DOI: 10.1111/cgf.13883

COMPUTER GRAPHICS *forum*
Volume 39 (2020), number 1 pp. 420–435

Gaussian Product Sampling for Rendering Layered Materials

Mengqi (Mandy) Xia[1], Bruce Walter[1], Christophe Hery[2,3] and Steve Marschner[1]

[1]Cornell University, Ithaca, NY, USA
{mx83, bruce.walter}@cornell.edu, srm@cs.cornell.edu
[2]Pixar Animation Studios, Emeryville, CA, USA
[3]Facebook Reality Labs, Sausalito, CA, USA
chery@fb.com

Abstract

To increase diversity and realism, surface bidirectional scattering distribution functions (BSDFs) are often modelled as consisting of multiple layers, but accurately evaluating layered BSDFs while accounting for all light transport paths is a challenging problem. Recently, Guo et al. *[GHZ18] proposed an accurate and general position-free Monte Carlo method, but this method introduces variance that leads to longer render time compared to non-stochastic layered models. We improve the previous work by presenting two new sampling strategies,* pair-product sampling *and* multiple-product sampling*. Our new methods better take advantage of the layered structure and reduce variance compared to the conventional approach of sequentially sampling one BSDF at a time. Our* pair-product sampling *strategy importance samples the product of two BSDFs from a pair of adjacent layers. We further generalize this to* multiple-product sampling*, which importance samples the product of a chain of three or more BSDFs. In order to compute these products, we developed a new approximate Gaussian representation of individual layer BSDFs. This representation incorporates spatially varying material properties as parameters so that our techniques can support an arbitrary number of textured layers. Compared to previous Monte Carlo layering approaches, our results demonstrate substantial variance reduction in rendering isotropic layered surfaces.*

Keywords: reflectance and shading models, rendering

ACM CCS: • Computing methodologies → Reflectance modelling

1. Introduction

In physically based rendering, materials are most often modelled using bidirectional scattering distribution functions (BSDFs), which describe how much light is reflected and/or transmitted and how the scattered light is distributed. Many BSDF models have been developed for a variety of surfaces, and single-surface materials, such as bare metal, can be described well with elementary BSDF models. But most materials require multiple components to model their appearance—for instance, a coloured plastic has an interface at the surface that reflects specularly and a body material below the surface that reflects diffusely. More complex materials can have more layers, such as a car finish made by coating a metal surface with a primer, then a base coat, and then a clear coat.

The common practice in graphics has been to simply add the BSDFs for multiple layers, as when modelling coloured materials with the sum of a diffuse component and a specular component. This approximation can produce reasonable results in simple cases, but it is inaccurate, and in physically based renderers, there is a practical problem: adding the layer BSDFs violates energy conservation. These problems can be solved by *ad hoc* corrections [HVL17, KC17], but the result is still not accurate—and with no reference to a correct answer, different rendering systems produce inconsistent results.

The correct way forward is to combine layered BSDFs by accurately solving the equations that describe how light transmits, reflects and inter-reflects among layers before it emanates out. Three approaches to this problem have emerged in recent research. *Layerlab* [JdJM14] converts BSDFs to a tabulated representation and computes BSDFs for layer stacks numerically. It is accurate and general, but it requires a precomputation that depends on the parameters of all the layers, so it is unworkable for textured materials. Belcour [Bel18] proposed a simpler approach for interactive rendering that

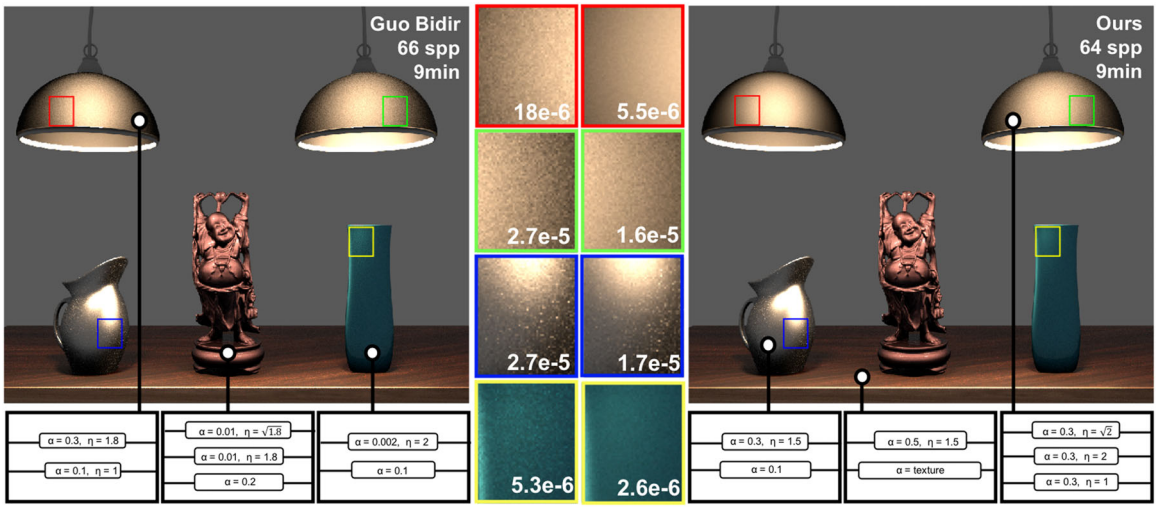

Figure 1: *We present a new Monte Carlo method for more efficiently rendering layered materials. This figure compares the bidirectional method in [GHZ18] and our pair-product sampling method. All the objects in this scene are characterized using Microfacet BSDF layers. We rendered only direct lighting to avoid the indirect noise that is unrelated to the layered BSDF evaluation. The RMSE of each colour box is reported. Our method reduces RMSE by 1.6x to 3.3x on the coloured boxes, corresponding to a 2.6x to 10.9x reduction in time.*

composes layers by estimating the statistics of the combined BSDF and representing the result as a sum of GGX lobes. This model is efficient, but approximate, and is not general enough to be a complete solution. Most recently, Guo *et al.* [GHZ18] proposed a general, accurate and precomputation-free method that uses Monte Carlo integration to estimate the values of layered BSDFs. This work serves an important stepping stone to composing layers in production rendering.

Guo *et al.*'s core innovation is to apply the same Monte Carlo sampling strategies that are used in standard scene rendering to the simpler plane-parallel problem. Although they improve significantly upon using forward path tracing to estimate layer BSDFs, these methods still have high variance, increasing render times by $2-12$ times over single-layer rendering, as reported in their paper. Thus, we focus on extending their work by developing more efficient Monte Carlo methods.

In this paper, we propose two new sampling methods for the plane-parallel layer-to-layer transport problem, which take better advantage of the layering structure to build lower-variance sampling strategies for many layer combinations. We will mainly compare with the bidirectional estimator in [GHZ18] as their unidirectional estimator is less efficient.

Contributions. In summary, we present the following theoretical and practical contributions:

- We introduce the *pair-product sampling* strategy, which can sample an inner direction according to the product of BSDFs from a pair of adjacent layers.
- We propose *multi-product sampling*, which extends pair-product and samples all inner directions at once according to the product of a chain of three or more BSDFs.
- We develop a novel and compact parameterized Gaussian representation of single-layer BSDFs. The two new sampling strategies are based on this representation. We express the Gaussian parameters as smooth functions of the BSDF parameters, so that only a one-time precomputation is required for each type of BSDF model.
- We demonstrate that these new sampling strategies lead to significant variance reduction, compared to Guo *et al.*'s conventional approach of serially sampling a single BSDF at a time, in rendering isotropic surface scattering layers.

Assumptions. Like previous work on layered materials, we make several assumptions in order to model a layered material as a BSDF.

- Layers are locally planar and parallel.
- Layers are thin compared to variations in illumination or material properties. Thus, any horizontal displacement of the exiting point relative to the incoming point is assumed to be negligible—in other words, we are modelling a BSDF rather than a BSSDF.
- Each layer's BSDF is known. That is, it is available in the standard form of a function of incoming and outgoing directions that can be evaluated and sampled.

In this paper, we will only discuss isotropic surface scattering layers, though volume scattering and anisotropic surface layers can also be handled using the estimators in [GHZ18], as their sampling strategies can be easily integrated. Extending our new sampling strategies to improve the handling of such layers is left as future work.

2. Related work

This section gives an overview of related work on layered models, approximating BSDFs using Gaussian distributions, and joint sampling of multiple scattering events in path space.

Table 1: Table of symbols.

η	index of refraction
α	surface roughness
θ	polar angle
ϕ	azimuthal angle
ω_i	incoming direction
ω_o	outgoing direction
ψ_k	direction ofd light flow within a layer stack
$\underline{\psi_k^{\perp}}$	ψ_k projected onto the unit disc
$\underline{\psi}$	slope space projection of direction ψ
$f(\omega_i, \omega_o)$	BSDF of a combined layer stack
f_k	BSDF for layer k
F_k	directional albedo for layer k
$f_{j_0 \cdots j_n}$	BSDF for a path type that visits layer $j_0 \cdots j_n$
p_k	importance sampling function associated with f_k
$w_{j_0 \cdots j_n}$	BSDF to probability ratio
$g_{j_0 \cdots j_n}$	estimator for $f_{j_0 \cdots j_n}$
$g^{s,t}$	estimator that samples s BSDFs at a time and takes t steps from ω_i
μ	mean of Gaussian distribution
Σ	covariance of Gaussian distribution
V	inverse of covariance matrix (precision matrix)
L	lower triangular matrix in Choleksy factorization
$N(x\|\mu, \Sigma)$	Gaussian distribution with variable x
$\mathcal{M}$	Gaussian mixture
s	Gaussian product normalization factor
γ_i	parameters of the ith Gaussian in the mixture
Q	quadratic polynomial
b	degree one coefficient in Q
c	constant in Q
$\mathbf{P}$	polynomial coefficients in the pameterization
$\mathbf{B}(t)$	Bernstein basis

2.1. Layered models

Several layered models have been proposed to solve particular specialized layer configurations. For example, the work of [DH96] focuses on modelling ageing of metals; the method presented by [Sta01] derives a BSDF model specific for skin; and [WWD*06] introduced a layered model for leaves.

