This technique typically uses the sampling pdfs of the techniques being combined to derive the weights, which requires the pdf values of the layered BSDFs. We introduce two solutions: an unbiased solution for estimating the exact pdf values in expectation, as well as a fast and approximate version which we demonstrate is sufficient for MIS (§5.3). In a supplementary document, we show that the estimators are still unbiased in the presence of approximate pdfs for MIS weighting and stochastic evaluation of both weights and function values.

5.1 BSDF sampling

Sampling a BSDF is the problem of drawing the outgoing direction ω_o given the incoming one ω_i (or the reverse), with a pdf proportional, exactly or approximately, to the value $f_l(\omega_i, \omega_o)$ (times the

cosine term, if possible). This is straightforward: we draw ω_o simply by following the stochastic process given by light interacting with the layered configuration. That is, we utilize a pure forward path tracing process that starts with a ray with direction $-\omega_i$ and explicitly simulates interactions between the ray and the layer's interfaces and internal media by sampling the corresponding BSDFs and phase functions, accumulating a throughput value along the way. When the ray eventually leaves the layer, its direction gives ω_o and the throughput of the full light transport path gives the stochastic sample weight. Formally, this weight is an estimate of the BSDF value, times the exitant cosine direction, divided by the sampling pdf in solid angle measure.

Although this simulation is analogous to standard Monte Carlo path tracing, it is usually much more efficient than tracing paths in the global scene thanks to the simplicity of the flat slab configuration (under which ray tracing becomes simple numerical computation, not requiring any acceleration structures).

5.2 BSDF evaluation

To evaluate our BSDF f_l at given incoming and outgoing directions ω_i and ω_o , we introduce two Monte Carlo based methods to evaluate the path integral from Eq. (8). The first one (§5.2.1) is analogous to a unidirectional path tracer with next-event estimation (NEE), while the second (§5.2.2) uses a bidirectional scheme.

5.2.1 Unidirectional simulation. In standard path tracing, a shading point would be directly connected to a light source in a process often called *direct illumination* or *next event estimation* (NEE), which is crucial for low-variance rendering. In an analogy to this technique, consider a shading point inside a single layer slab (whether on the bottom interface or a scattering point within the medium). We would like to create a path ending with ω_i , intuitively connecting it to an external directional light source with direction ω_i . However, direct connection between the shading point and the desired external direction is usually invalid due to the layer's top refractive interface.

To address this problem, we introduce our NEE scheme that directly connects scattering events across potentially rough refractive interfaces. Assume without loss of generality that our path tracing starts with direction ω_o . At each scattering event, we need to find a direction ω'_i so that $\omega_i \rightarrow \omega'_i$ follows the BSDF at the interface. To this end, we draw ω'_i by sampling the interface BSDF backwards, given ω_i . Finally, we simply multiply the accumulated throughput by the weight returned from the sampling routine, and the BSDF (or phase function) value at the scattering event.

Furthermore, this NEE connection can be combined with a path continuation (by sampling the phase function or interface BSDF), using MIS for the weighting. This is analogous to the MIS direct illumination used in many practical path tracers, with the difference that the path can cross a refractive boundary. Note the distinction between this *local MIS*, and the *global MIS* used by the scene-level transport algorithm (a standard path tracer in our results). An illustration of these two techniques, applied to a transmit-reflect-transmit (TRT) configuration, can be found in Figure 7-ab.

Previous work on next-event estimation in scattering volumes through refractive interfaces [Koerner et al. 2016; Walter et al. 2009]

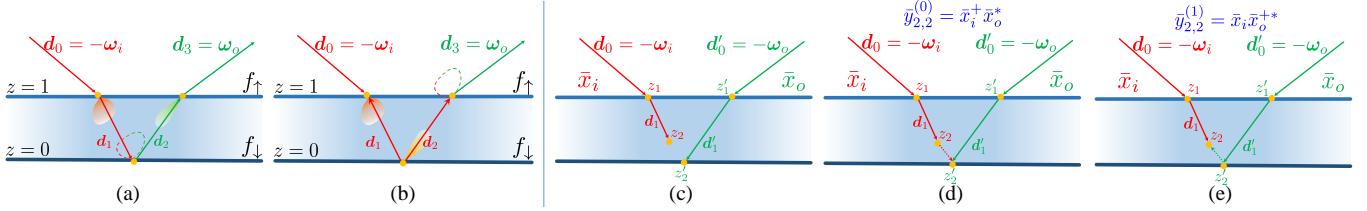


Fig. 7. Our Monte Carlo estimators for BSDF values. (a**b**) Unidirectional estimator uses two path sampling strategies for “shading” a vertex on the bottom layer: (a) sampling the BSDF f_{\uparrow} of the top interface and connecting at the bottom (next event estimation); or (b) sampling d_2 using f_{\downarrow} and connecting at the top (path continuation). These strategies are combined using local MIS. (c**e**) Bidirectional estimator: (c) Two subpaths with initial directions ω_i and ω_o . (d)e Two full light paths constructed by sampling an additional direction from each sub-path.

is related to our scheme, but focuses on arbitrary geometries, which is not necessary in the flat layer setting.

Extending this NEE scheme to cross multiple layer interfaces is somewhat tedious to implement, as care must be taken not to double-count light paths. We instead use the recursive nesting approach to multiple layers (§4.6) when using the unidirectional estimator, which handles these issues automatically.

5.2.2 Bidirectional simulation. Although our unidirectional solution works well in many cases, we introduce a new bidirectional approach that performs even better. Our approach is conceptually similar to bidirectional path tracing (BDPT) but is technically different in several ways due to our position-free path formulation.

Given the incoming and outgoing directions ω_i and ω_o , consider two light transport paths, generated from the light and camera, respectively.

$$\begin{aligned}\bar{x}_i &= (\mathbf{d}_0, z_1, \mathbf{d}_1, \dots, z_s) \\ \bar{x}_o &= (\mathbf{d}'_0, z'_1, \mathbf{d}'_1, \dots, z'_t),\end{aligned}\quad (15)$$

where $\mathbf{d}_0 = -\omega_i$ and $\mathbf{d}'_0 = -\omega_o$ (Figure 7-c). Now we can construct a full light path $\bar{y}_{s,t}$ connecting the s -th vertex in \bar{x}_i and the t -th vertex in \bar{x}_o (assuming the connection between z_s and z'_t does not cross any layer boundary):

$$\bar{y}_{s,t} = (\mathbf{d}_0, \dots, z_{s-1}, \mathbf{d}_{s-1}, z_s, \tilde{\mathbf{d}}, z'_t, -\mathbf{d}'_{t-1}, z'_{t-1}, \dots, -\mathbf{d}'_0). \quad (16)$$

Unlike traditional BDPT, where the connection term between two given subpaths endpoints is fixed, there exists infinitely many valid directions $\tilde{\mathbf{d}}$ connecting z_s and z'_t in our case, which gives us freedom to importance-sample the direction. In practice, we choose $\tilde{\mathbf{d}}$ in two ways by sampling additional directions \mathbf{d}_s and \mathbf{d}'_t by extending the two subpaths with an extra importance sampling step. We set $\tilde{\mathbf{d}}$ to \mathbf{d}_s and $-\mathbf{d}'_t$ respectively. This yields two light paths $\bar{y}_{s,t}^{(0)}$ and $\bar{y}_{s,t}^{(1)}$ (Figure 7-de), thus providing two samples of the path integral. Denote the extended subpaths by

$$\bar{x}_i^+ := (\mathbf{d}_0, z_1, \mathbf{d}_1, \dots, z_s, \mathbf{d}_s), \quad (17)$$

$$\bar{x}_o^+ := (\mathbf{d}'_0, z'_1, \mathbf{d}'_1, \dots, z'_t, \mathbf{d}'_t), \quad (18)$$

and let \bar{x}^* denote the adjoint (reversed) version of a light path \bar{x} , e.g., $\bar{x}_o^{++} = (-\mathbf{d}'_t, z'_t, -\mathbf{d}'_{t-1}, \dots, z'_1, -\mathbf{d}'_0)$.

