DOI: 10.1111/cgf.142623
EUROGRAPHICS 2021 / N. Mitra and I. Viola
(Guest Editors)
Volume 40 (2021), Number 2

Learning Multiple-Scattering Solutions for Sphere-Tracing of Volumetric Subsurface Effects

L. Leonard[1,2], K. Höhlein[2] and R. Westermann[2]

[1]University of Havana, Cuba
[2]Technical University of Munich, Germany

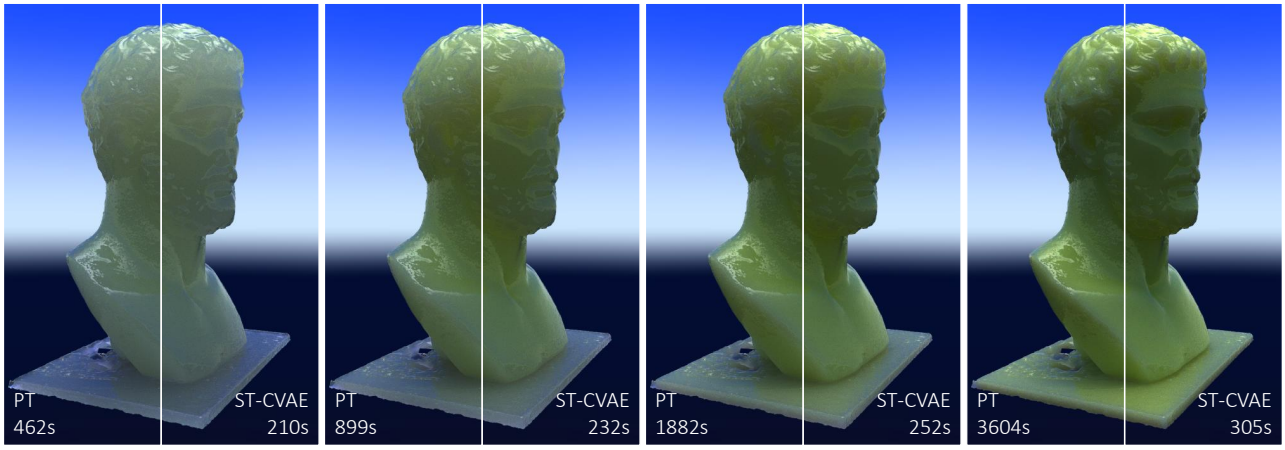


Figure 1: *Subsurface scattering simulation for homogeneous media with varying density and high anisotropy ($g = 0.875$) using standard path-tracing (PT) vs. our approach – a sphere-tracing algorithm using data-driven learning of multiple-scattering statistics (ST-CVAE). 8000 light paths per pixel are simulated. Density doubles from left to right, increasing the number of scatter events during path-tracing. Our approach achieves improved performance by summarizing long scattering chains into a single sphere-tracing step.*

Abstract

Accurate subsurface scattering solutions require the integration of optical material properties along many complicated light paths. We present a method that learns a simple geometric approximation of random paths in a homogeneous volume with translucent material. The generated representation allows determining the absorption along the path as well as a direct lighting contribution, which is representative of all scatter events along the path. A sequence of conditional variational auto-encoders (CVAEs) is trained to model the statistical distribution of the photon paths inside a spherical region in the presence of multiple scattering events. A first CVAE learns how to sample the number of scatter events, occurring on a ray path inside the sphere, which effectively determines the probability of this ray to be absorbed. Conditioned on this, a second model predicts the exit position and direction of the light particle. Finally, a third model generates a representative sample of photon position and direction along the path, which is used to approximate the contribution of direct illumination due to in-scattering. To accelerate the tracing of the light path through the volumetric medium toward the solid boundary, we employ a sphere-tracing strategy that considers the light absorption and can perform a statistically accurate next-event estimation. We demonstrate efficient learning using shallow networks of only three layers and no more than 16 nodes. In combination with a GPU shader that evaluates the CVAEs' predictions, performance gains can be demonstrated for a variety of different scenarios. We analyze the approximation error that is introduced by the data-driven scattering simulation and shed light on the major sources of error.

CCS Concepts

• ***Computing methodologies*** → ***Neural networks; Ray tracing;***

1. Introduction

The realistic rendering of translucent solid materials such as wax or skin requires simulating the scattering of light in the interior of the body. Subsurface scattering refers to the mechanism where the light that penetrates the boundary of a translucent object is scattered internally until it exits the body at a different location on the boundary. Due to the internal scattering of light and the resulting complicated light paths that have to be considered in the simulation, an accurate subsurface scattering solution becomes computationally expensive.

To reduce the computational requirements, some previous methods exploit special shape and material configurations, such as planar surface geometry, isotropic scattering, or separability of the bidirectional scattering-surface reflectance distribution function (BSSRDF) into location- and direction-dependent terms. While efficient analytic transport solutions can be developed for some special cases, rendering quality decreases for scenes where the underlying assumptions are violated. To improve the generalizability of rendering acceleration solutions, there has been growing interest in data-driven approaches that can infer approximate multiple-scattering solutions at render time.

One of the earliest methods precomputes long-distance light transport through a homogeneous medium using transition probabilities at varying spatial scales [MWM07]. This so-called shell tracing approach enables moving rays through the volume in large steps without considering individual scattering events. It builds upon the concept of sphere-tracing for surface rendering [Har96], to adapt the step size to how deep the ray is in the solid body. Recently, for rendering volumetric media without an explicit boundary representation, deep-scattering [KMM*17] has proposed to feed geometric information into a network by progressively evaluating a hierarchical volumetric geometry descriptor. The inference is based on an expressive multi-layer neural network. To address specifically the issue of boundary effects, a method for learning a shape-aware BSSRDF model from ground truth volumetric transport simulations using path tracing has been proposed by Vicini et al. [VKJ19]. The method avoids many limiting assumptions of prior analytic models regarding the planarity of surfaces and isotropy of volumetric transport, and it generalizes the built-in notion of spatio-directional separability. It assumes a local polynomial surface expansion around a shading point and uses this expansion to re-project predicted off-surface samples back onto the surface.

Our work builds upon prior works in data-driven subsurface scattering simulation by leveraging the core idea of "bypassing a potentially lengthy internal scattering process" [VKJ19] through a network that has learned to sample outgoing light locations on an object's surface from an incident light location. In contrast to existing work in subsurface scattering simulation, our method avoids an explicit encoding of the local surface geometry in the stochastic modelling of the scattering process, and it also infers the outgoing light direction from a reference distribution generated via volumetric path tracing, instead of using importance sampling. To achieve this, we combine shell and sphere tracing with network-based learning of long-distance light transport between a point and a spherical surface centered therein. A sequence of neural-network-based conditional variational auto-encoders (CVAEs) is trained to model the statistical distribution of light paths, and absorption along them, inside a spherical region in presence of multiple scattering events. We subsequently call the proposed combination of CVAEs with sphere tracing ST-CVAE. A first CVAE learns to sample the number of scattering events, i.e., the absorption, occurring along a path inside the sphere. Conditioned on this, a second model predicts the exit position and continuation direction of this path. By using shell tracing, light paths can be traced efficiently through the volumetric medium toward the solid boundary without an explicit encoding of the boundary geometry.

Furthermore, instead of modelling purely the one-to-one light transport between two surface points, our method supports the integration of direct in-scattering to achieve improved convergence. For this, a third network model generates a representative sample of photon position and direction along the path, which is used to approximate the contribution of direct illumination due to in-scattering. In combination with existing approaches for simulating the direct interior illumination, the path-tracing solution converges quickly even when none of the paths arrives directly at a light source.

We jointly train all three networks on a spherical geometry with unitary radius and varying interior density (different radii can be considered by adjusting the density appropriately). In the training and validation phase, ground truth transport solutions using Monte Carlo path tracing are used. An interesting result is that all proposed networks are light-weight, comprised of only up to three layers with no more than 16 nodes each. Despite the resulting compact internal latent-space representations, very realistic reconstructions of the modeled distributions are obtained. Due to this compactness, all inference steps can be implemented efficiently using short shader programs and matrix multiplication instructions on the GPU. We analyze the performance and accuracy of our approach on different geometries and material properties and investigate the generative modelling skills of each of our network models individually. ST-CVAE achieves quality on par with that of a reference Monte Carlo path tracing solution, at significantly improved performance. Code for reproducing the results is publicly available at [LHW21].

2. Related Work

Compared to traditional ray tracing algorithms that can assume a free traverse of the light between surfaces, ray tracing participating media represents a superior level in computational complexity. Due to the plethora of possible material interactions, the rendering equation for realistic scene settings does not possess a closed-form solution in presence of participating media. Thus, for synthesizing realistic images, approximate solutions are required.

Analytical approximation schemes Traditional approximation schemes include the exclusion of higher-order scattering effects, leading to the single-scattering approximation [SRNN05] or approaches based on luminance diffusion [KVH84, FPW92]. A thorough review of traditional rendering techniques for participating media can be found in [CPP*05]. Due to their greater generality, research has focused on diffusion-based methods more recently. Diffusion models approximate the light transport by virtue of bidirectional scattering-surface reflectance dis-

tribution functions (BSSRDFs), which account for the interaction of light with the object's surface and light transport underneath the surface of translucent objects. Though recent methods like [dI11, YZXW12, d'E13, FHK14] can generate visually appealing results at moderate rendering time, diffusion approximations typically involve restrictive assumptions on homogeneity in the considered material and surface geometry, which limits physical realism and applicability of the methods in more complex settings.

