A Learned Shape-Adaptive Subsurface Scattering Model

DELIO VICINI, Ecole Polytechnique Fédérale de Lausanne (EPFL)
VLADLEN KOLTUN, Intel Labs
WENZEL JAKOB, Ecole Polytechnique Fédérale de Lausanne (EPFL)

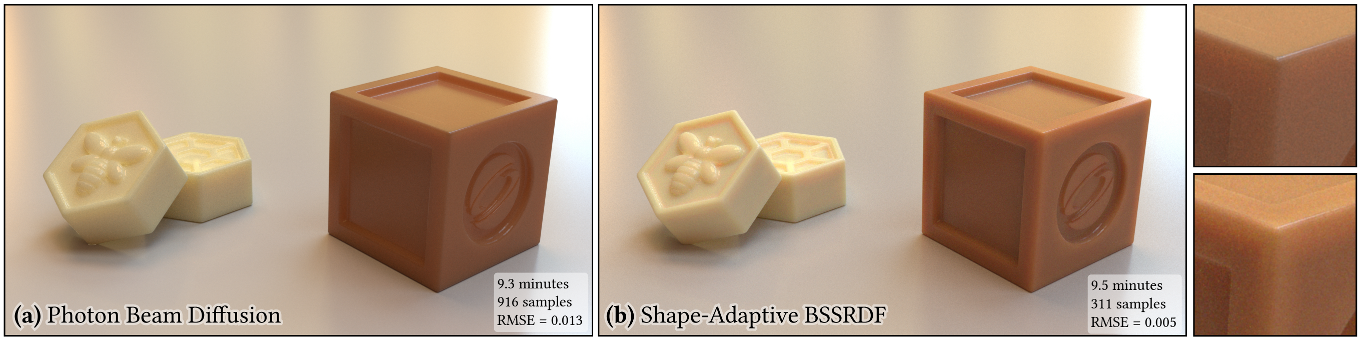


Fig. 1. Rendering of translucent soap blocks with significant anisotropy ($g = 0.9$). **(a)** Dipole diffusion-based models such as Photon Beam Diffusion [Habel et al. 2013] yield overall flat appearance due to their internal assumption of isotropy and plane-parallel light transport. **(b)** Our learned subsurface scattering model adapts to both geometry and anisotropy, producing more realistic appearance and lower error compared to a path-traced reference. Accounting for the geometry leads to increased scattering around the silhouette and overall higher contrast in regions with geometric detail. Our method has constant sampling weights, hence renderings converge with fewer samples. Please see the supplemental material for an interactive version of this figure including error maps.

Subsurface scattering, in which light refracts into a translucent material to interact with its interior, is the dominant mode of light transport in many types of organic materials. Accounting for this phenomenon is thus crucial for visual realism, but explicit simulation of the complex internal scattering process is often too costly. BSSRDF models based on analytic transport solutions are significantly more efficient but impose severe assumptions that are almost always violated, e.g. planar geometry, isotropy, low absorption, and spatio-directional separability. The resulting discrepancies between model and usage lead to objectionable errors in renderings, particularly near geometric features that violate planarity.

This article introduces a new shape-adaptive BSSRDF model that retains the efficiency of prior analytic methods while greatly improving overall accuracy. Our approach is based on a conditional variational autoencoder, which learns to sample from a reference distribution produced by a brute-force volumetric path tracer. In contrast to the path tracer, our autoencoder directly samples outgoing locations on the object surface, bypassing a potentially lengthy internal scattering process.

The distribution is conditional on both material properties and a set of features characterizing geometric variation in a neighborhood of the incident location. We use a low-order polynomial to model the local geometry as an implicitly defined surface, capturing curvature, thickness, corners, as well as cylindrical and toroidal regions. We present several examples of objects with challenging medium parameters and complex geometry and compare to ground truth simulations and prior work.

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

ACM Reference Format:
Delio Vicini, Vladlen Koltun, and Wenzel Jakob. 2019. A Learned Shape-Adaptive Subsurface Scattering Model. *ACM Trans. Graph.* 38, 4, Article 127 (July 2019), 15 pages. https://doi.org/10.1145/3306346.3322974

1 INTRODUCTION

A variety of materials ranging from organic substances like skin or fruits to liquids and translucent stone exhibit significant subsurface light transport, where light that enters the surface at one point exits some distance away. This process is generally modeled using the radiative transfer equation (RTE) and involves long sequences of scattering interactions within a medium that fills the object's interior. Although Monte Carlo methods can be used to solve the resulting integration problem, this tends to be impracticably expensive for many real-world materials due to their high albedo, anisotropy, and density.

For instance, Narasimhan et al. [2006] measured the scattering parameters of a large number of materials and report that milk has an albedo of $\alpha \approx 0.99959$, meaning that light will scatter an expected number of 2439 times before being absorbed. Furthermore, scattering is strongly forward-peaked ($g \approx 0.9$), and light will thus tend to penetrate deeply into the object.

An alternative approach for rendering subsurface scattering without the need for costly internal scattering simulations involves the notion of a *bidirectional scattering-surface reflectance distribution function* (BSSRDF). BSSRDF models directly encode surface-to-surface transport and are typically based on analytic solutions of simplified light transport problems, such as solutions of the

Authors' addresses: Delio Vicini, Ecole Polytechnique Fédérale de Lausanne (EPFL), delio.vicini@epfl.ch; Vladlen Koltun, Intel Labs, vladlen.koltun@intel.com; Wenzel Jakob, Ecole Polytechnique Fédérale de Lausanne (EPFL), wenzel.jakob@epfl.ch.

0730-0301/2019/7-ART127 $15.00
https://doi.org/10.1145/3306346.3322974

diffusion equation in a plane-parallel setting. While significantly more efficient, these models impose severe assumptions that are almost always violated—standard steps in the derivation of diffusion models for example postulate half-space or slab geometry and low anisotropy/absorption. Another assumption is that of *separability*, i.e. that the BSSRDF decomposes into a product of functions with spatial or directional dependence. The resulting discrepancies between model and actual usage lead to objectionable errors in rendered images, particularly near geometric features that violate planarity. In recent years, BSSRDF models have fallen out of favor due to these problems, with major rendering systems adopting or falling back to Monte Carlo sampling of the radiative transfer equation despite the associated computational expense [Christensen et al. 2018; Fascione et al. 2018].

In this article, we introduce a new shape-adaptive BSSRDF model with support for arbitrary homogeneous medium parameters. Our method retains the efficiency of classical BSSRDF models without imposing the typical assumptions of diffusion models, greatly improving overall accuracy. Our technique relies on a combination of three neural networks that together constitute a probabilistic generative model for BSSRDF sampling on general curved surfaces.

The first is a conditional variational autoencoder [Kingma and Welling 2013], which learns to sample from a reference distribution produced by a volumetric path tracer. The second is a multilayer perceptron that regresses the distribution's scale factor to allow for absorbing materials. Both are conditioned on a number of parameters that are pre-processed by a third feature network. These include material properties (albedo, extinction, anisotropy, and index of refraction), the incident direction, and a descriptor characterizing geometric variation in a neighborhood of the incident location. The networks admit a compact implementation that is efficient enough to support evaluation at every surface intersection. To describe the local geometry, we rely on a low-order trivariate polynomial that encodes an approximate signed-distance function, making the model adaptive to geometric detail including curvature, thickness, corners, as well as cylindrical and toroidal extrusions (e.g. handles).

We jointly train all three networks on a reference dataset of transport paths generated by a brute-force Monte Carlo simulation involving geometry retrieved from a repository of shapes of varying complexity, including CAD models and 3D scans. Training is scene-independent and only requires the set of final path vertices, where light refracts out of the training shapes. This yields an efficient BSSRDF sampling scheme with uniform weights that bypasses the lengthy internal scattering process. Our technique makes two central approximations. The first is that we model the geometry around a shading point as a smooth implicit surface, which potentially entails some loss of detail. The second is that we replace the ground-truth transport on this implicit surface by a learned distribution.

We analyze the performance and accuracy of our model on scenes with challenging geometry and material properties, and we also investigate the effect of our approximations individually. Our method achieves significantly lower error than state-of-the-art BSSRDF models including those of Habel et al. [2013], Frisvad et al. [2014], and Frederickx et al. [2017] using a comparable time budget.

2 RELATED WORK

The *radiative transfer equation* (RTE) arises in a wide variety of fields including atmospheric science, neutron transport, astronomy, and medical physics. The scalar form that is commonly used in computer graphics can be derived by analyzing scattering within a medium containing randomly distributed and uncorrelated particles [Chandrasekhar 1960; Ishimaru 1999]. A polarized form derived from wave optics was presented by Mishchenko [2006].

Monte Carlo methods. The most general techniques for solving the RTE are based on variants of Monte Carlo path tracing [Hanrahan and Krueger 1993; Kajiya 1986; Kajiya and Von Herzen 1984] and involve nested sampling of scattering events to form complete light paths. Paths can also be generated bidirectionally, e.g. using density estimators based on photon [Jensen and Christensen 1998] or ray primitives [Novák et al. 2012]. Path sampling approaches tend to be very costly when rendering highly-scattering media, where light potentially undergoes thousands of interactions before leaving the material. Absorption is another source of inefficiency, since considerable time may be spent generating paths that propagate deeply into the material without contributing to the rendered image. Dwivedi sampling [Křivánek and d'Eon 2014; Meng et al. 2016] improves statistical efficiency in such cases by biasing the random walk toward the surface.

Due to the high costs of volumetric path tracing, many works have proposed BSSRDF models that cast subsurface scattering into a considerably lower-dimensional integral over surface positions and incident directions. Explicit discretization of the 8-dimensional parameter space of BSSRDFs usually remains elusive, though simplifying assumptions such as homogeneous planar geometry can reduce the dimensionality sufficiently to make approximate tabulation of Monte Carlo simulations feasible [Donner et al. 2009].

Diffusion theory. The vast majority of BSSRDF models in computer graphics are based on analytic transport solutions derived from diffusion theory, which can be interpreted as a directional first-order expansion of the RTE [Ishimaru 1999]. Starting with the Green's function of the diffusion equation in an infinite volume, these methods approximate planar boundary conditions using the method of images, resulting in dipole [Jensen et al. 2001] or multipole [Donner and Jensen 2005] configurations that can be evaluated using efficient tree data structures [Jensen and Buhler 2002], ray tracing-based projections [Jensen et al. 2001; King et al. 2013; Walter et al. 2012] or discrete convolutions [d'Eon et al. 2007; Jimenez et al. 2009]. Since they only account for multiply scattered light, these models must be combined with a separate single scattering term [Hanrahan and Krueger 1993].

Classic diffusion tends to produce severe errors when simulating absorbing media, and later BSSRDF models adopt modified diffusion solutions and improved boundary conditions that rectify these problems [d'Eon 2012; D'Eon and Irving 2011; Habel et al. 2013]. Other diffusion-based models introduce support for anisotropic structures [Jakob et al. 2010] or an added dependence on the angle of incidence that leads to an anisotropic profile [Frisvad et al. 2014; Habel et al. 2013]. A general limitation of analytic diffusion models is the built-in plane-parallel assumption that introduces errors when rendering non-planar objects. Finite element solutions that solve the diffusion equation on a discrete domain composed of voxels or tetrahedral elements [Arbree et al. 2011; Stam 1995; Wang et al. 2008] improve accuracy but lack the efficiency of analytic methods. A dipole model for spherical geometry was presented by Kolchin [2010].

Feynman path integration. An interesting approach for highly forward-scattering media was pioneered by Tessendorf [1987]. In this setting, the trajectory of light can be approximated as a smoothly curving continuous path. Applied to the radiative transfer equation, this approximation turns the discrete integral over scattering interactions into a Feynman path integral over all possible continuous paths. Premože et al. [2003] use this framework to derive an approximate BSSRDF model based on the concept of the *most probable path.* Motivated by a connection to polymer physics, Frederickx et al. [2017] derive an approximate solution of the path integral in an infinite medium and use it to create a plane-parallel BSSRDF model based on a dipole configuration.