Let $v(z, \omega, \omega')$ and $s(z, z', \omega)$ be the vertex and segment contributions defined in eqs. (3) and (5). We can easily verify that

$$f(\bar{y}_{s,t}^{(0)}) = f(\bar{x}_i^+) f(\bar{x}_o^*) s(z_s, z'_t, \mathbf{d}_s) v(z'_t, -\mathbf{d}_s, -\mathbf{d}'_{t-1}), \quad (19)$$

$$f(\bar{y}_{s,t}^{(1)}) = f(\bar{x}_i) f(\bar{x}_o^{++}) v(z_s, -\mathbf{d}_{s-1}, -\mathbf{d}'_t) s(z_s, z'_t, -\mathbf{d}'_t). \quad (20)$$

It follows that the two Monte Carlo estimates will be:

$$\frac{f(\bar{y}_{s,t}^{(0)})}{p(\bar{y}_{s,t}^{(0)})} = \frac{f(\bar{x}_i^+)}{p(\bar{x}_i^+)} \frac{f(\bar{x}_o^*)}{p(\bar{x}_o)} s(z_s, z'_t, \mathbf{d}_s) v(z'_t, -\mathbf{d}_s, -\mathbf{d}'_{t-1}) \quad (21)$$

$$\frac{f(\bar{y}_{s,t}^{(1)})}{p(\bar{y}_{s,t}^{(1)})} = \frac{f(\bar{x}_i)}{p(\bar{x}_i)} \frac{f(\bar{x}_o^{++})}{p(\bar{x}_o^{++})} v(z_s, \mathbf{d}_{s-1}, \mathbf{d}'_t) s(z_s, z'_t, -\mathbf{d}'_t), \quad (22)$$

Note that in general $f(\bar{x}_o) \neq f(\bar{x}_o^*)$ and $f(\bar{x}_o^+) \neq f(\bar{x}_o^{++})$ due to non-reciprocal operations such as shading normals; care must be taken to compute correct throughputs of light subpaths, as detailed in Chapter 5 of Veach [1997].

The above discussion assumed a single light and single camera subpath. In practice, we combine all prefixes of the sampled subpaths. In particular, we sample subpaths of length n_i and n_o from the light and camera respectively (the lengths are chosen through Russian roulette):

$$\begin{aligned}\bar{x}_i &= (\mathbf{d}_0, z_1, \mathbf{d}_1, \dots, z_{n_i}, \mathbf{d}_{n_i}), \\ \bar{x}_o &= (\mathbf{d}'_0, z'_1, \mathbf{d}'_1, \dots, z'_{n_o}, \mathbf{d}'_{n_o}).\end{aligned}\quad (23)$$

For all s and t combinations, Eqs. (21) and (22) provide $2n_i n_o$ estimators of $f_l(\omega_i, \omega_o)$. Combining them using MIS gives us our bidirectional estimator for paths of length 2 or more vertices. We handle single vertex paths separately. The details of MIS weighting are discussed in the supplementary document.

5.3 Pdf estimation

Another important operation for practical BSDF models is to evaluate the probability density for sampling provided incoming and outgoing directions. That is, to evaluate $p(\omega_o | \omega_i)$, the probability density of ω_o given ω_i (assuming that the sampling process draws ω_o and fixes ω_i). This operator is used for weight computation in multiple importance sampling (using balance or power heuristics), a crucial technique for generating low-noise results using scene-level Monte Carlo rendering techniques. Note that this pdf is in the solid angle measure; it is a marginal pdf distinct from the path pdf $p(\bar{x})$.

Although $p(\omega_o | \omega_i)$ is usually easily available for traditional analytical BSDFs, no closed-form pdf exists in our case. Instead, the

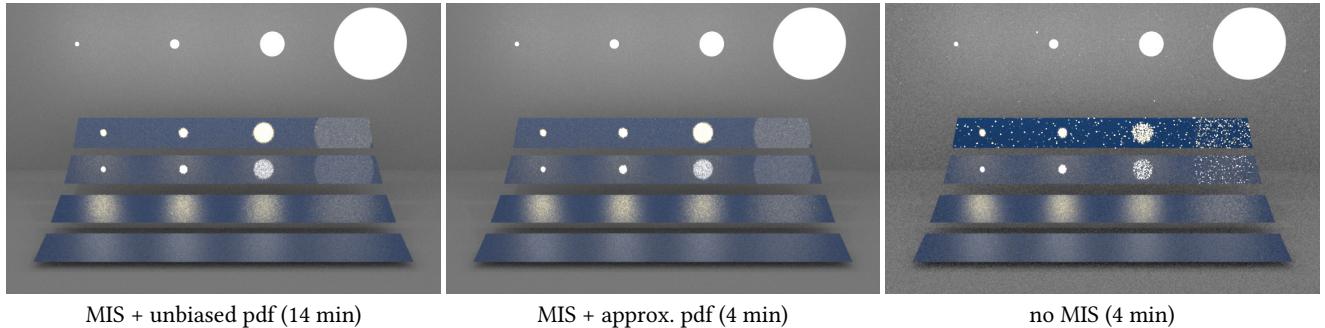


Fig. 8. Multiple importance sampling using our BSDFs. The slabs in this figure use a layered material with rough dielectric on the top, rough gold conductor on the bottom, and blueish homogeneous scattering medium in between. **Left:** Using the unbiased pdf from §5.3.1 for MIS in a traditional global path tracer. **Middle:** Using the approximate pdf from §5.3.2 is faster and gives equivalent quality. **Right:** Using no MIS is clearly inferior.

pdf evaluation has comparable form to the BSDF evaluation itself. It can be expressed using another position-free path integral:

$$p(\omega_o \mid \omega_i) = \int_{\Omega(\omega_i, \omega_o)} \mathcal{P}(\bar{x}) d\mu(\bar{x}), \quad (24)$$

where

$$\mathcal{P}(\bar{x}) := \left(\prod_{j=1}^k p(d_j \mid z_j, d_{j-1}) \right) \left(\prod_{j=1}^{k-1} p(z_{j+1} \mid z_j, d_{j-1}) \right), \quad (25)$$

with k denoting the number of vertices in \bar{x} . Note that $d_k = \omega_o$.

We introduce two nondeterministic methods, an unbiased (§5.3.1) and a fast approximate approach (§5.3.2), to estimate $p(\omega_o \mid \omega_i)$. These operations are not used in standard Monte Carlo light transport and are new, to our knowledge. In practice, the approximate approach can be used when exact estimations are unnecessary (as is the case for a global path tracer with MIS, which we use for our results). Note that the estimated $p(\omega_o \mid \omega_i)$ is only ever used for MIS weight computation. We never use approximate path pdfs for Monte Carlo estimates, as this would introduce bias. Our BSDF value estimators directly return path throughput with accurate pdf factored in.