Monte Carlo methods More flexible approaches for global illumination simulation arise from the application of Monte Carlo path tracing algorithms [KVH84, Rus89, HK93, LW96]. Monte Carlo path tracing in a converged setting yields physically accurate images that can serve as a "ground-truth" for validating alternative approaches. The recent survey by Novak et al. [NGHJ18] reviews the latest advances in Monte Carlo methods for solving the light transport in participating media. In path tracing, and as an alternative to ray-marching with constant step size, delta tracking is often used to determine free path lengths according to the optical depth in the participating medium. For importance sampling of the corresponding probability density function, Woodcock tracking [WMHL65] adjusts the sampling distances so that dense regions in a volume can be sampled appropriately, and it has been adapted to achieve unbiased sampling of sparse inhomogeneous media [YIC*10]. Free path sampling with probabilities not necessarily proportional to the volume transmittance has been realized by means of weighted delta tracking approaches, e.g. [NSJ14, KHLN17, RKDS19]. All these techniques reduce noise in the estimated light paths for participating media, and they enable an automatic selection of the step size according to the underlying material distribution. A significant amount of work has been devoted to the acceleration of Monte Carlo rendering. Popular approaches include importance sampling, e.g. [CCY67, VG95] and next-event estimation [CCY67, KNK*16, HDF15], which modulate the sampling distribution in a way that minimizes the variance of the estimator, or making use of hybrid methods to weigh out the advantages and disadvantages of Monte-Carlo-based and approximated rendering approaches, e.g. [HCJ13]. Similarity relations in the material parameter space have been exploited to identify analogies that yield an equivalent solution of the rendering equation but are faster to simulate, e.g. [CMA*99, ZRB14].

Shell tracing and sphere tracing Another approach to accelerate the rendering process is to summarize individual scattering events within a suitable transfer function and moving rays through the volume with maximally large, adaptive steps. One realization of this idea is known as shell tracing, which has been proposed for rendering participating media [LO07] and homogeneous random media [MWM07, MPH*15, MPG*16]. In the spirit of sphere tracing for surface rendering [Har96], related techniques build upon shortcutting photon paths through spherical regions of a homogeneous medium, enabling a large simulation step size in the interior of the body and avoiding the simulation of numerous scattering events inside the spheres. Traditionally, the sphere transition probabilities have been precomputed on a discretized grid and stored in multi-dimensional tables. The discretization, however, limits the expressiveness of the statistical model, and storage of high-dimensional tables in rapidly-accessible form can lead to severe memory requirements. Moon et al. [MWM07] approach this problem by compactly representing the tables using matrix factorization, which has the disadvantage of requiring decompression steps at rendering time. Müller et al. [MPG*16] utilize an uncompressed 4D table with a non-uniform discretization of individual dimensions to keep the memory footprint low. This table covers different albedos, phase functions, and densities. Lee and O'Sullivan [LO07] report a very moderate memory consumption but focus only on a single setting of material parameters, which naturally reduces the memory requirements, but disables application of the method in a general-purpose rendering pipeline. Meng et al. [MPH*15] introduce an alternative notion of shell transfer functions and derive a simple homogeneous approximation to circumvent the storage problem entirely.

Machine learning and neural networks for rendering Since Monte Carlo-based rendering intrinsically can be described as a statistical problem, the use of machine learning methods for rendering has recently attracted great interest. In the context of path tracing, the goal has been to use machine learning models to model sampling distributions or create guidance for efficient sample selection. Popular modeling approaches frequently involve Gaussian mixture models [VKŠ*14, HEV*16] or deep neural networks. Deep learning methods, thereby, can be classified into image-based adaptive sampling and denoising algorithms on the one hand, and multiple-scattering approximations on the other hand.

The recent approach by Kuznetsov et al. [KKR18] facilitates learning adaptivity in Monte Carlo path-tracing and denoising of the final image. A first network learns to adapt the number of additional paths from an initial image at the target resolution, which is generated via one path per pixel. A second denoising network learns to model the relationship between an image with increased variance in the color samples and the ground truth rendering [RACSKS*17, MMBJ17]. Recently, Weiss et al. [WITW20] used neural networks to learn the positions of sample locations in image space from a low-resolution image, using a differentiable sampling stage as well as a differentiable image reconstruction stage that can work on a sparse set of samples. Related to denoising approaches for path-traced images is the screen space shading approach by Nalbach et al. [NAM*17], where a network is trained to infer missing shading information from image information in a view-dependent G-buffer.

For surface graphics, Zheng and Zwicker use neural networks to model the relationships between scene parameters and light paths [ZZ19]. Deep-scattering [KMM*17] employs a radiance-predicting neural network for simulating scattering events in clouds. Geometry information is fed into the network by progressively evaluating a hierarchical volumetric geometry descriptor, and inference is based on an expressive multi-layer neural network, optimized with recent design features. Nevertheless, the method was developed for cloud rendering and does, as such, not explicitly account for boundary effects, which may play a major role in sub-surface scattering. Addressing specifically this issue, a method for learning a shape-aware BSSRDF model from ground truth volumetric transport simulations using path tracing has been proposed [VKJ19]. Exploiting the expressiveness of neural network models in a similar way, this method avoids many limiting assumptions of prior analytic models regarding the planarity of surfaces

and isotropy of volumetric transport, and it generalizes the built-in notion of spatio-directional separability. It assumes a locally expandable surface geometry, which is explicitly encoded using a trivariate polynomial around a shading point. In this way, an approximate signed-distance function to the surface geometry can be used to model the outgoing on-surface light distribution. Vicini et al. [VKJ19] overcome the limitation of the planar approximation via a learned approximation of subsurface scattering, by fitting the geometry with a cubic polynomial. Assuming a uniform distribution of the final outgoing direction and a single polynomial surface approximation, however, biases the final result.

3. Background and methods

The key idea underlying ST-CVAE is to train generative statistical models that can bypass multiple scattering events occurring inside a spherical volume of material with constant density and potentially anisotropic scattering characteristics. The learned representations are then used in the rendering process to generate paths through the volume using large steps (Figure 2a). Whenever the ray arrives at a sampling point, it tests for the largest step it can make without leaving the volume, and then evaluates a sequence of statistical models to infer the end position and direction after performing this step. The ray proceeds using delta tracking until the next collision event is determined and recurrently performs learning-based path tracing (Figure 2b). To determine the length of the step, a 3D signed distance function is computed for the current geometry in a preprocess. For this, the object's bounding volume is discretized using a 3D voxel grid, and for each voxel a conservative shortest Chebyshev distance to the object's surface is computed [DK19]. This representation enables to quickly find the radius of the sphere that can be safely traversed without intersecting the geometry. As the ray approaches the edges, it is forced to take smaller and smaller steps until the tracking process passes across the boundary surface. To efficiently trace against a polygonal boundary surface, we employ a GPU ray-tracer using NVIDIA's RTX ray tracing interface [NVI18] through the DirectX Raytracing API (DXR).

3.1. Scattering model and data generation

Learning is conducted in an offline process. We assume homogeneous extinction $\sigma_t = \sigma_a + \sigma_s$ and scattering albedo $\varphi = \sigma_s/\sigma_t$, where σ_a and σ_s are the absorption and scattering coefficient, respectively. Anisotropic scattering is modelled via the Henyey-Greenstein phase function [HG41]. The radiative transport [Cha50] can be described in terms of the volume rendering equation [KVH84],

$$L(x,\omega) = \int_{t=0}^{d_z} T(x,x_t)\left[\sigma_a L_e(x_t,\omega) + \sigma_s L_s(x_t,\omega)\right] dt + T(x,z)\, L_d(z,\omega),$$

where $x_t = x - \omega t$ and $z = x_{d_z}$. $L(x,\omega)$ describe the incoming radiance at position x from direction ω, $L_e(x,\omega)$, $L_s(x,\omega)$ and $L_d(x,\omega)$ denote emitted, scattered and direct radiance, and $T(x,y)$ is the transmittance along the ray (Beer-Lambert law [Lam60]), with $T(x,x_t) = \exp(-\sigma_t t)$ in the case of homogeneous medium. The phase function is considered in $L_s(x_t,\omega)$.

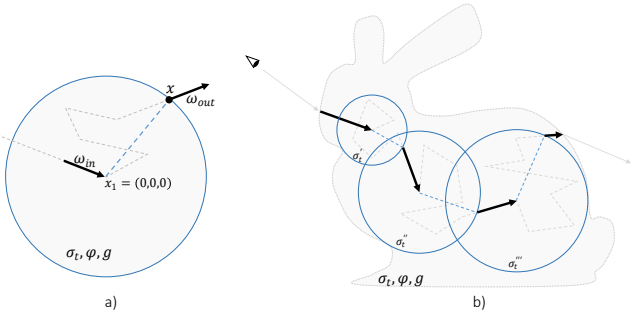


Figure 2: *Method overview. a) Given optical properties of the volumetric medium (σ_t, φ, g), the path (gray dotted lines) starting at $x_1 = (0,0,0)$ from the incoming direction $w_1 = w_{in}$ is bypassed by letting a network infer directly (blue) the random outgoing position x and direction w_{out} in a unitary radius sphere. b) After free-travel events (black arrows), the maximum sphere not intersecting the geometry is computed from a signed distance field, and the radius is used to control the adaptive density and the length of the next step along a light path. This process is repeatedly applied until the path leaves the object.*

For training the model, we assume that rays start at the center of a sphere of unitary radius and terminate when they cross the boundary of the sphere for the first time, after experiencing a sequence of scattering events in between. By assumption, the incident direction of the rays in the sphere center is set to $\omega_{in}^{CVAE} = (0,0,1)$ for all rays, and the first scattering event occurs directly in the center of the sphere. For each ray, Monte Carlo path tracing is used as a ground truth renderer [LW96, NGHJ18]. The distance between subsequent scattering events is estimated as $l = -\log(1-\xi)/\sigma_t$, where ξ is a uniformly distributed random number over the interval $[0,1)$. I.e., the mean free path is equal to the reciprocal of the extinction. At every scattering event, it is decided whether the path continues or terminates due to absorption using importance-based Russian roulette [EUVN87]. In the case of continuation, the scattering direction is computed by importance sampling the Henyey-Greenstein phase function.

ST-CVAE considers only the distribution of outgoing light on the sphere and ignores collisions inside the sphere. The distribution is determined by the following parameters: the number N of scattering events that occur inside the sphere, determining the likelihood of absorption A (ray terminates due to absorption, $A = 1$, or not, $A = 0$), as well as the exit position x on the sphere and the outgoing direction w of the ray. To further support the use of the proposed method in the more general context of path tracing, we consider the sampling of additional ray statistics, which we summarize as Σ.

