

A Non-Exponential Transmittance Model for Volumetric Scene Representations

DELIO VICINI, Facebook Reality Labs and École Polytechnique Fédérale de Lausanne (EPFL), Switzerland

WENZEL JAKOB, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland

ANTON KAPLANYAN, Facebook Reality Labs, USA

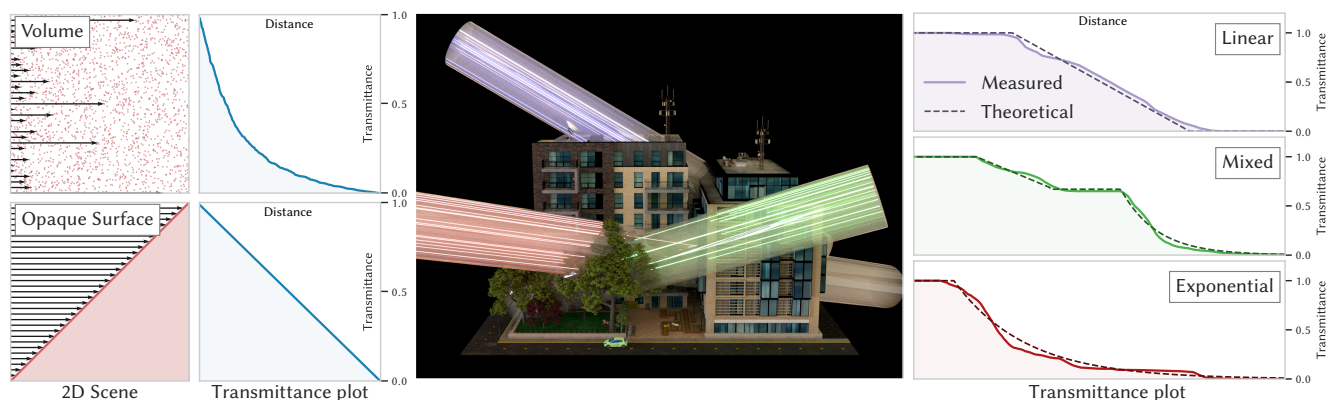


Fig. 1. *Left:* Consider the fraction of unoccluded rays in two flatland scenarios involving particles generated by a Poisson process (top) and an opaque surface (bottom). The former leads to an exponential decay that is the standard model underlying most existing volume rendering techniques. We consider a generalized notion of transmittance that also supports the unusual linear decay profile exhibited by the surface case. *Right:* We perform a similar experiment in a more complex scene by tracing rays within thick beams (middle) and tracking their free-flight distance. The distributions resulting from the three beams are plotted on the right. Hard surfaces induce linear transmittance (purple), while unstructured geometry like foliage resembles an uncorrelated medium that yields exponential transmittance (red). A beam that first traverses the trees and then a hard surface (green) encounters both linear and exponential transmittance. We also show a parametric fit (dotted plots) using either an exponential or linear model, or piecewise combination of the two in case of the mixed example.

We introduce a novel transmittance model to improve the volumetric representation of 3D scenes. The model can represent opaque surfaces in the volumetric light transport framework. Volumetric representations are useful for complex scenes, and become increasingly popular for level of detail and scene reconstruction. The traditional exponential transmittance model found in volumetric light transport cannot capture correlations in visibility across volume elements. When representing opaque surfaces as volumetric density, this leads to both bloating of silhouettes and light leaking artifacts. By introducing a parametric non-exponential transmittance model, we are able to approximate these correlation effects and significantly improve the accuracy of volumetric appearance representation of opaque scenes. Our parametric transmittance model can represent a continuum between the linear transmittance that opaque surfaces exhibit and the traditional exponential transmittance encountered in participating media and unstructured geometries. This covers a large part of the spectrum of geometric structures encountered in complex scenes. In order to handle the spatially varying transmittance correlation effects, we further extend the theory of non-exponential

participating media to a heterogeneous transmittance model. Our model is compact in storage and computationally efficient both for evaluation and for reverse-mode gradient computation. Applying our model to optimization algorithms yields significant improvements in volumetric scene appearance quality. We further show improvements for relevant applications, such as scene appearance prefiltering, image-based scene reconstruction using differentiable rendering, neural representations, and compare it to a conventional exponential model.

CCS Concepts: • **Computing methodologies** → **Ray tracing; Visibility; Antialiasing; Volumetric models.**

Additional Key Words and Phrases: level of detail, volume rendering, non-exponential media, transmittance, ray marching, differentiable rendering

ACM Reference Format:

Delio Vicini, Wenzel Jakob, and Anton Kaplanyan. 2021. A Non-Exponential Transmittance Model for Volumetric Scene Representations. *ACM Trans. Graph.* 40, 4, Article 136 (August 2021), 16 pages. <https://doi.org/10.1145/3450626.3459815>

Authors' addresses: Delio Vicini, delio.vicini@epfl.ch, Facebook Reality Labs and École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland; Wenzel Jakob, wenzel.jakob@epfl.ch, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland; Anton Kaplanyan, kaplanyan@gmail.com, Facebook Reality Labs, Redmond, USA.

© 2021 Copyright held by the owner/author(s). Publication rights licensed to ACM. This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in *ACM Transactions on Graphics*, <https://doi.org/10.1145/3450626.3459815>.

1 INTRODUCTION

Photorealistic representations of digital scenes require significant detail in both geometry and textures and can easily exceed the allotted storage or rendering budget. When the scene is observed from different positions, the level of detail may be grossly inappropriate, with an entire forest or city block visible in a small image region. *Level of detail* (LoD) techniques are therefore essential [Luebke et al. 2002] and widely used to represent complex photorealistic scenes in production environments [Schwank et al. 2016]. This is especially important for mobile and wearable augmented and virtual reality (AR/VR) devices, where immersion is paramount and resource limits place stringent requirements on rendering efficiency [Kaplanyan et al. 2019].

On the other hand, scalable scene representations that are purely based on decimating triangle meshes cannot efficiently filter the appearance of fine structures, such as tree leaves, where accurately modeling of partial visibility is required. Ultimately, at farther distances, any geometric structure only achieves partial coverage at the sub-pixel level and can be more efficiently approximated using a *volumetric representation* [Loubet and Neyret 2017]. This approach has also become commonplace for modeling captured scenes [Lombardi et al. 2019; Mildenhall et al. 2020].

However, one challenge is that the volumetric light transport theory behind this representation is originally derived to model absorption and scattering of a medium consisting of identically distributed and uncorrelated microscopic particles, but not opaque surfaces. The assumption of uncorrelated particles results in an *exponential* transmittance function. This exponential transmittance model is well suited for unstructured content such as the leaves of a tree, but it breaks down for structured and opaque geometry where the assumption of uncorrelated particles is violated. A distinguishing property of surfaces is their ability to be fully *opaque*, i.e., impenetrable to light, which is not possible with classic exponential volumetric light transport theory. When an opaque object is modeled as an exponential volume, a low density leads to a significant portion of light leaking through the surface. Increasing the volume density to compensate leads to bloated silhouettes and an overly opaque appearance of semitransparent parts of the model. Volumetric modeling of scenes containing both opaque and semitransparent elements remains a fundamental challenge in current volumetric modeling approaches.

In this paper, we build on the observation that approximating the transmittance behavior of an arbitrary scene requires a *non-exponential* transmittance model, as demonstrated in Figure 1. We build on the recent advances in correlated and non-exponential light transport theory [Bitterli et al. 2018; Jarabo et al. 2018] to improve the volumetric representation of 3D scenes. Our novel transmittance model captures the wide spectrum of transport behaviors induced by geometric configurations ranging from completely opaque surfaces to unstructured geometric aggregates. This allows to represent the whole scene in a single volumetric form while efficiently handling all scene elements within a unified light transport framework. We demonstrate this efficiency by achieving state-of-the-art results in multiple applications, such as appearance prefiltering (Figure 2). By using a unified volumetric framework to model the entirety of the

scattering in a scene, we can avoid the problem of separating the scene into partitions modeled using volumetric or surface scattering. A purely volumetric representation can also simplify the rendering algorithm.

Volumetric scene representations have recently gained significant momentum for inverse rendering applications [Lombardi et al. 2019; Mildenhall et al. 2020]. A differentiable renderer can be used to reconstruct a real scene by optimizing a volumetric scene representation to match reference photographs. Using a volumetric representation results in a more convex optimization problem than directly optimizing surface geometries. The resulting volume parameters can be easily stored in a neural network or a simple (uniform) grid. The volumetric scene representation is smooth and, in contrast to surface rendering, does not require any special treatment of visibility derivatives [Li et al. 2018].

We show that our non-exponential transmittance formulation can be useful for various tasks that require efficient volumetric representation of an opaque scene, including, prefiltering for level of detail, scene reconstruction using differentiable rendering, and neural rendering [Mildenhall et al. 2020]. In summary, our core contributions are:

- a unified volumetric representation that handles both opaque surfaces as well as aggregate geometries within a single volumetric light transport theory
- a new practical parametric model for heterogeneous non-exponential transmittance to account for correlations in a continuum from opaque to aggregate geometric configurations
- a new scene appearance prefiltering method based on our unified volumetric representation and a robust scene-scale parameter optimization for appearance prefiltering
- efficient image-based volumetric reconstruction of complex scenes using differentiable and neural rendering

In the remainder of the paper, we will first introduce our volumetric representation and transmittance model. We then demonstrate and discuss the application of our model to appearance prefiltering and image-based reconstruction.

2 RELATED WORK

Volumetric light transport is described by the *radiative transfer equation* (RTE) [Chandrasekhar 1960], which relates physical volume parameters to the scattering of radiance. The RTE is typically solved by using Monte Carlo integration. We refer to the state-of-the-art report by Novák et al. [2018] for a comprehensive review of the vast number of recent volume rendering techniques. In the following we will discuss related work on level of detail, non-exponential media, microflake theory and inverse rendering.

Volumetric appearance prefiltering for level of detail. Reducing the scene complexity by converting opaque surfaces into a volumetric representation has been a long standing and attracting direction in the field of level of detail [Crassin et al. 2011]. It is easy to filter a scene into a volumetric representation with a hierarchy of scales (e.g., an octree). In its hierarchical form, it allows to efficiently bound the per-pixel rendering complexity by choosing the

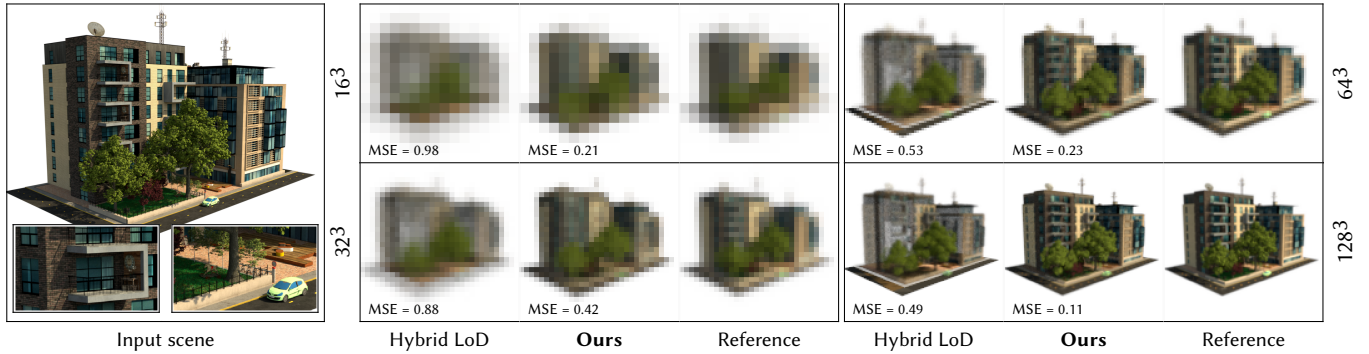


Fig. 2. Prefiltered rendering of a complex scene (13.8 million triangles) at different resolutions. In each rendering, the original scene is represented as a volume with a voxel grid resolution corresponding to the image resolution. Our method reproduces the appearance of the ground truth reference, while the state-of-the-art *Hybrid LoD* [Loubet and Neyret 2017] method has difficulties modeling the scene’s transmittance function leading to less accurate results. The original scene consists of 750MB of geometry data and 350MB of textures. Our volumetric representation compresses this down to 4.9MB (220x compression) for a resolution of 128³, and even further at lower resolutions. All results are rendered using global illumination.

prefiltered scale proportionally to the pixel’s footprint. The rendering of a volumetric representations can be accelerated by using sparse data structures [Crassin et al. 2009] and empty space skipping [Hadwiger et al. 2018; Morrical et al. 2019]. Beyer et al. [2015] provide an overview of efficient GPU volume rendering techniques used in scientific visualization.

