Intelligent Computing
A SCIENCE PARTNER JOURNAL

RESEARCH ARTICLE

PARS - Path recycling and sorting for efficient cloud tomography

Ido Czerninski*[*] and Yoav Y. Schechner

Viterbi Faculty of Electrical and Computer Engineering, Technion–Israel Institute of Technology, Haifa, Israel.

*Address correspondence to: idoczer@gmail.com

Citation: Czerninski I, Schechner YY. Path recycling and sorting for efficient cloud tomography. *Intell. Comput.* 2023;2:Article 0007. https://doi.org/10.34133/icomputing.0007

Submitted 3 August 2022
Accepted 3 December 2022
Published 8 March 2023

Copyright © 2023 Ido Czerninski and Yoav Y. Schechner. Exclusive licensee Zhejiang Lab. No claim to original U.S. Government Works. Distributed under a Creative Commons Attribution License (CC BY 4.0).

Inverse rendering estimates scene characteristics from image data. We derive an efficient framework for inverse rendering and specifically computed tomography (CT) of volumetric scattering objects. We focus on clouds, which have a key role in the climate system and require efficient analysis at a huge scale. Data for such reconstruction are multiview images of each cloud taken simultaneously. This acquisition mode is expected by upcoming future spaceborne imagers, such as CloudCT. Prior art shows that scattering CT can rely on Monte–Carlo (MC) light transport. This approach usually iterates differentiable radiative transfer, requiring many sampled paths per iteration. We present an acceleration approach: path recycling and sorting (PARS). It efficiently uses paths from previous iterations for estimating a loss gradient at the current iteration. This reduces the iteration run time. PARS enables further efficient realizations. Specifically, sorting paths according to their size accelerates implementations on a graphical processing unit (GPU). PARS, however, requires a correction operation for unbiased gradient estimation. This can be achieved by utilizing a well-established concept from MC integration methods, as we show in this paper. We derive the theory of PARS and demonstrate its efficiency on cloud tomography of both synthetic and real-world scenes. Moreover, we demonstrate PARS on simple reflectometry examples.

Downloaded from https://spj.science.org at Technion Israel Institute of Technology on October 07, 2024

Introduction

Climate has an important impact on our life. Thus, advancing climate research gains interest in the computer vision community [1–5] as well as other communities that use computer vision, such as remote sensing [6–12] and geophysics [6,13–16]. The climate is strongly affected by interaction with clouds [17], as they fully control the water cycle of the earth. To reduce major errors in climate predictions [18–20], this interaction requires a much finer analysis of clouds. Current remote sensing data analysis assumes that the atmosphere and clouds are modeled as very broad and uniform layers. This assumption yields biases [21].

This problem can be bypassed by a special kind of computed tomography (CT), which is based on scattering. This idea is behind the CloudCT space mission, funded by the European Research Council (ERC). Scattering-based CT recovers 3-dimensional (3D) properties of volumetric scattering objects. Light propagation in clouds is complicated by high orders of scattering, which dominate the signals. In this paper, we argue that this complexity suits the domain of image rendering. Rendering synthesizes images using a known physics-based description of a scene. Rendering is a major branch of computer graphics; thus, technologies have been developed to make it fast and scalable. These include graphical processing units (GPUs) [22–26] and Monte–Carlo (MC) light transport [27,28]. We, on the other hand, rather aim at an inverse problem. Modern rendering algorithms are differentiable; they can be used in gradient-based solutions to inverse problems. This approach is often referred to as inverse rendering.

Inverse rendering appears in computer vision and graphics problems, including reflectometry [29,30], speckle analysis [31,32], and scattering CT [3–5,26,33–38]. However, the current efficiency of rendering does not translate trivially to efficient inverse rendering. Available implementations are largely optimized for the forward problem [24], rather than its inverse. This deficiency is very significant in the context of scattering CT of atmospheric content. The domain there is huge, exacerbating efficiency and scalability problems. Another challenge posed by the setup scale is that it must be passive: The radiation source is the uncontrollable sun (Fig. 1), and multiple cameras observe the scene.

Recent work pays attention to the efficiency of inverse rendering. Sde-Chen et al. [5] suggested that scattering CT can be assisted by a neural network, which is trained using differentiable rendering [39]. Also, Sde-Chen et al. [5] argued that the neural inference result is better used as initialization, to be proceeded with gradient-based optimization of a physics-based radiative transfer model. Hence, efficient differentiable rendering is key for inverse rendering, also when neural nets are used. Zhang et al. [40] introduced a sampling framework that efficiently estimates derivatives of arbitrary scene variables. Mitsuba 2 [26] developed computations of rendering derivatives using automatic differentiation. Studies of Nimier-David et al. [25] and Vicini et al. [34] improved on the study of Nimier-David et al. [26] by efficiently combining computations of analytical derivatives in a rendering process. The Mitsuba rendering framework [26,41] may be used to implement the method presented in this paper. However, support for heterogeneous volumes with

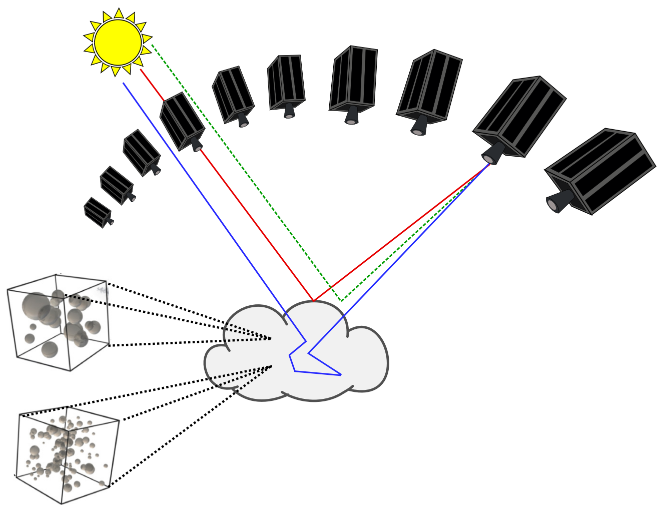

Fig. 1. Cloud tomography. A formation flight captures simultaneous multiview images of a cloud. These images are later used to recover the volume of the captured cloud. Credit: V. Holodovsky, M. Tzabari, and A. Levis.

spatially-varying phase functions, which is required for cloud tomography, is missing at the time of writing.

This paper introduces an efficient framework for the recovery of 3D distribution of scatterers, focusing on clouds. We propose a different direction for efficient inverse rendering. Our main insight is that inverse problems usually involve iterations, which incrementally refine scene variables. The iterations involve rendering operations. So, an inefficient aspect of rendering becomes amplified when run hundreds of times during iterated refinements. To counter this, we propose to recycle paths from prior iterations during inverse rendering. Moreover, we developed a differentiable rendering engine on a GPU that exploits path recycling. This is done by sorting the paths according to their size.

Previous work considered the concept of path reusing [42–44], but in the context of reducing forward rendering noise. This is done by sharing paths between adjacent pixels. We rather aim on exploiting similarities of consecutive iterations of a gradient-based algorithm to accelerate inverse rendering. Moreover, several sorting frameworks have been suggested [24,45,46] for rendering (not inverse). While they do not sort paths by their size, they may benefit path recycling and sorting (PARS).

The estimated variables change between iterations. Running fixed paths over a varying scene biases inverse rendering. Fortunately, previous work [37,47] showed that MC light transport can correct such discrepancies. In this paper, we use the ability to correct such discrepancies, to keep inverse rendering unbiased.

While unbiased, recycling may increase the error variance of rendered images. However, our aim is not image rendering per se, but an estimation of scene variables in a computer vision task. Indeed, our approach converges faster to a lower scene estimation error. Our approach is relevant to other inverse rendering problems. Thus, this paper demonstrates simple reflectometry using our approach.

Materials and Methods

This section first gives the necessary background for PARS. Then, this section introduces the PARS algorithm. In The forward model section, the image formation model is formulated as an integral over a set of light paths, as in [28]. Then, using previous knowledge [28], this section shows how the path integral can be estimated by MC rendering. Finally, a discussion is provided on the difficulties in the adaption of MC rendering to a GPU. In the Inverse problem section, we build upon [35,38] and formulate inverse rendering as an inverse problem. Then, we review how MC rendering is often used for estimating the gradient of an appearance loss. Afterward, we provide a contribution by deriving an analytical form of the gradient that considers 2 particle types in the atmosphere. In the Path recycling and sorting section, we introduce PARS—our method for fast, efficient, and scalable cloud tomography. Then, we give all the necessary derivations that justify the recycling of paths and adjust the MC rendering equations accordingly.

The forward model

The theoretical forward model is defined through a path integral. The integral is estimated using MC. This background preludes the adaption to scattering CT.

Path integral formulation

Vector $\boldsymbol{\omega}'$ is the 3D direction of incoming irradiance. Radiance is reflected to $\boldsymbol{\omega}$. A surface at 3D location $\boldsymbol{x}$ has normal $n(\boldsymbol{x})$ and bidirectional reflectance distribution function (BRDF) $\mathrm{f_r}(\boldsymbol{x},\boldsymbol{\omega}' \to \boldsymbol{\omega})$. Several scattering particle types exist in a medium. Each has an extinction coefficient $\beta^{\text{type}}(\boldsymbol{x})$, a scattering phase function $\mathrm{f_p^{type}}(\boldsymbol{x},\boldsymbol{\omega}' \to \boldsymbol{\omega}) = \mathrm{f_p^{type}}(\boldsymbol{x},\boldsymbol{\omega}' \cdot \boldsymbol{\omega})$, and a single scattering albedo ϖ^{type}. In an ensemble of particles, the total extinction coefficient, effective single scattering albedo, and total phase function are respectively

$$\beta(\boldsymbol{x}) = \sum_{\text{type}} \beta^{\text{type}}(\boldsymbol{x}),\ \varpi(\boldsymbol{x}) = \left[\sum_{\text{type}} \varpi^{\text{type}} \beta^{\text{type}}(\boldsymbol{x})\right] / \beta(\boldsymbol{x}),$$
$$\mathrm{f_p}(\boldsymbol{x},\boldsymbol{\omega}' \cdot \boldsymbol{\omega}) = \left[\sum_{\text{type}} \varpi^{\text{type}} \beta^{\text{type}}(\boldsymbol{x}) \mathrm{f_p^{type}}(\boldsymbol{x},\boldsymbol{\omega}' \cdot \boldsymbol{\omega})\right] / [\varpi(\boldsymbol{x})\beta(\boldsymbol{x})] \tag{1}$$

The transmittance between any 2 locations $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ is

$$T(\boldsymbol{x}_1,\boldsymbol{x}_2) \equiv exp\left(-\int_{\boldsymbol{x}_1}^{\boldsymbol{x}_2} \beta(\boldsymbol{x}) d\boldsymbol{x}\right) \tag{2}$$

A path of size $B+1 = |\boldsymbol{X}|$ from $\boldsymbol{x}_0$ to $\boldsymbol{x}_B$ is

$$\boldsymbol{X} = (\boldsymbol{x}_0,\boldsymbol{x}_1, \dots \boldsymbol{x}_b \dots ,\boldsymbol{x}_B) \tag{3}$$

where b is an interaction index. Let $d\boldsymbol{X} = \prod_{b=0}^{B} d\boldsymbol{x}_b$ be a path differential: It is a product of differential volume and surface elements [28]. Consider a light source at $\boldsymbol{x}_{\text{light}}$, whose radiance is L_e. Detector d has a response $W_d(\boldsymbol{x},\boldsymbol{\omega})$ to radiance propagating toward $\boldsymbol{\omega}$ (around $\boldsymbol{\omega}_d$) at $\boldsymbol{x}_d$, as in Fig. 2. Let

$$\mathcal{M} = \{m_v\}_{v=1}^{N_v} \tag{4}$$

be a set of N_v sought variables. For example, in scattering CT, $\mathcal{M}$ may be the extinction coefficient β in all voxels of a grid. In reflectometry, $\mathcal{M}$ can be a parameterization of $\mathrm{f_r}$. A forward model $\mathcal{F}_d(\mathcal{M})$ of the measured intensity at d is given by the path integral formulation of light transport [28]. This formulation

Downloaded from https://spj.science.org at Technion Israel Institute of Technology on October 07, 2024

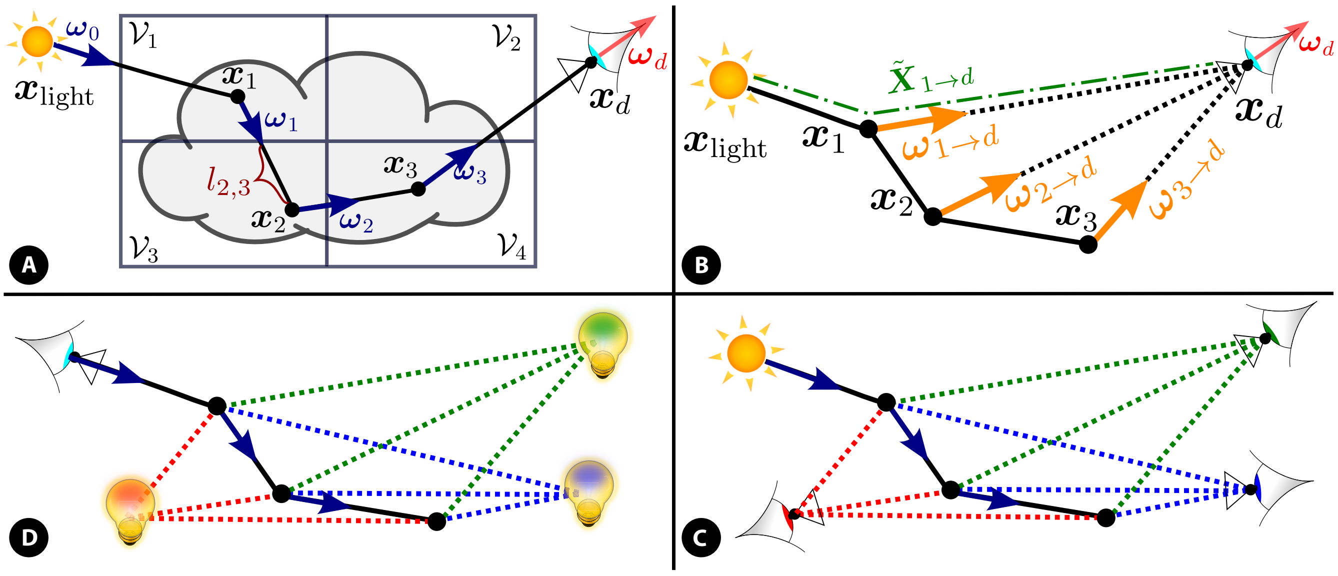

Fig. 2. Forward sampling (A) starts at x_{light} in direction ω_0. At x_1, the path scatters to ω_1 by interaction with a particle. Voxel 3 has domain $\mathcal{V}_3$. Its intersection with line segment x_1x_2 has length $l_{2,3}$. At each interaction, NEE (B) sends a ray to the detector (dotted lines). As defined in Eq. 10, $X^{1\to d} = (x_{\text{light}}, x_1, x_d)$ (dashed line). When the scene consists of a single light source and multiple viewpoints, forward sampling (C) allows for efficient implementations. The reason is that any path from the light source contributes to multiple cameras at each scattering location. These contributions can be parallelized. This scenario is relevant to cloud tomography. On the other hand, when the scene consists of multiple light sources and a single viewpoint, efficient implementations can be achieved by backward sampling (D). These contributions can be parallelized.

considers the set $\mathcal{P}$ of all possible paths X for which $x_0 = x_{\text{light}}$ and $x_B = x_d$:

$$\mathcal{F}_d(\mathcal{M}) = \int_{\mathcal{P}} f_d(\mathcal{M}, X)dX \tag{5}$$

The term $f_d(\mathcal{M},X)$ is the measurement contribution function (MCF) [28]. Intuitively, f_d is the contribution of path X to a measurement by the forward model. A detailed definition using Eqs. 1 and 2 is given in Eqs. S13 to S22.

MC light transport

Generally, Eq. 5 cannot be solved analytically. However, being an integral, it can be estimated by MC. Let $\mu_d(\mathcal{M},X)$ be any fixed probability density function (PDF) over $\mathcal{P}$. Let $\mathbb{E}$ express expectation. Then [28], Eq. 5 can be written as

$$\mathcal{F}_d(\mathcal{M}) = \int_{\mathcal{P}} \frac{f_d(\mathcal{M}, X)}{\mu_d(\mathcal{M}, X)}\mu_d(\mathcal{M}, X)dX = \mathbb{E}_{X\sim\mu_d}\left[\frac{f_d(\mathcal{M}, X)}{\mu_d(\mathcal{M}, X)}\right] \tag{6}$$

In this notation, path X is a random variable [37] with distribution μ_d. MC light transport estimates Eq. 6. This requires sampling paths through the scene [27,28], using μ_d. Consider a set $\Phi = \{X_i\}_{i=1}^N$ of $N = |\Phi|$ such paths, where i is a path index. Define

$$C_d(\mathcal{M},X) = \frac{f_d(\mathcal{M},X)}{\mu_d(\mathcal{M},X)} \tag{7}$$

Then, an unbiased MC estimation [28,47] of Eq. 6 is the following mean:

$$\hat{\mathcal{F}}_d(\mathcal{M},\Phi) = \frac{1}{|\Phi|}\sum_{X_i\in\Phi} C_d(\mathcal{M},X_i) \tag{8}$$

The integral (Eq. 5) is thus estimated by a finite sum. Eq. 8, is unbiased, but a poor choice of μ_d increases the estimation variance, as discussed in section S2.2.

There are several MC sampling techniques [47]. For example, forward sampling traces paths from a light source to a camera. Backward sampling traces paths from a camera to a light source. For simplicity, this section focuses on forward sampling, although our approach is implemented for both techniques.

The direction vector from x_b to x_{b+1} (Fig. 2) is ω_b. In a scattering medium, MC may sample a path as follows:

(i) Direction ω_0 is uniformly sampled and a ray $\mathcal{R}_0 = (x_{\text{light}}, \omega_0)$ is initialized.

Per iteration $b = 1,2,\ldots$:

(ii) On ray $\mathcal{R}_{b-1}$, a random location x_b is sampled through these substeps:

ii.a Uniformly sample $u \in [0,1]$.

ii.b Set $\tau = -\ln(1-u)$.

ii.c Numerically find an interaction location x_b such that $\tau = \int_{x_{b-1}}^{x_b} \beta(x)dx$.

(iii) Sample ω_b using $f_p(x_b, \omega_{b-1}\cdot\omega_b)$.

(iv) End if the path leaves the medium or hits a detector. Else, iterate again.