In account of these considerations, the likelihood of sampling a particular ray path with summary statistics $(N, A, x, \omega, \Sigma)$ is determined by a conditional probability density function (PDF) of the form $p(N, A, x, \omega, \Sigma | \sigma_t, g, \varphi)$, where the quantities to the right of the vertical bar affect the PDF only as conditions.

3.2. Model decomposition and inference

Over the last years, a variety of approaches have been proposed to efficiently learn complex statistical distributions via neural networks. These include (conditional) variational auto-encoders [KW13, MO14], generative adversarial networks [GPAM*14] or invertible neural networks [AKW*18], which all can be extended to learn conditional distributions, see e.g. [SLY15, MO14, ALK*19]. We propose to train conditional variational auto-encoders (CVAEs) to learn a representation of $p(N,A,x,\omega,\Sigma|\sigma_t,g,\varphi)$ that offers a good trade-off between computational model complexity, model flexibility, and training stability.

To enable sequential model evaluation and minimize the need for modeling complex correlation structures between random variables, we decompose $p(N,A,x,\omega,\Sigma|\sigma_t,g,\varphi)$ according to

$$\begin{aligned} p(N,A,x,\omega,\Sigma|\sigma_t,g,\varphi) = {} & p_N(N \mid \sigma_t,g) \\ & \times p_A(A \mid N,\varphi) \\ & \times p_{x,\omega}(x,\omega \mid N,A,\sigma_t,g) \\ & \times p_\Sigma(\Sigma \mid N,A,x,\omega,\sigma_t,g,\varphi). \end{aligned} \tag{1}$$

The full model is decomposed into three separate CVAEs, one for each of the distributions p_N, $p_{x,\omega}$, and p_Σ. We subsequently call these CVAEs LENGTHGEN, PATHGEN, and EVENTGEN (Figure 3). At inference time, the models are evaluated sequentially. The first model then infers the number N of scattering events along the path. The PDF p_N depends only on σ_t and g, because we assume in this step that a full path – starting at the sphere center and terminating when crossing the sphere boundary without being absorbed in between – is observed. This approach simplifies the generation of training data, especially in scenarios with significant absorption. Conditioned on N and φ, the probability of absorption A along the bypassed path follows a Bernoulli distribution

$$p_A(A \mid N,\varphi) = q(N,\varphi)^A (1-q(N,\varphi))^{1-A}, \tag{2}$$

with $q(N,\varphi) = 1-\varphi^N$. This admits simple sampling, so that training of a CVAE is not required. If sampling yields $A=1$, the path is considered as being absorbed and no further sphere-tracing steps need to be performed.

Note here that it would also be possible to omit the number of scattering events N from the inference procedure and model the probability of absorption directly through a CVAE or tabulation-based approach. However, especially in cases where the absorption probability is very high, for example, due to high anisotropy, dense medium, or high albedo, it can take a prohibitively large number of rays until the first non-absorbed is found. Then, obtaining a statistically accurate estimate of the absorption probability from simple sampling procedures becomes difficult. Modeling the distribution p_N instead, and deducing p_A through Equation (2), is statistically much better behaved and allows for higher accuracy in the estimation of p_A.

The second CVAE infers the position on the sphere surface where the ray exits the volume as well as the direction into which it continues (Figure 2a). Note here that $p_{x,\omega}$ is independent of φ. This can be assumed because the absorption process is treated separately before x and ω are sampled, and continuation of the path at x (in direction ω) is conditioned on the outcome of the absorption request A. If applicable, the third model finally resolves the summary statistics Σ. Since in general no dependencies can be excluded, p_Σ may depend on all previously sampled variables, as well as on all material parameters.

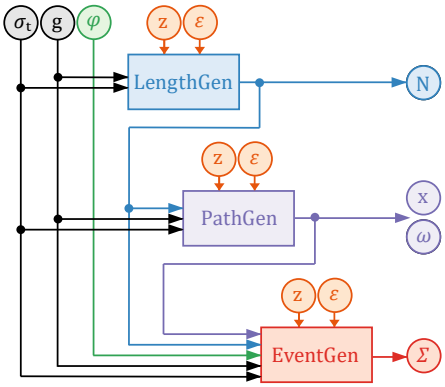


Figure 3: *Overview of the model architecture, consisting of three trainable CVAE models,* LENGTHGEN, PATHGEN, *and* EVENTGEN. *Variables z and* ε *refer to random numbers which are used for sampling the generative models.*

3.3. In-scattered direct illumination

One possible application of the third model, EVENTGEN, is the inclusion of next-event estimation for in-scattering of direct illumination [CCY67, NGHJ18] into the rendering process. In standard path tracing, the in-scattering of direct illumination is either neglected during the internal scattering process and considered only at the position where the path exits the volume, or the in-scattering is approximated at internal sampling points, e.g., by using precomputed photon maps [NNDJ12] or direct illumination using next-event estimation [CCY67, KNK*16, HDF15]. Since the samples at which internal scattering occurs are not available anymore at render time, we employ the EVENTGEN model and set $\Sigma = (X,W)$, wherein X indicates the position of the scattering event in the interior of the sphere and W the direction of the ray arriving at X. During training, (X,W) is sampled from the set of all scattering events on the path through the sphere with weights according to the contribution of the in-scattering events to the luminosity of the ray.

For example, the direct light contribution from a directional light source at position x_L with power Φ along a path inside a translucent medium can be considered at every internal sampling point of the ray path. Characterizing the ray through the occurring scattering events, $(x_1,\omega_1),\ldots,(x_N,\omega_N)$, the total in-scattering L due to direct illumination along the path is obtained as

$$L = \Phi \sum_{i=1}^{N} \varphi^i \rho_g(\omega_i, \omega_{x_L \to x_i}) \tau(x_L \to x_i). \tag{3}$$

Here, $\omega_{x_L \to x_i}$ indicates the direction of the incoming direct illumination at position x_i and $\tau(x_L \to x_i)$ denotes the extinction of the light when traveling from the light source through the volume to the scattering position.

Since in path tracing typically the expectation value $\langle L \rangle$ over

many paths is considered, we approximate L in a Monte Carlo fashion, by choosing one single representative sample (X,W) per path and considering the direct in-scattering from the light source only for this scattering event. A sample (X,W) is chosen to be the kth scattering event in a path $\{(x_i,\omega_i)\}_{i=1}^{N}$ with probability $\varphi^k/\Lambda^{(N)}$, where $\Lambda^{(N)}=\sum_{i=1}^{N}\varphi^i$. This approximation does not alter $\langle L\rangle$ if a sufficient number of rays is considered in the average.

The crucial step in applying this approximation lies in efficiently computing the incoming direction of the direct light $\omega_{x_L\to X}$ as well as the attenuation coefficient $\tau(x_L\to X)$ for all positions X. In absence of refractive boundaries, the expressions simplify to

$$\omega_{x_L\to X}=\frac{x_L-X}{\|x_L-X\|}\quad\text{and}\quad\tau(x_L\to X)=\exp(-d_t(x_L\to X)\sigma_t),$$

with $\|\cdot\|$ indicating the standard Euclidean vector norm, and $d_t(x_L\to X)$ referring to the distance along the direct path between x_L and X along which the light travels through the medium. In the more general case of refractive medium boundaries, alternative approaches need to be employed [WZHB09, Hol15, KNK*16]. Notably, the use of such approaches does not affect the applicability and performance gain of the path-tracing acceleration method proposed here.

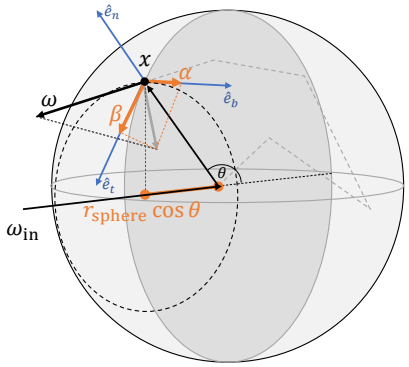


Figure 4: *Path parameterization. Due to rotation symmetry of the phase function, it is useful to parameterize exit position x and outgoing direction* ω *of a ray invariant with respect to rotations around* ω_{in}*. All exit positions on a circle of constant deviation angle* θ *with respect to* ω_{in} *(dotted circle) are equally likely. Exit position x of a path is therefore parameterized through* $\cos(\theta)$*. Coordinates of the outgoing direction* ω *are given in a local reference frame around x, with normal vector* $\hat{e}_n$*, binormal* $\hat{e}_b$ *and tangent* $\hat{e}_t$*, yielding coordinates* α *(projection on* $\hat{e}_b$*) and* β *(projection on* $\hat{e}_t$*).*

4. Learning multiple scattering in spherical volumes

While training the CVAEs can be performed in normalized conditions, i.e. in a unit sphere with all rays coming from the same direction, the parameters have to be adapted to the local frame of reference of the current ray at render time. Also, the radius of the sphere has to be adapted to the geometry of the scene-setting. To account for this, the output parameters of the models, i.e., x, ω, X, and W, have to be parameterized in a way that is invariant to rotations of the spatial coordinates and changes of the sphere radius. To account for this, we propose a parameterization scheme as shown in Figure 4.

4.1. Path parameterization

After sampling N and A, the outgoing position x on a sphere shell of radius r_{sphere} is parameterized in spherical coordinates through a tuple $(r_{\text{sphere}},\theta,\psi)$, where $\cos(\theta)=\cos(\omega_{\text{in}},x)$. Due to invariance of the distribution $p_{x,\omega}$ with respect to rotations around ω_{in}, ψ can be sampled as a random number from a uniform distribution between 0 and 2π. This invariance is inherited from the rotation invariance of the Henyey-Greenstein phase function $\rho_g(\omega,\omega')$ and can be used to restrict the learning-space of the CVAE to only the θ-coordinate. The rotation invariance furthermore simplifies the matching between real-space coordinates and CVAE reference system, since the orientation of the basis vectors orthogonal to ω_{in} can be chosen arbitrarily, as long as the CVAE-reference direction $\omega_{\text{in}}^{\text{CVAE}}=(0,0,1)$ is mapped appropriately to the true ω_{in} in real-space coordinates. Given a suitable rotation matrix R^{CVAE}, the direction x can be sampled as

$$x=r_{\text{sphere}}R^{\text{CVAE}}R_{\psi}^{(0,0,1)}\tilde{x},\tag{4}$$

where $\tilde{x}=(0,\sin(\theta),\cos(\theta))$, and $R_{\psi}^{(0,0,1)}$ denotes the rotation matrix corresponding to the random rotation of angle ψ around the CVAE-reference direction $\omega_{\text{in}}^{\text{CVAE}}=(0,0,1)$.