The main challenge with representing surfaces as participating media is in preserving their opaqueness as well as thin structures. This makes volumetric representations suffer either from bloated appearance (density is too high) or light leaking (density is too low), often leading to both at different viewing directions. In order to reduce this bloating and light leaking, Heitz et al. [2012] used a per-voxel, implicit plane to compute a view-dependent coverage mask. Volumetric models have also been used to prefilter the appearance of granular media such as snow and sand [Meng et al. 2015; Moon et al. 2007; Müller et al. 2016].

The state-of-the-art method for general, appearance-preserving LoD is the hybrid approach proposed by Loubet and Neyret [2017]. It performs a heterogeneous simplification of a scene by labelling parts of a surface mesh to use either a geometric or a voxelized volumetric representation at every rendering scale. While their method can achieve good results for both opaque geometry (e.g., a tree trunk) and aggregate geometry (e.g., tree leaves), this binary classification is an ill-posed problem, as there is a smooth transient phase between surface-like and volumetric appearance. Moreover, their binary classification algorithm is based on a heuristic considering local mesh topology, and can misclassify complex, unstructured geometry, resulting in opaque surfaces appearing semitransparent, as shown in Figure 2.

Non-exponential transport. Accounting for visibility correlation across particles or, as in our case, geometry is a challenging problem and has only recently been investigated explicitly in computer graphics. Bitterli et al. [2018] introduce a reciprocal path integral formulation to allow the use of non-exponential transmittance functions in participating media with correlated particles. Concurrent work by Jarabo et al. [2018] extends the non-exponential generalized Boltzmann equation (GBE) to be used in a rendering algorithm,

enabling to handle boundary surfaces. D’Eon [2018] introduced a weakly reciprocal non-exponential rendering formulation and investigated diffusion approximations [d’Eon 2019] and binary mixtures of scatterers [d’Eon 2019]. Several previous works [Bitterli et al. 2018; Davis and Mineev-Weinstein 2011; Guo et al. 2019; Jarabo et al. 2018] have introduced parametric families of non-exponential transmittance functions. These parametric models mostly cover the space of transmittance functions which fall off less quickly than the exponential function (e.g., due to clumping of medium particles). However, it is essential for our method to cover the spectrum between linear and exponential transmittance, which is not covered by these models. In this paper, we introduce an integral transmittance framework, that allows to cover the spectrum between exponential and linear transmittance behavior. We apply it to the problem of representing opaque surfaces as participating media.

Microflake theory. The appearance of a participating medium is not only determined by the transmittance, but also by the angular distribution of scattering, described by the *phase function*. Jakob et al. [2010] generalized the radiative transfer equation to introduce anisotropic *microflake* phase functions, which are the volumetric analog of microfacet BSDFs used in surface rendering. The microflake model is widely used for items such as woven fabric, since it very accurately preserves such appearances at even very fine scales [Zhao et al. 2012]. Heitz et al. [2015] introduced the SGGX microflake distribution, which enables practical and compact microflake representation of surfaces for appearance prefiltering.

Inverse and neural rendering. Volumetric representations have been popular for reconstructing 3D scenes from reference images. Gkioulekas et al. [2016] used image-based acquisition to reconstruct volumes (e.g., smoke), while accounting for multiple scattering. Zhao et al. [2016] applied differentiable volume rendering to the problem of downsampling volume parameters. Recently, Lombardi et al. [2019] used a volumetric representation to model facial performances. Volumetric representations have also recently been used to allow re-rendering under novel illumination conditions [Bi et al. 2020a,b]. Recent progress on differentiable rendering enables

brute-force gradient descent optimization of heterogeneous volumes using differentiable path tracing [Nimier-David et al. 2019]. A differentiable rendering approach has also been proposed to optimize the transmittance of a 3D printed light field display [Zheng et al. 2020]. Lately, neural radiance fields (NeRF) [Mildenhall et al. 2020] introduced a seminal compact volumetric scene representation represented by a neural network. The method reconstructs volumetric density and appearance from a set of reference views using classical participating media representation with exponential transmittance model. There have been multiple follow up works that utilize this representation for various tasks, such as relighting. In our work, we also demonstrate how our new transmittance model can be used with gradient descent to optimize the volumetric scene representation.

3 BACKGROUND

3.1 Transmittance

Our work focuses on representing arbitrary scenes in the volumetric rendering framework by introducing a new transmittance model. In the following, we recapitulate the definition of the transmittance and how it is connected to volumetric scene representation.

The transmittance function $T(\mathbf{x}, \mathbf{y})$ of a participating medium describes the *fractional visibility* between two points \mathbf{x} and \mathbf{y} . In the most general case, the transmittance function maps two 3D points to a scalar. By definition, it is non-increasing, attaining values in the interval $[0, 1]$, and reciprocal, i.e., $T(\mathbf{x}, \mathbf{y}) = T(\mathbf{y}, \mathbf{x})$. If the medium is made up of microscopic, uncorrelated particles, the transmittance follows the Beer-Lambert law and can be expressed as:

$$T(\mathbf{x}, \mathbf{y}) = \exp\left(-\int_0^{\|\mathbf{x}-\mathbf{y}\|} \sigma_t(t) dt\right), \quad (1)$$

where σ_t is the *extinction* coefficient of the medium. The extinction coefficient is the differential probability of interacting with a medium particle. We define $\sigma_t(t)$ as a shorthand notation for $\sigma_t\left(\mathbf{x} + \frac{\mathbf{y}-\mathbf{x}}{\|\mathbf{y}-\mathbf{x}\|}t\right)$, i.e., the spatially varying extinction coefficient evaluated at a distance t on the segment between \mathbf{x} and \mathbf{y} . The inner integral over extinction coefficients is also called the *optical depth* τ .

Non-exponential transmittance functions have recently been introduced [Bitterli et al. 2018; Jarabo et al. 2018] to handle more general cases of participating media, such as crystals, fabric, or water droplets formed in clouds, where microparticles have some structure and correlation. In particular, Jarabo et al. [2018] discuss how particles aligned in certain grid structures give rise to a linear transmittance function and show example renderings using homogeneous media. When representing a complex opaque 3D scene as a volumetric representation, transmittance can generally take one of two main modes, as shown in Figure 1: classic exponential mode (e.g., with unstructured geometry, like leaves), as well as linear mode (e.g., when hitting a planar opaque surface). In different parts of the scene, transmittance may vary continuously between these two extremes. Therefore, we build on the prior work on non-exponential media and extend it to non-homogeneous transmittance, where the transmittance behavior can change from one region of the medium to another, in order to handle correlations that occur in opaque 3D scenes.

3.2 Volumetric Light Transport

To render images with volumetric light scattering, the transmittance function needs to be inserted into the volume rendering equation, which is derived from the radiative transfer equation (RTE) [Chandrasekhar 1960]. The standard RTE models the scattering due to uncorrelated, exponential media. In the following, we only consider non-emissive volumes. Assuming that the next surface or volume boundary is at a distance z from the current location in the medium, the outgoing radiance at a location in the volume can be expressed as an integral over distance traveled in the medium:

$$L_o(\mathbf{x}, \boldsymbol{\omega}_o) = \int_0^z \alpha(\mathbf{x}_t) \sigma_t(\mathbf{x}_t) T(\mathbf{x}, \mathbf{x}_t) L_s(\mathbf{x}_t, \boldsymbol{\omega}_o) dt + T(\mathbf{x}, \mathbf{x}_z) L_o(\mathbf{x}_z, \boldsymbol{\omega}_o), \quad (2)$$

where T is the transmittance function, α is the medium's albedo and L_s is the inscattered radiance modulated by the phase function f_p ,

$$L_s(\mathbf{x}, \boldsymbol{\omega}_o) = \int_{S^2} f_p(\mathbf{x}_t, \boldsymbol{\omega}_o, \boldsymbol{\omega}_i) L_i(\mathbf{x}_t, \boldsymbol{\omega}_i) d\boldsymbol{\omega}_i. \quad (3)$$

To render images using path tracing, we construct light paths by alternating between sampling the distance to the next scattering event, the *free-flight distance*, and the direction of the ray after scattering. In an exponential medium, the free-flight distance is sampled proportionally to $\sigma_t(\mathbf{x}_t)T(\mathbf{x}, \mathbf{x}_t)$. The scattering direction can be sampled according to the phase function.

3.3 Non-Exponential Transport

The previously described volume rendering equation assumes the transmittance function to be exponential. One key observation made in previous work [Bitterli et al. 2018; d'Eon 2018; Jarabo et al. 2018] is that simply replacing T by an arbitrary function is not energy preserving. The solution to that issue lies in realizing that the term $\sigma_t(\mathbf{x}_t)T(\mathbf{x}, \mathbf{x}_t)$ inside the integral in Equation 2 is exactly the free-flight distance probability density function of the exponential medium. Physically, the transmittance is the probability of a photon traversing a region of space without collision. The free-flight distance PDF can therefore be obtained as the negative derivative of the transmittance T . That T is a factor in that term is simply due to it being an exponential function. Moving to a non-exponential transmittance, this whole term has to be replaced by the associated free-flight distance PDF, as opposed to merely replacing the function T in the integral. Denoting the free-flight distance PDF by T_{pdf} , we then obtain the following volume rendering equation:

$$L_o(\mathbf{x}, \boldsymbol{\omega}_o) = \int_0^z \alpha(\mathbf{x}_t) T_{\text{pdf}}(\mathbf{x}, \mathbf{x}_t) L_s(\mathbf{x}_t, \boldsymbol{\omega}_o) dt + T(\mathbf{x}, \mathbf{x}_z) L_o(\mathbf{x}_z, \boldsymbol{\omega}_o). \quad (4)$$

This simple formulation of the rendering equation has one caveat: If we want the light transport to be reciprocal, the transmittance function and free-flight distance distribution need to be different if the path starts on a surface or medium boundary [Bitterli et al. 2018; d'Eon 2018]. Our non-exponential transmittance formulation is aimed at allowing scenes to be described as a volume in a unified way. Therefore, for the following we assume this simplified, non-reciprocal model.

4 HETEROGENEOUS NON-EXPONENTIAL TRANSMITTANCE

4.1 Transmittance Model

In this section, we present our novel heterogeneous non-exponential transmittance formulation. In previous work on non-exponential media, the transmittance function is assumed to be parametric on the extinction $\sigma_t(t)$ and is fixed throughout the medium. The heterogeneity model proposed by Bitterli et al. [2018], similar to Camminady et al. [2017], expresses the transmittance in a heterogeneous medium as

$$T(\mathbf{x}, \mathbf{y}) = f(\tau) = f\left(\int_0^{\|\mathbf{x}-\mathbf{y}\|} \sigma_t(t) dt\right), \quad (5)$$

where f is the transmittance function, the extinction σ_t is a spatially varying function evaluated at distance t along the segment between \mathbf{x} and \mathbf{y} ; and τ is the optical depth.

