A Differential Theory of Radiative Transfer

CHENG ZHANG, University of California, Irvine
LIFAN WU, University of California, San Diego
CHANGXI ZHENG, Columbia University
IOANNIS GKIOULEKAS, Carnegie Mellon University
RAVI RAMAMOORTHI, University of California, San Diego
SHUANG ZHAO, University of California, Irvine

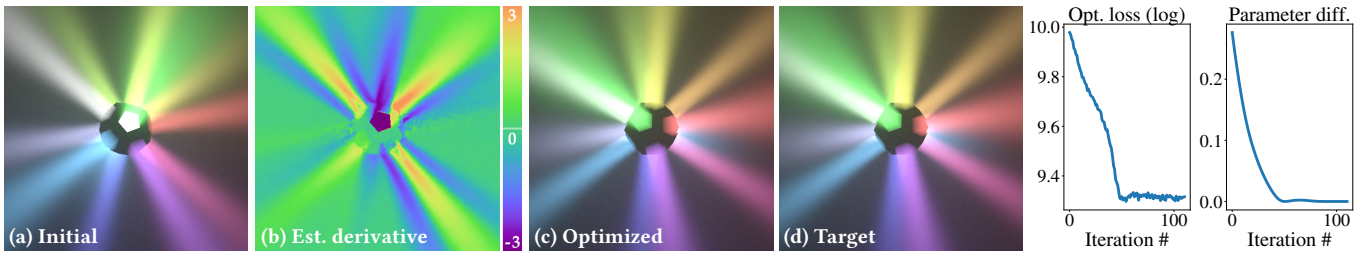

Fig. 1. We introduce a new differential theory of radiative transfer, which lays the foundation for computing the derivatives of radiometric measures with respect to arbitrary scene parameterizations (e.g., material properties and object geometries). The ability to evaluate these derivatives can facilitate gradient-based optimization for many diverse applications. As an example, here we optimize the pose of a dodecahedron emitting colored beams inside a participating medium. Given a target image (d) and an initial configuration (a), the optimization uses derivatives estimated by our method (b) to find parameters that produce rendered images (c) closely matching the target. Per-iteration optimization loss and difference between true and estimated parameters (both measured in L_2) are plotted on the right.

Physics-based differentiable rendering is the task of estimating the derivatives of radiometric measures with respect to scene parameters. The ability to compute these derivatives is necessary for enabling gradient-based optimization in a diverse array of applications: from solving analysis-by-synthesis problems to training machine learning pipelines incorporating forward rendering processes. Unfortunately, physics-based differentiable rendering remains challenging, due to the complex and typically nonlinear relation between pixel intensities and scene parameters.

We introduce a differential theory of radiative transfer, which shows how individual components of the radiative transfer equation (RTE) can be differentiated with respect to arbitrary differentiable changes of a scene. Our theory encompasses the same generality as the standard RTE, allowing differentiation while accurately handling a large range of light transport phenomena such as volumetric absorption and scattering, anisotropic phase functions, and heterogeneity. To numerically estimate the derivatives given by our theory, we introduce an unbiased Monte Carlo estimator supporting arbitrary surface and volumetric configurations. Our technique differentiates path contributions symbolically and uses additional boundary integrals to capture geometric discontinuities such as visibility changes.

We validate our method by comparing our derivative estimations to those generated using the finite-difference method. Furthermore, we use a few synthetic examples inspired by real-world applications in inverse rendering, non-line-of-sight (NLOS) and biomedical imaging, and design, to demonstrate the practical usefulness of our technique.

CCS Concepts: • **Computing methodologies** → **Rendering**.

Additional Key Words and Phrases: radiative transfer, differentiable rendering, Monte Carlo path tracing

ACM Reference Format:
Cheng Zhang, Lifan Wu, Changxi Zheng, Ioannis Gkioulekas, Ravi Ramamoorthi, and Shuang Zhao. 2019. A Differential Theory of Radiative Transfer. *ACM Trans. Graph.* 38, 6, Article 227 (November 2019), 16 pages. https://doi.org/10.1145/3355089.3356522

Authors' addresses: Cheng Zhang, University of California, Irvine, chengz20@uci.edu; Lifan Wu, University of California, San Diego, liw086@eng.ucsd.edu; Changxi Zheng, Columbia University, cxz@cs.columbia.edu; Ioannis Gkioulekas, Carnegie Mellon University, igkioule@andrew.cmu.edu; Ravi Ramamoorthi, University of California, San Diego, ravir@cs.ucsd.edu; Shuang Zhao, University of California, Irvine, shz@ics.uci.edu.

0730-0301/2019/11-ART227 $15.00
https://doi.org/10.1145/3355089.3356522

1 INTRODUCTION

A fundamental task of physics-based light transport simulation is to compute the radiant power (generally measured using radiance) at certain 3D locations and directions in a virtual scene, e.g., those corresponding to radiometric sensors. Such *forward* evaluations of light transport have been a focus of research efforts in computer graphics since the field's inception. These efforts have resulted in mature forward rendering algorithms, including Monte Carlo techniques, that can efficiently and accurately simulate complex light transport effects such as interreflections and subsurface scattering.

Mathematically, it is convenient to be capable of evaluating not only a given function but also its various transformations. One such

transformation of great importance is the function's derivatives. In the context of light transport simulation, this corresponds to *differentiable rendering*, the process of computing the derivatives of radiometric measurements with respect to differential changes of virtual scenes. Differentiable rendering techniques can enable, for example, *gradient-based optimization* when solving high-dimensional *inverse rendering* problems, where exhaustive search is typically impractical. Additionally, these techniques facilitate the efficient integration of physics-based light transport simulation in machine learning and probabilistic inference pipelines, where differentiability is a key requirement.

Unfortunately, and unlike their forward counterparts, derivatives of radiometric measurements remain challenging to estimate. On one hand, general-purpose numerical tools for computing derivatives such as finite differences are either too noisy or too biased when combined with forward Monte Carlo simulations, and scale poorly to high-dimensional gradients. On the other hand, existing differentiable rendering algorithms are not sufficiently general to support differentiation with respect to arbitrary parameters in the presence of nontrivial scene geometry and complex light transport effects.

At the core of the challenge is the lack of a comprehensive mathematical framework for differential radiometric quantities that encompasses all types of high-order light transport. This is in contrast with the forward rendering case, where this role is fulfilled by the radiative transfer and rendering equations, which have provided the foundation for the development of Monte Carlo forward rendering algorithms. We seek to fix this discrepancy by developing a *differential* theory of radiative transfer of the same generality as its forward form. Additionally, we use this theory to derive Monte Carlo algorithms capable of differentiating arbitrary radiative transfer effects (e.g., volumetric absorption and scattering, anisotropic phase functions, and heterogeneity) with respect to arbitrary scene parameterizations. This significantly extends prior differentiable rendering techniques that are restricted to differentiating only surface-based light transport [Li et al. 2018a].

Concretely, our contributions include:

- The derivation of differential forms of the radiative transfer and rendering equations, supporting differentiation with respect to arbitrary scene parameterizations (§4).
- An unbiased Monte Carlo algorithm for estimating the derivatives of radiometric measures in the presence of arbitrarily complex light transport (§5).

We validate our theory and algorithms by comparing our estimated derivatives with those produced using finite differences (Figure 10). Additionally, we show applications of our framework in synthetic examples of several inverse rendering problems (Figure 11), including examples relevant to non-line-of-sight (NLOS) and biomedical imaging (Figure 12), as well as design tasks (Figure 13).

2 RELATED WORK

Radiative transfer. Radiative transfer has been used in natural sciences and application areas, including astrophysics, neutron transport, material science, remote sensing and biomedical imaging [van de Hulst et al. 1957; Chandrasekhar 1960; Pomraning 1973; Ishimaru 1978; Mishchenko et al. 2006]. This framework was later introduced to computer graphics [Blinn 1982; Kajiya and Von Herzen 1984; Rushmeier and Torrance 1987], where it has been generalized to support light transport effects such as those arising from anisotropic media with organized structures [Jakob et al. 2010] and non-exponential media [Bitterli et al. 2018].

Monte Carlo volume rendering. Monte Carlo methods have been the "gold standard" for simulating photon and neutron transport in complex environments [Spanier and Gelbard 1969]. In computer graphics, volumetric path tracing and its variations (e.g., [Kajiya and Von Herzen 1984; Lafortune and Willems 1996; Cerezo et al. 2005]) produce unbiased and consistent estimates of radiometric measures. Later-developed Markov-Chain Monte Carlo (MCMC) methods (e.g., [Pauly et al. 2000; Kelemen et al. 2002; Jakob and Marschner 2012]) additionally enable efficient simulation of challenging effects such as volumetric caustics. For a comprehensive survey on Monte Carlo volume rendering techniques, we refer to the survey by Nováak et al. [2018].

Derivatives for rendering. Analytical derivatives have been used in computer graphics to compute pixel footprints [Igehy 1999], handle specular light paths [Chen and Arvo 2000; Jakob and Marschner 2012], use Hamiltonian Monte Carlo in rendering [Li et al. 2015], and enable interactive material editing [Hašan and Ramamoorthi 2013]. These derivatives are specialized for certain types of light transport effects, and usually neglect geometric discontinuities.

Arvo [1994] presented an analytical method for calculating the gradients of irradiance. This method is specialized for diffuse surfaces and requires clipping triangle meshes, making it difficult to generalize to arbitrary materials or scale to complex scenes.

Ramamoorthi et al. [2007] introduced a first-order analysis of light transport, focusing on effects such as soft shadows. Our work can be considered a generalization of this theoretical framework to include other effects, such as volumetric light transport.

Physics-based differentiable rendering. Differentiable rendering has been used to solve analysis-by-synthesis problems in volumetric scattering [Gkioulekas et al. 2013, 2016], cloth rendering [Khungurn et al. 2015], prefiltering of high-resolution volumes [Zhao et al. 2016], appearance modeling of human teeth [Velinov et al. 2018], fabrication of translucent materials [Sumin et al. 2019], reflectance and lighting estimation [Azinovic et al. 2019], and 3D reconstruction [Tsai et al. 2019]. All these methods compute radiance derivatives using algorithms specialized to specific light transport effects.

A few recent works have focused on extending differentiable rendering to more general settings. Che et al. [2018] developed a system capable of computing derivatives with respect to optical material (reflectance, scattering) and local shape (normal) properties. Unfortunately, this technique cannot differentiate with respect to global changes of the scene geometry: As we discuss in §3, this requires calculating derivatives of integrals with evolving boundaries, which in turn requires evaluating additional boundary integrals.

To overcome these challenges, Li et al. [2018a] generalized the work by Ramamoorthi et al. [2007] and presented a technique that

Table 1. List of symbols commonly used in this paper.

Symbol	Meaning	Def.
L, L^{ins}	interior and in-scattered radiances	(1, 5)
L_{s}, L_{s}^{r},	interfacial and reflected/refracted radiances	(7)
L^{e}, L_{s}^{e}	volumetric and interfacial emissions	(6, 7)
$\mathcal{K}_{\text{T}}$, $\mathcal{K}_{\text{C}}$	transport and collision operators	(2, 5)
$\mathcal{K}_{\text{S}}$	interfacial scattering operator	(9)
D	distance for the ray $(\boldsymbol{x}, -\boldsymbol{\omega})$ to intersect $\partial\Omega$	(3)
$T(\boldsymbol{x}, \boldsymbol{y})$	transmittance between $\boldsymbol{x}$ and $\boldsymbol{y}$	(4)
σ_{t}, σ_{s}, σ_{a}	extinction, scattering, and absorption coeff.	(4–6)
f_{p}	single-scattering phase function	(5)
f_{s}	cosine-weighted BSDF	(7)
$\mathrm{d}_{\boldsymbol{\pi}} f$	total derivative of f wrt. all scene parameters $\boldsymbol{\pi}$	
$\partial_{\pi} f$, $\dot{f}$	partial derivative of f wrt. one scene parameter π	
$\langle \boldsymbol{u}, \boldsymbol{v} \rangle$	dot (inner) product between vectors $\boldsymbol{u}$ and $\boldsymbol{v}$	
$\boldsymbol{x} \to \boldsymbol{y}$	unit vector pointing from $\boldsymbol{x}$ to $\boldsymbol{y}$, i.e., $\frac{\boldsymbol{y}-\boldsymbol{x}}{\|\boldsymbol{y}-\boldsymbol{x}\|}$	
$\boldsymbol{x}'$, $\dot{\boldsymbol{x}}'$	$\boldsymbol{x} - \tau\boldsymbol{\omega}$, $\dot{\boldsymbol{x}} - \tau\dot{\boldsymbol{\omega}}$	
$\boldsymbol{x}_0$, $\dot{\boldsymbol{x}}_0$	$\boldsymbol{x} - D\boldsymbol{\omega}$, $\dot{\boldsymbol{x}} - \dot{D}\boldsymbol{\omega} - D\dot{\boldsymbol{\omega}}$	

uses an edge-sampling process to handle the boundary terms. Despite significantly extending the range of parameters that can be differentiated, this work still neglects all volumetric light transport effects, by essentially differentiating only the rendering equation but not the radiative transfer equation. Our theory subsumes this work by showing how to differentiate both equations, resulting in the most general differentiable rendering framework to date.

Derivatives for vision. Having derivatives of rendered images also enables back-propagation through rendering processes, allowing them to be efficiently integrated into probabilistic programming [Kulkarni et al. 2015] and deep learning pipelines (e.g., as the decoder of an auto-encoder architecture [Vincent et al. 2010]). In fact, many recent works [Wu et al. 2017; Meka et al. 2018; Sengupta et al. 2018; Kato et al. 2018; Li et al. 2018b; Che et al. 2018] leverage various forms of rendering losses to regularize the training, and improve generalization, of neural networks. The renderers used in most of these works make restrictive simplifications such as direct-only illumination [Loper and Black 2014].

3 PRELIMINARIES

We start by briefly reviewing the radiative transfer framework (§3.1). We then define the radiance derivative with respect to arbitrary scene parameterizations (§3.2). Finally, we introduce mathematical background that paves the way for developing our theory of differential radiative transfer (§3.3). Table 1 summarizes our notation.

3.1 Radiative Transfer

Radiative transfer [Chandrasekhar 1960] uses energy conservation principles to model light transport in participating media. At its core is the *radiative transfer equation* (RTE): Consider a medium confined in a volume $\Omega \subseteq \mathbb{R}^3$ with boundary $\partial\Omega$. The *interior radiance field* $L(\boldsymbol{x}, \boldsymbol{\omega})$ is defined over positions $\boldsymbol{x} \in \Omega \setminus \partial\Omega$ and light-transport directions $\boldsymbol{\omega} \in \mathbb{S}^2$. The *steady-state* RTE is a linear integral equation on L that can be expressed in operator form as [Case and Zweifel 1967; Spanier and Gelbard 1969]:

$$L = \mathcal{K}_{\text{T}}\,\mathcal{K}_{\text{C}}\,L + Q. \tag{1}$$

Concretely, the *transport operator* $\mathcal{K}_{\text{T}}$ maps any function $g : (\Omega \setminus \partial\Omega) \times \mathbb{S}^2 \to \mathbb{R}_+$ to a new function

$$(\mathcal{K}_{\text{T}}\,g)(\boldsymbol{x}, \boldsymbol{\omega}) = \int_0^D T(\boldsymbol{x}', \boldsymbol{x})\, g(\boldsymbol{x}', \boldsymbol{\omega})\,\mathrm{d}\tau, \tag{2}$$

where: $\boldsymbol{x}' := \boldsymbol{x} - \tau\boldsymbol{\omega}$; D is the distance from $\boldsymbol{x}$ to the medium's boundary in the direction of $-\boldsymbol{\omega}$,

$$D = \inf\{\tau \in \mathbb{R}_+ \;:\; \boldsymbol{x} - \tau\boldsymbol{\omega} \in \partial\Omega\}; \tag{3}$$

and $T(\boldsymbol{x}', \boldsymbol{x})$ is the *transmittance* between $\boldsymbol{x}'$ and $\boldsymbol{x}$, that is, the fraction of light that transmits straight between $\boldsymbol{x}'$ and $\boldsymbol{x}$ without being absorbed or scattered away:

$$T(\boldsymbol{x}', \boldsymbol{x}) = \exp\left(-\int_0^{\tau} \sigma_{\text{t}}(\boldsymbol{x} - \tau'\boldsymbol{\omega})\,\mathrm{d}\tau'\right), \tag{4}$$

with σ_{t} denoting the medium's *extinction coefficient*.

The *collision operator* $\mathcal{K}_{\text{C}}$ maps the interior radiance field L to[1]

$$(\mathcal{K}_{\text{C}}\,L)(\boldsymbol{x}, \boldsymbol{\omega}) = \sigma_{\text{s}}(\boldsymbol{x}) \underbrace{\int_{\mathbb{S}^2} f_{\text{p}}(\boldsymbol{x}, -\boldsymbol{\omega}', \boldsymbol{\omega})\, L(\boldsymbol{x}, \boldsymbol{\omega}')\,\mathrm{d}\boldsymbol{\omega}'}_{=:\, L^{\text{ins}}(\boldsymbol{x}, \boldsymbol{\omega})}, \tag{5}$$

where σ_{s} and f_{p} denote the medium's *scattering coefficient* and *single-scattering phase function*, respectively. L^{ins} is usually termed as the *in-scattered* radiance.[2]

The *source term* Q of the RTE (1) is sometimes assumed given for problems with simple geometries (hence the name). Generally, with the presence of nontrivial interfaces, Q equals[3]

$$Q(\boldsymbol{x}, \boldsymbol{\omega}) = \underbrace{\int_0^D T(\boldsymbol{x}', \boldsymbol{x})\, \sigma_{\text{a}}(\boldsymbol{x}')\, L^{\text{e}}(\boldsymbol{x}', \boldsymbol{\omega})\,\mathrm{d}\tau}_{=\, (\mathcal{K}_{\text{T}}\,\sigma_{\text{a}}\,L^{\text{e}})(\boldsymbol{x}, \boldsymbol{\omega})} + T(\boldsymbol{x}_0, \boldsymbol{x})\, L_{\text{s}}(\boldsymbol{x}_0, \boldsymbol{\omega}), \tag{6}$$

where $\boldsymbol{x}_0 := \boldsymbol{x} - D\boldsymbol{\omega}$ is a point on the medium's boundary, and $\sigma_{\text{a}} := \sigma_{\text{t}} - \sigma_{\text{s}}$ is the *absorption coefficient*. On the right-hand side (RHS) of this equation, L^{e} represents the medium's *radiant emission*, and L_{s} indicates the *interfacial radiance* governed by the *rendering equation* (RE) for all $\boldsymbol{x} \in \partial\Omega$ and $\boldsymbol{\omega} \in \mathbb{S}^2$:

$$L_{\text{s}}(\boldsymbol{x}, \boldsymbol{\omega}) = \underbrace{\int_{\mathbb{S}^2} f_{\text{s}}(\boldsymbol{x}, -\boldsymbol{\omega}', \boldsymbol{\omega})\, L(\boldsymbol{x}, \boldsymbol{\omega}')\,\mathrm{d}\boldsymbol{\omega}'}_{=:\, L_{\text{s}}^{\text{i}}(\boldsymbol{x}, \boldsymbol{\omega})} + L_{\text{s}}^{\text{e}}(\boldsymbol{x}, \boldsymbol{\omega}), \tag{7}$$

where f_{s} is the *cosine-weighted BSDF*, L_{s}^{e} denotes the *interfacially emitted radiance*, and L_{s}^{r} indicates the *reflected/refracted radiance*.

