

# Volume Scattering Probability Guiding

KEHAN XU, ETH Zürich, Switzerland

SEBASTIAN HERHOLZ, Intel Corporation, Germany

MARCO MANZI, DisneyResearch|Studios, Switzerland

MARIOS PAPAS, DisneyResearch|Studios, Switzerland

MARKUS GROSS, DisneyResearch|Studios, Switzerland and ETH Zürich, Switzerland

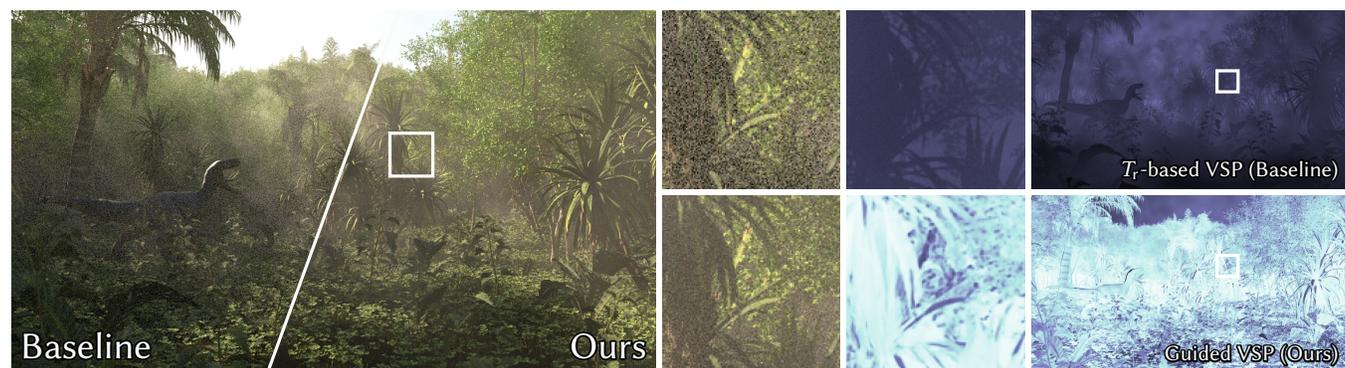


Fig. 1. Equal-time (5min) comparison of rendering a scene containing a thin high-albedo heterogeneous fog (JUNGLE) using our volume scattering probability guiding (VSPG) framework (**Ours**) against standard delta tracking (**Baseline**). The corresponding volume scattering probability (VSP) is visualized on the **right**, where brighter regions indicate higher probability. In this example, delta tracking does not generate enough volume samples (**top right**), resulting in a high variance in the volume contribution estimate, while VSPG automatically identifies the importance and variance of the volume contribution estimator and, therefore, increases the volume scatter probability (**bottom right**), leading to a significant noise reduction (**inset center**).

Simulating the light transport of volumetric effects poses significant challenges and costs, especially in the presence of heterogeneous volumes. Generating stochastic paths for volume rendering involves multiple decisions, and previous works mainly focused on directional and distance sampling, where the volume scattering probability (VSP), i.e., the probability of scattering inside a volume, is indirectly determined as a byproduct of distance sampling. We demonstrate that direct control over the VSP can significantly improve efficiency and present an unbiased volume rendering algorithm based on an existing resampling framework for precise control over the VSP. Compared to previous state-of-the-art, which can only increase the VSP without guaranteeing to reach the desired value, our method also supports decreasing the VSP. We further present a data-driven guiding framework to efficiently learn and query an approximation of the optimal VSP everywhere in the scene without the need for user control. Our approach can

easily be combined with existing path-guiding methods for directional sampling at minimal overhead and shows significant improvements over the state-of-the-art in various complex volumetric lighting scenarios.

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

Additional Key Words and Phrases: scattering probability guiding, volume rendering

## ACM Reference Format:

Kehan Xu, Sebastian Herholz, Marco Manzi, Marios Papas, and Markus Gross. 2024. Volume Scattering Probability Guiding. *ACM Trans. Graph.* 1, 1 (September 2024), 17 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

## 1 INTRODUCTION

Stochastic sampling methods of light paths, like path tracing, are the dominant techniques used in offline rendering. However, rendering complex media like clouds, fog, or smoke in high fidelity presents major challenges. These methods need to simulate numerous scattering events within these heterogeneous volumes, which can lead to either high computation costs or high-frequency noise due to variance in the pixel estimates. One effective remedy is importance sampling, which improves sampling of the various decisions encountered during path construction by placing samples proportional to the integrand. Prior work focuses mainly on directional sampling, which determines continuation directions after scattering events, and distance sampling, which determines locations of these events along rays inside volumes. However, the *volume scattering probability* (VSP), which determines the probability of any scattering

Authors' addresses: Kehan Xu, ETH Zürich, Zürich, Switzerland, kehanxu@student.ethz.ch; Sebastian Herholz, Intel Corporation, Karlsruhe, Germany, sebastian.herholz@gmail.com; Marco Manzi, DisneyResearch|Studios, Zurich, Switzerland, marco.manzi@disneyresearch.com; Marios Papas, DisneyResearch|Studios, Zürich, Switzerland, marios.papas@disneyresearch.com; Markus Gross, DisneyResearch|Studios, Zürich, Switzerland and ETH Zürich, Zürich, Switzerland, grossm@inf.ethz.ch.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

© 2024 Association for Computing Machinery.

0730-0301/2024/9-ART \$15.00

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

occurring along the ray within a volume, has often been overlooked and treated as a secondary aspect of distance sampling.

This work proposes a framework for independent control of the VSP, setting it apart from most previous work. Specifically, we present a data-driven framework to effectively learn and query approximate optimal VSPs, and a distance sampling algorithm, based on insights on delta tracking and its connection to resampling provided by Wrenninge and Villemin [2020], to sample according to those learned VSPs. As a motivational example, consider Fig. 1 depicting an optically thin high-albedo heterogeneous fog in a jungle. Traditional distance sampling methods, such as delta tracking [Coleman 1968; Woodcock et al. 1965], implicitly set the VSP solely based on the local volume properties and overlook the illumination in the scene. This leads to a low VSP across this scene, and the thin fog gets under-sampled. By contrast, our approach significantly reduces variance by considering the illumination when setting the VSP, which, in this case, leads to more paths scattering within the volume.

We propose a data-driven framework that *guides* the VSP throughout the scene based on cached information about the scene’s illumination. This is a form of path guiding as it optimizes path sampling by steering paths towards areas of higher importance based on continuously refined estimates of the light transport quantities. In our framework, we derive the spatio-directional varying optimal VSP and determine the essential quantities needed for effective guidance. We cache the light transport estimates in a screen-space buffer for primary rays and a spatio-directional data structure for secondary rays for efficient learning and querying. Additionally, our method can easily be combined with existing state-of-the-art path guiding methods for directional sampling at minimal overhead.

In summary, our contribution is as follows:

- Derivations of the provably optimal VSP based on certain assumptions about the underlying estimators (Sec. 5.1).
- A path guiding framework to learn and efficiently access the optimal spatio-directional varying VSP during rendering (Sec. 7).
- An unbiased distance sampling method allowing us to control the VSP precisely. Our method thereby utilizes the insight by Wrenninge and Villemin [2020] that delta tracking can be re-interpreted as a resampling process (Sec. 5.3).
- We demonstrate how our VSP guiding approach can be combined with directional path guiding to improve rendering quality over the state-of-the-art (Sec. 8).

## 2 RELATED WORK

Simulating volumetric light or particle transport is a well-researched topic in the rendering and neutron transport communities. In the context of rendering, the high-dimensional volume rendering equation (Eq. 1) is solved using extended versions of the Monte Carlo-based path tracing algorithm [Kajiya 1986]. In this section, we summarize the related areas of previous works and refer interested readers to the in-depth report by Novak et al. [2018].

*Traditional Distance Sampling.* For simple volumes, such as homogeneous ones, the transmittance component can be importance

sampled analytically. For piecewise constant volumes, regular tracking [Leppänen 2010] iteratively samples each constant interval of the volume analytically, but becomes inefficient in high-frequency volumes. Alternatively, ray marching [Danskin and Hanrahan 1992; Perlin and Hoffert 1989] marches through the volume at fixed-size steps, but introduces bias.

Null-collision algorithms, such as delta tracking [Coleman 1968; Woodcock et al. 1965] which we discuss in more detail in Sec. 3.3.1, allow for both efficient and unbiased importance sampling of the transmittance of heterogeneous volumes by homogenizing them with fictitious particles. Galtier et al. [2013] presented a formulation of null-collision algorithms using the integral formulation of the radiative transfer equation (RTE) [Chandrasekhar 1960]. This formulation enabled the design of several weighted delta tracking algorithms (e.g., [Kutz et al. 2017; Szirmay-Kalos et al. 2017, 2018]) that sample distances with PDFs that are not necessarily proportional to transmittance. The framework from Miller et al. [2019] enables delta tracking to be combined with other methods using Multiple Importance Sampling (MIS) [Veitch and Guibas 1995].

*Transmittance Estimation.* Explicit transmittance estimation is needed when the light attenuation due to participating media between two points must be computed, as is the case when explicitly sampling light sources with next-event estimation.

While an analytic solution is available for homogeneous volumes, delta tracking can be applied as an unbiased binary transmittance estimator for heterogeneous volumes. Ratio tracking [Novák et al. 2014] reduces the variance due to the probabilistic termination in delta tracking by weighting the samples by the probability of continuing the walk. Residual ratio tracking [Novák et al. 2014] combines ratio tracking with a control variate to reduce variance further. The next flight estimator [Kutz et al. 2017] sums up the uncollided transport through the majorant medium at each random walk position. Recent methods [Georgiev et al. 2019; Kettunen et al. 2021] utilize Taylor expansion to build an unbiased transmittance estimator that randomly selects locations along the ray instead of null-scattering random walks.

*Specialized Distance Sampling.* While traditional distance sampling techniques importance sample the transmittance term, more sophisticated and specialized methods aim at enhancing distance sampling by incorporating geometrical or illumination-dependent terms into the sampling procedure.

The equiangular sampling technique introduced by Kulla and Fajardo [2012] proposes a specialized estimator for single-scattered direct light contributions such as point, spot, or area light sources.

The joint-importance sampling framework by Georgiev et al. [2013] extends the concept to double scattering, including the phase function, and Hanika et al. [2022] improves it by alleviating the need of a tabulated representation of the phase function. Körner et al. [2016] utilizes the joint-importance sampling framework to enable single-scattered next-event estimation through refractive surfaces.

The resampling method from Wrenninge and Villemin [2020] underpins our method and was originally proposed for importance sampling of the product between direct illumination and the transmittance. They generate candidate samples using delta tracking and resample according to weights proportional to the product. The key

difference to our work is their focus on the distance sampling within the volume, while our work focuses on the VSP (Sec. 5.4).

Huang et al. [2021] propose another related resampling-based approach to importance sample emissive volumes. Similar to our work, they propose a data-driven approach, but its large overhead limits its use to primary rays only.

*Data-driven Importance Sampling for Volumes.* Data-driven importance sampling approaches, such as path guiding, use approximations of a scene’s light transport, learned in a pre-processing step or online during rendering, to optimize sampling decisions. The earlier works by Pegoraro et al. [2008] and Bashford-Rogers et al. [2012] on volumetric path guiding focused on importance sampling directional sampling decisions based on learned approximation of the incoming radiance distribution, but were limited to (near) isotropic volumes. More general is the approach of Herholz et al. [2019] as it considers the product with the phase function, as well as the additional volumetric sampling decisions such as distance sampling, Russian roulette, and splitting, all based on the zero-variance theory [Hoogenboom 2008]. While the distance sampling proposed in their work implicitly yields similar optimal VSPs as our work, their method incurs a large overhead since it relies on regular tracking. Relevant to our work are directional path-guiding methods that use spatio-directional data structures to cache approximations of the incident radiance distribution. The methods of Müller et al. [2017] and Ruppert et al. [2020] use adaptive KD-trees to store incident radiance models per spatial voxel region. Ruppert et al. [2020] robustly fits the directional distributions for each leaf node using parametric mixture models. Our work extends this representation by adding the required quantities to estimate the optimal volume scattering probability per location for each lobe of the mixture models (Sec. 7).

*Volume Scattering Probability.* Closest related to our work is the method by Villemin et al. [2018] that modifies the distance sampling PDF to increase the volume scatter probability to render thin media more effectively. The method, however, requires the user to set the VSP per scene or volume manually, and in heterogeneous volumes, it does not always reach the desired VSP while also not supporting decreasing the VSP. In contrast, our approach is fully automatic and adapts to local illumination, always reaches the approximate optimal VSP anywhere in the scene and allows to also decrease the VSP. We discuss the differences in more detail in Sec. 8.5.

### 3 BACKGROUND

In this section, we recap the fundamentals of volume rendering relevant to our presented work, such as the volume rendering equation (Sec. 3.1), volumetric path tracing (Sec. 3.2), common transmittance-based distance sampling (Sec. 3.3.1) and transmittance estimation (Sec. 3.3.2). Experienced readers may skip to Sec. 4.