This formulation, while being limited, has the advantage that it naturally fits into a reciprocal rendering framework and for some special transmittance functions even allows for unbiased estimation of the transmittance. However, as a limitation, it does not allow to continuously vary the transmittance behavior inside of a medium. This is necessary if we want to be able to capture different transmittance modes present in a single scene. Note that subdividing the scene into homogeneous regions with different non-exponential transmittances would not solve this problem, as this would be restricted to using a voxel grid representation with nearest neighbor interpolation. Jarabo et al. [2018] discuss handling heterogeneous volumes in a such a way, but also acknowledge its limitations and do not provide a practical algorithm.

Our goal in the following is to replace the transmittance function f by a spatially varying function. For scene representation, we are interested in the spectrum between exponential and linear transmittance. Conceptually, exponential transmittance is the consequence of the participating medium consisting of microscopic particles with uncorrelated locations. Previous work [Caglioti and Golse 2003; Jarabo et al. 2018] showed that a linear transmittance can theoretically be achieved by particles placed in a regular grid or crystal-like pattern. Such models could potentially be useful to form a direct connection from the scene's geometry to medium parameters. However, it seems to be difficult to smoothly transition from a linear to an exponential transmittance using such theoretical scatterer distributions. Further, recent progress in differentiable rendering makes it appealing to simply fit a parametric model by minimizing a loss function. Therefore, we focus on building a parametric model that can represent the desired appearance space. We then later show how its parameters can be optimized efficiently. Since we need to capture the full spectrum from linear to exponential transmittance, we base our model on a linear combination of the two. We define a transmittance function

$$f(\tau, \gamma) = \gamma \exp(-\tau) + (1 - \gamma) \max(0, 1 - \tau/2), \quad (6)$$

where γ varies between 0 and 1. We will refer to this parameter as *transmittance mode*. The division by two in the linear transmittance ensures that exponential and linear transmittance have the same mean free path.

Given this transmittance function, we need to be able to vary γ spatially. In the following, we present a novel transmittance framework that enables this. The derivation of the framework is independent of the concrete form of f . We construct our model such that it satisfies the following requirements:

- (1) The formulation needs to be an extension of the traditional exponential transmittance and previous heterogeneous non-exponential transmittance formulations
- (2) The transmittance function needs to be non-increasing
- (3) The transmittance function should be continuous for continuously varying parameters
- (4) Evaluating the transmittance should have similar memory and computational requirements as conventional heterogeneous media

With these requirements, we do not make any explicit assumptions about the correlation throughout the medium. We also do not strictly force the representation to be reciprocal. Our goal is to merely introduce the additional degree of freedom to spatially vary the transmittance behavior f . In order to achieve this, we write the transmittance as a recursive integral over transmittance function derivatives:

$$T(\mathbf{x}, \mathbf{y}) = 1 + \int_0^{\|\mathbf{x}-\mathbf{y}\|} \frac{\partial f}{\partial \tau} \left(f^{-1}(T(\mathbf{x}, \mathbf{x}_t), \gamma(t)), \gamma(t) \right) \sigma_t(t) dt \quad (7)$$

This formulation satisfies requirements (1) – (4), as we explain in the rest of this section and in Section 4.2. In particular, to show that requirement (1) holds, and to provide justification for our formulation, we first consider a constant transmittance mode, i.e., $f(\tau, \gamma) = f(\tau)$. In that case, we can directly derive our formulation from the previous model (Equation 5):

$$\begin{aligned} T(\mathbf{x}, \mathbf{y}) &= f\left(\int_0^{\|\mathbf{x}-\mathbf{y}\|} \sigma_t(t) dt\right) \\ &= 1 + \int_0^{\|\mathbf{x}-\mathbf{y}\|} \frac{\partial}{\partial t} \left[f\left(\int_0^t \sigma_t(s) ds\right) \right] dt \\ &= 1 + \int_0^{\|\mathbf{x}-\mathbf{y}\|} \frac{\partial f}{\partial \tau} \left(\int_0^t \sigma_t(s) ds \right) \sigma_t(t) dt \\ &= 1 + \int_0^{\|\mathbf{x}-\mathbf{y}\|} \frac{\partial f}{\partial \tau} \left(f^{-1}(T(\mathbf{x}, \mathbf{x}_t)) \right) \sigma_t(t) dt. \end{aligned}$$

The first step uses the fundamental theorem of calculus, the second step applies the chain rule, and in the third step we substitute the optical depth integral using the original transmittance definition, i.e., $\int_0^t \sigma_t(s) ds = f^{-1}(T(\mathbf{x}, \mathbf{x}_t))$. Georgiev et al. [2019] used a similar integral formulation to derive new transmittance estimators for exponential light transport. Our model is closely related to that formulation, see the supplemental material for details. Intuitively, our formulation allows to "stitch" together different transmittance behaviors in a way that ensures continuity of the transmittance with respect to both extinction and transmittance mode parameters. Note that our formulation is still bounded between 0 and 1, since the derivative of the non-increasing function f is negative, and has to be zero once $T(\mathbf{x}, \mathbf{x}_t)$ reaches zero.

While our model satisfies requirements (1) – (4) and is straightforward to implement, its simple form has the downside that it does not explicitly track correlation across voxels. Just modifying the transmittance behavior per voxel is still an approximation of the ground truth transmittance function. The only "state" that we carry along a ray is the transmittance up to the current location. As a consequence of this approximation, our model ends up being non-reciprocal. A reciprocal model most likely would need to track some form of correlation between encountered voxels to more faithfully represent the true transmittance function. Combining reciprocity and heterogeneous transmittance behavior remains an important avenue for future work. Since we use our model in conjunction with unidirectional path tracing, we did not find non-reciprocity to be a noticeable issue for any of our experiments. We provide an evaluation of the impact of non-reciprocity in Figure 9 in Section 7.

4.2 Evaluation and Sampling

In order to render with this generalized model, a renderer needs a way to efficiently evaluate and sample the transmittance function [Novák et al. 2018]. We use ray marching to estimate transmittance and sample free-flight distances, but exploring unbiased estimators is an interesting avenue for future work.

When using ray marching, we approximate the transmittance integral using a quadrature based on a set of evenly spaced locations along the ray. We found that naively approximating the integral from Equation 7 using quadrature does not work well. Our model was derived from the original transmittance formulation using the fundamental theorem of calculus. If we want this relation to hold when doing quadrature, we cannot simply evaluate the integrand as is. We need to use a discrete approximation of the derivative terms which are part of the integrand. Otherwise, the error of the quadrature can be almost arbitrarily bad.

We therefore replace the analytic derivative by a finite difference approximation, where we use a step size proportional to the medium extinction and the ray marching step size. We provide an expanded explanation and derivation in the supplemental material. The ray marching algorithm then simplifies to evaluating the following recursive expression:

$$T(\mathbf{x}, \mathbf{y}) \approx f \left(f^{-1}(T_{N-1}, \gamma(t_i)) + \sigma_t(t_i) \Delta_{\text{step}}, \gamma(t_i) \right) \quad (8)$$

where, N is the number of steps in the ray marching routine, Δ_{step} the step size and T_i refers to the transmittance from iteration i in the evaluation (with $T_0 = 1$). We provide pseudocode for the transmittance evaluation in Listing 1. This formulation has a simple intuitive meaning: It can be seen as iteratively decreasing the transmittance according to the local transmittance model and density. We found this particular evaluation scheme to work well in practical applications of our model.

Free-flight distance sampling. To use this model in a volumetric path tracer, we need to be able to sample free-flight distances according to it. This is can be done by numerically inverting the cumulative distribution function (CDF). We sample a uniform random number $u \sim \mathcal{U}(0, 1)$ and find t such that $T(\mathbf{x}, \mathbf{x}_t) = u$ by marching along the ray. We perform bisection search on the last segment to precisely determine the sampled distance.

```

1 def transmittance( $\sigma_t, \gamma$ ):
2   T = 1
3   for i in range(N):
4     t = i *  $\Delta_{\text{step}}$  # Evaluate distance along ray
5     T = f [  $f^{-1}(T, \gamma(t)) + \sigma_t(t) \cdot \Delta_{\text{step}}, \gamma(t)$  ]
6   return T

```

Listing 1. Pseudocode to evaluate the transmittance using our model

This works because the transmittance is defined to be one minus the CDF of the free-flight distance distribution. The PDF of the resulting sample is then the negative derivative of the final transmittance value. Since the transmittance is a definite integral evaluated from 0 to t , the derivative of the transmittance with respect to t is simply the integrand itself:

$$T_{\text{pdf}}(t) = -\frac{\partial f}{\partial \tau} \left(f^{-1}(T(\mathbf{x}, \mathbf{x}_t), \gamma(t)), \gamma(t) \right) \sigma_t(t). \quad (9)$$

For simple forward rendering of a scene with a monochromatic extinction coefficient, this PDF does not have to be evaluated explicitly, since the same term occurs in the non-exponential volume rendering integral and therefore cancels out.

5 VOLUMETRIC APPEARANCE MODEL

Our transmittance model is motivated by the problem of approximating complex scenes as scattering volumes. In this section, we describe the full set of parameters required by our fully volumetric scene representation. Our representation is parametrized by albedo α , phase function parameters and transmittance parameters. For our main experiments, all parameters are stored on a voxel grid and interpolated trilinearly between voxels. The spatial behavior of these parameters could also be modeled by a neural network to further reduce memory usage.

Transmittance. The transmittance is modeled by our heterogeneous transmittance model introduced in Section 4. We store both extinction σ_t and transmittance mode γ on voxel grids. Both of these parameters are monochrome, but the method could be extended to handle spectrally varying transmittance parameters.

Phase function. The phase function describes the angular distribution of scattering in a volume and plays a crucial role in representing complex appearance. Since we are interested in representing surfaces, it is preferable to use an anisotropic phase function [Jakob et al. 2010]. We use the SGGX microflake phase function [Heitz et al. 2015], as it is straightforward to use and has robust sampling and evaluation routines. Being a microflake phase function, the SGGX is defined by its normal distribution function (NDF). The NDF in this case is the distribution of normals on a spherical ellipsoid. This ellipsoid, and hence the SGGX phase function, is parametrized by a symmetric positive definite 3×3 matrix.

Unless specified otherwise, we use an SGGX microflake phase function with diffuse microflakes for all our results. One advantage of the SGGX phase function is that it also allows to partially model the directional dependence of transmittance. In microflake theory, the extinction of the medium is modulated by the projected area of

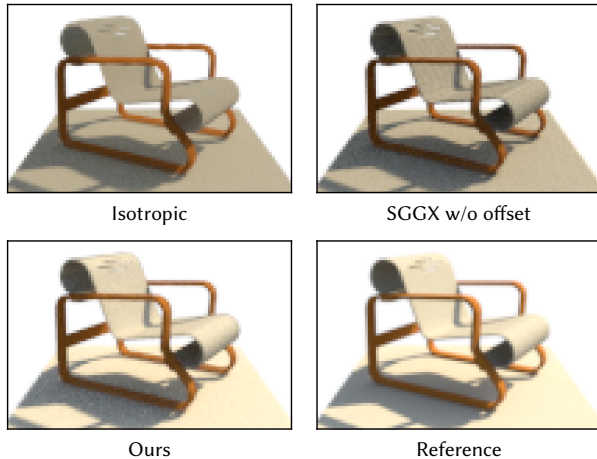


Fig. 3. This figure illustrates the difference between using an isotropic phase function, a standard SGGX phase function and an SGGX phase function with an offset after scattering (*Ours*). Applying an offset reduces energy loss and allows to more faithfully approximate the reference appearance.

the microflakes:

$$\sigma_{\tau}(\mathbf{x}, \omega_i) = \sigma_{\tau}(\mathbf{x}) \cdot \int_{\Omega} \langle \omega_i, \omega_m \rangle D(\omega_m) d\omega_m, \quad (10)$$

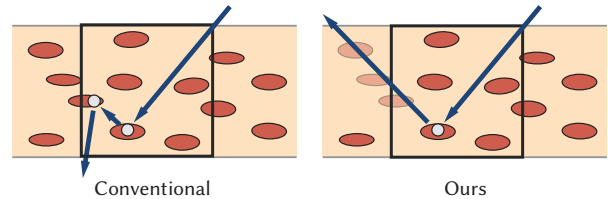
where D is the distribution of microflake normals and $\langle \cdot, \cdot \rangle$ is the dot product clamped to positive values. The SGGX phase function admits a closed-form solution to this integral. For a diffuse plane, the SGGX microflake distribution can almost perfectly model its discrete NDF. The projected microflake area will then tend to zero as we approach grazing angles, just as we would expect for the opacity of plane. We found this to be useful, since we otherwise do not model any directional dependence of the transmittance. Developing a closer coupling between microflake theory and non-exponential media remains an interesting avenue for future work. Further, the modeling power of the phase functions is one of the main factors limiting the generality of our methods.