Neural networks. Neural networks are a widely used building block for constructing expressive nonlinear function approximators, classifiers, and generative models (see LeCun et al. [2015] for a general overview). Applications in computer graphics include denoising of images rendered using Monte Carlo techniques [Bako et al. 2017; Chaitanya et al. 2017; Vogels et al. 2018], approximate shading of surfaces [Hermosilla et al. 2018] and heterogeneous volumes [Kallweit et al. 2017], and shading in screen space [Nalbach et al. 2017].

Two related recent articles by Müller et al. [2018] and Zheng and Zwicker [2018] propose methods that learn to sample light paths in a scene using extensions of the Real NVP ("non-volume preserving") network architecture [Dinh et al. 2016]. These methods could in principle be applied to render subsurface scattering, although the high dimensionality of volumetric path space is likely prohibitive. The main limitation of this approach is that training runs concurrently with rendering and adds considerable overhead.

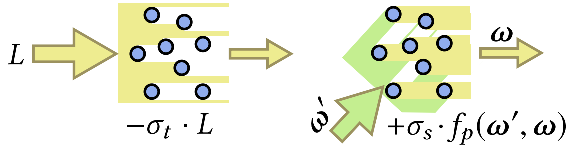
3 BACKGROUND

We will briefly review the standard radiative transfer formulation that gives rise to subsurface scattering along with similarity theory and the core concepts of the variational autoencoder.

Radiative transfer. Light transport within the interior of an object is generally simulated using the *radiative transfer equation* (RTE), which models the material as a suspension of scattering and absorbing particles that are too small and numerous to be individually discernible and can hence be represented using density functions. The RTE

$$\begin{aligned}(\boldsymbol{\omega}\cdot\nabla)\, L(\mathbf{x},\boldsymbol{\omega}) = &-\sigma_t L(\mathbf{x},\boldsymbol{\omega}) \\ &+ \sigma_s \int_{S^2} L(\mathbf{x},\boldsymbol{\omega}')\, f_p(\boldsymbol{\omega}',\boldsymbol{\omega})\,\mathrm{d}\boldsymbol{\omega}', \qquad (\mathbf{x}\in\Omega)\end{aligned} \tag{1}$$

is defined on a domain Ω and relates the directional change of radiance L in direction $\boldsymbol{\omega}$ to a sum of two effects: *extinction* proportional to $\sigma_t = \sigma_a + \sigma_s$ causes light traveling along the ray to be absorbed (σ_a) or scattered (σ_s) into other directions. *In-scattering* proportional to σ_s redirects light arriving from other directions $\boldsymbol{\omega}'$ so that it now travels in direction $\boldsymbol{\omega}$. The phase-function f_p describes the change in direction when light is scattered by the medium. We use the Henyey-Greenstein [1941] phase function, which is parameterized by its *mean cosine* $g \in [-1, 1]$. A value of $g = 0$ indicates an *isotropic* medium, where light becomes directionally diffuse after a single interaction, and $g \approx 1$ indicates strongly forward-peaked scattering that approximately preserves the direction of propagation. Most translucent materials are highly anisotropic with values of $g > 0.9$ [Narasimhan et al. 2006]. We denote the *scattering albedo* of a single medium interaction as $\alpha = \sigma_s \,/\, \sigma_t$.



Note that the RTE is only valid on object's interior, while surface positions satisfy the following boundary condition:

$$L_o(\mathbf{x},\boldsymbol{\omega}) = \int_{S^2} L_i(\mathbf{x},\boldsymbol{\omega}')\, f_s(\boldsymbol{\omega}',\boldsymbol{\omega})\, |\boldsymbol{\omega}'\cdot\boldsymbol{\omega}|\,\mathrm{d}\boldsymbol{\omega}'. \qquad (\mathbf{x}\in\partial\Omega) \tag{2}$$

Here, f_s is the surface's *bidirectional scattering distribution function* (BSDF), which models surface interactions including total internal reflection and angular deviation of rays moving through an index of refraction change. In this work, we only consider smooth boundaries with relative indices of refraction ranging from $\eta \approx 1.0 \ldots 1.5$, although our method could in principle be trained to work with other BSDFs. We refer the reader to Pharr et al. [2016] for a review of surface light transport.

The BSSRDF. Equations (1) and (2) together constitute a full specification of subsurface light transport. Given an incident radiance function $L_i(\mathbf{x},\boldsymbol{\omega})$ defined for all $\mathbf{x} \in \partial\Omega$ and outward-facing directions $\boldsymbol{\omega}$ (i.e. $\mathbf{n}_\mathbf{x}\cdot\boldsymbol{\omega} > 0$), they allow us to solve for the corresponding outgoing illumination $L_o(\mathbf{x},\boldsymbol{\omega})$, for example using a volumetric path tracer. Since the underlying physical process is entirely linear, the relationship between outgoing and incident radiance can be condensed into a continuous scattering kernel S:

$$L_o(\mathbf{x},\boldsymbol{\omega}) = \int_{\partial\Omega}\int_{S^2} L_i(\mathbf{x}',\boldsymbol{\omega}')\, S(\mathbf{x}',\boldsymbol{\omega}',\mathbf{x},\boldsymbol{\omega})\,\mathrm{d}\boldsymbol{\omega}'\,\mathrm{d}\mathbf{x}'. \tag{3}$$

The function S is known as the *bidirectional scattering-surface reflectance distribution function* (BSSRDF). At a first glance an eight-dimensional function, the BSSRDF is in fact much higher-dimensional due to its dependence on the properties of the interior medium $\varphi_\Omega = (\sigma_t, \sigma_s, g)$ and the shape of the boundary $\partial\Omega$.

The goal of this article is to realize a probabilistic generative algorithm for sampling $(\mathbf{x}',\boldsymbol{\omega}')$ proportional to $S_\varphi(\cdot,\cdot,\mathbf{x},\boldsymbol{\omega})$ given a fixed incident ray $\mathbf{r}_i = (\mathbf{x},\boldsymbol{\omega})$ and parameters $\varphi = (\varphi_\Omega, \varphi_{\partial\Omega})$ describing the interior and boundary. Note that the dependence of S_φ on both $\mathbf{r}_i$ and φ is *highly nonlinear*—we rely on neural networks to *learn* these nonlinear dependencies from ground-truth data.

Similarity theory. Consider a hypothetical medium without absorption ($\sigma_a = 0$) and maximal forward scattering ($g = 1$) that causes no angular deviation of light. For all intents and purposes, such a material is equivalent to vacuum regardless of its density. This observation applies more broadly: for any set of medium parameters φ_Ω, we can find *similar* alternative parameters φ'_Ω. The underlying theory of similarity relations was studied by Wyman et al. [1989], who proved equivalence under directionally band-limited illumination. We exploit similarity theory to re-parameterize φ_Ω in a more perceptually uniform space, improving training performance. Higher-order similarity relations were studied by Zhao et al. [2014] and could also be used with our method.

Variational autoencoders. To our knowledge, variational autoencoders have not been used in rendering to date, hence we provide a review motivated by the requirements of radiative transport. Our discussion roughly follows Doersch [2016].

Subsurface scattering can be identified with a set of unobserved latent variables $\mathbf{z} \in Z$ that fully characterize the underlying stochastic process. *Latent* here refers to our inability to know the precise value of $\mathbf{z}$ that caused light to eventually leave the surface at a particular position $\mathbf{x} \in \partial\Omega$. One choice that clearly fits this description is the path $\mathbf{z} = (\mathbf{z}_1, \ldots, \mathbf{z}_n)$ of internal scattering events (where $\mathbf{z}_1 = \mathbf{x}_i$ and $\mathbf{z}_n = \mathbf{x}$). However, the main objective of this work is to find an alternative that sidesteps the complexities of the high-dimensional space of interior light paths.

An arguably easier sampling technique could e.g. first decide whether to focus on the front or back surface of a potentially thin surface region as seen from $\mathbf{x}$, draw a sample from a suitable distribution on a Euclidean domain, and then map it onto the boundary $\partial\Omega$, where each step uses information encoded in an associated latent vector. Although such an algorithm is not readily available, it should be clear that dimensionality of Z can be much smaller than the full path space.

We use a *variational autoencoder* (VAE) [Kingma and Welling 2013] to learn a suitable mapping between sampled positions and latent variables $\mathbf{z}$. The VAE consists of two parts: a neural network $\mathbf{g}$ in the role of a high-capacity function approximator maps an as-of-yet unspecified vector of latent variables $\mathbf{z}$ to a surface position $\mathbf{g}(\mathbf{z}\,|\,\theta, \varphi)$—this is referred to as *decoding* $\mathbf{z}$. A separate *encoder* network $\mathbf{q}(\mathbf{x}\,|\,\theta, \varphi)$ takes a surface position $\mathbf{x}$ as input and returns randomly sampled latent variables corresponding to light transport that produces $\mathbf{x}$ with high probability. Both $\mathbf{g}$ and $\mathbf{q}$ are conditioned on trainable parameters θ of the neural networks and a descriptor φ characterizing the local surface and material properties. We will omit both for readability. Note that the encoder network internally draws samples from a probability distribution, hence $\mathbf{q}$ is a random variable. Although $\mathbf{g}$ is deterministic given $\mathbf{z}$, the following discussion simplifies if we consider it to be a normally distributed quantity with density $p_\mathbf{x}(\mathbf{x} \mid \mathbf{z}) = \mathcal{N}(\mathbf{x}; \mathbf{g}(\mathbf{z}), \sigma^2)$ and a small variance σ^2. Integrating over the latent variables then yields the marginal density produced by the decoder:

$$p_\mathbf{x}(\mathbf{x}) = \int_Z p_\mathbf{x}(\mathbf{x} \mid \mathbf{z})\, p_\mathbf{z}(\mathbf{z})\, \mathrm{d}\mathbf{z}, \tag{4}$$

where $p_\mathbf{z}(\mathbf{z})$ is the distribution of the latent variables. The aforementioned requirement that the encoder $\mathbf{q}$ produces latent variables which decode to $\mathbf{x}$ with high probability can be made precise by considering the Kullback-Leibler divergence between these distributions:

$$\begin{aligned} &D[p_\mathbf{q}(\mathbf{z}\,|\,\mathbf{x}) \parallel p_\mathbf{z}(\mathbf{z}\,|\,\mathbf{x})] \\ &\quad = E_{\mathbf{z}\sim\mathbf{q}(\mathbf{x})}\left[\log p_\mathbf{q}(\mathbf{z}\,|\,\mathbf{x}) - \log p_\mathbf{z}(\mathbf{z}\,|\,\mathbf{x})\right] \end{aligned} \tag{5}$$

and an application of Bayes' theorem to the second term yields

$$= E_{\mathbf{z}\sim\mathbf{q}(\mathbf{x})}\left[\log p_\mathbf{q}(\mathbf{z}\,|\,\mathbf{x}) - \log p_\mathbf{x}(\mathbf{x}\,|\,\mathbf{z}) - \log p_\mathbf{z}(\mathbf{z})\right] + \log p_\mathbf{x}(\mathbf{x})$$

which can be rearranged as

$$\begin{aligned} &\log p_\mathbf{x}(\mathbf{x}) - D[p_\mathbf{q}(\mathbf{z}\,|\,\mathbf{x}) \parallel p_\mathbf{z}(\mathbf{z}\,|\,\mathbf{x})] \\ &\quad = E_{\mathbf{z}\sim\mathbf{q}(\mathbf{x})}\left[\log p_\mathbf{x}(\mathbf{x}\,|\,\mathbf{z})\right] - D[p_\mathbf{q}(\mathbf{z}\,|\,\mathbf{x}) \parallel p_\mathbf{z}(\mathbf{z})] \end{aligned} \tag{6}$$

The above equation encapsulates the core of the variational autoencoder: the terms on the left hand side are both desirable optimization objectives that are challenging to compute or optimize individually. The first, $\log p_\mathbf{x}(\mathbf{x})$, is the log-likelihood of the generator, which we would like to maximize for values $\mathbf{x}_0, \mathbf{x}_1, \ldots$ drawn from a reference distribution, and the second KL-divergence term is the objective from Equation (5). Due to the equality, we can simultaneously handle both objectives by applying gradient ascent to the tractable right-hand side instead.