#### 3.1 Volume Rendering Equation

Rendering scenes containing volumes requires solving the Volume Rendering Equation (VRE). In its integral formulation, the VRE decomposes the incident radiance  $L$  arriving at a point  $\mathbf{x}$  from a

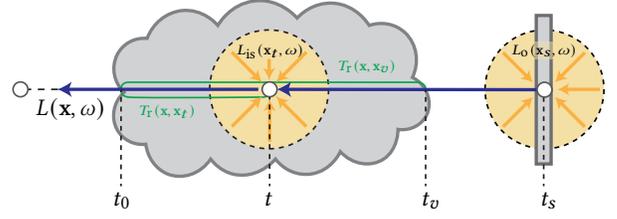


Fig. 2. The volume rendering equation (VRE) describing the radiance  $L(\mathbf{x}, \omega)$  involves the integral of the attenuated in-scattered radiance  $L_{is}(\mathbf{x}_t, \omega)$  along the volume and the outgoing surface radiance  $L_o(\mathbf{x}_s, \omega)$ .

direction  $\omega$  into a volume  $L_v$  and a surface contribution  $L_s$ :

$$L(\mathbf{x}, \omega) = \underbrace{\int_{t_0}^{t_v} \sigma_s(\mathbf{x}_t) T_r(\mathbf{x}, \mathbf{x}_t) L_{is}(\mathbf{x}_t, \omega) dt}_{L_v(\mathbf{x}, \omega)} + \underbrace{T_r(\mathbf{x}, \mathbf{x}_s) L_o(\mathbf{x}_s, \omega)}_{L_s(\mathbf{x}, \omega)}. \quad (1)$$

Fig. 2 visualizes the individual components of the VRE. The volume contribution  $L_v$  is defined by the integral of the attenuated in-scattered radiance  $L_{is}$  along a ray segment  $[t_0, t_v]$  covering the volume. The in-scattered radiance  $L_{is}$  at a point  $\mathbf{x}_t = \mathbf{x} + t\omega^1$  is the spherical integral of the incoming radiance  $L$  arriving at  $\mathbf{x}_t$  from a direction  $\omega'$ , that is scattered according to a phase function  $\rho$  in the outgoing direction  $\omega$ :

$$L_{is}(\mathbf{x}_t, \omega) = \int_{\Omega} \rho(\mathbf{x}_t, \omega, \omega') L(\mathbf{x}_t, \omega') d\omega'. \quad (2)$$

The attenuation of  $L_{is}$  between  $\mathbf{x}_t$  and  $\mathbf{x}$  is the product of the local scattering coefficient  $\sigma_s$  at  $\mathbf{x}_t$  and the volume transmittance  $T_r$  from  $\mathbf{x}_t$  to  $\mathbf{x}$ .

The transmittance function  $T_r(\mathbf{x}_1, \mathbf{x}_2)$  quantifies the attenuation of light between two points inside a volume due to absorption and out-scattering:

$$T_r(\mathbf{x}_1, \mathbf{x}_2) = \exp(-\tau(\mathbf{x}_1, \mathbf{x}_2)), \text{ with } \tau(\mathbf{x}_1, \mathbf{x}_2) = \int_{\mathbf{x}_1}^{\mathbf{x}_2} \sigma_t(\mathbf{x}) dx. \quad (3)$$

This function is defined by an exponential of *optical thickness*  $\tau$ , that is the integral of the extinction coefficient  $\sigma_t$  over the volume segment between  $\mathbf{x}_1$  and  $\mathbf{x}_2$ .

The surface contribution  $L_s$  is the outgoing radiance  $L_o$  from the first surface point  $\mathbf{x}_s$  behind or inside the volume that is attenuated by the volume. The outgoing radiance  $L_o$  is defined by the Rendering Equation (RE) [Kajiya 1986]:

$$L_o(\mathbf{x}, \omega) = L_e(\mathbf{x}, \omega) + \int_{\Omega} f_s(\mathbf{x}, \omega, \omega') \cos(\theta') L(\mathbf{x}, \omega') d\omega', \quad (4)$$

where  $L_e$  is the emitted radiance. The integral over the sphere is the incoming radiance  $L$  at  $\mathbf{x}$  from direction  $\omega'$ , scattered by the Bidirectional Scattering Distribution Function (BSDF)  $f_s$  into the outgoing direction  $\omega$ , and  $\theta'$  is the angle between the incoming radiance and the surface normal. The attenuation of  $L_o$  is defined

<sup>1</sup>For better readability of the equations throughout the text, we slightly abuse notation and denote positions corresponding to distances with a specific meaning as  $\mathbf{x}_k = \mathbf{x} + t_k\omega$  for  $k \in \{0, v, s\}$ .

by the transmittance  $T_r(\mathbf{x}, \mathbf{x}_s)$  between  $\mathbf{x}$  and  $\mathbf{x}_s$  which, under the assumption of a vacuum outside volumes, is equal to the volume transmittance  $T_r(\mathbf{x}_0, \mathbf{x}_v)$ .

### 3.2 Volumetric Path Tracing

The VRE typically lacks a closed-form solution and is commonly solved using a volumetric extension of the Monte Carlo-based path tracing algorithm [Kajiya 1986]. The algorithm numerically estimates the nested integrals of the VRE by tracing random paths through the scene, starting at  $\mathbf{x}$  in the direction  $\omega$ . The directions and distances of each path segment are, thereby, determined by the nested Monte Carlo estimators for the VRE's nested integrals.

Each path returns a Monte Carlo estimate  $\langle L(\mathbf{x}, \omega) \rangle$  of the incoming radiance which we can write as follows:

$$\langle L(\mathbf{x}, \omega) \rangle = \begin{cases} \frac{1}{P_{\text{vol}}} \langle L_v(\mathbf{x}, \omega) \rangle & \text{if } \xi < P_{\text{vol}}, \\ \frac{1}{1-P_{\text{vol}}} \langle L_s(\mathbf{x}, \omega) \rangle & \text{otherwise.} \end{cases} \quad (5)$$

To estimate  $L$ , the estimator first decides if the path should explore the volume contribution (i.e., calling  $\langle L_v \rangle$ ) or the surface contribution (i.e., calling  $\langle L_s \rangle$ ). This binary decision is driven by the volume scattering probability  $P_{\text{vol}}$ .

In the first case, when the volume contribution is picked, the estimator for  $L_v$  at the current vertex  $\mathbf{x}$  and for the direction  $\omega$  of the current path segment is:

$$\langle L_v(\mathbf{x}, \omega) \rangle = \frac{T_r(\mathbf{x}, \mathbf{x}_t) \sigma_s(\mathbf{x}_t) \langle L_{\text{is}}(\mathbf{x}_t, \omega) \rangle}{p_{\text{dist}}(t)}. \quad (6)$$

The estimator  $\langle L_v(\mathbf{x}, \omega) \rangle$  samples a random distance  $t$  according to some distance PDF  $p_{\text{dist}}$ , calculates the product of the transmittance  $T_r(\mathbf{x}, \mathbf{x}_t)$ , the local scattering coefficient  $\sigma_s$ , and an estimate of the in-scattered radiance  $L_{\text{is}}$  at  $\mathbf{x}_t$  and divides the result by  $p_{\text{dist}}$ .

In the second case, when the surface contribution is picked, the estimator  $\langle T_r(\mathbf{x}, \mathbf{x}_s) \rangle$  (Sec. 3.3.2) is used to evaluate the transmittance towards the next surface intersection  $\mathbf{x}_s$  which is multiplied by the estimated outgoing radiance  $L_o$ , emitted and reflected at  $\mathbf{x}_s$  in the direction  $\omega$ :

$$\langle L_s(\mathbf{x}, \omega) \rangle = \langle T_r(\mathbf{x}, \mathbf{x}_s) \rangle \langle L_o(\mathbf{x}_s, \omega) \rangle. \quad (7)$$

In both cases, the evaluations require calling the following nested estimators respectively for  $L_{\text{is}}$  and  $L_o$ :

$$\begin{aligned} \langle L_{\text{is}}(\mathbf{x}_t, \omega) \rangle &= \frac{\rho(\mathbf{x}_t, \omega, \omega') \langle L(\mathbf{x}_t, \omega') \rangle}{p(\omega')} \\ \langle L_o(\mathbf{x}_s, \omega) \rangle &= L_e(\mathbf{x}_s, \omega) + \frac{f_s(\mathbf{x}_s, \omega, \omega') \cos(\theta') \langle L(\mathbf{x}_s, \omega') \rangle}{p(\omega')}. \end{aligned} \quad (8)$$

Both estimators sample a random direction  $\omega'$  according to a directional PDF  $p(\omega')$  in which the path continues. The evaluation of the corresponding integrands from Eq. 2 and Eq. 4 trigger recursive calls of the estimator  $\langle L \rangle$  defined in Eq. 5.

Previous works mainly focused on importance sampling for directional decisions (in volumes or on surfaces) and distance sampling. The volume scattering probability  $P_{\text{vol}}$  is usually not driven explicitly but stems from transmittance-based distance sampling (see Sec. 3.3). This work focuses on optimizing  $P_{\text{vol}}$ , which can already lead to a significant variance reduction for  $\langle L \rangle$  (Sec. 8.3) while still

using common importance sampling strategies (e.g., transmittance-based distance sampling) for the remaining decisions.

### 3.3 Distance Sampling, Volume Scattering and Transmittance Evaluation

Deciding whether and where to scatter within a volume and estimating the transmittance between two points are crucial components for solving the VRE. We will examine these in more detail in the following.

**3.3.1 Distance Sampling and Scattering Probabilities.** Ideally, a distance  $t$  should be sampled according to a PDF  $p_{\text{dist}}(t)$  that is proportional to the integrand of  $L_v$  in Eq. 1. Unfortunately, this would require prior knowledge about the global light transport quantity  $L_{\text{is}}$  for all positions  $\mathbf{x}_t$  and outgoing directions  $\omega$  inside the volume, which is typically unknown. Instead, traditional distance sampling strategies focus on parts of the integrand that depend on known volume quantities (e.g.,  $\sigma_t$ ) and, therefore, sample  $t$  with a PDF  $p_{\text{dist}}$  that is equal to the product of the local extinction coefficient and transmittance:

$$p_{\text{dist}}(t) = \sigma_t(\mathbf{x}_t) T_r(\mathbf{x}, \mathbf{x}_t). \quad (9)$$

These distance sampling strategies are not bounded to the range of the volume and can sample distances behind the volume, leading to scattering events on the closest surface intersection  $\mathbf{x}_s$ . This feature leads to an implicit definition of the volume scattering probability  $P_{\text{vol}}$  that depends on the transmittance from  $\mathbf{x}$  to  $\mathbf{x}_s$ :

$$P_{\text{vol}} = \int_{t_0}^{t_v} \sigma_t(\mathbf{x}_t) T_r(\mathbf{x}, \mathbf{x}_t) dt = 1 - T_r(\mathbf{x}_0, \mathbf{x}_v), \quad (10)$$

while  $t_v$  is either the distance to the end of the volume or to the next surface interaction if the surface is inside the volume ( $\mathbf{x}_v = \mathbf{x}_s$ ). The probability  $P_{\text{surf}}$  for the next scattering event being on the closest surface is  $P_{\text{surf}} = 1 - P_{\text{vol}}$ . Throughout this paper, we will call strategies resulting in Eq. 9 and Eq. 10 *transmittance-based* distance sampling strategies.

For homogeneous volumes, where  $\sigma_t$  is constant, sampling a distance  $t$  (starting at  $t_0$ ) proportional to transmittance can be achieved analytically using the inverse CDF method:

$$t = -\frac{\ln(1 - \xi)}{\sigma_t}, \quad (11)$$

where  $\xi$  is a uniform random number in the range of  $[0, 1)$ .

**Delta Tracking.** For heterogeneous volumes, the inverse CDF method is typically not applicable to sample according to Eq. 9. Instead, delta tracking [Woodcock et al. 1965] is used, which first *homogenizes* the heterogeneous volume by filling it with so-called *null-collision particles*  $\sigma_\eta$  with a majorant density of  $\bar{\sigma}$ :

$$\bar{\sigma} = \sigma_\eta(\mathbf{x}) + \sigma_t(\mathbf{x}). \quad (12)$$

Afterward, the algorithm steps through the *homogenized* volume by sampling distances using the CDF method based on the majorant density (see Eq. 11), generating so-called event candidates. Each candidate can either be a null-scattering or a real-scattering event. Null-scattering events do not change the direction or the amount of light that passes through the volume, while real-scattering events

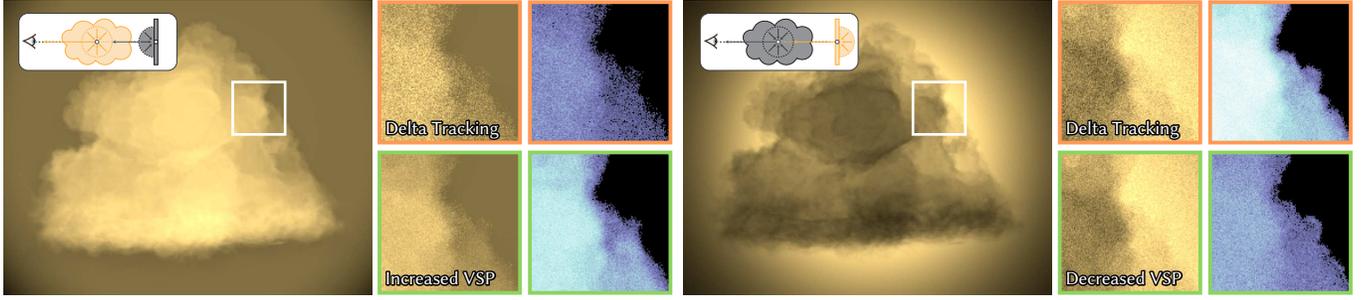


Fig. 3. Motivational example comparing how changing the volume scattering probability (VSP) affects the variance in images. We discuss these examples in detail in Sec. 4. Increasing the VSP in the **left** image and decreasing it in the **right** image lead to a significant variance reduction in both cases.

leads to its absorption and out-scattering. At each candidate position  $\mathbf{x}_i$  the scattering event type is selected based on the two probabilities:

$$P_{\text{real}}(\mathbf{x}_i) = \frac{\sigma_t(\mathbf{x}_i)}{\sigma} \quad \text{and} \quad P_{\text{null}}(\mathbf{x}_i) = 1 - P_{\text{real}}(\mathbf{x}_i). \quad (13)$$

If a null-scattering event is selected, the process continues sampling the distance to the next candidate position, and if a real-scattering event is selected, the process stops. The current position  $\mathbf{x}_i$  is returned as the position for the next volume scattering event. If no real-scattering event is selected before the procedure reaches the end of the volume or steps over a surface inside the volume, the next scattering event will be on the closest surface behind or inside the volume. Later in Sec. 5.3, we describe a reformulation of delta tracking under a resampling framework that was inspired by the insights of Wrenninge and Villemin [2020], which forms the basis of our methodology.

**3.3.2 Transmittance Evaluation.** In heterogeneous volumes, evaluating the transmittance requires solving the *optical thickness* (Eq. 3) between the two points  $\mathbf{x}_0$  and  $\mathbf{x}_1$ . Usually the *optical thickness* cannot be evaluated analytically and the transmittance is therefore estimated.

The delta tracking algorithm can be used as a binary estimator for this quantity: If the procedure generates a volume sample, the estimator returns 0; if it generates a surface sample, it returns 1.