Steps (i) to (iv) yield not only a path $X = (x_{\text{light}},\ldots,x_B)$ but also, implicitly, its corresponding $\mu_d(\mathcal{M},X)$. Then, the realization of Eq. 7 is

$$C_d(\mathcal{M},X) = 4\pi L_e W_d(x_B, \omega_{B-1}) \prod_{b=1}^{B-1} \varpi(x_b) \tag{9}$$

Next event estimation

The spatial support of $W_d(x,\omega)$ is that of a small pinhole at x_d. If x_B is away from x_d, then $C_d(\mathcal{M},X) = 0$ in Eq. 9. For paths that efficiently estimate Eq. 8, next event estimation (NEE) [47,48]

Downloaded from https://spj.science.org at Technion Israel Institute of Technology on October 07, 2024

is used. NEE sends a ray to $\boldsymbol{x}_d$ at each scattering event, closing a path from $\boldsymbol{x}_{\text{light}}$ to $\boldsymbol{x}_d$ (see Fig. 2). Recall Eq. 3. Express each $\boldsymbol{X}$ as $[(\boldsymbol{x}_0,\ldots,\boldsymbol{x}_b)\cup(\boldsymbol{x}_{b+1},\ldots,\boldsymbol{x}_B)]$. Then, define

$$\boldsymbol{X}^{b\to d} = (\boldsymbol{x}_0, \ldots, \boldsymbol{x}_b, \boldsymbol{x}_d) \tag{10}$$

For any b, $\boldsymbol{X}^{b\to d}$ starts at $\boldsymbol{x}_0 = \boldsymbol{x}_{\text{light}}$. NEE modifies Eq. 8, by using the set $\left\{\boldsymbol{X}_i^{b\to d}\right\}_{b=1}^{B}$ to compute B contributions of path $\boldsymbol{X}_i$:

$$\widehat{\mathcal{F}}_d(\mathcal{M},\Phi) = \frac{1}{|\Phi|}\sum_{X_i\in\Phi}\sum_{b=1}^{B} c_d^{\text{nee}}\left(\mathcal{M},\boldsymbol{X}_i^{b\to d}\right) \tag{11}$$

Here, $c_d^{\text{nee}}\left(\mathcal{M},\boldsymbol{X}^{b\to d}\right)$ is the contribution of the NEE path (10) to the estimated $\widehat{\mathcal{F}}_d(\mathcal{M},\Phi)$. It is similar to $C_d\left(\mathcal{M},\boldsymbol{X}^{b\to d}\right)$. However, $c_d^{\text{nee}}\left(\mathcal{M},\boldsymbol{X}^{b\to d}\right)$ weights in the probability P_{nee} of a path to exist unattenuated between $\boldsymbol{x}_b$ and $\boldsymbol{x}_d$:

$$c_d^{\text{nee}}\left(\mathcal{M},\boldsymbol{X}^{b\to d}\right) = C_d\left(\mathcal{M},\boldsymbol{X}^{b\to d}\right)P_{\text{nee}}\left(\mathcal{M},\boldsymbol{X}^{b\to d}\right) \tag{12}$$

Let $\boldsymbol{\omega}_{b\to d}$ be a direction vector from $\boldsymbol{x}_b$ toward $\boldsymbol{x}_d$. Then, the probability P_{nee} is

$$P_{\text{nee}}\left(\mathcal{M},\boldsymbol{X}^{b\to d}\right) = \text{f}_{\text{p}}\left(\boldsymbol{x}_b,\boldsymbol{\omega}_{b-1}\cdot\boldsymbol{\omega}_{b\to d}\right)T\left(\boldsymbol{x}_b,\boldsymbol{x}_d\right)\left\|\boldsymbol{x}_b-\boldsymbol{x}_d\right\|^{-2} \tag{13}$$

In forward sampling, each scattering event contributes to multiple viewpoints (Fig. 2). The computation of Eqs. 11 to 13 is deterministic and can be parallelized to all viewpoints. It is computationally far less costly than the sequential random sampling of a path $\boldsymbol{X}_i$. In a cloud tomography setup, illumination is by a single source (the sun) and imaging is by multiple cameras; thus, forward sampling is more efficient than backward sampling. Backward sampling is superior when a scene consists of a single camera and multiple light sources. In that case, NEE from each scattering event accounts for the contribution of all the light sources.

Parallelization on a GPU

MC light transport can be parallelized since paths are independent. A GPU excels at running programs that have high parallelism. However, a GPU architecture does not trivially benefit from this parallelism. A GPU executes tasks in blocks. Each block has usually 32 threads such that each thread represents a different path sampling instance. To fully utilize the GPU capabilities, all threads within the same block should execute the same commands. Unfortunately, different paths have different sizes. In traditional methods, the path size is known only after sampling ends. In the Path recycling and sorting section, we show that this can be countered using our approach: PARS.

The main building block of path sampling is the ability to sample random numbers, which are independent and uniform between 0 and 1. A GPU imitates this ability by generating a deterministic sequence, which is initialized by a scalar seed. This sequence has mathematical properties that appear as a random sequence having a uniform distribution. However, given a constant seed, this sequence is known completely. We exploit this property in the PARS algorithm section for encoding a whole path using only its corresponding seed.

Inverse problem

A vector of N_{d} true intensity measurements is $\boldsymbol{I}^{\text{gt}} = \left[I_1^{\text{gt}}, I_2^{\text{gt}}, \ldots, I_{N_{\text{d}}}^{\text{gt}}\right]$. Their corresponding forward model signals are set in a vector $\left[\mathcal{F}_1(\mathcal{M}), \ldots, \mathcal{F}_{N_{\text{d}}}(\mathcal{M})\right]$. An appearance loss is

$$\mathcal{L}\left(\mathcal{M}|\boldsymbol{I}^{\text{gt}}\right) = \frac{1}{2}\sum_{d=1}^{N_{\text{d}}}\left|\mathcal{F}_d(\mathcal{M}) - I_d^{\text{gt}}\right|^2 \tag{14}$$

An optimization problem seeks the true scene,

$$\widehat{\mathcal{M}} = \arg\min_{\mathcal{M}}\mathcal{L}\left(\mathcal{M}\,|\,\boldsymbol{I}^{\text{gt}}\right) \tag{15}$$

often using gradient-based iterations. This requires the gradient of $\mathcal{F}_d(\mathcal{M})$. Denote $\partial_{m_v} \triangleq \frac{\partial}{\partial m_v}$. The partial derivative of the forward model (5) is

$$\partial_{m_v}\mathcal{F}_d(\mathcal{M}) = \int_{\mathcal{P}}\partial_{m_v}f_d(\mathcal{M},\boldsymbol{X})d\boldsymbol{X} = \int_{\mathcal{P}}f_d(\mathcal{M},\boldsymbol{X})\partial_{m_v}\ln f_d(\mathcal{M},\boldsymbol{X})d\boldsymbol{X} \tag{16}$$

This is a path integral [49]. The analogy of Eqs. 5 and 16 can be used to estimate $\partial_{m_v}\mathcal{F}_d(\mathcal{M})$ through sampling as in the MC light transport section, followed by NEE as in the Next event estimation section. Then, similarly [30,35] to Eq. 11, an MC estimation of Eq. 16 is by this sum

$$\widehat{\partial}_{m_v}\mathcal{F}_d(\mathcal{M},\Phi) = \frac{1}{|\Phi|}\sum_{X_i\in\Phi}\sum_{b=1}^{B}c_d^{\text{nee}}\left(\mathcal{M},\boldsymbol{X}_i^{b\to d}\right)\partial_{m_v}\ln f_d\left(\mathcal{M},\boldsymbol{X}_i^{b\to d}\right) \tag{17}$$

From Eqs. 8, 14, and 17, the gradient of $\mathcal{L}$ is estimated by

$$\widehat{\partial}_{m_v}\mathcal{L}\left(\mathcal{M},\Phi|\boldsymbol{I}^{\text{gt}}\right) = \sum_{d=1}^{N_{\text{d}}}\left[\widehat{\mathcal{F}}_d(\mathcal{M},\Phi) - I_d^{\text{gt}}\right]\widehat{\partial}_{m_v}\mathcal{F}_d(\mathcal{M},\Phi) \tag{18}$$

Here, $\left[\widehat{\mathcal{F}}_d(\mathcal{M}) - \boldsymbol{I}_d^{\text{gt}}\right]$ is the estimation error, obtained by rendering. Then, plugging Eq. 17 in Eq. 18 results in

$$\widehat{\partial}_{m_v}\mathcal{L}\left(\mathcal{M},\Phi|\boldsymbol{I}^{\text{gt}}\right) = \frac{1}{|\Phi|}\sum_{X_i\in\Phi}\sum_{b=1}^{B}\sum_{d=1}^{N_{\text{d}}}\left[\widehat{\mathcal{F}}_d(\mathcal{M},\Phi) - I_d^{\text{gt}}\right]c_d^{\text{nee}}\left(\mathcal{M},\boldsymbol{X}_i^{b\to d}\right)\partial_{m_v}\ln f_d\left(\mathcal{M},\boldsymbol{X}_i^{b\to d}\right) \tag{19}$$

Scattering CT seeks properties of particles in a scene. Let us focus on 2 particle types: air molecules and cloud droplets. Their respective parameters are $\beta^{\text{a}}, \varpi^{\text{a}}, \text{f}_{\text{p}}^{\text{a}}$ and $\beta^{\text{c}}, \varpi^{\text{c}}, \text{f}_{\text{p}}^{\text{c}}$. Suppose $\beta^{\text{a}}(\boldsymbol{x})\varpi^{\text{c}},\varpi^{\text{a}}$ are known, and we seek $\beta^{\text{c}}(\boldsymbol{x})$. Approximate the cloud extinction coefficient in voxel v as a constant, β_v^{c}. Thus, $\mathcal{M} = \left\{\beta_v^{\text{c}}\right\}_{v=1}^{N_{\text{v}}}$. Voxel v has a domain $\mathcal{V}_v$ (see Fig. 2). Denote the line segment between $\boldsymbol{X}_{b-1}$ and $\boldsymbol{x}_b$ by $\overline{\boldsymbol{x}_{b-1}\boldsymbol{x}_b}$. The intersection of $\mathcal{V}_v$ with $\overline{\boldsymbol{x}_{b-1}\boldsymbol{x}_b}$ has length $l_{b,v}$. Let $D_b = \boldsymbol{\omega}_{b-1}\cdot\boldsymbol{\omega}_b$. Let $\mathbb{I}(\mathcal{A})$ be an indicator, which equals 1 if $\mathcal{A}$ occurs (0 otherwise). Then, we provide a contribution of a derivation that appears in section S4. This derivation yields:

$$\partial_{m_v}\ln f_d\left(\mathcal{M},\boldsymbol{X}^{b\to d}\right) = -\sum_{b'=1}^{b}l_{b',v} + \sum_{b'=1}^{b}\mathbb{I}\left(\boldsymbol{x}_{b'}\in\mathcal{V}_v\right)\left[\beta_v^{\text{c}} + \frac{\varpi^{\text{a}}\text{f}_{\text{p}}^{\text{a}}\left(\boldsymbol{x}_{b'}D_{b'}\right)}{\varpi^{\text{c}}\text{f}_{\text{p}}^{\text{c}}\left(\boldsymbol{x}_{b'}D_{b'}\right)}\beta_v^{\text{a}}\right]^{-1} \tag{20}$$

Downloaded from https://spj.science.org at Technion Israel Institute of Technology on October 07, 2024

Eq. 20 can be evaluated as a path $\boldsymbol{X}$ is traversed. For example, when a path $\boldsymbol{X}^{b\to d}$ traverses a distance l through a voxel v, it contributes a gradient contribution of

$$-l\left[\widehat{\mathcal{F}}_d(\mathcal{M},\Phi)-I_d^{\mathrm{gt}}\right]c_d^{\mathrm{nee}}\left(\mathcal{M},\boldsymbol{X}_i^{b\to d}\right) \tag{21}$$

Contributions to the gradient are accumulated to Eq. 20 as we traverse a path.

Path recycling and sorting

We propose PARS to accelerate inverse rendering. While each iteration may be accelerated independently [25,34], we exploit the iterative manner of a gradient algorithm. Consider 2 consecutive iterations and their respective extinction coefficients $\{\beta_v^{\mathrm{c}}\}_{v=1}^{N_{\mathrm{v}}}$. Since $\{\beta_v^{\mathrm{c}}\}_{v=1}^{N_{\mathrm{v}}}$ changes only slightly by a small gradient step, radiative transfer in the scene generally does not change drastically. PARS exploits this concept to largely eliminate the need for expensive path sampling. We suggest sampling paths only once each N_{r} (hyperparameter) iterations.

MC randomness is unfriendly to GPU realizations. GPUs work better on blocks having a similar path size, for compatibility with aspects mentioned in the Parallelization on a GPU section. So, we sort paths according to their size. Sorting alone speeds rendering run time by ≈× 4. Sorting requires overhead computations, but these need to be done only once per N_{r} iterations, right after path sampling. We now detail the method, together with an operation that maintains PARS unbiased.

PARS algorithm

PARS is illustrated in Fig. 3. A recycling period of N_{r} iterations starts at reference iteration t' and ends at $t'+N_{\mathrm{r}}$. The estimated set of scene variables at iteration t' is $\mathcal{M}_{\mathrm{ref}}=\mathcal{M}_{t'}$. The set of variables at a later iteration $t\geq t'$ is $\mathcal{M}_t$.

At reference iteration t':

{I} Path sampling: Following steps (i) to (iv) in the MC light transport section, sample paths. Here, sampling relies on $\mathcal{M}_{\mathrm{ref}}$, specifically in steps ii.c and (iii). The resulting set of paths is denoted $\Phi^{\mathrm{ref}}=\{\boldsymbol{X}_i\}_{i=1}^{N}$.

Complexity: Each sampling of a path segment crosses $\mathcal{O}\left(N_{\mathrm{voxel}}^{1/3}\right)$ voxels (as illustrated in Fig. 4).

Thus, path sampling time is $\mathcal{T}_{\mathrm{sample}}\in\mathcal{O}\left(N_{\mathrm{s}}N_{\mathrm{voxel}}^{1/3}\right)$, where

$$N_{\mathrm{s}}=\sum_{\boldsymbol{X}_i\in\Phi^{\mathrm{ref}}}|\boldsymbol{X}_i| \tag{22}$$

{II} Path sorting: Following path sampling, sort paths according to their size and associate them to subsets. Each subset has paths of the same size q,

$$\Phi_q^{\mathrm{ref}}=\left\{\boldsymbol{X}_i\in\Phi^{\mathrm{ref}}:|\boldsymbol{X}_i|=q\right\}. \tag{23}$$

Complexity: Sorting time is $\mathcal{T}_{\mathrm{sort}}\in\mathcal{O}(|\Phi_{\mathrm{ref}}|\ln|\Phi_{\mathrm{ref}}|)$.

For iteration $t=t',t'+1,\ldots,t'+N_{\mathrm{r}}-1$:

{III} Rendering through recycling: Compute $\forall q$: $\widehat{\mathcal{F}}_d\left(\mathcal{M}_t,\Phi_q^{\mathrm{ref}}\right)$.

In other words, at iteration t, a recycled path set (Φ_{ref}) from iteration $t'\leq t$ is used for rendering images relating to the medium at t. Then,

$$\widehat{\mathcal{F}}_d\left(\mathcal{M}_t,\Phi^{\mathrm{ref}}\right)=\frac{1}{|\Phi^{\mathrm{ref}}|}\sum_q|\Phi_q^{\mathrm{ref}}|\,\widehat{\mathcal{F}}_d\left(\mathcal{M}_t,\Phi_q^{\mathrm{ref}}\right). \tag{24}$$

Complexity: Let N_{view} be the number of viewpoints: this is also the number of NEE rays projected from each of the N_{s} interaction points. Each path segment and each NEE ray pass $\mathcal{O}\left(N_{\mathrm{voxel}}^{1/3}\right)$ voxels. So, without sorting, rendering time is $\mathcal{T}_{\mathrm{render}}\in\mathcal{O}\left(N_{\mathrm{view}}N_{\mathrm{s}}N_{\mathrm{voxel}}^{1/3}\right)$. Because of sorting, the rendering formulation in Eq. 24 is GPU friendly, as all paths in Φ_q^{ref} are known to have the same size. As a result, sorting accelerates $\mathcal{T}_{\mathrm{render}}$ to $\eta_{\mathrm{sorted}}\mathcal{T}_{\mathrm{render}}$, where $\eta_{\mathrm{sorted}}<1$.

The N_{view} viewpoints define a set of distinct $\boldsymbol{x}_d$ vectors. All NEE contributions are stored in memory as the set

$$C=\left\{c_d^{\mathrm{nee}}\left(\mathcal{M}_t,\boldsymbol{X}_i^{b\to d}\right)\right\}_{\forall i,b,\boldsymbol{x}_d} \tag{25}$$

Downloaded from https://spj.science.org at Technion Israel Institute of Technology on October 07, 2024

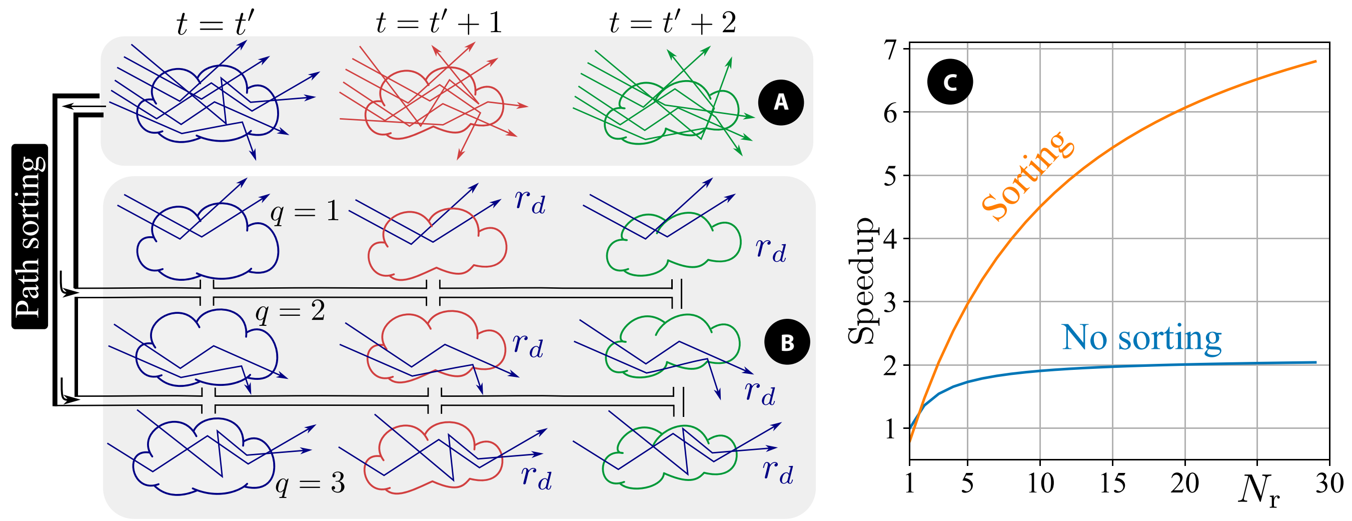

Fig. 3. (A) Traditionally, paths are sampled in each iteration. (B) PARS: At t', paths are sampled and then sorted by their size q. Sorted paths remain fixed and reused for N_{r} iterations. This better utilizes GPUs. For unbiased estimation of rendering and its gradient, path $\boldsymbol{X}_i$ has a correction factor $r_d(\mathcal{M}_t,\boldsymbol{X}|\mathcal{M}_{\mathrm{ref}})$. (C) PARS speedup.

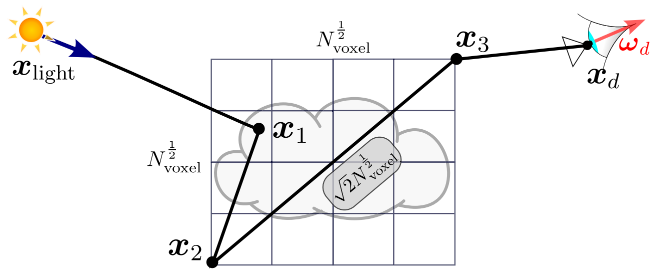
x_{light}
$N_{\text{voxel}}^{\frac{1}{2}}$
x_3
ω_d
x_d
x_1
$N_{\text{voxel}}^{\frac{1}{2}}$
$\sqrt{2}N_{\text{voxel}}^{\frac{1}{2}}$
x_2

Fig. 4. Sampling complexity illustration. Assume we have N_{voxel} voxels ordered in a 2D squared grid. The longest possible path segment is achieved by the diagonal line. In this example, the diagonal line is the line segment $\overline{x_2x_3}$. The diagonal line has $\sqrt{2}N_{\text{voxel}}^{\frac{1}{2}} \in \mathcal{O}\left(N_{\text{voxel}}^{\frac{1}{2}}\right)$ voxels. We, however, deal with 3D scenes. Thus, the path crosses $\mathcal{O}\left(N_{\text{voxel}}^{\frac{1}{3}}\right)$.