5.3.1 Unbiased pdf estimation. Both our Monte Carlo estimators introduced in §5.2.1 and §5.2.2 can be adapted to estimate the path integral in Eq. (24) in an unbiased manner. For instance, the estimators given by Eqs. (21) and (22) simply require a replacement of f by \mathcal{P} , and become:

$$\frac{\mathcal{P}(\bar{y}_{s,t}^{(0)})}{p(\bar{y}_{s,t}^{(0)})} = \frac{\mathcal{P}(\bar{x}_o^*)}{p(\bar{x}_o)} p(d_s \mid z'_t, -d'_{t-1}) p(z'_t \mid z_s, d_s), \quad (26)$$

$$\frac{\mathcal{P}(\bar{y}_{s,t}^{(1)})}{p(\bar{y}_{s,t}^{(1)})} = \frac{\mathcal{P}(\bar{x}_o^{+*})}{p(\bar{x}_o^+)} p(d'_t \mid z_s, d_{s-1}) p(z_s \mid z'_t, d'). \quad (27)$$

Note that some cancellation occurs because $p(x_i) = \mathcal{P}(x_i)$, but in general $p(x_o) \neq \mathcal{P}(x_o^*)$.

When jointly estimating the path integrals for the BSDF value (8) and the conditional probability (24), the light transport paths \bar{x} need

to be sampled *independently* to ensure unbiasedness. Please refer to the supplemental document for a proof.

5.3.2 Approximate pdf estimation. Although the adapted estimators defined in 5.3.1 provide unbiased pdf estimations, they introduce computational overhead comparable to the BSDF evaluation itself. Thus, for applications where unbiased pdfs are unnecessary, we introduce an approximation to accelerate the pdf estimation process. The key idea is to only consider short paths reflecting/refracting from interfaces, as these events have the largest effect on the pdf lobe shape, and add a constant (Lambertian) term to approximate the effect of volume scattering and longer paths.

In practice, we run Monte Carlo simulation on a simplified layer configuration where all volumetric media are removed. We further limit the maximal number of vertices on the light paths to $(2L + 1)$ when $\omega_i \cdot \omega_o > 0$ (i.e., $f_l(\omega_i, \omega_o)$ captures reflection) and $(L + 1)$ when $\omega_i \cdot \omega_o < 0$ (i.e., $f_l(\omega_i, \omega_o)$ captures transmission) where L denotes the number of layers. Lastly, we add a small constant term to the estimation result. The exact scaling of this term is not important for MIS weighting (as it will be overwhelmed by the pdfs of sharply peaked lobes) and we found setting it to 0.1 works well.

See Figure 9 for validation of the above pdf approaches against ground truth, and Figure 8 for a comparison between renderings using the unbiased and approximated pdf estimation results. All the other results in our paper are using approximated PDF for MIS. Unbiased PDF is much slower, because it requires long light paths, and has to be computed twice per shading event.

6 APPLICATIONS AND RESULTS

In this section, we first provide experimental validations (§6.1) and then showcase our method on a number of applications and demonstrate its effectiveness (§6.2). All the renderings are generated using the Mitsuba physically based renderer [Jakob 2010] with our layered model implemented as a BSDF plugin. Please see the accompanying video for animated versions of several results.

All the multi-layer results in the paper use our bidirectional estimator with the explicit implementation (although our BSDF plugin

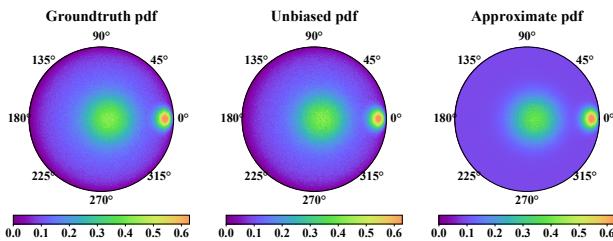


Fig. 9. Validation of our pdf estimates. The visualization applies a $\log(1+x)$ map for better shape perception. **Left:** Ground truth by sampling and binning. **Middle:** Using the unbiased pdf from §5.3.1. **Right:** Using the approximate pdf from §5.3.2 matches the shape of the most important features and approximates longer paths and volume scattering as diffuse.

also supports nesting BSDFs). This is because the former runs faster, as seen in Figure 15-c.

6.1 Validations

6.1.1 Cross validation. In Figures 4 and 9 as well as the supplemental material, we cross-validate our Monte Carlo estimators depicted in §5.2 by comparing our estimated BSDFs/pdfs to references generated using forward sampling (§5.1) and binning. Notice that the sampling procedure is a straightforward process that requires none of the complexity introduced by our path formulation and estimators.

6.1.2 White furnace tests. We conducted a few “white furnace tests” to demonstrate the energy conservation of our layered BSDFs (Figure 10). For all these examples, the BSDFs are constructed such that no energy is lost due to light-layer interactions. Under constant lighting (where identical amount of light comes from all directions), the object becomes invisible, demonstrating that our layered BSDFs indeed conserve energy properly.

6.2 Main Results

6.2.1 Application: Coating thickness/normal variation. Figure 11 shows renderings of a globe with a dielectric coating on top of a metallic substrate. In this example, both interfaces are colorless and the layer medium has a blue tint. In Figure 11-a, both interfaces are smooth, creating two overlapped reflections of the environment map with different amounts of blur. In Figure 11-b, the top interface of the globe is smooth, leading to one clear reflection. On the bottom (metallic) interface, we use a detailed height field to drive the normal variation as well as the medium thickness. The high-frequency variation of normal direction has resulted in detailed highlights on the bottom surface. Further, due to varying amounts of attenuation at different thickness, these highlights exhibit different colors: reflections from greater depths become darker and more saturated. In Figure 11-c, the height variation is instead applied to the top dielectric interface, causing the clear reflection of the environment to be replaced by a blurred one. Further, since the areas under the continents now have larger thickness, their colors become more saturated. Our layered BSDF model is capable of producing all these

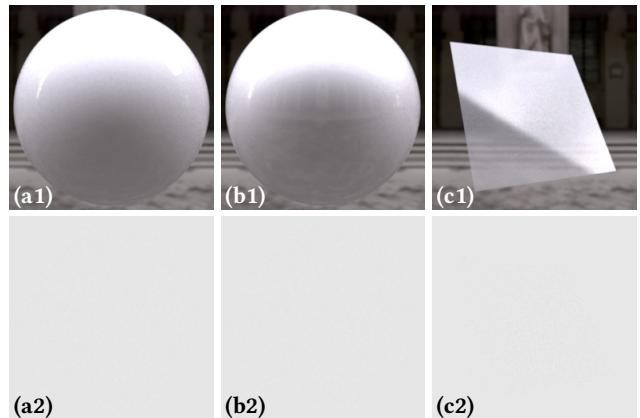


Fig. 10. White furnace tests. We demonstrate that our BSDFs conserve energy properly via three layered BSDF examples respectively given by **(a)** a dielectric and a diffuse interface; **(b)** a dielectric and a conductor interface and participating medium; **(c)** two dielectric interfaces and participating medium in between. All the interfaces and media have albedo one (so no energy is lost due to light-layer interactions). For each example, a simple object is rendered under both environmental (1) and constant (2) illuminations.

appearances using a simple set of parameters (thickness, roughness and medium absorption) in conjunction with spatial variation.

6.2.2 Application: Complex thin sheet transmission. Our physically based BSDF is capable of accurately modeling not only reflection but also transmission. Figure 12 shows two examples. The top row contains an example flat surface rendered with our layered BSDF under varying illuminations. This model involves dielectric interfaces with spatially varying roughnesses and a normal map applied to the front surface. The optical thickness at each location is obtained by multiplying a base density, which varies across the color channels, by the geometric height field matching the normal map. In other words, the optical densities (mean free paths) are spectrally varying, which results in subtle color variations across the surface (especially for transmitted light), a phenomenon that would be challenging to model accurately using existing BSDF models. Note again that all of these effects come from the BSDF model, as the scene geometry is a simple flat polygon.