Combining the RTE and the RE. The source term Q of the RTE (1) can also be expressed compactly using operators. Substituting Eq. (7) into Eq. (6) and regrouping the resulting terms, we have

$$Q = \mathcal{K}_{\text{S}}\,L + L^{(0)}, \tag{8}$$

where the *interfacial scattering operator* $\mathcal{K}_{\text{S}}$ is defined as

$$\begin{aligned}(\mathcal{K}_{\text{S}}\,L)(\boldsymbol{x}, \boldsymbol{\omega}) &= T(\boldsymbol{x}_0, \boldsymbol{x})\, L_{\text{s}}^{\text{r}}(\boldsymbol{x}_0, \boldsymbol{\omega}) \\ &= T(\boldsymbol{x}_0, \boldsymbol{x}) \int_{\mathbb{S}^2} f_{\text{s}}(\boldsymbol{x}_0, -\boldsymbol{\omega}', \boldsymbol{\omega})\, L(\boldsymbol{x}_0, \boldsymbol{\omega}')\,\mathrm{d}\boldsymbol{\omega}',\end{aligned} \tag{9}$$

[1] In this paper, we follow the convention that all directions point away from $\boldsymbol{x}$ when expressing BSDFs and phase functions, yielding the negative sign before $\boldsymbol{\omega}'$ in Eq. (5).
[2] Precisely, L^{ins} captures the in-scattered radiance before volumetric absorption.
[3] In Eq. (6), the absorption coefficient σ_{a} and radiant emission L^{e} are sometimes combined into a single volumetric source term. Here, we follow the work by Kutz et al. [2017] and separate the two terms. Additionally, when Ω is unbounded, D can be infinite for certain $\boldsymbol{x}$ and $\boldsymbol{\omega}$, causing the second term on the RHS to vanish.

which accounts for contributed radiance due to reflection and transmission at the medium's boundary. The second term in Eq. (8) is

$$L^{(0)}(\boldsymbol{x}, \boldsymbol{\omega}) := (\mathcal{K}_{\mathrm{T}}\, \sigma_{\mathrm{a}}\, L^{\mathrm{e}})(\boldsymbol{x}, \boldsymbol{\omega}) + T(\boldsymbol{x}_0, \boldsymbol{x})\, L^{\mathrm{e}}_{\mathrm{s}}(\boldsymbol{x}_0, \boldsymbol{\omega}), \tag{10}$$

indicating the contribution of radiant emission in the medium and from its boundary.

Based on Eqs. (8–10), the RTE (1) and the RE (7) can be combined into an integral equation on L with an operator form:

$$L = (\mathcal{K}_{\mathrm{T}}\, \mathcal{K}_{\mathrm{C}} + \mathcal{K}_{\mathrm{S}})\, L + L^{(0)}. \tag{11}$$

Our core theoretical contribution is to derive expressions for the derivatives of the interior radiance L by differentiating individual terms in this equation.

3.2 Scene Derivatives

We assume that the scene specifications (such as object geometries and material properties) are parameterized by a set of m parameters $\boldsymbol{\pi} = \{\pi_1, \pi_2, \ldots, \pi_m\}$, and each parameter varies continuously. The medium's domain and its boundary may also vary with these parameters. We therefore denote them as $\Omega(\boldsymbol{\pi})$ and $\partial\Omega(\boldsymbol{\pi})$, respectively.

To give a few examples, the translation of an object in 3D along a direction $\hat{\boldsymbol{\omega}}$ can be parameterized by $\boldsymbol{y}(\pi) = \boldsymbol{y} + \pi\hat{\boldsymbol{\omega}}$ for each point $\boldsymbol{y}$ of the object. The radiance field L depends on the object position and subsequently the parameter π. Further, any linear transformed version of the object can be expressed as $\boldsymbol{y}(\pi) = M(\pi)\,\boldsymbol{y}$, where $M \in \mathbb{R}^{3\times3}$ is a transformation matrix of parameter π.

Our problem. Provided a specific scene parameterization, we aim to develop a theory for computing the total derivative (or gradient) of the interior radiance L with respect to $\boldsymbol{\pi}$. We denote the total derivative as $\mathrm{d}_{\boldsymbol{\pi}} L \in \mathbb{R}^m$ and refer it as the *scene derivative* of L.

Let $L(\boldsymbol{x}, \boldsymbol{\omega}; \boldsymbol{\pi})$ be the interior radiance L at $\boldsymbol{x} \in \Omega(\boldsymbol{\pi}) \setminus \partial\Omega(\boldsymbol{\pi})$ with the direction $\boldsymbol{\omega} \in \mathbb{S}^2$. The scene derivative, $\mathrm{d}_{\boldsymbol{\pi}} L(\boldsymbol{x}(\boldsymbol{\pi}), \boldsymbol{\omega}(\boldsymbol{\pi}); \boldsymbol{\pi})$, is an $m \times 1$ vector whose i-th element is the partial derivative of L with respect to the i-th parameter π_i. That is,

$$\begin{aligned}&\partial_{\pi_i} L(\boldsymbol{x}(\boldsymbol{\pi}), \boldsymbol{\omega}(\boldsymbol{\pi}); \boldsymbol{\pi}) \\ &\qquad = \lim_{\epsilon \to 0} \frac{L(\boldsymbol{x}(\boldsymbol{\pi}_i'), \boldsymbol{\omega}(\boldsymbol{\pi}_i'); \boldsymbol{\pi}_i') - L(\boldsymbol{x}(\boldsymbol{\pi}), \boldsymbol{\omega}(\boldsymbol{\pi}); \boldsymbol{\pi})}{\epsilon},\end{aligned} \tag{12}$$

where $\boldsymbol{\pi}_i' := \{\pi_1, \ldots, \pi_{i-1}, \pi_i + \epsilon, \pi_{i+1}, \ldots, \pi_m\}$. We note that it is crucial to allow both $\boldsymbol{x}$ and $\boldsymbol{\omega}$ to depend on the scene parameters $\boldsymbol{\pi}$. As we will show later, this dependence emerges in a number of interesting applications such as (i) interfacial radiances L_{s} on evolving surfaces and (ii) radiometric measurements from moving sensors.

Without loss of generality, our derivation focuses on the partial derivative of L, denoted as $\dot{L} := \partial_\pi L$, with respect to one scene parameter $\pi \in \boldsymbol{\pi}$. For notational convenience, we omit the dependencies on $\boldsymbol{\pi}$ from $\boldsymbol{x}$, $\boldsymbol{\omega}$, and the radiances in the rest of the paper.

3.3 Differentiating Integrals

Deriving analytic expressions for the scene derivatives of radiances requires differentiating the integral operators in the RTE (11) and the RE (7). To this end, we use the Reynolds transport theorem [Leal 2007],[4] which originated in fluid mechanics and is a generalization of Leibniz's integral rule for differentiation [Flanders 1973].

Theorem 1: Reynolds transport theorem

Let f be a scalar-valued function defined on some n-dimensional manifold $\Omega(\pi)$ parameterized with some $\pi \in \mathbb{R}$. Additionally, let $\Gamma(\pi) \subset \Omega(\pi)$ be an $(n-1)$-dimensional manifold given by the union of the *external* boundary $\partial\Omega(\pi)$ and the *internal* one containing the discontinuous locations of f. Then, it holds that

$$\partial_\pi \left(\int_{\Omega(\pi)} f \,\mathrm{d}\Omega(\pi) \right) = \int_{\Omega(\pi)} \dot{f} \,\mathrm{d}\Omega(\pi) + \int_{\Gamma(\pi)} \langle \boldsymbol{n}, \dot{\boldsymbol{x}} \rangle\, \Delta f \,\mathrm{d}\Gamma(\pi), \tag{13}$$

where $\dot{f} := \partial_\pi f$, $\dot{\boldsymbol{x}} := \partial_\pi \boldsymbol{x}$, $\mathrm{d}\Omega$ and $\mathrm{d}\Gamma$ respectively denote the standard measures associated with Ω and Γ; and $\langle \cdot, \cdot \rangle$ indicates the dot (inner) product between two vectors. Further, $\boldsymbol{n}$ is the normal direction at each $\boldsymbol{x} \in \Gamma(\pi)$, and Δf is given by[a]

$$\Delta f(\boldsymbol{x}) := \lim_{\epsilon \to 0^-} f(\boldsymbol{x} + \epsilon\, \boldsymbol{n}) - \lim_{\epsilon \to 0^+} f(\boldsymbol{x} + \epsilon\, \boldsymbol{n}), \tag{14}$$

for all $\boldsymbol{x} \in \Gamma(\pi)$.

[a] When $\boldsymbol{x}$ approaches $\Gamma(\pi)$ from the exterior of the integral domain $\Omega(\pi)$, the corresponding one-sided limit in Eq. (14) is set to zero.

Intuitively, Theorem 1 states that the derivative of the integral of a continuous function f over some domain $\Omega(\pi)$ depends on (i) the derivative of f itself over $\Omega(\pi)$ (the first integral); and (ii) the "normal velocity" of the domain boundary and internal discontinuities $\Gamma(\pi)$ modulated by the differences between function values across these boundaries (the second integral)[5].

In a special case, for classical Riemann integrals where Ω is an interval $(a, b) \subset \mathbb{R}$ and the integrand f is differentiable everywhere, it holds that $\Gamma = \{a, b\}$, and the interface integral in Eq. (13) reduces to the sum of the integrand evaluated at a and b. Let $\boldsymbol{n} = 1$, then $\langle \boldsymbol{n}, \dot{\boldsymbol{x}} \rangle$ equals $\dot{a}$ at $x = a$ and $\dot{b}$ at $x = b$. Further, it is easy to verify that $\Delta f(b(\pi), \pi) = f(b(\pi), \pi)$ and $\Delta f(a(\pi), \pi) = -f(a(\pi), \pi)$, yielding

$$\begin{aligned}\partial_\pi \textstyle\int_{a(\pi)}^{b(\pi)} f(x, \pi) \,\mathrm{d}x = &\textstyle\int_{a(\pi)}^{b(\pi)} \dot{f}(x, \pi) \,\mathrm{d}x \\ &+ \dot{b}(\pi)\, f(b(\pi), \pi) - \dot{a}(\pi)\, f(a(\pi), \pi),\end{aligned} \tag{15}$$

which is precisely Leibniz's rule for differentiation.

In the rest of this paper, we detail how Theorem 1 can be used to calculate scene derivatives of radiances (§4), and introduce an unbiased Monte Carlo method for estimating those derivatives (§5).

4 DIFFERENTIABLE RADIATIVE TRANSFER

We now present our differential theory of radiative transfer that shows how the interior radiance L can be differentiated with respect to some scene parameter $\pi \in \mathbb{R}$. To this end, we derive the partial derivative $\dot{L} := \partial_\pi L$ by differentiating each of the operators on the right-hand side (RHS) of Eq. (11).

[4] We present as Theorem 1 a slightly generalized version of the original theorem.
[5] Previously, Li et al. [2018a] used a different but mathematically equivalent method to demonstrate the need of the boundary integral. This equivalence was further discussed in his Ph.D. thesis [2019].

Outline of Our Derivations

$$\dot{L} = \underbrace{\partial_\pi \mathcal{K}_\mathrm{T}\, \mathcal{K}_\mathrm{C}\, L}_{\text{§4.1, §4.2}} + \underbrace{\partial_\pi \mathcal{K}_\mathrm{S}\, L}_{\text{§4.3}} + \underbrace{\dot{L}^{(0)}}_{\text{§4.4}}.$$

Fig. 2. An **outline of our derivations** in §4. To derive $\dot{L}$, we differentiate the transport and collision operators $\mathcal{K}_\mathrm{T}$ and $\mathcal{K}_\mathrm{C}$ in §4.1, the interfacial scattering operator $\mathcal{K}_\mathrm{S}$ in §4.3, and the source term $L^{(0)}$ in §4.4.

Assumptions. As most participating media and translucent materials are non-emissive, we neglect the volumetric emission term L^e in Eqs. (6) and (10) when deriving $\dot{L}$ in this section. Additionally, we assume that: (i) the RTE and RE parameters σ_t, σ_s, f_p, L^e, f_s, and L^e_s are *continuous* spatially and directionally; and (ii) there are no *zero-measure* light sources (e.g., point and directional) or *ideal specular surfaces* (e.g., perfect mirrors). We generalize our derivations by relaxing some of these assumptions in the supplement.

Overview. Figure 2 outlines the structure of our derivations. As a preview, based on the assumptions above, the scene partial derivative $\dot{L}$ will take the form of Eq. (32) with dependencies between the involved quantities summarized in Figure 7.

4.1 Differentiation of the Transport & Collision Operators

We start with the first term on the RHS in Figure 2, $\partial_\pi \mathcal{K}_\mathrm{T}\, \mathcal{K}_\mathrm{C}\, L$. As we derive in Appendix A.1, this derivative comprises the terms:

$$\begin{aligned}(\partial_\pi \mathcal{K}_\mathrm{T}\, \mathcal{K}_\mathrm{C}\, L)(\boldsymbol{x}, \boldsymbol{\omega}) &= \int_0^D T(\boldsymbol{x}', \boldsymbol{x})\, \sigma_\mathrm{s}(\boldsymbol{x}')\, \dot{L}^\mathrm{ins}(\boldsymbol{x}', \boldsymbol{\omega})\, \mathrm{d}\tau \\ &+ \int_0^D T(\boldsymbol{x}', \boldsymbol{x})\, [\dot{\sigma}_\mathrm{s}(\boldsymbol{x}') - \Sigma_\mathrm{t}(\boldsymbol{x}, \boldsymbol{\omega}, \tau)\, \sigma_\mathrm{s}(\boldsymbol{x}')]\, L^\mathrm{ins}(\boldsymbol{x}', \boldsymbol{\omega})\, \mathrm{d}\tau \\ &+ \dot{D}\, T(\boldsymbol{x}_0, \boldsymbol{x})\, \sigma_\mathrm{s}(\boldsymbol{x}_0)\, L^\mathrm{ins}(\boldsymbol{x}_0, \boldsymbol{\omega}), \end{aligned} \tag{16}$$

where $\boldsymbol{x}' := \boldsymbol{x} - \tau\boldsymbol{\omega}$ is a point on the ray $(\boldsymbol{x}, -\boldsymbol{\omega})$; $\boldsymbol{x}_0 := \boldsymbol{x} - D\boldsymbol{\omega}$ is the location where the ray intersects the medium boundary; L^ins is the in-scattered radiance from Eq. (5); and $\Sigma_\mathrm{t}(\boldsymbol{x}, \boldsymbol{\omega}, \tau)$, defined as

$$\Sigma_\mathrm{t}(\boldsymbol{x}, \boldsymbol{\omega}, \tau) := \int_0^\tau \dot{\sigma}_\mathrm{t}(\boldsymbol{x} - \tau'\boldsymbol{\omega})\, \mathrm{d}\tau', \tag{17}$$

appears when we differentiate the transmittance $T(\boldsymbol{x}', \boldsymbol{x})$ (see Appendix A.2).

In Eqs. (16) and (17), $\dot{\sigma}_\mathrm{t}$ and $\dot{\sigma}_\mathrm{s}$ are essentially *material derivatives* given by [Leal 2007]

$$\dot{\sigma}(\boldsymbol{x}) = \frac{\partial \sigma}{\partial \pi}(\boldsymbol{x}) + \langle \dot{\boldsymbol{x}}, \nabla \sigma(\boldsymbol{x}) \rangle, \tag{18}$$

for $\sigma \in \{\sigma_\mathrm{t}, \sigma_\mathrm{s}\}$ with ∇ being the gradient operator.

Evaluating the RHS of Eq. (16) also requires $\dot{D}$, $L^\mathrm{ins}(\boldsymbol{x}_0, \boldsymbol{\omega})$, and $\dot{L}^\mathrm{ins}(\boldsymbol{x}', \boldsymbol{\omega})$. In what follows, we derive formulas for these terms.

Derivation of $\dot{D}$. In the transport operator of Eq. (2), the upper-bound of the line integral D is determined by the distance a light ray originating at $\boldsymbol{x}$ with direction $-\boldsymbol{\omega}$ travels before intersecting the medium boundary $\partial\Omega$. Suppose that the boundary $\partial\Omega(\pi)$ has an implicit representation $F(\boldsymbol{y}; \pi) = 0$ parameterized by π. Then, substituting $\boldsymbol{y}$ with the ray equation yields $F(\boldsymbol{x} - D\boldsymbol{\omega}; \pi) = 0$ and $D = F^{-1}(0; \boldsymbol{x}, \boldsymbol{\omega}, \pi)$, with F^{-1} being the inverse function of F with respect to D. Then, $\dot{D} = \partial_\pi F^{-1}(0; \boldsymbol{x}, \boldsymbol{\omega}, \pi)$.