Separation of local and global scattering. As described, the presented model does not in any way restrict multiple scattering inside a voxel. This means that even when using linear transmittance and a sufficiently high extinction, we can still end up with light paths eventually going through an opaque surface due to multiple scattering. This does not only result in light leaking, but also produces significant energy loss on the visible side of a surface.

One possible approach to this problem could be to use one-sided microflakes, which are transparent from the backside [Dupuy et al. 2016]. However, generalizing this concept to arbitrary scenes is difficult. We chose a simpler solution: after each medium scatter event, we offset the start of the next ray by the size of a voxel. This is similar to using an epsilon for shadow rays to prevent self intersections. The following illustration shows the issue. Conventional path tracing would simulate multiple scattering inside a voxel and the path could eventually reach the other side. By offsetting the ray origin, this is mostly prevented:

Table 1. The resulting *file size* and *percentage of empty voxels* of the sparse volumetric representation for several example scenes. The variation in compressed size is due to different levels of sparsity in the original scenes.

Resolution	16	32	64	128	Original
City building	49.7 KB	0.2 MB	1.1 MB	4.9 MB	1.1 GB
	42.2%	65.9%	80.5%	88.8%	
City	32.6 KB	0.1 MB	0.7 MB	3.4 MB	1.5 GB
	62.1%	80.3%	87.9%	92.4%	
Trees	35.4 KB	0.2 MB	1.0 MB	5.0 MB	452 MB
	58.8%	73.9%	82.6%	88.7%	
Checkerboards	53.7 KB	0.2 MB	0.9 MB	3.3 MB	36 KB
	37.6%	66.1%	83.9%	92.5%	
Fractal	51.7 KB	0.3 MB	1.5 MB	8.0 MB	42 MB
	39.9%	61.1%	73.4%	81.9%	



In Figure 3, we show the impact of this offset on an example scene. Offsetting the ray origin is related to shell tracing [Moon et al. 2007; Müller et al. 2016], where local transport is summarized over a spherical region and the light path proceeds from a location sampled on a sphere around the current location in the medium.

6 APPLICATION: APPEARANCE PREFILTERING

One application of our model is appearance prefiltering for level of detail, or scene compression. Given a complex scene, we try to approximate it using a lower resolution volumetric representation. In our volumetric appearance model, we have to determine the parameters that reproduce a certain target appearance. During rendering, we use a voxel grid resolution that is appropriate for the size of the image pixels. Therefore, we build a hierarchy of voxel grids covering different resolutions, doubling the resolution between each scale. We fit the medium parameters separately for each target resolution, as simply downsampling from the finest resolution would not preserve appearance. The fitting time is dominated by the runtime at the highest resolution, so fitting parameters independently at lower resolutions does not significantly impact processing time.

The fitting pipeline consists of multiple stages illustrated in Figure 4. First, we compute a binary voxelization of the scene, marking each non-empty voxel for further processing. We use Binvox [Min 2021; Nooruddin and Turk 2003] to determine the set of non-empty voxels. This first step is important for computational efficiency, since the total number of voxels in a grid of width n is n^3 , but the number of occupied surface voxels only increases at a rate of $O(n^2)$. Our goal is to create a *surface voxelization*, i.e., we do not fill in solid objects. This ensures that our representation remains sparse, and therefore efficient to render and store. We show compression rates for a few example scenes in Table 1.

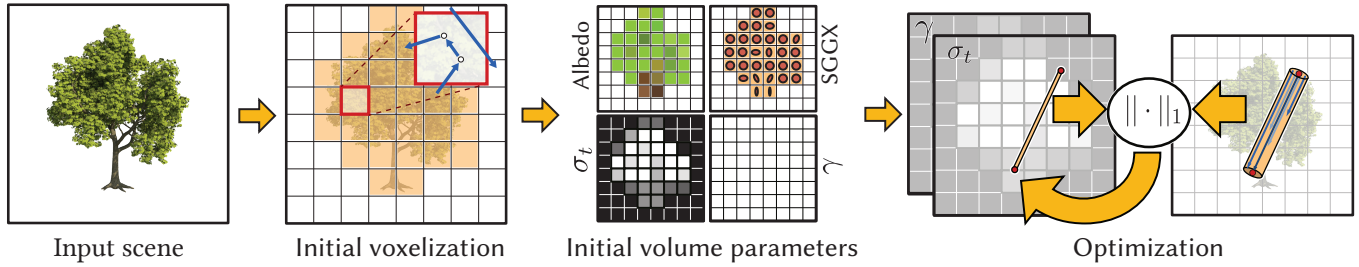


Fig. 4. Overview of our scene prefiltering pipeline. We first obtain an initial binary voxelization of the scene and then trace rays in each occupied voxel to determine local appearance parameters. We further refine the transmittance model by optimizing the σ_t and γ parameters using gradient descent. During the optimization, we sample random line segments in the scene and compute an L_1 loss between our transmittance model and the reference transmittance (estimated using ray tracing). This whole pipeline is run once for each target resolution.

We then trace light paths in each non-empty voxel using a path tracer and estimate the voxel’s albedo by averaging their throughput. Therefore, the albedo of a voxel already accounts for multiple scattering within this voxel.

Phase function. For the SGGX phase function, we fit parameters using the algorithm provided by Heitz et al. [2015]. The ellipsoid defining the SGGX phase function can be expressed by 3 eigenvectors and corresponding projected areas. To fit these parameters, we first obtain a distribution of surface normals inside the voxel by intersecting rays with the geometry. Given a large number of sampled normals, we then compute the covariance matrix of the components of these normals. By performing an eigendecomposition on this covariance matrix we obtain the eigenvectors of the ellipsoid. The projected areas are then computed by projecting the sampled normals onto these three eigenvectors.

6.1 Transmittance Optimization

There is no closed-form solution for the extinction σ_t and transmittance mode γ . Even when using the original exponential transmittance in combination with a microflake phase function, the extinction coefficient of a single voxel does not have a closed-form solution. This is due to the modulation of the extinction by the view-dependent projected area of the microflake distribution. In previous work [Loubet and Neyret 2017], they fit the extinction value locally by performing gradient descent. This can be done by tracing a number of rays for each voxel and then optimizing the extinction value to reproduce the observed directionally varying opacity behavior.

At first, this seems like the right way to solve this problem: to determine the per-voxel extinction, we should just consider what happens inside the region of the scene represented by that single voxel. However, there are two main issues with this approach. First, it completely ignores any correlation effects across voxels. The second issue is less obvious, but equally important: Due to the nature of the volumetric approximation, it is typically impossible to perfectly fit the reference scene. By fitting parameters independently per voxel, we effectively prioritize per voxel error over error at larger scales or even image space error. Empirically we found this to lead to significantly less accurate results than optimizing across *multiple* voxels. The benefit of this is illustrated in Figure 5. Both exponential

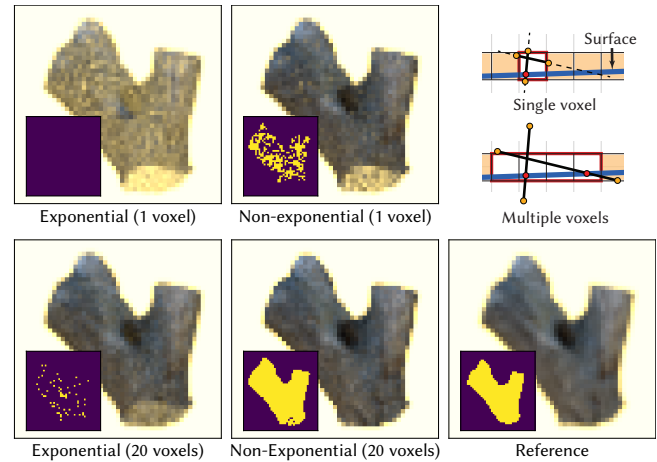


Fig. 5. Fitting over several voxels can drastically reduce error. In the top row, we fit medium parameters to represent the transmittance behavior of a single voxel. In the bottom row, the extinction is fit over beams of the length of 20 voxels. In these examples, the composited background has a constant RGB color of (4, 3, 0.9), which helps to highlight leakage. In the insets, we highlight pixels with more than 99.99% opacity. Note how the exponential representation does not even come close to being fully opaque. The illustration in the top right shows how fitting over multiple voxels better represents the absorption behavior for rays at grazing angles.

and non-exponential transmittance models strongly benefit from optimizing transmittance over several voxels.

To optimize our transmittance parameters, we first initialize the extinction coefficient of each non-empty voxel by tracing N rays through its region in the reference scene. We then compute

$$\sigma_t^{\text{init}} = -\frac{1}{s} \log \left(\frac{1}{N} \sum_{i=1}^N V_i \right), \quad (11)$$

where s is the side length of a single voxel and V_i the visibility of sample i (i.e., 1 if the ray passed through the voxel and zero otherwise). The transmittance mode γ is initialized to 1, which corresponds to exponential transmittance. This initialization gives the optimizer a starting guess to work from. We then optimize the

parameters by minimizing the following loss function:

$$L(\sigma_t, \gamma) = \int_{\mathcal{V}} \int_{\mathcal{V}} |T(\mathbf{x}, \mathbf{y}) - T^{\text{ref}}(\mathbf{x}, \mathbf{y})| p(\mathbf{x}, \mathbf{y}) \, d\mathbf{x} \, d\mathbf{y}, \quad (12)$$

where \mathbf{x} and \mathbf{y} are 3D points following a distribution $p(\mathbf{x}, \mathbf{y})$. In practice, we sample \mathbf{x} and \mathbf{y} by generating randomly oriented segments passing through occupied voxels. Practically, we found that segments of 20 voxels in length are sufficient for capturing even sophisticated correlations along grazing angles. The reference transmittance $T^{\text{ref}}(\mathbf{x}, \mathbf{y})$ is computed by tracing rays between the two points. In a scene consisting of hard surfaces, the reference transmittance is the binary visibility function. Since our volumetric approximation is bandlimited due to the voxel grid resolution, we also filter the reference transmittance with a small Gaussian kernel of a standard deviation of $s/6$, where s is the voxel size in world units. This means, that instead of just tracing rays between \mathbf{x} and \mathbf{y} , we randomly offset the ray origin in a plane perpendicular to the vector between the two points (as illustrated in Figure 4). This aids optimization, as it reduces variance in the evaluation of the reference transmittance. We optimize this loss over batches of 16k voxels simultaneously using the Adam optimizer [Kingma and Ba 2015] and a learning rate of 0.5. We run this optimization for 512 epochs over all occupied voxels.

6.2 Efficient Transmittance Gradient Computation

To optimize our transmittance model, we need to compute the gradient of the objective function with respect to the parameters. Simply using automatic differentiation to optimize transmittance parameters results in very high memory consumption, as the state of each loop iteration has to be stored. This was also observed by previous work on optimizing transmittance [Zheng et al. 2020]. Efficiently optimizing through a for-loop can be difficult using a conventional automatic differentiation framework.