The bottom row of Figure 12 shows renderings of a magnifying lens filled with scattering media with spatially varying thickness (which captures the shape of real convex lens). Note that the scene geometry is still just a flat surface. When coupled with different phase functions (Heney-Greenstein and von-Mises-Fisher, with different forward scattering parameters), a range of spatially varying and physically plausible blurring effects can be achieved.

Please see the supplemental images and video for more variations with similar configurations.

6.2.3 Application: Anisotropic layer media for fabrics. Our layered BSDF allows any phase functions within volumetric scattering layers, including anisotropic microflake phase functions [Heitz et al. 2015; Jakob et al. 2010; Zhao et al. 2011] capable of representing

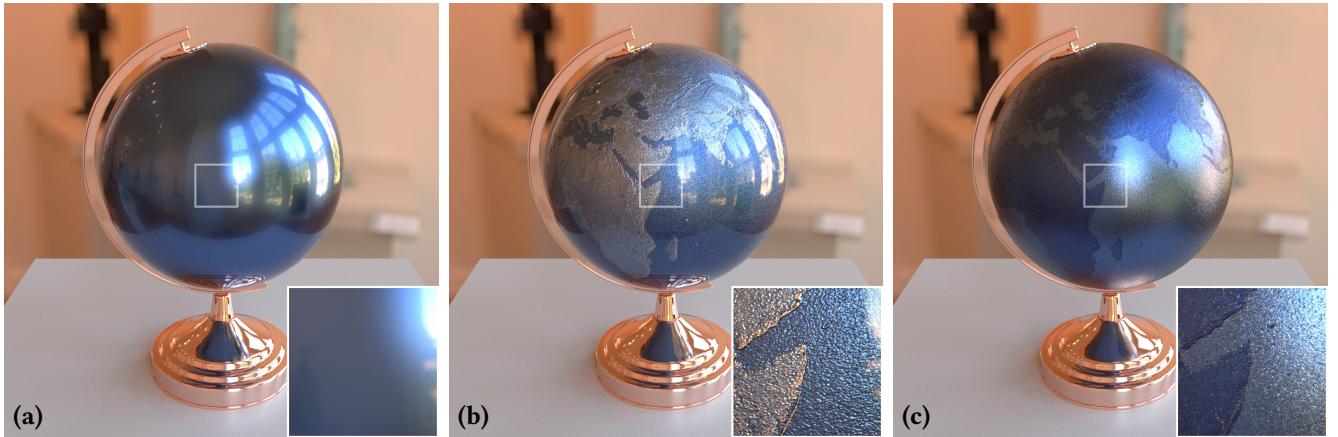


Fig. 11. Top vs. bottom height variation. Thanks to the physically-based nature of our layered BSDF model, manipulating heights on its top and bottom interfaces has greatly varying effects on the final appearance. The height variation drives both normals and thickness differences (and thus medium absorption). (a) No height variation. (b) Height variation applied to the bottom interface. (c) Height variation applied to the top interface.

Table 2. Render times of all our results (using our “unidir.” and “bidir.” estimators) as well as baseline models with “trivial” BSDFs (that require no stochastic evaluation). All the multi-layer models are described using nesting BSDFs for the unidirectional estimator and the explicit implementations for the bidirectional one. The baseline models exhibit different appearances and are created solely for performance comparison. All the timings are converted to a 6-core Intel i7-6800K CPU time, and those between parentheses indicate render time per mega-pixel. The numbers in bold correspond to methods used for creating the paper figures. Please refer to the supplemental material for all the other renderings.

	Image size	Spp	Render time		
			Unidir.	Bidir.	Trivial
Fig. 1 (a)	3000×2000	1024	2.5 h (25 m)	2.2 h (22 m)	38 m (6.3 m)
Fig. 11 (b)	1024×1024	256	2.2 m (2.1 m)	2.6 m (2.5 m)	1.3 m (1.2 m)
Fig. 12, top	800×1200	512	15.2 m (7.9 m)	24 m (12.5 m)	2.4 m (1.3 m)
Fig. 12, bot.	512×512	1024	6.4 m (6.1 m)	13 m (12.6 m)	1.6 m (1.5 m)
Fig. 13 (a)	876×584	256	1.1 m (2.2 m)	1.4 m (2.7 m)	0.6 m (1.1 m)
Fig. 13 (b)	876×584	256	1.1 m (2.2 m)	1.4 m (2.7 m)	0.5 m (0.9 m)
Fig. 13 (c)	876×584	256	2.5 m (4.9 m)	5.4 m (10.5 m)	0.5 m (0.9 m)
Fig. 14 (b)	640×540	256	1.5 m (4.3 m)	1.9 m (5.5 m)	0.5 m (1.4 m)
Fig. 15 (a)	1200×1400	256	6.7 m (4.0 m)	12 m (7.1 m)	3.7 m (2.2 m)
Fig. 15 (b)	1200×1400	256	7.0 m (4.2 m)	13 m (7.7 m)	3.7 m (2.2 m)
Fig. 15 (c)	1200×1400	256	67 m (40 m)	20 m (12 m)	4.7 m (2.8 m)

fabrics. Figure 13 shows three fabrics modeled using our model with “null” top and bottom interfaces (ones that allow light to travel through without reflecting or refracting it) and anisotropic layer media with spatially varying albedo and flake orientations (the optical density does not vary in these examples, though it could). The satin weave shows well aligned yarns have created smooth and strongly anisotropic highlights. The twill pattern has warp and weft yarns in different colors, leading to dual colored highlights. The velvet exhibits strong grazing-angle highlights, an effect that is challenging to model using traditional BSDF models. Our model successfully captures all the diverse appearances from all three fabrics and produces convincing impressions of these materials.

Figure 14 shows a fabric rendered using fiber orientation data acquired by micro-CT imaging [Zhao et al. 2011]. Our rendering uses a fiber orientation map derived from the full data, and matches the full volumetric simulation fairly closely, while being 40 times faster. The speedup is because ours is still a flat BSDF model with parameter mapping, as opposed to full volumetric tracing that requires expensive ray marching through massive data.

6.2.4 Application: Multiple layers. Lastly, in Figure 15, we show rendered results of a kettle with varying layer configurations. In column (a), the material has a single transparent water layer with a dielectric interface on the top and a metallic surface on the bottom. Both interfaces are normal mapped to capture the water drops and