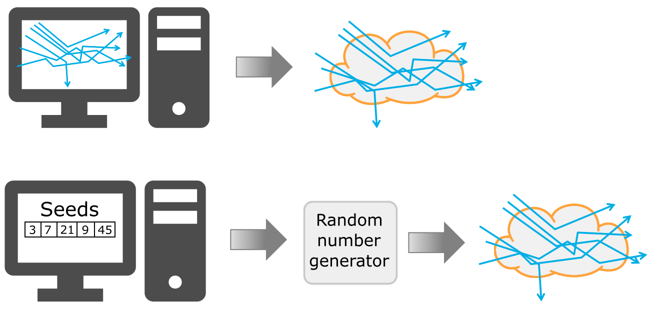

Fig. 5. Seed decoding. Top: Normally, path recycling requires storing them in memory. Bottom: We suggest storing only the scalar seeds that were used to generate the random paths. Then, the random number generator of a GPU can replicate the paths completely, because they are deterministic sequences given their corresponding seeds.

This requires $\mathcal{O}(N_s N_{\text{view}})$ memory.

{IV} Gradient through recycling: Plugging Eqs. 24 and 25 in Eq. 19 directly estimates $\left\{\partial_{m_v}\mathcal{L}\right\}_{v=1}^{N_{\text{voxel}}}$. For each voxel v, the partial derivative $\partial_{m_v}\mathcal{L}$ in Eq. 19 depends on v only through $\partial_{m_v} lnf_d$. The latter is given in Eq. 20.

Complexity: As mentioned in the Inverse problem section, we accumulate gradient contributions, while we traverse the paths. Then, as in {III} above, the time for gradient estimation without sorting is $\mathcal{T}_{\text{grad}} \in \mathcal{O}\left(N_{\text{view}}N_sN_{\text{voxel}}^{1/3}\right)$. We save time by using the stored $\mathcal{C}$ of Eq. 25 in Eq. 19.

In PARS, the gradient uses sorted paths Φ_q^{ref}, in analogy to Eq. 24. This is GPU friendly, as all paths in Φ_q^{ref} are known to have the same size. As a result, sorting accelerates $\mathcal{T}_{\text{grad}}$ to $\eta_{\text{sorted}}\mathcal{T}_{\text{grad}}$.

PARS, by definition, requires retrieving of paths. Paths can be retrieved using only a saved seed that the GPU had used to sample the path [50], as hinted in the Parallelization on a GPU section, illustrated in Fig. 5.

Then, retrieving paths does not add memory beyond storing $\mathcal{C}$. In inverse rendering, scene variables change between iterations. However, the paths here are sampled by the process in the MC light transport section, using scene variables of a different iteration. Thus, relying on Φ^{ref} in Eq. 24 introduces an estimation bias. However, Eq. 6 holds for any PDF μ_d. Therefore, in the next section, we use this freedom to correct this discrepancy.

Correction factor for unbiased estimation

In iteration t, we need to estimate $\mathcal{F}_d$ and $\partial_{m_v}\mathcal{F}_d$ using paths that correspond to an inconsistent estimated scene from another iteration. This discrepancy leads to bias that can potentially derail the optimization. Recall from Eq. 6 that MC provides freedom to choose a general PDF for paths [37]. We suggest using $\mu_d(\mathcal{M}_{\text{ref}}, X)$ as the PDF. Define a PARS correction factor,

$$r_d(\mathcal{M}_t, X|\mathcal{M}_{\text{ref}}) \triangleq \frac{\mu_d(\mathcal{M}_t, X)}{\mu_d(\mathcal{M}_{\text{ref}}, X)} \tag{26}$$

which is a ratio between PDFs. Then, recycling modifies Eq. 7 as follows:

$$C_d(\mathcal{M}_t, X|\mathcal{M}_{\text{ref}}) = \frac{f_d(\mathcal{M}_t, X)}{\mu_d(\mathcal{M}_{\text{ref}}, X)} = C_d(\mathcal{M}_t, X)r_d(\mathcal{M}_t, X|\mathcal{M}_{\text{ref}}) \tag{27}$$

Then, the NEE contribution from Eqs. 13 and 12 changes to:

$$c_d^{\text{nee}}(\mathcal{M}_t, X^{b\to d}|\mathcal{M}_{\text{ref}}) = C_d(\mathcal{M}_t, X^{b\to d}|\mathcal{M}_{\text{ref}})P_{\text{nee}}(\mathcal{M}_t, X^{b\to d}) \tag{28}$$

By using Eqs. 26 to 28 in Eqs. 11 to 18, PARS yields unbiased differentiable rendering. This is achieved by using a factor r_d that corrects the bias that originates from discrepancies between $\mathcal{M}_t$ and $\mathcal{M}_{\text{ref}}$. However, these discrepancies increase the variance of rendering and the gradient. Nevertheless, usually increased variance is tolerated during repeated gradient-based iterations. PARS pays off: It reduces computations and enables efficient use of a GPU by sorting.

Eq. 26 is a ratio of PDFs. Let T_t, β_t, $\mathrm{f}_{\mathrm{p}}^t$ denote the transmittance, extinction coefficient, and phase function of the medium at iteration t (see the Path integral formulation section). Let T_{ref}, β_{ref}, $\mathrm{f}_{\mathrm{p}}^{\text{ref}}$ denote the transmittance, extinction coefficient, and phase function of $\mathcal{M}_{\text{ref}}$ (at t'). Then, we show in section S3.2 that for scattering CT:

$$r_d(\mathcal{M}_t, X|\mathcal{M}_{\text{ref}}) = \frac{T_t(x_{B-1}, x_B)}{T_{\text{ref}}(x_{B-1}, x_B)} \prod_{b=1}^{B-1} \frac{\beta_t(x_b)\mathrm{f}_{\mathrm{p}}^t(x_b, D_b)T_t(x_b, x_{b-1})}{\beta_{\text{ref}}(x_b)\mathrm{f}_{\mathrm{p}}^{\text{ref}}(x_b, D_b)T_{\text{ref}}(x_b, x_{b-1})}. \tag{29}$$

Experimental design

We performed several inverse rendering experiments to demonstrate the achievement of PARS. For this cause, we implemented a 3D rendering engine to solve both scattering CT and reflectometry. To exploit parallelism, we implemented the engine on a GPU, using Numba [51]. Our engine supports both backward and forward sampling. The optimization is performed using momentum gradient descent [52].

Results

This section describes the setting of several inverse rendering experiments, followed by quantitative and visual results that are achieved by using PARS.

Downloaded from https://spj.science.org at Technion Israel Institute of Technology on October 07, 2024

Synthetic data

Consider 2 synthetic scenes: a solitude cloud [53] and a cloud field [53]. All have constant known $\beta^{a} = 0.04\left[\frac{1}{km}\right]$, $\varpi^{c} = 0.99$, $\varpi^{a} = 0.912$ [54] and are illuminated by the distant sun at the zenith. Air and clouds yield Rayleigh [55] and a Henyey–Greenstein [50] phase functions, respectively. The scenes are captured by 9 perspective cameras (Fig. 6 and Table 1). Rendering uses forward sampling and $|\Phi| = 5 \cdot 10^{7}$ paths. The data images $\mathbf{I}^{gt}$ include noise, as they are generated by random MC. In the solitude cloud scene, $SNR \approx \frac{1}{30}$.

The reconstruction process starts with space carving [56] to obtain a photo-hull $\mathcal{H}$ of the cloud. Then, we use a homogeneous $\overline{\beta}, \forall \mathbf{x} \in \mathcal{H}$ as an initialization for the scattering CT described in The forward model, Inverse problem, and Path recycling and sorting sections. We use $N_r = 10$ and a Tesla V100 DGXS 32GB. To assess the reconstruction quality, we follow [38] and use

$$\epsilon = \left\|\mathcal{M}^{*} - \widehat{\mathcal{M}}\right\|_{1} / \|\mathcal{M}^{*}\|_{1}, \delta = \left[\|\mathcal{M}^{*}\|_{1} - \left\|\widehat{\mathcal{M}}\right\|_{1}\right] / \|\mathcal{M}^{*}\|_{1} \tag{30}$$

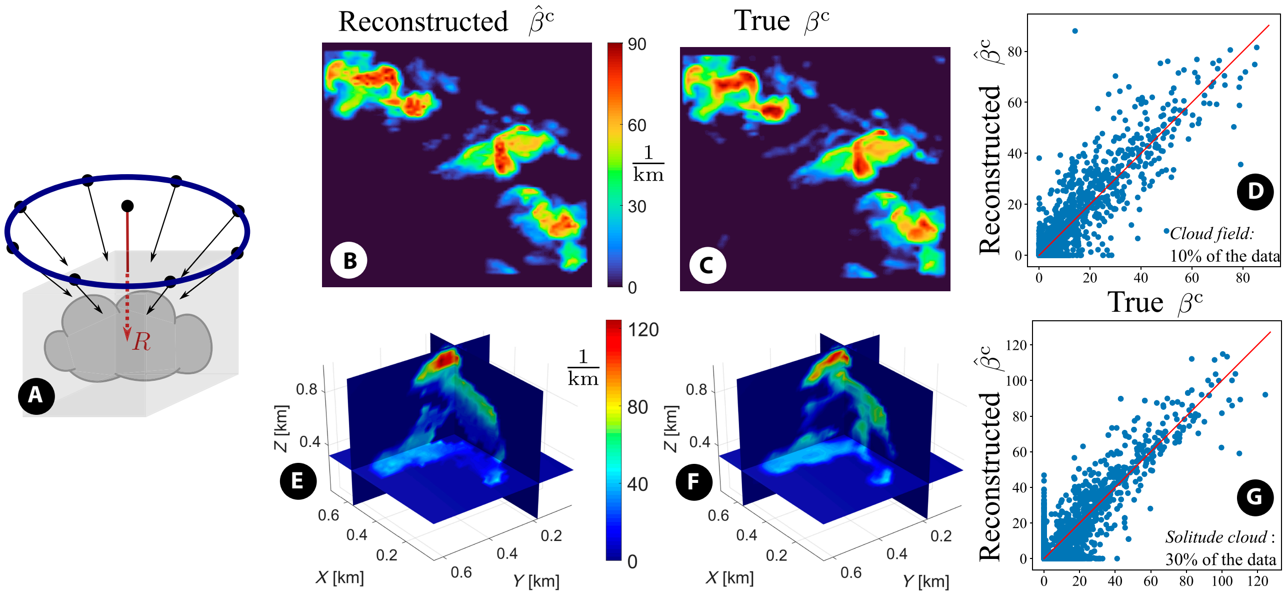

Fig. 6. (A) Cameras are at the zenith and on a horizontal ring. (B and C) 3D plots of the cloud field scene. These plots are rendered using maximum intensity projection (MIP). (D) Corresponding scatterplot. (E and F) 3D slices of the solitude cloud scene and a corresponding scatterplot (G).

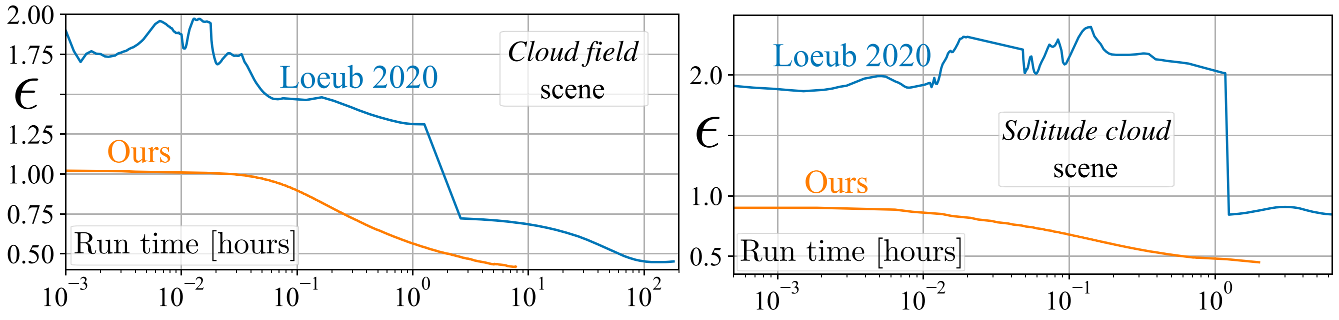

Fig. 7. Comparison to [38]. Left: The same reduction in ϵ as [38] is achieved an order of magnitude faster in the cloud field. Right: In the single cloud, PARS converges to a lower ϵ compared to [38].

Table 1. Scene properties. Cameras are at distance R from the domain center.

Scene	Bounding box	Max β^c	Voxels	Pixels	Camera	R
Solitude cloud	$0.6 \times 0.7 \times 1$ km^3	127 km^{-1}	$32 \times 36 \times 25$	76×76	29°	2 km
Cloud field	$1.6 \times 1.7 \times 1.1$ km^3	90 km^{-1}	$79 \times 84 \times 27$	86×86	33°	2 km

Downloaded from https://spj.science.org at Technion Israel Institute of Technology on October 07, 2024

The cloud field reconstruction converged an order of magnitude faster using PARS (Fig. 7), compared to [38], whose code is available at [53]. PARS reached results for the solitude cloud scene that are visually superior and quantitatively comparable to [38]. Results are displayed in Table 2 and Fig. 6.

An ablation study shows that PARS indeed accelerates inverse rendering. We validate that run time reduction is attributed to PARS, not only to a GPU. Therefore, we inverse rendered the solitude cloud and the cloud field using several values of N_r, with and without sorting. The results in Fig. 8 show that using $N_r = 10$ in our experiments is a sound choice. It is large enough to mask the path sampling overhead while maintaining a reasonable variance.

Table 2. Quantitative results and run time comparison. We compare to [2,38], by using results from [38]. Results that are not provided by [38] are marked by "-."

	Solitude cloud			Cloud field		
Method	ϵ%	δ%	Run time (h)	ϵ%	δ%	Run time (h)
Levis et al. [2]	72	−3.6	1	70	30	-
Loeub et al. [38]	85	−5.5	6.5	45	−28.8	127
PARS (ours)	45	4	2	42	−2.2	7

Real-world data

We follow the experimental approach of [2,3] and use real-world data acquired by JPL's AirMSPI [57], which flies onboard NASA's ER-2. Data use the 660-nm channel of a Pacific flight, acquired on 2013 February 6 at 20:27 GMT, around coordinates 32N 123W. This flight used a push-broom camera observing 9 viewing angles: ±70.5°, ±60°, ±45.6°, ±26.1°, and 0°, where ± indicates fore- and aft-views along the northbound flight path. We examine an atmospheric domain of size $2.75 \times 3.65 \times 2[\mathrm{km}^3]$, divided to $68 \times 91 \times 49$ voxels, each of size $40 \times 40 \times 40[\mathrm{m}^3]$.

Our implementation of backward sampling is the rendering engine. Exhaustive search analyzing the background of $\boldsymbol{I}^{\mathrm{gt}}$ found the ocean albedo to be 0.05. We initialize $\boldsymbol{\beta}$ inside the photo-hull $\mathcal{H}$ with $\widehat{\beta} = 1$. The number of paths is $|\Phi| = 7 \cdot 10^8$. We set $N_r = 10$. There is no ground truth for the cloud content. Hence, we check for consistency using cross-validation [2,4], by excluding one image from the recovery process. Thus, the optimization used 8 of the 9 raw views. The results are displayed in Fig. 9. The optimization converged after 70 min. When comparing pixel values to the ground truth, we achieve $\epsilon = 25.1\%$, $\delta = 1\%$.

PARS in reflectometry

Besides scattering CT, PARS can be used for other inverse rendering problems. For example, reflectometry retrieves an object's BRDF. There are several BRDF representations [30,58]. We show simple examples using a Phong BRDF [59]:

$$f_r(\boldsymbol{x},\boldsymbol{\omega}' \to \boldsymbol{\omega}) = \frac{\kappa_d}{\pi} + \kappa_s \frac{\gamma + 2}{2\pi}(\boldsymbol{\omega}^s \cdot \boldsymbol{\omega})^{\gamma}. \quad (31)$$

Here, $\gamma \geq 0$, $0 \leq \kappa_d + \kappa_s \leq 1$ and $\boldsymbol{\omega}^s$ is the specular direction of $\boldsymbol{\omega}'$ with respect to the normal $n(\boldsymbol{x})$. Two scenes include 14 small specular spheres and a large sphere in a box [60]. In scene Phong, κ_d is known. The set of sought variables is $\mathcal{M}_{\mathrm{phong}} = [\kappa_s, \gamma]$ of the large sphere. The number of paths is $|\Phi| = 7 \cdot 10^6$. In scene Wall,

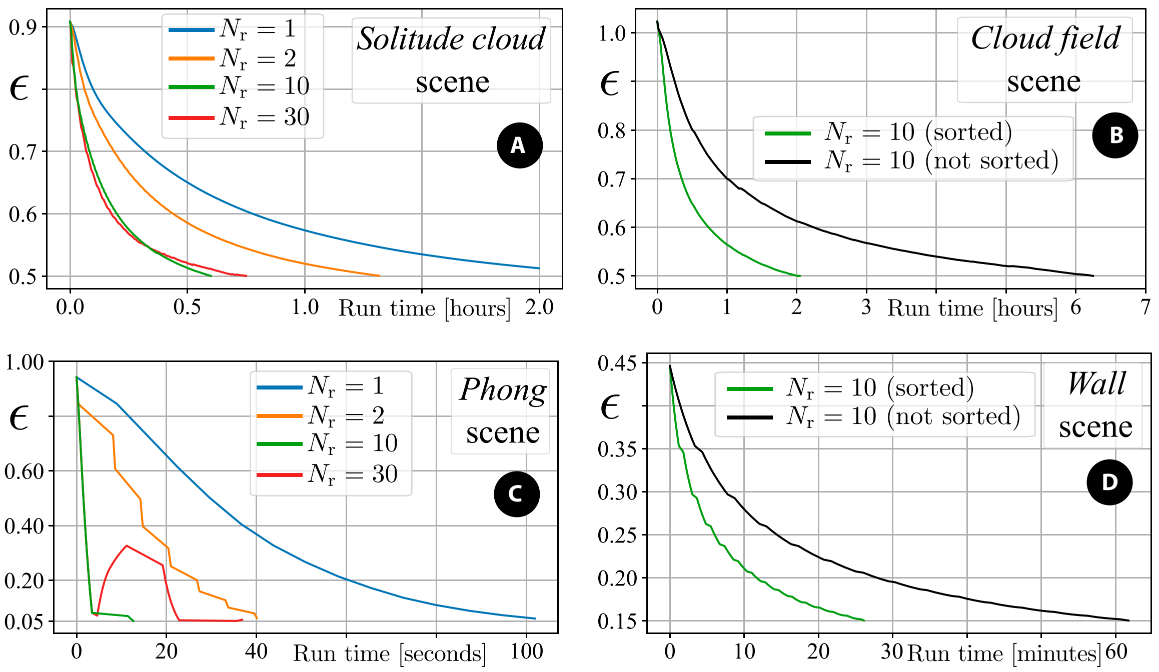

Fig. 8. ϵ versus run time of 4 synthetic scenes. (A and C) Comparison of different values of N_r (sorting is not used for $N_r = 1$). (B and D) Sorting versus no sorting, using $N_r = 10$.