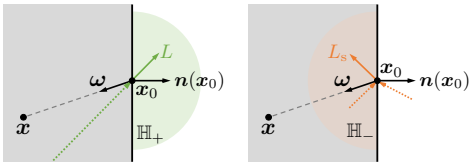

Fig. 3. Calculating the in-scattered radiance L^ins at location $\boldsymbol{x}_0 \in \partial\Omega$ with direction $\boldsymbol{\omega}$ pointing toward the interior of the medium (illustrated in gray). (a) When $\boldsymbol{\omega}' \in \mathbb{H}_+$ (i.e., pointing toward the exterior of the medium), the interior radiance $L(\boldsymbol{x}_0, \boldsymbol{\omega}')$ involving a line integral (indicated as the dashed line in green) from the interior is used. (b) When $\boldsymbol{\omega}' \in \mathbb{H}_-$, on the contrary, the interfacial radiance $L_\mathrm{s}(\boldsymbol{x}_0, \boldsymbol{\omega}')$ from the interior is used. This radiance is in turn determined by interior radiances reflected and refracted by the interface (shown as dashed lines in orange).

For piecewise-linear shapes, such as polygonal meshes, it holds that $F(\boldsymbol{x} - D\boldsymbol{\omega}; \pi) = \langle \boldsymbol{n}^\mathrm{face},\ \boldsymbol{x} - D\boldsymbol{\omega} - \boldsymbol{p}^\mathrm{face} \rangle$, where $\boldsymbol{n}^\mathrm{face}$ is the normal of the face containing the intersection point, and $\boldsymbol{p}^\mathrm{face}$ is an arbitrary point (e.g., a vertex) on that face. Note that both $\boldsymbol{n}^\mathrm{face}$ and $\boldsymbol{p}^\mathrm{face}$ can depend on the scene parameter π. Therefore,

$$\dot{D} = \partial_\pi F^{-1}(0; \boldsymbol{x}, \boldsymbol{\omega}, \pi) = \partial_\pi \frac{\langle \boldsymbol{n}^\mathrm{face}, \boldsymbol{x} - \boldsymbol{p}^\mathrm{face} \rangle}{\langle \boldsymbol{n}^\mathrm{face}, \boldsymbol{\omega} \rangle}. \tag{19}$$

In practice, $\dot{D}$ can also be obtained by differentiating the ray tracing process using *automated differentiation*.

In-scattered radiance at the boundary. The last term required for evaluating $\partial_\pi \mathcal{K}_\mathrm{T}\, \mathcal{K}_\mathrm{C}\, L$ in Eq. (16) involves the in-scattered radiance L^ins at boundary point $\boldsymbol{x}_0$. Caution is needed here since light transport behaves differently at the two sides of the boundary.

$L^\mathrm{ins}(\boldsymbol{x}_0, \boldsymbol{\omega})$ is given by the limit of $L^\mathrm{ins}(\boldsymbol{x}', \boldsymbol{\omega})$ as $\boldsymbol{x}'$ approaches the boundary location $\boldsymbol{x}_0$ from the interior of the medium along the direction of $-\boldsymbol{\omega}$. Namely, $L^\mathrm{ins}(\boldsymbol{x}_0, \boldsymbol{\omega}) = \lim_{\tau \to D^-} L^\mathrm{ins}(\boldsymbol{x}', \boldsymbol{\omega})$ with $\boldsymbol{x}' := \boldsymbol{x} - \tau\boldsymbol{\omega}$. This limit can be further expressed as

$$\begin{aligned} L^\mathrm{ins}(\boldsymbol{x}_0, \boldsymbol{\omega}) = &\int_{\mathbb{H}_+} f_\mathrm{p}(\boldsymbol{x}_0, -\boldsymbol{\omega}', \boldsymbol{\omega})\, L(\boldsymbol{x}_0, \boldsymbol{\omega}')\, \mathrm{d}\boldsymbol{\omega}' + \\ &\int_{\mathbb{H}_-} f_\mathrm{p}(\boldsymbol{x}_0, -\boldsymbol{\omega}', \boldsymbol{\omega})\, L_\mathrm{s}(\boldsymbol{x}_0, \boldsymbol{\omega}')\, \mathrm{d}\boldsymbol{\omega}', \end{aligned} \tag{20}$$

where $\mathbb{H}_+$ and $\mathbb{H}_-$ are the two hemispheres separated by the boundary at $\boldsymbol{x}_0$ (see Figure 3) given by

$$\begin{aligned} \mathbb{H}_+ &= \{\boldsymbol{\omega}' \in \mathbb{S}^2 \ :\ \langle \boldsymbol{n}(\boldsymbol{x}_0), \boldsymbol{\omega}' \rangle > 0\}, \\ \mathbb{H}_- &= \{\boldsymbol{\omega}' \in \mathbb{S}^2 \ :\ \langle \boldsymbol{n}(\boldsymbol{x}_0), \boldsymbol{\omega}' \rangle < 0\}, \end{aligned} \tag{21}$$

where $\boldsymbol{n}(\boldsymbol{x}_0)$ is the boundary normal pointing toward the exterior of the medium (i.e., $\langle \boldsymbol{n}(\boldsymbol{x}_0), \boldsymbol{\omega} \rangle < 0$).

4.2 Differentiation of the In-Scattered Radiance

Evaluating the RHS of Eq. (16) also requires the derivative of the in-scattered radiance L^ins of Eq. (5). Recall that L^ins is expressed as an integral of $f_\mathrm{p}(\boldsymbol{x}, -\boldsymbol{\omega}', \boldsymbol{\omega})\, L(\boldsymbol{x}, \boldsymbol{\omega}')$ over directions $\boldsymbol{\omega}'$. Note that $L(\boldsymbol{x}, \boldsymbol{\omega}')$ may have discontinuities in $\boldsymbol{\omega}'$ (for fixed $\boldsymbol{x}$) due to visibility changes. Thus, to differentiate L^ins, we must consider how the discontinuities change with respect to the scene parameter π.

Let $\mathbb{S}(\boldsymbol{x}) \subset \mathbb{S}^2$ be a set of *spherical curves* capturing all discontinuities of $f_\mathrm{p}(\boldsymbol{x}, -\boldsymbol{\omega}', \boldsymbol{\omega})\, L(\boldsymbol{x}, \boldsymbol{\omega}')$ with respect to $\boldsymbol{\omega}'$ (see the red curves

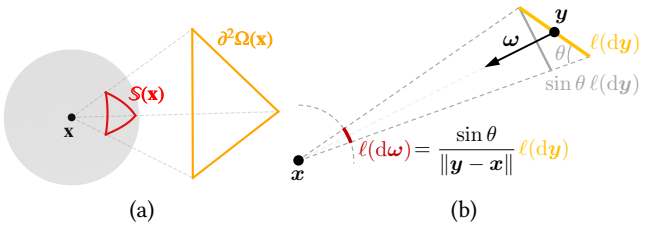

Fig. 4. (a) Definitions of $\mathbb{S}(\boldsymbol{x})$ and $\partial^2\Omega(\boldsymbol{x})$. (b) Deriving the change-of-measure ratio $\sin\theta/\|\boldsymbol{y}-\boldsymbol{x}\|$ in Eq. (23) by projecting a differential curve $\mathrm{d}\boldsymbol{y}$ to the surface of a unit sphere around $\boldsymbol{x}$.

in Figure 4). At an interior point $\boldsymbol{x} \in \Omega \setminus \partial\Omega$ and direction $\boldsymbol{\omega} \in \mathbb{S}^2$, the derivative of L^{ins} can be expressed using Theorem 1 as

$$\dot{L}^{\text{ins}}(\boldsymbol{x},\boldsymbol{\omega}) = \int_{\mathbb{S}^2} \partial_\pi \left[f_{\text{p}}(\boldsymbol{x},-\boldsymbol{\omega}',\boldsymbol{\omega})\, L(\boldsymbol{x},\boldsymbol{\omega}') \right] \mathrm{d}\boldsymbol{\omega}' + \underbrace{\int_{\mathbb{S}(\boldsymbol{x})} \langle \boldsymbol{n}_\perp, \dot{\boldsymbol{\omega}}' \rangle\, f_{\text{p}}(\boldsymbol{x},-\boldsymbol{\omega}',\boldsymbol{\omega})\, \Delta L(\boldsymbol{x},\boldsymbol{\omega}')\, \mathrm{d}\ell(\boldsymbol{\omega}')}_{=:\, B^{\text{ins}}(\boldsymbol{x},\boldsymbol{\omega})}. \tag{22}$$

The first term on the RHS of this equation is an integral over the unit sphere $\mathbb{S}^2$, which is independent of the scene parameter π, making the integral variable $\boldsymbol{\omega}'$ also be π-independent. To evaluate this term, the derivative of the phase function f_{p} is needed. In practice, f_{p} usually has analytical expressions, allowing $\dot{f}_p(\boldsymbol{x},-\boldsymbol{\omega}',\boldsymbol{\omega})$ to be obtained using symbolic differentiation.

The second term in Eq. (22) arises from the boundary integral term in Eq. (13) after Theorem 1 is applied. It emerges due to the discontinuities of $f_{\text{p}}(\boldsymbol{x},-\boldsymbol{\omega}',\boldsymbol{\omega})\, L(\boldsymbol{x},\boldsymbol{\omega}')$ with respect to $\boldsymbol{\omega}'$, corresponding to how the discontinuities "move" as the scene parameter π varies. We denote this integral as $B^{\text{ins}}(\boldsymbol{x},\boldsymbol{\omega})$, wherein $\dot{\boldsymbol{\omega}}'$ is the change rate (with respect to π) of the discontinuity location $\boldsymbol{\omega}'$, and $\ell(\boldsymbol{\omega}')$ indicates the curve length measure.

In Eq. (22), $\boldsymbol{n}_\perp$ is a vector in the tangent space of $\mathbb{S}^2$ at $\boldsymbol{\omega}'$ perpendicular to the discontinuity curve (see Figure 6). Further, ΔL is the radiance difference across a discontinuity curve in $\mathbb{S}(\boldsymbol{x})$. We discuss both terms in more detail later in this subsection.

While B^{ins} integrates over a set of spherical curves on $\mathbb{S}^2$, it is computationally more convenient to rewrite this integral in terms of boundary curves in the 3D space. As shown in Figure 5, the 3D boundary curves comprise all the *geometric edges* that cause discontinuities of $L(\boldsymbol{x},\boldsymbol{\omega}')$ in $\boldsymbol{\omega}'$ when viewed from $\boldsymbol{x}$, including (i) *boundary* edges associated with only one face; (ii) *silhouette* edges shared by a front-facing and a back-facing face; and (iii) front-facing *sharp* edges across which the surface normals are discontinuous.

Let $\partial^2\Omega(\boldsymbol{x}) \subset \partial\Omega$ denote all the boundary curves when viewed from $\boldsymbol{x}$. By changing the measure of curve length from $\mathbb{S}^2$ to the 3D Euclidean space, we rewrite the last term $B^{\text{ins}}(\boldsymbol{x},\boldsymbol{\omega})$ in Eq. (22) as

$$B^{\text{ins}}(\boldsymbol{x},\boldsymbol{\omega}) = \int_{\partial^2\Omega(\boldsymbol{x})} \left\langle \boldsymbol{n}_\perp, \partial_\pi(\boldsymbol{y}\to\boldsymbol{x}) \right\rangle f_{\text{p}}(\boldsymbol{x},\boldsymbol{x}\to\boldsymbol{y},\boldsymbol{\omega}) \\ \Delta L(\boldsymbol{x},\boldsymbol{y}\to\boldsymbol{x})\, V(\boldsymbol{x},\boldsymbol{y})\, \frac{\sin\theta}{\|\boldsymbol{y}-\boldsymbol{x}\|}\, \mathrm{d}\ell(\boldsymbol{y}), \tag{23}$$

where $\boldsymbol{y}\to\boldsymbol{x}$ indicates the normalized direction from $\boldsymbol{y}$ to $\boldsymbol{x}$, $\partial_\pi(\boldsymbol{y}\to\boldsymbol{x})$ is the same as $\dot{\boldsymbol{\omega}}'$, $V(\boldsymbol{x},\boldsymbol{y})$ denotes the mutual visibility between $\boldsymbol{x}$ and $\boldsymbol{y}$, and θ is the angle between the tangent direction

Fig. 5. Three types of edges (drawn in yellow) that can cause geometric discontinuities: (a) boundary, (b) silhouette, and (c) sharp.

at $\boldsymbol{y}$ and $\boldsymbol{y}\to\boldsymbol{x}$ (see Figure 4-b). As a result, $\ell(\boldsymbol{y})$ here is the length measure in Euclidean space.

Lastly, we discuss how to compute the ingredients needed for evaluating Eq. (22), namely, $\boldsymbol{n}_\perp$, $\dot{\boldsymbol{\omega}}' = \partial_\pi(\boldsymbol{y}\to\boldsymbol{x})$, and ΔL.

Normal. The normal $\boldsymbol{n}_\perp$ in Eqs. (22) and (23) emerges from the application of Theorem 1 and represents the normal vector of the discontinuity boundary on the integration manifold (i.e., $\mathbb{S}^2$). Therefore, $\boldsymbol{n}_\perp$ must be in the tangent space of $\boldsymbol{\omega}' \in \mathbb{S}^2$.

In practice, when the scene geometry is depicted using polygonal meshes, the boundary curves $\partial^2\Omega(\boldsymbol{x})$ in Eq. (23) are comprised of polygonal face edges. Consider an edge with endpoints $\boldsymbol{p}$ and $\boldsymbol{q}$. Its projection on a unit sphere centered at $\boldsymbol{x}$ is an arc on which every point has the same normal in the tangent space (see Figure 6-a):

$$\boldsymbol{n}_\perp = \frac{(\boldsymbol{p}-\boldsymbol{x})\times(\boldsymbol{q}-\boldsymbol{x})}{\|(\boldsymbol{p}-\boldsymbol{x})\times(\boldsymbol{q}-\boldsymbol{x})\|}. \tag{24}$$

Change rate of $\boldsymbol{\omega}'$. Boundary curves in $\partial^2\Omega(\boldsymbol{x})$ may vary with respect to the scene parameter π—for example, an object may move in 3D along a trajectory parameterized by π. Suppose that a point $\boldsymbol{y} \in \partial^2\Omega(\boldsymbol{x})$ has a change rate $\dot{\boldsymbol{y}}$. Then, the corresponding direction $\boldsymbol{\omega}' = (\boldsymbol{y}\to\boldsymbol{x})$ has the derivative (which we derive in Appendix A.3):

$$\dot{\boldsymbol{\omega}}' = \partial_\pi\left(\frac{\boldsymbol{x}-\boldsymbol{y}}{\|\boldsymbol{x}-\boldsymbol{y}\|}\right) = \frac{\dot{\boldsymbol{x}}-\dot{\boldsymbol{y}}}{\|\boldsymbol{x}-\boldsymbol{y}\|} - \boldsymbol{\omega}'\left\langle \boldsymbol{\omega}', \frac{\dot{\boldsymbol{x}}-\dot{\boldsymbol{y}}}{\|\boldsymbol{x}-\boldsymbol{y}\|}\right\rangle. \tag{25}$$

Evaluating ΔL. Recall that, in Eq. (22), $\Delta L(\boldsymbol{x},\boldsymbol{\omega}')$ indicates the difference in radiance across discontinuity boundaries with respect to $\boldsymbol{\omega}'$. This is given by the difference between two one-sided limits of $\Delta L(\boldsymbol{x},\hat{\boldsymbol{\omega}})$ when $\hat{\boldsymbol{\omega}}$ approaches the discontinuity boundary $\boldsymbol{\omega}'$ from both sides along the normal direction $\boldsymbol{n}_\perp$. Namely,

$$\Delta L(\boldsymbol{x},\boldsymbol{\omega}') = \lim_{\epsilon\to 0^-} L(\boldsymbol{x},\boldsymbol{\omega}'+\epsilon\,\boldsymbol{n}_\perp) - \lim_{\epsilon\to 0^+} L(\boldsymbol{x},\boldsymbol{\omega}'+\epsilon\,\boldsymbol{n}_\perp). \tag{26}$$

Based on this definition, we evaluate ΔL in Appendix A.4.

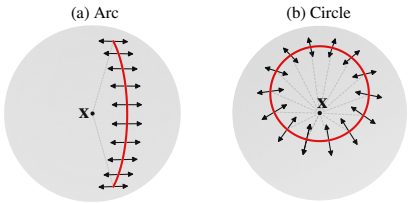

Fig. 6. The normal directions of arcs and circles (that are respectively the projections of line segments and spheres) as spherical curves.

4.3 Differentiation of the Interfacial Scattering Operator

We now move on to the second term in Figure 2, the derivative of $\mathcal{K}_\mathrm{S}\,L$. Differentiating $(\mathcal{K}_\mathrm{S}\,L)(\boldsymbol{x},\boldsymbol{\omega}) = T(\boldsymbol{x}_0,\boldsymbol{\omega})\,L^\mathrm{r}_\mathrm{s}(\boldsymbol{x}_0,\boldsymbol{\omega})$ yields

$$\begin{aligned}&(\partial_\pi \mathcal{K}_\mathrm{S}\,L)(\boldsymbol{x},\boldsymbol{\omega}) = \partial_\pi[T(\boldsymbol{x}_0,\boldsymbol{x})\,L^\mathrm{r}_\mathrm{s}(\boldsymbol{x}_0,\boldsymbol{x})]\\ &= T(\boldsymbol{x}_0,\boldsymbol{x})\,\dot{L}^\mathrm{r}_\mathrm{s}(\boldsymbol{x}_0,\boldsymbol{\omega}) + \dot{T}(\boldsymbol{x}_0,\boldsymbol{x})\,L^\mathrm{r}_\mathrm{s}(\boldsymbol{x}_0,\boldsymbol{\omega})\\ &= T(\boldsymbol{x}_0,\boldsymbol{x})\left[-\left(\Sigma_\mathrm{t}(\boldsymbol{x},\boldsymbol{\omega},D) + \dot{D}\,\sigma_\mathrm{t}(\boldsymbol{x}_0)\right)L^\mathrm{r}_\mathrm{s}(\boldsymbol{x}_0,\boldsymbol{\omega}) + \dot{L}^\mathrm{r}_\mathrm{s}(\boldsymbol{x}_0,\boldsymbol{\omega})\right].\end{aligned} \tag{27}$$

The expression of $\dot{T}(\boldsymbol{x}_0,\boldsymbol{x})$ is derived in Appendix A.2 and equals:

$$\dot{T}(\boldsymbol{x}_0,\boldsymbol{x}) = -T(\boldsymbol{x}_0,\boldsymbol{x})\left(\Sigma_\mathrm{t}(\boldsymbol{x},\boldsymbol{\omega},D) + \dot{D}\,\sigma_\mathrm{t}(\boldsymbol{x}_0)\right), \tag{28}$$

where Σ_t follows the definition in Eq. (17) but with the distance variable τ replaced by D. The last term $\dot{D}\,\sigma_\mathrm{t}(\boldsymbol{x}_0)$ appears because the total travel distance D in the medium depends on the medium's boundary that might change under π.