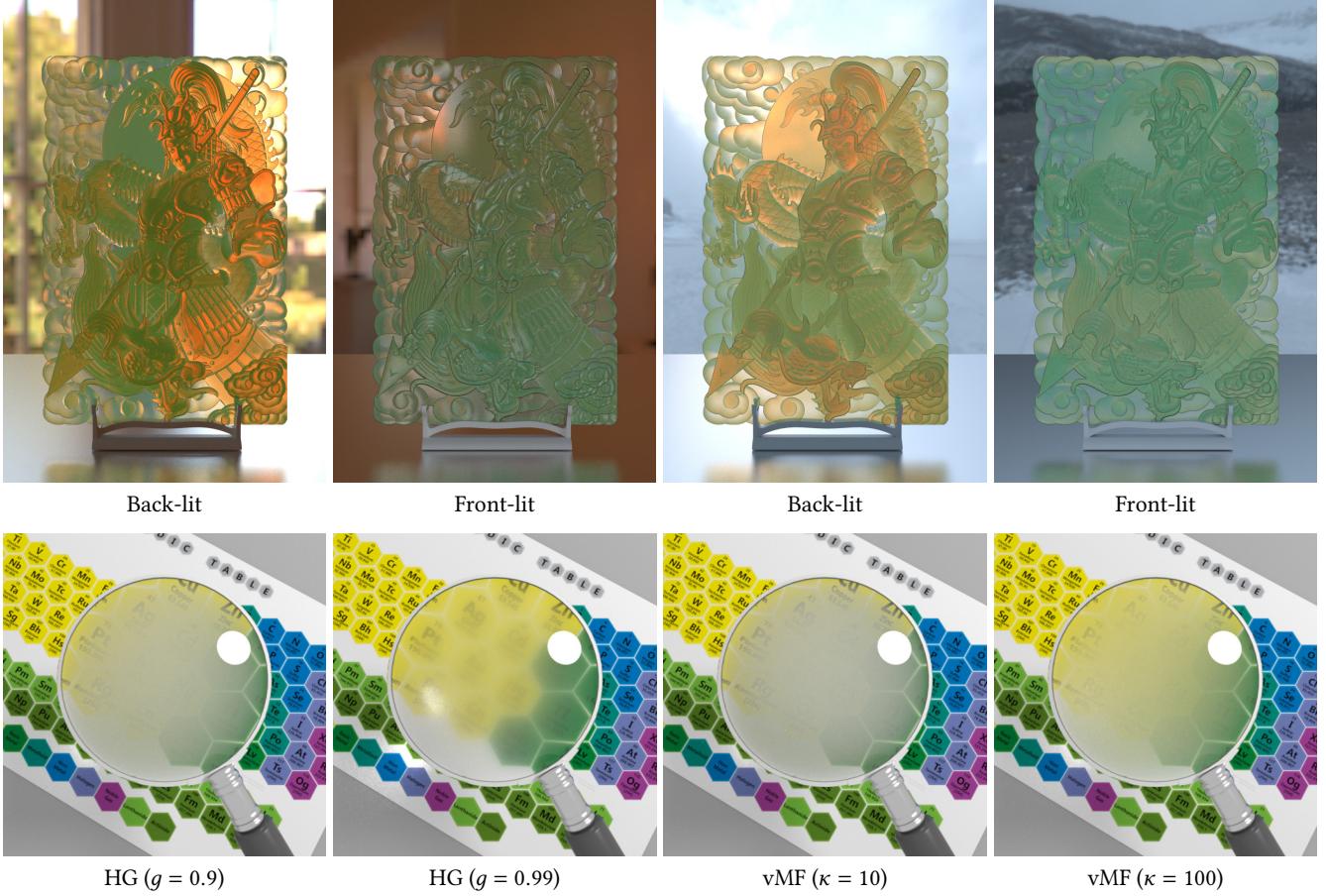


Fig. 12. Reflection and transmission: Our BSDF models are capable of accurately capturing not only reflection but also transmission from thin layers. **Top:** A flat surface rendered with our layered BSDF under varying illuminations. This model involves dielectric interfaces with spatially varying roughnesses, normal maps, and thickness. The optical densities (mean free paths) are spectrally varying, which results in subtle color variations across the surface. Note that the color (albedo) is not varying. **Bottom:** A flat surface with a layered BSDF of spatially varying thickness (which captures the shape of real convex lens). A range of spatially varying and physically plausible blurring effects can be achieved by varying phase functions.

the scratches, respectively. In column (b), the material shares the same bottom surface as in (a) but has a smooth top interface and a translucent coating layer with spatially varying optical thickness and albedo, making only part of the bottom surface directly visible. Lastly, in column (c), the material has a dual-layer configuration by stacking the layers from (a) on top of that from (b). Our method offers the flexibility to conveniently model all three cases with the last one described using the explicit implementation depicted in §4.6.

6.3 Performance

The Monte Carlo processes for sampling and evaluating our BSDFs do introduce computational overhead. Table 2 lists the performance numbers of all our results. Further, we provide baseline timings using “trivial” BSDFs (that require no stochastic evaluation) to the same scene geometries. Our performance does degrade with the

presence of optically thick and highly scattering media. However, as already demonstrated in Figure 2, rendering using our model is still significantly faster than explicitly simulating light transport in layered geometries.

6.4 Limitations and future work

Our model relies on the assumption of thin flat layers (Figure 5) and cannot capture effects caused by geometric or optical variations at the global scale. Examples include internal caustics and shadowing arising from major normal variations and color bleeding caused by light scattering through media with varying colors. Generalizing our technique to include bidirectional subsurface scattering distribution functions (BSSRDFs) is an interesting further topic. In addition, as our model simulates subsurface scattering using Monte Carlo path tracing, the performance may degrade with the presence of optically thick layers with many scattering events. Using fast approximated

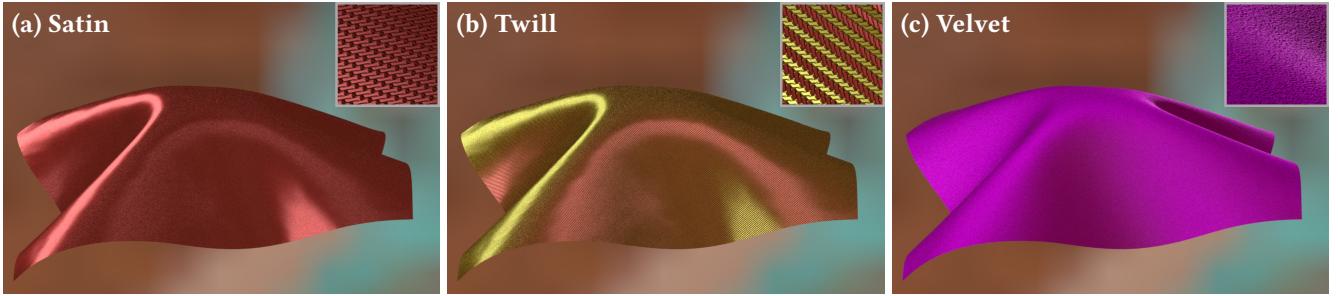


Fig. 13. Anisotropic media within layers. Our layered BSDF offers the generality to use anisotropic layer media with microflake phase functions. This example shows three fabrics modeled with our BSDF model with anisotropic layer media: (a) satin; (b) twill; and (c) velvet.

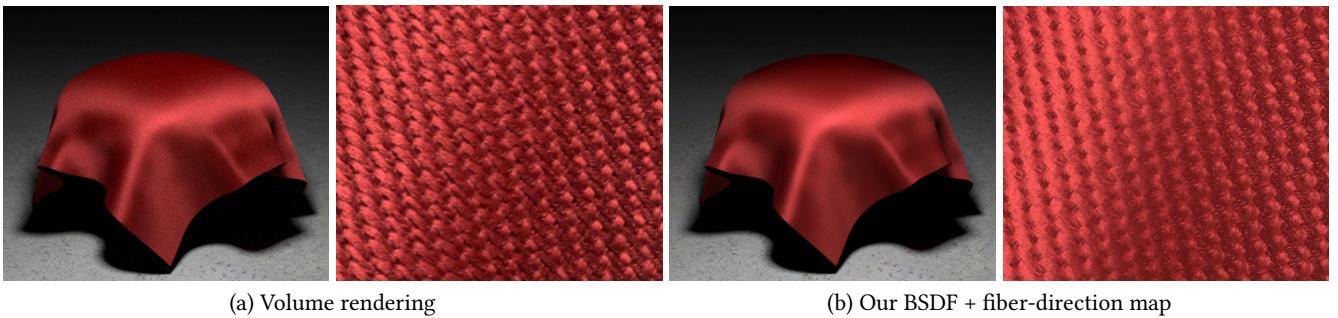


Fig. 14. Comparison to volumetric cloth. (a) Images rendered from micro-CT volumetric data, using the microflake phase function. (b) Renderings using our approach using a single microflake volumetric layer, where we are using fiber direction maps extracted from the volumetric data. Our rendering is 40× faster than the volumetric simulation.

solutions such as [Frisvad et al. 2014; Jensen et al. 2001] to capture multiple scattering may be a useful extension. Lastly, since we model light transport using traditional radiative transfer, wave effects such as thin film interference are not handled. An interesting challenge is to integrate wave optics into our model to accurately and efficiently handle light interference and phase shifts.