Downloaded from https://spj.science.org at Technion Israel Institute of Technology on October 07, 2024

there is a diffuse ($\kappa_s = 0$) texture on the wall behind the camera. The wall domain $\mathcal{W}$ has 150×150 pixels. Corresponding to RGB color channels, the BRDF parameters $\left[\kappa_d^R, \kappa_d^G, \kappa_d^B\right]$ are each in the range [0,1]. Then, $\mathcal{M}_{\text{Wall}} = \left\{ \left[\kappa_d^R(\boldsymbol{x}), \kappa_d^G(\boldsymbol{x}), \kappa_d^B(\boldsymbol{x})\right] \right\}_{\boldsymbol{x} \in \mathcal{W}}$. The number of paths is $|\Phi| = 8 \cdot 10^7$.

The results of inverse rendering are displayed in Figs. 8 and 10. While these tests are simple, they indicate that PARS can also be used in reflectometry. An equivalent derivation of Eq. 20 for reflectometry is provided in section S4.2.

PARS speedup

Traditionally, with neither recycling nor sorting, run time of N_r iterations is

$$\mathcal{T}^{\text{no}}_{\text{recyc}} = N_r \left(\mathcal{T}_{\text{sample}} + \mathcal{T}_{\text{render}} + \mathcal{T}_{\text{grad}} \right) \tag{32}$$

Run time of N_r iterations of PARS takes

$$\mathcal{T}_{\text{PARS}} = \mathcal{T}_{\text{sample}} + \mathcal{T}_{\text{sort}} + \eta_{\text{sorted}} N_r \left(\mathcal{T}_{\text{render}} + \mathcal{T}_{\text{grad}} \right) \tag{33}$$

Consider, for example, the parameters of our system and the solitude cloud scene (see the Results section). Then, using $|\Phi| = 5 \cdot 10^7$ paths and ≈ 7.05 GB memory (at peak): $\mathcal{T}_{\text{sample}} = 5.52$ [s], $\mathcal{T}_{\text{sort}} = 6.44$ [s], $\eta_{\text{sorted}} \left(\mathcal{T}_{\text{render}} + \mathcal{T}_{\text{grad}} \right) = 1.38$ [s], and $\eta_{\text{sorted}} = 0.23$. The averaged iteration speedup is:

$$\text{speedup} = \mathcal{T}^{\text{no}}_{\text{recyc}} / \mathcal{T}_{\text{PARS}} \tag{34}$$

as plotted in Fig. 3. We use $N_r = 10$, so we expect speedup ≈ 4.5.

Discussion

In every research, there are goals and methods. The main goal of this paper is to perform fast, efficient, and scalable cloud tomography. The available methods depend on current technologies. For example, a common practice is to enhance efficiency by translating an algorithm to a parallelized GPU code. Therefore, we have made an effort to achieve our goal by suggesting an algorithm that is tailored to GPUs. The approach exploits the seed of a random number generator, for saving memory. In the future, other technologies may overtake the GPU architecture. Then, the balance between memory speed, computation, and other resources may change. In order to make the best use of those future technologies, our approach may need to be adjusted.

While this paper takes a step toward accurate and efficient cloud tomography, a gap remains. A study on cloud-climate feedback requires an accurate representation of the cloud microphysics. Cloud microphysics include the water droplet distribution parameters and should preferably include specifically the water droplet effective radius and water droplet effective variance. However, at this point, we represent a cloud voxel by its extinction coefficient. So, PARS should be extended to include microphysical parameters in each voxel. This will need to be addressed in future research in order to fully exploit the methods of this work for climate research.

Using paths from a previous iteration can increase the variance of rendered images (the forward model). Moreover, it increases the variance of the gradient estimator, as if fewer paths are sampled. Therefore, excessively extending N_r may yield diminishing returns (Fig. 8). Eq. 18 essentially seeks the derivative $\partial_{m_v} \mathcal{L}$ of the loss function; the derivative $\partial_{m_v} \mathcal{F}_d$ is a means to that end. For a true unbiased $\partial_{m_v} \mathcal{L}$, the paths for

Downloaded from https://spj.science.org at Technion Israel Institute of Technology on October 07, 2024

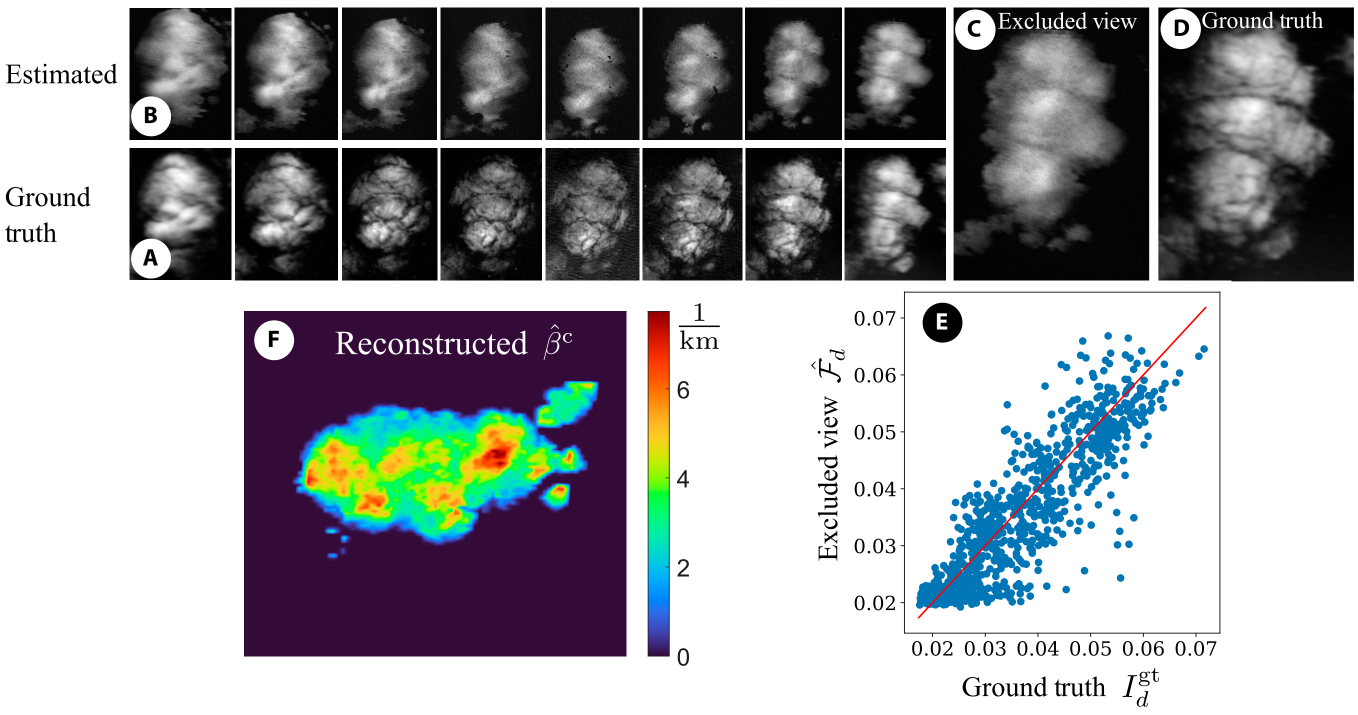

Fig. 9. AirMSPI real-world experiment. (A) Eight views that were used by the optimization. (B) Corresponding estimated images. (C) Excluded view is estimated in cross-validation. (D) Ground-truth image corresponding to (C). (E) Scatterplot of pixel values. (F) 3D cloud tomography using our method (rendered using MIP).

estimating $\mathcal{F}_d$ and $\partial_{m_v}\mathcal{F}_d$ should be exclusive. We use the same paths for both subtasks: It may cause a bias, but we found that in practice, this bias is insignificant in stochastic optimization. We cannot overrule the existence of cases where this bias has significance. Then, the path set used for $\mathcal{F}_d$ should not be the set used for $\partial_{m_v}\mathcal{F}_d$.

Inverse rendering is not only interesting for computer vision and graphics. It has benefits for wider society, specifically regarding climate and human health. Clouds have a strong effect on climate and bear a great source of climate uncertainty. 3D scattering CT of clouds (an inverse rendering problem) offers a new tool for tackling this uncertainty [61]. However, so far, solutions have been too slow to scale to meet scientific needs. Hence, it is critical to develop efficient methods to scale inverse rendering to huge atmospheric scenes.

PARS is not limited to cloud tomography and reflectometry. An additional inverse rendering task is x-ray CT through scatter [21] (illustrated in Fig. 11). Geva et al. [33] showed that accounting for scatter can significantly reduce dose while indicating the chemistry of organs, using MC for scattering CT. As this method relies on iterating MC random sampling of paths, it can benefit from our approach as well.

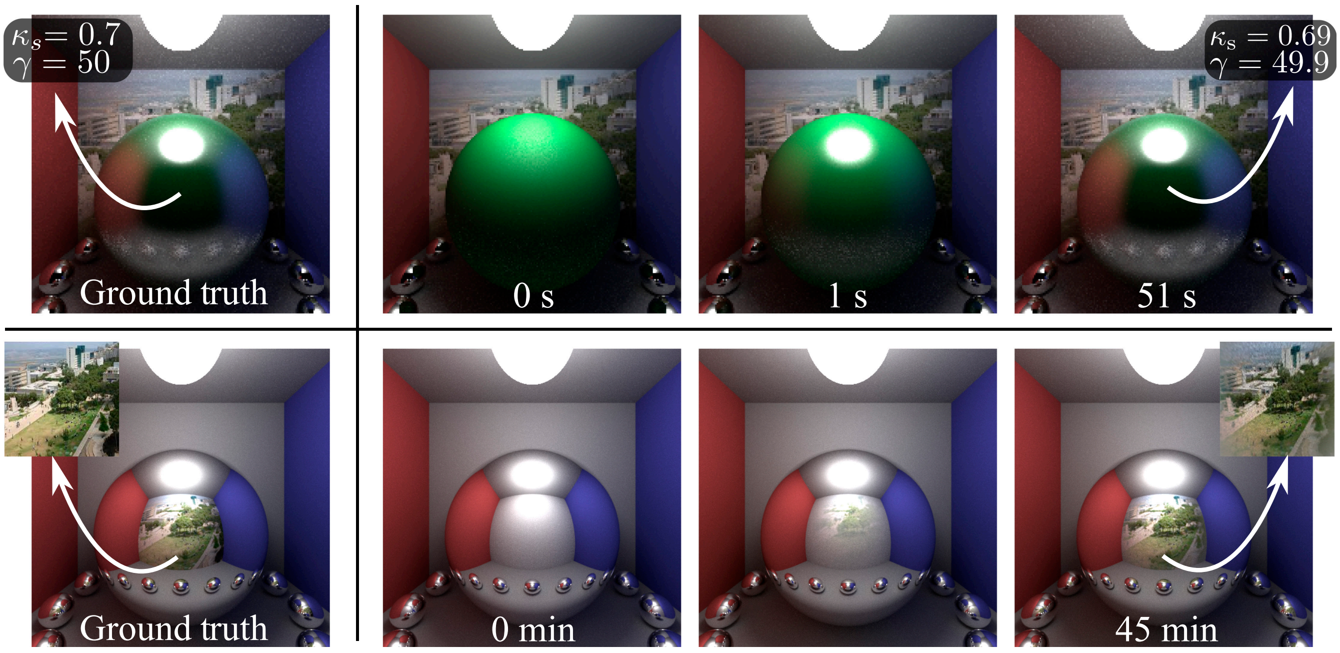

Fig. 10. BRDF recovery of scene Phong (top) and scene Wall (bottom).

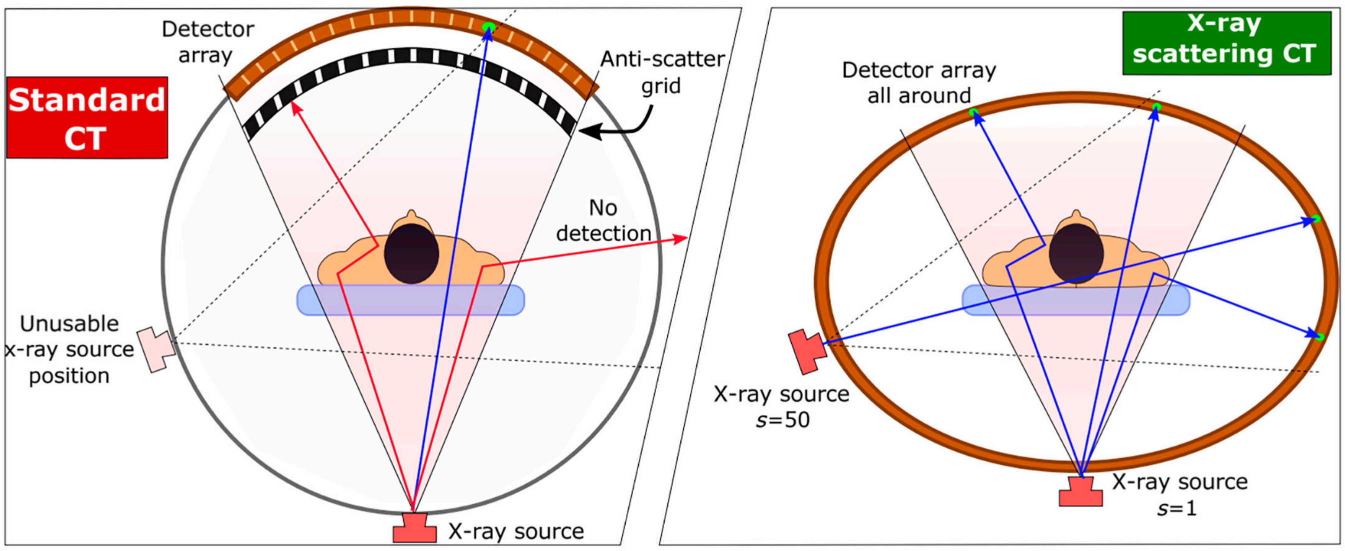

Fig. 11. In standard CT (left), an anti-scatter grid near the detectors blocks the majority of photons scattered by the body (red) and many nonscattered photons. An anti-scatter grid suits only one projection, necessitating rigid rotation of the anti-scatter grid with the source. Removing the anti-scatter grid (right) enables simultaneous multisource irradiation and allows all photons passing through the body to reach the detector. Courtesy of A. Geva [33].

Downloaded from https://spj.science.org at Technion Israel Institute of Technology on October 07, 2024

Acknowledgments

We thank V. Holodovsky and R. Ronen for useful discussions. We thank J. Erez, I. Talmon, and D. Yagodin for technical support. **Funding:** Y.Y.S. is the Mark and Diane Seiden Chair in Science at the Technion. He is a Landau Fellow, supported by the Taub Foundation. His work was conducted in the Ollendorff Minerva Center. Minvera is funded through the BMBF. This research is funded by the ERC under the European Union's Horizon 2020 research and innovation program (grant agreement no. 810370: CloudCT). **Author contributions:** I.C. conceived the idea, formulated the theory, and implemented the rendering engine. Y.Y.S. provided supervisory support and guidance. **Competing interests:** The authors declare that they have no competing interests.

Data Availability

The code of this work is available at https://github.com/idocz/PARS.

Supplementary Materials

Fig. S1. Riemann sum.
Fig. S2. MC integration.
Fig. S3. Importance sampling.
Fig. S4. Illustration of definitions.
Fig. S5. Illustration of $f_d(\boldsymbol{X})$ for $\boldsymbol{X} = \{\boldsymbol{x}_{\text{light}}, \boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_d\}$.
Fig. S6. Illustration of $\mathcal{V}_v$ and $l_{b,v}$ on a path.
Fig. S7. Visual comparisons with [38].
Fig. S8. ϵ versus run time of 4 synthetic scenes.

References

1. Aides A, Levis A, Holodovsky V, Schechner YY, Althausen D, Vainiger A. Distributed sky imaging radiometry and tomography. Paper presented at: International Conference on Computational Photography, ICCP, IEEE; 2020 Apr 24–26; St. Louis, MO. p. 1–12.
2. Levis A, Schechner YY, Aides A, Davis AB. Airborne three-dimensional cloud tomography. Paper presented at: International Conference on Computer Vision, ICCV, IEEE; 2015 Dec 7–13, Santiago, Chile. p. 3379–3387.
3. Levis A, Schechner YY, Davis AB. Multiple-scattering microphysics tomography. Paper presented at: Conference on Computer Vision and Pattern Recognition, CVPR, IEEE; 2017 July 21–26; Honolulu, HI. p. 6740–6749.
4. Ronen R, Schechner YY, Eytan E. 4D cloud scattering tomography. Paper presented at: International Conference on Computer Vision, ICCV, IEEE; 2021 Oct 10–17; Montreal, QC. p. 5520–5529.
5. Sde-Chen Y, Schechner YY, Holodovsky V, Eytan E. 3DeepCT: Learning volumetric scattering tomography of clouds. Paper presented at: International Conference on Computer Vision, ICCV, IEEE; 2021 Oct 10–17; Montreal, QC. p. 5671–5682.
6. Camps-Valls G, Tuia D, Zhu XX, Reichstein M. *Deep learning for the Earth sciences: A comprehensive approach to remote sensing, climate science and geosciences*. Hoboken, New Jersey: John Wiley & Sons; 2021.
7. Davis C, Emde C, Harwood R. A 3-D polarized reversed Monte Carlo radiative transfer model for millimeter and submillimeter passive remote sensing in cloudy atmospheres. *IEEE, Trans Geosci Remote Sens*. 2005;43(5):1096–1101.
8. Yang J, Gong P, Fu R, Zhang M, Chen J, Liang S, Xu B, Shi J, Dickinson R. The role of satellite remote sensing in climate change studies. *Nat Clim Chang*. 2013;3(10):875–883.
9. Yuan Q, Shen H, Li T, Li Z, Li S, Jiang Y, Xu H, Tan W, Yang Q, Wang J, et al. Deep learning in environmental remote sensing: Achievements and challenges. *Remote Sens Environ*. 2020;241:111–716.
10. Forster L, Mayer B. Ice crystal characterization in cirrus clouds III: Retrieval of ice crystal shape and roughness from observations of halo displays. *Atmos Chem Phys Discuss*. 2022;1–38.
11. Nataraja V, Schmidt S, Chen H, Yamaguchi T, Kazil J, Feingold G, Wolf K, Iwabuchi H. Segmentation-based multipixel cloud optical thickness retrieval using a convolutional neural network. *Atmos Meas Tech Discuss*. 2022;1–34.
12. Ramon D, Steinmetz F, Jolivet D, Compiègne M, Frouin R. Modeling polarized radiative transfer in the ocean-atmosphere system with the GPU-accelerated SMART-G Monte Carlo code. *J Quant Spectrosc Radiat Transf*. 2019;222:89–107.
13. Bolibar J, Rabatel A, Gouttevin I, Zekollari H, Galiez C. Nonlinear sensitivity of glacier mass balance to future climate change unveiled by deep learning. *Nat Commun*. 2022;13(1):409.
14. Davenport FV, Diffenbaugh NS. Using machine learning to analyze physical causes of climate change: A case study of U.S. Midwest extreme precipitation. *Geophys Res Lett*. 2021;48(15):e2021GL093787.
15. Lenaerts JT, Gettelman A, Van Tricht K, van Kampenhout L, Miller NB. Impact of cloud physics on the Greenland ice sheet near-surface climate: A study with the community atmosphere model. *J Geophys Res Atmos*. 2020;125(7):e2019JD031470.
16. Zhou X, Zhang J, Feingold G. On the importance of sea surface temperature for aerosol-induced brightening of marine clouds and implications for cloud feedback in a future warmer climate. *Geophys Res Lett*. 2021;48(24):e2021GL095896.
17. Fujita TT. Mesoscale classifications: Their history and their application to forecasting. In: *Mesoscale meteorology and forecasting*. 1986, p. 18–35.
18. Bony S, Stevens B, Frierson DM, Jakob C, Kageyama M, Pincus R, Shepherd TG, Sherwood SC, Siebesma AP, Sobel AH, et al. Clouds, circulation and climate sensitivity. *Nat Geosci*. 2015;8(4):261–268.
19. O. Boucher, D. Randall, P. Artaxo, C. Bretherton, G. Feingold, P. Forster, V.-M. Kerminen, Y. Kondo, H. Liao, U. Lohmann, et al. Climate change 2013: The physical science basis. Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change. Cambridge, England: Cambridge University Press; 2013. p. 571–657.
20. Ceppi P, Brient F, Zelinka MD, Hartmann DL. Cloud feedback mechanisms and their representation in global climate models. *Wiley Interdiscip Rev Clim Chang*. 2017;8(4):e465.
21. Várnai T, Marshak A. Statistical analysis of the uncertainties in cloud optical depth retrievals caused by three-dimensional radiative effects. *J Atmos Sci*. 2001;58(12):1540–1548.
22. Fascione L, Hanika J, Leone M, Droske M, Schwarzhaupt J, Davidovič T, Weidlich A, Meng J. Manuka: A batch-shading architecture for spectral path tracing in movie production. *ACM Trans Graph*. 2018;37(3):1–18.
23. Laine S, Karras T, Aila T. Megakernels considered harmful: Wavefront path tracing on GPUs. Paper presented at:

Downloaded from https://spj.science.org at Technion Israel Institute of Technology on October 07, 2024

Proceedings of High Performance Graphics; 2013; Anaheim, CA. p. 137–143.
24. Lee M, Green B, Xie F, Tabellion E. Vectorized production path tracing. Paper presented at: Proceedings of High Performance Graphics, ACM; 2017; Los Angeles, CA. pp. 1–11.
25. Nimier-David M, Speierer S, Ruiz B, Jakob W. Radiative backpropagation: An adjoint method for lightning-fast differentiable rendering. *ACM Trans Graph*. 2020;39(4):146:1–146:15.
26. Nimier-David M, Vicini D, Zeltner T, Jakob W. Mitsuba 2: A retargetable forward and inverse renderer. *ACM Trans Graph*. 2019;38(6):1–17.
27. Kajiya JT. The rendering equation. In *Computer graphics*. New York, NY, USA: Association for Computing Machinery; 1986. p. 143–150.
28. Veach E. Robust Monte Carlo methods for light transport simulation [dissertation]. Stanford University; 1997. vol. 1610.
29. Romeiro F, Zickler T. Paper presented at: Blind reflectometry. In: *ECCV*. Berlin, Heidelberg: Springer; 2010. p. 45–58.
30. Shem-Tov K, Bangaru SP, Levin A, Gkioulekas I. Towards reflectometry from interreflections. Paper presented at: International Conference on Computational Photography, ICCP, IEEE; 2020; Saint Louis, MO. p. 1–12.
31. Alterman M, Bar C, Gkioulekas I, Levin A. Imaging with local speckle intensity correlations: Theory and practice. *ACM Trans Graph*. 2021;40(3):1–22.
32. Bar C, Alterman M, Gkioulekas L, Levin A. Single scattering modeling of speckle correlation. Paper presented at: International Conference on Computational Photography, ICCP, IEEE; 2021 May 23–25; Haifa, Israel. p. 1–16.
33. Geva A, Schechner YY, Chernyak Y, Gupta R. Paper presented at: X-ray computed tomography through scatter. In: *ECCV*. Springer; 2018. p. 34–50.
34. Vicini D, Speierer S, Jakob W. Path replay backpropagation: Differentiating light paths using constant memory and linear time. *ACM Trans Graph*. 2021;40(4):1–14.
35. Gkioulekas I, Levin A, Zickler T. An evaluation of computational imaging techniques for heterogeneous inverse scattering. In: *ECCV*. Springer; 2016. p. 685–701.
36. Gkioulekas I, Zhao S, Bala K, Zickler T, Levin A. Inverse volume rendering with material dictionaries. *ACM Trans Graph*. 2013;32(6):162.
37. Khungurn P, Schroeder D, Zhao S, Bala K, Marschner S. Matching real fabrics with micro-appearance models. *ACM Trans Graph*. 2015;35(1):1–1.
38. Loeub T, Levis A, Holodovsky V, Schechner YY. Monotonicity prior for cloud tomography. Paper presented at: Proceedings of the European Conference on Computer Vision, ECCV; 2020 Aug 24–29; Glasgow, Scotland.
39. Levis A, Aides A. Pyshdom. 2019. https://github.com/aviadlevis/pyshdom
40. Zhang C, Miller B, Yan K, Gkioulekas I, Zhao S. Path-space differentiable rendering. *ACM Trans Graph*. 2020;39(4).
41. Jakob W, Speierer S, Roussel N Nimier-David M, Vicini D, Zeltner T, Nicolet B, Crespo M, Leroy V, Zhang Z, Mitsuba 3 renderer. version 3.1.1 (2022); https://mitsuba-renderer.org
42. Bekaert P, Sbert M, Halton J. Accelerating path tracing by re-using paths. In: *Eurographics workshop on rendering*. Debevec P, Gibson S, editors. Pisa, Italy: The Eurographics Association; 2002.
43. Bauszat P, Petitjean V, Eisemann E. Gradient-domain path reusing. *ACM Trans Graph*. 2017;36(6):229.
44. Hašan M, Ramamoorthi R. Interactive albedo editing in path-traced volumetric materials. *ACM Trans Graph*. 2013;32(11):11.
45. Áfra AT, Benthin C, Wald I, Munkberg J. Local shading coherence extraction for SIMD-efficient path tracing on CPUs. In: *High performance graphics*. Eurographics Association; 2016. p. 119–128.
46. Garanzha K, Loop C. Fast ray sorting and breadth-first packet traversal for GPU ray tracing. *Comput Graph Forum*. 2010;29(2):289–298.
47. Novák J, Georgiev I, Hanika J, Jarosz W. Monte Carlo methods for volumetric light transport simulation. In: *Computer graphics forum*. Wiley; 2018. vol. 37. p. 551–576.
48. Marshak A, Davis A. *3D radiative transfer in cloudy atmospheres*. Springer; 2005.
49. Zeltner T, Speierer S, Georgiev I, Jakob W. Monte Carlo estimators for differential light transport. *ACM Trans Graph*. 2021;40(4):1–16.
50. Binzoni T, Leung TS, Gandjbakhche AH, Ruefenacht D, Delpy D. The use of the Henyey–Greenstein phase function in Monte Carlo simulations in biomedical optics. *Phys Med Biol*. 2006;51(17):N313.
51. Lam SK, Pitrou A, Seibert S. Numba: A LLVM-based python JIT compiler. Paper presented at: Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC, ACM, 2015; Austin, TX. p. 1–6.
52. Ruder S. An overview of gradient descent optimization algorithms. arXiv. 2016. https://arxiv.org/abs/1609.04747
53. Loeub T, MitsubaGradient. 2019. https://github.com/tamarloeub/MitsubaGradient
54. Onasch TB, Massoli P, Kebabian PL, Hills FB, Bacon FW, Freedman A. Single scattering albedo monitor for airborne particulates. *Aerosol Sci Technol*. 2015;49(4):267–279.
55. Frisvad JR. Importance sampling the Rayleigh phase function. *J Opt Soc Am A Opt Image Sci Vis*. 2011;28(12):2436–2441.
56. Kutulakos KN, Seitz SM. A theory of shape by space carving. *Int J Comput Vis*. 2000;38(3):199–218.
57. Diner DJ, Xu F, Garay MJ, Martonchik JV, Rheingans BE, Geier S, Davis AB, Hancock B, Jovanovic VM, Bull MA, et al. The Airborne Multiangle SpectroPolarimetric Imager (AirMSPI): A new tool for aerosol and cloud remote sensing. *Atmos Meas Tech*. 2013;6(8):2007.
58. Ngan A, Durand F, Matusik W. Experimental analysis of BRDF models. *Render Tech*. 2005;2005(16):2.
59. Phong BT. Illumination for computer generated pictures. *Commun ACM*. 1975;18(6):311–317.
60. K. Zsolnai-Feh'er, Smallpaint: A global illumination renderer. 2018. https://users.cg.tuwien.ac.at/zsolnai/gfx/smallpaint/
61. Schilling K, Schechner YY, Koren I. CloudCT—Computed tomography of clouds by a small satellite formation. Paper presented at: IAA Symposium on Small Satellites for Earth Observation; 2019; TU Berlin.

Downloaded from https://spj.science.org at Technion Israel Institute of Technology on October 07, 2024

PARS - Path Recycling and Sorting for Efficient Cloud Tomography Supplementary Material

Ido Czerninski and Yoav Y. Schechner

Viterbi Faculty of Electrical and Computer Engineering,
Technion - Israel Inst. of Technology, Haifa, Israel

Abstract

This is a supplementary document to the main manuscript. This document provides a detailed theoretical derivation of *Path Recycling and Sorting* (PARS) in an inverse rendering framework. We start with basic Monte-Carlo (MC) integration practice and then show how it can be used to derive PARS. Additionally, we provide more simulated results.

1 Outline

This supplementary document contains five sections. Sec. 2 starts with basic 1D examples of Monte-Carlo (MC) integration. Building on Sec. 2, Sec. 3 first provides the path PDF μ_d that corresponds to both scattering and surface MC light transport. Then, Sec. 3 uses basic MC concepts to derive PARS correction factors for both problems. Sec. 4 provides a detailed derivation of the gradient of the forward model for both scattering tomography and reflectometry. Sec. 5 provides additional results.

2 Monte-Carlo Integration

This section gives several examples that illustrate MC integration. In the following examples, we consider a one-dimensional function. This is a basis for generalization to arbitrary dimensions [1].

Let $f : [0, 1] \to \mathbb{R}$ be a one dimensional function. We are interested in calculating the following integral:

$$I = \int_0^1 f(u)du. \tag{1}$$

Before suggesting an MC technique, one can trivially approximate the integral in equation Eq. (1) herein by deterministically dividing the segment $[0, 1]$ to N partitions of the same width $\frac{1}{N}$, as

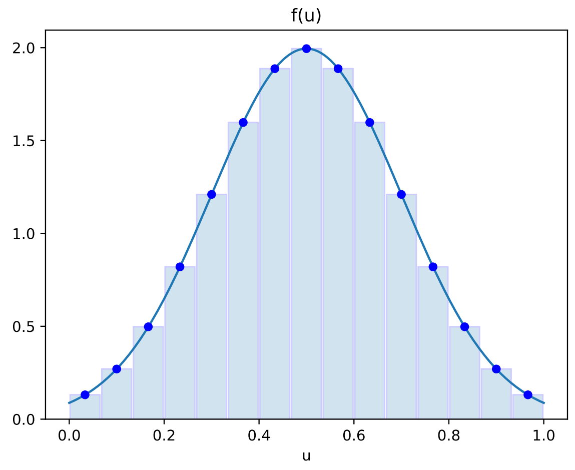

Figure 1: Riemann sum. Approximating the integral of the function $f(u)$ (dark blue) by dividing the segment $[0, 1]$ to 15 rectangles (light blue).

illustrated in Fig. 1 below. Let A_i be the area of a rectangle whose width is the i'th partition, and whose height is set by the corresponding value of f. Then, the integral can be approximated by:

$$I \approx \sum_{i=1}^{N} A_i. \tag{2}$$

The limit $N \to \infty$ yields the integral definition as a Riemann sum. A Riemann sum converges fast for 1-dimensional integrals, but it suffers from a slow convergence rate for higher dimensions. This is where MC shines.

2.1 Uniform Sampling

MC integration estimates Eq. (1) above by exploiting the similarity between a deterministic integral and the definition of random variable expectation. Consider N independent and identically distributed (IID) random samples. They are drawn from a uniform distribution over the segment $[0, 1]$, that is,

$$\{u_i\}_{i=1}^{N} \sim \mathcal{U}[0, 1]. \tag{3}$$

The corresponding probability density function (PDF) is

$$p_{\mathcal{U}}(u) = \begin{cases} 1 & \text{if } 0 \leq u \leq 1 \\ 0 & \text{else} \end{cases}. \tag{4}$$

MC suggests [1] the following unbiased estimator, illustrated in Fig. 2 herein:

$$\hat{I} = \frac{1}{N} \sum_{i=1}^{N} f(u_i). \tag{5}$$

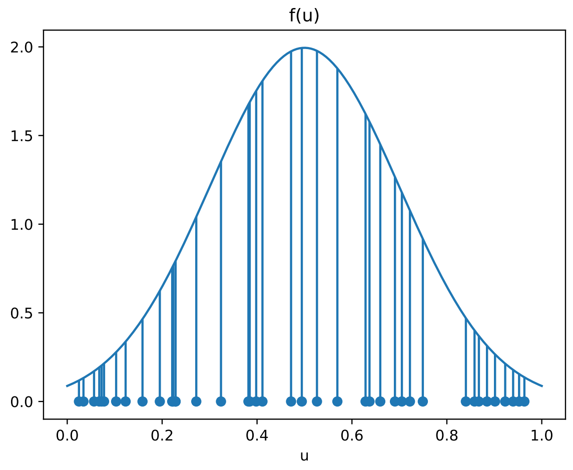

Figure 2: MC integration. Estimating the integral of the function $f(u)$ using uniform sampling.

This estimator can be analogously compared to Eq. (2) above, but with rectangles at random locations. Denote $\mathbb{E}[\cdot]$ as the expected value operator. MC relies on the law of large numbers from probability theory, which states that:

$$\lim_{N\to\infty} \frac{1}{N}\sum_{i=1}^{N} f(u_i) = \mathbb{E}\left[f(u)\right]. \tag{6}$$

By plugging the expectation definition we achieve convergence

$$\mathbb{E}\left[f(u)\right] = \int\limits_{-\infty}^{\infty} f(u)\cdot p_{\mathcal{U}}(u)du = \int\limits_{0}^{1} f(u)du = I. \tag{7}$$

The mean square error (MSE) for an unbiased estimator is the estimated variance:

$$\mathrm{MSE}(\hat{I}) = \mathbb{E}\left[(I-\hat{I})^2\right] = \mathbb{E}\left[(\hat{I}-\mathbb{E}[\hat{I}])^2\right] + \underbrace{(\mathbb{E}[\hat{I}]-I)^2}_{\text{unbiased}\to=0} = \mathrm{Var}(\hat{I}). \tag{8}$$

This variance decreases with N.

2.2 Importance Sampling

Uniform sampling is an acceptable choice for well-behaved functions. However, in cases where the function f has a range including both extremely low and high values, $\mathrm{Var}(\hat{I})$ decreases with N significantly more slowly, when Eq. (5) is used. This is seen in Fig. 3 herein. This problem can be mitigated by random sampling from a different PDF, which considers the shape of f.

Let μ be a general PDF. Let $\{u_i\}_{i=1}^{N} \sim \mu$ be N IID random samples drawn from μ. MC

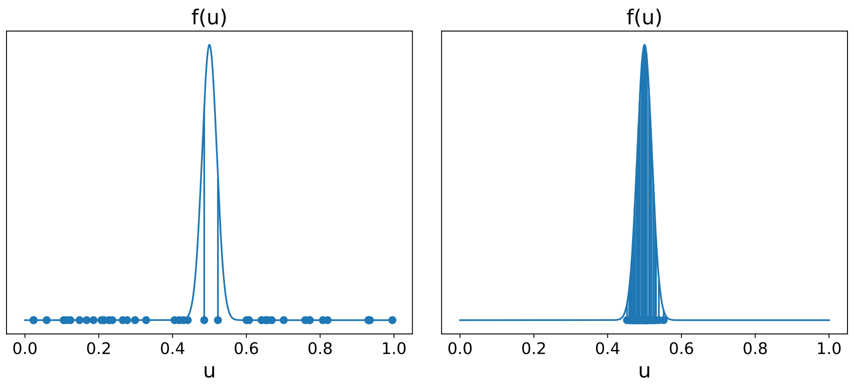

Figure 3: Importance sampling. Illustration of a function that is not suitable for uniform sampling. [Top] Uniform sampling misses the important features of $f(u)$. [Bottom] Importance sampling samples the random variable from a distribution μ that is similar to f.

suggests [1] the following estimator:

$$\hat{I} = \frac{1}{N}\sum_{i=1}^{N}\frac{f(u_i)}{\mu(u_i)}. \tag{9}$$

This is a generalization of Eq. (5) herein. Eq. (9) above uses an arbitrary PDF μ to sample the random variables. The division by $\mu(u_i)$ is a correction factor, considering the fact that some samples have a higher probability to be sampled than other samples. If μ yields a high probability of drawing a sample, the correction factor reduces the contribution of this particular sample (and vice versa).

Notice that the estimator in Eq. (9) above is unbiased, since

$$\begin{aligned}\mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N}\frac{f(u_i)}{\mu(u_i)}\right] &= \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}\left[\frac{f(u_i)}{\mu(u_i)}\right] \\ &= \frac{1}{N}\sum_{i=1}^{N}\int_0^1 \frac{f(u)}{\cancel{\mu(u)}}\cdot\cancel{\mu(u)}du \\ &= \frac{1}{N}\sum_{i=1}^{N} I = I.\end{aligned} \tag{10}$$

Eq. (10) above proves that any choice of μ (as long it is positive where f is positive) results in an unbiased estimator of I; however, we are interested in minimizing $\mathrm{Var}(\hat{I})$. It can be shown [1] that choosing μ to be proportional to f reduces $\mathrm{Var}(\hat{I})$. In other words, it is often desirable that in high probability μ draws samples that attain high values of f. This procedure is called *Importance Sampling.*

This background is related to Sec. 2.2 of the main manuscript. There, we state that we have the freedom to choose a path sampling PDF, μ_d. However, now, we mention that to maintain low $\mathrm{Var}(\hat{I})$, μ_d has to be “similar” enough to f_d. In a high dimensional integral, such as the path integral, this resemblance is even more important to maintain.

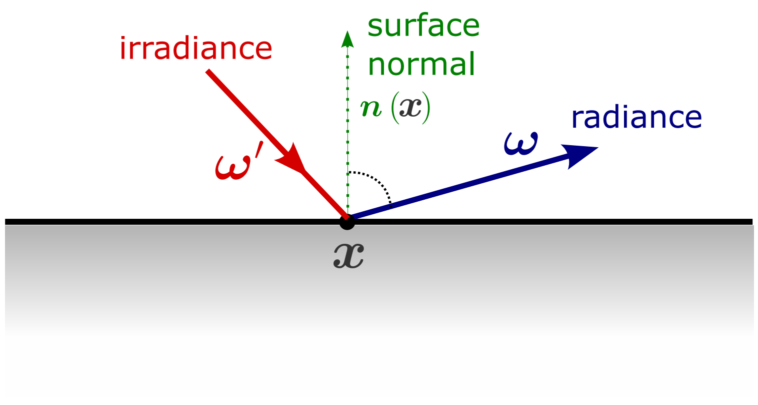

Figure 4: Illustration of definitions. herein. In the graphics literature, $\boldsymbol{\omega}'$ is often defined as the negative irradiance direction. We define $\boldsymbol{\omega}'$ as the irradiance direction, to be consistent with definitions in the literature on scattering.