The last component needed in Eq. (27) is $\dot{L}^\mathrm{r}_\mathrm{s}(\boldsymbol{x}_0,\boldsymbol{\omega})$, whose specific form is analogous to $\dot{L}^\mathrm{ins}$ from Eq. (22) since both L^ins and L^r_s involve integrations over $\mathbb{S}^2$ (with the difference that L^r_s modulates incident radiance with the cosine-weighted BSDF f_s, while L^ins uses the phase function f_p). Thus, $\dot{L}^\mathrm{r}_\mathrm{s}$ takes the form expressed in Eq. (29):

$$\begin{aligned}\dot{L}^\mathrm{r}_\mathrm{s}(\boldsymbol{x},\boldsymbol{\omega}) = &\int_{\mathbb{S}^2}\partial_\pi\left[f_\mathrm{s}(\boldsymbol{x},-\boldsymbol{\omega}',\boldsymbol{\omega})\,L(\boldsymbol{x},\boldsymbol{\omega}')\right]\mathrm{d}\boldsymbol{\omega}' +\\ &\int_{\partial^2\Omega(\boldsymbol{x})}\left\langle \boldsymbol{n}_\perp, \partial_\pi(\boldsymbol{y}\to\boldsymbol{x})\right\rangle f_\mathrm{s}(\boldsymbol{x},\boldsymbol{x}\to\boldsymbol{y},\boldsymbol{\omega})\\ &\Delta L(\boldsymbol{x},\boldsymbol{y}\to\boldsymbol{x})\,V(\boldsymbol{x},\boldsymbol{y})\,\tfrac{\sin\theta}{\|\boldsymbol{y}-\boldsymbol{x}\|}\,\mathrm{d}\ell(\boldsymbol{y}),\end{aligned} \tag{29}$$

where ΔL follows Eq. (26). Further, $\partial_\pi f_\mathrm{s}\,L$ in the first term depends on the derivative of the BSDF f_s which can generally be obtained using symbolic/automated differentiation.

4.4 Completing $\dot{L}$

The final term in Figure 2 is the derivative of $L^{(0)} = TL^\mathrm{e}_\mathrm{s}$. Since Eq. (27) has already provided $\partial_\pi TL^\mathrm{r}_\mathrm{s}$, $\dot{L}^{(0)}$ can be obtained by respectively replacing L^r_s and $\dot{L}^\mathrm{r}_\mathrm{s}$ in Eq. (27) with L^e_s and $\dot{L}^\mathrm{e}_\mathrm{s}$, yielding

$$\begin{aligned}&\dot{L}^{(0)}(\boldsymbol{x},\boldsymbol{\omega}) = \partial_\pi[T(\boldsymbol{x}_0,\boldsymbol{x})\,L^\mathrm{e}_\mathrm{s}(\boldsymbol{x}_0,\boldsymbol{\omega})]\\ &= T(\boldsymbol{x}_0,\boldsymbol{x})\left[-\left(\Sigma_\mathrm{t}(\boldsymbol{x},\boldsymbol{\omega},D) + \dot{D}\,\sigma_\mathrm{t}(\boldsymbol{x}_0)\right)L^\mathrm{e}_\mathrm{s}(\boldsymbol{x}_0,\boldsymbol{\omega}) + \dot{L}^\mathrm{e}_\mathrm{s}(\boldsymbol{x}_0,\boldsymbol{\omega})\right].\end{aligned} \tag{30}$$

Similar to $\dot{f}_p$ and $\dot{f}_s$, $\dot{L}^\mathrm{e}_\mathrm{s}(\boldsymbol{x}_0,\boldsymbol{\omega})$ can be obtained using symbolic differentiation.

Putting together. We now combine Eqs. (16, 27, 30) to get $\dot{L}$, the partial scene derivative of the interior radiance L.

Since $\dot{L}_\mathrm{s} = \dot{L}^\mathrm{r}_\mathrm{s} + \dot{L}^\mathrm{e}_\mathrm{s}$, adding Eq. (27) and (30) effectively yields the derivative of the source term Q:

$$\begin{aligned}&\dot{Q}(\boldsymbol{x},\boldsymbol{\omega}) = \left(\partial_\pi\,\mathcal{K}_\mathrm{S}\,L + \dot{L}^{(0)}\right)(\boldsymbol{x},\boldsymbol{\omega})\\ &= T(\boldsymbol{x}_0,\boldsymbol{x})\left[-\left(\Sigma_\mathrm{t}(\boldsymbol{x},\boldsymbol{\omega},D) + \dot{D}\,\sigma_\mathrm{t}(\boldsymbol{x}_0)\right)L_\mathrm{s}(\boldsymbol{x}_0,\boldsymbol{\omega}) + \dot{L}_\mathrm{s}(\boldsymbol{x}_0,\boldsymbol{\omega})\right].\end{aligned} \tag{31}$$

Adding Eq. (16) and (31) completes the derivation of $\dot{L}$, as shown in Eq. (32) below. The relations between the quantities on which $\dot{L}$ depends are visualized in Figure 7.

Scene derivative of the interior radiance L

$$\begin{aligned}\dot{L}(\boldsymbol{x},\boldsymbol{\omega}) = &\int_0^D T(\boldsymbol{x}',\boldsymbol{x})\left[\sigma_\mathrm{s}(\boldsymbol{x}')\,\dot{L}^\mathrm{ins}(\boldsymbol{x}',\boldsymbol{\omega}) + (\dot{\sigma}_\mathrm{s}(\boldsymbol{x}') - \Sigma_\mathrm{t}(\boldsymbol{x},\boldsymbol{\omega},\tau)\,\sigma_\mathrm{s}(\boldsymbol{x}'))\,L^\mathrm{ins}(\boldsymbol{x}',\boldsymbol{\omega})\right]\mathrm{d}\tau\\ &+ T(\boldsymbol{x}_0,\boldsymbol{x})\left[-\left(\Sigma_\mathrm{t}(\boldsymbol{x},\boldsymbol{\omega},D) + \dot{D}\,\sigma_\mathrm{t}(\boldsymbol{x}_0)\right)L_\mathrm{s}(\boldsymbol{x}_0,\boldsymbol{\omega}) + \dot{L}_\mathrm{s}(\boldsymbol{x}_0,\boldsymbol{\omega}) + \dot{D}\,\sigma_\mathrm{s}(\boldsymbol{x}_0)\,L^\mathrm{ins}(\boldsymbol{x}_0,\boldsymbol{\omega})\right],\end{aligned} \tag{32}$$

where Σ_t is defined in Eq. (17), $\dot{L}^\mathrm{ins}$ follows Eq. (22), and $\dot{L}_\mathrm{s} = \dot{L}^\mathrm{r}_\mathrm{s} + \dot{L}^\mathrm{e}_\mathrm{s}$ with $\dot{L}^\mathrm{r}_\mathrm{s}$ given by Eq. (29).

4.5 Summary and Discussion

We derived the (partial) scene derivative of the interior radiance L by differentiating individual components of the RTE (11) using Theorem 1. The key to the application of this theorem is to properly handle all the boundary integral terms which arise due to Eq. (13) and contribute significantly to the final derivatives, as shown in Figure 8 (middle vs. bottom).

When differentiating the transport operator $\mathcal{K}_\mathrm{T}$ that involves a line integral, as expressed in Eq. (16), this boundary term reduces to evaluating the integrand $\sigma_\mathrm{s}\,L^\mathrm{ins}$ at the boundary (Eq. (20)) multiplied by the change rate $\dot{D}$ of the boundary.

Besides the boundary term, according to Theorem 1, we also need to differentiate the integrand $\sigma_\mathrm{s}\,L^\mathrm{ins}$ itself. To this end, we in turn need the derivative of the in-scattered radiance L^ins that is defined in Eq. (5) and involves a spherical integral of interior radiance L modulated by the phase function f_p. The boundary term B^ins in this case takes the form of an integral over all 3D edges capable of creating geometric discontinuities, as expressed in Eq. (23).

The differentiation of the interfacial scattering operator $\mathcal{K}_\mathrm{S}$ of Eq. (9) requires the derivative of the reflected/refracted radiance L^r_s (Eq. (7)), as expressed in Eq. (27). This can be done in a similar manner as the calculation of $\dot{L}^\mathrm{ins}$ in Eq. (29), given that both of them comprise spherical integrals of the interior radiance L.

Relation to prior work. When differentiating radiance, Li et al. [2018a] only considers the RE (7), which is equivalent to our handling of the reflected/refracted radiance L^r_s in Eq. (29), and neglects the RTE (11). Thus, their technique cannot handle the volumetric phenomena governed by the RTE and is a special case of ours.

5 UNBIASED MONTE CARLO ESTIMATOR

We now show how to numerically estimate the scene derivative $\mathrm{d}_{\boldsymbol{\pi}}L$ of interior radiance L. To this end, we show in §5.1 how individual components $\dot{L} := \partial_\pi L$ can be computed based on our derivations from §4. Then, we discuss the differentiation of pixel intensities in §5.2. Lastly, we discuss handling of the total derivative $\mathrm{d}_{\boldsymbol{\pi}}L$ in §5.3.

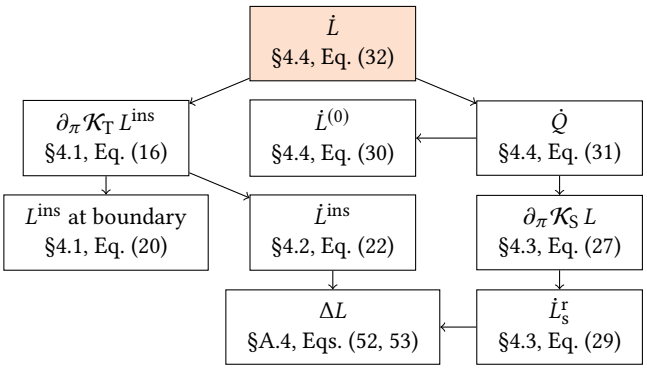

Fig. 7. A **summary of dependencies** between $\dot{L}$-related quantities derived in §4. An arrow from A to B indicates that A depends on B.

5.1 Monte Carlo Estimation of $\dot{L}$

To estimate $\dot{L}$ for some scene parameter $\pi \in \boldsymbol{\pi}$, we introduce a Monte Carlo technique analogous to unidirectional volumetric path tracing (VPT) that handles L and $\dot{L}$ jointly.

5.1.1 Next-Event Estimation. A variance-reduction technique essential to the efficiency of VPT is next-event estimation (NEE): instead of only "tallying" when a light transport path traced from the virtual sensor hits a light source, NEE-based methods tally at each volumetric and interfacial scattering location. We now show how NEE can be used for estimating $\dot{L}$.

Let $L^{(1)} := L - L^{(0)}$. Then, the in-scattered radiance L^ins (5) and reflected/refracted radiance L^r_s (7) can be expressed as sums of an *indirect* and a *direct* term of the form

$$\underbrace{\int_{\mathbb{S}^2} f(\boldsymbol{x}, -\boldsymbol{\omega}', \boldsymbol{\omega})\, L^{(1)}(\boldsymbol{x}, \boldsymbol{\omega}')\, \mathrm{d}\boldsymbol{\omega}'}_{\text{Indirect}} + \underbrace{\int_{\mathbb{S}^2} f(\boldsymbol{x}, -\boldsymbol{\omega}', \boldsymbol{\omega})\, L^{(0)}(\boldsymbol{x}, \boldsymbol{\omega}')\, \mathrm{d}\boldsymbol{\omega}'}_{\text{Direct}}, \tag{33}$$

where $f := f_\mathrm{p}$ for L^ins and $f := f_\mathrm{s}$ for L^r_s.

Since $\dot{L} = \dot{L}^{(1)} + \dot{L}^{(0)}$, the derivatives of L^ins and L^r_s given by Eqs. (22) and (29) can be rewritten as

$$\begin{aligned}\dot{L}^\mathrm{ins}(\boldsymbol{x}, \boldsymbol{\omega}) = {} & \underbrace{\int_{\mathbb{S}^2} \partial_\pi \left[f_\mathrm{p}(\boldsymbol{x}, -\boldsymbol{\omega}', \boldsymbol{\omega})\, L^{(1)}(\boldsymbol{x}, \boldsymbol{\omega}') \right] \mathrm{d}\boldsymbol{\omega}'}_{\text{Indirect}} + \\ & \underbrace{\int_{\mathbb{S}^2} \partial_\pi \left[f_\mathrm{p}(\boldsymbol{x}, -\boldsymbol{\omega}', \boldsymbol{\omega})\, L^{(0)}(\boldsymbol{x}, \boldsymbol{\omega}') \right] \mathrm{d}\boldsymbol{\omega}'}_{\text{Direct}} + \\ & \underbrace{\int_{\partial^2\Omega(\boldsymbol{x})} \left\langle \boldsymbol{n}_\perp, \partial_\pi(\boldsymbol{y} \to \boldsymbol{x}) \right\rangle f_\mathrm{p}(\boldsymbol{x}, \boldsymbol{x} \to \boldsymbol{y}, \boldsymbol{\omega})\, \Delta L(\boldsymbol{x}, \boldsymbol{y} \to \boldsymbol{x})\, V(\boldsymbol{x}, \boldsymbol{y})\, \frac{\sin\theta}{\|\boldsymbol{y}-\boldsymbol{x}\|}\, \mathrm{d}\ell(\boldsymbol{y})}_{\text{Edge}},\end{aligned} \tag{34}$$

and

$$\begin{aligned}\dot{L}^\mathrm{r}_\mathrm{s}(\boldsymbol{x}, \boldsymbol{\omega}) = {} & \underbrace{\int_{\mathbb{S}^2} \partial_\pi \left[f_\mathrm{s}(\boldsymbol{x}, -\boldsymbol{\omega}', \boldsymbol{\omega})\, L^{(1)}(\boldsymbol{x}, \boldsymbol{\omega}') \right] \mathrm{d}\boldsymbol{\omega}'}_{\text{Indirect}} + \\ & \underbrace{\int_{\mathbb{S}^2} \partial_\pi \left[f_\mathrm{s}(\boldsymbol{x}, -\boldsymbol{\omega}', \boldsymbol{\omega})\, L^{(0)}(\boldsymbol{x}, \boldsymbol{\omega}') \right] \mathrm{d}\boldsymbol{\omega}'}_{\text{Direct}} + \\ & \underbrace{\int_{\partial^2\Omega(\boldsymbol{x})} \left\langle \boldsymbol{n}_\perp, \partial_\pi(\boldsymbol{y} \to \boldsymbol{x}) \right\rangle f_\mathrm{s}(\boldsymbol{x}, \boldsymbol{x} \to \boldsymbol{y}, \boldsymbol{\omega})\, \Delta L(\boldsymbol{x}, \boldsymbol{y} \to \boldsymbol{x})\, V(\boldsymbol{x}, \boldsymbol{y})\, \frac{\sin\theta}{\|\boldsymbol{y}-\boldsymbol{x}\|}\, \mathrm{d}\ell(\boldsymbol{y})}_{\text{Edge}}.\end{aligned} \tag{35}$$

In Eqs. (33–35), the integrals involving $L^{(0)}$ (enclosed in green boxes) act as the *direct* terms and can be estimated using multiple importance sampling (MIS). The edge integrals (in orange boxes), on the contrary, depend on the differences ΔL of L across boundary curves (26) and need to be estimated by tracing additional *side paths* (see Figure 9 and discussions later). Lastly, the integrals involving $L^{(1)}$ (in purple boxes) are analogous to *indirect illumination* and need to be estimated recursively based on

$$\begin{aligned}& L^{(1)}(\boldsymbol{x}, \boldsymbol{\omega}) \\ & = \int_0^D \underbrace{T(\boldsymbol{x}', \boldsymbol{x})\, \sigma_\mathrm{s}(\boldsymbol{x}')\, L^\mathrm{ins}(\boldsymbol{x}', \boldsymbol{\omega})}_{\text{Volumetric}}\, \mathrm{d}\tau + \underbrace{T(\boldsymbol{x}_0, \boldsymbol{x})\, L^\mathrm{r}_\mathrm{s}(\boldsymbol{x}_0, \boldsymbol{\omega})}_{\text{Interfacial}},\end{aligned} \tag{36}$$

and

$$\begin{aligned}& \dot{L}^{(1)}(\boldsymbol{x}, \boldsymbol{\omega}) = \dot{L}(\boldsymbol{x}, \boldsymbol{\omega}) - \dot{L}^{(0)}(\boldsymbol{x}, \boldsymbol{\omega}) \\ & = \int_0^D \underbrace{T(\boldsymbol{x}', \boldsymbol{x}) \left[\sigma_\mathrm{s}(\boldsymbol{x}')\, \dot{L}^\mathrm{ins}(\boldsymbol{x}', \boldsymbol{\omega}) + \left(\dot{\sigma}_\mathrm{s}(\boldsymbol{x}') - \Sigma_\mathrm{t}(\boldsymbol{x}, \boldsymbol{\omega}, \tau)\, \sigma_\mathrm{s}(\boldsymbol{x}') \right) L^\mathrm{ins}(\boldsymbol{x}', \boldsymbol{\omega}) \right]}_{\text{Volumetric}} \mathrm{d}\tau \\ & + \underbrace{T(\boldsymbol{x}_0, \boldsymbol{x}) \left[- \left(\Sigma_\mathrm{t}(\boldsymbol{x}, \boldsymbol{\omega}, D) + \dot{D}\, \sigma_\mathrm{t}(\boldsymbol{x}_0) \right) L^\mathrm{r}_\mathrm{s}(\boldsymbol{x}_0, \boldsymbol{\omega}) + \dot{L}^\mathrm{r}_\mathrm{s}(\boldsymbol{x}_0, \boldsymbol{\omega}) + \dot{D}\, \sigma_\mathrm{s}(\boldsymbol{x}_0)\, L^\mathrm{ins}(\boldsymbol{x}_0, \boldsymbol{\omega}) \right]}_{\text{Interfacial}}.\end{aligned} \tag{37}$$

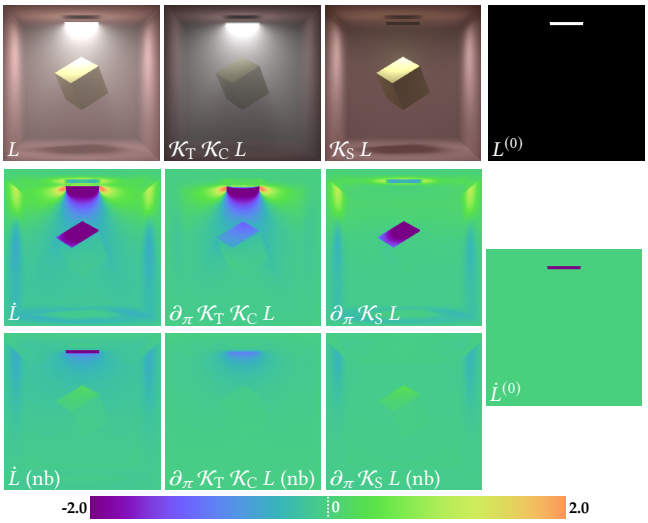

Fig. 8. **Component-wise visualizations** of interior radiance L and its (partial) scene derivative $\dot{L}$. In these images, each pixel stores a single radiance value corresponding to the ray going through the center of that pixel. (Top) original radiance values; (middle) radiance derivatives (of the red channel) with respect to the vertical displacement of the area light and the box encoded in false colors; (bottom) same as the middle but with all the boundary integrals neglected (indicated as "nb").