7 CONCLUSION

In this paper, we introduced a new BSDF model to capture the appearance of layered materials. Inside the evaluation and sampling routines of the layered BSDF, we run a Monte Carlo simulation of light transport within flat slabs. This is substantially faster than explicitly constructing the layer geometry, but also allows constructing light transport paths that would not easily be available to a generic light transport algorithm, due to our new position-free path formulation.

Within this framework, we introduced unbiased Monte Carlo techniques analogous to a forward path tracer with next event estimation (NEE) and a fully bidirectional estimator. We demonstrated the capabilities of our solution on a number of examples, featuring multiple layers with surface and volumetric scattering, surface and phase function anisotropy, and spatial variation in all parameters. This leads to the first BSDF layering solution that offers unbiased accuracy and full flexibility in setting the layer properties.

REFERENCES

- Laurent Belcour. 2018. Efficient Rendering of Layered Materials using an Atomic Decomposition with Statistical Operators. *ACM Trans. Graph.* (2018).
- R. L. Cook and K. E. Torrance. 1982. A Reflectance Model for Computer Graphics. *ACM Trans. Graph.* 1, 1 (1982), 7–24.
- Craig Donner, Tim Weyrich, Eugene d’Eon, Ravi Ramamoorthi, and Szymon Rusinkiewicz. 2008. A Layered, Heterogeneous Reflectance Model for Acquiring and Rendering Human Skin. *ACM Trans. Graph.* 27, 5 (2008), 140:1–140:12.
- Jeppe Revall Frisvad, Toshiya Hashisuka, and Thomas Kim Kjeldsen. 2014. Directional Dipole Model for Subsurface Scattering. *ACM Trans. Graph.* 34, 1 (2014), 5:1–5:12.
- Jie Guo, Jinghui Qian, Yanwen Guo, and Jingui Pan. 2017. Rendering Thin Transparent Layers with Extended Normal Distribution Functions. *IEEE transactions on visualization and computer graphics* 23, 9 (2017), 2108–2119.
- Pat Hanrahan and Wolfgang Krueger. 1993. Reflection from Layered Surfaces Due to Subsurface Scattering. In *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH ’93)*. ACM, New York, NY, USA, 165–174. <https://doi.org/10.1145/166117.166139>
- Eric Heitz, Jonathan Dupuy, Cyril Crassin, and Carsten Dachsbacher. 2015. The SGX Microflake Distribution. *ACM Trans. Graph.* 34, 4 (2015), 48:1–48:11.
- Eric Heitz, Johannes Hanika, Eugene d’Eon, and Carsten Dachsbacher. 2016. Multiple-scattering Microfacet BSDFs with the Smith Model. *ACM Trans. Graph.* 35, 4 (2016), 58:1–58:14.
- Wenzel Jakob. 2010. Mitsuba renderer. (2010). <http://www.mitsuba-renderer.org>.
- Wenzel Jakob, Adam Arbree, Jonathan T. Moon, Kavita Bala, and Steve Marschner. 2010. A Radiative Transfer Framework for Rendering Materials with Anisotropic Structure. *ACM Trans. Graph.* 29, 4 (2010), 53:1–53:13.
- Wenzel Jakob, Eugene d’Eon, Otto Jakob, and Steve Marschner. 2014. A Comprehensive Framework for Rendering Layered Materials. *ACM Trans. Graph.* 33, 4 (2014), 118:1–118:14.
- Henrik Wann Jensen, Stephen R. Marschner, Marc Levoy, and Pat Hanrahan. 2001. A Practical Model for Subsurface Light Transport. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*. 511–518.
- David Koerner, Jan Novák, Peter Kutz, Ralf Habel, and Wojciech Jarosz. 2016. Subdivision Next-event Estimation for Path-traced Subsurface Scattering. In *Proceedings of*



Fig. 15. Multi-layer BSDF. This result shows renderings of a kettle described with: (a) a single transparent layer with a dielectric top interface capturing the water drops over a conducting bottom surface with scratches; (b) a single translucent layer with spatially varying optical thicknesses and albedo over the same bottom surface of (a); (c) a dual layer configuration created by stacking the transparent layer (a) over the translucent one (b).

- the Eurographics Symposium on Rendering: Experimental Ideas & Implementations (EGSR '16)*, 91–96.
- Vincent Schüssler, Eric Heitz, Johannes Hanika, and Carsten Dachsbacher. 2017. Microfacet-based Normal Mapping for Robust Monte Carlo Path Tracing. *ACM Trans. Graph.* 36, 6 (Nov. 2017), 205:1–205:12.
- Jos Stam. 2001. An Illumination Model for a Skin Layer Bounded by Rough Surfaces. In *Proceedings of the 12th Eurographics Workshop on Rendering Techniques*. Springer-Verlag, London, UK, UK, 39–52. <http://dl.acm.org/citation.cfm?id=647653.732287>
- Eric Veach. 1997. *Robust Monte Carlo Methods for Light Transport Simulation*. Ph.D. Dissertation. Stanford University.
- Bruce Walter, Stephen R. Marschner, Hongsong Li, and Kenneth E. Torrance. 2007. Microfacet Models for Refraction Through Rough Surfaces. In *Proceedings of the 18th Eurographics Conference on Rendering Techniques*. 195–206.
- Bruce Walter, Shuang Zhao, Nicolas Holzschuch, and Kavita Bala. 2009. Single Scattering in Refractive Media with Triangle Mesh Boundaries. *ACM Trans. Graph.* 28, 3 (2009), 92:1–92:8.
- Andrea Weidlich and Alexander Wilkie. 2007. Arbitrarily Layered Micro-facet Surfaces. In *Proc. 5th Int. Conf. Comput. Graph. Interactive Techn.* 171–178.
- Tizian Zeltner and Wenzel Jakob. 2018. The Layer Laboratory: A Calculus for Additive and Subtractive Composition of Anisotropic Surface Reflectance. *ACM Trans. Graph.* (2018).
- Shuang Zhao, Wenzel Jakob, Steve Marschner, and Kavita Bala. 2011. Building Volumetric Appearance Models of Fabric Using Micro CT Imaging. *ACM Trans. Graph.* 30, 4 (2011), 44:1–44:10.

APPENDIX

A DETAILED DERIVATIONS

We now provide detailed derivations for the key equations in §4.

Position-free radiative transfer equation. Traditionally, the radiative transfer equation (RTE) involves an integral over free-flight distance t :

$$L_v(z, \omega) = S(z, \omega) + \int_0^{t'} \exp(-t\sigma_t) \int_{S^2} \hat{f}_p(\omega', \omega) L_v(z', \omega') d\omega' dt, \quad (28)$$

where $z' := z - t \cos \omega$ and t' denotes the distance between z and the closest layer boundary. Since $t = (z - z')/\cos \omega$, changing the integration variable from t to z' in Eq. (28) yields an additional factor of $(\cos \omega)^{-1}$ which in turn gives our position-free RTE (9). Notice that the change-of-variable ratio only appears within the integration (and not in the source term S).

Cosines in path contribution. The contribution f of a light path \bar{x} can be obtained by repeatedly expanding the rendering equation (11) and our position-free RTE (9).