We therefore implemented both the transmittance evaluation and its gradient manually using CUDA. The key idea to make this efficient is to evaluate the derivatives in *reverse mode*. While normally the term reverse mode is used in the context of automatic differentiation, the exact same principle applies when computing derivatives by hand. Using reverse-mode differentiation is crucial to optimizing many parameters at once, without having to run the forward computation once per parameter. We further use the fact that we can easily invert an iteration of the transmittance computation loop. This allows to implement the transmittance optimization without any temporary storage. We therefore only store the wavefront of rays to evaluate and the volume parameters. The idea of using reversible computations to reduce memory usage when computing derivatives has for example also been applied to neural networks [Gomez et al. 2017].

In our implementation, we first evaluate the transmittance using ray marching in the forward direction. Then we compute the loss and the gradient of the loss. This gradient is then propagated in a reversed ray marching loop, where we accumulate gradients of the extinction and transmittance mode parameters. In each iteration of the reverse loop, we compute the previous loop state using the inverse transmittance function. We validated our analytic gradients against finite differences and automatic differentiation using

Enoki [Jakob 2019]. The pseudocode for our reverse transmittance evaluation is given in Listing 2. The adjoint function takes the loss gradient δT and result of the forward pass, T , as inputs. Additionally, we pass the transmittance parameters and corresponding gradient variables, in which the gradients are accumulated.

```

1 def transmittance_adjoint( $\delta T$ ,  $T$ ,  $\sigma_t$ ,  $\gamma$ ,  $\delta\sigma_t$ ,  $\delta\gamma$ ):
2   for i in reversed(range(N)):
3     t = i ·  $\Delta_{\text{step}}$ 
4      $\hat{\sigma}_t = \sigma_t(t)$  # Get current medium parameters
5      $\hat{\gamma} = \gamma(t)$ 
6      $T = f[f^{-1}(T, \hat{\gamma}) - \hat{\sigma}_t \cdot \Delta_{\text{step}}, \hat{\gamma}]$  # Inverse loop iteration
7      $\tau = f^{-1}(T, \hat{\gamma}) + \hat{\sigma}_t \cdot \Delta_{\text{step}}$  # Optical depth
8     # Accumulate  $\sigma_t$  and  $\gamma$  gradients
9      $\delta\sigma_t(t) += \delta T \cdot \Delta_{\text{step}} \cdot f_{\tau}(\tau, \hat{\gamma})$ 
10     $\delta\gamma(t) += \delta T \cdot [f_{\tau}(\tau, \hat{\gamma}) \cdot f_{\gamma}^{-1}(T, \hat{\gamma}) + f_{\gamma}(\tau, \hat{\gamma})]$ 
11     $\delta T *= f_{\tau}(\tau, \hat{\gamma}) \cdot f_T^{-1}(T, \hat{\gamma})$  # Update Jacobian
12  return T

```

Listing 2. Pseudocode for the memoryless adjoint transmittance evaluation. Subscripts to functions denote partial derivatives, e.g., $f_{\tau} = \partial f / \partial \tau$.

7 APPLICATION: IMAGE-BASED RECONSTRUCTION

As another application, our non-exponential transmittance model can be applied to the problem of image-based volumetric scene reconstruction. Using a volumetric representation is advantageous for differentiable rendering, as it does not have to handle edge discontinuities [Li et al. 2018]. A model based on volume rendering is always continuous, as long as the underlying volumetric grids use trilinear interpolation (or a continuous neural network).

Given a set of reference images, we can optimize extinction values, transmittance mode, phase function parameters and albedo on a voxel grid to match the reference appearance. In our experiments, we use voxel grid resolutions of up to 256^3 . For all parameters we start with a uniform grid as our initial guess. We optimize parameters using differentiable rendering and up to 4 bounces of indirect illumination. A higher number of indirect bounces could be used, but this increases computation time and did not significantly change our results. We use a coarse-to-fine optimization routine to improve the convexity of the problem, similar to the volumetric optimizations in Nimier-David et al. [2019]. We start at a volume and image resolution of 4^3 and 4^2 , respectively, and then increase both resolutions after a number of iterations up to reaching the final volume resolution. The low starting resolution not only makes the problem more convex, but also makes the initial optimization iterations cheaper to render.

We implemented a differentiable volume renderer using CUDA and analytic derivatives. Computing gradients of a rendered image is a straightforward extension of the code used to optimize transmittance between two points (see Listing 2). Using a unified volumetric representation means that our renderer does not need to support surfaces. We use volumetric path tracing with next event estimation, and then use an approach similar to recent adjoint methods [Nimier-David et al. 2020; Stam 2020] to compute parameter gradients.

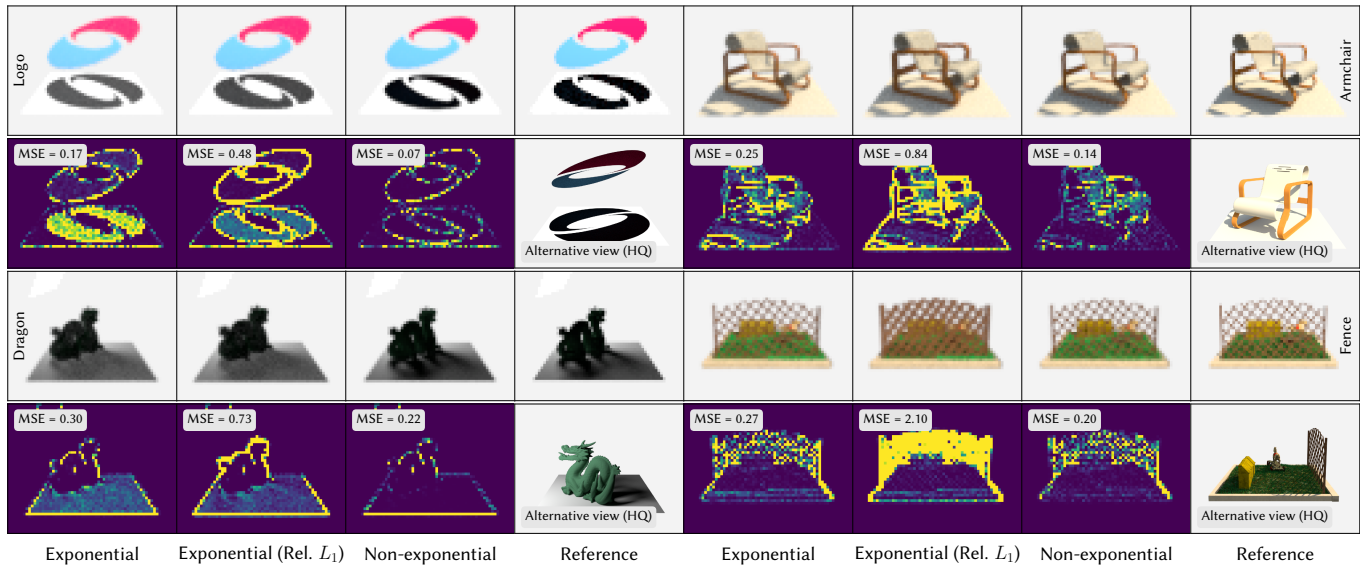


Fig. 6. These results show differences between using exponential vs. our more generic non-exponential transmittance. We also compare to the result obtained by optimizing the exponential model using a relative loss, which favors higher extinction values. Regardless of the loss function, the exponential model often leads to light leaking, resulting in overly bright shadows. Using a relative loss, we get worse results on semitransparent structures, such as the bloated appearance of the fence in the bottom right scene.

We perform the optimization under a uniform illumination and then re-render the result using a novel illumination condition. The illumination condition we use for the optimization is advantageous as no hard shadows are cast into the scene. Since this paper is about the new transmittance model, we leave the issue of optimizing under more difficult illumination conditions to future work. Our optimization uses a silhouette loss to further reduce ambiguities between fore- and background.

8 RESULTS

We implemented our method on top of Mitsuba 2 [Nimier-David et al. 2019] and Enoki [Jakob 2019]. A large part of the pipeline has been written utilizing the Python frontend to Mitsuba. Expensive computations are carried out on an Nvidia RTX Titan GPU using CUDA and we trace rays using OptiX [Parker et al. 2010]. For our appearance prefiltering results, we implemented a dynamic mip mapping scheme based on ray differentials [Igehy 1999]. Using the ray differentials, we estimate the size of a pixel in world space as we intersect the volume bounding box. We then probabilistically choose one of the two volume resolutions which most closely match the scale of the pixels. For simplicity, we use the same volume resolution for the whole light path. Our supplemental video contains several zoom ins where we choose the mip level in this way. The following results in the paper all have been computed at a fixed scale to facilitate comparisons to previous work.

Performance. We evaluate the performance of our methods using both exponential and non-exponential transmittance models. When prefiltering the "City building" scene (see Figure 2) at a resolution of 128^3 voxels, it takes around 1 minute to compute albedo, SGGX parameters and initial extinction values. Optimizing the transmittance

then takes 4 minutes using an exponential model and around 4.5 minutes using our non-exponential transmittance, which amounts to around 12.5% overhead. Rendering the optimized volume on the CPU is around 35% slower when using our non-exponential model than when using the exponential model. For reference, path tracing the same scene using Embree [Wald et al. 2014] is around 5× faster than rendering the volumetric representation. This is not surprising, as our volume rendering implementation was only lightly optimized.

Image-based reconstruction of a volumetric scene representation is significantly more challenging than appearance prefiltering and takes around 1 hour for the results shown in Figure 14 (256^3 voxels). The non-exponential model takes around 10% more time to optimize. However, the CUDA implementation could be simplified to support purely exponential media more efficiently.

Exponential vs. non-exponential transmittance. For appearance prefiltering, we show the practical benefits from switching to a non-exponential transmittance in Figure 6. We ran our transmittance optimization both for the exponential and our general non-exponential model. For both sets of results, we separate local and global scattering by offsetting the starting location of rays. This is a form of non-exponentiality, but a standard exponential medium would result in significant energy loss, as shown in Figure 3.

We found that the exponential model suffers from significant leaking artifacts. One possible approach to reduce leaking is to more strongly penalize this type of error. Therefore, we experimented with using a relative loss, where the L_1 transmittance loss is divided by the ground truth transmittance (clamped to a small epsilon to prevent division by zero). This increases the loss value if the reference transmittance is zero or close to zero. We found that this helped to

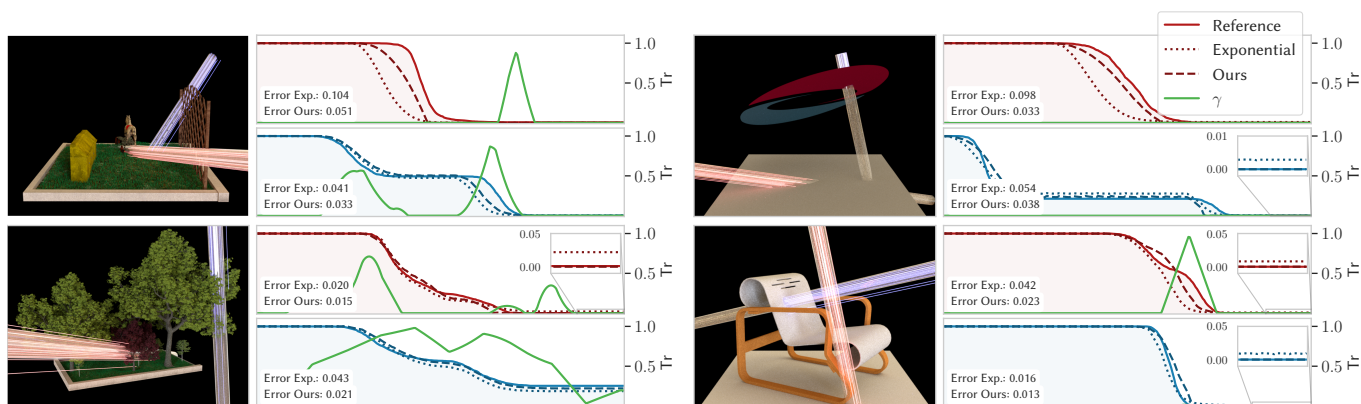


Fig. 7. We plot the reference simulated transmittance (solid line) against both our non-exponential and exponential transmittance fits. In all these results we use a voxel grid of 16^3 resolution and compare to the ground-truth transmittance computed using ray tracing in a beam of the same footprint. The exponential transmittance often decreases too rapidly for surfaces, which leads to bloating. However, in other scenarios, despite this rapid decrease in transmittance, the exponential model leads to significant leaking, as highlighted in the inset plots, due to its inability to easily reach zero. The non-exponential model fits the transmittance curves more faithfully and prevents leakage.

reduce leaking, but at the same time drastically increased bloating, especially for semitransparent regions such as the fence in Figure 6. This indicates that the exponential model cannot easily be fixed by just changing the optimization routine. This is a trade-off between opaqueness and amount of bloating inherent to the exponential transmittance model. Our more general non-exponential model can find a better compromise between these conflicting goals. We found it to mostly eliminate leaking as well as reducing bloating compared to the exponential model.