The ratio tracking estimator [Novák et al. 2014] is a lower variance non-binary transmittance estimator; Instead of making stochastic scattering or termination decisions at each sampled location, it deterministically continues until the sample is past the volume or at the first surface. This procedure results in candidates at location  $\mathbf{x}_1, \dots, \mathbf{x}_n$  where the distance to  $\mathbf{x}_n$  is smaller or equal to the distance to  $\mathbf{x}_t$ . The ratio tracking estimator  $\langle T_r \rangle_{\text{ratio}}$  is then the product of the null-scattering probabilities  $P_{\text{null}}$  of all candidates:

$$\langle T_r(\mathbf{x}, \mathbf{x}_t) \rangle_{\text{ratio}} = \prod_{i=1}^n P_{\text{null}}(\mathbf{x}_i). \quad (14)$$

Later, in Secs. 5.3 and 5.4, this definition of the ratio tracking estimator is used to reformulate delta tracking as a resampling process and to derive our new distance sampling algorithm that can reach arbitrary VSP values.

## 4 MOTIVATION AND GOAL

The transmittance-based distance sampling method discussed in the previous section (Sec. 3.3.1) does only consider local volume properties and neglects the global information, such as the contributions of the nested estimators.

It is, therefore, unable to optimally importance sample all terms in Eq. 5. In Fig. 3, we show on two simple examples how considering the contribution of the nested estimators when sampling the VSP can be beneficial.

The first example (Fig. 3, **left**) shows an optically thin and brightly lit volume in front of a dark surface. In this case, the volume sampling probability  $P_{\text{vol}}$  is small. Thus, only a small fraction of the camera rays actually scatter in the volume. However, the contribution of the total radiance estimator  $\langle L(\mathbf{x}, \omega) \rangle$  is dominated by the contribution of the volume  $\langle L_{\text{is}}(\mathbf{x}, \omega) \rangle$ , thus increasing  $P_{\text{vol}}$  reduces the variance of the rendered image.

The second example (Fig. 3, **right**) shows an optically thick and dimly lit volume in front of a back-lit rough glass wall. In this case,  $P_{\text{vol}}$  is large. Thus, only a small fraction of camera rays pass through the volume to scatter on the back-lit surface. However, the contribution of the total radiance estimator  $\langle L(\mathbf{x}, \omega) \rangle$  is dominated by the contribution of the surface contribution estimator  $\langle L_{\text{o}}(\mathbf{x}, \omega) \rangle$ , thus decreasing  $P_{\text{vol}}$  reduces the variance of the rendered image.

Note that we guide the volume sampling probability in both examples following the algorithm presented in Sec. 5.4.

*Goals of Our Method.* We aim to design a practical and widely applicable VSP guiding method that reduces variance and is fully automatic, lightweight, and robust.

By deriving our optimal VSP values from optimal sampling theories (Sec. 5.1), our VSP guiding framework does not require manual tuning of hyperparameters. Our method is integrated into a path-guiding framework, which we modified to automatically provide continuously improved optimal VSP estimates to our distance sampling routines (Sec. 7).

To keep our approach lightweight, we focus on VSP guiding instead of product distance guiding [Herholz et al. 2019] since the former avoids the need to query information about the in-scattered radiance  $L_{\text{is}}$  along the ray during sampling.

Finally, to make our method robust, we need to ensure that our method does not lead to a variance increase when compared to traditional transmittance-based distance sampling. We, therefore,

designed a *scaled transmittance-based* distance sampling strategy (Sec. 5.2) that closely aligns the sampling distribution to be proportional to the one from transmittance-based methods while allowing flexible adjustment of the resulting volume sampling probability.

## 5 METHODOLOGY

In this section, we outline our algorithm of volume scattering probability guiding (VSPG or VSP guiding). We first define the optimal volume scattering probability (VSP) (Sec. 5.1) and how we want this to affect distance sampling (Sec. 5.2). Next, we formulate delta tracking as a resampling algorithm that allows us to modify the distance sampling distribution through resampling weights (Sec. 5.3). This serves as a basis to finally describe a resampling algorithm that achieves distance sampling with the desired properties and enables us to reach an arbitrary VSP (Sec. 5.4). We analyze the properties of the resulting distribution in Sec. 6 and the integration of the method within a VSP guiding framework in Sec. 7. An extension of the resampling algorithm for chromatic media is described in Appendix A.

### 5.1 Optimal Volume Scattering Probability

In the following, we derive two optimal formulations for  $P_{\text{vol}}$  given different assumptions about the underlying estimators. The corresponding probabilities are denoted as  $P_{\text{vol}}^{\text{1st}}$  and  $P_{\text{vol}}^{\text{2nd}}$  for being related to the first and second moments of the estimators.

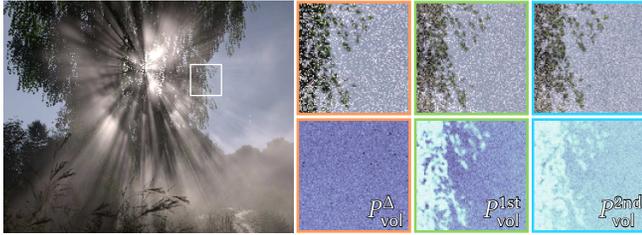


Fig. 4. Equal-sample comparison (64 spp) on LANDSCAPE of no guiding (left) to VSPG using the contribution-based criterion (Sec. 5.1.1) (middle) and the variance-based criterion (Sec. 5.1.2) (right). The resulting  $P_{\text{vol}}$  is shown at the bottom. Variance-based criterion detects the need for a higher  $P_{\text{vol}}$  than contribution-based to eliminate the overall noise.

**5.1.1 Contribution-based Criterion.** The optimal volume and surface sample distribution under the zero-variance framework should be proportional to the respective contribution values (i.e., first moments)  $L_v(\mathbf{x}, \omega)$  and  $L_s(\mathbf{x}, \omega)$  [Herholz et al. 2019]. In this framework, the recursive estimators for  $L_v(\mathbf{x}, \omega)$  and  $L_s(\mathbf{x}, \omega)$  are assumed to have zero variance, and the optimal VSP is:

$$P_{\text{vol}}^{\text{1st}} = \frac{L_v(\mathbf{x}, \omega)}{L_v(\mathbf{x}, \omega) + L_s(\mathbf{x}, \omega)} = \frac{L_v(\mathbf{x}, \omega)}{L(\mathbf{x}, \omega)}. \quad (15)$$

**5.1.2 Variance-based Criterion.** Achieving zero-variance sampling for complex integrals like those in the VRE is often not possible in practice, and thus, the assumptions of the zero-variance framework are typically violated. However, it is straightforward to derive the optimal VSP for nested estimators with non-zero variance [Rath et al. 2020]:

$$P_{\text{vol}}^{\text{2nd}} = \frac{\sqrt{M_{\text{vol}}}}{\sqrt{M_{\text{vol}}} + \sqrt{M_{\text{surf}}}}, \quad (16)$$

where  $M_{\text{vol}}$  and  $M_{\text{surf}}$  are the second moments of the volume and surface contribution estimators:

$$M_{\text{vol}} = \mathbb{E}[\langle L_v(\mathbf{x}, \omega) \rangle^2] = \int_{t_0}^{t_v} \frac{\langle T_r(\mathbf{x}, \mathbf{x}_t) \sigma_s(\mathbf{x}_t) \langle L_{is}(\mathbf{x}_t, \omega) \rangle \rangle^2}{p(\mathbf{x}_t)} d\mathbf{x}_t, \quad (17)$$

$$\begin{aligned} M_{\text{surf}} &= \mathbb{E}[\langle L_s(\mathbf{x}, \omega) \rangle^2] = \mathbb{E}[\langle T_r(\mathbf{x}, \mathbf{x}_s) \rangle^2] \mathbb{E}[\langle L_o(\mathbf{x}_s, \omega) \rangle^2] \\ &= \mathbb{E}[\langle T_r(\mathbf{x}, \mathbf{x}_s) \rangle^2] \int_{\Omega} \frac{\langle f_s(\mathbf{x}_s, \omega, \omega') \cos(\theta') \langle L(\mathbf{x}_s, \omega') \rangle \rangle^2}{p(\omega')} d\omega'. \end{aligned} \quad (18)$$

For simplicity's sake, we dropped the emission term of  $L_o$  in Eq. 18.

In practice, we do not know the exact values of the contributions  $L_v(\mathbf{x}, \omega)$ ,  $L_s(\mathbf{x}, \omega)$  or the second moments  $M_{\text{vol}}$ ,  $M_{\text{surf}}$  and need to estimate them. Details on the computation of these estimates are provided in Sec. 7. To simplify the discussion in the remaining sections, we unify the notation of the optimal VSP to  $P_{\text{vol}}^*$ , which can be replaced by  $P_{\text{vol}}^{\text{1st}}$  or  $P_{\text{vol}}^{\text{2nd}}$  unless noted otherwise. Fig. 4 provides an example of the comparison between the two criteria.

Note that, in both variants, the optimal VSP does not require knowledge of the distribution of  $L_{is}$  along the ray, but only of related quantities that are integrated over the ray. This allows for efficient caching and querying of the related quantities as discussed in Sec. 7.

### 5.2 Scaled Transmittance-based Distance Sampling

In our proposed framework, we focus on guiding the VSPs to follow the optimal ones defined in Sec. 5.1. At the same time, however, we aim to distribute samples *within* the volume proportional to the product of transmittance and the local extinction coefficient, like in delta tracking. We achieve this with a scaled distance sampling PDF of:

$$p_{\text{dist}}(t) = C \sigma_t(\mathbf{x}_t) T_r(\mathbf{x}, \mathbf{x}_t) \quad (19)$$

with constant  $C$  such that:

$$P_{\text{vol}}^* = \int_{t_0}^{t_v} C \sigma_t(\mathbf{x}_t) T_r(\mathbf{x}, \mathbf{x}_t) dt = C(1 - T_r(\mathbf{x}, \mathbf{x}_v)). \quad (20)$$

Developing a distance sampling scheme for homogeneous volumes that adheres to this  $p_{\text{dist}}(t)$  using the inverse CDF method is straightforward. Based on  $P_{\text{vol}}^*$ , we first decided whether to scatter inside the volume or not. If so, a distance between  $t_0$  and  $t_v$  can be sampled analytically based on transmittance according to Eq. 9 in [Kulla and Fajardo 2012]. However, achieving this distance sampling PDF with a heterogeneous medium is more challenging. For this purpose, we utilize the resampling framework described in the next section.

### 5.3 Delta Tracking as a Resampling Method

As Wrenninge and Villemin [2020] pointed out, delta tracking can be reformulated as a resampling method.

In the first step, ratio tracking is used to step through the volume or to the next surface intersection to generate a transmittance estimate  $\langle T_r(\mathbf{x}_0, \mathbf{x}_s) \rangle_{\text{ratio}}$ . Unlike normal ratio tracking, we keep the event candidates for further reuse. This way we obtain  $M - 1$  event candidates at  $\mathbf{x}_1, \dots, \mathbf{x}_{M-1}$  inside the volume and one surface sample

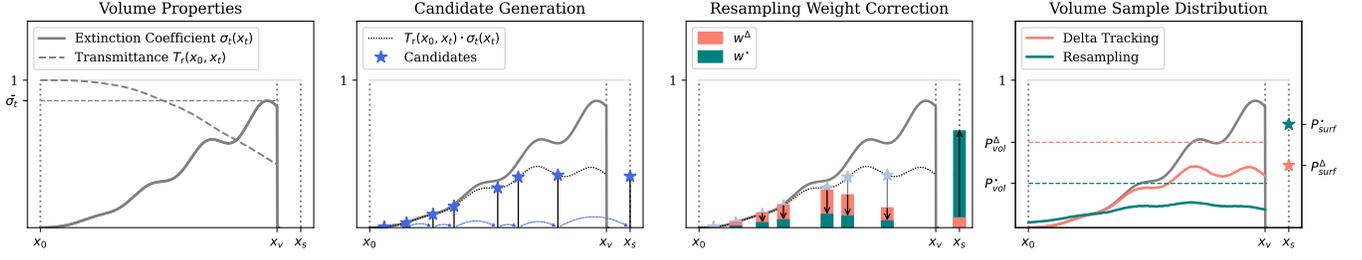


Fig. 5. Visualization of the resampling procedure. **From left to right:** (1) a simple 1D slice of a heterogeneous volume; (2) we generate candidate samples using ratio tracking and use them to estimate transmittance; (3) we then scale the delta tracking resampling weight  $w^\Delta$  to become  $w^*$  (Eq. 24); (4) we resample a single candidate from these event candidates according to the scaled weights  $w^*$ ; the resulting sample distributions obtained from using  $w^\Delta$  and  $w^*$  are illustrated. Note that our proposed algorithm reaches the desired volume scattering probability precisely.

$\mathbf{x}_M = \mathbf{x}_s$ . Further, we associate each candidate with a resampling weight:

$$w^\Delta(\mathbf{x}_i) = P_{\text{real}}(\mathbf{x}_i) \left( \prod_{j=1}^{i-1} P_{\text{null}}(\mathbf{x}_j) \right) \quad (21)$$

for the volume candidates, and:

$$w^\Delta(\mathbf{x}_M) = \prod_{j=1}^{M-1} P_{\text{null}}(\mathbf{x}_j) = \langle T_r(\mathbf{x}_0, \mathbf{x}_s) \rangle_{\text{ratio}} \quad (22)$$

for the surface candidate. The need to store all the candidates explicitly is avoided by using reservoir sampling; this is elaborated in Sec. 5.5.

In the second step, we pick a single sample from the set of  $M$  candidates proportionally to the resampling weights. The resampling weights define a valid probability density over the candidates for resampling since  $\sum_{i=1}^M w^\Delta(\mathbf{x}_i) = 1$ . The resulting sample distribution and VSP, denoted by  $p_{\text{dist}}^\Delta(\mathbf{x}) = w^\Delta(\mathbf{x})$  and  $P_{\text{vol}}^\Delta$ , are by construction identical to delta tracking. This resampling formulation allows us to adjust the sampling distribution and the resulting VSP by modifying the resampling weights.