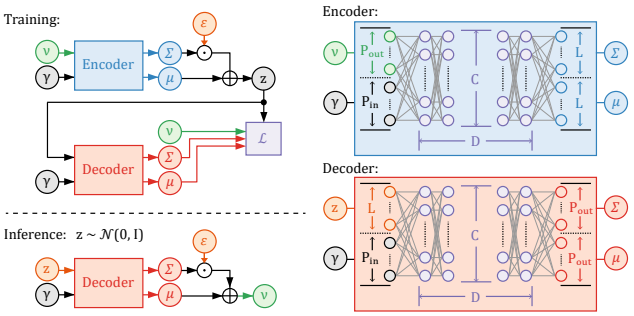
Given x, a local frame of reference can be obtained in x by considering the coordinate system

$$\hat{e}_{\text{n}}=\hat{x},\quad\hat{e}_{\text{b}}=\omega_{\text{in}}\times\hat{x},\quad\hat{e}_{\text{t}}=\hat{e}_{\text{b}}\times\hat{e}_{\text{n}},$$

with $\hat{x}=x/r_{\text{sphere}}$, and $\hat{e}_{\text{n}}$, $\hat{e}_{\text{b}}$ and $\hat{e}_{\text{t}}$ referring to normal, binormal and tangent vectors of the frame of reference around x. The operator $\times$ denotes the cross product in $\mathbb{R}^3$. Within this system, the out-going direction ω can be parameterized through a tuple (α,β), wherein $\alpha=\cos(\omega,\hat{e}_{\text{b}})$ and $\beta=\cos(\omega,\hat{e}_{\text{t}})$. Again exploiting rotation invariance, ω can be sampled in the CVAE reference system and can be transformed to real-space by applying a transformation similar to that in Equation 4.

The treatment of summary statistics Σ has to be considered separately and depending on which quantities are required. For $\Sigma=(X,W)$, with X and W describing position and direction of a representative scattering event, a symmetry-guided restriction of sample space is not admissible since the statistics may depend on x and ω in a complex manner. Therefore, X and W are sampled as real-valued 3D vectors in the CVAE system of reference with W being normalized subsequently. Both quantities are then transformed to real-space coordinates by applying a suitable rotation matrix and scaling.

To account for varying radii r_{sphere} in the model setting, the extinction coefficient of the medium is re-scaled according to $\sigma_{\text{t}}\to\sigma_{\text{t}}r_{\text{sphere}}$ before applying the models. This is justified because of homogeneity of the medium and independence of scattering events



(a) *Working principle of CVAE.* **(b)** *Model realization in this work.*

Figure 5: *(a) Working principle of the CVAE. At training time, both encoder and decoder are used to enable closed-loop training of the model. At sampling time only the decoder is evaluated. Parameters* ε *are random variables to evaluate the Gaussian posterior distributions, parametrized by mean* μ *and covariance* Σ *of the respective model. (b) Configuration of encoder (blue) and decoder (red) networks for CVAEs. Both models are determined through the number of input parameters,* P_{in} *(dimension of the conditioning variables,* γ*), the number of output parameters,* P_{out} *(dimension of the variables to be predicted,* ν*), the latent space dimension L, the depth of the networks D, and the number of node channels per layer C. Encoder and decoder share the same settings of D, C and L, and output estimates for mean* $\mu_{Enc/Dec}$ *and diagonal covariance matrix* $\Sigma_{Enc/Dec}$ *of the respectively modeled distributions.*

therein. All other variables are unaffected by the transformation between model- and real-space coordinates.

4.2. Model realization

Each CVAE model is realized via an encoder and a decoder, as shown in Figures 5a and 5b. The encoders establish mappings between their respective random variates ν, and a probabilistic latent-space representation z, from which samples can be drawn. The mappings are conditioned on the material parameters and previously sampled random variables γ. For the three models, we have $\nu_{\text{Length}} = N$, $\nu_{\text{Path}} = (x, \omega)$ and $\nu_{\text{Scatter}} = \Sigma$, as well as $\gamma_{\text{Length}} = (\sigma_t, g)$, $\gamma_{\text{Path}} = (N, A, \sigma_t, g)$ and $\gamma_{\text{Scatter}} = (N, A, x, \omega, \sigma_t, g, \varphi)$, according to the distribution splitting in Equation (1). We assume a Gaussian shape of the encoder posterior, i.e.

$$q_{\text{Enc}}(z \mid \nu, \gamma) = \mathcal{N}(\mu_{\text{Enc}}(\nu, \gamma), \Sigma_{\text{Enc}}(\nu, \gamma)),$$

where the mappings $\mu_{\text{Enc}}(\nu, \gamma)$ and $\Sigma_{\text{Enc}}(\nu, \gamma)$ represent the conditional mean and diagonal covariance of the distribution, both learned through the networks. The decoders learn to invert the action of the respective encoder and are parameterized accordingly through

$$p_{\text{Dec}}(\nu \mid z, \gamma) = \mathcal{N}(\mu_{\text{Dec}}(z, \gamma), \Sigma_{\text{Dec}}(z, \gamma)).$$

During rendering, only the decoders are used and need to be kept as simple as possible. All our models follow the scheme shown in Figure 5b and are determined through the number of input parameters P_{in}, the number of output parameters, P_{out}, the latent space dimension L, the number of hidden layers in the networks, i.e., the depth of the networks D, and the number of node channels per layer C. Both encoder and decoder networks use Softplus activation functions [ZYL*15] between hidden layers. The outputs of the final layers, $\mu_{\text{Enc/Dec}}$ and $\Sigma_{\text{Enc/Dec}}$, are transformed by linear activation functions. Note, however, that for the diagonal covariance matrices we predict $\log(\Sigma_{\text{Enc/Dec}})$ instead of $\Sigma_{\text{Enc/Dec}}$ directly. Both encoder and decoder networks share the same settings for D and C, which were optimized in empirical experiments. Parameter selection was guided by efficiency considerations. Since matrix multiplications on the GPU are carried out in matrix sections of size 4×4, we adapted D, C and L so that especially the decoder model is able to efficiently utilize the parallelization capabilities on the GPU. In particular, we considered multiples of four for C-values, to speed-up matrix multiplications between hidden layers, and selected L so that the input size of the decoder, $P_{\text{in}} + L$, is a multiple of four. Concerning D, small values were preferred, as deeper models require longer computation time due to the sequential network evaluation. The models were trained using an Adamax optimizer [KB14], implemented in the PyTorch interface in Python. The settings of the hyper parameters D (depth, i.e. number of layers of the models), C (width, i.e. number of nodes per layer) and L (latent space dimension) used for setting up the CVAEs are summarized in Table 1.

Table 1: *Examined and final parameter settings for the CVAE models* LENGTHGEN, PATHGEN *and* EVENTGEN. *Final parameter settings are highlighted.*

	P_{in}	P_{out}	D	C	L
LENGTHGEN	2	1	1, **2**, 3	4, **8**, 12	**2**, 6
PATHGEN	3	3	1, 2, **3**	8, 12, **16**	**5**, 9
EVENTGEN	7	6	1, 2, **3**	**12**, 16, 20	**5**, 9

5. Results and Comparison

We conduct a number of experiments using different scenarios to analyse the fitting accuracy of the model predictions with respect to the real distributions, and we evaluate the performance of network-based path prediction during rendering. In all scenarios the same network configuration is used, i.e., networks are not retrained to specialize for particular tasks. For training, 4×10^6 ray samples are used with the material parameters $\sigma_t \in [0, 200]$ and $g \in [-1, 1]$ to fully cover the parameter space. The height of the rendered models is always normalized to 1m.

We further compare network-based inference to tabulation-based lookup of precomputed multi-dimensional function values, using non-uniform quantization as proposed by Müller et al. [MPG*16]. For this, we first analyse the effect of tabulation regarding accuracy, speed and memory consumption, by replacing network-based prediction by tabulation-based lookup in our implementation. Since Müller et al. use a different model in their approach, i.e., one that predicts only the exit position but not the continuation direction of a path on the sphere surface, we also analyze the differences that are caused by these model assumptions. It turns out that the use of a more simple model can significantly reduce the memory requirement of tabulation approaches at comparable accuracy, yet making