Similar to the traditional path integral formulation, for each vertex z_i corresponding to an interface event (i.e., reflection or refraction), a cosine term $|\cos \mathbf{d}_i|$ is needed to ensure the measure of projected solid angle. On the other hand, a segment of our light path connecting two depths z_i and z_{i+1} via direction \mathbf{d}_i can yield an additional $|\cos \mathbf{d}_i|^{-1}$ when z_{i+1} corresponds to a volumetric scattering. Thus, for each i , the path contribution involve a factor of $|\cos \mathbf{d}_i|^{\alpha_i}$ with:

- $\alpha_i = 1$ if z_i and z_{i+1} are both on interfaces;
- $\alpha_i = 0$ (i.e., no $\cos \mathbf{d}_i$ term) if (i) z_i is volumetric and z_{i+1} lies on an interface (so that no $\cos \mathbf{d}_i$ terms appear during expansion), or (ii) z_i is interfacial and z_{i+1} is volumetric (so that both $|\cos \mathbf{d}_i|$ and $|\cos \mathbf{d}_i|^{-1}$ are present, canceling out each other);
- $\alpha_i = -1$ if z_i and z_{i+1} are both volumetric vertices.

Eq. (6) provides a compact way to encode these rules.

B EFFICIENT WEIGHT COMPUTATION

Weights of Light Transport Paths.

Given a light subpath \bar{x}_i and a camera subpath \bar{x}_o with n_i and n_o vertices respectively, our bidirectional estimator combines $2n_i n_o$ estimators of the form $f(\bar{y}_{s,t}^{(u)})/p_{s,t}^{(u)}(\bar{y}_{s,t}^{(u)})$ with $s \in \{1, 2, \dots, n_i\}$, $t \in \{1, 2, \dots, n_o\}$, and $u \in \{0, 1\}$ via the multiple importance sampling (MIS) framework. This yields a combined estimator:

$$\sum_{s=1}^{n_i} \sum_{t=1}^{n_o} \sum_{u=0}^1 w_{s,t}^{(u)}(\bar{y}_{s,t}^{(u)}) \frac{f(\bar{y}_{s,t}^{(u)})}{p_{s,t}^{(u)}(\bar{y}_{s,t}^{(u)})}, \quad (29)$$

where the weight $w_{s,t}^{(u)}$, when using the balanced heuristics [Veach 1997], is given by

$$w_{s,t}^{(u)}(\bar{y}_{s,t}) = \left(\sum_{s'=1}^{s+t-1} \sum_{u'=0}^1 \frac{p_{s',s+t-s'}^{(u')}(\bar{y}_{s,t})}{p_{s,t}^{(u)}(\bar{y}_{s,t})} \right)^{-1} \quad (30)$$

for any path $\bar{y}_{s,t}$ with $(s+t)$ vertices.

Notice that, compared to standard bidirectional path tracing that combines $n_i n_o$ estimators, our position-free formulation offers twice the number of estimators since the direction connecting two depths is not unique (see Eq. (16) of the main paper).

Efficient Weight Computation.

Computing Eqs. (29) and (30) for all s and t naïvely has a time complexity of $O(n_i n_o (n_i + n_o))$ and is too slow to be practical. We now present our method that runs in $O(n_i n_o)$ time. Our approach is conceptually similar to Veach’s method for standard BDPT but differs in the exact mathematical form due to our position-free path formulation (see §4 for the paper).

Let $\bar{y}_{s,t} = (\mathbf{d}_0, z_1, \mathbf{d}_1, \dots, z_n, \mathbf{d}_n)$ with $n = s+t$. For all $s', t' \in \{1, 2, \dots, n\}$, define

$$p_{s'}^{(0)} := \prod_{i=1}^{s'-1} p(\mathbf{d}_i | z_i, \mathbf{d}_{i-1}) p(z_{i+1} | z_i, \mathbf{d}_i), \quad (31)$$

$$p_{t'}^{(1)} := \prod_{i=n-t'+1}^{n-1} p(-\mathbf{d}_i | z_{i+1}, -\mathbf{d}_{i+1}) p(z_i | z_{i+1}, -\mathbf{d}_i), \quad (32)$$

which denote the probability for constructing two subpaths containing the first s' and last t' vertices of \bar{y} , respectively. Then, for all u' , s' and t' , it holds that

$$p_{s',t'}^{(u')}(\bar{y}_{s,t}) = p_{s'}^{(0)} p_{t'}^{(1)} q_{s'}^{(u')}, \quad (33)$$

where

$$q_{s'}^{(u')} := \begin{cases} p(\mathbf{d}_{s'} | z_{s'}, \mathbf{d}_{s'-1}) & \text{if } u = 0, \\ p(-\mathbf{d}_{s'} | z_{s'+1}, -\mathbf{d}_{s'+1}) & \text{if } u = 1. \end{cases} \quad (34)$$

It follows that

$$\frac{p_{s',t'}^{(u')}(\bar{y}_{s,t})}{p_{s,t}^{(u)}(\bar{y}_{s,t})} = \frac{p_{s'}^{(0)} p_{t'}^{(1)} q_{s'}^{(u')}}{p_s^{(0)} p_t^{(1)} q_s^{(u)}}. \quad (35)$$

Note that, for any $s' < s$, we have

$$p_{s',t'}^{(u')}(\bar{y}_{s,t}) = p_{s'}^{(0)} \frac{p_{t'}^{(1)}}{p_{t+1}^{(1)}} q_{s'}^{(u')} p_{t+1}^{(1)}. \quad (36)$$

It follows that

$$\sum_{s'=1}^{s-1} \sum_{u'=0}^1 \frac{p_{s',t'}^{(u')}(\bar{y}_{s,t})}{p_{s,t}^{(u)}(\bar{y}_{s,t})} = \frac{p_{t+1}^{(1)}}{p_t^{(1)} q_s^{(u)}} \underbrace{\sum_{s'=1}^{s-1} \sum_{u'=0}^1 \frac{p_{s'}^{(0)} p_{t+1}^{(1)} q_{s'}^{(u')}}{p_s^{(0)}}}_{=: P_s^{(0)}}. \quad (37)$$

Since

$$\frac{p_{t'}^{(1)}}{p_{t+1}^{(1)}} = \prod_{i=s'+1}^{s-1} p(-\mathbf{d}_i | z_{i+1}, -\mathbf{d}_{i+1}) p(z_i | z_{i+1}, -\mathbf{d}_i), \quad (38)$$

it is easy to verify that $P_s^{(0)}$ depends only on depths $z_{s'}$ and directions $\mathbf{d}_{s'}$ with $s' \leq s$, which are all from the subpath \bar{x}_i . Further, $P_{s'}^{(0)}$ remains constant for all paths $\bar{y}_{s,t}$ with $s > s'$. This allows us to precompute $P_s^{(0)}$ using \bar{x}_i for $s = 1, 2, \dots, n_i$. To this end, $P_s^{(0)}(\bar{y})$ can be efficiently evaluated using the following relation:

$$P_s^{(0)} = \begin{cases} 0 & (s = 0), \\ \frac{p_{s-1}^{(0)}}{p_s^{(0)}} \left(P_{s-1}^{(0)} \frac{p_{t+2}^{(1)}}{p_{t+1}^{(1)}} + \sum_{u'} q_{s-1}^{(u')} \right) & (s > 1). \end{cases} \quad (39)$$

Using Eq. (39), we can compute $P_s^{(0)}(\bar{x}_i)$ for $s = 1, 2, \dots, n_i$ in $O(n_i)$ time.