#### 5.4 Modify the Resampling Weight

Next, we adjust the resampling weights of the above resampling algorithm to achieve the desired VSP  $P_{\text{vol}}^*$  while keeping the sample distribution within the volume approximately proportional to the product of transmittance and the local extinction coefficient (Eq. 9). We propose to use two constant weight scaling factors, one applied to all volume candidate weights ( $C_{\text{vol}}$ ), and another applied to the single surface candidate weight ( $C_{\text{surf}}$ ):

$$C_{\text{vol}} = \frac{P_{\text{vol}}^*}{1 - \langle T_r(\mathbf{x}_0, \mathbf{x}_s) \rangle_{\text{ratio}}}, \quad C_{\text{surf}} = \frac{1 - P_{\text{vol}}^*}{\langle T_r(\mathbf{x}_0, \mathbf{x}_s) \rangle_{\text{ratio}}}. \quad (23)$$

Here  $\langle T_r(\mathbf{x}_0, \mathbf{x}_s) \rangle_{\text{ratio}}$  is the ratio tracking estimator obtained from the volume candidates during the first step of the resampling algorithm. The scaled resampling weights are thus:

$$w^*(\mathbf{x}_i) = \begin{cases} C_{\text{vol}} w^\Delta(\mathbf{x}_i) & \text{volume sample } \mathbf{x}_i (1 \leq i < M), \\ C_{\text{surf}} w^\Delta(\mathbf{x}_i) & \text{surface sample } \mathbf{x}_M. \end{cases} \quad (24)$$

With  $\sum_{i=1}^M w^*(\mathbf{x}_i) = 1$ , these scaled resampling weights still define a valid probability density. It is straightforward to show that

using those resampling weights instead of those defined in Eq. 21 and Eq. 22 changes the probability of sampling a volume event after the resampling step from  $1 - T_r(\mathbf{x}_0, \mathbf{x}_s)$  to the desired  $P_{\text{vol}}^*$ . A visualization of the resampling procedure is provided in Fig. 5.

#### 5.5 Practical Considerations

In the following, we discuss multiple approaches to optimize our previously introduced resampling method for guiding the VSP.

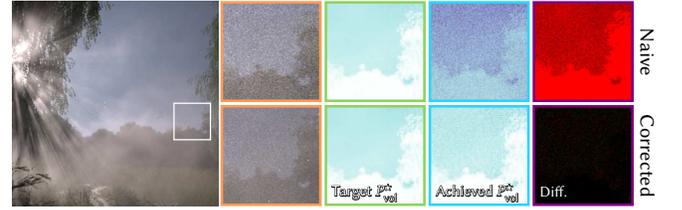


Fig. 6. Equal-sample comparison (64spp) on LANDSCAPE without (**top**) and with (**bottom**) zero volume event candidate correction. **Green** shows the target  $P_{\text{vol}}^*$ , **blue** the achieved  $P_{\text{vol}}^*$ , and **purple** the difference between achieved VSP and target  $P_{\text{vol}}^*$ .

*Zero Volume Event Candidate.* The number of scatter candidates to resample from cannot be controlled directly since their generation process is ratio tracking. In the occasional case when the ratio tracking estimator samples the first distance to be greater than  $t_v$ , no volume scatter candidate will be created. The probability of such an event occurring depends on the majorant of the volume  $\bar{\sigma}$  and the distance to the volume boundary  $t_v$  and is equal to majorant transmittance:  $\exp(-\bar{\sigma}t_v)$ . Whenever this happens, the probability to sample the volume is 0 and not  $P_{\text{vol}}^*$ . The resulting  $P_{\text{vol}}$  from our algorithm is thus:

$$P_{\text{vol}} = \begin{cases} 0 & \text{with probability } \exp(-\bar{\sigma}t_v), \\ P_{\text{vol}}^* & \text{with probability } 1 - \exp(-\bar{\sigma}t_v). \end{cases}$$

which, in expectation, yields  $\mathbb{E}[P_{\text{vol}}] = P_{\text{vol}}^*(1 - \exp(-\bar{\sigma}t_v))$ . In practice, this means the resulting volume scattering probability will be lower than the desired  $P_{\text{vol}}^*$ .

We have two ways to improve this behavior; either by increasing the majorant  $\bar{\sigma}$  leading to decreasing  $\exp(-\bar{\sigma}t_v)$  or by increasing

$P_{\text{vol}}^*$  by a factor of  $1/(1 - \exp(-\bar{\sigma}t_v))$ . The first approach increases the number of scatter candidates and thus the cost of our resampling algorithm, while the second one is limited to situations when  $P_{\text{vol}}^*/(1 - \exp(-\bar{\sigma}t_v)) \leq 1$ .

Our proposed solution combines both ideas by increasing the majorant  $\bar{\sigma}$  as little as possible, such that the second approach becomes applicable. In detail, we first increase the majorant to:

$$\bar{\sigma}' = \max\left(\bar{\sigma}, -\ln(1 - P_{\text{vol}}^*)/t_v\right), \quad (25)$$

and then we increase the volume scattering probability used for computing  $C_{\text{vol}}$  and  $C_{\text{surf}}$  (Eq. 23) to:

$$P_{\text{vol}}^{\star'} = P_{\text{vol}}^*/(1 - \exp(-\bar{\sigma}'t_v)), \quad (26)$$

such that:  $\mathbb{E}[P_{\text{vol}}] = P_{\text{vol}}^{\star'}(1 - \exp(-\bar{\sigma}'t_v)) = P_{\text{vol}}^*$ . This ensures that  $P_{\text{vol}}^{\star'} \leq 1$  and therefore allows us to always reach the desired  $P_{\text{vol}}^*$ . We show the effect of this addition to our algorithm in Fig. 6.

*Defensive resampling.* As we will discuss in Sec. 7, we have to estimate  $P_{\text{vol}}^*$  during rendering. Unfortunately, we can not bound the estimation error of  $P_{\text{vol}}^*$ , and, in rare cases, bad estimates can lead to our resampling algorithm performing worse than naive delta tracking. We mitigate this by interpolating between the scaled and original resampling weights, i.e.:

$$w^\alpha = \alpha w^* + (1 - \alpha)w^\Delta \quad (27)$$

where  $\alpha$  is a user-defined weight that controls how closely we want to sample according to  $P_{\text{vol}}^*$ . This corresponds to using Multiple Importance Sampling (MIS) with the one-sample model using the balance heuristic [Veach and Guibas 1995]. The resulting VSP is  $P_{\text{vol}}^\alpha = \alpha P_{\text{vol}}^* + (1 - \alpha)P_{\text{vol}}^\Delta$ . We found  $\alpha = 0.75$  to work well in all our experiments.<sup>2</sup>

*Reservoir Sampling.* To bound memory consumption of our resampling method, we utilize weighted reservoir sampling [Chao 1982] instead of storing all candidates explicitly before resampling. Reservoir sampling processes elements in order from an input stream of  $M$  candidates and stores a *reservoir* of  $N$  samples ( $N = 1$  in our case) from these candidates. Each incoming element  $x_i$  with weight  $w(x_i)$  replaces the sample in the reservoir with the replacement probability  $P_{\text{res}} = w(x_i)/\sum_{j=1}^i w(x_j)$ .

For our method, the resampling weight  $w^*(x_i)$  (Eq. 24) includes either  $C_{\text{vol}}$  or  $C_{\text{surf}}$ . Both can be calculated only after finishing sampling all the event candidates  $\{x_1, \dots, x_M\}$ . While the factor  $C_{\text{vol}}$  is unknown during reservoir sampling for volume event candidates, it is shared among all volume candidate resampling weights and cancels out in the replacement probability.

After streaming all the volume candidates, the final decision is between the volume candidate  $x_j$  inside the reservoir and the surface candidate  $x_M$ . The denominator in  $P_{\text{res}}$  is then 1 which means the final replacement probability is simply  $C_{\text{surf}}w^\Delta(x_M) = w^*(x_M)$ .

The pseudocode in Alg. 1 shows our resampling approach using reservoir sampling, defensive resampling, and the zero volume event candidate correction.

<sup>2</sup>Nevertheless, lower values like  $\alpha = 0.5$  might be beneficial in high-variance setups.

---

**ALGORITHM 1:** Volume Scattering Probability Guiding

---

```

1 Function VSPG( $\bar{\sigma}, t_v, P_{\text{vol}}^*, \alpha$ ):
2   Reservoir r
3    $t \leftarrow 0, \langle T_r \rangle_{\text{ratio}} \leftarrow 1, w_{\text{sum}} \leftarrow 0$ 
4    $\bar{\sigma}', P_{\text{vol}}^{\star'} \leftarrow \text{ZeroVolumeCandidateCompensation}(\bar{\sigma}, t_v, P_{\text{vol}}^*, \alpha)$ 
5   while true do
6      $t \leftarrow t - \frac{\ln(1-\xi)}{\bar{\sigma}'}$  // Distance sampling, Eq. 11
7      $x_i \leftarrow x + t\omega$  // Generate a volume candidate
8     if  $t \geq t_v$  then
9       break
10     $\langle T_r \rangle_{\text{ratio}} \leftarrow P_{\text{null}}(x_i)\langle T_r \rangle_{\text{ratio}}$ 
11     $w_{\text{sum}} \leftarrow w_{\text{sum}} + w^\Delta(x_i)$  // Eq. 21
12     $r.\text{update}(x_i, \frac{w^\Delta(x_i)}{w_{\text{sum}}})$ 
13  end
14   $x_M \leftarrow x + t_v\omega$  // Generate the surface candidate
15  /* Defensive resampling */
16   $w_{\text{sum}}^\alpha \leftarrow \alpha(1 - P_{\text{vol}}^{\star'}) + (1 - \alpha)w_{\text{sum}}$  // Eq. 27
17   $w^\alpha(x_M) \leftarrow \alpha P_{\text{vol}}^{\star'} + (1 - \alpha)w^\Delta(x_M)$  // Eq. 22, Eq. 27
18   $w_{\text{sum}}^\alpha \leftarrow w_{\text{sum}}^\alpha + w^\alpha(x_M)$ 
19   $r.\text{update}(x_M, \frac{w^\alpha(x_M)}{w_{\text{sum}}^\alpha})$  //  $w_{\text{sum}}^\alpha = 1$ 
20  /* Set path segment throughput */
21   $P_{\text{vol}}^\alpha \leftarrow \alpha P_{\text{vol}}^{\star'} + (1 - \alpha)(1 - \langle T_r \rangle_{\text{ratio}})$  // Resulting VSP
22   $r.T_p \leftarrow \frac{1 - \langle T_r \rangle_{\text{ratio}}}{P_{\text{vol}}^\alpha}$  or  $\frac{\langle T_r \rangle_{\text{ratio}}}{1 - P_{\text{vol}}^\alpha}$  // Eq. 28
23  return  $r$ 
24 Function ZeroVolumeCandidateCompensation( $\bar{\sigma}, t_v, P_{\text{vol}}^*, \alpha$ ):
25   $\bar{\sigma}' \leftarrow \max(\bar{\sigma}, -\ln(1 - P_{\text{vol}}^*)/t_v)$  // Eq. 25
26   $P_{\text{vol}}^{\star'} \leftarrow P_{\text{vol}}^*/(1 - \exp(-\bar{\sigma}'t_v))$  // Eq. 26
27  return  $\bar{\sigma}', P_{\text{vol}}^{\star'}$ 

```

---

## 6 SAMPLE DISTRIBUTION ANALYSIS

The resampling algorithm aims to achieve a VSP of  $P_{\text{vol}}^*$  while keeping the relative distribution of samples *within* the volume as proportional as possible to delta tracking. We will discuss if and when this is achieved in the following section.

We first analyze the path segment's throughput from our proposed resampling scheme. As opposed to delta tracking, which has a constant path segment throughput of 1, our sampling scheme yields the following:

$$\frac{p_{\text{dist}}^\Delta(x_i)}{w^*(x_i)} \sum_j w^*(x_j) = \begin{cases} \frac{1 - \langle T_r(x_0, x_s) \rangle_{\text{ratio}}}{P_{\text{vol}}^*} & \text{for } x_i (1 \leq i < M), \\ \frac{\langle T_r(x_0, x_s) \rangle_{\text{ratio}}}{1 - P_{\text{vol}}^*} & \text{for } x_M. \end{cases} \quad (28)$$

This follows from  $\sum_{i=1}^M w^*(x_i) = 1$  and the definitions in Eq. 24 and Eq. 23. Examining this expression shows that the ratio tracking transmittance estimate  $\langle T_r(x_0, x_s) \rangle_{\text{ratio}}$  is not canceled out. This has two consequences:

First, unlike in delta tracking, the throughput itself becomes noisy (see Fig. 7, **bottom-left**). However, our experimental results in Sec. 8.3 imply that transmittance typically is not a dominating source of noise (i.e., variance) in the renderings. Thus importance sampling  $\langle L_o(x, \omega) \rangle$  and  $\langle L_{\text{is}}(x, \omega) \rangle$  through better VSP sampling usually still decreases the variance.

Second, using our proposed resampling procedure will slightly flatten the distribution of the samples within the volume. The underlying reason for this behavior lies in the correlation between the resampled volume location  $\mathbf{x}_t$  and the transmittance estimate  $\langle T_r(\mathbf{x}_0, \mathbf{x}_s) \rangle_{\text{ratio}}$  used in the weights  $w^*$  (Eq. 23), which stems from sharing the event candidates for resampling and transmittance estimation. This correlation causes the transmittance estimates  $\langle T_r(\mathbf{x}_0, \mathbf{x}_s) \rangle_{\text{ratio}}$  at any specific resampled location  $\mathbf{x}_t$  to be biased. Note that the unbiasedness of our algorithm is guaranteed by the transmittance estimates being unbiased when integrated over all resampled locations. To see this on a trivial example, suppose a resampled location  $\mathbf{x}_i$  has  $\sigma_t(\mathbf{x}_i) = \bar{\sigma}$ . Then any ratio tracking estimator (Eq. 14) for  $T_r(\mathbf{x}_0, \mathbf{x}_s)$  that includes this candidate will be zero since  $P_{\text{null}}(\mathbf{x}_i) = 0$ . This clearly underestimates the transmittance, as the volume is heterogeneous. Fig. 7 shows this behavior on a more complex example. This correlation leads to over-estimation in low-density regions and under-estimation in high-density regions (Fig. 7, **top-right**). Since the throughput is multiplied by  $1 - \langle T_r(\mathbf{x}_0, \mathbf{x}_s) \rangle_{\text{ratio}}$ , high volume density regions correspond to high throughput and low volume density regions to low throughput (Fig. 7, **bottom-left**). As our total estimate of  $L$  is unbiased, the sampling density inside the volume has to decrease in regions with higher throughput and vice versa (Fig. 7, **bottom-right**).