Algorithm 1 Unbiased Monte Carlo estimator of $L^{(1)}$ and $\dot{L}^{(1)}$

1: **function** RADIANCE($\boldsymbol{x}, \dot{\boldsymbol{x}}, \boldsymbol{\omega}, \dot{\boldsymbol{\omega}}$)
2: Calculate D and $\dot{D}$ based on $\boldsymbol{x}$, $\dot{\boldsymbol{x}}$, $\boldsymbol{\omega}$, and $\dot{\boldsymbol{\omega}}$
3: Sample free-flight distance τ
4: **if** $\tau < D$ **then** ▷ volumetric
5: $\boldsymbol{x}' \leftarrow \boldsymbol{x} - \tau\boldsymbol{\omega}$, $\dot{\boldsymbol{x}}' \leftarrow \dot{\boldsymbol{x}} - \tau\dot{\boldsymbol{\omega}}$ ▷ $\boldsymbol{x}' \in \Omega \setminus \partial\Omega$
6: $\alpha \leftarrow \sigma_{\mathrm{s}}(\boldsymbol{x}')/\sigma_{\mathrm{t}}(\boldsymbol{x}')$
7: $\beta \leftarrow (\dot{\sigma}_{\mathrm{s}}(\boldsymbol{x}') - \Sigma_{\mathrm{t}}(\boldsymbol{x}, \boldsymbol{\omega}, \tau)\,\sigma_{\mathrm{s}}(\boldsymbol{x}'))/\sigma_{\mathrm{t}}(\boldsymbol{x}')$ ▷ (17)
8: Draw $\boldsymbol{\omega}' \sim f_{\mathrm{p}}(\boldsymbol{x}', -\boldsymbol{\omega}', \boldsymbol{\omega})$
9: $(L^{(1)}, \dot{L}^{(1)}) \leftarrow$ RADIANCE$(\boldsymbol{x}', \dot{\boldsymbol{x}}', \boldsymbol{\omega}', \boldsymbol{0})$ ▷ recursion
10: $L^{\mathrm{ins}} \leftarrow L^{(1)}$
11: $\dot{L}^{\mathrm{ins}} \leftarrow \dot{L}^{(1)} + (\dot{f}_p(\boldsymbol{x}', -\boldsymbol{\omega}', \boldsymbol{\omega})/f_{\mathrm{p}}(\boldsymbol{x}', -\boldsymbol{\omega}', \boldsymbol{\omega}))\,L^{(1)}$
12: $L^{\mathrm{ins}} \leftarrow L^{\mathrm{ins}} + \int_{\mathbb{S}^2} f_{\mathrm{p}}(\boldsymbol{x}', -\boldsymbol{\omega}', \boldsymbol{\omega})\,L^{(0)}(\boldsymbol{x}', \boldsymbol{\omega}')\,\mathrm{d}\boldsymbol{\omega}'$
13: $\dot{L}^{\mathrm{ins}} \leftarrow \dot{L}^{\mathrm{ins}} + \int_{\mathbb{S}^2} \partial_\pi[f_{\mathrm{p}}(\boldsymbol{x}', -\boldsymbol{\omega}', \boldsymbol{\omega})\,L^{(0)}(\boldsymbol{x}', \boldsymbol{\omega}')]\,\mathrm{d}\boldsymbol{\omega}'$
14: $\dot{L}^{\mathrm{ins}} \leftarrow \dot{L}^{\mathrm{ins}} +$ EDGEINTEGRAL$(\boldsymbol{x}', \dot{\boldsymbol{x}}', \boldsymbol{\omega}, f_{\mathrm{p}})$ ▷ Alg. 2
15: $L^{(1)} \leftarrow \alpha\,L^{\mathrm{ins}}$, $\dot{L}^{(1)} \leftarrow \alpha\,\dot{L}^{\mathrm{ins}} + \beta\,L^{\mathrm{ins}}$
16: **else** ▷ interfacial
17: $\boldsymbol{x}_0 \leftarrow \boldsymbol{x} - D\boldsymbol{\omega}$, $\dot{\boldsymbol{x}}_0 \leftarrow \dot{\boldsymbol{x}} - \dot{D}\boldsymbol{\omega} - D\dot{\boldsymbol{\omega}}$ ▷ $\boldsymbol{x}_0 \in \partial\Omega$
18: Draw $\boldsymbol{\omega}'$ with probability $p(\boldsymbol{\omega}')$
19: $\alpha \leftarrow f_{\mathrm{s}}(\boldsymbol{x}_0, \boldsymbol{\omega}', \boldsymbol{\omega})/p(\boldsymbol{\omega}')$
20: $\beta \leftarrow \Sigma_{\mathrm{t}}(\boldsymbol{x}, \boldsymbol{\omega}, D) + \dot{D}\,\sigma_{\mathrm{t}}(\boldsymbol{x}_0)$ ▷ (17)
21: $(L^{(1)}, \dot{L}^{(1)}) \leftarrow$ RADIANCE$(\boldsymbol{x}_0, \dot{\boldsymbol{x}}_0, \boldsymbol{\omega}', \boldsymbol{0})$ ▷ recursion
22: $L^{\mathrm{r}}_{\mathrm{s}} \leftarrow \alpha\,L^{(1)}$
23: $\dot{L}^{\mathrm{r}}_{\mathrm{s}} \leftarrow \alpha\,\dot{L}^{(1)} + (\dot{f}_{\mathrm{s}}(\boldsymbol{x}_0, -\boldsymbol{\omega}', \boldsymbol{\omega})/p(\boldsymbol{\omega}'))\,L^{(1)}$
24: $L^{\mathrm{r}}_{\mathrm{s}} \leftarrow L^{\mathrm{r}}_{\mathrm{s}} + \int_{\mathbb{S}^2} f_{\mathrm{s}}(\boldsymbol{x}_0, -\boldsymbol{\omega}', \boldsymbol{\omega})\,L^{(0)}(\boldsymbol{x}_0, \boldsymbol{\omega}')\,\mathrm{d}\boldsymbol{\omega}'$
25: $\dot{L}^{\mathrm{r}}_{\mathrm{s}} \leftarrow \dot{L}^{\mathrm{r}}_{\mathrm{s}} + \int_{\mathbb{S}^2} \partial_\pi[f_{\mathrm{s}}(\boldsymbol{x}_0, -\boldsymbol{\omega}', \boldsymbol{\omega})\,L^{(0)}(\boldsymbol{x}_0, \boldsymbol{\omega}')]\,\mathrm{d}\boldsymbol{\omega}'$
26: $\dot{L}^{\mathrm{r}}_{\mathrm{s}} \leftarrow \dot{L}^{\mathrm{r}}_{\mathrm{s}} +$ EDGEINTEGRAL$(\boldsymbol{x}_0, \dot{\boldsymbol{x}}_0, \boldsymbol{\omega}, f_{\mathrm{s}})$ ▷ Alg. 2
27: $L^{\mathrm{ins}} \leftarrow$ INSCATRADIANCE$(\boldsymbol{x}_0, \boldsymbol{\omega})$ ▷ Appendix B
28: $L^{(1)} \leftarrow L^{\mathrm{r}}_{\mathrm{s}}$, $\dot{L}^{(1)} \leftarrow -\beta\,L^{\mathrm{r}}_{\mathrm{s}} + \dot{L}^{\mathrm{r}}_{\mathrm{s}} + \dot{D}\,\sigma_{\mathrm{s}}(\boldsymbol{x}_0)\,L^{\mathrm{ins}}$
29: **end if**
30: **return** $(L^{(1)}, \dot{L}^{(1)})$
31: **end function**

Algorithm 2 Monte Carlo estimator of the edge integrals

1: **function** EDGEINTEGRAL($\boldsymbol{x}, \dot{\boldsymbol{x}}, \boldsymbol{\omega}, f$)
2: Draw $\boldsymbol{y}$ from $\partial\Omega(\boldsymbol{x})$ with probability $p(\boldsymbol{y})$
3: $\boldsymbol{\omega}' \leftarrow (\boldsymbol{y} \to \boldsymbol{x})$
4: Calculate $\boldsymbol{n}_\perp$, $\dot{\boldsymbol{\omega}}'$ using Eqs. (24, 25)
5: $\Delta L \leftarrow$ DELTARADIANCE$(\boldsymbol{x}, \dot{\boldsymbol{x}}, \boldsymbol{\omega}', \dot{\boldsymbol{\omega}}')$ ▷ Appendix B
6: **return** $\langle \boldsymbol{n}_\perp, \dot{\boldsymbol{\omega}}' \rangle\, f(\boldsymbol{x}, -\boldsymbol{\omega}', \boldsymbol{\omega})\,\Delta L\,\frac{V(\boldsymbol{x},\boldsymbol{y})\,\sin\theta}{\|\boldsymbol{y}-\boldsymbol{x}\|\,p(\boldsymbol{y})}$
7: **end function**

5.1.2 Our Estimator. Based on Eqs. (33–37), we introduce an unbiased approach (Algorithm 1) that jointly estimates $L^{(1)}$ and $\dot{L}^{(1)}$ using Monte Carlo integration.

Sampling free-flight distance. Similar to VPT, our algorithm starts with drawing the free-flight distance τ (Line 3 of Algorithm 1). If $\tau < D$, the *volumetric* terms (in pink boxes) from Eqs. (36) and (37) are estimated at $\boldsymbol{x}' = \boldsymbol{x} - \tau\boldsymbol{\omega}$; otherwise, the *interfacial* terms (in yellow boxes) are calculated at $\boldsymbol{x}_0 = \boldsymbol{x} - D\boldsymbol{\omega}$. We detail how these terms are estimated in the following.

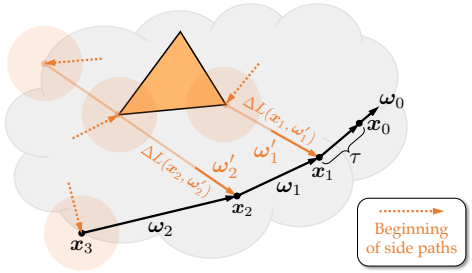

Fig. 9. An **illustration** of the joint estimation of $L^{(1)}(\boldsymbol{x}_0, \boldsymbol{\omega}_0)$ and $\dot{L}^{(1)}(\boldsymbol{x}_0, \boldsymbol{\omega}_0)$ using Algorithm 1 for some $\boldsymbol{x}_0 \in \Omega \setminus \partial\Omega$ and $\boldsymbol{\omega}_0 \in \mathbb{S}^2$. Similar to conventional volumetric path tracing, our algorithm begins with drawing a free-flight distance τ (Line 3), yielding a volumetric event at $\boldsymbol{x}_1 = \boldsymbol{x}_0 - \tau\boldsymbol{\omega}_0$ in this example. Then, a new direction $\boldsymbol{\omega}_1$ is drawn (Line 8) followed by a recursive estimation of the indirect terms $L^{(1)}(\boldsymbol{x}_1, \boldsymbol{\omega}_1)$ and $\dot{L}^{(1)}(\boldsymbol{x}_1, \boldsymbol{\omega}_1)$ (Line 9). Next, the direct terms are estimated at $\boldsymbol{x}_1$ (Lines 12 and 13). Lastly, the edge integral term $B^{\mathrm{ins}}(\boldsymbol{x}_1, \boldsymbol{\omega}_0)$ is handled by drawing another direction $\boldsymbol{\omega}_1'$ in which $L(\boldsymbol{x}_1, \boldsymbol{\omega}_1')$ is discontinuous (Line 14). In this example, $\Delta L(\boldsymbol{x}_1, \boldsymbol{\omega}_1')$ arises from a sharp edge and can be estimated by tracing one side path via Eq. (53). Similarly, the joint estimation of $L^{(1)}(\boldsymbol{x}_1, \boldsymbol{\omega}_1)$ and $\dot{L}^{(1)}(\boldsymbol{x}_1, \boldsymbol{\omega}_1)$ boils down to (i) calculating the indirect terms $L^{(1)}(\boldsymbol{x}_2, \boldsymbol{\omega}_2)$ and $\dot{L}^{(1)}(\boldsymbol{x}_2, \boldsymbol{\omega}_2)$ recursively; and (ii) estimating the direct and edge terms. In this illustration, $\Delta L(\boldsymbol{x}_2, \boldsymbol{\omega}_2')$ results from a silhouette edge and is estimated by tracing two side paths via Eq. (52). When estimating $L^{(1)}(\boldsymbol{x}_2, \boldsymbol{\omega}_2)$ and $\dot{L}^{(1)}(\boldsymbol{x}_2, \boldsymbol{\omega}_2)$, $\boldsymbol{x}_3 \in \partial\Omega$ corresponds to an interfacial event where the in-scattered radiance L^{ins} is estimated using Eq. (20).

Volumetric. The volumetric terms in Eqs. (36) and (37) are estimated jointly by Lines 5–15 of Algorithm 1. When the volumetric terms are picked, $\tau < D$, and the probability density of τ is $p(\tau) = \sigma_{\mathrm{t}}(\boldsymbol{x}')\,T(\boldsymbol{x}', \boldsymbol{x})$. Dividing $p(\tau)$ from the integrand cancels out the transmittance terms, yielding

$$\langle L^{(1)} \rangle = \alpha\,L^{\mathrm{ins}}, \quad \langle \dot{L}^{(1)} \rangle = \alpha\,\dot{L}^{\mathrm{ins}} + \beta\,L^{\mathrm{ins}}, \tag{38}$$

with α and β given by Lines 6 and 7. To estimate $L^{\mathrm{ins}}(\boldsymbol{x}', \boldsymbol{\omega})$ and $\dot{L}^{\mathrm{ins}}(\boldsymbol{x}', \boldsymbol{\omega})$, the main ingredients of Eq. (38), we apply Monte Carlo integration to Eqs. (33) and (34) jointly as follows.

- We estimate the indirect terms by drawing a random $\boldsymbol{\omega}'$ according to the phase function f_{p} and obtain $L^{(1)}(\boldsymbol{x}', \boldsymbol{\omega}')$ and $\dot{L}^{(1)}(\boldsymbol{x}', \boldsymbol{\omega}')$ recursively (Line 9 of Algorithm 1). Techniques like Russian roulette can be used here to avoid infinite recursion.
- Then, the indirect terms of Eqs. (33) and (34) are respectively given by $L^{(1)}(\boldsymbol{x}', \boldsymbol{\omega}')$ and
$$\partial_\pi\left[f_{\mathrm{p}}(\boldsymbol{x}', -\boldsymbol{\omega}', \boldsymbol{\omega})\,L^{(1)}(\boldsymbol{x}, \boldsymbol{\omega}')\right]/f_{\mathrm{p}}(\boldsymbol{x}', -\boldsymbol{\omega}', \boldsymbol{\omega})$$
$$= \dot{L}^{(1)}(\boldsymbol{x}, \boldsymbol{\omega}') + \frac{\dot{f}_p(\boldsymbol{x}', -\boldsymbol{\omega}', \boldsymbol{\omega})}{f_{\mathrm{p}}(\boldsymbol{x}', -\boldsymbol{\omega}', \boldsymbol{\omega})} L^{(1)}(\boldsymbol{x}, \boldsymbol{\omega}'),$$
which corresponds to Lines 10 and 11.
- The direct term can be estimated directly using MIS (Lines 12 and 13).
- The edge-integral term is computed via the EDGEINTEGRAL function (Line 14) that uniformly samples a point $\boldsymbol{y}$ from an edge and calculates the corresponding ΔL (Algorithm 2).

Interfacial. The interfacial terms in Eqs. (36) and (37) are handled with Lines 17–28 of Algorithm 1. Since $p(\tau) = T(\boldsymbol{x}_0, \boldsymbol{x})$, dividing it from the direct terms cancels out the transmittances, yielding

$$\langle L^{(1)} \rangle = L_{\mathrm{s}}^{\mathrm{r}}, \quad \langle \dot{L}^{(1)} \rangle = -\beta\, L_{\mathrm{s}}^{\mathrm{r}} + \dot{L}_{\mathrm{s}}^{\mathrm{r}} + \dot{D}\, \sigma_{\mathrm{s}}\, L^{\mathrm{ins}}, \tag{39}$$

with β calculated in Line 20. We estimate $L_{\mathrm{s}}^{\mathrm{r}}(\boldsymbol{x}_0, \boldsymbol{\omega})$ and $\dot{L}_{\mathrm{s}}^{\mathrm{r}}(\boldsymbol{x}_0, \boldsymbol{\omega})$, the main ingredients of Eq. (39), based on Eqs. (33) and (35) as follows.