The standard VAE construction introduces two assumptions at this point—the first is that the latent variables are distributed according to a standard normal distribution, which is reasonable since $\mathbf{z}$ can internally be warped to any suitable distribution as long as the encoder and decoder have high capacity and Z is sufficiently high-dimensional. Furthermore, it is assumed that $\mathbf{q}$ consists of a deterministic neural network portion that computes the mean $\boldsymbol{\mu}$ and a diagonal covariance matrix Σ of a multivariate normal distribution that is sampled to produce $\mathbf{z}$. This provides an explicit expression for the second term on the right hand side of Equation (6):

$$D[p_\mathbf{q}(\mathbf{z}\,|\,\mathbf{x}) \parallel p_\mathbf{z}(\mathbf{z})] = \frac{1}{2}\left[\operatorname{tr}\Sigma + \boldsymbol{\mu}^T\boldsymbol{\mu} - \dim Z - \log\det\Sigma\right] \tag{7}$$

The expectation term in Equation (6) is approximated using a 1-sample Monte Carlo estimator that encodes and decodes $\mathbf{x}$ once:

$$E_{\mathbf{z}\sim\mathbf{q}(\mathbf{x})}\left[\log p_\mathbf{x}(\mathbf{x}\,|\,\mathbf{z})\right] \approx \log p_\mathbf{x}(\mathbf{x}\,|\,\mathbf{q}(\mathbf{x})) = \log \mathcal{N}(\mathbf{x}; \mathbf{g}(\mathbf{q}(\mathbf{x})), \sigma^2). \tag{8}$$

After optimizing Equation (6) with respect to the neural network coefficients θ for many sampled material configuration $\varphi_i \in \Phi$, the VAE can generate samples that approximately follow the reference distribution S_φ. For this, we simply discard the encoder portion and directly feed the decoder with normal variates.

4 METHOD

We now turn to the specifics of our BSSRDF model. We discuss central assumptions and present our neural network architecture and associated training procedure. Following this, we introduce geometry and material descriptors and explain how the model is used at render time.

Assumptions. Existing models for subsurface scattering generally assume that the BSSRDF factorizes into a product of the form

$$S(\mathbf{x}', \boldsymbol{\omega}', \mathbf{x}, \boldsymbol{\omega}) = S_d(\boldsymbol{\omega}')\, S_p(\|\mathbf{x}' - \mathbf{x}\|)\, S_d(\boldsymbol{\omega}), \tag{9}$$

where $S_d(\omega) = F_t(\theta)/\pi$ is directionally diffuse term modulated by the boundary's Fresnel transmission, and S_p is a radially symmetric

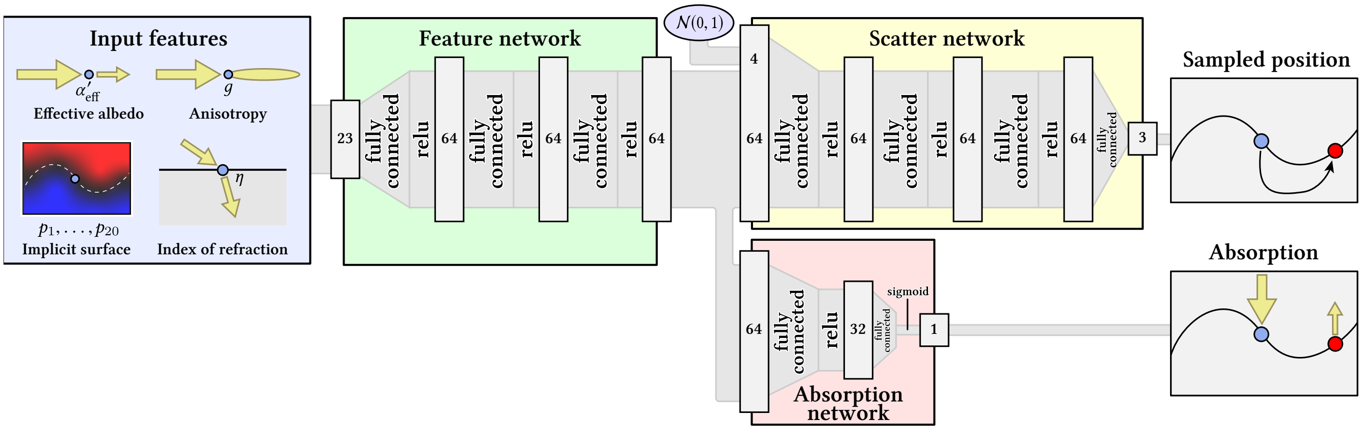


Fig. 2. Architecture of our learned BSSRDF at render time. The input consists of a set of features (blue) characterizing both material parameters and the local geometry of a shading point. These features are transformed by a deterministic feature network (green) shared by later parts of the model. A scatter network (yellow) concatenates the transformed features with a normally distributed 4D sample and outputs a 3D position sampled from the learned distribution. The absorption network also takes the transformed features as input and outputs a constant that scales the resulting distribution to support absorbing materials.

1D function or a more general 2D function [Frisvad et al. 2014] representing the solution of a plane-parallel light transport problem.

Our model generalizes this factorization to

$$S(\mathbf{x}', \boldsymbol{\omega}', \mathbf{x}, \boldsymbol{\omega}) = S_d(\boldsymbol{\omega}')\, S_p(\mathbf{x}', \mathbf{x}, \boldsymbol{\omega}) \tag{10}$$

where S_p has a complex learned dependence on all of $\mathbf{x}, \mathbf{x}', \boldsymbol{\omega}$, as well as the shape of the boundary $\partial\Omega$. The $\boldsymbol{\omega}'$ argument is handled as before—in practical terms, this means that our algorithm samples outgoing positions $\mathbf{x}'$ on the boundary, while the outgoing direction is handled using standard techniques (e.g. multiple importance sampling of $S_d(\boldsymbol{\omega}')$ and light sources). Section 6 discusses ways in which this last assumption could be removed in the future.

Architecture. Our BSSRDF model consists of three major conceptual units shown in Figure 2. Note that this illustration depicts the usage at render time, while training uses a different arrangement discussed shortly.

When the BSSRDF model is sampled at a particular incident location $\mathbf{x}$, we first extract a feature descriptor characterizing the local material and geometric properties. The features influence both the characteristic shape of the BSSRDF and its overall scale in absorbing materials, and these two aspects are handled in separate branches of our architecture. Part of the calculation that controls their dependence on the input features should arguably be shared by both branches, and our architecture reflects this: the features first pass through a dedicated *feature network* consisting of three fully connected layers with rectified linear units as activation functions, which output a 64-dimensional feature vector.

The *scatter network* takes the pre-processed feature vector and concatenates it with a four-dimensional normal variate to generate a 3D position from the learned distribution using three fully connected layers. Note that the scatter network is simply the decoder $\mathbf{g}(\mathbf{z})$ of our trained variational autoencoder, and the normal variate constitutes a sample from the latent distribution $p_{\mathbf{z}}(\mathbf{z})$. The jointly trained encoder $\mathbf{q}(\mathbf{x})$, which is separate from the feature network and not shown in Figure 2, is not needed at render time and can be discarded. We observed superior performance when using a latent space, whose dimension exceeds that of the target distribution (2D), hence we use a four-dimensional latent variables $\mathbf{z}$.

The scatter network is trained using samples generated by a volumetric path tracer, which uses standard techniques (e.g. importance sampling, Russian Roulette, etc.) to ensure that generated samples have uniform importance weights. This, however, means that one crucial piece of information is still missing: what fraction of incident light eventually leaves the surface without being absorbed along the way? This fraction is a function of the input position, geometry, and material properties. To support absorbing materials, we must therefore also regress a scale factor for the distribution $\mathbf{g}(\mathbf{z})$, which is the role of the *absorption network*. This part is independent of $\mathbf{z}$ and can hence branch off without access to the latent variable. We use two progressively smaller fully connected layers and a sigmoidal activation function that ensures an output in the range $[0, 1]$.

Projection. The use of simplified domains is a general source of difficulty in the context of BSSRDF importance sampling: the generated positions generally do not lie on the scene manifold $\mathcal{M}$, and this discrepancy must be reconciled before the BSSRDF is usable in physically based rendering systems.

For instance, many prior BSSRDF models assume that S factorizes into a product, whose spatial component is a radially symmetric function that only depends on a scalar radius $r = \|\mathbf{x}' - \mathbf{x}\|$ (Equation 9). Sampling S then involves generating a radius $r \sim S_p$ and azimuth $\phi \sim U(0, 2\pi)$, at which point the resulting polar coordinate must be projected onto underlying surface. A number techniques have been proposed to carry out such a projection using ray tracing [Jensen et al. 2001; King et al. 2013; Walter et al. 2012].

The central difficulty here is that the surface is not flat. In fact, the intersection with an ε-ball $\mathcal{M} \cap B(\varepsilon)$ may contain an arbitrarily small or even unbounded amount of surface area rather than the expected amount $\pi\varepsilon^2$. This discrepancy causes energy conservation issues in all plane-parallel BSSRDFs: depending on how projection is implemented, they produce conspicuous dark regions or energy

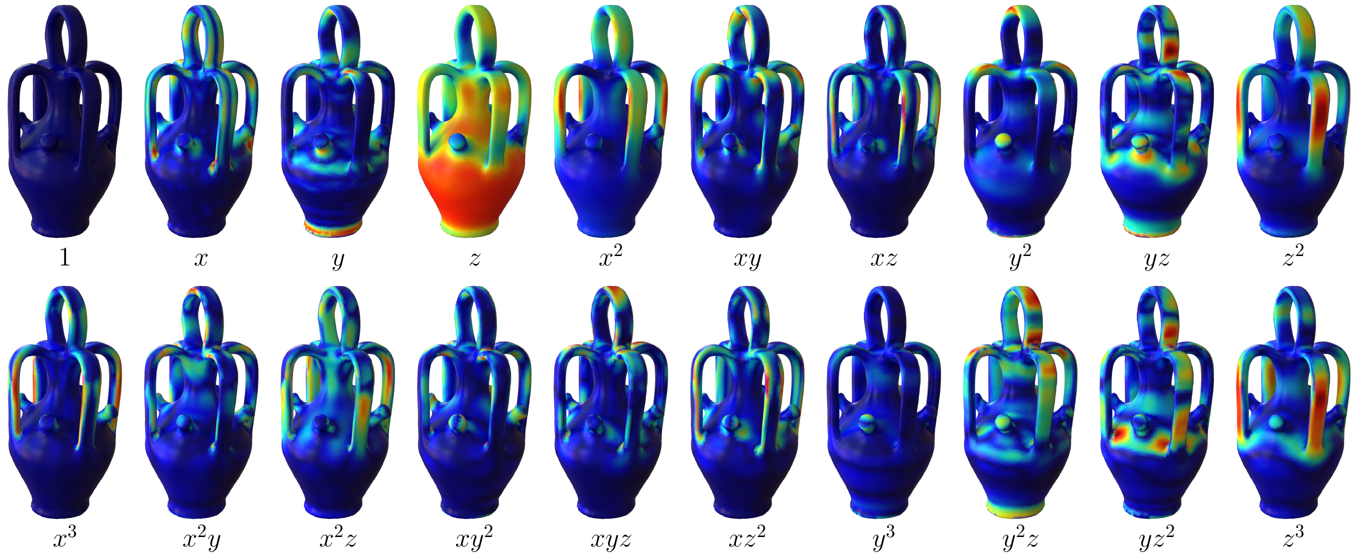


Fig. 3. Our shape descriptor approximates the local geometry using a low-order polynomial that encodes an implicit surface approximation. This figure visualizes polynomial coefficients in a tangential coordinate system computed using a moving least squares fit. Different parts of the descriptor identify features including components of surface normals, curvature, knobs, and handles.

gains that can even cause the rendering process to diverge (Section 5). This lack of energy conservation is one of the reasons why BSSRDF models have fallen out of favor in recent years, with major rendering systems falling back to brute-force sampling of the RTE [Christensen et al. 2018; Fascione et al. 2018].

Our approach is to model the surface geometry using a *trivariate* degree-3 polynomial $P(\mathbf{x}) = c_1 + c_2x + c_3y + \ldots c_{20}z^3$ that defines an implicit surface in a suitably chosen neighborhood of a shading point. This neighborhood may also encompass the opposing side of a surface, which is important when rendering thin regions of an object. The coefficients $c_1 \ldots c_{20}$ are part of the descriptor provided to the feature network (see Figure 3 for a visualization), and the network is trained to sample positions on the zero level set (i.e. $P(\mathbf{x}) = 0$) of this polynomial.