Similarly, for all $t' < t$, we have

$$\sum_{t'=1}^{t-1} \sum_{u'=0}^1 \frac{p_{s',t'}^{(u')}(\bar{y}_{s,t})}{p_{s,t}^{(u)}(\bar{y}_{s,t})} = \underbrace{\frac{p_{s+1}^{(0)}}{p_s^{(0)} q_s^{(u)}} \sum_{t'=1}^{t-1} \sum_{u'=0}^1 \frac{\frac{p_{s'}^{(0)}}{p_{s+1}^{(0)}} p_{t'}^{(1)} q_{n-t'}^{(u')}}{p_t^{(1)}}}_{=: P_t^{(1)}}, \quad (40)$$

where $P_t^{(1)}$ only depends on \bar{x}_o can be computed in $O(n_o)$ time.

With both $P_s^{(0)}$ and $P_t^{(1)}$ precomputed, Eq. (30) becomes

$$w_{s,t}^{(u)}(\bar{y}_{s,t}) = \left(1 + P_s^{(0)} + P_t^{(1)} + \sum_{u'=0}^1 \frac{p_{s-1,t+1}^{(u')}(\bar{y}_{s,t}) + p_{s+1,t-1}^{(u')}(\bar{y}_{s,t})}{p_{s,t}^{(u)}(\bar{y}_{s,t})} \right)^{-1}, \quad (41)$$

which can be computed in constant time. This leads to a full bidirectional estimator with time complexity $O(n_i n_o)$.

C MIS WITH STOCHASTIC FUNCTION AND WEIGHT EVALUATION

Introduction.

While Monte Carlo integration and multiple importance sampling (MIS) are widely used in practice, we use extended versions of these techniques: our MIS weighting is based on approximate (not exact) pdfs, and our weight and function evaluation are both stochastic (i.e. they consume additional random numbers, and are equal to the true weight and function value only in expectation). For this reason, we review standard Monte Carlo and MIS estimators, and show that our extensions still lead to unbiased results.

Monte Carlo estimator.

Let $f(x)$ be an integrable function on domain D , and let X be a random variable on domain D with probability distribution $p(x)$, such that $p(x) > 0$ whenever $f(x) \neq 0$. An integral

$$I = \int_D f(x) dx \quad (42)$$

can be approximated by the unbiased estimator

$$X_f = \frac{f(X)}{p(X)}. \quad (43)$$

It is easy to see that X_f is an unbiased estimate of I :

$$E_X[X_f] = \int_D p(x) \frac{f(x)}{p(x)} dx = \int_D f(x) dx = I. \quad (44)$$

Note, the cancellation of $p(x)$ is always possible due to the assumption that $p(x) > 0$ for all x where $f(x)$ is non-zero.

Combining estimators through MIS.

Multiple importance sampling (MIS) combines two different sampling strategies (random variables) X_1 and X_2 on D , with pdfs $p_1(x)$ and $p_2(x)$, to compute the integral I more robustly. This is achieved by choosing weighting functions $w_1(x)$ and $w_2(x)$ such that $w_1(x) + w_2(x) = 1$ for all $x \in D$.

Furthermore, we shall require that if $p_1(x) = 0$ or $p_2(x) = 0$, the corresponding $f(x) = 0$.

The integral I is thus split into I_1 and I_2 :

$$I = I_1 + I_2 = \int_D w_1(x)f(x) dx + \int_D w_2(x)f(x) dx. \quad (45)$$

The following are unbiased estimators for I_1 and I_2 :

$$X_f^1 = \frac{w_1(X)f(X)}{p_1(X)} \quad X_f^2 = \frac{w_2(X)f(X)}{p_2(X)}. \quad (46)$$

This can be seen as follows:

$$E_X[X_f^1] = \int_D p_1(x) \frac{w_1(x)f(x)}{p_1(x)} dx = \int_D w_1(x)f(x) dx = I_1, \quad (47)$$

and the same argument works for I_2 . Again, the reason the cancellation of $p_1(x)$ works is that either it is non-zero, or $f(x) = 0$, due to the assumption above.

Also note that we made no assumptions on the weights other than that they sum to 1. In particular, there is no requirement that the weights be derived from exact pdfs, and we are free to base them on approximate pdfs, among other choices.

Stochastic function evaluation.

Now suppose that the function evaluation is itself stochastic, i.e. it is an unbiased estimator $f(x, R)$ of the true value of $f(x)$, that uses a uniform random number R on the interval $[0, 1]$ during its evaluation. The argument can be easily extended for the case of consuming multiple uniform random numbers. We use a single random number in the proof for brevity.

Because the function estimator is unbiased, we have $E_R[f(x, R)] = \int_0^1 f(x, r) dr = f(x)$ for all x . Therefore, our full estimator becomes:

$$X_f = \frac{f(X, R)}{p(X)}. \quad (48)$$

We can see that this estimator is still unbiased, by computing its expected value over X and R :

$$\begin{aligned}
E_{X,R}[X_f] &= \int_D \int_0^1 p(x) \frac{f(x,r)}{p(x)} dr dx \\
&= \int_D p(x) \frac{\int_0^1 f(x,r) dr}{p(x)} dx \\
&= \int_D p(x) \frac{f(x)}{p(x)} dx = I
\end{aligned} \tag{49}$$

Stochastic weight and function evaluation.

When both the weight evaluation and the function evaluation in an MIS estimator are stochastic, the resulting estimator is still unbiased, provided that the random numbers used by the weight and the function are independent (which enables us to rewrite the joint integral over both random choices into separate integrals). Specifically, consider an unbiased estimator $w_1(x, R_1)$ of the true value of $w_1(x)$, and an unbiased estimator $f(x, R_2)$ of the true value of $f(x)$, based on uniform random numbers R_1 and R_2 on the interval $[0, 1]$. (again, this can be easily extended for the case of consuming multiple uniform random numbers.) The estimator for integral I_1 will become:

$$X_f^1 = \frac{w_1(X, R_1)f(X, R_2)}{p_1(X)} \tag{50}$$

We can see that this estimator is unbiased, by computing its expected value over X, R_1 and R_2 :

$$\begin{aligned}
E_{X,R_1,R_2}[X_f^1] &= \int_D \int_0^1 \int_0^1 p_1(x) \frac{w_1(x, r_1)f(x, r_2)}{p_1(x)} dr_1 dr_2 dx \\
&= \int_D p_1(x) \frac{\int_0^1 w_1(x, r_1) dr_1 \cdot \int_0^1 f(x, r_2) dr_2}{p_1(x)} dx \\
&= \int_D p_1(x) \frac{w_1(x)f(x)}{p_1(x)} dx = I_1.
\end{aligned} \tag{51}$$

The same argument can be used for X_f^2 .

Discussion.

Application to direct illumination integral. In our application, the integral of interest I is normally the direct illumination estimate at a shading point. The function $f(x)$ involves the product of the BSDF and illumination values; this is integrated over the unit sphere (or unit hemisphere for BRDFs with no transmission), which is the domain D . The random variables X_1 and X_2 are outgoing directions ω_o chosen by light sampling and BSDF sampling, respectively. For the case of light sampling, we need to stochastically evaluate the MIS weight and BSDF value for the chosen ω_o ; these evaluations will consume vectors of uniform random numbers R_1 and R_2 , respectively.

No approximation of pdfs in estimator denominators. While in the main paper we use approximate stochastic pdfs to define the weights, we never approximate the pdfs in the denominators of our estimators. In our case, the accurate values of these pdfs are already baked into the f/p estimates returned by the position-free Monte Carlo simulations.

Sum of stochastic weights. The sum of the stochastic approximations to weights w_1 and w_2 will generally not be exactly 1, but this is not required. We simply require that

- (1) the expected values of the weights sum to 1, so that the integral I separates correctly into I_1 and I_2 ,
- (2) X_f^1 and X_f^2 are unbiased estimators for I_1 and I_2 , respectively.

The combination of these properties implies an unbiased estimate for I .