- The indirect terms are computed jointly by sampling $\boldsymbol{\omega}'$ according to the local BSDF and acquiring $L(\boldsymbol{x}_0, \boldsymbol{\omega}')$ and $\dot{L}(\boldsymbol{x}_0, \boldsymbol{\omega}')$ recursively (Lines 21–23 of Algorithm 1).
- The direct terms are estimated directly using MIS (Lines 24 and 25).
- Similar to the volumetric case, the edge-integral term is estimated using the EdgeIntegral function (Line 26).
- Additionally, the in-scattered radiance $L^{\mathrm{ins}}(\boldsymbol{x}_0, \boldsymbol{\omega})$, which arises from the differentiation of $\mathcal{K}_{\mathrm{T}}\, \mathcal{K}_{\mathrm{S}}\, L$ shown in Eq. (16), needs to be calculated (Line 27) according to Eq. (20).

The background colors of individual lines in Algorithm 1 indicate the correspondence to the indirect (purple), direct (green), and edge (orange) terms in Eqs. (33–35).

Lastly, calculating $L^{\mathrm{ins}}(\boldsymbol{x}_0, \boldsymbol{\omega})$ (Line 27 of Algorithm 1) and ΔL (Line 5 of Algorithm 2) requires tracing additional *side paths*. Since both $L^{\mathrm{ins}}(\boldsymbol{x}_0, \boldsymbol{\omega})$ and ΔL depend on only the interior radiance L but not its derivative $\dot{L}$, the side paths traced to estimate these terms can be handled using conventional volumetric path tracing. For more details, please refer to Appendix B.

We illustrate the execution of Algorithm 1 in Figure 9.

5.2 Differentiating Pixel Intensities

Physics-based renderers typically output entire images instead of individual radiance values, and differentiating rendered images requires calculating scene derivatives of pixel intensities.

Specifically, the intensity of each pixel is typically modeled as a radiometric measurement given by the inner product between the radiance L and some (normalized) reconstruction kernel K:

$$I = \int_{[0,1)^2} L(\boldsymbol{x}(\boldsymbol{u}), \boldsymbol{\omega}(\boldsymbol{u}))\, K(\boldsymbol{u})\, \mathrm{d}\boldsymbol{u}, \tag{40}$$

where $\boldsymbol{x} : [0,1)^2 \to \mathcal{P}$ maps $[0,1)^2$ to the pixel footprint $\mathcal{P}$ on the image plane in 3D, and $\boldsymbol{\omega}(\boldsymbol{u})$ depends on the camera model and equals $\boldsymbol{x}(\boldsymbol{u}) \to \boldsymbol{x}^{\mathrm{cam}}$ for a perspective camera centered at $\boldsymbol{x}^{\mathrm{cam}}$.

When K is independent of the scene parameter π and continuous in $\boldsymbol{u}$, Eq. (40) can be differentiated via

$$\begin{aligned} \dot{I} = &\int_{[0,1)^2} \dot{L}(\boldsymbol{x}(\boldsymbol{u}), \boldsymbol{\omega}(\boldsymbol{u}))\, K(\boldsymbol{u})\, \mathrm{d}\boldsymbol{u} + \\ &\int_{\partial[0,1)^2} \langle \boldsymbol{n}, \dot{\boldsymbol{x}}(\boldsymbol{u}) \rangle\, \Delta L(\boldsymbol{x}(\boldsymbol{u}), \boldsymbol{\omega}(\boldsymbol{u}))\, K(\boldsymbol{u})\, \mathrm{d}\ell(\boldsymbol{u}), \end{aligned} \tag{41}$$

where $\partial[0,1)^2$ denotes all boundary curves representing discontinuities of L (in $\boldsymbol{u}$). Further, $\boldsymbol{n}$ and $\dot{\boldsymbol{x}}$ capture the normal and change rate of these curves.

Notice that, when the camera's location, orientation, or field of view depends on π (which is required for optimizing camera parameters), so do the corresponding $\boldsymbol{x}(\boldsymbol{u})$ and $\boldsymbol{\omega}(\boldsymbol{u})$ in Eqs. (40) and (41). This necessitates our derivations of $\dot{L}(\boldsymbol{x}, \boldsymbol{\omega})$ in §4 with $\boldsymbol{x}$ and $\boldsymbol{\omega}$ being π-dependent.

To estimate $\dot{I}$, we draw $\boldsymbol{u}_1$ uniformly from $[0,1)^2$ and $\boldsymbol{u}_2$ from $\partial[0,1)^2$ with probability density $p(\boldsymbol{u}_2)$. Then,

$$\langle \dot{I} \rangle = \dot{L}(\boldsymbol{x}(\boldsymbol{u}_1), \boldsymbol{\omega}(\boldsymbol{u}_1))\, K(\boldsymbol{u}_1) + \frac{\langle \boldsymbol{n}, \dot{\boldsymbol{x}}(\boldsymbol{u}_2) \rangle \Delta L(\boldsymbol{x}(\boldsymbol{u}_2), \boldsymbol{\omega}(\boldsymbol{u}_2))\, K(\boldsymbol{u}_2)}{p(\boldsymbol{u}_2)}, \tag{42}$$

where the first term on the RHS can be estimated using Algorithm 1 while the second can be handled by tracing side paths (similar to the primary edge sampling process in Li et al.'s work [2018a]).

5.3 Discussion

Estimating total derivatives. To estimate the total scene derivative $\mathrm{d}_{\boldsymbol{\pi}} L$, Algorithm 1 only needs to be executed once in a *vectorized* fashion. Specifically, one can replace all partial derivatives with their (vector-valued) total derivative counterparts and perform component-wise calculations between all total-derivative-related quantities. In this way, all paths traced by Algorithm 1 are effectively shared to calculate derivatives with respect to all scene parameters.

Homogeneous media. Our derivations in §4 and §5.1 enjoy the generality for handling heterogeneous media with spatially varying radiative transfer parameters σ_{t}, σ_{s}, and f_{p}.

On the other hand, for homogeneous media with π-invariant parameters, $\nabla\sigma_{\mathrm{s}} = \nabla\sigma_{\mathrm{t}} \equiv \mathbf{0}$, causing (i) $\beta \equiv 0$ for Lines 7 and 20 of Algorithm 1; and (ii) $\dot{L}^{(1)}$ and $\dot{L}^{(0)}$ in Eqs. (30) and (37) to become:

$$\dot{L}^{(0)}(\boldsymbol{x}, \boldsymbol{\omega}) = \mathrm{e}^{-\sigma_{\mathrm{t}} D} \left[\dot{L}_{\mathrm{s}}^{\mathrm{e}}(\boldsymbol{x}_0, \boldsymbol{\omega}) - \dot{D}\, \sigma_{\mathrm{t}}\, L_{\mathrm{s}}^{\mathrm{e}}(\boldsymbol{x}_0, \boldsymbol{\omega}) \right], \tag{43}$$

$$\begin{aligned} \dot{L}^{(1)}(\boldsymbol{x}, \boldsymbol{\omega}) = &\int_0^D \mathrm{e}^{-\sigma_{\mathrm{t}}\tau}\, \sigma_{\mathrm{s}}\, \dot{L}^{\mathrm{ins}}(\boldsymbol{x}', \boldsymbol{\omega})\, \mathrm{d}\tau + \\ &\mathrm{e}^{-\sigma_{\mathrm{t}} D} \left[\dot{L}_{\mathrm{s}}^{\mathrm{r}}(\boldsymbol{x}_0, \boldsymbol{\omega}) + \dot{D} \left(\sigma_{\mathrm{s}}\, L^{\mathrm{ins}}(\boldsymbol{x}_0, \boldsymbol{\omega}) - \sigma_{\mathrm{t}}\, L_{\mathrm{s}}^{\mathrm{r}}(\boldsymbol{x}_0, \boldsymbol{\omega}) \right) \right]. \end{aligned} \tag{44}$$

Computational efficiency. Since our estimator is built upon ordinary Monte Carlo (OMC), it has a theoretical convergence rate of $O(1/\sqrt{N})$ with N being the number of (main) sample paths. In practice, because of the additional side paths needed to be traced in each iteration for estimating ΔL and L^{ins}, Algorithm 1 is significantly more expensive than conventional VPT by having a time complexity of $O(K^2)$, as opposed to $O(K)$, for each sample path of length K. Designing new Monte Carlo estimators of scene derivatives with better performance is an exciting future research direction.

6 RESULTS

We implemented Algorithms 1 and 2, as well as Algorithms 3 and 4 in Appendix B, on the CPU in C++ using the Embree ray tracer.

Performance. As discussed in §5.3, unbiased and consistent estimations of radiance derivatives are significantly more expensive than those of radiance values due to the need to trace side paths. In practice, our estimation of a single derivative runs about 10× slower than conventional volumetric path tracing for radiance estimation (with the same number of samples per pixel). Fortunately, as our method allows multiple derivatives to be estimated jointly, the amortized overhead for multiple parameters is much lower. For example, when computing derivatives with respect to 10 scene parameters, our method is only 2× slower than handling one parameter.

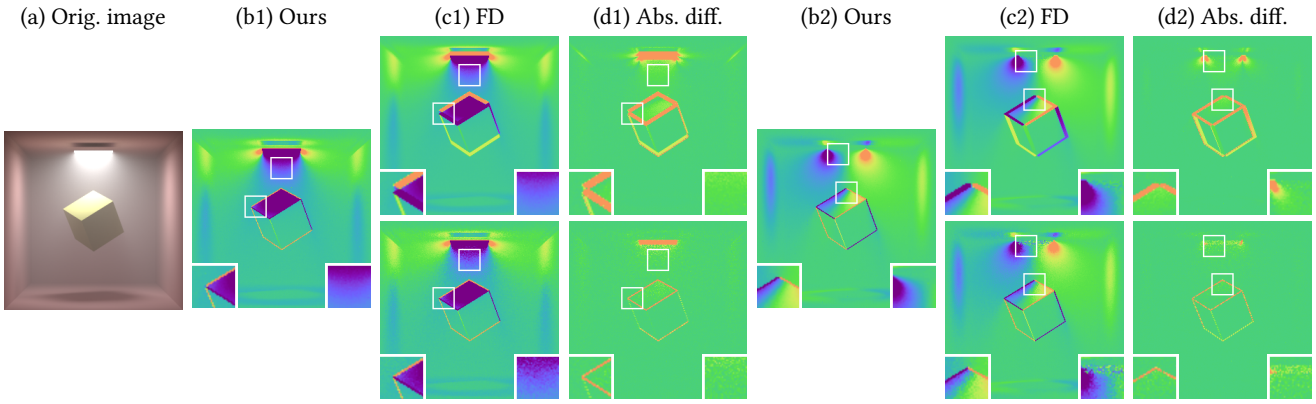

Fig. 10. **Equal-time comparisons** between derivatives estimated by (b) our method and (c) finite difference (FD) with respect to two distinctive scene parameters: (b1–d1) vertical displacement of the area light and the cube; (b2–d2) horizontal displacement of those objects. Pixel intensities of the derivative images are encoded in false colors using the same color map as Figure 8. The FD results on the first row use a larger spacing (0.1) and suffer from high bias (see the insets on the bottom left); those on the second row use a smaller spacing (0.01) and have high variance while still being biased (as shown by the insets on the bottom right). Our technique, in contrast, produces cleaner and unbiased derivative estimations.

Table 2. Optimization configurations and performance statistics for the inverse-rendering examples. Time is measured in CPU core minutes per iteration.

Scene		# Param.	# Iter.	Time
Teaser	(Fig. 1)	2	110	22
Glass	(Fig. 11)	4	80	12.2
Camera	(Fig. 11)	3	220	9.3
Reflector	(Fig. 11)	9	200	23.6
Shadow	(Fig. 12)	2	60	7.6
Multilayer	(Fig. 12)	3	50	31
Spotlight	(Fig. 13)	12	110	11.2
Logo	(Fig. 13)	101	100	27.2

6.1 Validation & Evaluation

To validate our derivations and implementation efficiency, we compare radiance derivatives estimated in equal time with our method (Algorithm 1) and finite difference (FD).

Figure 10 shows an example using the same Cornell-box-like scene as Figure 8. As a comparison, columns (b1) and (c1) both display derivatives with respect to the vertical displacement of the light and the cube,[6] resulted from our method and FD, respectively. Columns (b2) and (c2), on the other hand, visualize derivatives with respect to the horizontal displacement of those objects.

In both cases, FD results obtained with larger spacing (top) suffer from severe bias around object boundaries. Those with smaller spacing (bottom), on the contrary, suffer from higher noise while still being biased. Our technique is capable of producing much cleaner and unbiased derivative estimations.

For more validation results, please see the supplemental material.

[6]Compared to Figure 10, our derivatives images in Figure 8 do not show "edges" around moving objects since each pixel in those images corresponds to the derivative of one radiance value (as oppose to integrated radiances over the pixel).

6.2 Main Results

We now show inverse rendering results using gradients estimated with our approach. All these examples involve participating media/translucent objects and changing scene geometry, making previous methods inadequate.

In these examples, we take a target image I^{target} as input and search for scene parameters $\boldsymbol{\pi}^*$ that minimize the difference between the target and the rendered image $I(\boldsymbol{\pi})$ generated using these parameters. Namely,

$$\boldsymbol{\pi}^* = \arg\min_{\boldsymbol{\pi}} \left\| I(\boldsymbol{\pi}) - I^{\text{target}} \right\|. \tag{45}$$

In practice, we choose the norm $\|\cdot\|$ as the L_2 difference between Gaussian pyramids of the corresponding images[7]. We use the Adam method [Kingma and Ba 2014] to drive the gradient-descent processes. Please refer to Table 2 for the performance numbers and optimization configurations, and to the supplement for more results and animated versions of Figures 1, 11, 12, and 13.

Generic inverse rendering. Figure 1 illustrates an inverse rendering example where the pose of a light-emitting dodecahedron is optimized. From an initial configuration (a) that is quite different from the target (d), the optimization driven by derivatives estimated with our method (b) manages to find the global optimum in 110 iterations, as demonstrated by the parameter difference plot. Note that the optimization loss does not go to zero as noisy derivatives are used for faster iterations.

We show three more examples in Figure 11. The top row contains an example with an apple inside a cube with an absorbing medium and a rough dielectric interface. We jointly optimize the location of the apple inside the cube as well as the surface roughness of the cube itself. The second row of this image shows an example where the camera position and pose are optimized. The last row has an

[7]Our technique can be used with any differentiable metric, and we formulate the inverse rendering problem using the L_2 norm for its simplicity. Additionally, the norm is calculated based on Gaussian pyramids to make the optimization less prone to local minima. This, however, does not guarantee the optimization to be convex in general.

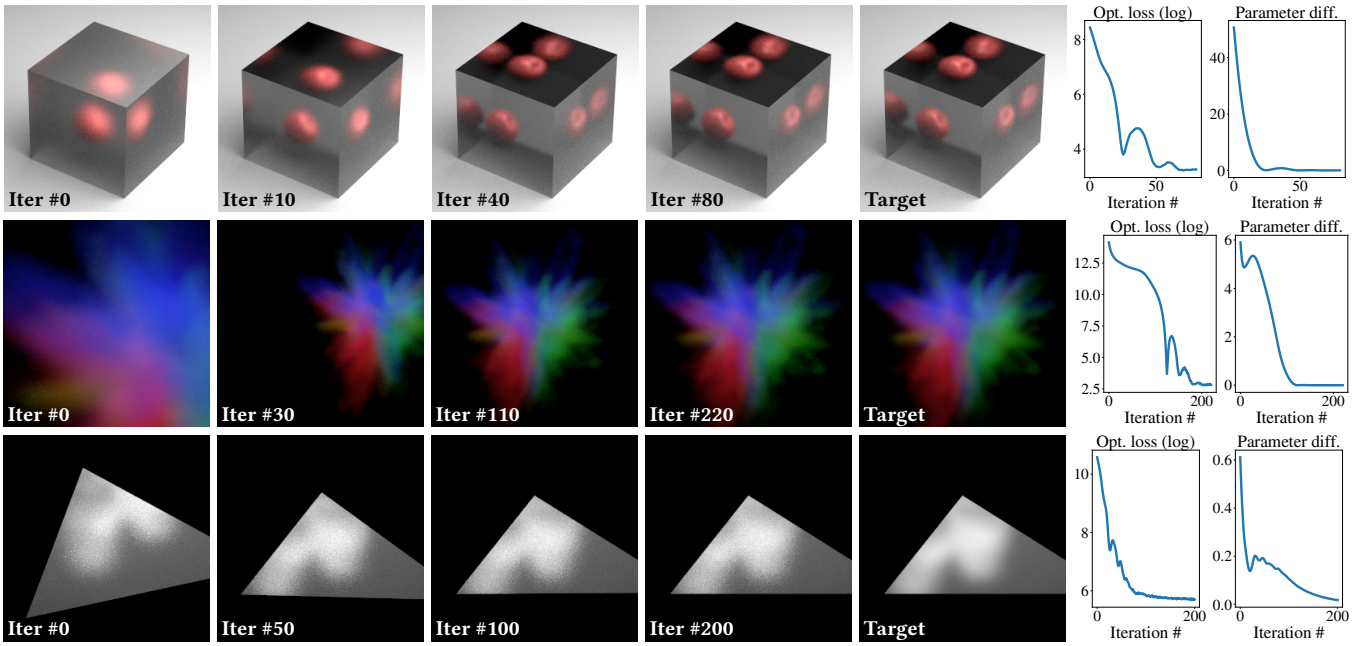

Fig. 11. **Inverse rendering results** obtained using gradients estimated with our technique (Algorithm 1). Please see the supplemental material for extended versions of these results visualizing the full optimization processes.

example where the camera looks at a smoke indirectly through a glossy triangular reflector. We optimize the 3D locations of each vertex of the reflector separately.

Imaging-inspired examples. In Figure 12, we show two examples inspired by real-world imaging problems. On the top of this figure, we show an example where a heterogeneous medium casts a colored shadow onto the ground. We optimize for the orientation as well as the optical density (as a global scaling factor) of this medium jointly by only looking at the shadow, mimicking a non-line-of-sight (NLOS) imaging scenario [Xin et al. 2019; O'Toole et al. 2018; Saunders et al. 2019; Velten et al. 2012]. Using gradients estimated with our method, the global optimum is reached within 60 iterations.