This does not completely solve the problem, however: although our network indeed learns to generate points $\mathbf{x}$ that are very close to the zero level set, they are not guaranteed to satisfy $P(\mathbf{x}) = 0$ exactly. Furthermore, the polynomial is only an approximation of the true scene geometry, hence a projection is still needed. The polynomial does, however, provide an excellent clue regarding where a nearby surface can be found—specifically in the direction of the scaled gradient $-P(\mathbf{x})\nabla P(\mathbf{x})$, where

$$\nabla P(\mathbf{x}) = \begin{bmatrix} c_2 & + & 2c_5x & + & \cdots \\ c_3 & + & 2c_8y & + & \cdots \\ c_4 & + & 2c_{10}z & + & \cdots \end{bmatrix}. \tag{11}$$

In practice, we trace rays in both directions $\pm\nabla P(\mathbf{x})$ and use the closer intersection if multiple are found. This scheme is guaranteed not to introduce additional energy, and the shape-adaptivity tends to resolve issues with energy loss encountered by prior work. We discuss several examples later in Section 5.

Material descriptor. The second input of the feature network is a small tuple of coefficients $\varphi_{\partial\Omega}$ characterizing the material properties of both boundary and interior. The parameterization of these inputs is in essence arbitrary but has a significant effect on the network's training and generalization performance. We seek a parameterization that is as perceptually uniform as possible, meaning that a small change $\varphi_{\partial\Omega} + \delta$ should lead to a predictable and proportional change in a certain aspect of the material appearance. Preferably, the parameter range should also be bounded to a known interval.

The RTE and boundary conditions discussed in Section 3 are dependent on the index of refraction η, extinction σ_t, scattering coefficient σ_s and scattering anisotropy g. Although in principle admissible, these parameters are a poor choice due to their non-uniformity and redundancy. For instance, σ_s and σ_t are both unbounded and specify collision rates for certain types of medium interactions per unit distance, controlling the overall density and albedo of the medium. However, the density of a medium is also related to the size of the object containing it: scaling the exterior geometry by λ is equivalent to scaling σ_s and σ_t by λ. To avoid this redundancy, we can instead use the scattering albedo defined in terms of their ratio $\alpha = \sigma_s \,/\, \sigma_t \in [0, 1]$.

The second problematic aspect is the dependence on anisotropy g, which controls two significant aspects of the appearance. The most apparent effect is that the medium becomes progressively less dense as $g \longrightarrow 1$ (Section 3) which we would rather like to encode separately using an appropriate re-scaling of the geometry. The second effect is a disproportionately higher increase of translucency around thin regions and silhouettes. To separate these two effects, we rely on first-order similarity theory [Wyman et al. 1989], which provides related *reduced* parameters

$$\sigma_s' = (1-g)\,\sigma_s, \quad \sigma_t' = \sigma_a + \sigma_s', \quad \alpha' = \sigma_s' \,/\, \sigma_t', \quad g' = 0 \tag{12}$$

of an isotropic medium that is similar in appearance and considerably less dense if the original medium was highly forward-scattering

($g \approx 1$). We use the reduced extinction σ_t' to scale the polynomial fitting kernel (see Equation 16) and retain the non-reduced anisotropy g to learn the effect of anisotropy on thin regions and silhouettes.

The last step concerns the albedo α' which indicates the fraction of incident illumination that remains after interacting with one of the particles that make up the reduced medium. Since light tends to scatter *many times* before leaving the medium, the relation between α' and the *effective albedo* α'_{eff} of the object when observed from the outside is highly nonlinear (Figure 4). We rely on a bijective mapping proposed by Pharr et al. [2016] that approximately relates these two quantities, providing a more perceptually uniform parameterization.

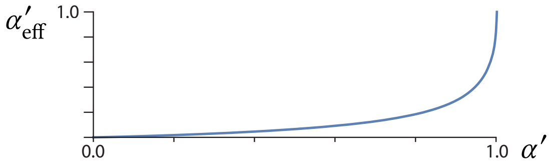


Fig. 4. The relation between reduced and effective albedo [Pharr et al. 2016].

$$\alpha'_{\text{eff}} = 1 - \frac{1}{8}\log(e^8 + \alpha'(1 - e^8)). \tag{13}$$

Our final set of material parameters then consists of the index of refraction η, the effective albedo α'_{eff}, and the non-reduced anisotropy g.

Neighborhood size. Our technique uses a polynomial fit (Figure 3) to approximate the geometry in a local neighborhood. The choice of this neighborhood is of crucial importance: an overly local fit will be sensitive to minute shape variations that have limited influence on the surrounding radiative transport, while a large neighborhood will lack adaptivity.

The size of this neighborhood is inversely proportional to the reduced extinction σ_t', which indicates the density of the medium. Media with a higher albedo or anisotropy should generally also require a larger neighborhood, but the influence of these parameters is less clear. Ideally, our choice should be related to the spread of samples $\mathbf{x}'$ drawn from a ground-truth BSSRDF $S_{\text{gt}}(\mathbf{x}', \boldsymbol{\omega}', \mathbf{x}, \boldsymbol{\omega})$ realized via brute-force Monte Carlo simulation. We performed a large set of such simulations in a plane-parallel configuration and computed the *median absolute deviation*

$$\text{MAD} = \text{median}(\|\mathbf{x}_k' - \mathbf{x}\|), \tag{14}$$

which provides a robust estimator of the spread of a distribution that is conceptually similar to variance. Here, $\mathbf{x}$ and $\mathbf{x}_k' (k = 1, \ldots, N)$ denote incident and outgoing positions, respectively. Based on the resulting data, we fit a linear interpolant in g, α' and α'_{eff}:

$$\text{MAD}(\alpha', \alpha'_{\text{eff}}, g) \approx \frac{1}{4}g + \frac{1}{4}\alpha' + \alpha'_{\text{eff}}. \tag{15}$$

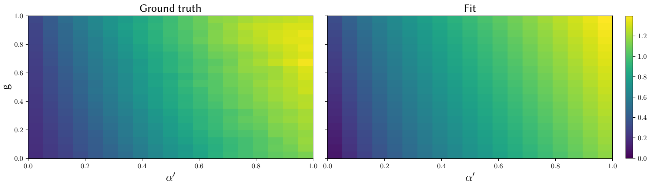


Fig. 5. **Left:** medium absolute deviation of the scattered distance as a function of anisotropy and reduced albedo. **Right:** our fit.

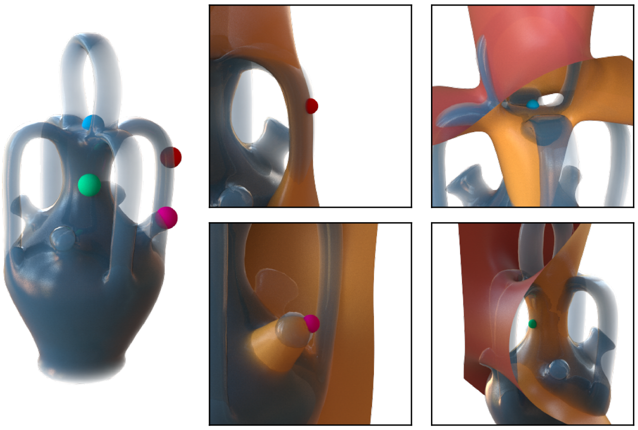


Fig. 6. Our shape descriptor consists of a trivariate degree-3 polynomial that approximates the input geometry in a local neighborhood. The four insets on the right visualize the zero level set of polynomials fit to the highlighted locations (exterior surface: brown, interior surface: red).

A visualization of the original data and our fit is shown in Figure 5. Using the above function, we define the standard deviation σ_n of a suitable neighborhood as

$$\sigma_n(\alpha', g, \sigma_t') = 2\,\text{MAD}(\alpha', g)\,/\,\sigma_t'. \tag{16}$$

We scale the geometry by the reciprocal of this standard deviation in order make our neural network invariant to changes in scene scale and medium density.

Polynomial shape descriptors. Our local polynomial approximation is inspired by the moving least-squares algorithm of Shen et al. [2004]. The main difference to their approach is our use of higher-degree polynomials and smooth weighting kernels[1].

Various challenges arise when fitting polynomials to the vertices of a boundary $\partial\Omega$ defined in terms of a highly non-uniform tessellation. To sidestep all such problems, we initially sample m uniformly distributed points $\mathbf{b}_1 \ldots \mathbf{b}_m \in \partial\Omega$ with associated surface normals $\mathbf{n}_1 \ldots \mathbf{n}_m$, where

$$m = \max\left\{1024, 2\,\sigma_n^{-2}\,A(\partial\Omega)\right\} \tag{17}$$

grows in proportion to the surface area $A(\partial\Omega)$ and neighborhood size. We use the resulting point cloud as an approximation of the original geometry and organize it in a k-d tree. Both sampled points and the acceleration data structure are only required for precomputation and can be discarded at render time.

To fit a polynomial to a neighborhood centered around a given position $\mathbf{x}$, we perform a weighted least-squares optimization that minimizes the following quadratic energy

$$\mathcal{E}(\mathbf{c}) = \frac{1}{m}\sum_{i=1}^{m} w(\|\mathbf{b}_i'\|)\left[P_{\mathbf{c}}(\mathbf{b}_i')^2 + \|\nabla P_{\mathbf{c}}(\mathbf{b}_i') - \mathbf{n}_i\|^2\right] + \mu\|\mathbf{c}\|^2 \tag{18}$$

where $\mathbf{b}_i' = \sigma_n^{-1}(\mathbf{b} - \mathbf{x})$ and $w(r) = e^{-r^2/2}$ defines a Gaussian window that preferentially weights constraints near $\mathbf{x}$. The first term inside the square brackets forces the polynomial to take on small values

[1] Shen et al. fit polynomials of order zero and use singular weighting kernels that must be integrated over triangles using an involved semi-analytic quadrature scheme.

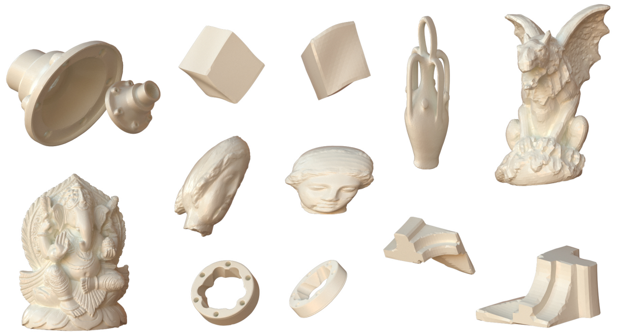


Fig. 7. Visualization of several meshes in our training dataset. Each shape is instantiated many times using randomly sampled transformations.

on the boundary $\partial\Omega$, and the second term constrains its gradients to point away from the boundary. The last term regularizes the problem by penalizing solutions with large coefficients (we use $\mu = 0.0001$). We found it beneficial to also add a small number of uniformly weighted point constraints as an additional regularizer. For this we sample 32 points uniformly over the mesh and scale their constraints by a constant of 0.01.

In essence, we search for a polynomial that behaves like a *signed distance function* in the neighborhood of the boundary. We cast Equation (18) into the associated normal equations and solve the resulting 20×20 linear system using a Cholesky decomposition. Figure 6 visualizes the polynomial surfaces fit to several neighborhoods.

Note that the computation produces a polynomial P that is centered around the shading point $\mathbf{x}$ and expressed in scaled coordinates where a distance of 1 unit corresponds to the neighborhood size σ_n in world space. To accelerate optimization of Equation (18), we clamp the weighting kernel w to zero after a distance of 3 standard deviations and use the previously constructed k-d tree to find points within this distance.

Adaptation to the incident direction. Thus far, we have not discussed how our model adapts to the incident direction ω. We do so by rotating the point cloud $(\mathbf{b}'_i, \mathbf{n}_i)$ into a coordinate system where the z-axis corresponds to the initial propagation direction of light. Our model learns to recognize this cue and uses it to produce appropriately stretched-out and shifted distributions at glancing angles.

To avoid potential inconsistencies during training and rendering, we augment Equation 18 with a hard constraint $c_1 = 0$, which ensures that a polynomial approximating the geometry near $\mathbf{x}$ indeed passes through this point.