In Figure 7 we plot the transmittance obtained after optimizing medium parameters, and the corresponding values of γ . These plots demonstrate that the optimizer is able to automatically detect regions with different modes (opaque vs. aggregate uncorrelated) and pick the proper transmittance mode for it by selecting the γ parameter value. It further showcases the issues related to bloating and leaking found in the exponential model. We can also see that both the non-exponential and exponential model are approximations and do not give a perfect match to the reference transmittance.

Figure 8 visualizes the γ values after optimization. The optimizer prefers using linear transmittance for solid objects, while it resorts to the classic exponential transmittance for aggregate or unstructured detail, such as leaves towards the outside of the trees. We additionally discuss the optimization convergence for a few example scenes in the supplemental material.

Reciprocity. Our transmittance model is not reciprocal. This means that evaluating the transmittance $T(\mathbf{x}, \mathbf{y})$ might not match evaluating $T(\mathbf{y}, \mathbf{x})$. We evaluate the practical impact of this limitation by switching the evaluation direction of the transmittance when tracing shadow rays. We render several volumetric scene representations both using the unmodified path tracer and the implementation with the reversed shadow rays. The results of this experiment are shown in Figure 9. The differences caused by non-reciprocal behavior are almost imperceptible. This indicates that the non-reciprocity is not a limiting factor in practice. We provide some additional evaluation in the supplemental material.

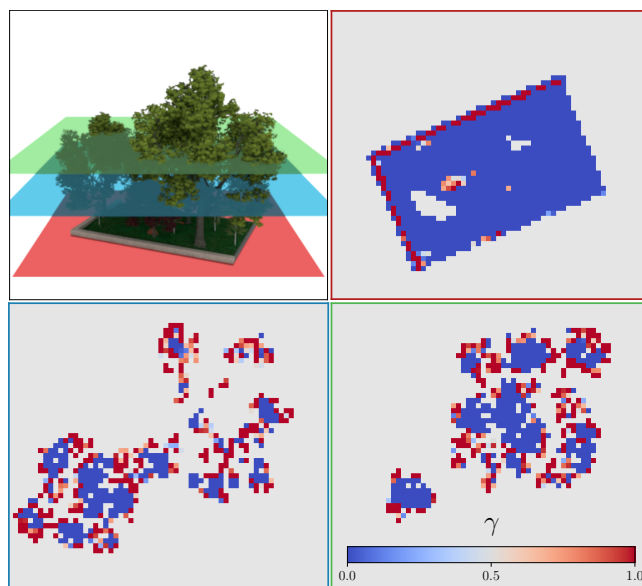


Fig. 8. We visualize the fitted values for γ over several horizontal slices in the same scene. The voxel grid used here has a resolution of 64^3 . These slices show that the optimizer indeed automatically classifies voxels containing opaque geometry to use linear transmittance (blue) and voxels containing more unstructured geometry (e.g. leaves) towards using exponential transmittance (red).

Specular surfaces. The problem of finding the right transmittance model is mostly orthogonal to the problem of defining the phase function. We therefore use simple diffuse BSDFs in most of our scenes. However, our prefiltering method can also be applied to scenes with specular surfaces, as we show in Figure 10. Our volumetric model reproduces the overall appearance of the golden bunny, but leads to a slight loss in sharpness of the reflection. It

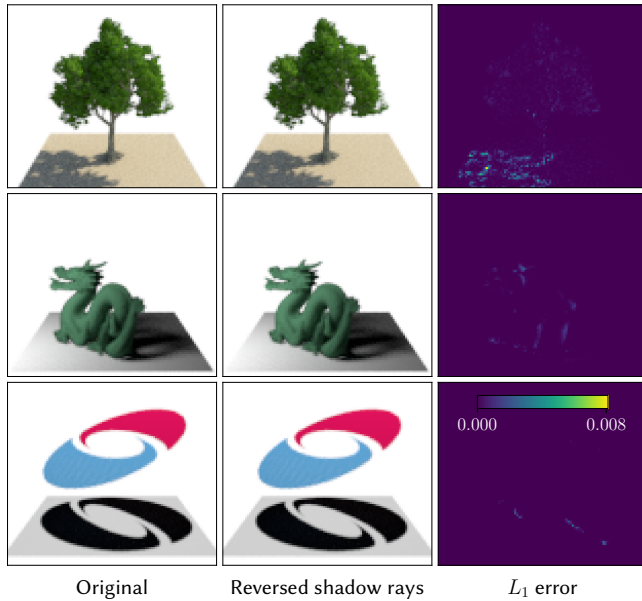


Fig. 9. We evaluate the practical impact of our model’s non-reciprocity by rendering our volume as usual, but evaluating shadow rays in the reverse direction (second column). The difference to the original rendering using the unmodified path tracer is almost imperceptible. All error maps are scaled consistently and normalized to make the small differences visible.

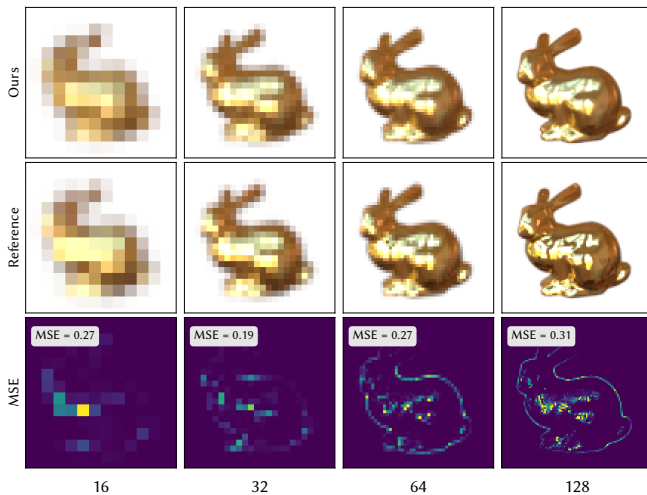


Fig. 10. By switching the SGGX phase function to use specular microflakes, we can also compute level of detail for metallic objects.

remains an important avenue of future work to define a general phase function models to handle the full spectrum of diffuse, plastic and metal objects.

Comparison to prior work. We compare our prefiltering method to the state-of-the-art Hybrid LoD method [Loubet and Neyret 2017]. Figure 11 shows the comparison on several example scenes. We used the original implementation provided by the authors, which

we extended to be able to load Mitsuba 2 scenes. We run both methods at resolutions 16^2 and 64^2 and compare to the ground truth of rendering the reference scene at the same resolution. The Hybrid LoD method produces good results on volume-friendly scenes (i.e., aggregate details), e.g., the “Trees” scene. However, it often misclassifies complex and mixed types of geometry as a volume (e.g., for the “City building” and “City” scenes), which leads to a semitransparent appearance that is far from the ground truth. The “Checkerboards” scene also shows how this behavior is scale dependent. At the higher resolution, it correctly classifies all surfaces as surfaces. However, as the resolution is decreased it switches to a volumetric representation, which changes the appearance of the object drastically. Our method does not perform a binary classification and therefore can maintain a consistent approximation quality across scales. The optimizer automatically lands at the best transmittance mode, also including the spectrum between linear (opaque) and exponential (aggregate) transmittance modes. We show a higher resolution rendering of a LoD volume computed using our method compared to Hybrid LoD to show the issues of this binary classification in Figure 12. The binary classification results in drastic changes of appearance across a single LoD.

Application to neural radiance fields. We also ran some experiments where we modified the NeRF code base [Mildenhall et al. 2020] to use a non-exponential transmittance. The results of this are shown in Figure 13. Using a non-exponential transmittance, we can achieve some improvements in the sharpness of the generated images. On the “red plane” scene, the unmodified NeRF model struggles to represent the vertical plane. Our non-exponential model manages to produce more accurate results. The combination of non-exponential transmittance with neural networks to solve real computer vision problems, however, remains an interesting future direction.

Application to image-based reconstruction. As described in Section 7, we can also apply our non-exponential transmittance model to improve image-based reconstruction using differentiable path tracing. In Figure 14, we show reconstructions of several example scenes using an exponential and our non-exponential representation. All results use a voxel grid resolution of 256^3 , which means that we optimize ca. 184 million parameters. However, since a lot of the voxels are empty, the optimization problem remains tractable. We optimize for 64 views of the synthetic scene at once. The optimization uses around 21 GB of VRAM, but the code could be optimized further.

Both the exponential and our non-exponential model converge to a meaningful volumetric approximation of the reference scene. However, the exponential model again suffers from leaking, despite the explicit silhouette loss. This is primarily visible in thin structures. At the same time, this exponential model also suffers from more bloating than our non-exponential model, as shown well in the details of the “Lego” scene. The exponential model fills in parts of the beams where holes should be present, while leading to leaking in other parts. This shows that using an exponential transmittance model is not sufficient when trying to relight a volumetric scene reconstruction, which is consistent with our observations made for