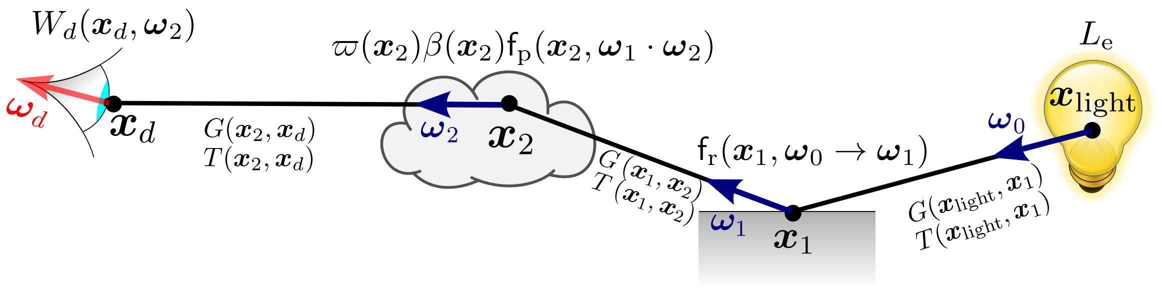
$W_d(\boldsymbol{x}_d, \boldsymbol{\omega}_2)$
$\varpi(\boldsymbol{x}_2)\beta(\boldsymbol{x}_2)\mathsf{f}_\mathrm{p}(\boldsymbol{x}_2, \boldsymbol{\omega}_1 \cdot \boldsymbol{\omega}_2)$
L_e
$\boldsymbol{x}_\mathrm{light}$
$\boldsymbol{\omega}_d$
$\boldsymbol{x}_d$
$G(\boldsymbol{x}_2, \boldsymbol{x}_d)$
$T(\boldsymbol{x}_2, \boldsymbol{x}_d)$
$\boldsymbol{\omega}_2$
$\boldsymbol{x}_2$
$\boldsymbol{\omega}_0$
$\mathsf{f}_\mathrm{r}(\boldsymbol{x}_1, \boldsymbol{\omega}_0 \to \boldsymbol{\omega}_1)$
$G(\boldsymbol{x}_1, \boldsymbol{x}_2)$
$T(\boldsymbol{x}_1, \boldsymbol{x}_2)$
$\boldsymbol{\omega}_1$
$\boldsymbol{x}_1$
$G(\boldsymbol{x}_\mathrm{light}, \boldsymbol{x}_1)$
$T(\boldsymbol{x}_\mathrm{light}, \boldsymbol{x}_1)$

Figure 5: Illustration of $f_d(\boldsymbol{X})$ for $\boldsymbol{X} = \{\boldsymbol{x}_\mathrm{light}, \boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_d\}$. Here, $\boldsymbol{\omega}_b$ is a vector direction from $\boldsymbol{x}_{b-1}$ towards $\boldsymbol{x}_b$. The first location $\boldsymbol{x}_0 = \boldsymbol{x}_\mathrm{light}$ is the light source location, which contributes a factor of L_e to f_d. The path propagation to $\boldsymbol{x}_1$ contributes a factor of $G(\boldsymbol{x}_\mathrm{light}, \boldsymbol{x}_1) T(\boldsymbol{x}_\mathrm{light}, \boldsymbol{x}_1)$ to f_d. At $\boldsymbol{x}_1$ the surface reflects the path towards $\boldsymbol{\omega}_1$. This event contributes a factor of $\mathsf{f}_\mathrm{r}(\boldsymbol{x}_1, \boldsymbol{\omega}_0 \to \boldsymbol{\omega}_1)$ to f_d. Then, at $\boldsymbol{x}_2$, the path is scattered by a medium towards $\boldsymbol{\omega}_2$. This event contributes a factor of $\varpi(\boldsymbol{x}_2)\beta(\boldsymbol{x}_2)\mathsf{f}_\mathrm{p}(\boldsymbol{x}_2, \boldsymbol{\omega}_1 \cdot \boldsymbol{\omega}_2)$ to f_d. Finally, the path hits a detector and f_d gains a factor of $W_d(\boldsymbol{x}_d, \boldsymbol{\omega}_2)$.

3 Derivation of PDFs in MC Light Transport

This section shows a derivation of the PDF μ_d that corresponds to the sampling process from Sec. 2.2 [2] of the main manuscript. Moreover, this section describes basic MC light transport for surfaces-only setting and provides the corresponding μ_d. In addition, this section derives the correction factors for both scattering tomography and reflectometry.

3.1 Definitions

First, we recall definitions from the main manuscript. A surface at 3D location $\boldsymbol{x}$ has normal $\mathbf{n}(\boldsymbol{x})$ and BRDF $\mathsf{f}_\mathrm{r}(\boldsymbol{x}, \boldsymbol{\omega}' \to \boldsymbol{\omega})$. Here $\boldsymbol{\omega}'$ is the 3D direction vector of incoming irradiance, while radiance is reflected to $\boldsymbol{\omega}$ (illustrated in Fig. 4 herein).

In this paper, we deal with a variation of a Phong BRDF with no absorption. Let $\boldsymbol{\omega}^\mathrm{s}$ be the specular direction of $\boldsymbol{\omega}'$ with respect to the normal $\mathbf{n}(\boldsymbol{x})$, thus:

$$\boldsymbol{\omega}^{\mathrm{s}} = \boldsymbol{\omega}' - 2 \cdot [\mathbf{n}(\boldsymbol{x}) \cdot \boldsymbol{\omega}'] \, \mathbf{n}(\boldsymbol{x}) \,. \tag{11}$$

Then, the BRDF is:

$$\mathsf{f}_{\mathrm{r}}(\boldsymbol{x}, \boldsymbol{\omega}' \to \boldsymbol{\omega}) = \frac{\kappa_{\mathrm{d}}}{\pi} + \kappa_{\mathrm{s}} \frac{\gamma + 2}{2\pi} (\boldsymbol{\omega}^{\mathrm{s}} \cdot \boldsymbol{\omega})^{\gamma}. \tag{12}$$

In this notation $0 \leq \kappa_{\mathrm{d}} \leq 1$ is diffuse albedo, $0 \leq \kappa_{\mathrm{s}} \leq 1$ controls the specular part of the BRDF and $\gamma \geq 0$ controls its shininess.

Multiple scattering particle types exist in a medium. Each has an extinction coefficient $\beta^{\mathrm{type}}(\boldsymbol{x})$, a single scattering albedo ϖ^{type}, and a scattering *phase function* $\mathsf{f}_{\mathrm{p}}^{\mathrm{type}}(\boldsymbol{x}, \boldsymbol{\omega}' \to \boldsymbol{\omega}) = \mathsf{f}_{\mathrm{p}}^{\mathrm{type}}(\boldsymbol{x}, \boldsymbol{\omega}' \cdot \boldsymbol{\omega})$. In the ensemble of particles, the total extinction coefficient, effective single scattering albedo, and total phase function are respectively

$$\beta(\boldsymbol{x}) = \sum_{\mathrm{type}} \beta^{\mathrm{type}}(\boldsymbol{x}) \,, \tag{13}$$

$$\varpi(\boldsymbol{x}) \equiv \frac{\sum_{\mathrm{type}} \varpi^{\mathrm{type}} \beta^{\mathrm{type}}(\boldsymbol{x})}{\beta(\boldsymbol{x})}, \tag{14}$$

$$\mathsf{f}_{\mathrm{p}}(\boldsymbol{x}, \boldsymbol{\omega}' \cdot \boldsymbol{\omega}) = \frac{\sum_{\mathrm{type}} \varpi^{\mathrm{type}} \beta^{\mathrm{type}}(\boldsymbol{x}) \mathsf{f}_{\mathrm{p}}^{\mathrm{type}}(\boldsymbol{x}, \boldsymbol{\omega}' \cdot \boldsymbol{\omega})}{\varpi(\boldsymbol{x}) \, \beta(\boldsymbol{x})}. \tag{15}$$

Detector d has a response $W_d(\boldsymbol{x}, \boldsymbol{\omega})$ to radiance propagating towards $\boldsymbol{\omega}$ (around $\boldsymbol{\omega}_d$) at $\boldsymbol{x} = \boldsymbol{x}_d$ (see Fig. 5 above). A point light source at $\boldsymbol{x}_{\mathrm{light}}$ emits radiance L_{e}. Let $\boldsymbol{X} = (\boldsymbol{x}_0, \ldots \boldsymbol{x}_b \ldots, \boldsymbol{x}_B)$ be a path of an arbitrary size $B + 1 = |\boldsymbol{X}|$. Here b represents an interaction index. The direction vector from $\boldsymbol{x}_b$ to $\boldsymbol{x}_{b+1}$ is $\boldsymbol{\omega}_b$. The transmittance between $\boldsymbol{x}_{b-1}$ and $\boldsymbol{x}_b$ is

$$T(\boldsymbol{x}_{b-1}, \boldsymbol{x}_b) \equiv \exp\left(- \int_{\boldsymbol{x}_{b-1}}^{\boldsymbol{x}_b} \beta(\boldsymbol{x}) \, d\boldsymbol{x} \right). \tag{16}$$

Let $\boldsymbol{\omega}_{\boldsymbol{x} \to \boldsymbol{y}}$ be a vector direction between $\boldsymbol{x}$ and $\boldsymbol{y}$, and

$$D(\boldsymbol{x}, \boldsymbol{y}) = \begin{cases} |\mathbf{n}(\boldsymbol{x}) \cdot \boldsymbol{\omega}_{\boldsymbol{x} \to \boldsymbol{y}}| & \text{if } \boldsymbol{x} \in \text{surface} \\ 1 & \text{if } \boldsymbol{x} \in \text{medium} \end{cases}. \tag{17}$$

Define respectively a geometric term and a directional interaction function, by

$$G(\boldsymbol{x}_{b-1}, \boldsymbol{x}_b) \equiv \frac{D(\boldsymbol{x}_{b-1}, \boldsymbol{x}_b) D(\boldsymbol{x}_b, \boldsymbol{x}_{b-1})}{||\boldsymbol{x}_b - \boldsymbol{x}_{b-1}||^2} \tag{18}$$

$$\mathsf{f}_{\mathrm{s}}(\boldsymbol{x}_{b-1}, \boldsymbol{x}_b, \boldsymbol{x}_{b+1}) \equiv \begin{cases} \mathsf{f}_{\mathrm{r}}(\boldsymbol{x}_b, \boldsymbol{\omega}_{b-1} \to \boldsymbol{\omega}_b) & \text{if } \boldsymbol{x}_b \in \text{surface} \\ \varpi \beta(\boldsymbol{x}_b) \, \mathsf{f}_{\mathrm{p}}(\boldsymbol{x}_b, \boldsymbol{\omega}_{b-1} \cdot \boldsymbol{\omega}_b) & \text{if } \boldsymbol{x}_b \in \text{medium} \end{cases}. \tag{19}$$

Using Eqs. (16,18,19) herein, define $\forall b \in \{1, ..., B-1\}$:

$$g_b\left(\boldsymbol{x}_{b-1}, \boldsymbol{x}_b, \boldsymbol{x}_{b+1}\right) = \mathsf{f}_{\mathrm{s}}\left(\boldsymbol{x}_{b-1}, \boldsymbol{x}_b, \boldsymbol{x}_{b+1}\right) T(\boldsymbol{x}_{b-1}, \boldsymbol{x}_b) G\left(\boldsymbol{x}_{b-1}, \boldsymbol{x}_b\right) \tag{20}$$

and

$$g_B\left(\boldsymbol{x}_{B-1}, \boldsymbol{x}_B, \boldsymbol{x}_{B+1}\right) = T(\boldsymbol{x}_{B-1}, \boldsymbol{x}_B) G\left(\boldsymbol{x}_{B-1}, \boldsymbol{x}_B\right). \tag{21}$$

Let $\mathcal{M} = \{m_v\}_{v=1}^{N_{\mathrm{v}}}$ be a set of N_{v} sought variables that define a scene description. Here v represents a variable index. Then [1], the measurement contribution function (MCF) is

$$f_d\left(\mathcal{M}, \boldsymbol{X}\right) = L_{\mathrm{e}} W_d\left(\boldsymbol{x}_B, \boldsymbol{\omega}_{B-1}\right) \prod_{b=1}^{B} g_b\left(\boldsymbol{x}_{b-1}, \boldsymbol{x}_b, \boldsymbol{x}_{b+1}\right). \tag{22}$$

Notice that $\boldsymbol{x}_{B+1}$ does not exist. We use it for the ease of notation. The way f_d is constructed is illustrated in Fig. 5 herein. Let $\mathcal{P}$ be the set of all possible paths, connecting $\boldsymbol{x}_{\mathrm{light}}$ to $\boldsymbol{x}_d$. Following Eq. (22) herein, the path integral formulation of the forward model is

$$\mathcal{F}_d\left(\mathcal{M}\right) = \int_{\mathcal{P}} f_d(\mathcal{M}, \boldsymbol{X}) d\boldsymbol{X}. \tag{23}$$

3.2 Scattering Tomography

In a scattering medium, MC light transport samples a path as consecutive rays $\mathcal{R}_b = (\boldsymbol{x}_b, \boldsymbol{\omega}_b)$, as follows:

`(i)` Direction $\boldsymbol{\omega}_0$ is uniformly sampled and a ray $\mathcal{R}_0 = (\boldsymbol{x}_{\mathrm{light}}, \boldsymbol{\omega}_0)$ is initialized. The PDF that corresponds to step `(i)` is

$$\mu_{\mathrm{ray}}(\mathcal{R}_0) = \frac{1}{4\pi} \tag{24}$$

Per iteration $b = 1, 2, ...$:

`(ii)` On ray $\mathcal{R}_{b-1}$, a random location $\boldsymbol{x}_b$ is sampled through these sub-steps:

`ii.a` Uniformly sample $u \in [0, 1]$.

`ii.b` Set $\tau = -\ln(1-u)$.

`ii.c` Numerically find distance l_b to the next interaction, such that

$$\tau = \int_0^{l_b} \beta\left(\boldsymbol{x}_{b-1} + l\boldsymbol{\omega}_{b-1}\right) dl. \tag{25}$$

The PDF [2] that corresponds to `(ii.a)`-`(ii.d)` is

$$\mu_l\left(l_b | \mathcal{R}_{b-1}\right) = \beta\left(\boldsymbol{x}_{b-1} + l_b\boldsymbol{\omega}_{b-1}\right) T(\boldsymbol{x}_{b-1}, \boldsymbol{x}_{b-1} + l_b\boldsymbol{\omega}_{b-1}). \tag{26}$$

`(iii)` Sample $\boldsymbol{\omega}_b$ using $\mathsf{f}_{\mathrm{p}}\left(\boldsymbol{x}_b, \boldsymbol{\omega}_{b-1} \cdot \boldsymbol{\omega}_b\right)$. The corresponding PDF here is trivially the phase function f_{p}, thus:

$$\mu_{\boldsymbol{\omega}}\left(\boldsymbol{\omega}_b | \mathcal{R}_{b-1}, \boldsymbol{x}_b\right) = \mathsf{f}_{\mathrm{p}}\left(\boldsymbol{x}_b, \boldsymbol{\omega}_{b-1} \cdot \boldsymbol{\omega}_b\right). \tag{27}$$

This yields ray $\mathcal{R}_b$.

`(iv)` End if the path leaves the medium or hits a detector. Else, iterate again.

Therefore [2], using Eq. (18,26,27) herein,

$$\begin{aligned}\mu_{\mathrm{ray}}\left(\mathcal{R}_b | \mathcal{R}_{b-1}\right) &= \mu_l\left(l_b | \mathcal{R}_{b-1}\right) \mu_{\boldsymbol{\omega}}\left(\boldsymbol{\omega}_b | \mathcal{R}_{b-1}, \boldsymbol{x}_b\right) G\left(\boldsymbol{x}_{b-1}, \boldsymbol{x}_b\right) \\ &= \mathsf{f}_{\mathrm{p}}\left(\boldsymbol{x}_b, \boldsymbol{\omega}_{b-1} \cdot \boldsymbol{\omega}_b\right) \beta\left(\boldsymbol{x}_b\right) T(\boldsymbol{x}_{b-1}, \boldsymbol{x}_b) G\left(\boldsymbol{x}_{b-1}, \boldsymbol{x}_b\right).\end{aligned} \tag{28}$$

The path end is at $\boldsymbol{x}_B$. Therefore, $\boldsymbol{\omega}_B$ is not defined. Thus, we define $\mathcal{R}_B \equiv (\boldsymbol{x}_B)$. The total path PDF is

$$\begin{aligned}\mu_d\left(\mathcal{M}, \boldsymbol{X}\right) = \mu_{\mathrm{ray}}(\mathcal{R}_0) \prod_{b=1}^{B} \mu_{\mathrm{ray}}\left(\mathcal{R}_b | \mathcal{R}_{b-1}\right) = \frac{1}{4\pi} T(\boldsymbol{x}_{B-1}, \boldsymbol{x}_B) G\left(\boldsymbol{x}_{B-1}, \boldsymbol{x}_B\right) \cdot \\ \prod_{b=1}^{B-1} \beta\left(\boldsymbol{x}_b\right) \mathsf{f}_{\mathrm{p}}\left(\boldsymbol{x}_b, \boldsymbol{\omega}_{b-1} \cdot \boldsymbol{\omega}_b\right) T(\boldsymbol{x}_{b-1}, \boldsymbol{x}_b) G\left(\boldsymbol{x}_{b-1}, \boldsymbol{x}_b\right).\end{aligned} \tag{29}$$

That is,

$$\mu_d\left(\mathcal{M}, \boldsymbol{X}\right) = \frac{1}{4\pi} \prod_{b=1}^{B} g_b\left(\boldsymbol{x}_{b-1}, \boldsymbol{x}_b, \boldsymbol{x}_{b+1}\right). \tag{30}$$

As presented in Eq. (7) of the main manuscript, the path contribution is

$$C_d\left(\mathcal{M}, \boldsymbol{X}\right) = \frac{f_d\left(\mathcal{M}, \boldsymbol{X}\right)}{\mu_d\left(\mathcal{M}, \boldsymbol{X}\right)}. \tag{31}$$

Using the explicit form of μ_d (Eq. 30 above), the path contribution is

$$C_d(\mathcal{M}, \boldsymbol{X}) = 4\pi L_{\mathrm{e}} W_d(\boldsymbol{x}_B, \boldsymbol{\omega}_{B-1}) \prod_{b=1}^{B-1} \varpi(\boldsymbol{x}_b). \tag{32}$$

Eq. (32) above corresponds to Eq. (9) from the main manuscript. Using a set $\Phi = \{\boldsymbol{X}_i\}_{i=1}^{N}$ of $N = |\Phi|$ sampled paths, MC estimates [2] $\mathcal{F}_d\left(\mathcal{M}\right)$ by

$$\hat{\mathcal{F}}_d\left(\mathcal{M}, \Phi\right) = \frac{1}{|\Phi|} \sum_{\boldsymbol{X}_i \in \Phi} C_d\left(\mathcal{M}, \boldsymbol{X}_i\right). \tag{33}$$

Recall the definition of the MCF f_d in Eq. (22) herein and notice the resemblance of f_d to μ_d. This is not a coincidence. The MC sampling procedure is tailored for applying importance sampling on the MCF. As discussed in Sec. 2.2 herein, the resemblance between f_d and μ_d reduces $\mathrm{Var}[\hat{\mathcal{F}}_d\left(\mathcal{M}, \Phi\right)]$.

Now we can derive an explicit form of the correction factor for scattering tomography. Let $T_t, \beta_t, \mathsf{f}_\mathrm{p}^t$ denote the transmittance, extinction coefficient, and phase function of the medium $\mathcal{M}_t$ at iteration t. Let $T_\mathrm{ref}, \beta_\mathrm{ref}, \mathsf{f}_\mathrm{p}^\mathrm{ref}$ denote the transmittance, extinction coefficient, and phase function of the reference medium $\mathcal{M}_\mathrm{ref}$ (iteration t'). Then, using Eq. (28) herein:

$$\begin{aligned} r_d(\mathcal{M}_t, \boldsymbol{X}|\mathcal{M}_\mathrm{ref}) &= \frac{\mu_d\left(\mathcal{M}_t, \boldsymbol{X}_i\right)}{\mu_d\left(\mathcal{M}_\mathrm{ref}, \boldsymbol{X}_i\right)} \\ &= \frac{T_t\left(\boldsymbol{x}_{B-1}, \boldsymbol{x}_B\right)}{T_\mathrm{ref}\left(\boldsymbol{x}_{B-1}, \boldsymbol{x}_B\right)} \prod_{b=1}^{B-1} \frac{\beta_t\left(\boldsymbol{x}_b\right) \mathsf{f}_\mathrm{p}^t\left(\boldsymbol{x}_b, D_b\right) T_t\left(\boldsymbol{x}_{b-1}, \boldsymbol{x}_b\right)}{\beta_\mathrm{ref}\left(\boldsymbol{x}_b\right) \mathsf{f}_\mathrm{p}^\mathrm{ref}\left(\boldsymbol{x}_b, D_b\right) T_\mathrm{ref}\left(\boldsymbol{x}_{b-1}, \boldsymbol{x}_b\right)}. \end{aligned} \tag{34}$$

Eq. (34) above corresponds to Eq. (29) of the main manuscript.