A variety of general layered models have also been proposed in the graphics literature. Layerlab [JdJM14] introduced a numerical framework to accurately synthesize layered structure. Their work followed the directional basis idea in [Sta01] but extended the prior work by handling arbitrarily layered materials. Zeltner *et al.* [ZJ18] later extended Layerlab to handle anisotropic materials. However, both works [JdJM14, ZJ18] require an expensive per-BSDF precomputation and as a result are not suitable for spatially varying materials. Weidlich *et al.* [WW07] proposed a flexible and simple layer framework, but this framework does not correctly account for multiple scattering within the layers. Existing solutions in the industry [HVL17, KC17] have the same problem due to the approximations used to simplify light scattering events. Belcour [Bel18] recently presented a real time layering model that makes use of lower-order moments to approximate the overall BSDF as a sum of GGX lobes. Figure 2 compares Belcour's method and our method with the ground truth, and shows that unlike Belcour's method, our method is unbiased. Our Gaussian representation can similarly be viewed as tracking low-order moments but Belcour's statistics are specialized to GGX distributions, which do not provide easy sampling of products. Moreover, Belcour's method does not provide direct mapping between variance and roughness but only a transformation on roughness where multiple bounces behave linearly. So, his GGX specialized statistics cannot convert to Gaussians. Since our product sampling requires direct access to the parameters, we need to construct another parameterization. Very recently, Bati *et al.* [BPB19] conducted a numerical evaluation of two variants of Belcour's method and two variants of Weidlich and Wilkie's model on layered materials, and they analysed approximation errors in more detail.

Our approach is most similar to that of [GHZ18], which we will refer as Guo's bidirectional method. Both our approach and their approach perform Monte Carlo estimation on the light scattering integral, and both achieve a significant speed-up compared to forward path tracing the layer stack (Figure 5). Guo's bidirectional method samples each layer BSDF, while our method allows the influence of multiple BSDFs to be considered simultaneously during sampling. This is accomplished by constructing local Gaussian approximations for the BSDFs, and reduces the sampling noise thereby improving the Monte Carlo efficiency.

2.2. Representing BSDFs with Gaussians

Since Gaussians and Gaussian mixtures are efficient to generate samples from, flexible in fitting and representing distributions via the expectation−maximization algorithm, and convenient in calculating products, they have been proven to be useful in rendering. For example, they have been used for rendering hair [MJC*03], specular microstructure [YHMR16, YHW*18] and caustic design [PJJ*11].

Different domains have been adopted for fitting Gaussian distributions. LEAN mapping [OB10, HKL14] and LEADR mapping [DHI*13] both represent BSDFs as bivariate Gaussians in the surface slope domain and develop real-time filtering techniques for bump maps, normal maps and displacement maps. Vorba *et al.* [VKv*14] proposed an online learning procedure that trains progressively from particles to handle scenes with complex lighting. Their method involves fitting Gaussian distributions in a projected unit square domain with area preserving mapping of [SC97]. An alternative is to use a disc mapping [HEV*16, HES*18]. Our method is similar to [HES*18] in the sense that we both define the Gaussian parameters as smooth functions of BSDF parameters. However, our method is different in that each 2D BSDF slice is represented using a bivariate Gaussian in outgoing direction slope domain. Compared to the other spaces used above, this domain is unbounded so that it avoids potential error in the Gaussian fitting introduced by the boundary, while it remains convenient to take the product of two fitted Gaussians. Also, we express Gaussian parameters as functions of multi-dimensional polynomials and rational functions instead of B-splines. Like [JRJ11, VKv*14, HEV*16, HES*18], we adopt fitted Gaussian distributions for importance sampling.

In addition, spherical Gaussians have also been adopted to solve various rendering problems. For example, Tsai and Shih [TS06] presented a precomputed radiance transfer method using spherical radial basis functions (SRBFs) and for real-time rendering under high-frequency lighting environments. Iwasaki *et al.* [IFDN12]

Figure 2: *A double layer sphere comparison shows that Belcour's is an approximation to the ground truth, while our method converges exactly to the ground truth.*

improved the previous method by introducing the integral spherical Gaussian and reducing the need for precomputed data. Anisotropic spherical Gaussians [XSD*13] have also been used to more faithfully represent anisotropic lightings and BRDFs. Xu *et al.* [XMR*11] derived a compact 1D circular Gaussian representation for hair scattering and enables closed-form integration with SRBF lights at run time. Spherical Gaussians would be an alternative to our slope-space Gaussians, but our slope-space approach is a natural choice for layered surfaces because it is limited naturally to the hemisphere and stretches out the angles near grazing in a way that makes the functions easier to fit.

2.3. Importance sampling products of functions

Our pair-product and multiple-product strategies are able to generate samples according to the product of two or more BSDFs. This idea of joint sampling scattering events also appears in a somewhat different setting in [HEV*16, GKH*13]. Gaussian mixtures are used to represent the illumination and the reflectance factors and perform product importance sampling in [HEV*16]. However, their work requires per-material preprocessing. Georgiev *et al.* [GKH*13] proposed a joint path importance sampling method to efficiently render participating media. Their method constructs paths that account for the product of anisotropic phase functions and geometric terms across sequences of path vertices. They derive analytic expressions for isotropic scattering and tabulate conditional probability density functions (PDFs) in the joint distribution for importance sampling anisotropic scattering. For the layering problem, the analytic joint distribution is not available, and our multi-product method instead builds a approximation to the joint distribution based on the parameterized Gaussian representation.

Some other works have dealt with joint sampling products of functions using wavelets. Clarberg *et al.* [CJAMJ05] generalize wavelet products to higher dimensional spaces and apply them to rendering measured BRDFs in complex distant lighting environments. Sun and Mukherjee [SM06] extend wavelet products from double/triple product integrals to general multi-function product integrals. Both works use the Haar wavelet basis, and the multi-product formulation relies on a heavy-weight wavelet tree structure. Moreover, the multi-product generalization [SM06] cannot handle object rotation, which is not as general as our parameterized Gaussian representation.

3. Background and overview

In this section, we review the path integral and Monte Carlo estimator for the layering problem, explain our motivation for improving

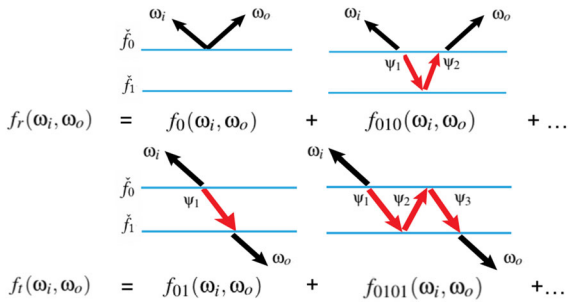

Figure 3: *The top row and the bottom row illustrate the front-side reflection $f_r(\omega_i, \omega_o)$ and transmission $f_t(\omega_i, \omega_o)$, respectively. Each of them is a sum of different components and the first two components are drawn. Except for $f_0(\omega_i, \omega_o)$, all the other components are integrals over the red interior directions ψ_j.*

the previous Monte Carlo layering methods and give an outline of the paper. Please refer to Table 1 for the notation used in this section.

3.1. Path integral and Monte Carlo estimator

Define single layer BSDF as f_k, where layer index k starts from 0. Our job is to compute the total BSDF of the layer stack, denoted as f. This BSDF is a sum over all possible paths by which light arriving from direction ω_i can travel through the layers and eventually exit in direction ω_o. We use $f_{j_0\cdots j_n}$ to denote the BSDF for a particular path type of length n that visits layer $j_0 \cdots j_n$ sequentially. For example, for a single-layer material, $f = f_0$; for a two-layer material, the front-side reflection f_r and transmission f_t components of the material's BSDF can be written as:

$$f_r(\omega_i, \omega_o) = f_0(\omega_i, \omega_o) + f_{010}(\omega_i, \omega_o) + f_{01010}(\omega_i, \omega_o) + \cdots \quad (1)$$

$$f_t(\omega_i, \omega_o) = f_{01}(\omega_i, \omega_o) + f_{0101}(\omega_i, \omega_o) + \cdots \quad (2)$$

The back-side reflection and transmission from the back side to the front side are similar. The key problem in rendering layered surfaces is to compute multi-index f's. For contributions other than f_0, there are many possible paths joining ω_i to ω_o, which can be described by the path's directions inside the layer stack, denoted as ψ_k. These interior directions start with index 1 and point in the direction of light flow. For a path of length n, we further define the convention that $\psi_0 = -\omega_i$ and $\psi_{n+1} = \omega_o$. Figure 3 illustrates these concepts in the context of a two-layer material. We use the same position-free formulation as Guo [GHZ18], where the horizontal positions do not matter; only the vertical depth and directions are relevant to the light transport integral. With the position-free assumption, contribution to the BSDF due to a path type with more than one interaction can be computed by integrating over interior directions:

$$f_{j_0\cdots j_n}(\psi_0, \psi_{n+1}) = \int \cdots \int f_{j_0}(-\psi_0, \psi_1) \cdots f_{j_n}(-\psi_n, \psi_{n+1})\, d\psi_1^{\perp} \cdots d\psi_n^{\perp}. \quad (3)$$

In the above integral, $f_0, \ldots, f_n$ denote individual layer BSDFs in a layer stack. $d\psi_k^\perp$ is the projected solid angle measure with respect to ψ_k (so $d\psi_k^\perp = |\cos\psi_k|\, d\psi_k$) and the integrals are over the corresponding hemispheres. In this paper, we will consistently use projected solid angle measures for the integrals and probabilities to avoid cluttering the equations with cosine terms. We will also assume basic radiance conventions [Vea97] to avoid writing the index of refraction terms with each transmission event.

Following [GHZ18], we use Monte Carlo integration to estimate these integrals. In the general case, to estimate a length n component $f_{j_0\cdots j_n}(\omega_i, \omega_o)$, we generate n internal directions $\psi_1 \cdots \psi_n$ and the corresponding estimator is:

$$g_{j_0\cdots j_n}(\psi_1, \ldots, \psi_n|\omega_i, \omega_o) = \frac{f_{j_0}(\omega_i, \psi_1)\cdots f_{j_n}(-\psi_n, \omega_o)}{p(\psi_1, \ldots, \psi_n|\omega_i, \omega_o)}, \quad (4)$$

where $p(\psi_1, \ldots, \psi_n|\omega_i, \omega_o)$ is the (joint) probability of generating the internal directions with the chosen strategy. Choosing the distribution p is the key element of defining an algorithm to evaluate the BSDF of the multilayer surface. Ideally, one wants this probability to match the numerator as closely as possible so the estimator will have low variance. Section 4 will discuss strategies to sample these interior directions.