The distribution's flattening correlates with the average number of candidates from the resampling procedure. A simple way to increase the number of candidates is to increase the majorant  $\bar{\sigma}$  used for candidate generation. This simultaneously reduces the variance in our transmittance estimate due to higher quality transmittance estimates and reduces the flattening of the sample distribution in the volume. In the limit, the variance in the throughput tends to zero, and the distribution of volume samples becomes proportional to the one from delta tracking (Fig. 8). However, since increasing the number of candidates also increases the cost of our sampling procedure and we did not observe obvious negative effects due to the flattening in our evaluation, a careful analysis of this trade-off seems warranted. We leave this as future work.

## 7 VOLUME SCATTERING PROBABILITY GUIDING (VSPG) FRAMEWORK

Previous sections derived and analyzed our VSP-driven distance sampling algorithm, which can achieve desired VSPs while still being approximately proportional to the product of transmittance and local extinction. The following section explains the data structure we use to represent and query the optimal VSP values for each position and direction in a scene. These two pieces combined to build our VSP guiding framework. The data structure consists of two components: an image space auxiliary VSP buffer (Sec. 7.1) for guiding the VSP for primary camera rays and a 5-dimensional spatio-directional VSP data structure (Sec. 7.2) for guiding the VSPs for secondary rays.

### 7.1 Primary Ray Image Space VSP Buffers

To estimate  $P_{\text{vol}}^*$  for all primary camera rays, we store estimates of the required quantities, like  $L_v$  (Eq. 6),  $L_s$  (Eq. 7),  $M_{\text{vol}}$  (Eq. 17), or

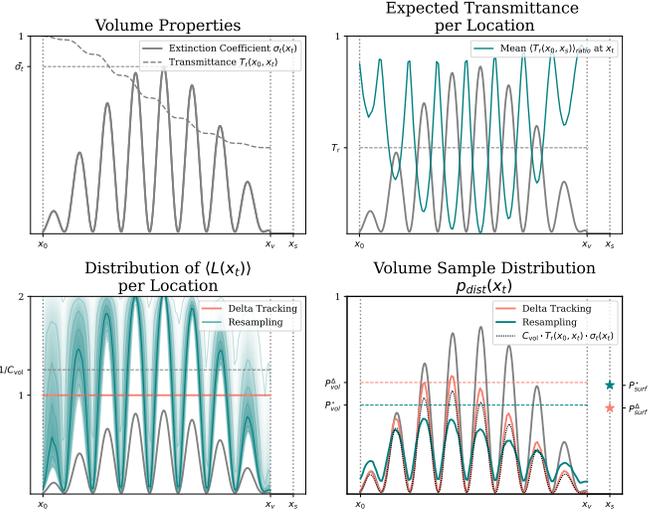


Fig. 7. We demonstrate on a simple 1D volume (**top-left**) the effect of our resampling procedure on the sample distribution within the volume (**bottom-right**) and the throughput of the samples (**bottom-left**). The cause for the change in distribution is the correlation of the transmittance estimate with the re-sampled location (**top-right**). The horizontal line  $1/C_{\text{vol}}$  (**bottom-left**) represents the throughput averaged over all samples across all locations.

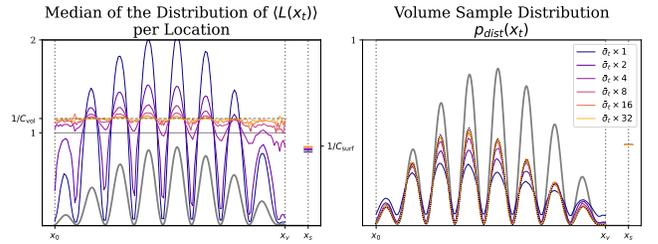


Fig. 8. We analyze the effect of loosening the majorant  $\bar{\sigma}$  to increase the average number of candidates per sample. We show on the **left** that the median of the throughput distribution per location  $\mathbf{x}_t$  becomes more constant and approaches the correct throughput  $1/C_{\text{vol}}$  (Eq. 23) and on the **right** that the resulting volume sample distribution approaches the desired  $C_{\text{vol}} T_r(\mathbf{x}_0, \mathbf{x}_t) \sigma_t(\mathbf{x}_t)$ . This experiment uses the same 1D volume as Fig. 7.

$M_{\text{surf}}$  (Eq. 18), in an auxiliary image space buffer (Fig. 9). The buffer is updated incrementally during rendering after each progression. To remove noise in the buffers, we denoise them during the rendering progressions at  $\{2^0, 2^1, \dots, 2^n\}$  samples per pixel. At the beginning of the path sampling process, the auxiliary buffer is queried to estimate  $P_{\text{vol}}^*$  for the pixel's primary camera ray. Depending on the VSP guiding type, i.e. contribution or variance-based,  $P_{\text{vol}}^*$  is either estimated according to Eq. 15 or Eq. 16. To enable quick and robust learning of the VSP values, we use a constant VSP value of 0.5 for the first rendering progression when the VSP buffer contains no information. Fig. 9 shows an example of the auxiliary image



Fig. 9. Image space auxiliary buffer for estimating the VSP for the primary camera rays for the JUNGLE scene. The **left** shows the estimates for the surface and volume contributions,  $L_s$  and  $L_v$ , as well as the resulting  $P_{\text{vol}}^{\text{1st}}$  (Eq. 15) for each pixel. The **right** shows the estimates for the square roots of the second moments  $M_{\text{surf}}$  and  $M_{\text{vol}}$  of the surface and volume contributions and the resulting  $P_{\text{vol}}^{\text{2nd}}$  (Eq. 16).

buffer for the JUNGLE scene, including the resulting probabilities for contribution-based and variance-based guiding of the VSP.

## 7.2 Secondary Ray VSP Data Structure

For secondary rays, we need to be able to query  $P_{\text{vol}}^*$  at any point  $\mathbf{x}$  and for any direction  $\omega$  in the scene, independent of whether  $\mathbf{x}$  is on a surface or inside a volume. This requires storing our approximation of  $P_{\text{vol}}^*$ , or the required quantities to derive it, in a 5-dimensional data structure.

Therefore, we adopt and extend the 5-dimensional path guiding structure presented by Ruppert et al. [2020] based on parallax-aware von Mises-Fisher mixture models (VMMs). This structure enables us querying the parameter set  $\Theta(\mathbf{x}) = \{\pi_1, \mu_1, \kappa_1, \dots, \pi_K, \mu_K, \kappa_K\}$  of a von Mises-Fisher mixture model  $\mathcal{V}$  approximating the incoming radiance distribution at  $\mathbf{x}$ :

$$\mathcal{V}(\omega | \Theta(\mathbf{x})) = \sum_{k=1}^K \pi_k v(\omega | \mu_k, \kappa_k) \propto L(\omega | \Theta(\mathbf{x})). \quad (29)$$

The mixture represents a normalized spherical distribution by a weighted sum of  $K$  von-Mises Fisher lobes  $v$ , each parameterized by a weight  $\pi_k$ , a mean vector  $\mu_k$  and a concentration  $\kappa_k$ . Each mixture lobe represents a portion of the overall incoming radiance distribution. Our method extends this representation by storing auxiliary data  $c_k^*$  (one floating point number) for each component. This auxiliary data defines each lobe's contribution to the optimal volume scattering probability. The final  $P_{\text{vol}}^*$  for an arbitrary position  $\mathbf{x}$  and direction  $\omega$  is evaluated as follows:

$$P_{\text{vol}}^*(\mathbf{x}, \omega) = \sum_{k=1}^K c_k^* a_k(\omega | \Theta(\mathbf{x})), \quad (30)$$

where the soft assignment function  $a_k$  determines the contribution the  $k$ th lobe of the mixture has to the incoming radiance approximation from direction  $\omega$ :

$$a_k(\omega | \Theta(\mathbf{x})) = \frac{\pi_k v(\omega | \mu_k, \kappa_k)}{\mathcal{V}(\omega | \Theta(\mathbf{x}))}. \quad (31)$$

The auxiliary data  $c_k^*$  is calculated and updated together with the directional guiding structure, i.e. after the mixture parameters for the incoming radiance distribution are updated (see [Ruppert et al. 2020] for details). In this process, a set of training samples  $\{s_1, \dots, s_N\}$  is provided to update the mixture. Each training sample  $s_i$  contains a

direction  $\omega_i$  and weight  $w_i = \langle L(\mathbf{x}, \omega_i) \rangle$  that estimates the incoming radiance arriving at  $\mathbf{x}$  from  $\omega_i$  (Eq. 5). Before calculating  $c_k^*$  we first aggregate the weights statistics  $W_{\text{surf},k}^*$  and  $W_{\text{vol},k}^*$  for the surface and volume contributions for each component  $k$ :

$$W_{\text{surf},k}^* = \frac{\sum_{n=1}^N b(s_i) w_n^* a_k(\omega_n | \Theta(\mathbf{x}))}{\sum_{n=1}^N b(s_i) a_k(\omega_n | \Theta(\mathbf{x}))}, \quad (32)$$

$$W_{\text{vol},k}^* = \frac{\sum_{n=1}^N (1 - b(s_i)) w_n^* a_k(\omega_n | \Theta(\mathbf{x}))}{\sum_{n=1}^N (1 - b(s_i)) a_k(\omega_n | \Theta(\mathbf{x}))}, \quad (33)$$

with  $b(s_i)$  being a binary function that either returns 1 or 0 depending on whether the source of  $w_i$  is a volume or surface interaction:

$$b(s_i) = \begin{cases} 1 & s_i \text{ is a volume contribution,} \\ 0 & \text{otherwise.} \end{cases} \quad (34)$$

The denominators in Eq. 32 and Eq. 33 determine the partial numbers  $N_{\text{surf},k}$  and  $N_{\text{vol},k}$ , of surface and volume samples associated to the  $k$ th lobe. Depending on the desired volume scattering probability type (i.e., contribution or variance-based),  $W_{\text{surf},k}^*$  and  $W_{\text{vol},k}^*$  can either be the first or second-moment contribution aggregates, meaning  $w_n^*$  is either  $w_i$  or  $w_i^2$  respectively. At the end of the training auxiliary data  $c_k^{\text{1st}}$  or  $c_k^{\text{2nd}}$  is calculated as follows:

$$c_k^{\text{1st}} = \frac{P_{\text{vol},k} W_{\text{vol},k}^{\text{1st}}}{(1 - P_{\text{vol},k}) W_{\text{surf},k}^{\text{1st}} + P_{\text{vol},k} W_{\text{vol},k}^{\text{1st}}}, \quad (35)$$

$$c_k^{\text{2nd}} = \frac{\sqrt{(P_{\text{vol},k})^2 W_{\text{vol},k}^{\text{2nd}}}}{\sqrt{(1 - P_{\text{vol},k})^2 W_{\text{surf},k}^{\text{2nd}} + (P_{\text{vol},k})^2 W_{\text{vol},k}^{\text{2nd}}}}.$$

Since the weights  $w_i$  are results of the estimator  $\langle L \rangle$  (Eq. 5) we need to correct the aggregated values of  $W_{\text{vol},k}^*$  and  $W_{\text{surf},k}^*$  to represent statistics of the two nested estimators  $\langle L_v \rangle$  (Eq. 6) and  $\langle L_s \rangle$  (Eq. 7). We do so by multiplying  $W_{\text{vol},k}^*$  and  $W_{\text{surf},k}^*$  with the actual volume and surface scattering probability  $P_{\text{vol},k}$  and  $1 - P_{\text{vol},k}$  for each lobe, which can be estimated by the number of actual generated surface and volume samples assigned to the  $k$ -th lobe:

$$P_{\text{vol},k} \approx \frac{N_{\text{vol},k}}{N_{\text{surf},k} + N_{\text{vol},k}}. \quad (36)$$

## 8 RESULTS AND IMPLEMENTATION

This section presents details about our implementation of our proposed VSPG framework (Sec. 8.1), a detailed description of the experimental setup (Sec. 8.2), and in-depth discussions about the evaluations of multiple experiments (Secs. 8.3–8.7).

### 8.1 Implementation

We integrated our VSPG framework, including our VSP-driven distance sampling algorithm, into the CPU backend of PBRT-v4 [Pharr et al. 2023] together with an integration of directional path guiding using Intel’s Open Path Guiding Library (OpenPGL) [Herholz and Dittebrandt 2022]. This library supports directional guiding on surfaces and inside volumes based on the parallax-aware von Mises-Fisher mixture model representation presented by Ruppert et al. [Ruppert et al. 2020]. We extended OpenPGL as described in Sec. 7.2 to support querying  $P_{\text{vol}}^*$  for any point  $\mathbf{x}$  and direction  $\omega$  in the scene. For denoising our image space VSP buffer (Sec. 7.1), we use Intel’s Open Image Denoise (OIDN) [Áfra 2024]. The following experiments are run on a dual-socket machine with two Intel Xeon Platinum 8260 CPUs, resulting in 48 cores, a total of 96 threads, and 2 terabytes of RAM. To avoid any interference from the operating system, we run our experiments using only 90 threads.

### 8.2 Experimental Setup

Since our work consists of two components, the VSPG framework (Sec. 7) and the resampling-based VSP distance sampling algorithm (Sec. 5.4), we designed an experimental setup to evaluate both.

To evaluate the performance of our VSPG framework, we compare it against two volumetric path tracer baselines: one that does not use any path guiding and one that uses directional path guiding. Both baselines use delta tracking for distance sampling (Sec. 3.3.1), leading to a transmittance-based VSP (Eq. 10). The unguided baseline samples directions using BSDF and phase function-based importance sampling. The directional guiding baseline importance samples the directions using the product of the learned incident radiance and the cosine term on surfaces and using the product of the learned incident radiance and the phase function in volumes. In both cases, we use defensive sampling [Hesterberg 1995] via multiple importance sampling (MIS) [Veach and Guibas 1995] to combine it with BSDF resp. phase-function-based importance sampling with a selection probability of 0.5. In addition, we use next-event estimation for light sampling in combination with MIS in all compared methods.