The bottom of Figure 12 shows an example where an object is embedded within a two-layer material modeled after human skin using measured properties [Collier et al. 2003; Walker et al. 2003; Chang et al. 2009]. We search for the location of the object, a common problem in biomedical imaging. Optimization with our estimated derivatives converges to the global optimum in 50 iterations.

Design-inspired examples. Figure 13 contains two examples inspired by design applications using arbitrary images (i.e., not 3D renderings) as input. On the top of this figure, we show an example where the orientations and colors of two small spotlights are jointly optimized. The input to this example is a simple drawing of the light beams. On the bottom is an example containing micro-scale translucent grains. We jointly optimize for the colors (i.e., single-scattering albedo) and physical sizes of these grains (in clusters) so that their macro-scale appearance matches the input image (i.e., the SIGGRAPH logo).

7 DISCUSSION AND CONCLUSION

Limitations and future work. Our derivations rely on a few simplifying assumptions stated at the start of §4. Although some of them (e.g., no zero-measure light source) are relaxed in the supplemental material, others require more sophisticated algorithms to handle. For instance, with ideal specular surfaces, discontinuities of the radiance $L(\boldsymbol{x}, \boldsymbol{\omega})$ in $\boldsymbol{\omega}$ need to also include mirror images of the geometric edges, which can be nontrivial to identify. Also, to support media where the transmittance is non-exponential, some of our derivations will need to be adjusted.

Another exciting direction for future research is to improve the computational efficiency of our Monte Carlo estimator (Algorithm 1). Developing more sophisticated estimators (e.g., those analogous to bidirectional path tracing) is an interesting future topic and necessitates the establishment of a path-integral form of Eq. (32).

Conclusion. We introduced a differential theory of radiative transfer by showing how the radiative transfer equation (RTE) can be differentiated with respect to arbitrary scene parameterizations. To this end, we derived the derivatives for individual terms of the RTE given by the transport, collision, and interfacial scattering operators. Our theory enjoys the generality to handle a large variety of radiative transfer phenomena including volumetric absorption, single and multiple scattering, anisotropic phase functions, and heterogeneity.

To numerically estimate these derivatives, we presented an unbiased Monte Carlo method analogous to volumetric path tracing (VPT) to compute scene derivatives of interior and interfacial radiances. Our technique symbolically differentiates contributions of

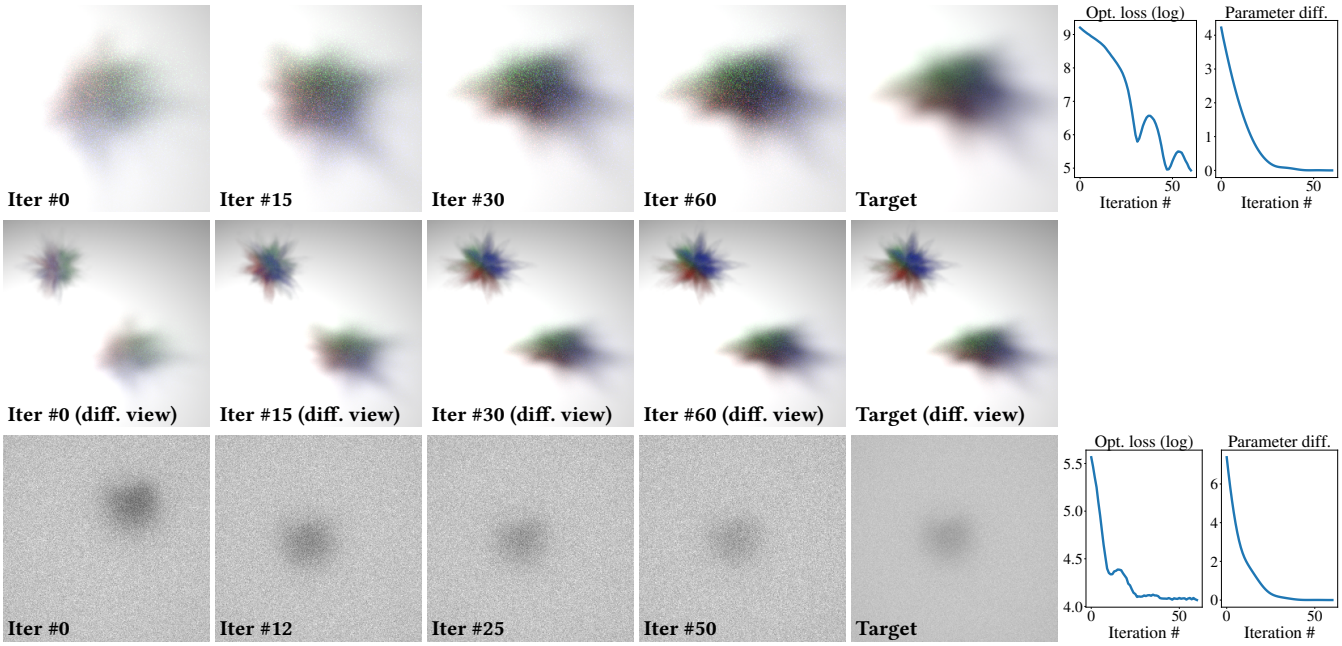

Fig. 12. **Imaging-inspired** inverse rendering examples obtained using gradients estimated with our technique (Algorithm 1). We use low sample counts for faster iterations, which causes the optimization loss to remain greater than zero (due to Monte Carlo noise) but does not affect the finding of target scene parameters (as the parameter differences do go to zero). Please refer to the supplemental material for animated versions of these results.

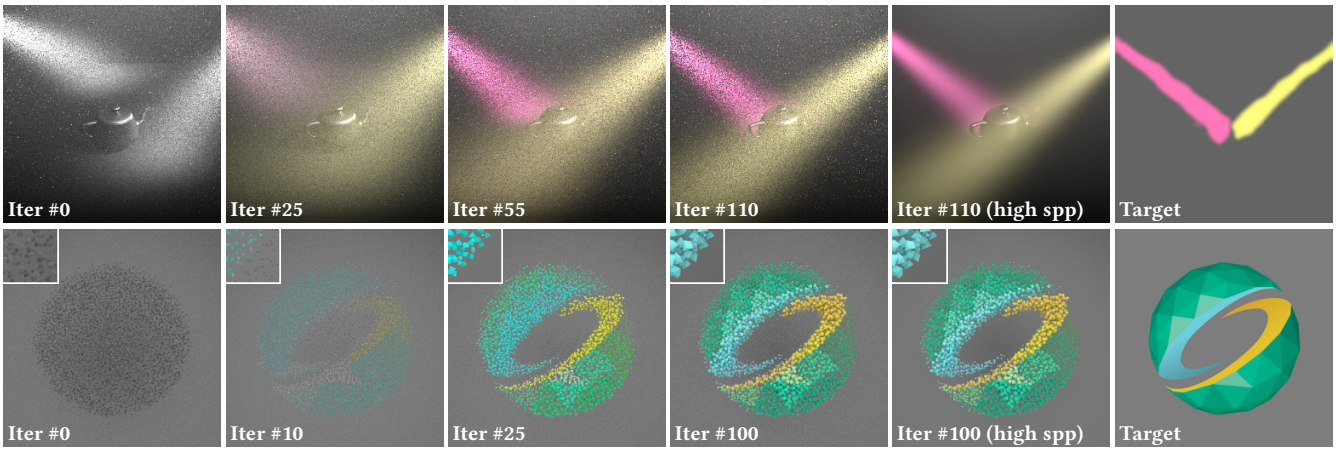

Fig. 13. **Design-inspired** inverse rendering examples obtained using gradients estimated with our technique (Algorithm 1). Similar to the imaging-inspired examples (Figure 12), we use low sample count for faster iterations. Please see the supplemental material for animated versions of these results showing the full optimization processes.

light transport paths and traces additional side paths toward geometric edges to account for the change of discontinuity boundaries.

We validated the correctness of our theory and implementation by comparing our derivative estimations to finite-difference results. Further, we demonstrated the practical usefulness of our technique via a few synthetic examples inspired by real-world applications in imaging and design.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their constructive comments. We are grateful to Jerome Spanier for many helpful suggestions. This work was supported in part by NSF grants 1717178, 1730147, 1816041, 1900849, and 1900927, an NVIDIA fellowship, the Ronald L. Graham Chair, and the UC San Diego Center for Visual Computing.

REFERENCES

James Arvo. 1994. The Irradiance Jacobian for partially occluded polyhedral sources. In *SIGGRAPH '94*. 343–350.

Dejan Azinovic, Tzu-Mao Li, Anton Kaplanyan, and Matthias Niessner. 2019. Inverse Path Tracing for Joint Material and Lighting Estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Benedikt Bitterli, Srinath Ravichandran, Thomas Müller, Magnus Wrenninge, Jan Novák, Steve Marschner, and Wojciech Jarosz. 2018. A Radiative Transfer Framework for Non-exponential Media. *ACM Trans. Graph.* 37, 6 (2018), 225:1–225:17.

James F. Blinn. 1982. Light reflection functions for simulation of clouds and dusty surfaces. In *SIGGRAPH '82*. 21–29.

Kenneth M Case and Paul Frederick Zweifel. 1967. Linear transport theory. (1967).

Eva Cerezo, Frederic Pérez, Xavier Pueyo, Francisco J Seron, and François X Sillion. 2005. A survey on participating media rendering techniques. *The Visual Computer* 21, 5 (2005), 303–328.

Subrahmanyan Chandrasekhar. 1960. *Radiative Transfer*. Courier Corporation.

Vivide Tuan-Chyan Chang, Peter S Cartwright, Sarah M Bean, Greg M Palmer, Rex C Bentley, and Nirmala Ramanujam. 2009. Quantitative physiology of the precancerous cervix in vivo through optical spectroscopy. *Neoplasia* 11, 4 (2009), 325–332.

Chengqian Che, Fujun Luan, Shuang Zhao, Kavita Bala, and Ioannis Gkioulekas. 2018. Inverse transport networks. *arXiv preprint arXiv:1809.10820* (2018).

Min Chen and James Arvo. 2000. Theory and application of specular path perturbation. *ACM Trans. Graph.* 19, 4 (2000), 246–278.

Tom Collier, Dizem Arifler, Anais Malpica, Michele Follen, and Rebecca Richards-Kortum. 2003. Determination of epithelial tissue scattering coefficient using confocal microscopy. *IEEE Journal of selected topics in quantum electronics* 9, 2 (2003), 307–313.

Harley Flanders. 1973. Differentiation under the integral sign. *The American Mathematical Monthly* 80, 6 (1973), 615–627.

Ioannis Gkioulekas, Anat Levin, and Todd Zickler. 2016. An Evaluation of Computational Imaging Techniques for Heterogeneous Inverse Scattering. In *Computer Vision – ECCV 2016*. Springer International Publishing, 685–701.

Ioannis Gkioulekas, Shuang Zhao, Kavita Bala, Todd Zickler, and Anat Levin. 2013. Inverse volume rendering with material dictionaries. *ACM Trans. Graph.* 32, 6 (2013), 162:1–162:13.

Milovš Hašan and Ravi Ramamoorthi. 2013. Interactive albedo editing in path-traced volumetric materials. *ACM Trans. Graph.* 32, 2 (2013), 11:1–11:11.

Homan Igehy. 1999. Tracing ray differentials. In *SIGGRAPH '99*. 179–186.

Akira Ishimaru. 1978. *Wave propagation and scattering in random media*. Academic press New York.

Wenzel Jakob, Adam Arbree, Jonathan T. Moon, Kavita Bala, and Steve Marschner. 2010. A radiative transfer framework for rendering materials with anisotropic structure. *ACM Trans. Graph.* 29, 4 (2010), 53:1–53:13.

Wenzel Jakob and Steve Marschner. 2012. Manifold exploration: A Markov chain Monte Carlo technique for rendering scenes with difficult specular transport. *ACM Trans. Graph.* 31, 4 (2012), 58:1–58:13.

James T. Kajiya and Brian P Von Herzen. 1984. Ray tracing volume densities. *SIGGRAPH Comput. Graph.* 18, 3 (1984), 165–174.

Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. 2018. Neural 3D mesh renderer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3907–3916.

Csaba Kelemen, László Szirmay-Kalos, György Antal, and Ferenc Csonka. 2002. A simple and robust mutation strategy for the metropolis light transport algorithm. In *Computer Graphics Forum*, Vol. 21. Wiley Online Library, 531–540.

Pramook Khungurn, Daniel Schroeder, Shuang Zhao, Kavita Bala, and Steve Marschner. 2015. Matching real fabrics with micro-appearance models. *ACM Trans. Graph.* 35, 1 (2015), 1:1–1:26.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

Tejas D. Kulkarni, Pushmeet Kohli, Joshua B. Tenenbaum, and Vikash Mansinghka. 2015. Picture: A Probabilistic Programming Language for Scene Perception. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Peter Kutz, Ralf Habel, Yining Karl Li, and Jan Novák. 2017. Spectral and decomposition tracking for rendering heterogeneous volumes. *ACM Trans. Graph.* 36, 4 (2017), 111:1–111:16.

Eric P Lafortune and Yves D Willems. 1996. Rendering participating media with bidirectional path tracing. In *Rendering techniques' 96*. Springer, 91–100.

L. Gary Leal. 2007. *Advanced Transport Phenomena: fluid mechanics and convective transport processes*. Vol. 7. Cambridge University Press.

Tzu-Mao Li. 2019. *Differentiable visual computing*. MIT PhD Dissertation.

Tzu-Mao Li, Miika Aittala, Frédo Durand, and Jaakko Lehtinen. 2018a. Differentiable Monte Carlo ray tracing through edge sampling. *ACM Trans. Graph.* 37, 6 (2018), 222:1–222:11.

Tzu-Mao Li, Jaakko Lehtinen, Ravi Ramamoorthi, Wenzel Jakob, and Frédo Durand. 2015. Anisotropic Gaussian mutations for Metropolis light transport through Hessian-Hamiltonian dynamics. *ACM Trans. Graph.* 34, 6 (2015), 209:1–209:13.

Zhengqin Li, Zexiang Xu, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. 2018b. Learning to reconstruct shape and spatially-varying reflectance from a single image. *ACM Trans. Graph.* 37, 6 (2018), 269:1–269:11.

Matthew M Loper and Michael J Black. 2014. OpenDR: An approximate differentiable renderer. In *European Conference on Computer Vision*. Springer, 154–169.

Abhimitra Meka, Maxim Maximov, Michael Zollhoefer, Avishek Chatterjee, Hans-Peter Seidel, Christian Richardt, and Christian Theobalt. 2018. Lime: Live intrinsic material estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6315–6324.

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

Jan Novák, Iliyan Georgiev, Johannes Hanika, and Wojciech Jarosz. 2018. Monte Carlo methods for volumetric light transport simulation. In *Computer Graphics Forum*, Vol. 37. Wiley Online Library, 551–576.

Matthew O'Toole, David B Lindell, and Gordon Wetzstein. 2018. Confocal non-line-of-sight imaging based on the light-cone transform. *Nature* 555, 7696 (2018), 338.

Mark Pauly, Thomas Kollig, and Alexander Keller. 2000. Metropolis light transport for participating media. In *Rendering Techniques 2000*. Springer, 11–22.

GC Pomraning. 1973. The equations of radiation hydrodynamics. *International Series of Monographs in Natural Philosophy, Oxford: Pergamon Press, 1973* (1973).

Ravi Ramamoorthi, Dhruv Mahajan, and Peter Belhumeur. 2007. A First-order Analysis of Lighting, Shading, and Shadows. *ACM Trans. Graph.* 26, 1 (2007), 2:1–2:21.

Holly E. Rushmeier and Kenneth E. Torrance. 1987. The zonal method for calculating light intensities in the presence of a participating medium. In *SIGGRAPH '87*. 293–302.

Charles Saunders, John Murray-Bruce, and Vivek K Goyal. 2019. Computational periscopy with an ordinary digital camera. *Nature* 565, 7740 (2019), 472.

Soumyadip Sengupta, Angjoo Kanazawa, Carlos D Castillo, and David W Jacobs. 2018. SfSNet: Learning shape, reflectance and illuminance of faces in the wild'. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6296–6305.

Jerome Spanier and Ely M Gelbard. 1969. *Monte Carlo principles and neutron transport problems*. The Addison-Wesley Publishing Company.

Denis Sumin, Tobias Rittig, Vahid Babaei, Thomas Nindel, Alexander Wilkie, Piotr Didyk, Bernd Bickel, Jaroslav Křivánek, Karol Myszkowski, and Tim Weyrich. 2019. Geometry-aware Scattering Compensation for 3D Printing. *ACM Trans. Graph.* 38, 4 (2019), 111:1–111:14.

Chia-Yin Tsai, Aswin C. Sankaranarayanan, and Ioannis Gkioulekas. 2019. Beyond Volumetric Albedo – A Surface Optimization Framework for Non-Line-Of-Sight Imaging. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Hendrik Christoffel van de Hulst et al. 1957. Light scattering by small particles. (1957).

Zdravko Velinov, Marios Papas, Derek Bradley, Paulo Gotardo, Parsa Mirdehghan, Steve Marschner, Jan Novák, and Thabo Beeler. 2018. Appearance Capture and Modeling of Human Teeth. *ACM Trans. Graph.* 37, 6 (2018), 207:1–207:13.

Andreas Velten, Thomas Willwacher, Otkrist Gupta, Ashok Veeraraghavan, Moungi G Bawendi, and Ramesh Raskar. 2012. Recovering three-dimensional shape around a corner using ultrafast time-of-flight imaging. *Nature communications* 3 (2012), 745.

Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. 2010. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research* 11, Dec (2010), 3371–3408.

DC Walker, BH Brown, AD Blackett, J Tidy, and RH Smallwood. 2003. A study of the morphological parameters of cervical squamous epithelium. *Physiological measurement* 24, 1 (2003), 121.

Jiajun Wu, Joshua B Tenenbaum, and Pushmeet Kohli. 2017. Neural scene de-rendering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 699–707.

Shumian Xin, Sotiris Nousias, Kiriakos N Kutulakos, Aswin C Sankaranarayanan, Srinivasa G Narasimhan, and Ioannis Gkioulekas. 2019. A theory of fermat paths for non-line-of-sight shape reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Shuang Zhao, Lifan Wu, Frédo Durand, and Ravi Ramamoorthi. 2016. Downsampling scattering parameters for rendering anisotropic media. *ACM Trans. Graph.* 35, 6 (2016), 166:1–166:11.