3.2. Motivation

Existing Monte Carlo layering approaches all use given importance sampling functions of individual layers sequentially to generate the inner directions. In other words, in the previous work, $p(\psi_1, \ldots, \psi_n|\omega_i, \omega_o)$ in Equation (4) is always a product of single-layer importance sampling functions. The problem with this approach is that the joint probability distribution can account for at most n of the $n + 1$ BSDFs, leaving one extra BSDF in the numerator, and this can introduce high variance. Ideally, we would like to have a joint distribution that is exactly proportional to the whole BSDF chain, but we do not have a closed form solution to this high-dimensional function. Therefore, we turn to combining lower dimensional local approximations to construct high-dimensional PDFs that better importance sample the whole BSDF chain. To this end, we propose pair-product sampling and multiple-product sampling. Pair-product sampling generates one of the inner directions using a distribution that accounts for a pair of BSDFs of the adjacent layers. Multiple-product sampling, on the other hand, generates several inner directions for a particular path type at once using a distribution that accounts for a chain of three or more BSDFs. Both methods introduce new sampling distributions that account for all the BSDF terms in Equation (4), so they can reduce variance compared to the previous Monte Carlo approaches. Although Guo *et al.*'s bidirectional method performs multiple importance sampling (MIS) on multiple estimators $g_{j_0\cdots j_n}$, there are cases where none of the estimators have low variance.

3.3. Paper overview

Section 4 discusses four sampling strategies: forward sampling, Guo's bidirectional [GHZ18], our pair-product sampling and our multiple-product sampling, in detail. Sections 5 and 6 describe the details of the specific methods we use to sample products of two or more BSDFs. They are based on approximating the individual BSDFs by parameterized bivariate Gaussian mixtures in a one-time preprocess. These separate approximations are then combined into bi- or multivariate distributions that can be sampled exactly.

Section 7 provides implementation details and Section 8 contains the rendering results including texturing examples and equal-time comparisons of Guo *et al.*'s method to ours in different scenarios. Section 9 discusses limitations and potential extensions of the current work and concludes the paper.

4. Layer BSDF evaluation

Our work addresses the local problem of integrating over light paths within a layered material, producing a BSDF that is then used in the global problem of tracking light flow between surfaces in the scene. Although both the local and the global problem are expressed as path integrals and bidirectional methods can be used for either, we treat the two problems separately. While we will focus mainly on techniques to evaluate the BSDF of the combined layer stack, later we will also discuss how to sample its BSDF and provide heuristic weights for combining BSDF and light sampling in the global integrator. First, we will discuss the two strategies based on path tracing and bidirectional path tracing. Then, we will describe the two new strategies proposed in this paper: pair-product sampling and multi-product sampling.

These sampling strategies to evaluate layered BSDFs all assume that each layer's BSDF f_j comes with an associated importance sampling distribution p_j. For example, given ψ, importance sampling can generate ψ' with a probability $p_j(\psi'|\psi)$ that closely matches the BSDF $f_j(\psi, \psi')$. Define the ratio of BSDF to probability be

$$w_j(\psi, \psi') = \frac{f_j(\psi, \psi')}{p_j(\psi'|\psi)}. \quad (5)$$

A good importance sampling function should have a nearly constant ratio

$$w_j(\psi, \psi') \approx \int f_j(\psi, \omega)\, d\omega^\perp. \quad (6)$$

Let $F_j(\psi) = \int f_j(\psi, \omega)\, d\omega^\perp$. In this paper, we assume all BSDFs are reciprocal, so that the same probability distribution serves for sampling the outgoing direction given the incoming direction or vice versa, but in implementation it is often useful to keep the two cases separate.

4.1. Forward sampling

Forward sampling, Figure 4(a), is a simple but not especially effective strategy for sampling the interior directions in a layered material. The ψs are chosen sequentially in order, each sampled from one layer's BSDF conditioned on the previous. For instance, if we apply this method to calculate $f_{010}(\omega_i, \omega_o)$, p will be the product of p_0 and p_1 and specializing 4 we obtain:

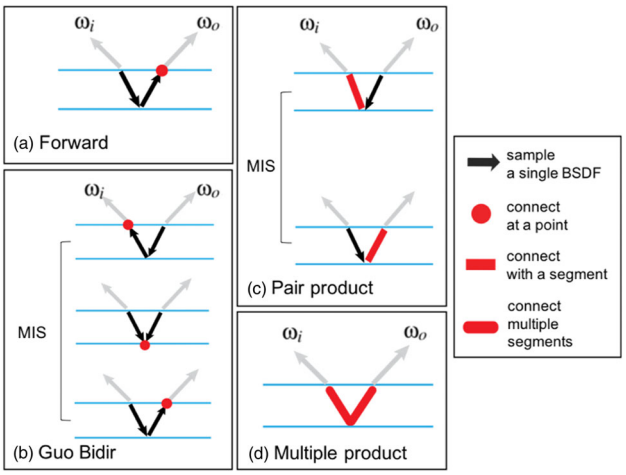

Figure 4: *Sampling strategy illustration for the f_{010} component. (a) The forward sampling strategy generates the inner directions by forward sampling using the given importance sampling functions of single layer BSDFs, connecting at a point that lies on the same layer with the outgoing direction. (b) Guo et al.'s bidirectional method performs MIS on three different connecting strategies, each of them connecting at a point where two subpaths meet at the same layer. (c) Pair-product sampling performs MIS on two connecting strategies; each of them generates one inner direction using the given importance sampling function and then connects a segment with either the incoming or the outgoing direction. This connecting segment is chosen by sampling from a Gaussian distribution that approximates the product of the BTDF of the top layer and the BRDF of the bottom layer. (d) Multiple-product sampling connects directly through multiple segments; this connection is made by sampling from a multivariate Gaussian distribution that approximates the product of the full BSDF chain in the integrand of f_{010}.*

$$g_{010}(\psi_1, \psi_2|\omega_i, \omega_o) = \frac{f_0(\omega_i, \psi_1) f_1(-\psi_1, \psi_2) f_0(-\psi_2, \omega_o)}{p_0(\psi_1|\omega_i)\, p_1(\psi_2| - \psi_1)} \tag{7}$$

$$= w_0(\omega_i, \psi_1) w_1(-\psi_1, \psi_2) f_0(-\psi_2, \omega_o) \tag{8}$$

$$\approx F_0(\omega_i) F_1(-\psi_1) f_0(-\psi_2, \omega_o). \tag{9}$$

This brute force method can have slow convergence because each sampled direction takes into account only one BSDF and incident direction. When the top layer roughness is small, $f_0(-\psi_2, \omega_o)$ can introduce large variance.

4.2. Guo's bidirectional method

A more robust alternative to forward sampling is Guo *et al.*'s bidirectional method. Their method makes use of a collection of sequential sampling strategies similar to forward sampling, but which choose the inner directions in different sequences. For instance, for the f_{010} component, with two interior directions ψ_1 and ψ_2 to be sampled, there are three strategies (Figure 4b):

0. Choose $\psi_2 \sim p_0(-\psi_2|\omega_o)$, then $\psi_1 \sim p_1(-\psi_1|\psi_2)$;
1. Choose $\psi_1 \sim p_0(\psi_1|\omega_i)$ and $\psi_2 \sim p_0(-\psi_2|\omega_o)$; or
2. Choose $\psi_1 \sim p_0(\psi_1|\omega_i)$, then $\psi_2 \sim p_1(\psi_2| - \psi_1)$.

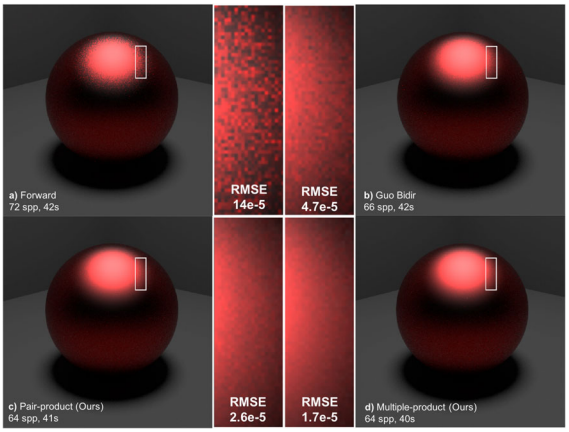

Figure 5: *Equal-time comparison of four strategies: (a) forward sampling, (b) Guo's bidirectional, (c) pair-product sampling and (d) multiple-product sampling. The scene contains a double layer sphere with top layer roughness 0.3 and bottom layer roughness 0.1. We render up to the f_{010} component for this test. The pair-product sampling obtains 5.4x and 1.8x reduction in RMSE compared to the forward sampling and the bidirectional, respectively. These correspond to 29.2x and 3.2x reduction in render time. The multiple-product sampling obtains 8.2x and 2.8x reduction in RMSE compared to the forward sampling and the bidirectional method, corresponding to 67.2x and 7.8x reduction in render time.*

Option (2) is the same as forward sampling. In general, when there are n interior directions to be generated, the mth strategy chooses $\psi_1, \ldots, \psi_m$ sequentially starting from the incident direction and $\psi_n, \ldots, \psi_{m+1}$ sequentially starting from the outgoing direction.