To evaluate the performance of our resampling-based VSP distance sampling algorithm, we compare it against the work by Villemin et al. [2018], which is, to our current knowledge, the only method explicitly adjusting the VSP. Their method uses an extension of weighted delta tracking that also adjusts the distance sampling scheme for event candidates, which they call *normalized distance sampling* (NDS). In addition to NDS, they also use a *null-collision probability bias* (NCPB) scheme that applies a gamma function to the null-collision probability (Eq. 13). Finding the correct gamma value for a given VSP value requires prior knowledge about the transmittance through the volume. It is mainly used for primary rays using a pre-computed image-space transmittance buffer. To

best represent Villemin et al.’s work, we evaluate both versions of NDS, one without (NDS) and one with NCPB (NDS+). For the latter, we use a pre-computed transmittance buffer and exclude the preprocessing time from the reported rendering times. Further, we use an improved version of their NCPB scheme ([Villemin et al. 2018], Sec 3.3) since the one proposed in their paper did not work as intended in our experiments (see Appendix B). We combine both variants with our VSPG framework for primary and secondary rays since the original approach requires manual adjustment of the VSP for each scene globally or per volume. We denote the full methods as VSPG (NDS) and VSPG (NDS+) throughout the text to emphasize the use of our framework.

For both NDS-based VSPG variants, we use the variance-based optimal VSP (Sec. 5.1.2) as this showed slight benefits over the contribution-based alternative. For VSPG that uses our resampled-based VSP distance sampling, denoted by VSPG (Resampling), we show results with both optimal VSPs and discuss the differences in Sec. 8.6. Since guiding the VSP for secondary rays always requires a path guiding data structure, we additionally enable directional path guiding for all methods that use our VSPG framework.

We use the mean *relative* squared error (relMSE) [Rousselle et al. 2011] with a 0.1-percentile outlier removal as our error metric. For better readability, we scale all relMSE values by a factor of 10x. Further, for all renderings, we used a maximum path depth of 15 and decided to disable stochastic Russian Roulette to avoid any variance increase caused by early path termination.

### 8.3 Scenes & Results

We evaluate our method on various scenes with different volume types and illumination conditions. Fig. 10 shows equal-time comparisons between our proposed VSP guiding variants and the existing delta tracking and directional guiding baselines, and Tab. 1 presents some additional statistics such as average path lengths (avg. pl) or the average number of density queries per path segment (avg. dq).

*JUNGLE, KITCHEN and LANDSCAPE.* These scenes share a similar setup with optically thin heterogeneous media illuminated by environmental lighting. The lighting in the scenes is dominated by low-order volumetric scattering, which is responsible for volumetric shadows and light shafts. Compared to the baselines, all VSPG methods are able to increase the volume scatter probability and provide quality improvements under equal time.

*UNDERWATER.* This underwater scene is filled with a thin but large homogeneous medium representing water. The water surface stems from an accurate water simulation and is modeled as a rough dielectric material with small surface roughness ( $\alpha = 1e - 4$ ). Paths have to scatter through the dielectric surface to reach the environment map. This scene strongly benefits from directional path guiding to direct paths toward the sun. Our VSPG framework instructs an overall decrease of the VSP such that more paths reach the water surface and refract towards the sun instead of scattering away inside the water. VSPG (Resampling) thus decreases the VSP on the way to the water surface, leading to shorter average path lengths and thus more samples per pixel under equal time (see Tab. 1). VSPG (NDS) and VSPG (NDS+), on the other hand, cannot decrease the

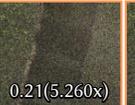
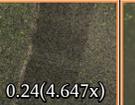
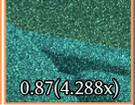
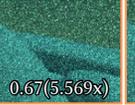
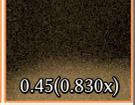
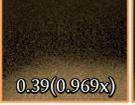
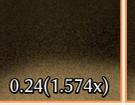
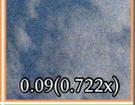
	No Guiding	Dir. Guiding	Dir. + VSPG (NDS), Variance	Dir. + VSPG (NDS+), Variance	Dir. + VSPG (Resampling), Contribution	Dir. + VSPG (Resampling), Variance	Reference
Jungle							
	1.10	1.10(1.000x)	0.71(1.557x)	0.32(3.488x)	0.21(5.260x)	0.24(4.647x)	relMSE
	40SPP	37SPP	36SPP	37SPP	39SPP	38SPP	300s
Kitchen							
	0.56	0.29(1.960x)	0.23(2.418x)	0.20(2.746x)	0.23(2.498x)	0.24(2.329x)	relMSE
	206SPP	158SPP	136SPP	137SPP	128SPP	125SPP	300s
Landscape							
	0.95	0.79(1.201x)	0.32(2.981x)	0.27(3.470x)	0.27(3.454x)	0.27(3.461x)	relMSE
	56SPP	67SPP	65SPP	65SPP	63SPP	62SPP	300s
Underwater							
	3.73	0.87(4.288x)	0.99(3.777x)	0.98(3.824x)	0.67(5.569x)	0.73(5.076x)	relMSE
	124SPP	127SPP	111SPP	113SPP	133SPP	129SPP	300s
Lantern							
	0.38	0.45(0.830x)	0.39(0.969x)	0.38(0.984x)	0.24(1.574x)	0.24(1.537x)	relMSE
	98SPP	50SPP	44SPP	44SPP	47SPP	46SPP	120s
Earth							
	0.07	0.09(0.722x)	0.08(0.857x)	0.06(1.130x)	0.10(0.654x)	0.11(0.587x)	relMSE
	166SPP	115SPP	130SPP	128SPP	65SPP	58SPP	120s

Fig. 10. Comparison of unguided path tracing and directional path guiding to various variants of our VSP guiding framework, which are always combined with directional path guiding. We show our VSPG framework described in Sec. 7 with different distance sampling algorithms: VSPG (NDS) and VSPG (NDS+) use algorithms based on Villemin et al. [2018] and VSPG (Resampling) uses the algorithm described in Sec. 5.4. For VSPG (Resampling), we show results with both optimal VSP formulations separately (Sec. 5.1). Our error metric is relMSE [Rousselle et al. 2011] with a 0.1-percentile outlier removal, and the displayed error value is multiplied by a factor of 10 for readability. The first four scenes are evaluated with 5-minute renderings and the latter two with 2-minute renderings. Sec. 8.2 describes the used experimental setup in more detail and Sec. 8.3 discusses these results.

VSP (Sec. 8.5) and end up with a higher error than the directional path guiding baseline due to their overhead, which results in lower SPP numbers.

**LANTERN.** This scene is mainly illuminated by light sources inside the paper lanterns placed behind the optically dense heterogeneous smoke. An additional light source on the top dimly illuminates the scene. For rays that pass through the volume and have their first surface intersection on the bright lantern (orange inset), surface contribution dominates, and our VSPG framework instructs more surface scattering events to happen. Again, VSPG (NDS) and VSPG (NDS+) cannot decrease the volume sampling probability, and we only observe improvements from VSPG (Resampling). For rays whose first surface intersection is the dark background (green inset), all VSPG methods improve over the baselines as they can all increase the VSP. This scene exemplifies the occasional need to both increase and decrease of the VSP even within the same volume, demonstrating the strength of automatic VSP framework.

**EARTH.** This scene is illuminated by an environment map containing a sun on the right side. We model the atmosphere above the earth’s surface with an exponentially decaying optically thin medium and the cloudscapes with an optically dense high-frequency heterogeneous volume with an albedo of 1.0. Here, all VSPG methods, except for VSPG (NDS), increase the volume scatter probability and are significantly better in the rim (green inset), where rays traverse through the optically thin atmosphere. In this scene, we observe an overall increase in the error values for both our VSPG (Resampling) approaches. This increase is related to the lower numbers of SPPs that can be evaluated at equal-time rendering due to a significant increase in average path lengths when using our algorithm (see Tab. 1). This is because most of the background behind the volume is dark, or even black, and therefore not contributing to the estimate of  $L$ . Consequently, VSPG will almost always prefer scattering inside the volume, even if its contribution to the subsequent path to the final pixel estimate is low, therefore trapping the path inside the volume. In contrast to VSPG (NDS) and VSPG (NDS+), which often can not reach the target VSP (see Sec. 8.5), VSPG (Resampling) usually reaches its VSP target, which, in this case, results in a much longer average path length (e.g., VSPG (NDS): 2.16 and VSPG (Resampling): 5.41). A way to compensate for this behavior is to use Russian roulette (RR) [Arvo 1986], starting at an early path depth (e.g., 1) as demonstrated in Fig. 11. Enabling RR leads to more similar average path lengths across the different VSPG methods (e.g., VSPG (NDS): 1.50 and VSPG (Resampling): 1.86) and, therefore, to similar SPPs and error values. We explicitly disabled RR in our evaluation since we observed that, in scenes with dense low-albedo volumes (e.g., LANTERN), it could become the primary source of variance, negating any positive effect of directional or VSP guiding. Implementing guided RR, as presented by Vorba et al. [2016] and extended to volumes by Herholz et al. [2019], should counter this source of inefficiency by terminating paths that have a low contribution to the estimate of  $L$  for the final pixel value.

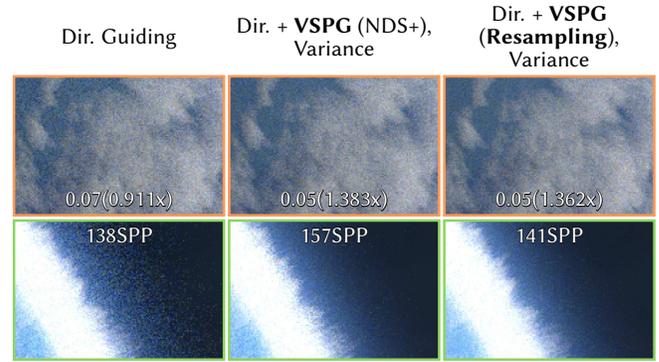


Fig. 11. Equal-time (2min) comparison with Russian roulette enabled: **Left** directional path guiding with delta tracking (avg. pl 1.42), **center** VSPG (NDS+) (avg. pl 1.50), and **right** VSPG (Resampling) (avg. pl 1.86).

#### 8.4 Primary and Secondary VSP Guiding

Our implementation (Sec. 7) of the VSPG framework allows guiding the VSP for the primary rays and secondary rays. Depending on the scene, the importance of both can vary significantly. Especially in scenes where the volume is directly visible, and the volume contribution is mainly single scattering (e.g., JUNGLE, KITCHEN, or LANDSCAPE), guiding the VSP of the camera rays has the biggest influence on the overall variance reduction and guiding secondary VSP decisions only leads to a minimal additional variance reduction. An example of the dominant effect of primary ray VSP guiding is shown for KITCHEN in Fig. 12 (**top**). Secondary ray VSP guiding, on the other hand, is important in situations where the volume is either not directly visible (e.g., bounded by a translucent surface) or when the dominant contribution comes from secondary scattering. An example where primary ray VSP guiding has little effect and secondary ray VSP guiding is dominant is shown for UNDERWATER in Fig. 12 (**bottom**).

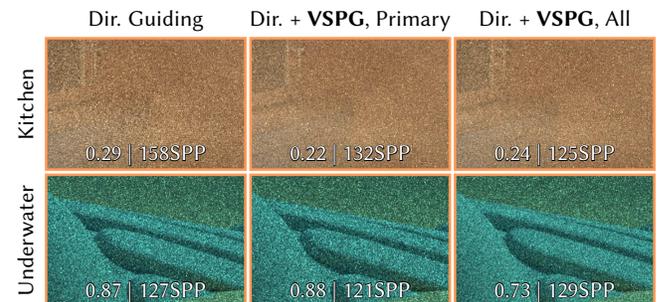


Fig. 12. Equal-time (5min) renderings showing primary-only and all ray VSP guiding in different scenes. **Left** does not use any VSP guiding, while **Center** guides the VSPs for primary ray, and **Right** for primary and secondary rays.

#### 8.5 Resampling versus NDS

The NDS-based variants of VSPG can only increase the VSP as opposed to VSPG (Resampling) that allows to both increase or decrease the VSP. This puts them at a fundamental disadvantage over the

Scene (time)	No Guiding				Directional Guiding (Dir.)				Dir. + VSPG (NDS)				Dir. + VSPG (NDS+)				Dir. + VSPG (Resampling)			
	spp	error	avg. pl	avg. dq	spp	error	avg. pl	avg. dq	spp	error	avg. pl	avg. dq	spp	error	avg. pl	avg. dq	spp	error	avg. pl	avg. dq
JUNGLE (5min)	40	1.10	6.41	0.27	37	1.10	6.42	0.24	36   36	0.65   0.71	6.32   6.36	0.36   0.32	37   37	0.28   0.32	6.11   6.20	0.36   0.32	39   38	<b>0.21</b>   0.24	5.76   5.94	0.55   0.45
KITCHEN (5min)	206	0.56	7.68	0.09	158	0.29	6.74	0.09	135   136	0.22   0.23	6.83   6.83	0.12   0.12	138   137	<b>0.20</b>   0.20	6.82   6.83	0.12   0.11	128   125	0.23   0.24	6.75   6.80	0.17   0.16
LANDSCAPE (5min)	56	0.95	4.99	0.05	67	0.79	3.91	0.06	66   65	0.32   0.32	3.81   3.85	0.18   0.18	65   65	0.28   <b>0.27</b>	3.82   3.87	0.17   0.17	63   62	0.27   0.27	4.02   4.12	0.55   0.51
UNDERWATER (5min)	124	3.73	9.80	0.72	127	0.87	6.92	0.67	114   111	0.95   0.99	7.10   7.33	0.68   0.69	103   113	1.02   0.98	7.10   7.33	0.68   0.69	133   129	<b>0.67</b>   0.73	5.98   6.20	0.60   0.61
LANternS (2min)	98	0.38	4.73	1.52	50	0.45	6.17	1.19	43   44	0.39   0.39	6.23   6.22	1.23   1.22	44   44	0.38   0.38	6.26   6.24	1.22   1.21	47   46	<b>0.24</b>   0.24	5.03   5.13	3.02   2.94
EARTH (2min)	166	0.07	2.18	4.39	115	0.09	2.07	4.77	136   130	0.07   0.08	2.12   2.12	4.78   4.78	131   128	<b>0.06</b>   0.06	2.16   2.16	4.64   4.64	65   58	0.10   0.11	4.73   5.41	8.45   8.60

Table 1. Rendering statistics for the different algorithms for the scenes shown in Fig. 10. The stats show the number of samples achieved in equal time, the resulting relMSE, the average path lengths, and the number of average volume density queries per path segment. For VSPG, we show the numbers for contribution and variance-based VSP guiding, each separated with a "|". For each scene, the method with the lowest relMSE is highlighted in bold.

resampling method when decreasing the VSP is the optimal choice (e.g., UNDERWATER and LANTERN). In such a situation, they degrade to delta tracking, whereas our resampling method is able to correctly reduce the VSP.