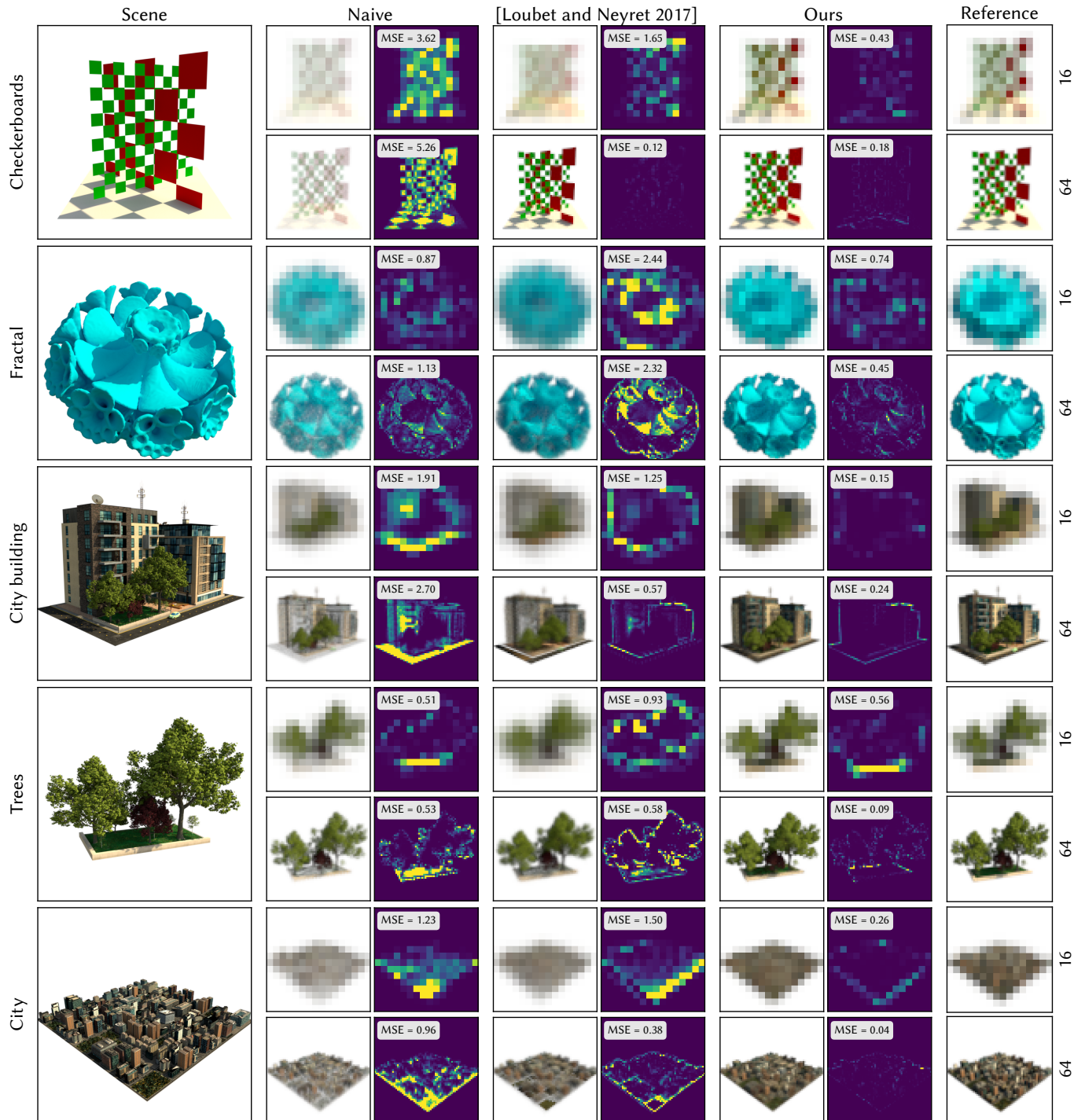


Fig. 11. Prefiltering results on a variety of scenes with different complexity. *Reference* is the path traced ground truth geometry. The reference is rendered using a Gaussian pixel filter to bandlimit the signal. *Naive* is a naive volumetric approximation of the scene. This is the starting point for our optimization. All LoD methods use a voxel resolution which matches the resolution of the rendered images. *Loubet and Neyret 2017* is the previous state of the art in automatic volumetric level of detail and *Ours* is our non-exponential model. All results are rendered using 1024 samples per pixel and we visualize mean squared errors (MSE) compared to the reference image. All MSE values have been multiplied by a factor of 100 for readability.

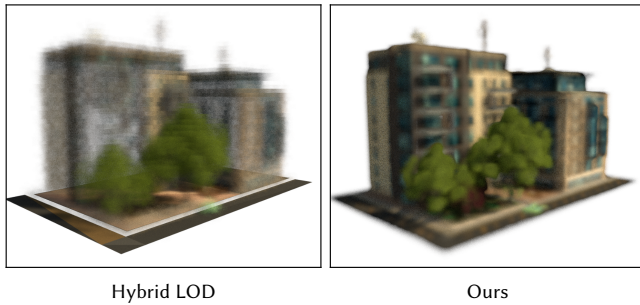


Fig. 12. If we render low resolution LoD volumes at a higher pixel resolution, we can see that the Hybrid LoD method by Loubet and Neyret [2017] struggles to correctly separate surfaces and volumes. Our method results in a smoother model, as it does not apply a binary classification between surface-like and volume-like scattering.

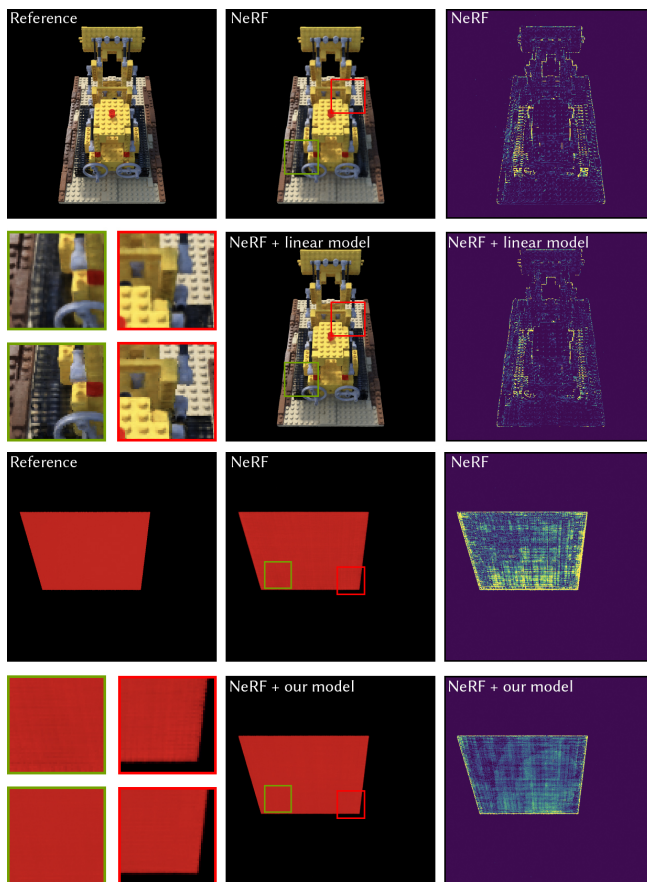


Fig. 13. Neural volumetric representation of the "Lego" scene (top) using NeRF [Mildenhall et al. 2020] with the original exponential transmittance model and with a purely linear model. We also applied exponential NeRF and a version using our transmittance model to the "Red plane" scene (bottom).

the level of detail use case. Overall, we observe that the improvement on image-based reconstruction is smaller than for appearance prefiltering. Image-based optimization using global illumination is

a harder problem than fitting transmittance parameters against a known 3D scene. Additionally, the image-based optimization does not enforce *sparsity* of the volume density field. The lack of sparsity constraints makes the results less sensitive to the transmittance model than for appearance prefiltering, where we explicitly enforce sparsity.

9 CONCLUSION AND FUTURE WORK

In this paper, we introduced a new volumetric transmittance model, which allows to represent both surface-like and volumetric appearance in a single coherent framework. Our model does not require explicit binary classification and we show that its parameters can be optimized efficiently using gradient-based optimization. We further show improvements to image-based reconstruction using differentiable rendering. Our model can easily be integrated into existing systems and is simple to implement.

While we found our transmittance formulation to work well for the shown applications, it is not constrained to be reciprocal. Developing a reciprocal formulation would be interesting and would make the model applicable in the context of bidirectional algorithms. It would also be interesting to investigate if the transmittance could be sampled and evaluated in an unbiased way, instead of using biased ray marching. One of the main practical limitations is the representation power of the phase function. The SGGX phase function [Heitz et al. 2015] is able to represent diffuse and metallic surfaces, but more work is required to define more general phase functions to handle plastic-like or dielectric materials. The phase function further serves an important role in preventing leaking due to multiple scattering. With the SGGX phase functions, more sampled directions will point into the surface as roughness of the normal distribution is increased. This can result in leaking of multiple scattering on the back of a surface and energy loss on the front side.

Overall, we hope that our work enables future research to go beyond just exponential transmittance models. We believe that this is a necessary step towards further unification of surface and volume rendering. With the recent gain in popularity of differentiable rendering, unifying these two worlds can benefit practical scene reconstruction approaches.

ACKNOWLEDGMENTS

We thank Matt Chapman for creating some of the example scenes (City building, City, Trees), Michael Goesele and Christoph Lassner for proofreading and the anonymous reviewers for their feedback and suggestions to improve the paper. We further thank Benedikt Bitterli and Adrian Jarabo for answering our questions about non-exponential media. We thank Guillaume Loubet for help running his Hybrid LoD method. We used several 3D models from the Stanford 3D scanning repository. The "Lego" scene has been created by Blendswap user Heinzelnisse and the "Drums" scene by Blendswap user bryanajones.

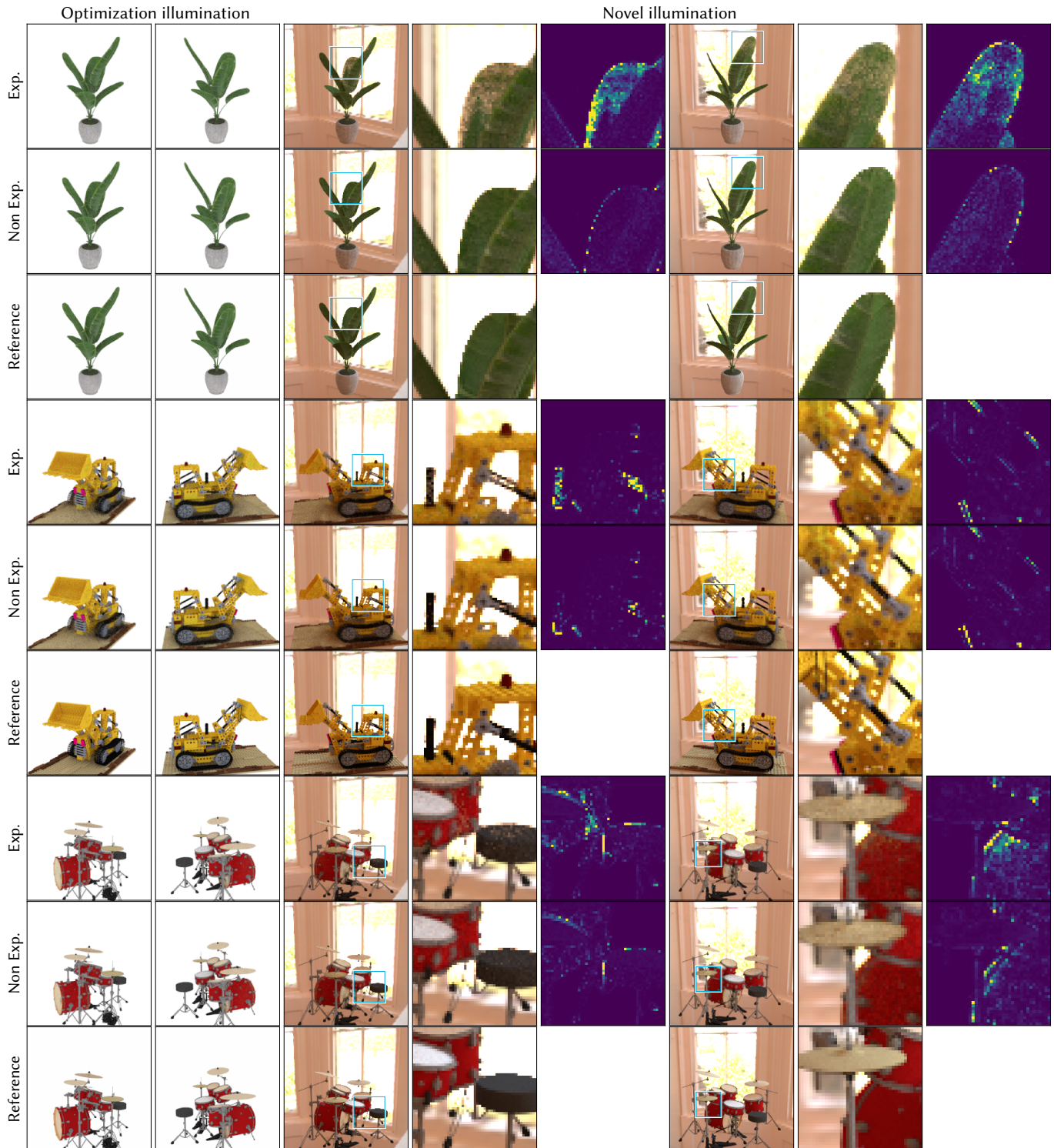


Fig. 14. Results of reconstructing scenes using an image-based optimization. We compare our non-exponential transmittance model with the exponential model and the ground truth reference. All volumes use a resolution of 256^3 and we use the same resolution to render the resulting images.