Any of these sampling strategies can be used as an importance sampling distribution for the reflection integral, leading to estimators that are similar to the estimator for forward path tracing, but with different layer BSDFs left in the estimator. For instance, the three sampling strategies for the $f_{010}(\omega_i, \omega_o)$ component lead to the estimators

$$g^{1,0}_{010}(\psi_1, \psi_2|\omega_i, \omega_o) = f_0(\omega_i, \psi_1)\, w_1(-\psi_1, \psi_2)\, w_0(-\psi_2, \omega_o), \tag{10}$$

$$g^{1,1}_{010}(\psi_1, \psi_2|\omega_i, \omega_o) = w_0(\omega_i, \psi_1)\, f_1(-\psi_1, \psi_2)\, w_0(-\psi_2, \omega_o), \tag{11}$$

$$g^{1,2}_{010}(\psi_1, \psi_2|\omega_i, \omega_o) = w_0(\omega_i, \psi_1)\, w_1(-\psi_1, \psi_2)\, f_0(-\psi_2, \omega_o), \tag{12}$$

where $g^{s,t}$ means the estimator samples s BSDFs at a time and takes t steps from ω_i and here $s = 1$. Each individual estimator has the potential for high variance due to the unsampled layer BSDF. [GHZ18] relies on the robustness of the bidirectional method [VG95] to multiple importance sample these estimators and this works well as long as there is at least one good connecting strategy for each path type. But it is entirely possible that all of the individual estimators are poor and Guo's bidirectional method will still exhibit high variance.

4.3. Pair-product sampling

The first of our new sampling strategies, *pair-product sampling*, solves this problem by using importance sampling distributions based on the product of two adjacent layers' BSDFs. For example, for layers j and k, we would like to have a pair sampling distribution $p_{jk}(\psi|\omega,\omega')$ with the ratio w_{jk}

$$w_{jk}(\omega,\psi,\omega') = \frac{f_j(\omega,\psi)\, f_k(-\psi,\omega')}{p_{jk}(\psi|\omega,\omega')} \tag{13}$$

closely matching the normalization factor

$$w_{jk}(\omega,\psi,\omega') \approx \int f_j(\omega,\psi)\, f_k(-\psi,\omega')\, d\psi^{\perp} = f_{jk}(\omega,\omega').$$

To construct a path using pair-product sampling, we use sequential sampling for m steps from ω_i and $n-m-1$ steps from ω_o, then choose ψ_{m+1} from a product sampling distribution. In this way, there are no individual BSDFs left in the estimator. For example, for the f_{010} case ($n=2$), there are two pair-product strategies, illustrated in Figure 4(c):

0. Choose $\psi_2 \sim p_0(-\psi_2|\omega_o)$, then choose $\psi_1 \sim p_{01}(\psi_1|\omega_i,\psi_2)$;
1. Choose $\psi_1 \sim p_0(\psi_1|\omega_i)$, then choose $\psi_2 \sim p_{10}(\psi_2|-\psi_1,\omega_o)$.

The resulting estimators for the two strategies are:

$$\begin{aligned} g^{2,0}_{010}(\psi_1,\psi_2|\omega_i,\omega_o) &= w_{01}(\omega_i,\psi_2)w_0(\omega_o) \approx f_{01}(\omega_i,\psi_2)w_0(\omega_o),\\ g^{2,1}_{010}(\psi_1,\psi_2|\omega_i,\omega_o) &= w_0(\omega_i)w_{10}(-\psi_1,\omega_o)\\ &\approx w_0(\omega_i)f_{10}(-\psi_1,\omega_o). \end{aligned} \tag{14}$$

Approximately, only the normalization factors from the sampling probabilities remain in the estimators. Thus, the pair-product estimator effectively combines a pair of BSDFs, replacing the variance of sampling the individual BSDFs with the variance of sampling their combination, which is smoother. In many cases, this new strategy significantly decreases noise where the material does not include a diffusive layer to provide a low-variance place for the sequential strategies to connect the incoming and outgoing subpaths. To ensure robustness, we MIS all the pair-product sampling estimators, e.g. MIS $g^{2,0}_{010}$ and $g^{2,1}_{010}$ when computing $f_{010}(\omega_i,\omega_o)$. This pair-product sampling strategy relies on an importance sampling distribution for individual layers that makes it efficient to compute products. Our parameterized Gaussian representation, described in Section 5, provides this.

4.4. Multiple-product sampling

Although pair-product sampling improves on the performance of sequential sampling, it does not solve the problem completely. It removes the unsampled BSDF that is left in the estimator by the sequential strategies, but it replaces that BSDF with the normalization factor f_{jk}. Even if the pair-product sampling procedure has low variance with respect to the direction it is choosing, the normalization factor f_{jk} is still a function of other, previously chosen, random directions in the path. This dependence can produce high variance. For example, $f_{01}(\omega_i,\psi_2)$ in $g^{2,0}_{010}(\psi_1,\psi_2)$ of Equation (14) can introduce large variance if its value changes rapidly w.r.t. ψ_2. The normalization factor can be recognized as the composition of two adjacent BSDFs along the path, which means that in essence the pair-product strategy is simply the sequential strategy applied to a path one event shorter.

Our second new sampling strategy takes this idea farther, constructing sampling distributions for the product of s sequential BSDFs rather than just two; an example for $s=3$ is:

$$p_{jk\ell}(\psi_a,\psi_b|\omega,\omega') = \frac{f_j(\omega,\psi_a)f_k(-\psi_a,\psi_b)f_\ell(-\psi_b,\omega'),}{w_{jk\ell}(\omega,\psi_a,\psi_b,\omega')} \tag{15}$$

where

$$\begin{aligned} w_{jk\ell}(\omega,\psi_a,\psi_b,\omega') &\approx f_{jk\ell}(\omega,\omega')\\ &= \int\int f_j(\omega,\psi_a)f_k(-\psi_a,\psi_b)f_\ell(-\psi_b,\omega')d\psi_a^{\perp}d\psi_b^{\perp}. \end{aligned}$$

With the ability to sample a pair of directions from a product of three BSDFs, we can construct paths by sampling m directions starting from ω_i and $n-m-2$ directions starting from ω_o, then jointly choosing the remaining two from the $p_{jk\ell}$ distribution. The resulting estimator has one fewer term; for instance, the estimator for $f_{010}(\omega_i,\omega_o)$ is simply:

$$g^{3,0}_{010}(\psi_1,\psi_2|\omega_i,\omega_o) = w_{010}(\omega_i,\psi_1,\psi_2,\omega_o) \approx f_{010}(\omega_i,\omega_o).$$

In this case, the entire path is being sampled in a single operation, and an ideal importance sampling distribution would lead to a zero-variance estimator. Although in general it is impossible to compute $f_{010}(\omega_i,\omega_o)$, replacing the chain in the estimator by a single sampling weight $w_{010}(\omega_i,\psi_1,\psi_2,\omega_o) \approx f_{010}(\omega_i,\omega_o)$ that has lower variance than the original terms can further reduce variance.

The same approach can be taken to arbitrarily long subpaths, but in practice as the paths get longer the approximations get poorer, eventually leading to estimators that do not outperform the simpler ones. In the present paper, we only demonstrate sampling products of up to three BSDFs by sampling the first length-three component using multiple-product sampling. This method almost always outperforms the other strategies on three-vertex components, such as the $f_{010}(\omega_i,\omega_o)$ component. We discuss future improvements of the longer paths in Section 8.

5. Layer sampling algorithms

In this section, we describe the sampling scheme for the pair-product and multi-product sampling strategies described in the previous section. They rely on an approximate representation of BSDFs in terms of Gaussians. The details of the fitting will be discussed in Section 6.

5.1. Gaussian representation

The core idea of this paper is that the simplicity of the layer reflection integral in the plane-parallel setting allows us to take better advantage of the structure of the integrand than is possible

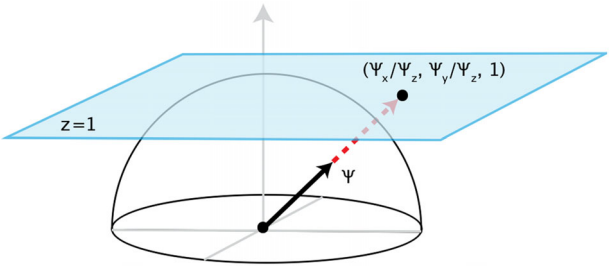

Figure 6: *Given an incoming direction, the 2D BSDF slice is approximated using a bivariate Gaussian in the outgoing direction's slope domain, which essentially projects the outgoing direction onto the plane $z = 1$. In this domain, the Microfacet model with the Beckmann distribution can be well fitted using a single Gaussian.*

in general scenes. The mechanism by which we express this structure is approximating BSDFs with Gaussian distributions and sums of Gaussians.

Motivation for the representation. Given two layer BSDFs f_j and f_k, pair-product sampling requires working with an expression of the form:

$$f_j(\omega, \psi)\, f_k(-\psi, \omega'),$$

with ω and ω' fixed. This can be seen as a product of two functions of ψ that are slices of f_j and f_k with one argument fixed. To efficiently sample this product, we would like to represent the 2D BSDF functions $f_j(\omega, \cdot)$ and $f_k(\cdot, \omega')$ using a type of distribution that is easy to sample from, is preserved under multiplication and fits various BSDFs well. With these considerations, we chose bivariate Gaussian mixtures to approximate these BSDFs. $f_j(\omega, \cdot)$ is written as:

$$f_j(\omega, \psi) \approx \sum_{i=1}^{n} N_{j,i}(\overline{\psi}; \gamma_{j,i}(\eta, \alpha, \omega)), \tag{16}$$

where $\gamma_{j,i}(\eta, \alpha, \omega)$ is represented as an explicit function of index of refraction η, roughness α and incident direction ω. $\gamma_{j,i}(\eta, \alpha, \omega)$ is a vector of parameters for the ith Gaussian, which includes its mean, covariance and mixture weight. $\overline{\psi}$ is the slope space projection of ψ—that is, a 2D vector with coordinates $(\psi_x/\psi_z, \psi_y/\psi_z)$ (Figure 6). We selected slope space for the domain of this representation because it makes common BSDFs easy to fit: a Microfacet model with Beckmann slope distribution can be well approximated using a single Gaussian, and with the GGX slope distribution can be well approximated using two Gaussians.

Gaussian multiplication and sampling. The product of two Gaussian mixtures is a new Gaussian mixture (see Appendix A). So, the product can be evaluated and sampled efficiently, which is what is needed for our product sampling. Ease of sampling is a key motivation for choosing to use Gaussians. A Gaussian with known mean and covariance can be sampled using the Box–Muller transform [BM58].