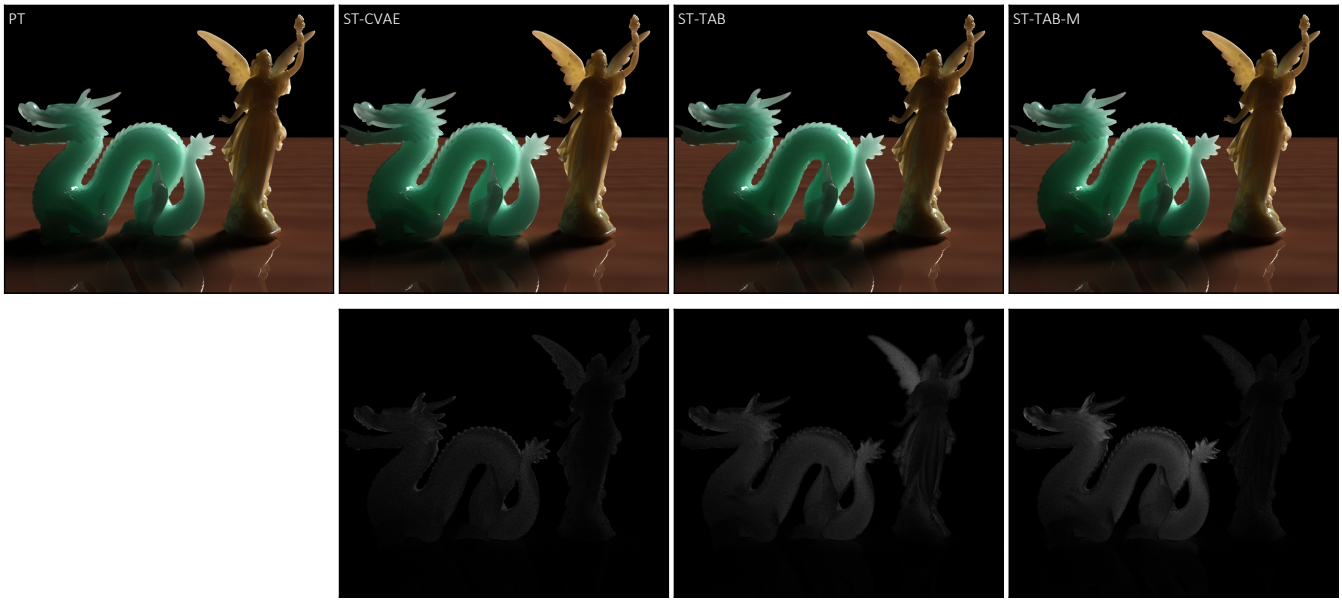


Figure 6: *Quality comparison between standard path tracing (PT), our approach (ST-CVAE), our approach with networks exchanged by tabulation (ST-TAB), and the tabulation approach by Müller et al. [MPG*16] (ST-TAB-M). Top: rendered images. Bottom: Per-pixel intensity difference to PT. Error magnitudes are enhanced by a factor of 4. The dragon model has* $\sigma_t = 0.5\,mm^{-1}$ *and anisotropy of* $g = 0.9$*, and Lucy has* $\sigma_t = (.4, .6, .8)\,mm^{-1}$ *and anisotropy of* $g = 0.6$*. ST-TAB requires* 32% *more time than ST-CVAE, and ST-TAB-M takes* 12% *less time.*

the approach less extensible to more sophisticated illumination effects.

5.1. Visual accuracy

In Figure 6, we compare the rendering results of standard path tracing (PT), our approach in combination with neural network-based inference (ST-CVAE), our approach with tabulation instead of network-based inference (ST-TAB), and the shell-tracing model using tabulation by Müller et al. [MPG*16] (ST-TAB-M). In-scattered direct illumination is not included at this stage, so only LengthGen and PathGen are employed by ST-CVAE. From a perceptual point of view, hardly any differences can be observed between ST-CVAE and PT. Quantitatively, the renderings using ST-CVAE achieve a root mean square error to PT between 0.01 and 0.02 and, thus, come very close to the ground truth. Nevertheless, the question arises whether neural networks are indeed required to realize our proposed method, or if similar results can be achieved with conceptually simpler approaches, such as tabulation-based methods. Replacing network-based inference in our model with non-uniform tabulation at reasonable accuracy, we arrive at a memory consumption of the order of GiB. Even with a table size of 2.5 GiB, significant differences to PT can be observed. Biases are especially high in optically thin regions, such as the wings of Lucy in Figure 6, indicating convergence of the rendering to a vastly different solution. Even though further increasing the table size is feasible within the memory constraints of current GPU devices, the precomputation time grows considerably with increasing table size. This is due to the many light paths that have to be evaluated to achieve an adequate accuracy of the tabulated probabilities. In comparison, the ST-CVAE requires a total of only 2 MiB of memory for storing the network architectures, revealing the advantage of network-based approaches for modelling high-dimensional functions.

Another way of lowering the memory footprint of tabulation has been proposed by Müller et al. [MPG*16]. By imposing separability conditions on the underlying probability distribution, the dimension of the stored tables can be restricted to 4D. Our experiments show that the assumptions by Müller et al. lead to reasonably accurate results, yet with observably larger bias than both realizations of our method. This suggests that resolving the separability assumption has indeed a positive effect on the rendering quality.

Figure 7 further compares the quality of PT and ST-CVAE, not considering in-scattered direct illumination. We use a low-density material under backside lighting to generate prominent translucency effects in both optically thin and thick material parts. In all experiments, we use as many paths as required to reach the same prescribed standard deviation of per-pixel illumination values. In this way, we can assume convergence of all approaches, yet possibly towards different solutions. Difference images are used to show the location and values of per-pixel differences.

5.2. Performance analysis

In the low-density setting in Figure 7, ST-CVAE achieves a speed-up of roughly 1.5x compared to PT (232 s vs. 363 s). The speed-up increases with increasing material density, so that in the high-density case ST-CVAE is already 4.4x faster than PT (432 s vs. 1884 s). This trend continues when the optical density of the ma-

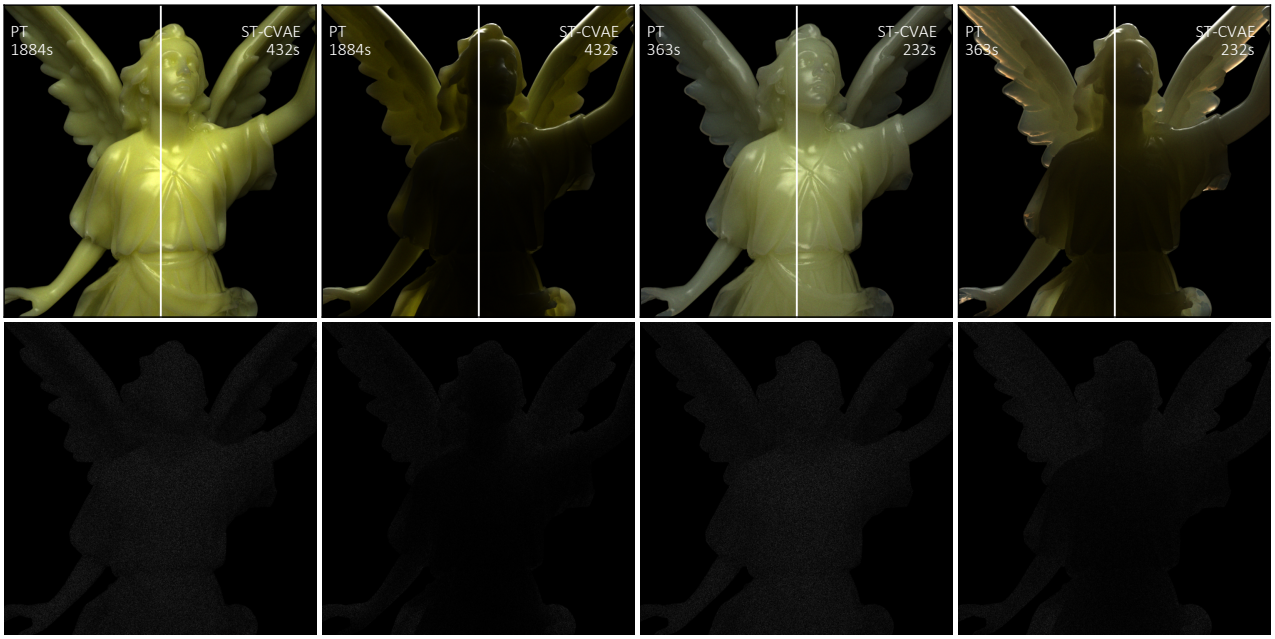


Figure 7: *Quality comparison between standard path tracing (PT) and our approach (ST-CVAE). Top row, from left to right: High-density material under front lighting and back-lighting conditions ($\sigma_t = (0.512\,mm^{-1}, 0.614\,mm^{-1}, 0.768\,mm^{-1})$ and $\varphi = (0.99999, 0.99995, 0.975)$ for channels $(red, green, blue)$), and low-density material ($\sigma_t = (0.128\,mm^{-1}, 0.1536\,mm^{-1}, 0.192\,mm^{-1})$) under front-lighting and back-lighting conditions. Bottom row: Per-pixel intensity difference with respect to PT. Error magnitudes are enhanced by a factor of 4. All images were rendered using 16000 paths per pixel. Rendering times as given.*

terial is further increased. As seen in the teaser (Figure 1), speed-ups of roughly 11x or more can be achieved for sufficiently dense materials.

Given the increase in relative performance gain of ST-CVAE with respect to PT, we further shed light on the different factors causing this behaviour. Therefore, we rendered different geometric objects with varying parameter settings and examined the number of scattering events per path as well as the rendering performance depending on object geometry and optical material density. Figure 8 encodes the number of scattering events occurring per path for different geometries. The corresponding rendering times of ST-CVAE and PT are shown in Figure 9.

The Stanford bunny in the top row of Figure 8 is characterized by a compact shape, with many convex parts. As a result, light can penetrate deeply into the body, yielding long light paths with a large number of scattering events. As expected, the number of scattering events per path increases strongly with increasing optical density in PT (left). ST-CVAE, in contrast, can take advantage of the voluminous convex shape to construct large spheres which efficiently propagate the rays through the medium. Even though also with ST-CVAE the number of scattering events increases with increasing optical density, the rate is much lower. While in the case of the highest density PT requires to simulate up to 3000 scattering events per path, ST-CVAE propagates the rays in less than 50 steps. The need for sequential path evaluations, which cannot be parallelized on the GPU, is thus reduced by a factor of up to 60x, and drastically reduces the time until long sequences of scattering events converge. The rendering times in Figure 9 confirm these findings. Notably, compared to ST-CVAE the time increases at a higher rate when using PT. In the case of highest optical density, a rendering time of about 15 ms for ST-CVAE stands against a rendering time of more than 210 ms with PT, yielding a speed-up by a factor of more than 14x.

When rendering the statue model in the bottom row of Figure 8, however, some limitations of our approach become apparent. Since the statue possesses many thin features, which do not form a voluminous material body, the sphere tracing approach is unable to build large spheres and propagate rays efficiently. Especially in the case of low densities, ST-CVAE requires only slightly less scattering steps than PT. In combination with comparatively costly model evaluations in our approach, the performance gain becomes rather low, yielding a speed-up of less than 2x (see Figure 9). Nevertheless, for larger material densities the number of scattering events per path increases much stronger in PT than in ST-CVAE, so that again a significant performance gain can be achieved, i.e. roughly 5x at the highest density.