Training dataset. We train our model on a reference dataset of paths generated using brute-force Monte Carlo simulation of transport through geometry retrieved from a repository of CAD models and 3D-scanned shapes of varying complexity [Jakob et al. 2015; Myles et al. 2014]. We furthermore augment the dataset by instantiating each shape a number of times using randomly sampled transformations, which increases the diversity of the dataset particularly with regards to the number of thin regions (Figure 7).

For each shape, we proceed to sample 100K incident positions and directions per shape using uniform and cosine-weighted distributions, respectively. Each sample is furthermore associated with an independently chosen index of refraction $\eta \sim U(1, 1.5)$, effective albedo $\alpha'_{\text{eff}} \sim U(0, 1)$ and anisotropy $g \sim U(0, 0.95)$. Once these parameters are decided, we are able to complete the feature vector with the computed coefficients $c_1 \ldots c_{20}$ of the local polynomial fit.

Polynomial path tracer. When simulating scattering directly on the ground truth geometry, we often encounter configurations where the polynomial shape descriptor cannot explain all light transport phenomena. To reduce the difficulty of the learning task[2], we do not train directly on the input meshes, but rather generate transport paths within the local polynomial surface approximations. This implies that our networks should in principle be able to replicate the training data from the provided features perfectly, in the sense that the training data contains no information that is not explained by the associated features. The role of the mesh dataset, then, is to provide a straightforward way of generating random sets of polynomials with the right statistics.

To simulate light transport in the polynomial representation, we must intersect rays $\mathbf{r}(t) = \mathbf{o} + t \cdot \mathbf{d}$ with the zero level set to find the solution $P(\mathbf{r}(t)) = 0$ with the smallest $t > 0$. Note that acceleration techniques for signed distance functions [Keinert et al. 2014] are not directly applicable—although the polynomial is optimized to resemble a signed distance function near the boundary, our fit makes no guarantees regarding its global behavior. We use a simplistic root finding scheme that involves ray marching with a fixed step size until the sign $P(\mathbf{r}(t))$ changes, which is followed by Newton-Bisection to locate the root given a bracketing interval. The step size must be set appropriately to match the minimum size of features in the shape dataset. Although more efficient methods are conceivable, optimization of this step is of limited value since the polynomial ray tracer only runs once to generate the training dataset. Generation of the full training dataset requires around 30 minutes.

The volumetric path tracer uses Russian roulette [Pharr et al. 2016] at each iteration to guarantee that the final set of path vertices has uniform weights. We exclude light paths where the light simply passes through the object without interacting with the underlying medium, since these paths are more easily handled separately at render time. We refract incident directions into the material using Snell's law. Internal reflections do not generate training samples—we only record paths after sampling a refraction towards the exterior.

Loss function. Training minimizes a joint loss function consisting of three components:

$$\begin{aligned} \mathcal{L}_{VAE}(\mathbf{x}_i) &= D[p_{\mathbf{q}}(\mathbf{z} \,|\, \mathbf{x}_i) \;\|\; p_{\mathbf{z}}(\mathbf{z})] \\ &\quad + w_0 \|\mathbf{x}_i - \mathbf{g}(\mathbf{q}(\mathbf{x}_i))\|_H \\ &\quad + w_1 \|\mathbf{a}_i - \mathbf{a}(\mathbf{a}_i)\|^2 \end{aligned} \tag{19}$$

where g is the decoder, q is the encoder, $\mathbf{z}$ are the latent variables, and $p_{\mathbf{q}}$ and $p_{\mathbf{z}}$ denote associated densities compared using a Kullback-Leibler distance D. The first part of Equation 19 is the VAE's latent

[2] We initially trained with transport on the original geometry and found that the resulting unexplained "noise" acted as a significant regularizer that became an impediment to the overall quality (in essence, the network learned to ignore the input features).

variable term, which constrains the latent space to follow a standard normal distribution. Together with the second term, they represent the maximum likelihood estimator introduced as part of the discussion of the variational autoencoder in Section 3. We use a Huber loss $\|\cdot\|_H$ instead of the regular L_2 loss to make training more robust. The third term constrains our absorption network to match ground-truth estimates of absorption probabilities, which are computed by an estimator that traces 1024 light paths at the same incident location. We use $w_0 = 100$ and $w_1 = 5000$ (the absorption takes on relatively small values and requires a higher weight to ensure accuracy.)

Rendering. Usage of our BSSRDF in a physically based rendering system requires efficient access to local shape descriptors. We precompute this information ahead of time by fitting polynomials to local neighborhoods around each vertex of the mesh. We retain the resulting coefficients and bilinearly interpolate them during ray-triangle intersection.

Note that the previously discussed fit depends on the direction of incidence, which would ordinarily prevent this type of precomputation. Fortunately, we can simply perform the fit in world space and suitably rotate the polynomial at render time. To do so, we substitute the rotation matrix into the polynomial and expand the resulting expression.

Sampling the BSSRDF produces a single position $\mathbf{x}' \in \partial\Omega$ drawn from the learned distribution for a particular set of material features. We then apply multiple importance sampling of the term $S_d(\boldsymbol{\omega}')$ and the scene's emitters to estimate the integral over the final directional dimension of Equation 10.

The sampling scheme discussed thus far assumes that the material is monochromatic. To support colored materials, we use splitting to estimate the integral with the BSSRDF for each color channel. To avoid exponential growth of the resulting computation graph, we only do this first time the BSSRDF is encountered along a path, after which we randomly sample a single channel. A simpler approximate alternative would be to assume that the material is monochromatic at this point. Note that despite the added expense of splitting at the first interaction, our method achieves competitive performance due to the uniform weights of generated samples.

Similar to prior BSSRDF models [D'Eon and Irving 2011], our method supports textured spatial variation if the material descriptor is controlled by a custom shader that is evaluated at the shading point $\mathbf{x}$. From a physical viewpoint, such approaches are of course dubious: the BSSRDF assumes homogeneity, and this type of query is furthermore non-reciprocal. Nonetheless, such approximations do not produce visible artifacts and are commonly used, e.g., to render human faces like the one shown in Figure 9.

5 RESULTS

We implemented our model as a new `Subsurface` plugin in the Mitsuba renderer [Jakob 2010] and generated most figures (including timing comparisons in Figure 11) using Mitsuba's standard path tracer on cluster nodes with Intel Xeon 6132 processors and 128 GB of memory. Note that only a small portion of this memory was actually used—our model adds an overhead on the order of 80 bytes per vertex for polynomial coefficients, and storage of neural network weights (~ 153 KiB) is negligible.

We generate training data as part of an offline process using a polynomial path tracer implemented in Mitsuba. One helpful advantage of learning from a physical simulation is the relative ease of generating very large reference datasets: we sample 16 million complete transport paths annotated with material parameters for training and 4 million test samples for inspecting training convergence (9 GiB on disk). The training data does not contain any of the shapes that we use to demonstrate rendering of subsurface transport in this article.

Due to our relatively shallow architecture and lack of convolutional layers, we found that training using GPU acceleration yielded little to no speedup, hence we performed all training on the CPU. We jointly train all three networks with respect to the loss defined in Equation 19, using 3.5 million batches of size 32, which takes roughly 4 hours on a Intel Xeon node using 12 cores. Our training implementation is based on Tensorflow [Abadi et al. 2015] and the Adam [Kingma and Ba 2014] optimizer with a learning rate of $2 \cdot 10^{-4}$. After training concludes, we write the final set of coefficients to disk for later use at render time.

When deployed in a physically based renderer, our model must be evaluated millions of times per second, hence we require an efficient implementation of the underlying arithmetic consisting mainly of fixed-size matrix-vector multiplications (Figure 2). We initially tried to export the Tensorflow computation graph to a native C++ implementation using Tensorflow's *accelerated linear algebra* (XLA) just-in-time compiler but only obtained modest speedups relative to the original Tensorflow graph. Following this, we manually implemented the network in C++ using Eigen [Guennebaud et al. 2010], which resulted in a speedup on the order of 200×.

We now evaluate the performance and accuracy of our model compared to ground truth references and prior work, in particular the *Forward Scattering Dipole* [Frederickx and Dutré 2017], the *Directional Dipole* [Frisvad et al. 2014] (both using sampling-based implementations provided by Frederickx et al.) and *Photon Beam Diffusion* [Habel et al. 2013] (using the implementation from PBRT [Pharr et al. 2016], which we ported to Mitsuba to facilitate timing comparisons). Note that PBRT's photon beam BSSRDF uses ray tracing-based projection [King et al. 2013] to map samples onto the surface.

Soap blocks. Figure 1 shows several anisotropic soap blocks rendered using photon beam diffusion and our method. Compared to the relatively flat-looking rendering produced by the plane-parallel dipole model, our shape-adaptive model increases scattering through thin silhouette regions and improves contrast in regions with geometric detail, producing a more realistic appearance and lower bias compared to a path-traced reference (please see the supplement for detailed comparisons including other methods). At equal render time (including preprocessing), our model produces 311 samples compared to 916 samples generated using photon beam diffusion. Since our model yields uniform importance sampling weights, the noise level in the rendered image is similar for both methods, despite the difference in the number of samples.

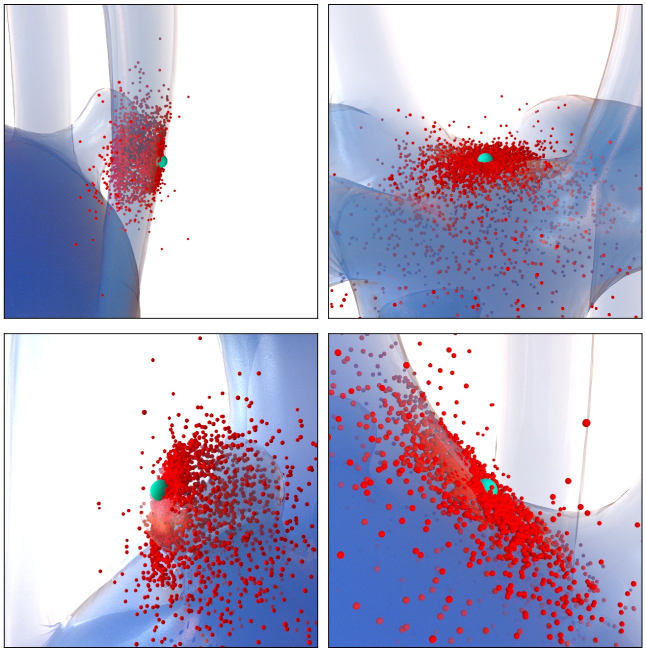
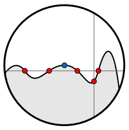
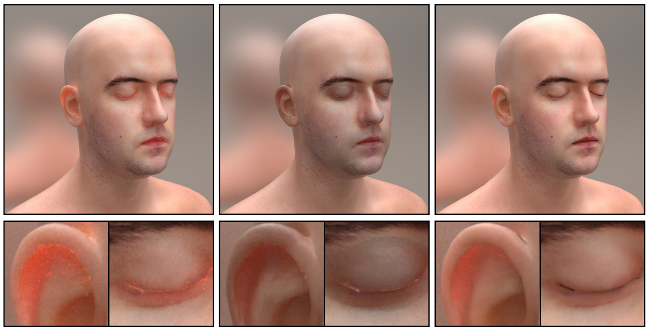


Fig. 8. Our BSSRDF model learns to sample point distributions that adapt to the underlying geometry. The visualization shows sampled outgoing positions (red) generated by our neural network architecture given incident locations (cyan). Note that the red points correspond to the "raw" network output without the additional projection that would be applied in actual usage.

Shape-adaptive distributions. Figure 8 visualizes several "raw" point distributions produced on the highly curved `botijo` shape that contains both knob-shaped extrusions and toroidal regions (handles). The figure shows that our BSSRDF model indeed learns to exploit the information contained in the local polynomial descriptor to produce point sets that closely align with the underlying polynomial (and hence, the input geometry). In actual usage, the points are furthermore projected onto the underlying triangle mesh. We do so by tracing rays along the gradient direction of the polynomial, which leads to a particularly robust projection operation.