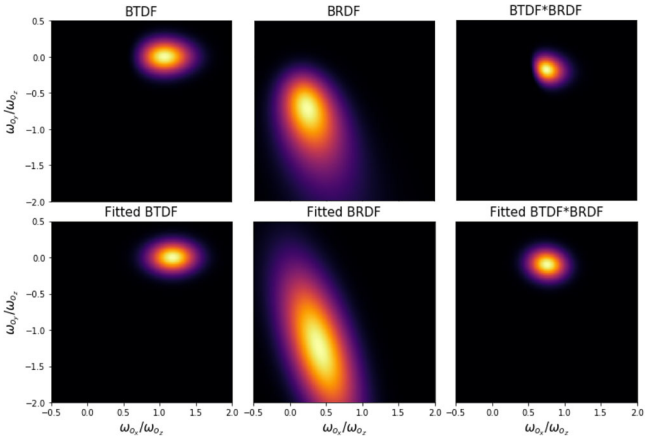

Figure 7: *The top row shows a 2D BTDF slice, a 2D BRDF slice and their product, all using the slope domain for the outgoing direction. The bottom row shows the two corresponding slices of the parameterized Gaussian fitting and their product. Note that the product of the two BSDFs can be approximated much better by the product of the two Gaussians than it is by either of the individual BSDFs. The single Gaussian in this case does not fit the BRDF slice very well compared to the BTDF slice fitting because this BRDF slice has stronger asymmetry, while Gaussian is symmetric. In practice, the product distribution is close enough to be a very useful sampling distribution.*

5.2. Pair-product sampling

The pair-product sampling operation requires importance sampling a product of two layer BSDFs with an approximation of the distribution in Equation (13). Given Gaussian fits for the two BRDFs, we can compute samples by forming the product of the Gaussian approximations, then sampling it. This results in a three-step process to sample $p_{jk}(\omega, \omega')$:

- Using the fitted model, compute the parameters $\gamma_{j,i}$ for $i = 1, \ldots, n_j$ and $\gamma_{k,i}$ for $i = 1, \ldots, n_k$ of the n_j and n_k Gaussians that approximate $f_j(\omega, \cdot)$ and $f_k(\cdot, \omega')$.
- Using Equations (A.1) through (A.4), compute the parameters $\gamma_{jk,i}$ for $i = 1, \ldots, n_j n_k$ for the $n_j n_k$ Gaussians in the product of the two mixture models.
- Generate a sample $\overline{\psi}$ from the product mixture, and convert it from slope space to a direction vector ψ.
- The probability of the sample ψ in the unit hemisphere space can be calculated using its probability in slope space multiplied with the Jacobian $1/|\psi_z|^3$.

Because this method explicitly computes the probability distribution it is sampling, the PDF of the generated sample is easily computed, and it is also easy to answer queries about the PDF of generating a particular direction. The cost of this operation rises with the product of the two model sizes, but we typically use very small mixtures with $n = 1$ or 2. The overhead cost is low compared to the stochastic layer BSDF evaluation, as shown in the results.

5.3. Multiple-product sampling

Instead of sampling one direction from a Gaussian that approximates a BSDF pair, we can leverage the same representation to generate multiple directions at once, thereby including more of the integrand in the sampling distribution. We will first describe the most useful version of this method, which samples two directions from a product of three BSDFs (Equation (15)) in the case of single-component Gaussians.

Our approach to this problem mirrors the pair-product sampling method in that we construct a Gaussian model and then sample from it, but in this case we construct a 4D Gaussian model and sample two directions jointly in a single operation. We build the high-dimensional Gaussian based on the same parameterized 2D Gaussian distribution that we use for pair-product sampling. There are two steps. The first step is to build a local 4D approximation to the middle BSDF function $f_k(\psi_a, \psi_b)$ around a chosen expansion point. Then, the second step is to chain this local approximation together with the two 2D Gaussians that are the approximations to $f_j(\omega, \cdot)$ and $f_\ell(\cdot, \omega')$, for the fixed directions ω and ω', to get the final Gaussian distribution.

4D BSDF approximation. Our parameterized representation can be viewed as a function that maps from a pair of directions to approximated BSDF values ($\Omega \times \Omega' \mapsto \mathbb{R}$). Given index of refraction η and roughness α, $\gamma_{k,i}(\eta, \alpha, \omega)$ maps from an incident direction ψ_a on a layer to the parameters of a 2D Gaussian approximation of $f_k(\psi_a, \cdot)$. Evaluating the resulting Gaussian for a particular exitant direction ψ_b gives an approximation of the layer's BSDF $f_k(\psi_a, \psi_b)$. This 4D function is not a Gaussian, though, because the mapping from incident direction to 2D Gaussian is polynomial based.

We can construct a 4D Gaussian that approximates the 4D BSDF using the parameterized fit. We compute the first and second derivatives of the parameterized fit, evaluated at the chosen expansion point and then construct a 4D function of the appropriate form that matches these derivatives (see Appendix B).

Building the sampling distribution. The desired distribution is an approximation to the product $f_j f_k f_\ell$, which we construct in three steps:

1. Construct the Gaussians $N_j \approx f_j(\omega, \cdot)$ and $N_\ell \approx f_\ell(\cdot, \omega')$.
2. Let ψ_a^* and ψ_b^* be the means (maxima) of N_j and N_ℓ, respectively, and compute the 4D Gaussian N_k that approximates f_k around (ψ_a^*, ψ_b^*).
3. Compute the parameters of the 4D Gaussian
$$N_4(\psi_a, \psi_b) = \frac{1}{s} N_j(\psi_a) N_k(\psi_a, \psi_b) N_\ell(\psi_b),$$
where s is the normalization factor.
4. Modify the negative eigenvalues of the precision matrix of N_4 to a small positive number ϵ if it is not positive definite.

In this section, we have described how to conduct pair-product sampling and multiple-product sampling based on a parameterized Gaussian representation, in order to reduce variance in stochastic layer BSDF evaluation. These methods depend upon parameterized Gaussian models, the construction of which we describe next.

6. Fitting the Gaussian model to BSDFs

This section will describe the Gaussian representation and fitting in detail. The methods of Section 5 are based on an approximation of a BSDF in terms of a Gaussian mixture that depends on several variables. In this section, we provide the details of this approximation and how it is computed. Note that the precomputed representation encompasses all possible parameter values for a given BSDF model, so fitting is a one-time, scene-independent precomputation that only needs to be done once for a given BSDF model.

Algorithm 1. Parameterized 2D fitting

Input: A BSDF model, number of Gaussians in each slice (default 1), incident direction polynomial order (default 2), roughness polynomial order (default 2), index of refraction polynomial order (default 1).
Output: The coefficients **P** in (18) and (19).
for *each* η *in IOR list* **do**
  for *each* α *in roughness list* **do**
    for *each* θ *in incident polar angle list* **do**
      Evaluate BSDF value on a uniform grid with adpated boundary.
      Numerical integration to calculate covariance $\Sigma_{(\eta,\alpha,\theta)}$ -> Used in a)
      Draw samples according to discrete BSDF values -> Used in b)
    end
  end
end
Non-linear Optimization
a) For roughness >= 0.1:
Use SLSQP to solve for $\mathbf{P} = argmin\mathcal{L}_1$, $\mathcal{L}_1$ defined in (20).
b) For roughness < 0.1:
Use SLSQP to solve for $\mathbf{P} = argmin\mathcal{L}_2$, $\mathcal{L}_2$ defined in (21).

Representation. Consider an isotropic BSDF for a surface with index of refraction η and roughness α. Let θ be the polar angle of the incident direction. To incorporate the material properties along with the incident direction into the fitted representation, we want to find a mapping

$$\gamma : \{\eta\} \times \{\alpha\} \times \{\theta\} \mapsto \mathcal{M},$$

where $\mathcal{M}$ is a bivariate Gaussian mixture. In other words, the bivariate Gaussian parameters: mean μ, covariance Σ and mixture weight q are represented as explicit functions of the index of refraction, η, roughness α and incident polar angle θ. For brevity, we omit the subscripts of γ and $\mathcal{M}$.

There are many ways to construct this parameterization. However, we would like a compact parameterization that ensures symmetric positive definiteness (SPD) of the Gaussian covariance. Based on these considerations, we write the precison matrix of the Gaussian using its Cholesky decomposition:

$$V = LL^T. \tag{17}$$

Instead of directly parameterizing the entries of Σ, we parameterize the entries of L. By constraining the diagonal entries of L to be

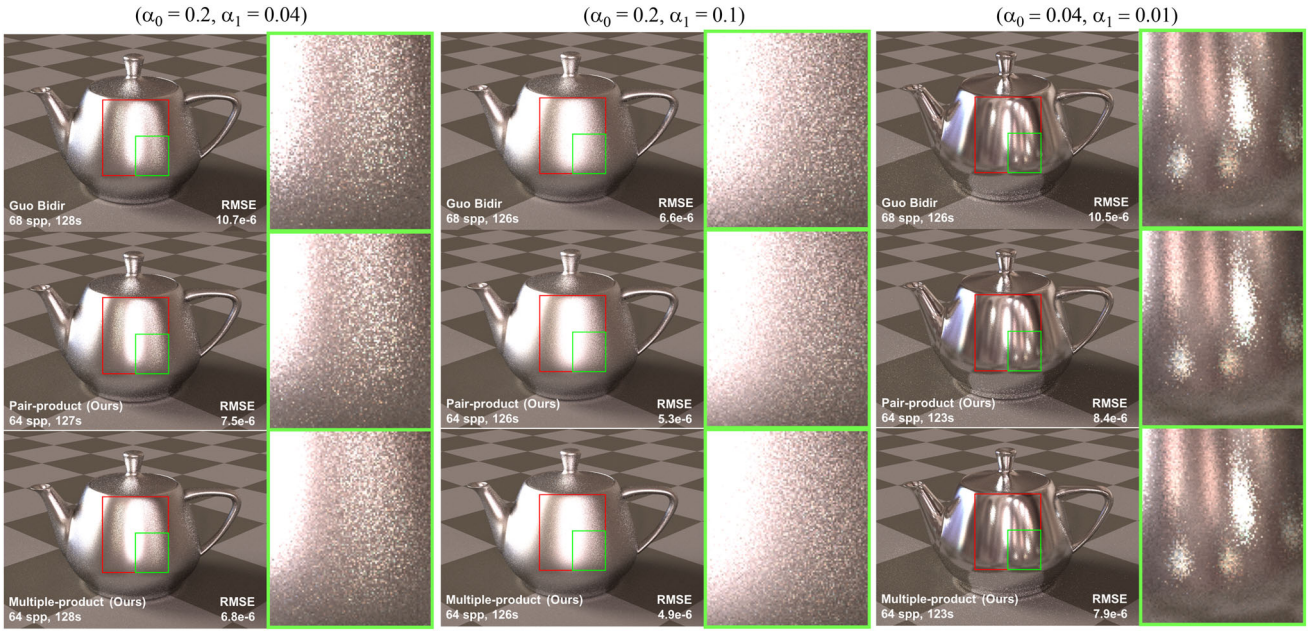

Figure 8: *A double layer teapot scene rendered using Guo et al.'s bidirectional (top row), our pair-product sampling (middle row) and our multiple-product sampling (bottom row). We rendered global illumination and the full path length through the layer stack. The green box shows the zoom area. The red box is used to calculate RMSE. Compared to Guo's bidirectional, the pair- product sampling obtains on average 1.3x reduction in RMSE, which corresponds to 1.7x reduction in render time; the multiple-product sampling obtains on average 1.4x reduction in RMSE, which corresponds to 2.0x reduction in render time.*