From these considerations, we conclude that ST-CVAE can be employed at high efficiency if standard path tracing leads to exceedingly long scattering paths. Thus, another factor limiting the efficiency of our approach is absorption, which has not been considered strongly in previous experiments. Figure 10 explores this effect in detail. For this experiment, we rendered a geometric object with constant material parameters and under fixed lighting conditions, while varying the absorption rate of the medium $(1 - \varphi)$. It

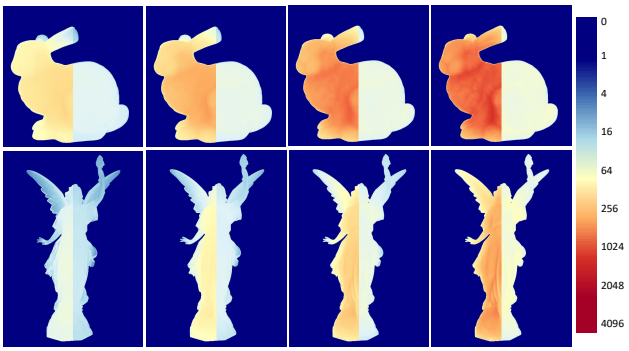


Figure 8: *Number of scattering event simulations per path for different geometries and densities. Standard path tracing (left) vs. our approach, ST-CVAE (right). Densities increase from left to right.*

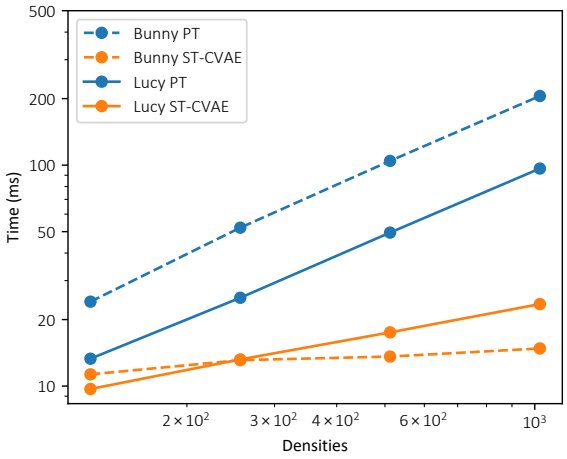


Figure 9: *Time complexity of our approach (ST-CVAE) vs. standard path tracing (PT) for different geometries and optical densities.*

can be seen that higher absorption leads to shorter ray paths and, thus, shortens the rendering time. At higher absorption, the advantage of ST-CVAE decreases.

5.3. Quality of the learned distributions

To evaluate the accuracy of plain model predictions, we conduct the following experiment. For a predefined set of material parameters, $\sigma_t \in \{1,5,25,125\}$, $g \in \{-0.7,0.0,0.4,0.9\}$, $\varphi = 1$, we use standard path tracing to generate a distribution of ground-truth rays, starting in the center of a unit sphere in direction $\omega_{in} = (0,0,1)$ and evaluate the ray statistics $(N,\theta,\alpha,\beta,X,W)$ for each of the rays. The resulting distribution is then compared to the distribution that is obtained from sampling the respective CVAE models. For all experiments, we draw 10000 samples per parameter setting (σ_t, g), for both ground truth and CVAE statistics.

The LENGTHGEN distribution of N samples is compared to the corresponding ground truth in Figure 11. Ground truth statistics

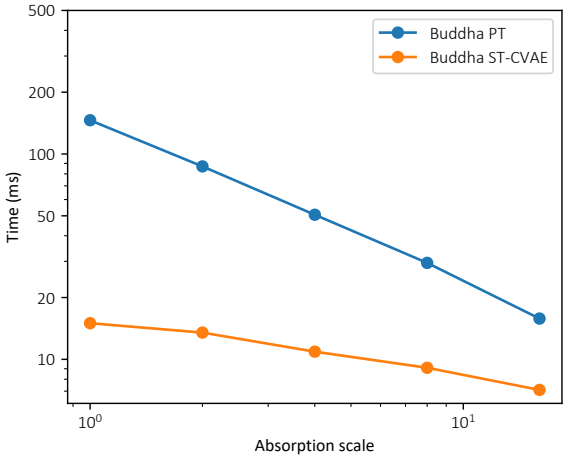


Figure 10: *Performance impact of the absorption parameter. Top: Geometric model rendered with different absorption albedos, using standard path tracing (left half images) and ST-CVAE (right half images). The initial absorption albedo (left-most image) is* $(2\times10^{-3}, 2\times10^{-4}, 2\times10^{-3})$*, and it is doubled in every next image. Bottom: Rendering times for the above model as a function of the absorption parameter.*

are summarized in a log-scale histogram, shown in blue, and overlaid by an orange curve reflecting the statistics obtained from sampling LENGTHGEN. It can be seen that for small values of σ_t (top row), a large fraction of rays leaves the sphere already after the first scattering event, which by assumption occurs in the center of the sphere. This results in a bipartite structure of the histogram, consisting of a pronounced peak of histogram counts around $N = 1$ and a smooth tail indicating rays that are scattered more than once. LENGTHGEN learns to compensate for this and is able to reproduce the peak, while slightly over-estimating the peak amplitude. Nevertheless, the distribution of scattering counts in the case of $N > 1$ is reproduced at reasonable accuracy. For larger values of σ_t, the significance of the peak at $N = 1$ decreases and the ground-truth statistics appear to converge to a unimodal shape, similar to a log-Gaussian. For larger values of σ_t, notably $\sigma_t = 25, 125$, the matching of the distributions is good, and also the dependence on g is captured appropriately. The highest inaccuracy is observed at intermediate scattering coefficients, $\sigma_t = 5$, where the multi-scattering contributions are dominant, but single scattering still has a notable contribution. There, it seems difficult for the model to predict the balance between both effects.The single-scattering contribution is underestimated, which may yield an explanation for the comparatively limited accuracy of our approach in presence of optically thin features (see Figure 7). These observations can be used, however, to bias the sampling statistics of σ_t and g at training time. By increasing the number of samples towards these difficult regions, a

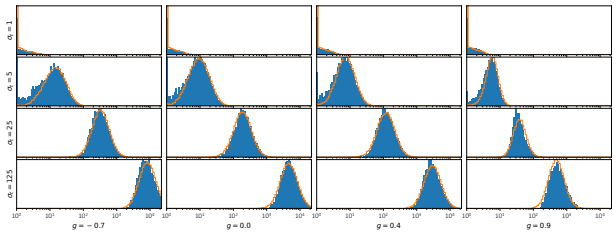


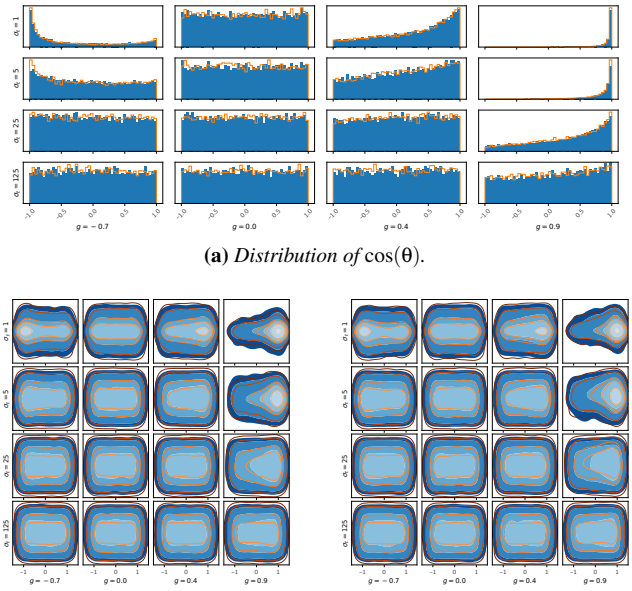
Figure 11: *Distribution of N as obtained from* LENGTHGEN *(orange) vs. ground truth (blue) for different settings of σ_t and g.*

more accurate fit can be achieved. Alternatively, an improved fitting accuracy can be achieved by excluding the single-scattering case from the CVAE statistics, thus simplifying the learning task for the CVAE, and treating single-scattering with a specialized algorithm. The proposed procedure, however, involves branching on the GPU and was found to significantly increase rendering times. The implementation of an efficient splitting approach remains to be studied in future work.

For the PATHGEN model, three parameters need to be validated against ground truth. The sampling statistics of $\cos(\theta)$ are shown in Figure 12a. Again, the blue histogram refers to ground truth statistics, whereas the orange line reflects the PATHGEN predictions. Notably, strongly anisotropic scattering is observed only in cases where σ_t is small or g is close to extreme values -1 or 1. The overall fitting accuracy between learned distributions and ground truth is good. Only for strongly anisotropic cases, e.g., $g = 0.9$ and $g = -0.7$, the model reveals minor difficulties with handling singly-scattered paths that exit in backward direction.

To validate the distribution of α and β, we generate a contour map, which shows the distribution function of α and β as a function of $\cos(\theta)$. Two-dimensional histograms of the distribution of $(\cos(\theta), \alpha)$ and $(\cos(\theta), \beta)$ are obtained and normalized by the number of observations. The resulting bin probability values are finally smoothed with a Gaussian filter to reduce visual clutter. The ground-truth distribution is shown as blue-filled contours, the learned distribution is overlaid as orange iso-contours at probability iso-values matching those of the ground truth. The comparison is shown in Figures 12b and 12c. Similar to the distribution of $\cos(\theta)$, the largest anisotropy is observed for g-values close to extremes, i.e., $g = 0.9$ or $g = -0.7$, and small σ_t. Nevertheless, the learned distributions fit the ground truth at high fidelity.