Energy conservation. Figure 9 highlights energy conservation issues in standard BSSRDF techniques that map samples from a plane-parallel transport model onto a curved surface. This involves integrating incident radiance over an ε-ball centered around the shading point (blue), whose radius is related to the size of the diffusion kernel, typically using a set of orthogonal projections [King et al. 2013]. Since the surface is not flat, the projection may "find" an arbitrarily large or small amount of surface area. Without additional precautions, this introduces unwanted energy into the simulation that can even cause the rendering process to diverge (Figure 9a). The projection may also be constrained to collect illumination from at most a single surface intersection per projection, which simply turns excessively bright regions into conspicuous dark patches (Figure 9b). In contrast, our model generates points that are already very close to the surface due to the underlying polynomial model, whose gradient enables a particularly robust projection operation. Our projection never introduces additional energy and significantly reduces issues with energy loss encountered by plane-parallel transport models (Figure 9c).





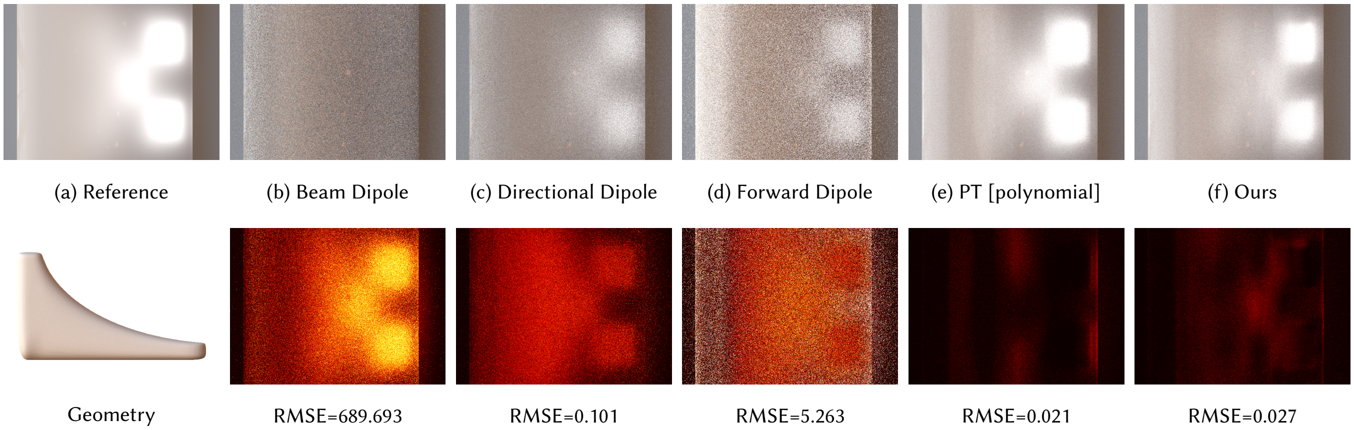
(a) Beam Dipole (variant 1) **(b)** Beam Dipole (variant 2) **(c)** Ours

Fig. 9. Energy conservation. Modern diffusion dipole models rely on ray tracing to project sampled 2D positions onto the underlying surface. A fundamental challenge of this projection is that the intersection of an ε-ball with curved scene geometry may contain an arbitrarily small or large amount of surface area rather than the expected amount $\pi\varepsilon^2$. Depending on how the resulting models are normalized, they either suffer from unrealistic energy gains **(a)** or losses **(b)**. Both sample generation and projection of our model **(c)** are guided by a local approximation of the underlying shape. Our model never creates energy, and the shape-adaptivity significantly reduces energy loss in practice.

Forward-scattering medium. Figure 10 shows a checkerboard light source viewed through a tapered dielectric material containing a highly forward-peaked medium. The checkerboard pattern becomes increasingly attenuated and blurry towards the thicker end due to multiple anisotropic scattering. The directional dipole, the forward dipole, and our method are able to reproduce this effect, while a classical dipole produces an opaque surface appearance. Interestingly, small differences between our model and the reference can also be found in the polynomial reference.

Comparison matrix. Figure 11 shows a large matrix of renderings of varied shapes made with different methods and material parameters: from top to bottom, we show results produced using a volumetric path tracer ("*Path Tracing*"), a volumetric path tracer with diffuse angular response at the outgoing position ("*PT. [diffuse]*") matching our factorization in Equation 10. The next row additionally replaces the geometry using a local polynomial model after refracting into the object, and hence corresponds closest to our learned model ("*PT. [poly]*"). The fourth row is our model, and the next three are photon beam diffusion, the directional dipole, and the forward scattering dipole.

As in the teaser scene, the beam diffusion model produces a relatively flat appearance (see e.g. the crevices of the wavy shape or the bunny in the two first columns). A more striking difference is the energy loss in columns 3, 4, and 5. The diffusion radius is relatively large compared to the geometry given these material parameters, causing the convolution with the diffusion kernel to find insufficient


(a) Reference (b) Beam Dipole (c) Directional Dipole (d) Forward Dipole (e) PT [polynomial] (f) Ours

Geometry RMSE=689.693 RMSE=0.101 RMSE=5.263 RMSE=0.021 RMSE=0.027

Fig. 10. Rendering of a forward scattering medium (g = .95, α = .99999) enclosed in a tapered shape, illuminated by a checkerboard light source. The directional dipole **(c)**, the forward dipole **(d)** and our learned BSSRDF **(f)** account for the shape-dependent decay and directional blur. In our case, the rendering reproduces the appearance of the polynomial path tracer **(e)**, which is a good approximation of the ground-truth transport **(a)**. The rendering by photon beam diffusion [Habel et al. 2013] **(b)** is almost opaque. Images **(b)**–**(f)** were rendered using 256 samples per pixel. For each technique we visualize the relative L_2 error compared to the converged ground-truth rendering **(a)**.

surface area. This is particularly apparent in the progression from larger to smaller cubes, cylinders, and spheres.

The forward dipole model produces excellent results for many configurations—among the dipole methods we compare to, it is the only BSSRDF model with a non-separable dependence on both incident and outgoing direction. A fundamental limitation of the model appears to be its importance sampling technique. At 1024 samples per pixel, it produces renderings with significant variance exceeding that of the Monte Carlo reference, using the highest amount of computation on average.

We observe excellent agreement between the learned BSSRDF and the original path-traced reference. As before, small deviations between our model and the reference can typically also be found in the polynomial reference. This seems to indicate that future work on local shape descriptors could further improve the quality of learned BSSRDF representations.

Light transport through slabs. Figures 12 and 13 visualize light transport through slabs of varying thickness and anisotropy. In Figure 12, all illumination arrives perpendicularly from the top, while Figure 13 also shows the dependence on the angle of incidence. In each case, we plot the distribution of emitted radiance at the top and bottom surfaces, and the left side shows a ground-truth reference produced by a volumetric path tracer. The histograms demonstrate that our model has learned to predict the spread, offset, slant, and intensity of the distribution of scattered light on both surfaces. We remark that our model was not specifically trained on parallel slabs that were used to generate this figure. One visible difference is that the distributions produced by our model are often less regular than the reference solution. We only noticed this in visualizations of the raw profiles, and the difference seems to have little practical relevance on usage in a physically based renderer.

6 CONCLUSION

We introduced a technique for learning an adaptive and scene-independent BSSRDF model from ground truth volumetric transport simulations. Our method avoids many limiting assumptions of prior analytic models regarding the planarity of surfaces and isotropy of volumetric transport, and it significantly generalizes the built-in notion of spatio-directional separability.

Our new algorithm does still share certain limitations with prior models—for instance, it assumes that the dependence in the outgoing direction is essentially diffuse and separable from the remainder of the model. This is certainly not true in thin anisotropic materials, where scattered radiation retains a narrow directional profile. While our training learns from a reciprocal light transport simulation, the network architecture is not built specifically to enforce reciprocity of the resulting model. We generally observed that errors produced by our method were also found in reference images produced by a path tracer using the same polynomial approximation, hence descriptors are a promising direction for further quality improvements. Another limitation is that our model implements a point sampling operation but does not provide a natural way of evaluating the associated density function. This is fine for usage in unidirectional rendering algorithms where knowing the sampling weight suffices, but may prove to be an impediment in other settings. Interestingly, these limitations are related: to use a fully non-separable BSSRDF in a physically-based renderer, a density function is a strict requirement to compute the material response given sampled positions on light sources. We believe that both limitations may be addressed in the future using an architecture based on invertible warps, such as the Real NVP [Dinh et al. 2016] method that was recently extended by Müller et al. [2018] and Zheng and Zwicker [2018].

Overall, we believe that our work makes a convincing case that a probabilistic generative model can be trained in a scene-independent way to replace a costly sampling process within a larger Monte Carlo simulation.

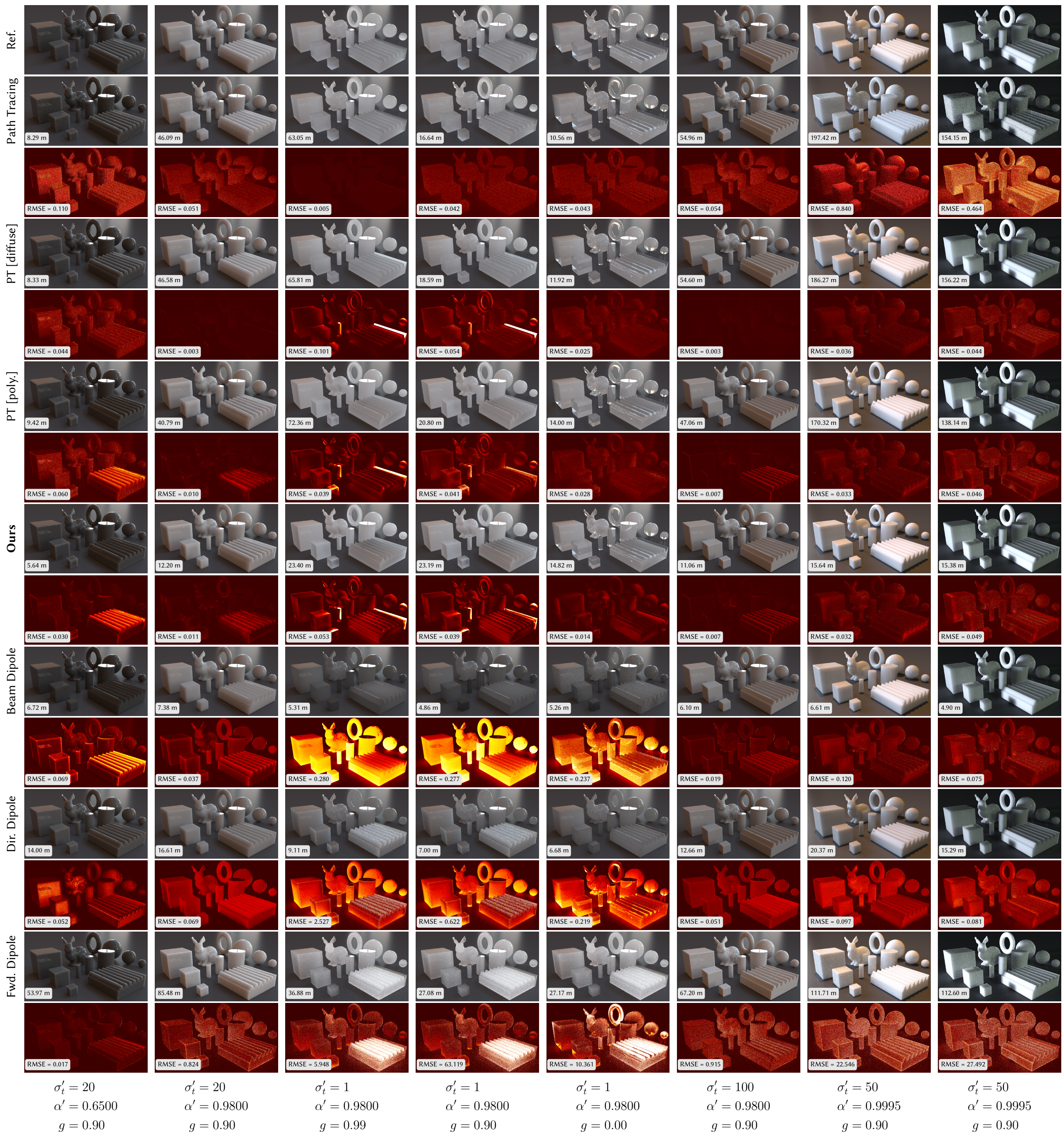


Fig. 11. Assorted shapes rendered using our model and other techniques for subsurface scattering, covering a range of material parameters (first 6 columns) and lighting conditions (last 2 columns). Note that an interactive version of this figure is provided in the supplemental material. *Ref.* is a path traced reference with 16-64K samples/pixel; the remaining rows all use 1024 samples/pixel and are followed by relative L_2 error maps. Insets specify rendering times and RMSE error values. *Path Tracing* denotes a standard volumetric path tracer, *PT. [diffuse]* is a path tracer that diffusely scatters light entering the object, *PT. [poly.]* is a path tracer that additionally uses the same polynomial surface approximations as our neural network. *Beam Dipole* [Habel et al. 2013], *Dir. Dipole* [Frisvad et al. 2014] and *Fwd. Dipole* [Frederickx and Dutré 2017] are previous state-of-the-art dipole methods.