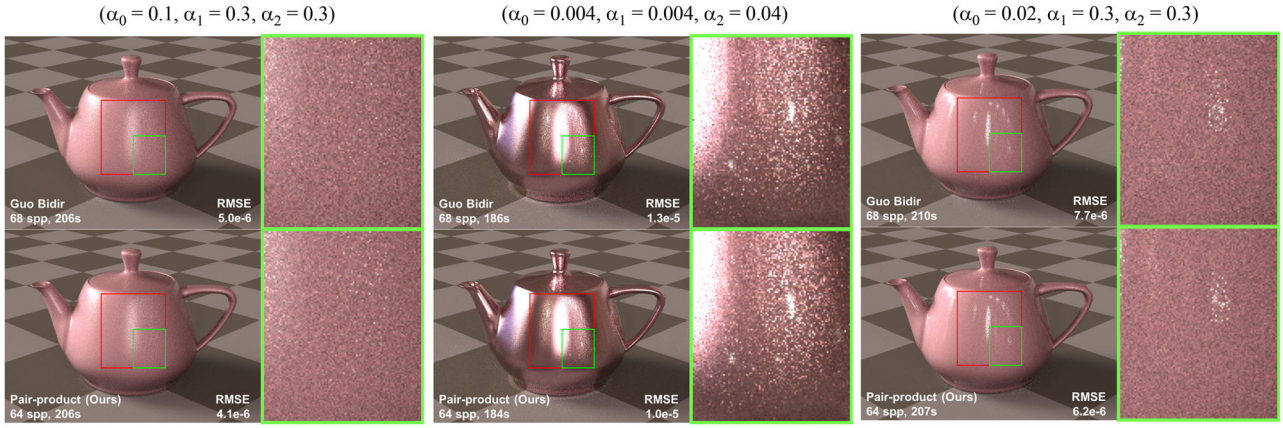

Figure 9: *A triple layer teapot scene rendered using Guo et al.'s bidirectional (top row) and our pair-product sampling (bottom row). We rendered global illumination and the full path length through the layer stack. The green box shows the zoom area. The red box is used to calculate RMSE. The pair-product sampling obtains on average 1.2x reduction in RMSE, which corresponds to 1.4x reduction in render time.*

positive, we ensure that the precision matrix V, and therefore the covariance that is its inverse, is SPD because for any $\mathbf{x} \in \mathbb{R}^2$,

$$\mathbf{x}^T V \mathbf{x} = \mathbf{x}^T L L^T \mathbf{x} > 0.$$

To constrain the diagonal entries of L to be positive, we use a Bézier curve to express L and constrain the control points of the curve to be positive. Each component of μ and L is represented using a multidimensional polynomial. Let I, J, K be the orders in η, α and θ, respectively, and $\mathbf{B}_k^K(\theta)$ be the kth degree-K Bernstein polynomial evaluated at θ. The entries of μ and L are represented as:

$$\mu_x = \sum_{k=0}^{K} \sum_{j=0}^{J} \sum_{i=0}^{I} \mathbf{P}_{x_{ijk}} \eta^i \alpha^j \theta^k, \tag{18}$$

$$L_{11} = \sum_{k=0}^{K} \frac{\mathbf{B}_k^K(\theta)}{\sum_{j=0}^{J} \sum_{i=0}^{I} \mathbf{P}_{l_{11ijk}} \eta^i \alpha^j}. \tag{19}$$

μ_y is represented similarly to μ_x, and L_{21} and L_{22} are represented similarly to L_{11}. To sum up, we need (K+1)(J+1)(I+1)

coefficients to represent each entry and there are five entries $(\mu_x, \mu_y, L_{11}, L_{21}, L_{22})$. So, we need 5(K+1)(J+1)(I+1) numbers in total.

In practice, we split the fitting into cases by reflection or refraction, and by low roughness or high roughness. For reflection, we observe that leaving out the Fresnel term does not have much influence on the final result and it saves us one dimension in the fitting because it is independent of IOR. For low roughness, we observe that a small fitting error in the mean would introduce a lot more variance in the samples compared to higher roughness. Thus, we adopt a modified representation and fitting scheme for roughness less than 0.1. For these cases, the specular direction serves as a quite accurate prediction of the mean. The specular direction converted to outgoing slope domain is denoted using μ^* and we let the mean of Gaussian be $\mu = \mu*$. L is represented the same as in (19). Note that the fitting discontinuity at 0.1 does not cause a problem in Monte Carlo estimation since we only use the fitted distribution for sampling.

Fitting algorithm. Algorithm 1 implements the parameterized 2D fitting method that outputs a parameterized representation of BSDF with respect to material properties and incident direction. This algorithm requires only the ability to evaluate the target BSDF function. We first collect information from the given BSDF for fitting: for a set of parameters $\{\eta, \alpha, \theta\}$, we evaluate BSDF values on a regular grid. As we mentioned previously, we split the fitting into cases by roughness.

When roughness is greater or equal to 0.1, we generate points by assuming that BSDF value is constant over each grid cell. Then, we solve a nonlinear optimization problem for the coefficients of the polynomials. The loss function is defined as minus the sum of log probability of the fitted Gaussian mixture generating the collected points:

$$\mathcal{L}_1 = -\sum_{\eta}\sum_{\alpha}\sum_{\theta}\sum_{m} ln\left(\mathcal{M}\left(\mathbf{x} = \mathbf{x}_m \mid \mu_{\theta\alpha\eta},\ \Sigma_{\theta\alpha\eta}\right)\right), \quad (20)$$

where we discretized the range of index of refraction, roughness and incident angle and η, α and θ are the discretization values; m is the index of the generated points.

When roughness is less than 0.1, we calculate the covariance matrix of each BSDF using numerical integration and compute the desired Cholesky factorization $\overline{L_{ijk}}$ of the precision matrix. Then, we solve another nonlinear optimization problem with the loss function defined as the sum of Frobenius norm of the difference L matrix:

$$\mathcal{L}_2 = \sum_{\eta}\sum_{\alpha}\sum_{\theta} \|L_{\theta\alpha\eta} - \overline{L_{\theta\alpha\eta}}\|_F. \quad (21)$$

To solve the nonlinear optimization problems, we use sequential least squares programming [Kra88]. In terms of initialization, we use parameterized fitting of a partial set of the variables to accelerate convergence.

Fitting performance. In practice, the representation is compact. For example, second- or third-order polynomials and one Gaussian per slice are good enough for fitting Microfacet Beckmann. A complete fitting to Microfacet Beckmann has 18–135 coefficients for each case. The total running time of the fitting is about 23 min. Although this preprocessing time seems long, the precomputation only needs to be conducted once for one kind of BSDF model and it is independent of scene and material properties. This makes our work much more practical in handling textured materials compared to [JRJ11]. Figure 7 visualizes the fitting. The top row shows a 2D BTDF slice, a 2D BRDF slice and their product. The bottom row shows the two corresponding slices of the parameterized Gaussian fitting and their product.

7. Implementation

We implemented our layered BSDF model in the PBRT framework. We mainly focus on layered BSDF evaluation in the previous sections. The global integrator also requires the ability to sampling the BSDF (i.e. generate one argument direction given the other) and to estimate the probability of this sampling. For these operations, we adopt the sampling and sample probability estimation methods from [GHZ18], which we summarize here for completeness.

BSDF sampling. Given incoming direction ω_i, we perform forward path tracing inside the layer stack until the path exits the layer stack with some outgoing direction ω_o. Assume this particular path through the layer stack contains directions $\psi_0, \ldots, \psi_{n+1}$, where $\psi_0 = -\omega_i$ and $\psi_{n+1} = \omega_o$. The nth corresponding weight for this sample is:

$$w(\psi_0, \ldots, \psi_{n+1}) = \frac{\prod_{k=0}^{n} f_{j_k}(-\psi_k, \psi_{k+1})}{\prod_{k=0}^{n} p_(\psi_{k+1}| -\psi_k)}. \quad (22)$$

PDF estimation. To estimate $p(\omega_o|\omega_i)$, we would ideally calculate the following:

$$p(\omega_o|\omega_i) = \sum_{\Omega} \int \cdots \int p(\psi_1|\omega_i) \cdots p(\omega_o|\psi_n)\, d\psi_1^{\perp} \cdots d\psi_n^{\perp}, \quad (23)$$

where Ω is the space of all path types. However, except for the single layer PDF, usually these integrals have no closed form solution. PDF estimation is typically only used to compute multiple-importance heuristic weights for combining BSDF and light sampling. Thus, for this purpose, we can return Monte Carlo estimates of this integral without introducing bias, as long as they are statistically independent of our BSDF estimates. Please refer to the Supplementary Material of [GHZ18] for the proof. Since this integral is very similar to the integral we are estimating for BSDF evaluation, the same methods can be used to estimate it. As suggested by Guo *et al.*, often these PDF estimates have lower accuracy requirements and can be estimated more approximately to reduce costs.