REFERENCES

- Johanna Beyer, Markus Hadwiger, and Hanspeter Pfister. 2015. State-of-the-Art in GPU-Based Large-Scale Volume Visualization. *Computer Graphics Forum* 34, 8 (Dec. 2015), 13–37.
- Sai Bi, Zexiang Xu, Pratul Srinivasan, Ben Mildenhall, Kalyan Sunkavalli, Miloš Hašan, Yannick Hold-Geoffroy, David Kriegman, and Ravi Ramamoorthi. 2020a. Neural Reflectance Fields for Appearance Acquisition. arXiv:2008.03824
- Sai Bi, Zexiang Xu, Kalyan Sunkavalli, Miloš Hašan, Yannick Hold-Geoffroy, David Kriegman, and Ravi Ramamoorthi. 2020b. Deep Reflectance Volumes: Relightable Reconstructions from Multi-view Photometric Images. In *ECCV*. 25 pages.
- Benedikt Bitterli, Srinath Ravichandran, Thomas Müller, Magnus Wrenninge, Jan Novák, Steve Marschner, and Wojciech Jarosz. 2018. A radiative transfer framework for non-exponential media. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* 37, 6 (2018), 225:1–225:17.
- Emanuele Caglioti and Francois Golse. 2003. On the Distribution of Free Path Lengths for the Periodic Lorentz Gas III. *Communications in Mathematical Physics* 236, 2 (May 2003), 199–221.
- Thomas Camminady, Martin Frank, and Edward W. Larsen. 2017. Nonclassical particle transport in heterogeneous materials. In *M&C 2017 : International Conference on Mathematics & Computational Methods Applied to Nuclear Science & Engineering*. Subrahmanyam Chandrasekhar. 1960. *Radiative transfer*. Dover publications, New York.
- Cyril Crassin, Fabrice Neyret, Sylvain Lefebvre, and Elmar Eisemann. 2009. GigaVoxels: Ray-Guided Streaming for Efficient and Detailed Voxel Rendering. In *Proceedings of the 2009 Symposium on Interactive 3D Graphics and Games* (Boston, Massachusetts). Association for Computing Machinery, New York, NY, USA, 15–22.
- Cyril Crassin, Fabrice Neyret, Miguel Sainz, Simon Green, and Elmar Eisemann. 2011. Interactive indirect illumination using voxel cone tracing. *Computer Graphics Forum* 30, 7 (2011), 1921–1930.
- Anthony Davis and Mark Mineev-Weinstein. 2011. Radiation propagation in random media: From positive to negative correlations in high-frequency fluctuations. *Journal of Quantitative Spectroscopy and Radiative Transfer* 112 (03 2011), 632–645.
- Eugene d'Eon. 2019. A Reciprocal Formulation of Nonexponential Radiative Transfer. 3: Binary Mixtures. arXiv:1903.08783
- Jonathan Dupuy, Eric Heitz, and Eugene d'Eon. 2016. Additional Progress towards the Unification of Microfacet and Microflake Theories. In *Proceedings of the Eurographics Symposium on Rendering: Experimental Ideas and Implementations* (Dublin, Ireland) (*Computer Graphics Forum (Proceedings of EGSR)*). Eurographics Association, Goslar, DEU, 55–63.
- Eugene d'Eon. 2018. A Reciprocal Formulation of Nonexponential Radiative Transfer. 1: Sketch and Motivation. *Journal of Computational and Theoretical Transport* 47, 1-3 (2018), 84–115.
- Eugene d'Eon. 2019. A Reciprocal Formulation of Nonexponential Radiative Transfer. 2: Monte Carlo Estimation and Diffusion Approximation. *Journal of Computational and Theoretical Transport* 48, 6 (2019), 201–262.
- Iliyan Georgiev, Zackary Misso, Toshiya Hachisuka, Derek Nowrouzezahrai, Ja roslav Křivánek, and Wojciech Jarosz. 2019. Integral formulations of volumetric transmittance. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* 38, 6 (2019), 17 pages.
- Ioannis Gkioulekas, Anat Levin, and Todd Zickler. 2016. An Evaluation of Computational Imaging Techniques for Heterogeneous Inverse Scattering. In *ECCV*, Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling (Eds.). Springer International Publishing, 685–701.
- Aidan N. Gomez, Mengye Ren, Raquel Urtasun, and Roger B. Grosse. 2017. The Reversible Residual Network: Backpropagation without Storing Activations. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Long Beach, California, USA) (*NIPS'17*). Curran Associates Inc., Red Hook, NY, USA, 2211–2221.
- Jie Guo, Yanjun Chen, Bingyang Hu, Ling-Qi Yan, Yanwen Guo, and Yuntao Liu. 2019. Fractional Gaussian Fields for Modeling and Rendering of Spatially-Correlated Media. *ACM Trans. Graph. (Proc. SIGGRAPH)* 38, 4, Article 45 (July 2019), 13 pages.
- Markus Hadwiger, Ali K. Al-Awami, Johanna Beyer, Marco Agus, and Hanspeter Pfister. 2018. SparseLeap: Efficient Empty Space Skipping for Large-Scale Volume Rendering. *IEEE Transactions on Visualization and Computer Graphics* 24, 1 (2018), 974–983.
- Eric Heitz, Jonathan Dupuy, Cyril Crassin, and Carsten Dachsbacher. 2015. The SGX Microflake Distribution. *ACM Trans. Graph. (Proc. SIGGRAPH)* 34, 4, Article 48 (July 2015), 11 pages.
- Eric Heitz and Fabrice Neyret. 2012. Representing Appearance and Pre-Filtering Subpixel Data in Sparse Voxel Octrees. In *Proceedings of the Fourth ACM SIGGRAPH / Eurographics Conference on High-Performance Graphics* (Paris, France) (*EGGH-HPG'12*). Eurographics Association, Goslar, DEU, 125–134.
- Homan Igehy. 1999. Tracing Ray Differentials. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 99)*. ACM Press/Addison-Wesley Publishing Co., USA, 179–186.
- Wenzel Jakob. 2019. Enoki: structured vectorization and differentiation on modern processor architectures. <https://github.com/mitsuba-renderer/enoki>.
- Wenzel Jakob, Jonathan T. Moon, Adam Arbree, Kavita Bala, and Steve Marschner. 2010. A Radiative Transfer Framework for Rendering Materials with Anisotropic Structure. *ACM Trans. Graph. (Proc. SIGGRAPH)* 29, 10 (July 2010), 53:1–53:13.
- Adrian Jarabo, Carlos Aliaga, and Diego Gutierrez. 2018. A Radiative Transfer Framework for Spatially-Correlated Materials. *ACM Trans. Graph. (Proc. SIGGRAPH)* 37, 4, Article 83 (July 2018), 13 pages.
- Anton Kaplanyan, Anton Sochenov, Thomas Leimkuehler, Mikhail Okunev, Todd Goodall, and Rufo Gizem. 2019. DeepFovea: Neural Reconstruction for Foveated Rendering and Video Compression using Learned Statistics of Natural Videos. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* 38, 4 (2019), 212:1–212:13.
- Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, Conference Track Proceedings*.
- Tzu-Mao Li, Miika Aittala, Frédo Durand, and Jaakko Lehtinen. 2018. Differentiable Monte Carlo Ray Tracing through Edge Sampling. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* 37, 6 (2018), 222:1–222:11.
- Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. 2019. Neural Volumes: Learning Dynamic Renderable Volumes from Images. *ACM Trans. Graph. (Proc. SIGGRAPH)* 38, 4, Article 65 (July 2019), 14 pages.
- Guillaume Loubet and Fabrice Neyret. 2017. Hybrid Mesh-Volume LoDs for All-Scale Pre-Filtering of Complex 3D Assets. *Computer Graphics Forum* 36, 2 (2017), 431–442.
- David Luebke, Martin Reddy, Jonathan D. Cohen, Amitabh Varshney, Benjamin Watson, and Robert Huebner. 2002. *Level of Detail for 3D Graphics*. Morgan Kaufmann Publishers Inc.
- Johannes Meng, Marios Pappas, Ralf Habel, Carsten Dachsbacher, Steve Marschner, Markus Gross, and Wojciech Jarosz. 2015. Multi-scale Modeling and Rendering of Granular Materials. *ACM Trans. Graph. (Proc. SIGGRAPH)* 34, 4, Article 49 (July 2015), 13 pages.
- Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *ECCV*. 25 pages.
- Patrick Min. 2004–2021. binvox. <http://www.patrickmin.com/binvox>. Accessed: 2021-04-20.
- Jonathan T. Moon, Bruce Walter, and Stephen R. Marschner. 2007. Rendering Discrete Random Media Using Precomputed Scattering Solutions. In *Proceedings of the 18th Eurographics Conference on Rendering Techniques* (Grenoble, France) (*Computer Graphics Forum (Proceedings of EGSR)*). Eurographics Association, Goslar, DEU, 231–242.
- Nate Morrill, Will Usher, Ingo Wald, and Valerio Pascucci. 2019. Efficient Space Skipping and Adaptive Sampling of Unstructured Volumes Using Hardware Accelerated Ray Tracing. In *2019 IEEE Visualization Conference (VIS)*. 256–260.
- Thomas Müller, Marios Pappas, Markus Gross, Wojciech Jarosz, and Jan Novák. 2016. Efficient Rendering of Heterogeneous Polydisperse Granular Media. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* 35, 6 (Dec. 2016), 168:1–168:14.
- Merlin Nimier-David, Sébastien Speierer, Benoît Ruiz, and Wenzel Jakob. 2020. Radiative Backpropagation: An Adjoint Method for Lightning-Fast Differentiable Rendering. *ACM Trans. Graph. (Proc. SIGGRAPH)* 39, 4, Article 146 (July 2020), 15 pages.
- Merlin Nimier-David, Delio Vicini, Tizian Zeltner, and Wenzel Jakob. 2019. Mitsuba 2: A Retargetable Forward and Inverse Renderer. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* 38, 6 (Nov. 2019), 17 pages.
- Fakir S. Nooruddin and Greg Turk. 2003. Simplification and Repair of Polygonal Models Using Volumetric Techniques. *IEEE Transactions on Visualization and Computer Graphics* 9, 2 (2003), 191–205.
- Jan Novák, Iliyan Georgiev, Johannes Hanika, and Wojciech Jarosz. 2018. Monte Carlo Methods for Volumetric Light Transport Simulation. *Computer Graphics Forum (Proceedings of Eurographics - State of the Art Reports)* 37, 2 (May 2018), 26 pages.
- Steven G. Parker, James Bigler, Andreas Dietrich, Heiko Friedrich, Jared Hoberock, David Luebke, David McAllister, Morgan McGuire, Keith Morley, Austin Robison, and Martin Stich. 2010. OptiX: A General Purpose Ray Tracing Engine. *ACM Trans. Graph. (Proc. SIGGRAPH)*, Article 66 (2010), 13 pages.
- Alexander Schwank, Callum James James, and Tony Micilotta. 2016. The Trees of The Jungle Book. In *ACM SIGGRAPH Talks* (Anaheim, California). Association for Computing Machinery, New York, NY, USA, Article 21, 2 pages.
- Jos Stam. 2020. Computing Light Transport Gradients using the Adjoint Method. *CoRR* (2020), 23 pages. arXiv:2006.15059
- Ingo Wald, Sven Woop, Carsten Benthin, Gregory S. Johnson, and Manfred Ernst. 2014. Embree: a kernel framework for efficient CPU ray tracing. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 1–8.
- Shuang Zhao, Wenzel Jakob, Steve Marschner, and Kavita Bala. 2012. Structure-aware synthesis for predictive woven fabric appearance. *ACM Trans. Graph. (Proc. SIGGRAPH)* 31, 4 (2012), 75.
- Shuang Zhao, Lifan Wu, Frédo Durand, and Ravi Ramamoorthi. 2016. Downsampling Scattering Parameters for Rendering Anisotropic Media. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* 35, 6, Article 166 (Nov. 2016), 11 pages.
- Quan Zheng, Vahid Babaei, Gordon Wetzstein, Hans-Peter Seidel, Matthias Zwicker, and Gurprit Singh. 2020. Neural Light Field 3D Printing. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* 39, 6, Article 207 (2020), 12 pages.