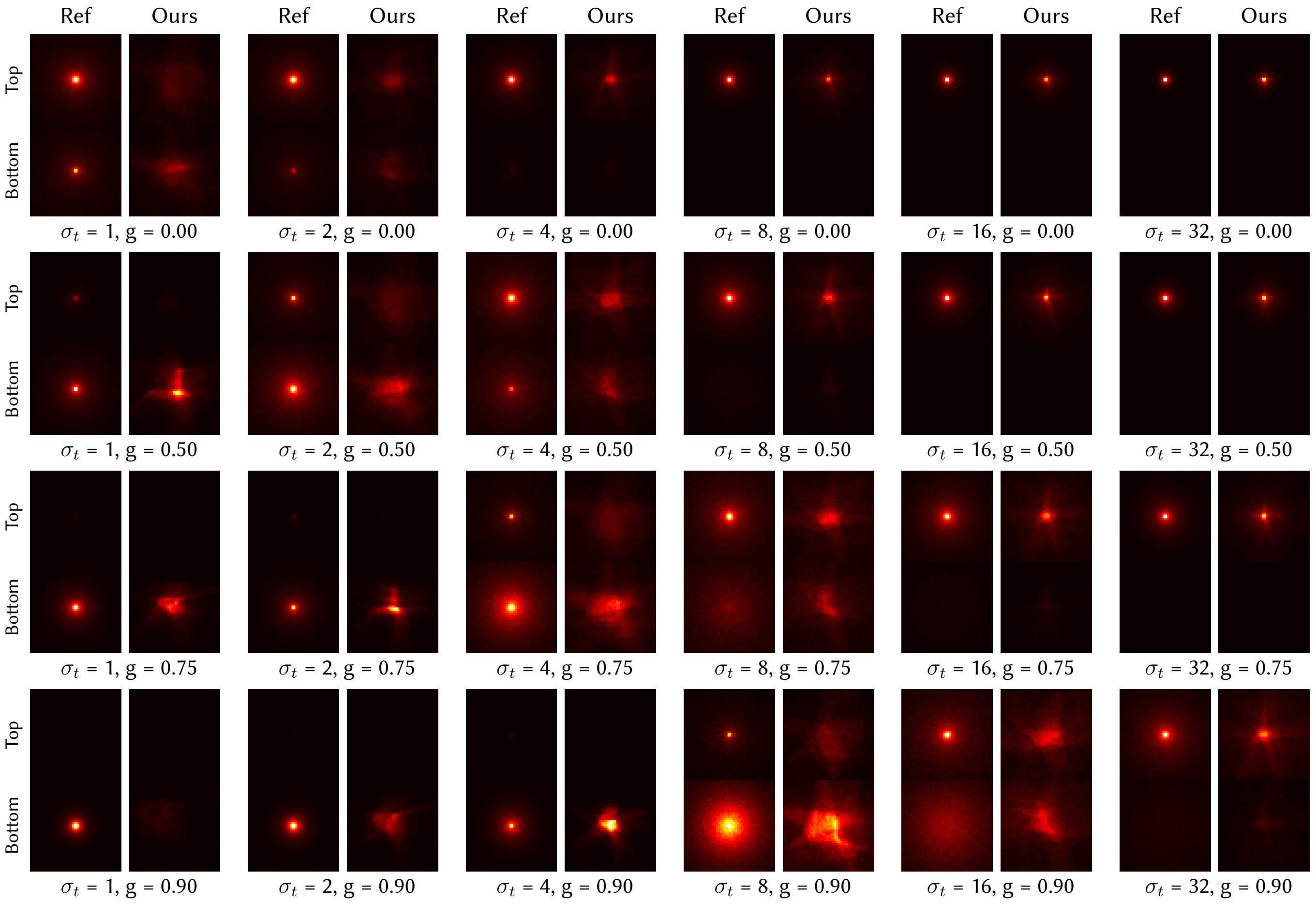


Fig. 12. Visualization of the distribution of scattered radiance on slabs of varying optical thickness illuminated by a perpendicularly incident collimated beam. We compare our method to ground-truth volumetric path tracing for different material configurations with a constant scattering albedo of $\alpha = 0.99$. Despite not having been trained on parallel slabs, our model correctly predicts the spread, offset, and intensity of the distribution on both surfaces.

ACKNOWLEDGMENTS

We thank Olesya Jakob for modeling the scene shown in Figure 1, Merlin Nimier-David for creating Figure 7 and providing feedback about the paper, and Thijs Vogels for additional proofreading. This research was supported by the Swiss National Science Foundation (SNSF) as part of grant 200021_184629.

REFERENCES

Martin Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Jia Yangqing, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mane, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viegas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. https://www.tensorflow.org/

Adam Arbree, Bruce Walter, and Kavita Bala. 2011. Heterogeneous Subsurface Scattering Using the Finite Element Method. *IEEE Transactions on Visualization and Computer Graphics* 17, 7 (July 2011), 956–969. https://doi.org/10.1109/TVCG.2010.117

Steve Bako, Thijs Vogels, Brian McWilliams, Mark Meyer, Jan Novák, Alex Harvill, Pradeep Sen, Tony DeRose, and Fabrice Rousselle. 2017. Kernel-predicting Convolutional Networks for Denoising Monte Carlo Renderings. *ACM Trans. Graph. (Proc. SIGGRAPH)* 36, 4 (jul 2017), 97:1—-97:14. https://doi.org/10.1145/3072959.3073708

Chakravarty R Alla Chaitanya, Anton S Kaplanyan, Christoph Schied, Marco Salvi, Aaron Lefohn, Derek Nowrouzezahrai, and Timo Aila. 2017. Interactive Reconstruction of Monte Carlo Image Sequences Using a Recurrent Denoising Autoencoder. *ACM Trans. Graph.* 36, 4 (jul 2017), 98:1—-98:12. https://doi.org/10.1145/3072959.3073601

Subrahmanyan Chandrasekhar. 1960. *Radiative transfer.* Dover publications, New York.

Per Christensen, Julian Fong, Jonathan Shade, Wayne Wooten, Brenden Schubert, Andrew Kensler, Stephen Friedman, Charlie Kilpatrick, Cliff Ramshaw, Marc Bannister, Brenton Rayner, Jonathan Brouillat, and Max Liani. 2018. RenderMan: An Advanced Path-Tracing Architecture for Movie Rendering. *ACM Trans. Graph.* 37, 3, Article 30 (Aug. 2018), 21 pages. https://doi.org/10.1145/3182162

Eugene d'Eon. 2012. A better dipole. http://www.eugenedeon.com/project/a-better-dipole/

Eugene D'Eon and Geoffrey Irving. 2011. A Quantized-diffusion Model for Rendering Translucent Materials. *ACM Trans. Graph. (Proc. SIGGRAPH 2011)* 30, 4 (jul 2011), 56:1–56:14. https://doi.org/10.1145/2010324.1964951

Eugene d'Eon, David Luebke, and Eric Enderton. 2007. Efficient Rendering of Human Skin. In *Proceedings of the 18th Eurographics Conference on Rendering Techniques (EGSR'07).* Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 147–157. https://doi.org/10.2312/EGWR/EGSR07/147-157

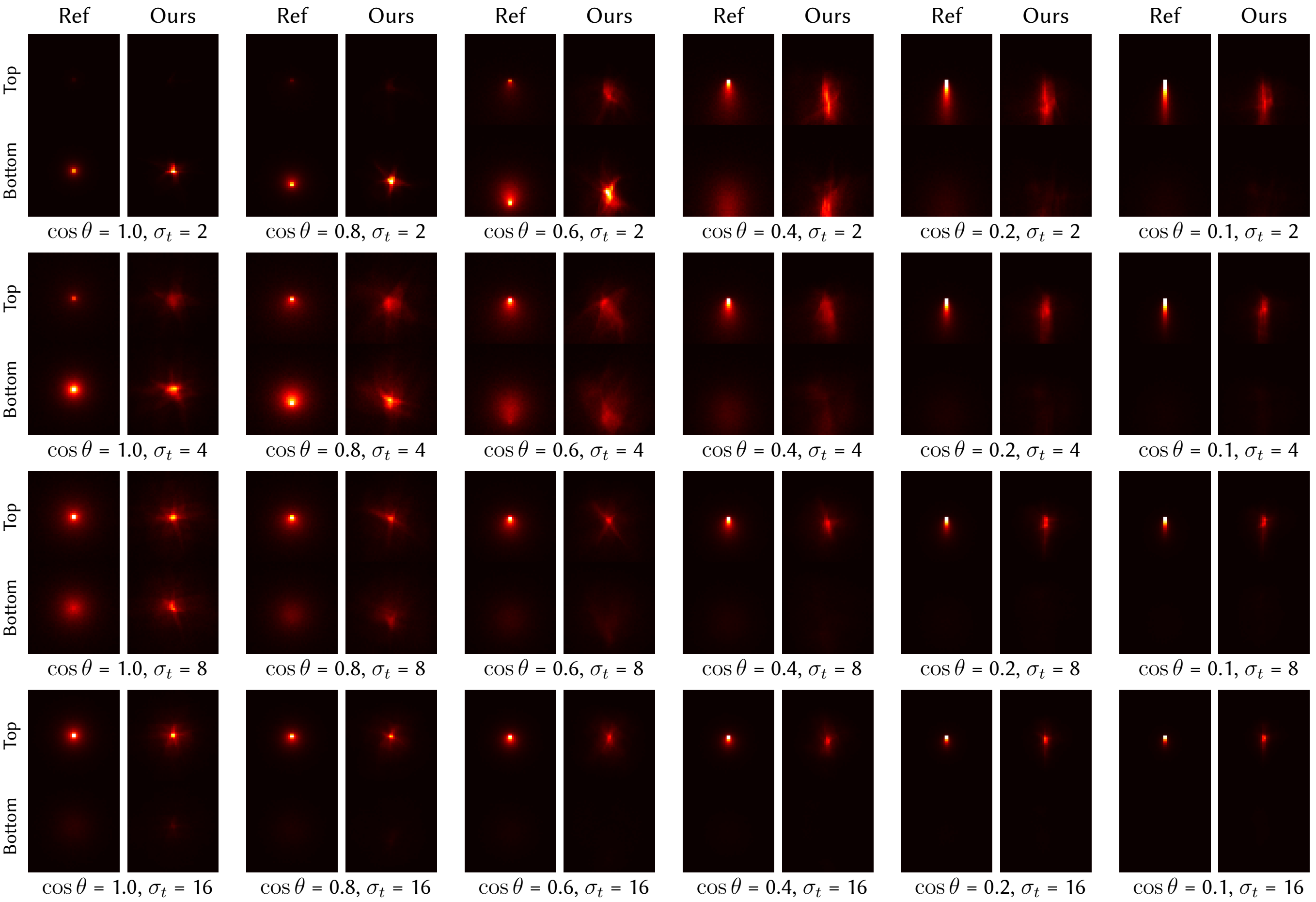

Ref
Ours
Top
Bottom
$\cos\theta = 1.0, \sigma_t = 2$
$\cos\theta = 0.8, \sigma_t = 2$
$\cos\theta = 0.6, \sigma_t = 2$
$\cos\theta = 0.4, \sigma_t = 2$
$\cos\theta = 0.2, \sigma_t = 2$
$\cos\theta = 0.1, \sigma_t = 2$
$\cos\theta = 1.0, \sigma_t = 4$
$\cos\theta = 0.8, \sigma_t = 4$
$\cos\theta = 0.6, \sigma_t = 4$
$\cos\theta = 0.4, \sigma_t = 4$
$\cos\theta = 0.2, \sigma_t = 4$
$\cos\theta = 0.1, \sigma_t = 4$
$\cos\theta = 1.0, \sigma_t = 8$
$\cos\theta = 0.8, \sigma_t = 8$
$\cos\theta = 0.6, \sigma_t = 8$
$\cos\theta = 0.4, \sigma_t = 8$
$\cos\theta = 0.2, \sigma_t = 8$
$\cos\theta = 0.1, \sigma_t = 8$
$\cos\theta = 1.0, \sigma_t = 16$
$\cos\theta = 0.8, \sigma_t = 16$
$\cos\theta = 0.6, \sigma_t = 16$
$\cos\theta = 0.4, \sigma_t = 16$
$\cos\theta = 0.2, \sigma_t = 16$
$\cos\theta = 0.1, \sigma_t = 16$