8. Results

In this section, we demonstrate the effectiveness and efficiency of our new sampling strategy on layered materials with various combinations of parameters. All renderings are generated using a

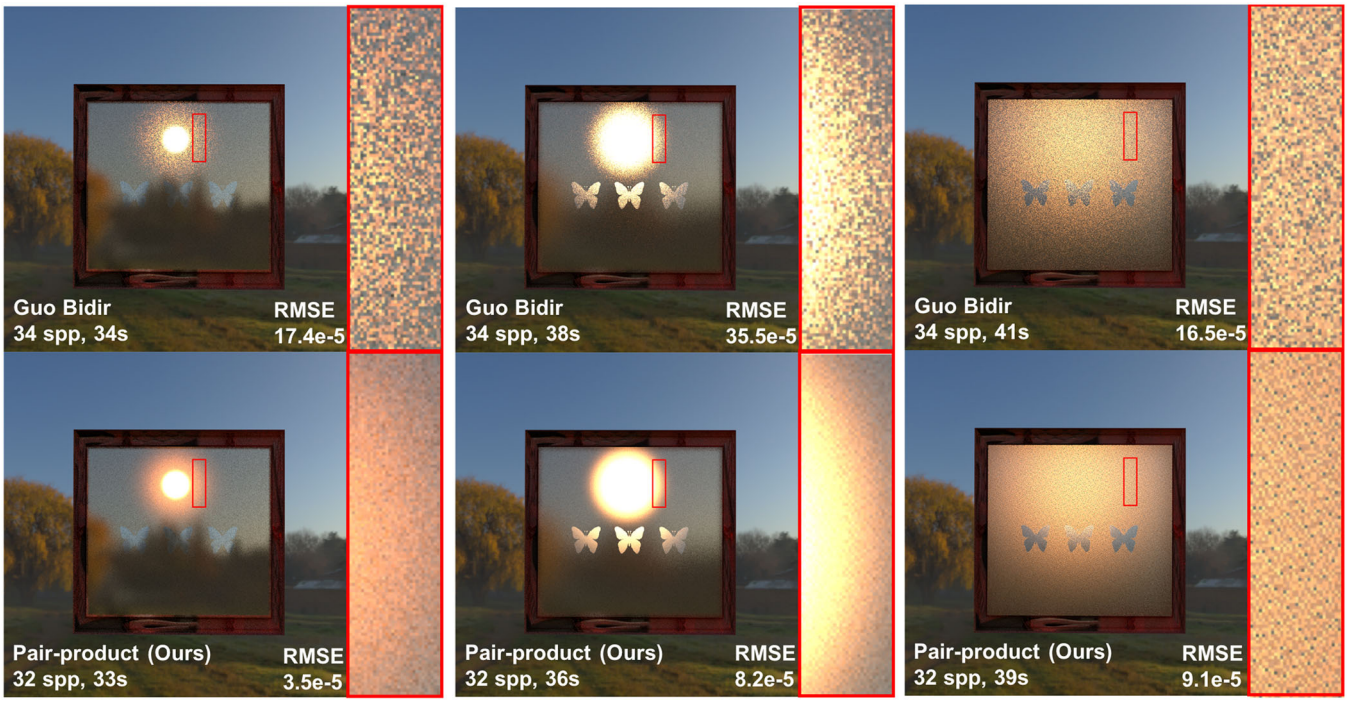

Figure 10: *This scene contains a double transmissive layer material (the framed area) with textured albedo and roughness. The scene is rendered using Guo et al.'s bidirectional (top row) and our pair-product sampling (bottom row). We rendered global illumination and the full path length through the layer stack. The red box shows the zoom area and is also used to calculate RMSE. The pair-product sampling obtains 1.8x to 5.0x reduction in RMSE, which corresponds to 3.2x to 25.0x reduction in render time.*

Figure 11: *A double textured layer car scene rendered using our pair-product sampling strategy. The top layer uses procedural noise to texture the roughness, and the bottom layer uses image maps to texture albedo and roughness. We rendered global illumination and the full path length through the layer stack with 1024 spp. The running time is 46 min.*

C++ implementation in the PBRT framework [PJH16] on an 8 core i7-6700K machine. The timings are reported using wall-clock time. For our performance analysis, we compute the per-pixel root mean square error (RMSE) across four renders of a specific patch of the image, with respect to a reference image, then take the mean over all the pixels in the patch. We compute RMSE reduction by calculating the ratio of RMSE of Guo *et al.*'s bidirectional method to our methods. Since Monte Carlo integration converges at a rate of $O(\sqrt{N})$ with N being the number of samples, we also report effective render time reduction as the square of RMSE reduction.

Reflective layer. Figure 5 compares forward sampling, Guo's bidirectional, our pair-product and our multiple-product on a double layer sphere scene, rendered with global illumination and up to the f_{010} component. The pair-product sampling obtains 5.4x and 1.8x reduction in RMSE compared to the forward sampling and the bidirectional, respectively. These correspond to 29.2x and 3.2x reduction in render time. The multiple-product sampling obtains 8.2x and 2.8x reduction in RMSE compared to the forward sampling and the bidirectional method, corresponding to 67.2x and 7.8x reduction in render time.

In Figure 8, we show an equal-time comparison of Guo *et al.*'s bidirectional method, our pair-product sampling and our multiple-product sampling on a double layer teapot scene, with global illumination and the full path length through the layer stack. We used the red box to compute RMSE. The pair-product sampling obtains 1.3x reduction in RMSE compared to Guo's bidirectional method, which corresponds to 1.7x reduction in render time; the multiple-product sampling obtains 1.4x reduction in RMSE compared to the bidirectional method, which corresponds to 2.0x reduction in render time.

Figure 9 compares Guo *et al.*'s bidirectional method to our method on a triple-layer reflective material with equal running

Figure 12: *This scene contains a triple transmissive layer material (the framed area) with textured albedo and roughness. The scene is rendered using Guo et al.'s bidirectional (top row) and our pair-product sampling (bottom row). We rendered global illumination and the full path length through the layer stack. The red box shows the zoom area and is also used to calculate RMSE. The pair-product sampling obtains 1.5x to 2.9x reduction in RMSE, which corresponds to 2.3x to 8.4x reduction in render time.*

time. We rendered global illumination and the full path length through the layer stack. Similar to the double layer case, we used the red box to compute RMSE. The pair- product sampling obtains 1.2x reduction in RMSE, which corresponds to 1.4x reduction in render time. We observe the improvement in this three layer case is slightly smaller than that in the double layer case. Our pair-product sampling can be seen as shortening the path through layer stack by one and replacing a pair of high variance terms in the estimator with a single lower variance term. We expect this improvement to be less obvious when the path length is longer.

Transmissive layer. Figure 10 shows an equal time comparison for transmission through a two-layer material, with global illumination and the full path length through the layer stack. The lighting is an environment map with a distant light source behind the framed area. Both the layers have textured albedo and textured roughness, with the butterflies having different colour and roughness from the rest of the surface. We used the red box to compute RMSE and our pair-product sampling obtains 2.6x to 5.1x reduction in RMSE, which corresponds to 6.8x to 26.0x reduction in render time.

Figure 12 shows an equal time comparison for triple transmission with the settings as the double layer case. Our pair-product sampling obtains 1.5x to 2.9x reduction in RMSE, which corresponds to 2.3x to 8.4x reduction in render time.

Textured material. Our layer model is able to handle textured materials. Each layer can have independent albedo, roughness and index of refraction maps. Figure 11 showcases a double layer material to render a car with scratched surface. The top layer uses procedural noise to texture the roughness, and the bottom layer has textured albedo and roughness via image maps. We rendered global illumination and the full path length through the layer stack.

Figure 1 compares Guo's bidirectional method and our pair-product sampling method on a scene with reflective layered materials, transmissive layered materials and textured layered materials. We rendered direct lighting only to avoid the indirect illumination noise that is unrelated to the layered BSDF evaluation. We report RMSE of each coloured box. Our method reduces RMSE by 1.6x to 3.3x on the coloured boxes, corresponding to 2.6x to 10.9x reduction in time.

9. Conclusion and future work

In this paper, we improved the previous Monte Carlo layer approaches by introducing two new more efficient sampling strategies for stochastic layer BSDF evaluation. The two strategies are pair-product sampling and multiple-product sampling. Pair-product sampling importance samples the product of two adjacent layers' BSDFs. Multiple-product sampling method is a generalization to the pair-product sampling method that can importance sample a chain of three or more BSDFs. They reduce stochastic layer evaluation variance by combining multiple BSDFs together and generating samples from distributions that are more proportional to the BSDF chain. Both sampling methods are based on a novel and compact parameterized Gaussian BSDF representation. This representation expresses 2D BSDF slices as bivariate Gaussians and writes

Gaussian parameters as smooth functions of BSDF parameters. Since it incorporates material properties, our layer model can handle arbitrary numbers of textured layers and only a one-time precomputation is required for one type of BSDF. Our results demonstrate substantial variance reduction compared to previous Monte Carlo layering approaches in rendering isotropic surface layers.

At the same time, our method has the following limitations, which leaves plenty of avenues for future work.

Generality. We have only demonstrated our new sampling strategies on isotropic surface scattering layers. We could easily integrate the BDPT estimator in [GHZ18], and for anisotropic and volume layers the system would fall back to the BDPT estimators. To handle anisotropic materials using our method, another dimension needs to be introduced into the parameterized representation. Depending on how anisotropic the material is, we might need to increase the number of Gaussians in the mixture. In order to handle volume scattering, we would fit a separate volume Gaussian representation for a given phase function. Specifically, we would separate the representation to upper and lower hemisphere to fit in our product sampling framework. Then, tracing the volume and generating connections from the volume should be added to the current framework.

Multiple-product sampling for longer paths. The pair-product sampling strategy works for an arbitrary layer configuration and an arbitrary path length. The multiple-product sampling method, in theory, works for an arbitrarily long path. However, in practice, we observe it only improves convergence when we sample three BSDFs at a time. The approximation error increases as the chain grows, especially for larger roughness, making this strategy not as effective as the simpler ones. One future direction would be utilizing other approximations instead of Gaussian mixtures in order to construct joint PDF approximations for the entire paths. They should more accurately approximate the whole chain of BSDFs as the chain grows longer, potentially leading to much larger gains in performance.