3.3 Reflectometry

Let a scene consist of only opaque surfaces. Then, MC light transport samples a path as follows [1]:
`(i)` Direction $\boldsymbol{\omega}_0$ is uniformly sampled and a ray $\mathcal{R}_0 = (\boldsymbol{x}_\mathrm{light}, \boldsymbol{\omega}_0)$ is initialized.
Per iteration $b = 1, 2, \ldots$:
`(ii)` Set $\boldsymbol{x}_b$ as the closest intersection between $\mathcal{R}_{b-1}$ and a visible surface.
`(iii)` Let Ω be the unit hemisphere and

$$a_b = \int_\Omega \mathsf{f}_\mathrm{r}\left(\boldsymbol{x}_b, \boldsymbol{\omega}_{b-1} \to \boldsymbol{\omega}_b\right) \left[\boldsymbol{\omega}_b \cdot \mathbf{n}\left(\boldsymbol{x}_b\right)\right] d\boldsymbol{\omega}_b \tag{35}$$

be the albedo of the surface at $\boldsymbol{x}_b$. Then, sample $\boldsymbol{\omega}_b$ using

$$\mu_{\boldsymbol{\omega}}\left(\boldsymbol{\omega}_b|\mathcal{R}_{b-1}, \boldsymbol{x}_b\right) = \frac{\mathsf{f}_\mathrm{r}\left(\boldsymbol{x}_b, \boldsymbol{\omega}_{b-1} \to \boldsymbol{\omega}_b\right) \left[\boldsymbol{\omega}_b \cdot \mathbf{n}\left(\boldsymbol{x}_b\right)\right]}{a_b}. \tag{36}$$

`(iv)` End if the path leaves the medium or hits a detector. Else, iterate again.

In analogy [1] to Eq. (29) herein and using Eq. (18) herein

$$\mu_d\left(\mathcal{M}, \boldsymbol{X}\right) = \frac{1}{4\pi} G\left(\boldsymbol{x}_{B-1}, \boldsymbol{x}_B\right) \prod_{b=1}^{B-1} \frac{\mathsf{f}_\mathrm{r}\left(\boldsymbol{x}_b, \boldsymbol{\omega}_{b-1} \to \boldsymbol{\omega}_b\right) G\left(\boldsymbol{x}_{b-1}, \boldsymbol{x}_b\right)}{a_b}. \tag{37}$$

Here the transmittance is omitted. Therefore, G must include a visibility indicator between points. The path contribution (based on Eqs. 22,37 herein) is then

$$C_d\left(\mathcal{M}, \boldsymbol{X}\right) = \frac{f_d\left(\mathcal{M}, \boldsymbol{X}\right)}{\mu_d\left(\mathcal{M}, \boldsymbol{X}\right)} = 4\pi L_\mathrm{e} W_d\left(\boldsymbol{x}_d, \boldsymbol{\omega}_{B-1}\right) \prod_{b=1}^{B-1} a_b. \tag{38}$$

We derive again an explicit form of the correction factor, but for reflectometry. Let $\mathsf{f}_\mathrm{r}^t, \mathsf{f}_\mathrm{r}^\mathrm{ref}$ be the BRDFs of the surface at iterations t and t', respectively. Then:

$$r_d(\mathcal{M}_t, \boldsymbol{X}|\mathcal{M}_\mathrm{ref}) = \frac{\mu_d\left(\mathcal{M}_t, \boldsymbol{X}_i\right)}{\mu_d\left(\mathcal{M}_\mathrm{ref}, \boldsymbol{X}_i\right)} = \prod_{b=1}^{B-1} \frac{\mathsf{f}_\mathrm{r}^t\left(\boldsymbol{x}^i, \boldsymbol{\omega}_{b-1}^i \to \boldsymbol{\omega}_b^i\right)}{\mathsf{f}_\mathrm{r}^\mathrm{ref}\left(\boldsymbol{x}^i, \boldsymbol{\omega}_{b-1}^i \to \boldsymbol{\omega}_b^i\right)}. \tag{39}$$

4 Partial Derivatives of the Forward Model

This section provides an analytical form of the forward model partial derivatives $\partial \mathcal{F}_d(\mathcal{M})/\partial m_v$ that are given in Sec. 3 of the main manuscript. Denote $\partial_{m_v} \triangleq \frac{\partial}{\partial m_v}$. Differentiating the forward model with respect to m_v results in:

$$\begin{aligned}\partial_{m_v}\mathcal{F}_d(\mathcal{M}) &= \int_{\mathcal{P}} \partial_{m_v} f_d(\mathcal{M}, \boldsymbol{X})\, d\boldsymbol{X} = \int_{\mathcal{P}} f_d(\mathcal{M},\boldsymbol{X}) \frac{\partial_{m_v} f_d(\mathcal{M},\boldsymbol{X})}{f_d(\mathcal{M},\boldsymbol{X})} d\boldsymbol{X} \\ &= \int_{\mathcal{P}} f_d(\mathcal{M},\boldsymbol{X}) \partial_{m_v} \ln f_d(\mathcal{M},\boldsymbol{X}) d\boldsymbol{X}.\end{aligned} \tag{40}$$

Eq. (40) above is the path integral formulation of the partial derivative of the forward model. It corresponds to Eq. (15) of the main manuscript. Applying Eq. (40) $\forall m_v \in \mathcal{M}$ results in a path integral formulation of the gradient of the forward model with respect to $\mathcal{M}$.

Deriving the derivative of the model boils down to deriving the derivative $\partial_{m_v} \ln f_d$. Eq. (22) herein shows that the MCF is a multiplication of $B+1$ functions, where $B+1$ is the path size. We follow [3] to derive a compact form that holds for an arbitrary B.

Using Eq. (22) herein in Eq. (40) herein:

$$\partial_{m_v} \ln f_d(\mathcal{M},\boldsymbol{X}) = \partial_{m_v} \ln \left[L_{\mathrm{e}} W_d \prod_{b=1}^{B} g_b \right] = \partial_{m_v} \left[\ln L_{\mathrm{e}} + \ln W_d + \sum_{b=1}^{B} \ln g_b \right] \tag{41}$$

Since L_{e} and the detector response function W_d do not depend on m_v (in our cases):

$$\partial_{m_v} \ln f_d(\mathcal{M},\boldsymbol{X}) = \partial_{m_v} \left[\sum_{b=1}^{B} \ln g_b \right] = \sum_{b=1}^{B} \frac{\partial_{m_v} g_b}{g_b}. \tag{42}$$

To conclude,

$$\partial_{m_v} \ln f_d(\mathcal{M},\boldsymbol{X}) = \sum_{b=1}^{B} \frac{\partial_{m_v} g_b(\boldsymbol{x}_{b-1}, \boldsymbol{x}_b, \boldsymbol{x}_{b+1})}{g_b(\boldsymbol{x}_{b-1}, \boldsymbol{x}_b, \boldsymbol{x}_{b+1})}. \tag{43}$$

4.1 Scattering Tomography Derivatives

Scattering CT seeks properties of particles in a scene. Let us focus on two particle types: air molecules and cloud droplets. Their respective parameters are $\beta^{\mathrm{a}}, \varpi^{\mathrm{a}}, \mathsf{f}_{\mathrm{p}}^{\mathrm{a}}$ and $\beta^{\mathrm{c}}, \varpi^{\mathrm{c}}, \mathsf{f}_{\mathrm{p}}^{\mathrm{c}}$. Suppose $\beta^{\mathrm{a}}(\boldsymbol{x}), \varpi^{\mathrm{c}}, \varpi^{\mathrm{a}}$ are known, and we seek $\beta^{\mathrm{c}}(\boldsymbol{x})$. Approximate the cloud extinction coefficient in voxel v as a constant, β_v^{c}. Thus, $\mathcal{M} = \{\beta_v^{\mathrm{c}}\}_{v=1}^{N_{\mathrm{v}}}$.

We now provide a detailed derivation of $\partial_{m_v} \ln f_d(\mathcal{M},\boldsymbol{X})$, which is used by Eq. (15) of the main manuscript. Denote the line segment between $\boldsymbol{x}_{b-1}$ and $\boldsymbol{x}_b$ by $\overline{\boldsymbol{x}_{b-1}\boldsymbol{x}_b}$. Voxel v has a domain $\mathcal{V}_v$. The intersection of $\mathcal{V}_v$ with $\overline{\boldsymbol{x}_{b-1}\boldsymbol{x}_b}$ has length $l_{b,v}$ (illustrated in Fig. 6 herein). The transmittance

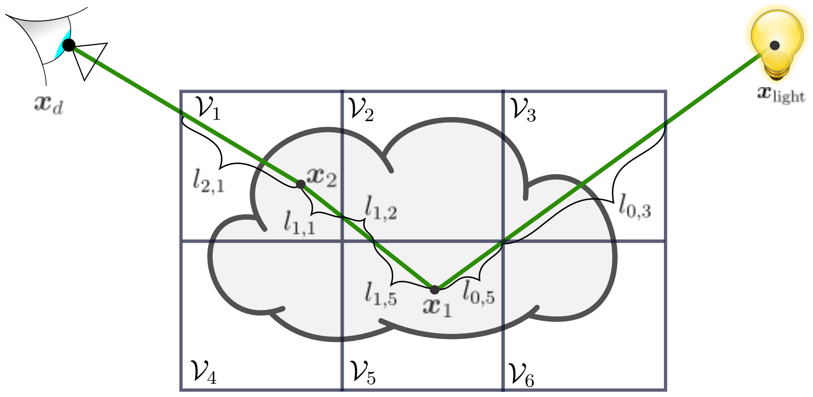

Figure 6: Illustration of $\mathcal{V}_v$ and $l_{b,v}$ on a path.

of the medium on the line segment $\overline{\boldsymbol{x}_{b-1}\boldsymbol{x}_b}$ is

$$T(\boldsymbol{x}_{b-1},\boldsymbol{x}_b) \approx \exp\left[-\sum_{v=1}^{N_{\rm u}} (\beta_v^{\rm a} + \beta_v^{\rm c}) \cdot l_{b,v}\right]. \tag{44}$$

We evaluate each summand of Eq. (43) herein separately. For $b = B$ (Eq. 21 herein), we get:

$$\frac{\frac{\partial}{\partial \beta_v^{\rm c}} g_B\left(\boldsymbol{x}_{B-1},\boldsymbol{x}_B,\boldsymbol{x}_{B+1}\right)}{g_B\left(\boldsymbol{x}_{B-1},\boldsymbol{x}_B,\boldsymbol{x}_{B+1}\right)} = \frac{\frac{\partial}{\partial \beta_v^{\rm c}}\left[T(\boldsymbol{x}_{B-1},\boldsymbol{x}_B) G\left(\boldsymbol{x}_{B-1},\boldsymbol{x}_B\right)\right]}{T(\boldsymbol{x}_{B-1},\boldsymbol{x}_B) G\left(\boldsymbol{x}_{B-1},\boldsymbol{x}_B\right)}. \tag{45}$$

Notice from Eqs. (18,44) herein that $T(\boldsymbol{x}_{B-1},\boldsymbol{x}_B)$ is the only term that may depend on $\beta_v^{\rm c}$. Therefore,

$$\frac{\frac{\partial}{\partial \beta_v^{\rm c}} g_B\left(\boldsymbol{x}_{B-1},\boldsymbol{x}_B,\boldsymbol{x}_{B+1}\right)}{g_B\left(\boldsymbol{x}_{B-1},\boldsymbol{x}_B,\boldsymbol{x}_{B+1}\right)} = \frac{\frac{\partial}{\partial \beta_v^{\rm c}}\left[T(\boldsymbol{x}_{B-1},\boldsymbol{x}_B)\right] G\left(\boldsymbol{x}_{B-1},\boldsymbol{x}_B\right)}{T(\boldsymbol{x}_{B-1},\boldsymbol{x}_B) G\left(\boldsymbol{x}_{B-1},\boldsymbol{x}_B\right)} = \frac{\frac{\partial}{\partial \beta_v^{\rm c}} T(\boldsymbol{x}_{B-1},\boldsymbol{x}_B)}{T(\boldsymbol{x}_{B-1},\boldsymbol{x}_B)}. \tag{46}$$

The transmittance derivative is

$$\begin{aligned}
\frac{\partial}{\partial \beta_v^{\rm c}} T(\boldsymbol{x}_{B-1},\boldsymbol{x}_B) &= \frac{\partial}{\partial \beta_v^{\rm c}} \exp\left[-\sum_{v'=1}^{N_{\rm u}} \left(\beta_{v'}^{\rm a} + \beta_{v'}^{\rm c}\right) l_{B,v'}\right] \\
&= -l_{1,v} \exp\left[-\sum_{v'=1}^{N_{\rm u}} \left(\beta_{v'}^{\rm a} + \beta_{v'}^{\rm c}\right) l_{B,v'}\right] \\
&= -l_{B,v} T(\boldsymbol{x}_{B-1},\boldsymbol{x}_B).
\end{aligned} \tag{47}$$

Plugging Eq. (47) herein in Eq. (46) above,

$$\frac{\frac{\partial}{\partial \beta_v^{\rm c}} g_B\left(\boldsymbol{x}_{B-1},\boldsymbol{x}_B,\boldsymbol{x}_{B+1}\right)}{g_B\left(\boldsymbol{x}_{B-1},\boldsymbol{x}_B,\boldsymbol{x}_{B+1}\right)} = -l_{B,v}. \tag{48}$$

For $b \in \{1, ..., B-1\}$:

$$\frac{\frac{\partial}{\partial \beta_v^{\mathrm{c}}} g_b\left(\boldsymbol{x}_{b-1}, \boldsymbol{x}_b, \boldsymbol{x}_{b+1}\right)}{g_b\left(\boldsymbol{x}_{b-1}, \boldsymbol{x}_b, \boldsymbol{x}_{b+1}\right)} = \frac{\frac{\partial}{\partial \beta_v^{\mathrm{c}}}\left[\mathsf{f}_{\mathrm{s}}\left(\boldsymbol{x}_{b-1}, \boldsymbol{x}_b, \boldsymbol{x}_{b+1}\right) T(\boldsymbol{x}_{b-1}, \boldsymbol{x}_b) G\left(\boldsymbol{x}_{b-1}, \boldsymbol{x}_b\right)\right]}{\mathsf{f}_{\mathrm{s}}\left(\boldsymbol{x}_{b-1}, \boldsymbol{x}_b, \boldsymbol{x}_{b+1}\right) T(\boldsymbol{x}_{b-1}, \boldsymbol{x}_b) G\left(\boldsymbol{x}_{b-1}, \boldsymbol{x}_b\right)}. \tag{49}$$

From Eq. (18) herein,$G\left(\boldsymbol{x}_{b-1}, \boldsymbol{x}_b\right)$ does not depend on β_v^{c}. Hence

$$\begin{aligned}\frac{\frac{\partial}{\partial \beta_v^{\mathrm{c}}} g_b\left(\boldsymbol{x}_{b-1}, \boldsymbol{x}_b, \boldsymbol{x}_{b+1}\right)}{g_b\left(\boldsymbol{x}_{b-1}, \boldsymbol{x}_b, \boldsymbol{x}_{b+1}\right)} &= \frac{\frac{\partial}{\partial \beta_v^{\mathrm{c}}}\left[\mathsf{f}_{\mathrm{s}}\left(\boldsymbol{x}_{b-1}, \boldsymbol{x}_b, \boldsymbol{x}_{b+1}\right) T(\boldsymbol{x}_{b-1}, \boldsymbol{x}_b)\right]}{\mathsf{f}_{\mathrm{s}}\left(\boldsymbol{x}_{b-1}, \boldsymbol{x}_b, \boldsymbol{x}_{b+1}\right) T(\boldsymbol{x}_{b-1}, \boldsymbol{x}_b)} \\ &= \frac{\frac{\partial}{\partial \beta_v^{\mathrm{c}}} \mathsf{f}_{\mathrm{s}}\left(\boldsymbol{x}_{b-1}, \boldsymbol{x}_b, \boldsymbol{x}_{b+1}\right)}{\mathsf{f}_{\mathrm{s}}\left(\boldsymbol{x}_{b-1}, \boldsymbol{x}_b, \boldsymbol{x}_{b+1}\right)} + \frac{\frac{\partial}{\partial \beta_v^{\mathrm{c}}} T(\boldsymbol{x}_{b-1}, \boldsymbol{x}_b)}{T(\boldsymbol{x}_{b-1}, \boldsymbol{x}_b)}.\end{aligned} \tag{50}$$

In analogy to Eqs. (47,48) herein, the second term in Eq. (50) above is given by:

$$\frac{\frac{\partial}{\partial \beta_v^{\mathrm{c}}} T(\boldsymbol{x}_{b-1}, \boldsymbol{x}_b)}{T(\boldsymbol{x}_{b-1}, \boldsymbol{x}_b)} = -l_{b,v}. \tag{51}$$

Let

$$D_b = \boldsymbol{\omega}_b \cdot \boldsymbol{\omega}_{b-1}. \tag{52}$$

Recall the definition of f_{s} for a scattering medium (Eq. 19 herein) Then,

$$\mathsf{f}_{\mathrm{s}}\left(\boldsymbol{x}_{b-1}, \boldsymbol{x}_b, \boldsymbol{x}_{b+1}\right) = \varpi\left(\boldsymbol{x}_b\right) \beta\left(\boldsymbol{x}_b\right) \mathsf{f}_{\mathrm{p}}\left(\boldsymbol{x}_b, D_b\right). \tag{53}$$

Recall Eqs. (13,14) herein. Then:

$$\begin{aligned}\varpi\left(\boldsymbol{x}_b\right) \beta\left(\boldsymbol{x}_b\right) &= \frac{\varpi^{\mathrm{c}} \beta^{\mathrm{c}}\left(\boldsymbol{x}_b\right) + \varpi^{\mathrm{a}} \beta^{\mathrm{a}}\left(\boldsymbol{x}_b\right)}{\beta\left(\boldsymbol{x}_b\right)} \beta\left(\boldsymbol{x}_b\right) \\ &= \varpi^{\mathrm{c}} \beta^{\mathrm{c}}\left(\boldsymbol{x}_b\right) + \varpi^{\mathrm{a}} \beta^{\mathrm{a}}\left(\boldsymbol{x}_b\right).\end{aligned} \tag{54}$$

We need to write Eq. (53) herein using the unknowns. If $\boldsymbol{x}_b \in \mathcal{V}_v$, then

$$\beta_{\mathrm{s}}^{\mathrm{c}}\left(\boldsymbol{x}_b\right) = \varpi^{\mathrm{c}} \beta_v^{\mathrm{c}}, \quad \beta_{\mathrm{s}}^{\mathrm{a}}\left(\boldsymbol{x}_b\right) = \varpi^{\mathrm{a}} \beta_v^{\mathrm{a}}. \tag{55}$$