We conclude that the major challenges for all models lie in learning a reasonable PDF at small values of σ_t and in correctly resolving the dependence of the PDF on g. From a practical perspective, learning results can be improved, however, by specializing the model at training time towards those parameter regions, which are particularly relevant during upcoming rendering processes. By over-weighting ray samples with particular (σ_t, g)-settings at training time, these regions of parameter spaces are assigned a larger weight in the loss function and, thus, affect the outcome of the training process more strongly. If, for example, only media with largely forward-directed scattering are considered during rendering, there is no need to include negative values of g at training time. For our purposes, however, we did not make use of such optimizations.



(a) *Distribution of* $\cos(\theta)$.

(b) *Distribution of* α *(vertical axis) as a function of* $\cos(\theta)$ *(horizontal axis).*

(c) *Distribution of* β *(vertical axis) as a function of* $\cos(\theta)$ *(horizontal axis).*

Figure 12: *Distribution of* PATHGEN *samples for* $\cos(\theta)$, α *and* β *(orange) vs. ground truth (blue) for different settings of σ_t and g.*

5.4. In-scattered direct illumination

To illustrate the use of EVENTGEN, we compare the renderings of a geometric object using all three network models, LENGTHGEN, PATHGEN and EVENTGEN, with those generated with PT, and with the same path tracer with next-event estimation (NEE) using direct illumination. To obtain a fair comparison, the images were rendered using the same time budget of 120 seconds.

In our current implementation of PT and ST-CVAE with direct illumination, both approaches draw a shadow ray to a single light source in the scene setting to account for the direct illumination. As such, refractive object boundaries are not handled accurately, resulting in biased results compared to standard path tracing. Nevertheless, Figure 13 illustrates that ST-CVAE yields a considerably lower variance than standard PT and PT with next-event estimation. This is a promising result and suggests that CVAEs can be employed successfully for learning summary statistics of the bypassed path sections. Furthermore, the performance gain is independent of how refractive boundaries are treated, since approaches enabling an accurate integration of refractions affect our model and the reference path tracer in the same way.

6. Discussion and Conclusion

In this work we have introduced and analyzed a network pipeline comprised of CVAEs that learns to bypass multiple-scattering paths in anisotropic volumetric material. The resulting pipeline can be embedded into a sphere-tracing approach to speed up the performance of Monte Carlo path tracing. For different parameter settings and geometries, we have demonstrated visual quality on par

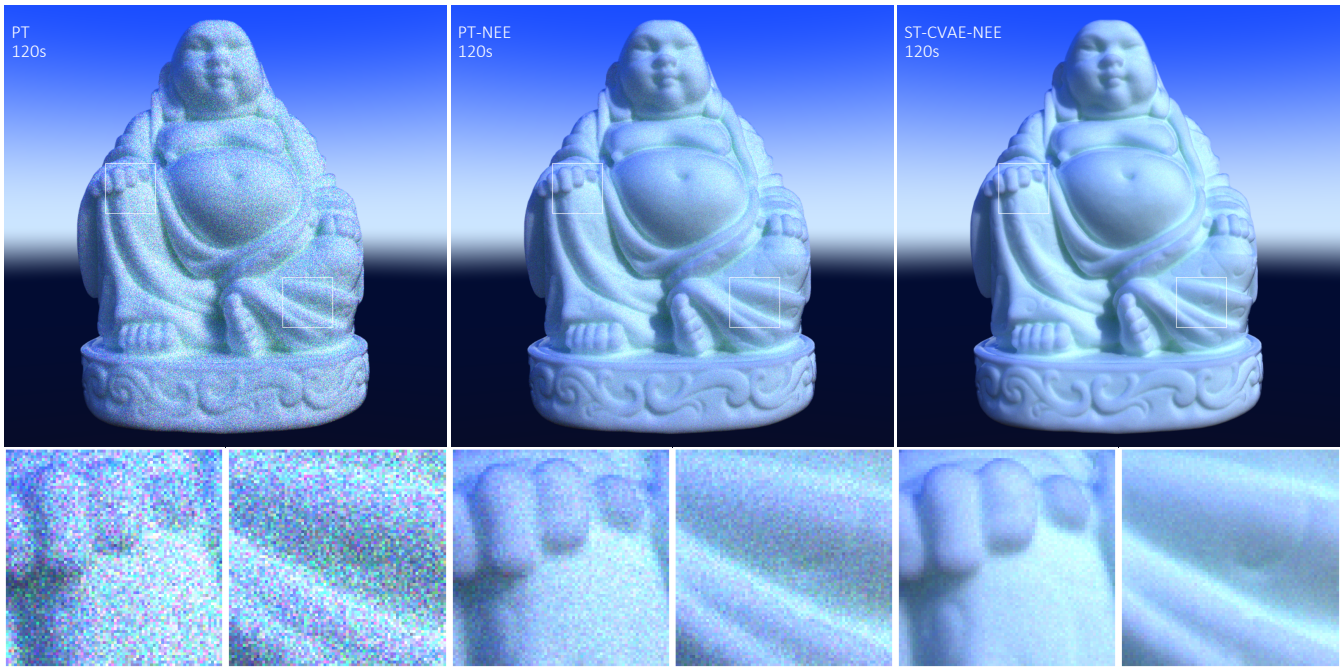


Figure 13: *Convergence improvement using next-event estimation (NEE). All images are rendered in* 120 *s. Left: Standard path tracing (735 paths per pixel), excluding next-event estimation. Middle: Standard path tracing (294 paths per pixel), including next-event estimation using in-scattered direct illumination at every internal sample. Right: ST-CVAE (1900 paths per pixel), including next-event estimation using one representative sample (X,W) per sphere-tracing step.*

with that of ground truth path tracing, as well as performance gains for objects with large volumes of material.

We see the limitations of our method mainly related to the underlying statistical model. As pointed out in Section 3.2, our approach assumes a large-scale homogeneous body and, in NEE-mode, the absence of refractive boundaries. Performance gains are most prominent in the case of dense media with convex shapes. For optically thin objects with fine-scale details, performance gains are rather moderate. This is because the computational gain due to fewer scattering event simulations is weighed out by the increased cost per sphere-tracing step. While the conditions under which our approach can be applied are not perfectly met in many real-world applications, it is often possible to approximately fulfill the requirements by using a multi-scale rendering approach, as described, e.g., in [MPH*15]. Surface-related effects, such as fine-granular surface structures, and refraction effects can be handled with a standard path tracer, while the lighting contribution of photons scattered deeply inside the volume are treated with an accelerated volumetric path tracer. Especially in the case of high scattering albedo, where our model has its strength, fine-scale structures are smeared out by scattering processes right underneath the surface due to inhomogeneity, so that the volume interior can be considered homogeneous.

From an implementation perspective, we see the limited interpretability of the inference steps that happen inside the network. Verification of the fitting accuracy, thus, has to be achieved empirically, which may be difficult given the multi-dimensional parameter spaces on which the models operate. Furthermore, accurate fitting of the models should only be expected within the range of data that the model has seen during training, since the extrapolation behavior of neural networks remains to be understood better. From the viewpoint of model design and training, differences in the hyper-parameters may strongly influence the final performance of the models, so that finding an optimal parameter selection may require a certain amount of empirical experimentation. Nevertheless, once a suitable set of hyper-parameters has been selected, we found the models to train reliably and with reproducible performance.

In the future, we will consider a case-sensitive implementation of the sphere-tracing algorithm to further minimize the overhead in computation time in low-density or thin object parts. For shell tracing, a fall-back strategy to standard path tracing was already used to avoid sphere tracing in the limit of very small radii [MWM07]. Efficiently parallelizing CVAE evaluation and standard path tracing on GPU architectures concurrently, however, appears challenging. Furthermore, the proposed combination of deep learning and sphere tracing opens the possibility to efficiently render volumetric objects like clouds. In this context, we believe that the integration of direct in-scattering can significantly improve performance.

7. Acknowledgements

Open access funding enabled and organized by Projekt DEAL. [Correction added on 17 Jan 2022, after first online publication: Projekt Deal funding statement has been added.]

References

[AKW*18] ARDIZZONE L., KRUSE J., WIRKERT S., RAHNER D., PELLEGRINI E. W., KLESSEN R. S., MAIER-HEIN L., ROTHER C., KÖTHE U.: Analyzing inverse problems with invertible neural networks. *arXiv preprint arXiv:1808.04730* (2018). 5

[ALK*19] ARDIZZONE L., LÜTH C., KRUSE J., ROTHER C., KÖTHE U.: Guided image generation with conditional invertible neural networks. *arXiv preprint arXiv:1907.02392* (2019). 5

[CCY67] COVEYOU R. R., CAIN V., YOST K.: Adjoint and importance in monte carlo application. *Nuclear Science and Engineering 27*, 2 (1967), 219–234. 3, 5

[Cha50] CHANDRASEKHAR S.: Radiative transfer. clarendon, 1950. 4

[CMA*99] CHATIGNY S., MORIN M., ASSELIN D., PAINCHAUD Y., BEAUDRY P.: Hybrid monte carlo for photon transport through optically thick scattering media. *Applied optics 38*, 28 (1999), 6075–6086. 3

[CPP*05] CEREZO E., PÉREZ F., PUEYO X., SERON F. J., SILLION F. X.: A survey on participating media rendering techniques. *The Visual Computer 21*, 5 (2005), 303–328. 2

[d'E13] D'EON E.: A dual-beam method-of-images 3d searchlight bssrdf. *arXiv preprint arXiv:1311.0955* (2013). 3

[dI11] D'EON E., IRVING G.: A quantized-diffusion model for rendering translucent materials. *ACM transactions on graphics (TOG) 30*, 4 (2011), 1–14. 3

[DK19] DEAKIN L., KNACKSTEDT M.: Accelerated volume rendering with chebyshev distance maps. In *SIGGRAPH Asia 2019 Technical Briefs*. 2019, pp. 25–28. 4

[EUVN87] ECKHARDT R., ULAM S., VON NEUMANN J.: the monte carlo method. *Los Alamos Science*, 15 (1987), 131. 4

[FHK14] FRISVAD J. R., HACHISUKA T., KJELDSEN T. K.: Directional dipole model for subsurface scattering. *ACM Transactions on Graphics (TOG) 34*, 1 (2014), 1–12. 3