Acknowledgements

We are grateful to Mark Meyer, Tom Duff and Ryusuke Villemin for many helpful suggestions. This work was supported by the National Science Foundation under award 1704540.

Appendix A: Gaussian product

Let $N_1(\mu_1,\ \Sigma_1)$ and $N_2(\mu_2,\ \Sigma_2)$ be two Gaussian distributions, s be the normalization factor and $N_{12}(\mu,\ \Sigma)$ be the Gaussian distribution after normalization:

$$N_{12}(\mu,\ \Sigma) = \frac{1}{s} N_1(\mu_1,\ \Sigma_1) N_2(\mu_2,\ \Sigma_2), \tag{A.1}$$

where

$$\mu = \Sigma\left(\Sigma_1^{-1}\mu_1 + \Sigma_2^{-1}\mu_2\right), \tag{A.2}$$

$$\Sigma = \left(\Sigma_1^{-1} + \Sigma_2^{-1}\right)^{-1}, \tag{A.3}$$

and constant

$$s = N(\mu_1 | \mu_2,\ \Sigma_1 + \Sigma_2). \tag{A.4}$$

The product of two Gaussian mixtures with n and m components, respectively, is therefore a new Gaussian mixture with nm components.

Appendix B: 4D Gaussian approximation to BSDF

The Gaussian distribution can be written as the exponential of a quadratic polynomial

$$N(x|\mu,\ V) = e^{Q} = e^{-\frac{1}{2}(x-\mu)^T V (x-\mu)+c} = e^{-\frac{1}{2}x^T V x + bx + c}. \tag{B.1}$$

Here, μ, V and c are the 4D mean, 4×4 inverse covariance (precision) matrix and log normalization constant, and $b = \mu^T V$. With a second-order Taylor expansion, we can get a function that matches the parameterized fit exactly in value and first and second derivatives at a point. It is sufficient to only consider the quadratic polynomial. We solve the linear system

$$\frac{dQ}{dx} = -V x_0 + b,$$

$$\frac{d^2 Q}{dx^2} = -V,$$

to find a quadratic polynomial $Q(x)$ that matches to second order. Since in our construction V is symmetric but not necessarily positive definite (SPD), it is not guaranteed that we will end up with a distribution that can be sampled. In practice, we find that for most of the cases, after multiplying the two 2D Gaussians that approximate $f_j(\omega, \cdot)$ and $f_\ell(\cdot, \omega')$, we will end up with a Gaussian whose covariance is SPD. For the non-SPD cases, we modify the negative eigenvalues to a small positive number ϵ.

References

[Bel18] Belcour L.: Efficient rendering of layered materials using an atomic decomposition with statistical operators. *ACM Transactions on Graphics 37*, 4 (July 2018), 73:1–73:15.

[BM58] Box G. E. P., Muller M. E.: A note on the generation of random normal deviates. *Annals of Mathematical Statistics 29*, 2 (June 1958), 610–611.

[BPB19] Bati M., Pacanowski R., Barla P.: *Numerical Analysis of Layered Materials Models*. Research Report, Institut d'Optique Graduate School; Université de Bordeaux; CNRS; INRIA. Jun, 2019. https://hal.inria.fr/hal-02157966.

[CJAMJ05] Clarberg P., Jarosz W., Akenine-Möller T., Jensen H. W.: Wavelet importance sampling: Efficiently evaluating products of complex functions. *ACM Transactions on Graphics 24*, 3 (July 2005), 1166–1175.

[DH96] Dorsey J., Hanrahan P.: Modeling and rendering of metallic patinas. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1996), SIGGRAPH '96, ACM, pp. 387–396.

[DHI*13] Dupuy J., Heitz E., Iehl J.-C., Poulin P., Neyret F., Ostromoukhov V.: Linear efficient antialiased displacement and reflectance mapping. *ACM Transactions on Graphics 32*, 6 (2013), 211:1–211:11.

[GHZ18] Guo Y., Hašan M., Zhao S.: Position-free Monte Carlo simulation for arbitrary layered BSDFs. In *SIGGRAPH Asia 2018 Technical Papers* (New York, NY, USA, 2018), SIGGRAPH Asia '18, ACM, pp. 279:1–279:14.

[GKH*13] Georgiev I., Křivánek J., Hachisuka T., Nowrouzezahrai D., Jarosz W.: Joint importance sampling of low-order volumetric scattering. *ACM Transactions on Graphics 32*, 6 (November 2013), 164:1–164:14.

[HES*18] Herholz S., Elek O., Schindel J., Křivánek J., Lensch H. P. A.: A unified manifold framework for efficient BRDF sampling based on parametric mixture models. In *Proceedings of the Eurographics Symposium on Rendering: Experimental Ideas & Implementations* (Goslar, Germany, 2018), EGSR '18, Eurographics Association, pp. 41–52.

[HEV*16] Herholz S., Elek O., Vorba J., Lensch H., Křivánek J.: Product importance sampling for light transport path guiding. *Computer Graphics Forum 35*, 4 (July 2016), 67–77.

[HKL14] Hery C., Kass M., Ling J.: *Geometry into Shading*, 2014.

[HVL17] Hery C., Villemin R., Ling J.: Pixar's foundation for materials. *SIGGRAPH Course, Physically Based Shading* (2017).

[IFDN12] Iwasaki K., Furuya W., Dobashi Y., Nishita T.: Real-time rendering of dynamic scenes under all-frequency lighting using integral spherical Gaussian. *Computer Graphics Forum 31* (2012), 727–734.

[JdJM14] Jakob W., d'Eon E., Jakob O., Marschner S.: A comprehensive framework for rendering layered materials. *ACM Transactions on Graphics 33*, 4 (July 2014), 118:1–118:14.

[JRJ11] Jakob W., Regg C., Jarosz W.: Progressive expectation-maximization for hierarchical volumetric photon mapping. In *Proceedings of the 22nd Eurographics Conference on Rendering* (Aire-la-Ville, Switzerland, 2011), EGSR '11, Eurographics Association, pp. 1287–1297.

[KC17] Kulla C., Conty A.: Revisiting physically based shading at imageworks. *SIGGRAPH Course, Physically Based Shading* (2017).

[Kra88] Kraft D.: A software package for sequential quadratic programming. *Forschungsbericht- Deutsche Forschungs- und Versuchsanstalt fur Luft- und Raumfahrt* (1988).

[MJC*03] Marschner S. R., Jensen H. W., Cammarano M., Worley S., Hanrahan P.: Light scattering from human hair fibers. *ACM Transactions on Graphics 22*, 3 (2003), 780–791.

[OB10] Olano M., Baker D.: Lean mapping. In *Proceedings of the 2010 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games* (New York, NY, USA, 2010), I3D '10, ACM, pp. 181–188.

[PJH16] Pharr M., Jakob W., Humphreys G.: *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann, 2016.

[PJJ*11] Papas M., Jarosz W., Jakob W., Rusinkiewicz S., Matusik W., Weyrich T.: Goal-based caustics. *Computer Graphics Forum 30* (2011), 503–511.

[SC97] Shirley P., Chiu K.: A low distortion map between disk and square. *Journal of Graphics Tools 2*, 3 (December 1997), 45–52.

[SM06] Sun W., Mukherjee A.: Generalized wavelet product integral for rendering dynamic glossy objects. In *ACM SIGGRAPH 2006 Papers* (New York, NY, USA, 2006), SIGGRAPH '06, ACM, pp. 955–966.

[Sta01] Stam J.: An illumination model for a skin layer bounded by rough surfaces. In *Proceedings of the 12th Eurographics Conference on Rendering* (Aire-la-Ville, Switzerland, 2001), EGWR'01, Eurographics Association, pp. 39–52.

[TS06] Tsai Y.-T., Shih Z.-C.: All-frequency precomputed radiance transfer using spherical radial basis functions and clustered tensor approximation. *ACM Transactions on Graphics 25*, 3 (July 2006), 967–976.

[Vea97] Veach E.: *Robust Monte Carlo Methods for Light Transport Simulation*. Vol. 1610. PhD thesis, Stanford University, 1997.

[VG95] Veach E., Guibas L. J.: Optimally combining sampling techniques for Monte Carlo rendering. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1995), SIGGRAPH '95, ACM, pp. 419–428.

[VKv*14] Vorba J., Karlík O., Šik M., Ritschel T., Křivánek J.: On-line learning of parametric mixture models for light transport simulation. *ACM Transactions on Graphics 33*, 4 (July 2014), 101:1–101:11.

[WW07] Weidlich A., Wilkie A.: Arbitrarily layered micro-facet surfaces. In *Proceedings of the 5th International Conference on Computer Graphics and Interactive Techniques in Australia and Southeast Asia* (New York, NY, USA, 2007), GRAPHITE '07, ACM, pp. 171–178.

[WWD*06] Wang L., Wang W., Dorsey J., Yang X., Guo B., Shum H.-Y.: Real-time rendering of plant leaves. In *ACM SIGGRAPH 2006 Courses* (New York, NY, USA, 2006), SIGGRAPH '06, ACM, p. 5.

[XMR*11] XU K., MA L.-Q., REN B., WANG R., HU S.-M.: Interactive hair rendering and appearance editing under environment lighting. In *Proceedings of the 2011 SIGGRAPH Asia Conference* (New York, NY, USA, 2011), SA '11, ACM, pp. 173:1–173:10.

[XSD*13] XU K., SUN W.-L., DONG Z., ZHAO D.-Y., WU R.-D., HU S.-M.: Anisotropic spherical Gaussians. *ACM Transactions on Graphics 32*, 6 (November 2013), 209:1–209:11.

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

[YHW*18] YAN L.-Q., HAŠAN M., WALTER B., MARSCHNER S., RAMAMOORTHI R.: Rendering specular microgeometry with wave optics. *ACM Transactions on Graphics 37*, 4 (July 2018), 75:1–75:10.

[ZJ18] ZELTNER T., JAKOB W.: The layer laboratory: A calculus for additive and subtractive composition of anisotropic surface reflectance. *ACM Transactions on Graphics 37*, 4 (July 2018), 74:1–74:14.