Fig. 13. Visualization of the distribution of scattered radiance on slabs of varying optical thickness illuminated by collimated beam with a varying angle of incidence. All insets use a fixed scattering albedo of $\alpha = 0.99$ and anisotropy of $g = 0.75$. Despite not having been trained on parallel slabs, our model learns to generate slanted profiles at the top surface, and it accounts for the spread and offset of the distribution at the bottom surface.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. 2016. Density estimation using Real NVP. arXiv:cs.LG/1605.08803

Carl Doersch. 2016. Tutorial on Variational Autoencoders. stat/1606 (2016), 23. arXiv:1412.6980 https://arxiv.org/abs/1606.05908

Craig Donner and Henrik Wann Jensen. 2005. Light Diffusion in Multi-layered Translucent Materials. In *ACM Trans. Graph. (Proc. SIGGRAPH) (SIGGRAPH '05)*. ACM, New York, NY, USA, 1032–1039. https://doi.org/10.1145/1186822.1073308

Craig Donner, Jason Lawrence, Ravi Ramamoorthi, Toshiya Hachisuka, Henrik Wann Jensen, and Shree Nayar. 2009. An Empirical BSSRDF Model. *ACM Trans. Graph.* 28, 3, Article 30 (July 2009), 10 pages. https://doi.org/10.1145/1531326.1531336

Luca Fascione, Johannes Hanika, Mark Leone, Marc Droske, Jorge Schwarzhaupt, Tomáš Davidovič, Andrea Weidlich, and Johannes Meng. 2018. Manuka: A Batch-Shading Architecture for Spectral Path Tracing in Movie Production. *ACM Trans. Graph.* 37, 3, Article 31 (Aug. 2018), 18 pages. https://doi.org/10.1145/3182161

Roald Frederickx and Philip Dutré. 2017. A Forward Scattering Dipole Model from a Functional Integral Approximation. *ACM Trans. Graph.* 36, 4, Article 109 (July 2017), 13 pages. https://doi.org/10.1145/3072959.3073681

Jeppe Revall Frisvad, Toshiya Hachisuka, and Thomas Kim Kjeldsen. 2014. Directional Dipole Model for Subsurface Scattering. *ACM Trans. Graph.* 34, 1, Article 5 (Dec. 2014), 12 pages. https://doi.org/10.1145/2682629

Gaël Guennebaud, Benoît Jacob, et al. 2010. Eigen v3. http://eigen.tuxfamily.org.

Ralf Habel, Per H Christensen, and Wojciech Jarosz. 2013. Photon Beam Diffusion: A Hybrid Monte Carlo Method for Subsurface Scattering. *Computer Graphics Forum (Proceedings of EGSR)* 32, 4 (jun 2013), 27–37. https://doi.org/10.1111/cgf.12148

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

Louis G. Henyey and Jesse L. Greenstein. 1941. Diffuse radiation in the galaxy. *The Astrophysical Journal* 93 (1941), 70–83.

Pedro Hermosilla, Sebastian Maisch, Tobias Ritschel, and Timo Ropinski. 2018. Deep-learning the Latent Space of Light Transport. arXiv:cs.GR/1811.04756

Akira Ishimaru. 1999. *Wave propagation and scattering in random media*. Vol. 12. John Wiley & Sons.

Wenzel Jakob. 2010. Mitsuba renderer.

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. (Proc. SIGGRAPH)* 29, 4 (2010), 53:1–53:13. https://doi.org/10.1145/1778765.1778790

Wenzel Jakob, Marco Tarini, Daniele Panozzo, and Olga Sorkine-Hornung. 2015. Instant Field-Aligned Meshes. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)* 34, 6 (Nov. 2015), 189:1–189:15. https://doi.org/10.1145/2816795.2818078

Henrik Wann Jensen and Juan Buhler. 2002. A Rapid Hierarchical Rendering Technique for Translucent Materials. *ACM Trans. Graph.* 21, 3 (July 2002), 576–581. https://doi.org/10.1145/566654.566619

Henrik Wann Jensen and Per H. Christensen. 1998. Efficient Simulation of Light Transport in Scenes with Participating Media Using Photon Maps. (1998), 311–320. https://doi.org/10.1145/280814.280925

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 (SIGGRAPH '01)*. ACM, New York, NY, USA, 511–518. https://doi.org/10.1145/383259.383319

Jorge Jimenez, Veronica Sundstedt, and Diego Gutierrez. 2009. Screen-space perceptual rendering of human skin. *ACM Transactions on Applied Perception* 6, 4, Article 23 (2009), 15 pages.

James T. Kajiya. 1986. The Rendering Equation. In *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '86)*. ACM, New York, NY, USA, 143–150. https://doi.org/10.1145/15922.15902

James T. Kajiya and Brian P. Von Herzen. 1984. Ray Tracing Volume Densities. In *Proceedings of the 11th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '84)*. ACM, New York, NY, USA, 165–174. https://doi.org/10.1145/800031.808594

Simon Kallweit, Thomas Müller, Brian McWilliams, Markus Gross, and Jan Novák. 2017. Deep Scattering: Rendering Atmospheric Clouds with Radiance-Predicting Neural Networks. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* 36, 6 (nov 2017), 231:1–231:11. https://doi.org/10.1145/3130800.3130880

Benjamin Keinert, Henry Schäfer, Johann Korndörfer, Urs Ganse, and Marc Stamminger. 2014. Enhanced Sphere Tracing. In *Smart Tools and Apps for Graphics - Eurographics Italian Chapter Conference*, Andrea Giachetti (Ed.). The Eurographics Association. https://doi.org/10.2312/stag.20141233

Alan King, Christopher Kulla, Alejandro Conty, and Marcos Fajardo. 2013. BSSRDF Importance Sampling. In *ACM SIGGRAPH 2013 Talks (SIGGRAPH '13)*. ACM, New York, NY, USA, Article 48, 1 pages. https://doi.org/10.1145/2504459.2504520

Diederik P Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *CoRR* abs/1412.6 (2014), 15. arXiv:1412.6980 http://arxiv.org/abs/1412.6980

Diederik P. Kingma and Max Welling. 2013. Auto-Encoding Variational Bayes. *CoRR* abs/1312.6114 (2013), 14.

Konstantin Kolchin. 2010. Surface Curvature Effects on Reflectance from Translucent Materials. *CoRR* abs/1010.2623 (2010), 4. arXiv:1010.2623 http://arxiv.org/abs/1010.2623

Jaroslav Křivánek and Eugene d'Eon. 2014. A Zero-variance-based Sampling Scheme for Monte Carlo Subsurface Scattering. In *ACM SIGGRAPH 2014 Talks (SIGGRAPH '14)*. ACM, New York, NY, USA, Article 66, 1 pages. https://doi.org/10.1145/2614106.2614138

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature* 521 (2015), 436–444.

Johannes Meng, Johannes Hanika, and Carsten Dachsbacher. 2016. Improving the Dwivedi Sampling Scheme. *Computer Graphics Forum (Proceedings of Eurographics Symposium on Rendering)* 35, 4 (2016), 37–44.

Michael I Mishchenko, Larry D Travis, and Andrew A Lacis. 2006. *Multiple scattering of light by particles: radiative transfer and coherent backscattering*. Cambridge University Press.

Ashish Myles, Nico Pietroni, and Denis Zorin. 2014. Robust Field-aligned Global Parametrization. *ACM Trans. Graph.* 33, 4, Article 135 (July 2014), 14 pages. https://doi.org/10.1145/2601097.2601154

Thomas Müller, Brian McWilliams, Fabrice Rousselle, Markus Gross, and Jan Novák. 2018. Neural Importance Sampling. arXiv:cs.LG/1808.03856

Oliver Nalbach, Elena Arabadzhiyska, Dushyant Mehta, Hans-Peter Seidel, and Tobias Ritschel. 2017. Deep Shading: Convolutional Neural Networks for Screen-Space Shading. *Computer Graphics Forum (Proc. EGSR 2017)* 36, 4 (2017), 65–78.

Srinivasa G. Narasimhan, Mohit Gupta, Craig Donner, Ravi Ramamoorthi, Shree K. Nayar, and Henrik Wann Jensen. 2006. Acquiring scattering properties of participating media by dilution. *ACM Trans. Graph.* 25, 3 (2006), 1003–1012.

Jan Novák, Derek Nowrouzezahrai, Carsten Dachsbacher, and Wojciech Jarosz. 2012. Virtual Ray Lights for Rendering Scenes with Participating Media. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 31, 4, Article 60 (jul 2012), 11 pages. https://doi.org/10.1145/2185520.2185556

Matt Pharr, Wenzel Jakob, and Greg Humphreys. 2016. *Physically Based Rendering: From Theory to Implementation (3rd ed.)* (3rd ed.). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. 1266 pages.

Simon Premože, Michael Ashikhmin, and Peter Shirley. 2003. Path Integration for Light Transport in Volumes. In *Proceedings of the 14th Eurographics Workshop on Rendering (EGRW '03)*. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 52–63. http://dl.acm.org/citation.cfm?id=882404.882413

Chen Shen, James F. O'Brien, and Jonathan R. Shewchuk. 2004. Interpolating and Approximating Implicit Surfaces from Polygon Soup. *ACM Trans. Graph. (Proc. SIGGRAPH)* 23, 3 (Aug. 2004), 896–904. https://doi.org/10.1145/1015706.1015816

Jos Stam. 1995. Multiple scattering as a diffusion process. In *Rendering Techniques '95*. Springer Vienna, Vienna, 41–50.

Jerry Tessendorf. 1987. Radiative transfer as a sum over paths. *Physical review A* 35, 2 (1987), 872.

Thijs Vogels, Fabrice Rousselle, Brian McWilliams, Gerhard Röthlin, Alex Harvill, David Adler, Mark Meyer, and Jan Novák. 2018. Denoising with Kernel Prediction and Asymmetric Loss Functions. *ACM Trans. Graph.* 37, 4, Article 124 (July 2018), 15 pages. https://doi.org/10.1145/3197517.3201388

Bruce Walter, Pramook Khungurn, and Kavita Bala. 2012. Bidirectional Lightcuts. *ACM Trans. Graph.* 31, 4, Article 59 (July 2012), 11 pages. https://doi.org/10.1145/2185520.2185555

Jiaping Wang, Shuang Zhao, Xin Tong, Stephen Lin, Zhouchen Lin, Yue Dong, Baining Guo, and Heung-Yeung Shum. 2008. Modeling and Rendering of Heterogeneous Translucent Materials Using the Diffusion Equation. *ACM Trans. Graph.* 27, 1, Article 9 (March 2008), 18 pages. https://doi.org/10.1145/1330511.1330520

Douglas R. Wyman, Michael S. Patterson, and Brian C. Wilson. 1989. Similarity relations for the interaction parameters in radiation transport. *Appl. Opt.* 28, 24 (Dec 1989), 5243–5249. https://doi.org/10.1364/AO.28.005243

Shuang Zhao, Ravi Ramamoorthi, and Kavita Bala. 2014. High-order Similarity Relations in Radiative Transfer. *ACM Trans. Graph.* 33, 4, Article 104 (July 2014), 12 pages. https://doi.org/10.1145/2601097.2601104

Quan Zheng and Matthias Zwicker. 2018. Learning to Importance Sample in Primary Sample Space. arXiv:cs.LG/1808.07840