[FPW92] FARRELL T. J., PATTERSON M. S., WILSON B.: A diffusion theory model of spatially resolved, steady-state diffuse reflectance for the noninvasive determination of tissue optical properties in vivo. *Medical physics 19*, 4 (1992), 879–888. 2

[GPAM*14] GOODFELLOW I., POUGET-ABADIE J., MIRZA M., XU B., WARDE-FARLEY D., OZAIR S., COURVILLE A., BENGIO Y.: Generative adversarial nets. In *Advances in neural information processing systems* (2014), pp. 2672–2680. 5

[Har96] HART J. C.: Sphere tracing: A geometric method for the antialiased ray tracing of implicit surfaces. *The Visual Computer 12*, 10 (1996), 527–545. 2, 3

[HCJ13] HABEL R., CHRISTENSEN P. H., JAROSZ W.: Photon beam diffusion: a hybrid monte carlo method for subsurface scattering. In *Computer Graphics Forum* (2013), vol. 32, Wiley Online Library, pp. 27–37. 3

[HDF15] HANIKA J., DROSKE M., FASCIONE L.: Manifold next event estimation. In *Computer graphics forum* (2015), vol. 34, Wiley Online Library, pp. 87–97. 3, 5

[HEV*16] HERHOLZ S., ELEK O., VORBA J., LENSCH H., KŘIVÁNEK J.: Product importance sampling for light transport path guiding. In *Computer Graphics Forum* (2016), vol. 35, Wiley Online Library, pp. 67–77. 3

[HG41] HENYEY L. G., GREENSTEIN J. L.: Diffuse radiation in the galaxy. *The Astrophysical Journal 93* (1941), 70–83. 4

[HK93] HANRAHAN P., KRUEGER W.: Reflection from layered surfaces due to subsurface scattering. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques* (1993), pp. 165–174. 3

[Hol15] HOLZSCHUCH N.: Accurate computation of single scattering in participating media with refractive boundaries. In *Computer graphics forum* (2015), vol. 34, Wiley Online Library, pp. 48–59. 6

[KB14] KINGMA D. P., BA J.: Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014). 7

[KHLN17] KUTZ P., HABEL R., LI Y. K., NOVÁK J.: Spectral and decomposition tracking for rendering heterogeneous volumes. *ACM Transactions on Graphics (TOG) 36*, 4 (2017), 1–16. 3

[KKR18] KUZNETSOV A., KALANTARI N. K., RAMAMOORTHI R.: Deep adaptive sampling for low sample count rendering. In *Computer Graphics Forum* (2018), vol. 37, Wiley Online Library, pp. 35–44. 3

[KMM*17] KALLWEIT S., MÜLLER T., MCWILLIAMS B., GROSS M., NOVÁK J.: Deep scattering: Rendering atmospheric clouds with radiance-predicting neural networks. *ACM Transactions on Graphics (TOG) 36*, 6 (2017), 1–11. 2, 3

[KNK*16] KOERNER D., NOVÁK J., KUTZ P., HABEL R., JAROSZ W.: Subdivision next-event estimation for path-traced subsurface scattering. In *EGSR (EI&I)* (2016), pp. 91–96. 3, 5, 6

[KVH84] KAJIYA J. T., VON HERZEN B. P.: Ray tracing volume densities. *ACM SIGGRAPH computer graphics 18*, 3 (1984), 165–174. 2, 3, 4

[KW13] KINGMA D. P., WELLING M.: Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013). 5

[Lam60] LAMBERT J. H.: *Photometria sive de mensura et gradibus luminis, colorum et umbrae*. Klett, 1760. 4

[LHW21] LEONARD L., HÖHLEIN K., WESTERMANN R.: Learning Multiple-Scattering Solutions for Sphere-Tracing of Volumetric Subsurface Effects, 2021. URL: https://doi.org/10.5281/zenodo.4555744, doi:10.5281/zenodo.4555744. 2

[LO07] LEE R., O'SULLIVAN C.: Accelerated light propagation through participating media. In *Volume Graphics* (2007), pp. 17–23. 3

[LW96] LAFORTUNE E. P., WILLEMS Y. D.: Rendering participating media with bidirectional path tracing. In *Rendering Techniques '96* (Vienna, 1996), Pueyo X., Schröder P., (Eds.), Springer Vienna, pp. 91–100. 3, 4

[MMBJ17] MARA M., MCGUIRE M., BITTERLI B., JAROSZ W.: An efficient denoising algorithm for global illumination. In *Proceedings of High Performance Graphics* (New York, NY, USA, jul 2017), ACM. 3

[MO14] MIRZA M., OSINDERO S.: Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784* (2014). 5

[MPG*16] MÜLLER T., PAPAS M., GROSS M., JAROSZ W., NOVÁK J.: Efficient rendering of heterogeneous polydisperse granular media. *ACM Transactions on Graphics (TOG) 35*, 6 (2016), 1–14. 3, 7, 8

[MPH*15] MENG J., PAPAS M., HABEL R., DACHSBACHER C., MARSCHNER S., GROSS M. H., JAROSZ W.: Multi-scale modeling and rendering of granular materials. *ACM Trans. Graph. 34*, 4 (2015), 49–1. 3, 12

[MWM07] MOON J. T., WALTER B., MARSCHNER S. R.: Rendering discrete random media using precomputed scattering solutions. In *Proceedings of the 18th Eurographics conference on Rendering Techniques* (2007), pp. 231–242. 2, 3, 12

[NAM*17] NALBACH O., ARABADZHIYSKA E., MEHTA D., SEIDEL H.-P., RITSCHEL T.: Deep shading: convolutional neural networks for screen space shading. In *Computer graphics forum* (2017), vol. 36, Wiley Online Library, pp. 65–78. 3

[NGHJ18] NOVÁK J., GEORGIEV I., HANIKA J., JAROSZ W.: Monte carlo methods for volumetric light transport simulation. In *Computer Graphics Forum* (2018), vol. 37, Wiley Online Library, pp. 551–576. 3, 4, 5

[NNDJ12] NOVÁK J., NOWROUZEZAHRAI D., DACHSBACHER C., JAROSZ W.: Virtual ray lights for rendering scenes with participating media. *ACM Transactions on Graphics (TOG) 31*, 4 (2012), 1–11. 5

[NSJ14] NOVÁK J., SELLE A., JAROSZ W.: Residual ratio tracking for estimating attenuation in participating media. *ACM Transactions on Graphics (TOG) 33*, 6 (2014), 179–1. 3

[NVI18] NVIDIA: Nvidia rtx ray tracing developer documentation. developer.nvidia.com/rtx/raytracing, 2018. [Online; accessed 21-March-2019]. 4

[RACSKS*17] R. ALLA CHAITANYA C., S. KAPLANYAN A., SCHIED C., SALVI M., LEFOHN A., NOWROUZEZAHRAI D., AILA T.: Interactive reconstruction of monte carlo image sequences using a recurrent denoising autoencoder. *ACM Transactions on Graphics 36* (07 2017), 1–12. 3

[RKDS19] REHAK J., KERBY L., DEHART M., SLAYBAUGH R.: Weighted delta-tracking in scattering media. *Nuclear Engineering and Design 342* (2019), 231–239. 3

[Rus89] RUSHMEIER H. E.: *Realistic image synthesis for scenes with radiatively participating media.* PhD thesis, 1989. 3

[SLY15] SOHN K., LEE H., YAN X.: Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems 28*, Cortes C., Lawrence N. D., Lee D. D., Sugiyama M., Garnett R., (Eds.). Curran Associates, Inc., 2015, pp. 3483–3491. 5

[SRNN05] SUN B., RAMAMOORTHI R., NARASIMHAN S. G., NAYAR S. K.: A practical analytic single scattering model for real time rendering. *ACM Transactions on Graphics (TOG) 24*, 3 (2005), 1040–1049. 2

[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* (1995), pp. 419–428. 3

[VKJ19] VICINI D., KOLTUN V., JAKOB W.: A learned shape-adaptive subsurface scattering model. *ACM Transactions on Graphics (TOG) 38*, 4 (2019), 1–15. 2, 3, 4

[VKŠ*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 (TOG) 33*, 4 (2014), 1–11. 3

[WITW20] WEISS S., IŞIK M., THIES J., WESTERMANN R.: Learning adaptive sampling and reconstruction for volume visualization. *arXiv preprint arXiv:2007.10093* (2020). 3

[WMHL65] WOODCOCK E., MURPHY T., HEMMINGS P., LONGWORTH S.: Techniques used in the gem code for monte carlo neutronics calculations in reactors and other systems of complex geometry. In *Proc. Conf. Applications of Computing Methods to Reactor Problems* (1965), vol. 557. 3

[WZHB09] WALTER B., ZHAO S., HOLZSCHUCH N., BALA K.: Single scattering in refractive media with triangle mesh boundaries. In *ACM SIGGRAPH 2009 papers*. 2009, pp. 1–8. 6

[YIC*10] YUE Y., IWASAKI K., CHEN B.-Y., DOBASHI Y., NISHITA T.: Unbiased, adaptive stochastic sampling for rendering inhomogeneous participating media. *ACM Transactions on Graphics (TOG) 29*, 6 (2010), 1–8. 3

[YZXW12] YAN L.-Q., ZHOU Y., XU K., WANG R.: Accurate translucent material rendering under spherical gaussian lights. In *Computer Graphics Forum* (2012), vol. 31, Wiley Online Library, pp. 2267–2276. 3

[ZRB14] ZHAO S., RAMAMOORTHI R., BALA K.: High-order similarity relations in radiative transfer. *ACM Transactions on Graphics (TOG) 33*, 4 (2014), 1–12. 3

[ZYL*15] ZHENG H., YANG Z., LIU W., LIANG J., LI Y.: Improving deep neural networks using softplus units. In *2015 International Joint Conference on Neural Networks (IJCNN)* (2015), IEEE, pp. 1–4. 7

[ZZ19] ZHENG Q., ZWICKER M.: Learning to importance sample in primary sample space. *Computer Graphics Forum 38* (05 2019), 169–179. 3