Plugging Eq. (55) above into Eq. (53) herein,

$$\mathsf{f}_{\mathrm{s}}\left(\boldsymbol{x}_{b-1}, \boldsymbol{x}_b, \boldsymbol{x}_{b+1}\right) = \varpi^{\mathrm{c}} \beta_v^{\mathrm{c}} \cdot \mathsf{f}_{\mathrm{p}}^{\mathrm{c}}\left(\boldsymbol{x}_b, D_b\right) + \varpi^{\mathrm{a}} \beta_v^{\mathrm{a}} \cdot \mathsf{f}_{\mathrm{p}}^{\mathrm{a}}\left(\boldsymbol{x}_b, D_b\right). \tag{56}$$

Thus,

$$\frac{\frac{\partial}{\partial \beta_v^{\mathrm{c}}} \mathsf{f}_{\mathrm{s}}\left(\boldsymbol{x}_{b-1}, \boldsymbol{x}_b, \boldsymbol{x}_{b+1}\right)}{\mathsf{f}_{\mathrm{s}}\left(\boldsymbol{x}_{b-1}, \boldsymbol{x}_b, \boldsymbol{x}_{b+1}\right)} = \frac{\varpi^{\mathrm{c}} \mathsf{f}_{\mathrm{p}}^{\mathrm{c}}\left(\boldsymbol{x}_b, D_b\right)}{\mathsf{f}_{\mathrm{s}}\left(\boldsymbol{x}_{b-1}, \boldsymbol{x}_b, \boldsymbol{x}_{b+1}\right)} = \left[\beta_v^{\mathrm{c}} + \frac{\varpi^{\mathrm{a}} \mathsf{f}_{\mathrm{p}}^{\mathrm{a}}\left(\boldsymbol{x}_b, D_b\right)}{\varpi^{\mathrm{c}} \mathsf{f}_{\mathrm{p}}^{\mathrm{c}}\left(\boldsymbol{x}_b, D_b\right)} \beta_v^{\mathrm{a}}\right]^{-1}. \tag{57}$$

Otherwise, if $\boldsymbol{x}_b \notin \mathcal{V}_v$, then

$$\frac{\frac{\partial}{\partial \beta_v^{\mathrm{c}}} \mathsf{f}_{\mathrm{s}}\left(\boldsymbol{x}_{b-1}, \boldsymbol{x}_b, \boldsymbol{x}_{b+1}\right)}{\mathsf{f}_{\mathrm{s}}\left(\boldsymbol{x}_{b-1}, \boldsymbol{x}_b, \boldsymbol{x}_{b+1}\right)} = 0. \tag{58}$$

We note that [4] provides similar derivations. However, the factor $\dfrac{\varpi^{\mathrm{a}} \mathsf{f}_{\mathrm{p}}^{\mathrm{a}}\left(\boldsymbol{x}_b, D_b\right)}{\varpi^{\mathrm{c}} \mathsf{f}_{\mathrm{p}}^{\mathrm{c}}\left(\boldsymbol{x}_b, D_b\right)}$ in Eq. (57) is missing from their derivation. Let $\mathbb{I}(\mathcal{A})$ be an indicator, which equals 1 if $\mathcal{A}$ occurs (0 otherwise). Plugging Eqs. (51,57,58) above in Eq. (50) herein,

$$\frac{\frac{\partial}{\partial \beta_v^{\mathrm{c}}} g_b\left(\boldsymbol{x}_{b-1}, \boldsymbol{x}_b, \boldsymbol{x}_{b+1}\right)}{g_b\left(\boldsymbol{x}_{b-1}, \boldsymbol{x}_b, \boldsymbol{x}_{b+1}\right)} = -l_{b,v} + \mathbb{I}(\boldsymbol{x}_b \in \mathcal{V}_v) \left[\beta_v^{\mathrm{c}} + \frac{\varpi^{\mathrm{a}} \mathsf{f}_{\mathrm{p}}^{\mathrm{a}}\left(\boldsymbol{x}_b, D_b\right)}{\varpi^{\mathrm{c}} \mathsf{f}_{\mathrm{p}}^{\mathrm{c}}\left(\boldsymbol{x}_b, D_b\right)} \beta_v^{\mathrm{a}}\right]^{-1}. \tag{59}$$

Based on Eqs. (48,59) herein, we conclude:

$$\partial_{m_v} \ln f_d\left(\mathcal{M}, \boldsymbol{X}\right) = -\sum_{b=1}^{B} l_{b,v} + \sum_{b=1}^{B-1} \mathbb{I}(\boldsymbol{x}_b \in \mathcal{V}_v) \left[\beta_v^{\mathrm{c}} + \frac{\varpi^{\mathrm{a}} \mathsf{f}_{\mathrm{p}}^{\mathrm{a}}\left(\boldsymbol{x}_b, D_b\right)}{\varpi^{\mathrm{c}} \mathsf{f}_{\mathrm{p}}^{\mathrm{c}}\left(\boldsymbol{x}_b, D_b\right)} \beta_v^{\mathrm{a}}\right]^{-1}. \tag{60}$$

Eq. (60) above corresponds to Eq. (20) of the main manuscript.

4.2 Reflectometry (Phong model) Derivatives

Now, we evaluate Eq. (43) herein for a reflectometry toy example. Eq. (20) herein in a surface-only scene is

$$g_b\left(\boldsymbol{x}_{b-1}, \boldsymbol{x}_b, \boldsymbol{x}_{b+1}\right) = \mathsf{f}_{\mathrm{r}}\left(\boldsymbol{x}_b, \boldsymbol{\omega}_{b-1} \to \boldsymbol{\omega}_b\right) G\left(\boldsymbol{x}_{b-1}, \boldsymbol{x}_b\right). \tag{61}$$

Thus, using Eqs. (18,19,20),

$$\frac{\frac{\partial}{\partial m_v} g_b\left(\boldsymbol{x}_{b-1}, \boldsymbol{x}_b, \boldsymbol{x}_{b+1}\right)}{g_b\left(\boldsymbol{x}_{b-1}, \boldsymbol{x}_b, \boldsymbol{x}_{b+1}\right)} = \frac{\frac{\partial}{\partial m_v}\left[\mathsf{f}_{\mathrm{r}}\left(\boldsymbol{x}_b, \boldsymbol{\omega}_{b-1} \to \boldsymbol{\omega}_b\right) G\left(\boldsymbol{x}_{b-1}, \boldsymbol{x}_b\right)\right]}{\mathsf{f}_{\mathrm{r}}\left(\boldsymbol{x}_b, \boldsymbol{\omega}_{b-1} \to \boldsymbol{\omega}_b\right) G\left(\boldsymbol{x}_{b-1}, \boldsymbol{x}_b\right)} = \frac{\frac{\partial}{\partial m_v}\left[\mathsf{f}_{\mathrm{r}}\left(\boldsymbol{x}_b, \boldsymbol{\omega}_{b-1} \to \boldsymbol{\omega}_b\right)\right]}{\mathsf{f}_{\mathrm{r}}\left(\boldsymbol{x}_b, \boldsymbol{\omega}_{b-1} \to \boldsymbol{\omega}_b\right)}. \tag{62}$$

We are interested in estimating $\mathcal{M} = \{\kappa_{\mathrm{d}}, \kappa_{\mathrm{s}}, \gamma\}$ of a surface patch S. Let $\boldsymbol{\omega}_{b-1}^{\mathrm{s}}$ be the of specular reflection of $\boldsymbol{\omega}_{b-1}$ and

$$D_{b,\mathrm{s}} = \boldsymbol{\omega}_{b-1}^{\mathrm{s}} \cdot \boldsymbol{\omega}_b. \tag{63}$$

Then, define $\forall \boldsymbol{x}_b \in S$,

$$\mathsf{f}_{\mathrm{r}}\left(\boldsymbol{x}_b, \boldsymbol{\omega}_{b-1} \to \boldsymbol{\omega}_b\right) = \frac{\kappa_{\mathrm{d}}}{\pi} + \kappa_{\mathrm{s}} \frac{\gamma + 2}{2\pi} D_{b,\mathrm{s}}^{\gamma}. \tag{64}$$

Let $\boldsymbol{x}_b \in S$. For $m_v = \kappa_{\mathrm{d}}$:

$$\frac{\frac{\partial}{\partial \kappa_{\mathrm{d}}} \mathsf{f}_{\mathrm{r}}\left(\boldsymbol{x}_b, \boldsymbol{\omega}_{b-1} \to \boldsymbol{\omega}_b\right)}{\mathsf{f}_{\mathrm{r}}\left(\boldsymbol{x}_b, \boldsymbol{\omega}_{b-1} \to \boldsymbol{\omega}_b\right)} = \frac{\frac{1}{\pi}}{\mathsf{f}_{\mathrm{r}}\left(\boldsymbol{x}_b, \boldsymbol{\omega}_{b-1} \to \boldsymbol{\omega}_b\right)}. \tag{65}$$

For $m_v = \kappa_{\mathrm{s}}$,

$$\frac{\frac{\partial}{\partial \kappa_{\mathrm{s}}} \mathsf{f}_{\mathrm{r}}\left(\boldsymbol{x}_b, \boldsymbol{\omega}_{b-1} \to \boldsymbol{\omega}_b\right)}{\mathsf{f}_{\mathrm{r}}\left(\boldsymbol{x}_b, \boldsymbol{\omega}_{b-1} \to \boldsymbol{\omega}_b\right)} = \frac{\frac{\gamma+2}{2\pi} D_{b,\mathrm{s}}^{\gamma}}{\mathsf{f}_{\mathrm{r}}\left(\boldsymbol{x}_b, \boldsymbol{\omega}_{b-1} \to \boldsymbol{\omega}_b\right)}. \tag{66}$$

For $m_v = \gamma$:

$$\frac{\frac{\partial}{\partial \gamma} \mathsf{f}_{\mathrm{r}}\left(\boldsymbol{x}_b, \boldsymbol{\omega}_{b-1} \to \boldsymbol{\omega}_b\right)}{\mathsf{f}_{\mathrm{r}}\left(\boldsymbol{x}_b, \boldsymbol{\omega}_{b-1} \to \boldsymbol{\omega}_b\right)} = \frac{\frac{1}{\pi} \kappa_{\mathrm{s}} \left[\frac{D_{b,\mathrm{s}}^{\gamma}}{2} + \frac{\gamma+2}{2} D_{b,\mathrm{s}}^{\gamma} \ln\left(D_{b,\mathrm{s}}\right)\right]}{\mathsf{f}_{\mathrm{r}}\left(\boldsymbol{x}_b, \boldsymbol{\omega}_{b-1} \to \boldsymbol{\omega}_b\right)} = \frac{\kappa_{\mathrm{s}} \frac{\gamma+2}{2\pi} D_{b,\mathrm{s}}^{\gamma} \left[\frac{1}{\gamma+2} + \ln\left(D_{b,\mathrm{s}}\right)\right]}{\mathsf{f}_{\mathrm{r}}\left(\boldsymbol{x}_b, \boldsymbol{\omega}_{b-1} \to \boldsymbol{\omega}_b\right)}. \tag{67}$$

Suppose $\boldsymbol{x}_b$ is not on the path S ($\boldsymbol{x}_b \notin S$). Then,

$$\frac{\frac{\partial}{\partial m_v} \mathsf{f}_{\mathrm{r}}\left(\boldsymbol{x}_b, \boldsymbol{\omega}_{b-1} \to \boldsymbol{\omega}_b\right)}{\mathsf{f}_{\mathrm{r}}\left(\boldsymbol{x}_b, \boldsymbol{\omega}_{b-1} \to \boldsymbol{\omega}_b\right)} = 0. \tag{68}$$

Plugging Eqs. (65,66,67,68) herein in Eq. (43) herein results in

$$\frac{\partial}{\partial \kappa_{\mathrm{d}}} \ln f_d\left(\mathcal{M}, \boldsymbol{X}\right) = \sum_{b=1}^{B-1} \mathbb{I}(\boldsymbol{x}_b \in S) \frac{\frac{1}{\pi}}{\mathsf{f}_{\mathrm{r}}\left(\boldsymbol{x}_b, \boldsymbol{\omega}_{b-1} \to \boldsymbol{\omega}_b\right)}, \tag{69}$$

$$\frac{\partial}{\partial \kappa_{\mathrm{s}}} \ln f_d\left(\mathcal{M}, \boldsymbol{X}\right) = \sum_{b=1}^{B-1} \mathbb{I}(\boldsymbol{x}_b \in S) \frac{\frac{\gamma+2}{2\pi} D_b^{\gamma}}{\mathsf{f}_{\mathrm{r}}\left(\boldsymbol{x}_b, \boldsymbol{\omega}_{b-1} \to \boldsymbol{\omega}_b\right)}, \tag{70}$$

and

$$\frac{\partial}{\partial \gamma} \ln f_d\left(\mathcal{M}, \boldsymbol{X}\right) = \sum_{b=1}^{B-1} \mathbb{I}(\boldsymbol{x}_b \in S) \frac{\kappa_{\mathrm{s}} \frac{\gamma+2}{2\pi} D_b^{\gamma} \left(\frac{1}{\gamma+2} + \ln D_b\right)}{\mathsf{f}_{\mathrm{r}}\left(\boldsymbol{x}_b, \boldsymbol{\omega}_{b-1} \to \boldsymbol{\omega}_b\right)}. \tag{71}$$

5 Results

This section provides a visual comparison with [4] (Fig. 7 herein). Then, it extends the ablation studies of the main manuscript (see Fig. 8 herein).

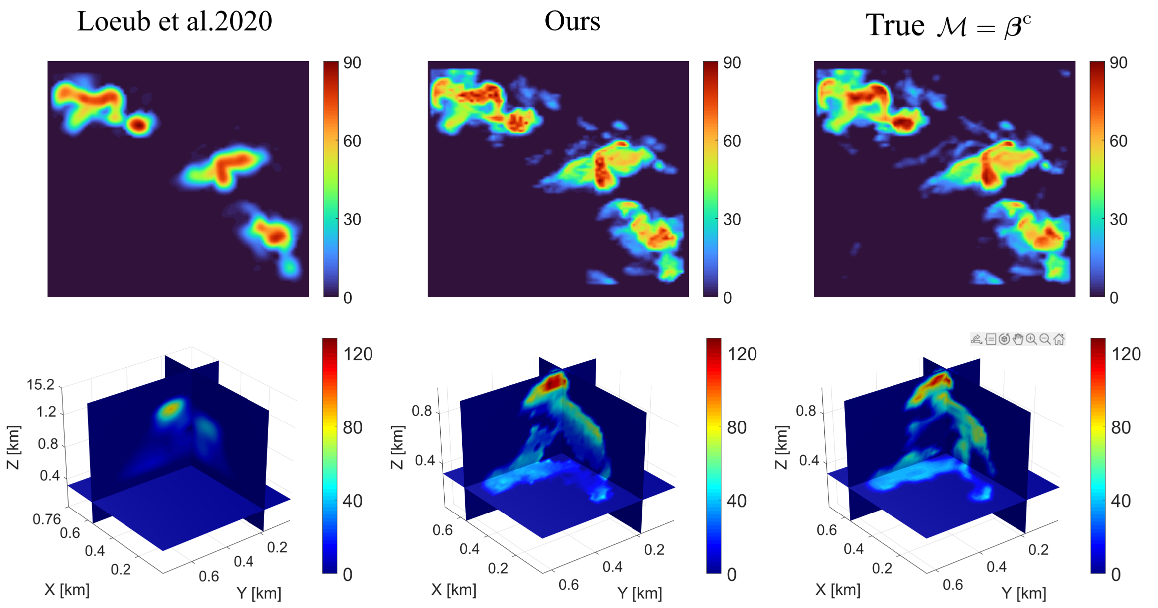

Figure 7: Visual comparisons with [4]. [Top] 3D comparison of the *cloud field* scene. These plots are rendered using Maximum Intensity Projection (MIP). [Bottom] 2D slices comparison of the *solitude cloud* scene.

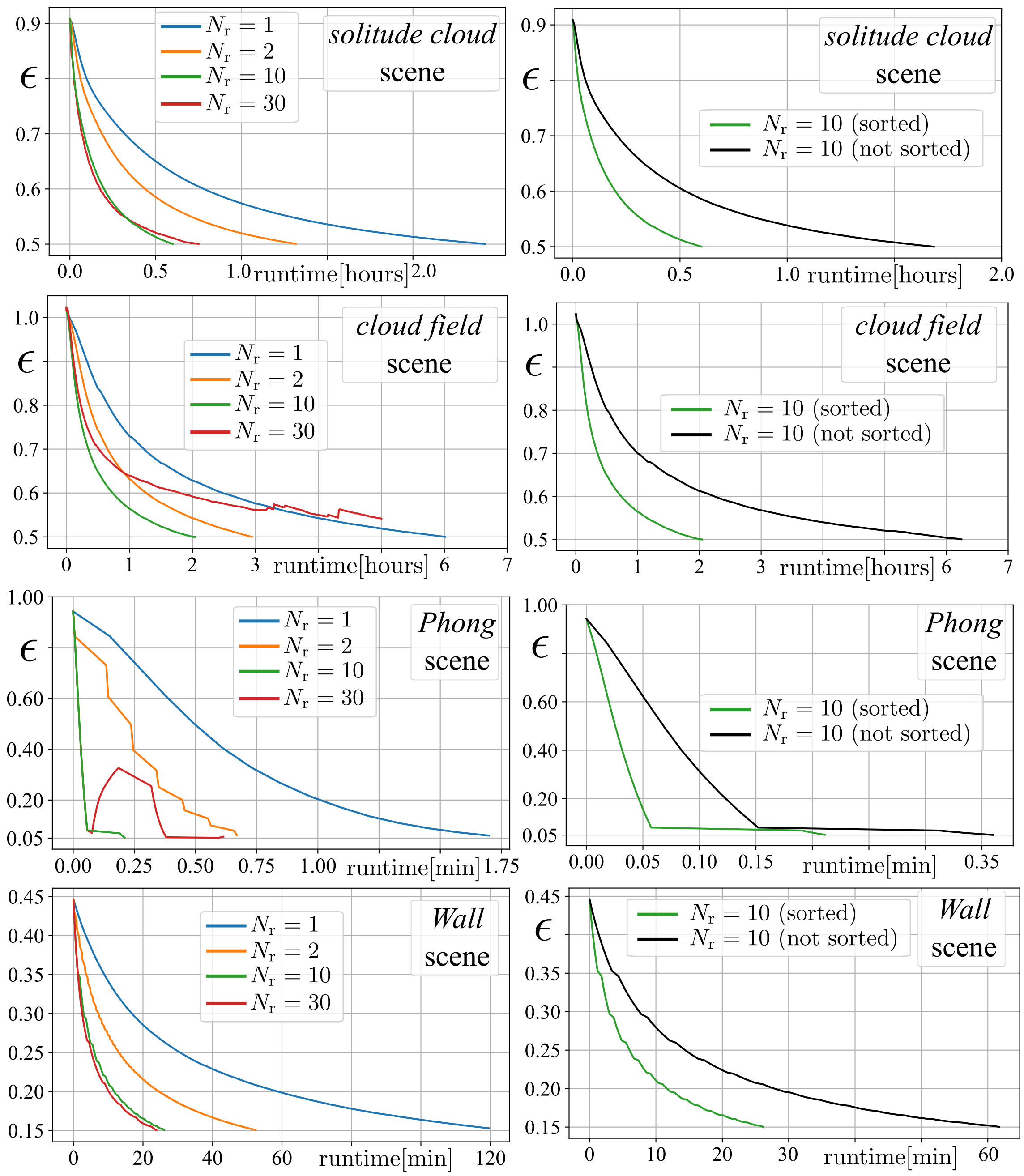

Figure 8: ϵ vs. runtime of 4 synthetic scenes. A recycling period of N_r iterations starts at *reference* iteration t' and ends at $t' + N_r$. [Left] Comparison of different values of N_r (sorting is not used for $N_r = 1$). [Right] Sorting vs. no sorting, using $N_r = 10$. While for $N_r = 1$, each iteration contributes more to the reduction of the error measures, with $N_r = 10$ and sorting we can run more iterations per second, leading to overall speedup.