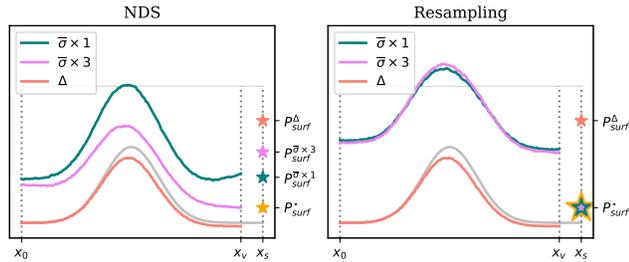


Fig. 13. Sample distribution of a 1D function using NDS on the left and our resampling method on the right. The distribution from delta tracking is shown in orange, with the volume density curve in gray. For both methods, we show the resulting distribution when using 1× and 3× the majorant. The stars on the right indicate the respective achieved surface scattering probabilities.

Even when increasing the VSP is optimal, the NDS-based methods have issues compared to our resampling method. Fig. 13 shows a simple 1D experiment that compares the properties of the resulting distance sampling distribution of both algorithms. NDS does not fully reach the desired  $P_{vol}^*$ , even when using the null-collision bias scheme, while the resampling method precisely reaches it. Further, NDS is negatively affected by loosening the majorant, as the discrepancy between the resulting and desired VSP increases, while our resampling method's reached VSP remains optimal and even benefits from loose majorants (Sec. 6). An example of how scaling the majorant influences VSPG (NDS) and VSPG (Resampling) is shown in Fig. 14. Note that VSPG (NDS+) could somewhat mitigate this issue on primary rays, but cannot be applied on secondary rays.

### 8.6 Contribution vs. Variance-Based VSP Guiding

Even if variance-based VSP guiding is theoretically preferable, the results, in terms of overall image error (i.e., relMSE), are often similar to contribution-based VSP (see Fig. 10 and Tab. 1). We observed that in most of our test scenes, the relative variance of  $\langle L_s \rangle$  and  $\langle L_v \rangle$  are often similar, leading to similar variance-based and contribution-based VSP values. These values significantly differ in cases where one of the estimators has a much higher or lower variance (e.g., narrow light shafts or environmental background). At the same time, the ratio between the surface and volume contributions and

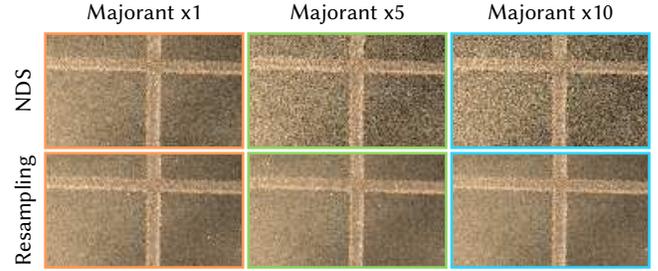


Fig. 14. Comparing the effect of scaling the majorant has in KITCHEN using VSPG with different distance sampling strategies: **Top** NDS and **bottom** our resampling approach.

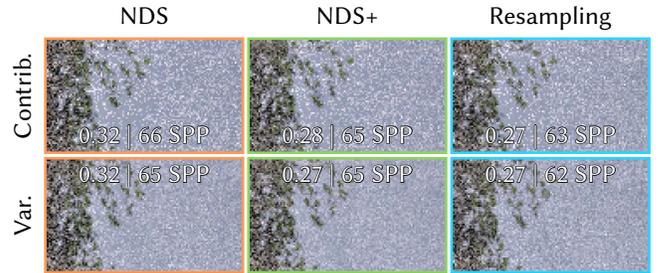


Fig. 15. Equal-time (5min) comparison of VSPG using contribution (**top**) or variance-based (**bottom**) VSP guiding for different distance sampling methods: **Left** NDS, **center** NDS+, and **right** our resampling approach.

variances needs to differ significantly. In our scenes, these cases rarely occur, and if they do, they only affect local regions in the image (e.g., LANDSCAPE: light shafts around the tree), barely affecting the overall averaged image error (Fig. 15).

### 8.7 Computational and Memory Overhead

In the following, we analyze the computational and memory overhead of our VSPG framework over the baselines.

We split the computational overhead into per-path segments and additional training overhead. We measured an average per-path segment overhead of 16.5% for directional guiding and 26.6% for directional and VSP guiding compared to unguided path tracing. The overhead of learning the additional VSP data for the 5D structure is minimal; in average, the training time changed from 6.8% to 7.2% of the total render time. Mainly due to the denoising overhead, training the image-space VSP buffer costs an additional 3.1%.<sup>3</sup>

<sup>3</sup>This overhead can be significantly reduced when using a GPU for denoising.

Scene (time)	Dir. Guiding		Dir. + VSPG (Resampling)	
	# caches	size (MB)	# caches	size (MB)
JUNGLE (5min)	8752	28	8401	32
KITCHEN (5min)	28272	91	27761	106
LANDSCAPE (5min)	12765	40	12677	48
UNDERWATER (5min)	37983	113	31998	122
LANTERN (2min)	10542	34	8140	31
EARTH (2min)	15012	49	16319	62

Table 2. Memory statistics for the 5D guiding structures for our test scenes (Fig. 10) when only using directions guiding (**left**) and when adding VSP guiding (**right**). The statistics also include the number of guiding caches.

The additional memory required to store the VSP guiding information is split between the image space VSP buffer (Sec. 7.1) and the 5D VSP guiding structure (Sec. 7.2). For the image space VSP buffer, we need to store 7 additional floats per pixel; the surface and volume contribution and the VSP value. Further, for the image-space denoiser, we need to store additional auxiliary data per pixel, such as unfiltered surface and volume contribution, albedo, normal, and samples per pixel, which results in 13 additional floats. This results in 55MB for the image-space VSP buffer and 102MB for the denoising data for an image with a resolution of 1920x1080. For the 5D VSP guiding structure, we measured a 15% size increase per guiding cache compared to just storing the directional guiding data. In Tab. 2, we show the memory consumption of the 5D guiding structure with and without VSP guiding as well as the number of guiding caches generated by the different methods.

## 9 DISCUSSION AND FUTURE WORK

*Density Query Costs.* Since our resampling-based distance sampling algorithm (Sec. 5.4) requires stepping through the entire volume, or at least to the first surface intersection inside the volume, the number of density queries increases compared to standard delta tracking or NDS (see Tab. 1). While we did not observe any negative impact on efficiency in our implementation and on our tested scenes, we believe that this could change in other, more production-related scenarios where individual density queries are more expensive (i.e., procedural volumes) or in scenes with extremely large and dense volumes (e.g., cloudscape). In this case, integrating some Russian roulette-based termination approach, as proposed by Wrenninge and Villemin [2020], might be required to reduce the overhead, especially in cases with low target VSPs.

*Product Distance Guiding.* Our method currently focuses on only guiding the VSP while keeping a transmittance-based PDF similar to traditional distance strategies. Ideally, one would also consider the distributions of the in-scattered or emitted volume radiance and their product with transmittance. Wrenninge and Villemin’s method [2020], which forms the basis of our resampling method, also utilizes equiangular sampling [Kulla and Fajardo 2012] to sample according to the full product for single scattering contributions. Therefore, it should be straightforward to combine VSP and distance guiding to leverage both advantages. Fig. 16 shows the results of a naive integration of the product distance sampling into our resampling framework, where at each candidate position, an approximation of the in-scattered radiance is queried from the guiding structure and integrated into the resampling process. While the

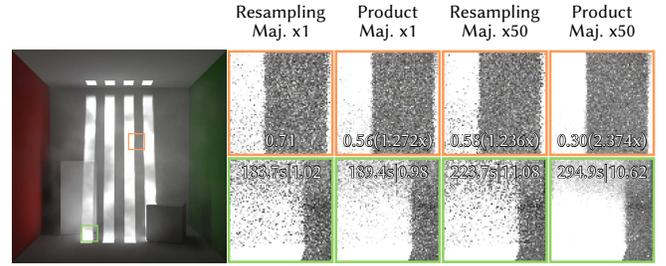


Fig. 16. Equal-sample comparison (64spp) of our resampling algorithm to its extension to support product distance guiding using different majorant scaling factors (**left**: x1 and **right**: x50) to increase the number of candidates.

narrow light shafts in the scene are a perfect use case for product distance sampling, we only observe minor improvements to standard resampling when the average number of event candidates inside the volume is low (**left**). To see a clear improvement, we need to increase the average number of event candidates inside the volume from 1 to 10 by increasing the volume’s majorant by 50, which almost doubles the runtime (**right**). While the variance reduction outweighs the cost increase in this example, this might not be true in other less extreme scenarios, where product distance sampling will have a much lower impact on variance reduction. This is especially true in scenes with a more uniform in-scattered radiance distribution, and the additional density and guiding cache query overhead can quickly decrease efficiency. This is a limitation our naive integration shares with previous works like the ones of Herholz et al. [2019] or Huang et al. [2021]. A potential avenue for future work to overcome this limitation would be to extend the VMM lobes with additional auxiliary data, such as a 1D representation of the in-scattered radiance distribution along the direction of the lobe that is cheap to learn and cheap to evaluate.

*Guiding other Binary Decisions.* This work shows, surprisingly, the effectiveness of guiding binary decisions that are usually only considered implicitly as a byproduct of other sampling techniques, such as transmittance-based distance sampling. The volume scattering probability is not the only such decision that has been neglected so far. The binary decision to reflect or refract on dielectric (potentially multi-layered) interfaces falls into the same category. This decision is usually driven by the local Fresnel term only and completely ignores the contribution and variance coming from the reflecting or refractive direction. We believe our current framework could easily be extended to this and other use cases.

*High-Quality Transmittance Estimates.* Our resampling method allows for using any unbiased transmittance estimate for the surface candidate. This allows us, for instance, to replace the surface candidate’s ratio tracking estimator used in Sec. 5.3 with an unbiased, potentially more expensive, high-quality transmittance estimator (e.g., [Kettunen et al. 2021]). Early experiments show that this can significantly reduce variance for the primary rays when the surface contribution dominates and the variance of the ratio tracking estimator is high (Fig. 17). However, a careful analysis is required to trade off the increased computational cost with the gain in quality

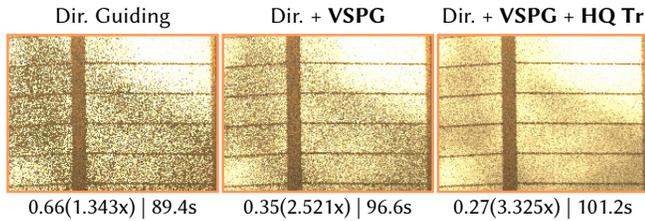


Fig. 17. Equal-sample renderings (32spp) of the LANTERN scene. In this inset, the main contribution comes from the surface behind the volume and the ratio tracking estimator has high variance. **Left:** Directional path guiding (using delta tracking). **Center:** VSPG using the ratio tracking estimator for the surface candidate. **Right:** VSPG using the transmittance estimator from Kettunen et al. [2021] for the surface candidate.

to avoid negative impacts on the rendering efficiency. We leave this as future work.

## 10 CONCLUSION

We presented a practical and widely applicable unbiased method for guiding the volume sampling probability, which significantly improves the rendering efficiency of volumetric effects. Through our extensive analysis and evaluation, we demonstrated that the volume sampling probability is an essential contributor to variance when rendering scenes with volumetric effects, and controlling this probability with a systematic method can yield significant efficiency improvements. Our method is lightweight, robust, and fully automatic without any hyper-parameters, as demonstrated in various volumetric scenarios across realistic scenes commonly used in production. Moreover, our method is easy to integrate into existing volumetric path tracing renderers, especially those that have already adopted the data structures needed for directional path guiding.

## ACKNOWLEDGMENTS

We thank Christoph Peters, Tiziano Portenier, Cliff Ramshaw, Sergio Sancho, and Alex Rath for proofreading the paper and providing helpful feedback, as well as Julian Fong, Ethan Jaszewski, and Cliff Ramshaw for discussions on delta tracking and its connection to resampling. Special thanks go to George Kyriazis for keeping our render servers up and running, to Cristiano Siqueira for the JUNGLE scene and the sleepless weekends to port it to PBRT, and to Violaine Fayolle and Dorian Van Essen for creating the LANTERN scene for this project. We also want to thank the following persons and institutions for providing access to scenes or data sets used in this work: Jay-Artist (KITCHEN), Jan-Walter Schliep, Burak Kahraman, and Timm Dapper from Laubwerk (LANDSCAPE), Pablo Vazquez (UNDERWATER) and Tom Patterson (EARTH). We further thank Disney for the cloud asset in Fig. 3 which was originally created by Kevin Udy and distributed under the CC BY-SA 3.0 license.

## REFERENCES

Attila T. Áfra. 2024. Intel® Open Image Denoise. <https://www.openimagedenoise.org>.  
 James Arvo. 1986. Backward ray tracing. In *ACM SIGGRAPH Course Notes*. 259–63.  
 Thomas Bashford-Rogers, Kurt Debattista, and Alan Chalmers. 2012. A significance cache for accelerating global illumination. *Comput. Graph. Forum* 31, 6 (2012), 1837–1851.

Subrahmanyan Chandrasekhar. 1960. *Radiative transfer*. Courier Corporation.  
 Min-Te Chao. 1982. A general purpose unequal probability sampling plan. *Biometrika* 69, 3 (1982), 653–656.  
 W. A. Coleman. 1968. Mathematical Verification of a Certain Monte Carlo Sampling Technique and Applications of the Technique to Radiation Transport Problems. *Nuclear Science and Engineering* 32, 1 (1968), 76–81. <https://doi.org/10.13182/NSE68-1>  
 John Danskin and Pat Hanrahan. 1992. Fast algorithms for volume ray tracing. In *Proceedings of the 1992 Workshop on Volume Visualization* (Boston, Massachusetts, USA) (VVS '92). Association for Computing Machinery, New York, NY, USA, 91–98. <https://doi.org/10.1145/147130.147155>  
 Mathieu Galtier, Stephane Blanco, Cyril Caliot, Christophe Coustet, Jérémie Dauchet, Mouna El Hafi, Vincent Eymet, Richard Fournier, Jacques Gautrais, Anais Khuong, et al. 2013. Integral formulation of null-collision Monte Carlo algorithms. *Journal of Quantitative Spectroscopy and Radiative Transfer* 125 (2013), 57–68.  
 Iliyan Georgiev, Jaroslav Krivánek, Toshiya Hachisuka, Derek Nowrouzezahrai, and Wojciech Jarosz. 2013. Joint Importance Sampling of Low-Order Volumetric Scattering. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)* 32, 6 (Nov. 2013). <https://doi.org/10/gbd5qs>  
 Iliyan Georgiev, Zackary Misso, Toshiya Hachisuka, Derek Nowrouzezahrai, Jaroslav Krivánek, and Wojciech Jarosz. 2019. Integral formulations of volumetric transmittance. *ACM Transactions on Graphics (TOG)* 38, 6 (2019), 1–17.  
 Johannes Hanika, Andrea Weidlich, and Marc Droske. 2022. Once-more scattered next event estimation for volume rendering. *Computer Graphics Forum (Proceedings of Eurographics Symposium on Rendering)* 41, 4 (2022). <https://doi.org/10.1111/cgf.14583>  
 Sebastian Herholz and Addis Dittebrandt. 2022. Intel® Open Path Guiding Library. <http://www.openppl.org>.  
 Sebastian Herholz, Yangyang Zhao, Oskar Elek, Derek Nowrouzezahrai, Hendrik P. A. Lensch, and Jaroslav Krivánek. 2019. Volume Path Guiding Based on Zero-Variance Random Walk Theory. *ACM Trans. Graph.* 38, 3, Article 25 (jun 2019), 19 pages. <https://doi.org/10.1145/3230635>  
 Tim Hesterberg. 1995. Weighted Average Importance Sampling and Defensive Mixture Distributions. *Technometrics* 37, 2 (1995), 185–194. <http://www.jstor.org/stable/1269620>  
 J. Eduard Hoogenboom. 2008. Zero-variance Monte Carlo schemes revisited. *Nuclear Science and Engineering* 160, 1 (sep 2008), 1–22.  
 Wei-Feng Huang, Peter Kutz, Yining Karl Li, and Matt Jen-Yuan Chiang. 2021. Unbiased Emission and Scattering Importance Sampling For Heterogeneous Volumes. In *ACM SIGGRAPH 2021 Talks*. 1–2.  
 James T. Kajiya. 1986. The rendering equation. *SIGGRAPH Comput. Graph.* 20 (1986).  
 Markus Kettunen, Eugene D'Eon, Jacopo Pantaleoni, and Jan Novák. 2021. An unbiased ray-marching transmittance estimator. *ACM Trans. Graph.* 40, 4, Article 137 (jul 2021), 20 pages. <https://doi.org/10.1145/3450626.3459937>  
 David Körner, Jan Novák, Peter Kutz, Ralf Habel, and Wojciech Jarosz. 2016. Subdivision Next-Event Estimation for Path-Traced Subsurface Scattering. In *Proceedings of EGSR (Experimental Ideas and Implementations)*. The Eurographics Association. <https://doi.org/10/gf6rzj>  
 Christopher Kulla and Marcos Fajardo. 2012. Importance sampling techniques for path tracing in participating media. In *Computer graphics forum*, Vol. 31. Wiley Online Library, 1519–1528.  
 Peter Kutz, Ralf Habel, Yining Karl Li, and Jan Novák. 2017. Spectral and decomposition tracking for rendering heterogeneous volumes. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 1–16.  
 Jaakko Leppänen. 2010. Performance of Woodcock delta-tracking in lattice physics applications using the Serpent Monte Carlo reactor physics burnup calculation code. *Annals of Nuclear Energy* 37, 5 (2010), 715–722.  
 Bailey Miller, Iliyan Georgiev, and Wojciech Jarosz. 2019. A null-scattering path integral formulation of light transport. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 38, 4 (July 2019). <https://doi.org/10/gf6rzb>  
 Thomas Müller, Markus Gross, and Jan Novák. 2017. Practical Path Guiding for Efficient Light-Transport Simulation. *Comput. Graph. Forum* 36, 4 (jul 2017), 91–100. <https://doi.org/10.1111/cgf.13227>  
 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).  
 Jan Novák, Andrew Selle, and Wojciech Jarosz. 2014. Residual Ratio Tracking for Estimating Attenuation in Participating Media. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)* 33, 6 (Nov. 2014). <https://doi.org/10/f6r2nq>  
 Vincent Pegoraro, Ingo Wald, and Steven G. Parker. 2008. Sequential monte carlo integration in low-anisotropy participating media. In *Proceedings of the Nineteenth Eurographics Conference on Rendering (Sarajevo, Bosnia and Herzegovina) (EGSR '08)*. Eurographics Association, Goslar, DEU, 1097–1104. <https://doi.org/10.1111/j.1467-8659.2008.01247.x>  
 Ken Perlin and Eric M Hoffert. 1989. Hypertexture. In *Proceedings of the 16th annual conference on Computer graphics and interactive techniques*. 253–262.

- Matt Pharr, Wenzel Jakob, and Greg Humphreys. 2023. *Physically based rendering: From theory to implementation* (4rd ed.). Morgan Kaufmann.
- Alexander Rath, Pascal Grittmann, Sebastian Herholz, Petr Vévoda, Philipp Slusallek, and Jaroslav Krivánek. 2020. Variance-aware path guiding. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 151–1.
- Fabrice Rousselle, Claude Knaus, and Matthias Zwicker. 2011. Adaptive sampling and reconstruction using greedy error minimization. *ACM Transactions on Graphics (TOG)* 30, 6 (2011), 1–12.
- Lukas Ruppert, Sebastian Herholz, and Hendrik P. A. Lensch. 2020. Robust Fitting of Parallax-Aware Mixtures for Path Guiding. *ACM Trans. Graph.* 39, 4, Article 147 (aug 2020), 15 pages. <https://doi.org/10.1145/3386569.3392421>
- László Szirmay-Kalos, Iliyan Georgiev, Milán Magdics, Balázs Molnár, and Dávid Légrády. 2017. Unbiased Light Transport Estimators for Inhomogeneous Participating Media. *Computer Graphics Forum* 36, 2 (2017). EUROGRAPHICS 2017.
- László Szirmay-Kalos, Milán Magdics, and Mateu Sbert. 2018. Multiple Scattering in Inhomogeneous Participating Media Using Rao-Blackwellization and Control Variates. *Computer Graphics Forum* 37, 2 (2018), 63–74. <https://doi.org/10.1111/cgf.13342>
- Eric Veach and Leonidas J. Guibas. 1995. Optimally Combining Sampling Techniques for Monte Carlo Rendering. Vol. 29. 419–428. <https://doi.org/10/d7b6n4>
- Ryusuke Villemin, Magnus Wrenninge, and Julian Fong. 2018. Efficient Unbiased Rendering of Thin Participating Media. *Journal of Computer Graphics Techniques (JCGT)* 7, 3 (13 September 2018), 50–65. <http://jcggt.org/published/0007/03/03/>
- Jiří Vorba and Jaroslav Krivánek. 2016. Adjoint-driven Russian roulette and splitting in light transport simulation. *ACM Trans. Graph.* 35, 4 (2016), 42:1–42:11.
- E Woodcock, T Murphy, P Hemmings, and S Longworth. 1965. 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*, Vol. 557. Argonne National Laboratory.
- Magnus Wrenninge and Ryusuke Villemin. 2020. *Product Importance Sampling of the Volume Rendering Equation using Virtual Density Segments*. <https://graphics.pixar.com/library/CandidateSampling/paper.pdf> Technical Note 20-01, Pixar Animation Studios.

## A RENDERING CHROMATIC MEDIA

Our resampling algorithm described in Sec. 5.4 can be extended to fit into the null scattering path integral framework [Miller et al. 2019]. This allows to perform MIS between color channels for efficient rendering of chromatic media with spectrally varying coefficients  $\sigma_t^{\vec{c}}, \sigma_\eta^{\vec{c}}, \sigma^{\vec{c}}$ .

For the resampling algorithm, we randomly select one color component  $c$  among the channels in  $\vec{c}$  and adopt this channel’s volume properties to generate event candidates and compute resampling weights. For the sample  $\mathbf{x}_i$  drawn from this procedure, the multi-channel throughput  $\langle T_p^{\vec{c}}(\mathbf{x}_i) \rangle$  is the scalar throughput  $\langle T_p^c(\mathbf{x}_i) \rangle$  (Eq. 28) from the resampling algorithm multiplied by a channel-wise correction factor  $\frac{f^{\vec{c}}(\mathbf{x}_i)}{p^{\vec{c}}(\mathbf{x}_i)}$ .

This correction factor is computed in the same way as the throughput in the null scattering path integral framework, based on the selected real scattering vertex  $\mathbf{x}_i$  and all the preceding null scattering vertices  $\mathbf{x}_1, \dots, \mathbf{x}_{i-1}$ . The numerator  $f^{\vec{c}}(\mathbf{x}_i)$  corresponds to sampling the integrand of the null scattering extension of the VRE [Galtier et al. 2013] according to  $\mathbf{x}_1, \dots, \mathbf{x}_i$ :

$$f^{\vec{c}}(\mathbf{x}_i) = \overline{T}_r^{\vec{c}}(\mathbf{x}_0, \mathbf{x}_i) \sigma_t^{\vec{c}}(\mathbf{x}_i) \left( \prod_{j=1}^{i-1} \sigma_\eta^{\vec{c}}(\mathbf{x}_j) \right), \quad (37)$$

with  $\overline{T}_r^{\vec{c}}(\mathbf{x}_0, \mathbf{x}_i) = \exp(-\overline{\sigma^{\vec{c}}} \|\mathbf{x}_0 - \mathbf{x}_i\|)$  being the majorant transmittance. The probability of sampling this path with color channel  $c$  is:

$$p^c(\mathbf{x}_i) = \overline{T}_r^c(\mathbf{x}_0, \mathbf{x}_i) \sigma_t^c(\mathbf{x}_i) \left( \prod_{j=1}^{i-1} \sigma_\eta^c(\mathbf{x}_j) \right), \quad (38)$$

and being able to analytically compute  $p^c(\mathbf{x}_i)$  for every color channel  $c \in \vec{c} = \{c_1, \dots, c_N\}$  allows MIS between all channels:

$$p^{\vec{c}}(\mathbf{x}_i) = \frac{1}{N} \sum_{j=1}^N p^{c_j}(\mathbf{x}_i). \quad (39)$$

The EARTH scene in Fig. 10 shows the application of our resampling method on rendering a chromatic medium.

## B MODIFIED NULL-COLLISION PROBABILITY BIAS

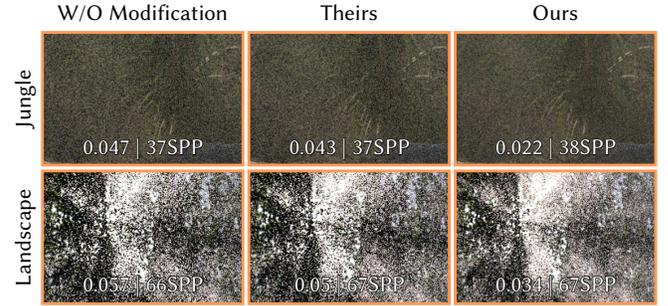


Fig. 18. We show the effect of null-collision probability bias (NCPB) on normalized density sampling (NDS). **Left** shows NDS without NCPB. **Center** and **right** show NDS with NCPB based on Eq. 40 and Eq. 41, respectively.

In Sec 3.3 of Villemin et al. [2018], the authors purpose to modify the real collision probability  $P_{\text{real}}$  to be  $P'_{\text{real}}$ :

$$P'_{\text{real}} = P_{\text{real}}^{\frac{1}{\gamma'}} \text{ with } \gamma' = 1 + (1 - \langle T_r(\mathbf{x}_0, \mathbf{x}_s) \rangle)(\gamma - 1),$$

where  $\langle T_r(\mathbf{x}_0, \mathbf{x}_s) \rangle$  is the transmittance estimation of the whole ray, and the paper sets  $\gamma = 2$  for all test cases. This results in:

$$\gamma' = 2 - \langle T_r(\mathbf{x}_0, \mathbf{x}_s) \rangle, \quad (40)$$

yielding a value in the range  $[1, 2]$ .

In thin media, the goal is to increase the volume scattering probability. Therefore we would like  $P'_{\text{real}}$  to be as large as possible. For a given  $P_{\text{real}} \in (0, 1)$ ,  $P'_{\text{real}}$  becomes larger when the exponent  $\frac{1}{\gamma'}$  is smaller, which occurs when  $\gamma$  is larger.

However,  $T_r(\mathbf{x}_0, \mathbf{x}_s)$  is high for thin volumes, resulting in  $\gamma'$  being close to 1, causing only a minor change from  $P_{\text{real}}$  to  $P'_{\text{real}}$ . We believe the authors intended for  $\gamma'$  to be high when  $T_r(\mathbf{x}_0, \mathbf{x}_s)$  is high. Therefore, we modify the equation to:

$$\gamma' = 1 + \langle T_r(\mathbf{x}_0, \mathbf{x}_s) \rangle, \quad (41)$$

which still keeps  $\gamma' \in [1, 2]$ , but increases  $P'_{\text{real}}$  further from  $P_{\text{real}}$  in thin media.

We compare the effects of the two  $\gamma'$  implementations (Eq. 40 and Eq. 41) in Fig. 18. Using our formula results in achieving a  $P_{\text{vol}}$  closer to the target  $P_{\text{vol}}^*$  and rendering with less error in both scenes. This experiment supports our observation.