A DETAILED DERIVATIONS

A.1 Deriving $\partial_\pi \mathcal{K}_\mathrm{T}\,\mathcal{K}_\mathrm{C}\,L$ (16)

According to Eqs. (2) and (5),

$$(\mathcal{K}_\mathrm{T}\,\mathcal{K}_\mathrm{C}\,L)(\boldsymbol{x},\boldsymbol{\omega}) = \int_0^D T(\boldsymbol{x}',\boldsymbol{x})\,\sigma_\mathrm{s}(\boldsymbol{x}')\,L^\mathrm{ins}(\boldsymbol{x}',\boldsymbol{\omega})\,\mathrm{d}\tau, \tag{46}$$

where $\boldsymbol{x}' := \boldsymbol{x} - \tau\boldsymbol{\omega}$. The assumptions stated at the beginning of §4 ensure that the integrand $T\,\sigma_\mathrm{s}\,L^\mathrm{ins}$ is continuous in τ for $0 < \tau <$

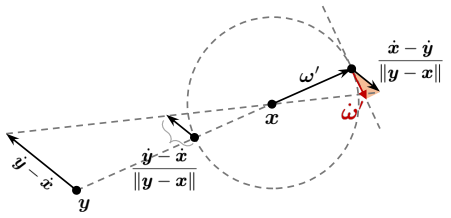

Fig. 14. Calculating the change rate of $\boldsymbol{\omega}'$ defined as the unit vector pointing from $\boldsymbol{y}$ to $\boldsymbol{x}$.

D. Thus, applying Theorem 1, which reduces to Leibniz's rule for differentiation in this case, to Eq. (46) gives

$$\begin{aligned}(\partial_\pi \mathcal{K}_\mathrm{T}\, \mathcal{K}_\mathrm{C}\, L)(\boldsymbol{x},\boldsymbol{\omega}) = &\int_0^D \partial_\pi \left[T(\boldsymbol{x}',\boldsymbol{x})\,\sigma_\mathrm{s}(\boldsymbol{x}')\,L^\mathrm{ins}(\boldsymbol{x}',\boldsymbol{\omega})\right]\mathrm{d}\tau \\ &+ \dot{D}\,T(\boldsymbol{x}_0,\boldsymbol{x})\,\sigma_\mathrm{s}(\boldsymbol{x}_0)\,L^\mathrm{ins}(\boldsymbol{x}_0,\boldsymbol{\omega}),\end{aligned} \tag{47}$$

where $\boldsymbol{x}_0 := \boldsymbol{x} - D\boldsymbol{\omega}$. Further, the derivative of the transmittance $T(\boldsymbol{x}',\boldsymbol{x})$, which we derive in Appendix A.2, equals

$$\dot{T}(\boldsymbol{x}',\boldsymbol{x}) = -T(\boldsymbol{x}',\boldsymbol{x})\,\Sigma_\mathrm{t}(\boldsymbol{x},\boldsymbol{\omega},\tau). \tag{48}$$

Lastly, combining Eqs. (47) and (48) yields Eq. (16) in §4.1.

A.2 Differentiating Transmittance

Multiple terms we derived in §4 involve the derivative of the transmittance between two points $\boldsymbol{x}$ and $\boldsymbol{x} - t\boldsymbol{\omega}$ in the medium given by

$$T(\boldsymbol{x}-t\boldsymbol{\omega},\boldsymbol{x}) = \exp\left(-\int_0^t \sigma_\mathrm{t}(\boldsymbol{x}-\tau\boldsymbol{\omega})\,\mathrm{d}\tau\right). \tag{49}$$

Applying Theorem 1 to this equation leads to

$$\begin{aligned}\dot{T}(\boldsymbol{x}-t\boldsymbol{\omega},\boldsymbol{x}) &= \partial_\pi \exp\left(-\int_0^t \sigma_\mathrm{t}(\boldsymbol{x}-\tau\boldsymbol{\omega})\,\mathrm{d}\tau\right) \\ &= -T(\boldsymbol{x}-t\boldsymbol{\omega},\boldsymbol{x})\,\partial_\pi \int_0^t \sigma_\mathrm{t}(\boldsymbol{x}-\tau\boldsymbol{\omega})\,\mathrm{d}\tau \\ &= -T(\boldsymbol{x}-t\boldsymbol{\omega},\boldsymbol{x})\left(\Sigma_\mathrm{t}(\boldsymbol{x},\boldsymbol{\omega},t) + \dot{t}\,\sigma_\mathrm{t}(\boldsymbol{x}-t\boldsymbol{\omega})\right),\end{aligned} \tag{50}$$

where Σ_t is defined in Eq. (17) as $\Sigma_\mathrm{t}(\boldsymbol{x},\boldsymbol{\omega},t) = \int_0^t \dot{\sigma}_\mathrm{t}(\boldsymbol{x}-\tau\boldsymbol{\omega})\,\mathrm{d}\tau$.

A.3 Deriving the Change Rate of $\boldsymbol{\omega}'$ (25)

Let $\boldsymbol{x}$ and $\boldsymbol{y}$ be two 3D locations. Then, the change rate of $\boldsymbol{\omega}' := (\boldsymbol{y} \to \boldsymbol{x})$ with respect to a scene parameter π can be calculated as follows (see Figure 14): (i) project $(\dot{\boldsymbol{y}} - \dot{\boldsymbol{x}})$ to the surface of a unit sphere centered at $\boldsymbol{x}$, yielding $(\dot{\boldsymbol{y}}-\dot{\boldsymbol{x}})/\|\boldsymbol{y}-\boldsymbol{x}\|$; (ii) invert the projected velocity to align it with $\boldsymbol{\omega}'$, yielding $(\dot{\boldsymbol{x}} - \dot{\boldsymbol{y}})/\|\boldsymbol{y} - \boldsymbol{x}\|$; (iii) take the component (illustrated in red) of the inverted velocity that is within the tangent plane at $\boldsymbol{\omega}'$, yielding $\dot{\boldsymbol{\omega}}'$ expressed in Eq. (25).

A.4 Evaluating ΔL Defined in Eq. (26)

Given a point $\boldsymbol{x}$ interior of the medium, the discontinuities of $L(\boldsymbol{x},\boldsymbol{\omega})$ in $\boldsymbol{\omega}$, in the absence of zero-measure light sources and ideal specular surfaces, generally arise when $\boldsymbol{\omega} = (\boldsymbol{y} \to \boldsymbol{x})$ with $\boldsymbol{y}$ located on one of the boundary, silhouette, and sharp edges (as illustrated in Figure 5). In what follows, we derive ΔL based on the type of edge on which $\boldsymbol{y}$ resides.

Case 1. Boundary and silhouette edges can cause the following discontinuities with respect to $\boldsymbol{\omega}$:

- The distance D for the ray $(\boldsymbol{x}, -\boldsymbol{\omega})$ to hit the medium boundary, which is involved in the transport operator $\mathcal{K}_\mathrm{T}$ (2).
- The location $\boldsymbol{x}_0 = \boldsymbol{x} - D\boldsymbol{\omega}$ where the ray intersects the boundary, which is needed by the source term Q (6).

For each $\boldsymbol{\omega}$ where D and $\boldsymbol{x}_0$ are discontinuous, let

$$D^- := \lim_{\epsilon\to 0^-} D(\boldsymbol{x},\boldsymbol{\omega}+\epsilon\,\boldsymbol{n}), \quad D^+ := \lim_{\epsilon\to 0^+} D(\boldsymbol{x},\boldsymbol{\omega}+\epsilon\,\boldsymbol{n}), \tag{51}$$

where $\boldsymbol{x}^- := \boldsymbol{x} - D^-\boldsymbol{\omega}$ and $\boldsymbol{x}^+ := \boldsymbol{x} - D^+\boldsymbol{\omega}$ denote the one-sided limits of D and $\boldsymbol{x}_0$ (see Figure 15). Then, $\Delta L(\boldsymbol{x},\boldsymbol{\omega})$ is given by the difference of the interior radiance L evaluated with $D = D^-$, $\boldsymbol{x}_0 = \boldsymbol{x}^-$ and $D = D^+$, $\boldsymbol{x}_0 = \boldsymbol{x}^+$. Precisely,

$$\begin{aligned}&\Delta L(\boldsymbol{x},\boldsymbol{\omega}) \\ &= \left[\int_0^{D^-} T(\boldsymbol{x}',\boldsymbol{x})\,(\mathcal{K}_\mathrm{C}\,L)(\boldsymbol{x}',\boldsymbol{\omega})\,\mathrm{d}\tau + T(\boldsymbol{x}^-,\boldsymbol{x})\,L_\mathrm{s}(\boldsymbol{x}^-,\boldsymbol{\omega})\right] - \\ &\quad \left[\int_0^{D^+} T(\boldsymbol{x}',\boldsymbol{x})\,(\mathcal{K}_\mathrm{C}\,L)(\boldsymbol{x}',\boldsymbol{\omega})\,\mathrm{d}\tau + T(\boldsymbol{x}^+,\boldsymbol{x})\,L_\mathrm{s}(\boldsymbol{x}^-,\boldsymbol{\omega})\right] \\ &= \int_{D^+}^{D^-} T(\boldsymbol{x}',\boldsymbol{x})\,\sigma_\mathrm{s}(\boldsymbol{x}')\,L^\mathrm{ins}(\boldsymbol{x}',\boldsymbol{\omega})\,\mathrm{d}\tau \\ &\quad + T(\boldsymbol{x}^-,\boldsymbol{x})\,L_\mathrm{s}(\boldsymbol{x}^-,\boldsymbol{\omega}) - T(\boldsymbol{x}^+,\boldsymbol{x})\,L_\mathrm{s}(\boldsymbol{x}^+,\boldsymbol{\omega}).\end{aligned} \tag{52}$$

Notice that it is possible for $D^+ > D^-$. In this case, the integral from D^+ to D^- in Eq. (52) takes a negative value.

Case 2. Another type of L's discontinuity in $\boldsymbol{\omega}$ occurs when $\boldsymbol{x}_0$ lies on a *sharp edge*. In this case, the surface normal $\boldsymbol{n}$ at $\boldsymbol{x}_0$ has jump discontinuities (despite D and $\boldsymbol{x}_0$ being continuous themselves). This makes ΔL to be entirely determined by the difference of the source term Q evaluated with the two different surface normals, denoted as $\boldsymbol{n}^-$ and $\boldsymbol{n}^+$, from the two faces sharing the sharp edge. With the volumetric emission L^e neglected, $Q(\boldsymbol{x},\boldsymbol{\omega}) = T(\boldsymbol{x}_0,\boldsymbol{x})\,L_\mathrm{s}(\boldsymbol{x}_0,\boldsymbol{\omega})$ and

$$\begin{aligned}&\Delta L(\boldsymbol{x},\boldsymbol{\omega}) = T(\boldsymbol{x}_0,\boldsymbol{x})\,\Delta L_\mathrm{s}(\boldsymbol{x}_0,\boldsymbol{\omega}) \\ &= T(\boldsymbol{x}_0,\boldsymbol{x})\left(\int_{\mathbb{S}^2} \Delta f_\mathrm{s}(\boldsymbol{x}_0,-\boldsymbol{\omega}',\boldsymbol{\omega})\,L(\boldsymbol{x}_0,\boldsymbol{\omega}')\,\mathrm{d}\boldsymbol{\omega}' + \Delta L_\mathrm{s}^\mathrm{e}(\boldsymbol{x}_0,\boldsymbol{\omega})\right),\end{aligned} \tag{53}$$

where Δf_s and $\Delta L_\mathrm{s}^\mathrm{e}$ denote the difference between the BSDFs and emitted radiances evaluated with the normals $\boldsymbol{n}^-$ and $\boldsymbol{n}^+$ (see Figure 16).

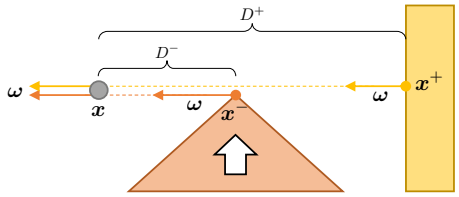

Fig. 15. Calculating ΔL resulting from a silhouette edge. In this 2D example, the orange triangle is translating upward with a constant velocity relative to $\boldsymbol{x}$ (while the yellow box stays stationary), and its top vertex creates a geometric discontinuity when viewed from $\boldsymbol{x}$. Let $\boldsymbol{\omega}'$ denote the direction corresponding to this discontinuity (i.e., the ray $(\boldsymbol{x}, -\boldsymbol{\omega}')$ intersects the orange triangle at its top vertex). Based on these observations, D^-, D^+, $\boldsymbol{x}^-$, and $\boldsymbol{x}^+$ defined in Eq. (51) are illustrated.

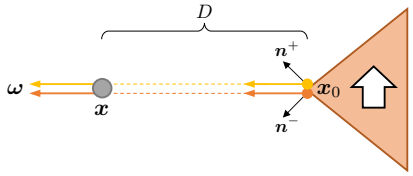

Fig. 16. Calculating ΔL resulting from a sharp edge. In this case, ΔL is solely determined by the difference between the BSDF and the surface emission at $\boldsymbol{x}_0$ evaluated at the two normals $\boldsymbol{n}^-$ and $\boldsymbol{n}^+$, as expressed in Eq. (53).

B DETAILS ON OUR MONTE CARLO ESTIMATOR

We now provide more details on the estimation of radiance change ΔL and in-scattered radiance L^{ins}.

Estimating ΔL. Computing the edge integrals (Lines 14 and 26 of Algorithm 1) requires estimating the change of radiance ΔL across discontinuity boundaries (Line 5 of Algorithm 2).

Recall that the discontinuity of L, under the assumptions stated at the beginning of §4, arises due to the geometric discontinuities resulting from boundary, silhouette, and sharp edges (Figure 5). Based on this observation, Eqs. (52) and (53) derived in Appendix A.4 evaluate ΔL under two cases. These equations can further be estimated using Monte Carlo integration, as shown in Algorithm 3.

Discontinuities caused by boundary and silhouette edges (52) are handled by Lines 3–18 of Algorithm 3. Let $D^{\text{near}} := \min(D^-, D^+)$ and $D^{\text{far}} := \max(D^-, D^+)$. Then, due to Eq. (52), it holds that

$$\begin{aligned} &|\Delta L(\boldsymbol{x}, \boldsymbol{\omega})| \\ &= \int_{D^{\text{near}}}^{D^{\text{far}}} T(\boldsymbol{x}', \boldsymbol{x})\, \sigma_{\text{s}}(\boldsymbol{x}')\, L^{\text{ins}}(\boldsymbol{x}', \boldsymbol{\omega})\, \mathrm{d}\tau + T(\boldsymbol{x}^{\text{far}}, \boldsymbol{x})\, L_{\text{s}}(\boldsymbol{x}^{\text{far}}, \boldsymbol{\omega}) \\ &\quad - T(\boldsymbol{x}^{\text{near}}, \boldsymbol{x})\, L_{\text{s}}(\boldsymbol{x}^{\text{near}}, \boldsymbol{\omega}), \end{aligned} \tag{54}$$

where $\boldsymbol{x}^{\text{near}} := \boldsymbol{x} - D^{\text{near}}\boldsymbol{\omega}$ and $\boldsymbol{x}^{\text{far}} := \boldsymbol{x} - D^{\text{far}}\boldsymbol{\omega}$. After obtaining $|\Delta L|$, it is easy to verify that ΔL equals $|\Delta L|$ if $D^- > D^+$ and $-|\Delta L|$ otherwise. Further, Eq. (54) can be estimated by tracing two side paths: one for the first two terms on the RHS side (Lines 9–14) and the other for the last term (Line 6).

Lastly, discontinuities caused by sharp edges (53) can be handled by tracing only one side path, as shown in Lines 3–18 of Algorithm 3.

Estimating L^{ins}. To estimate $L^{\text{ins}}(\boldsymbol{x}, \boldsymbol{\omega})$ given by Eq. (5) for interior points $\boldsymbol{x} \in \Omega \setminus \partial\Omega$ and Eq. (20) for boundary ones $\boldsymbol{x} \in \partial\Omega$, we randomly draw a direction $\boldsymbol{\omega}'$ according to the phase function at $\boldsymbol{x}$ and estimate $L(\boldsymbol{x}, \boldsymbol{\omega}')$ or $L_{\text{s}}(\boldsymbol{x}, \boldsymbol{\omega}')$ using standard volumetric path tracing. Algorithm 4 contains the pseudocode for this process.

Algorithm 3 Monte Carlo estimation of ΔL

1: **function** DELTARADIANCE($\boldsymbol{x}$, $\dot{\boldsymbol{x}}$, $\boldsymbol{\omega}$, $\dot{\boldsymbol{\omega}}$)
2: **if** $\boldsymbol{\omega}$ corresponds to a boundary/silhouette edge **then** ▷ (52)
3: Calculate D^-, D^+ using Eq. (51)
4: $D^{\text{near}} \leftarrow \min(D^-, D^+)$, $D^{\text{far}} \leftarrow \max(D^-, D^+)$
5: $\boldsymbol{x}^{\text{near}} \leftarrow \boldsymbol{x} - D_{\text{near}}\,\boldsymbol{\omega}$, $\boldsymbol{x}^{\text{far}} \leftarrow \boldsymbol{x} - D_{\text{far}}\,\boldsymbol{\omega}$
6: $\Delta L \leftarrow -L_{\text{s}}(\boldsymbol{x}^{\text{near}}, \boldsymbol{\omega})$ ▷ side path
7: Draw free-flight distance τ from $\boldsymbol{x}^{\text{near}}$ with direction $\boldsymbol{\omega}$
8: $\tau \leftarrow \tau + D^{\text{near}}$
9: **if** $\tau < D^{\text{far}}$ **then**
10: $\boldsymbol{x}' \leftarrow \boldsymbol{x} - \tau\boldsymbol{\omega}$
11: $\Delta L \leftarrow \Delta L + \frac{\sigma_{\text{s}}(\boldsymbol{x}')}{\sigma_{\text{t}}(\boldsymbol{x}')}$ INSCATRADIANCE($\boldsymbol{x}', \boldsymbol{\omega}$) ▷ Alg. 4
12: **else**
13: $\Delta L \leftarrow \Delta L + L_{\text{s}}(\boldsymbol{x}^{\text{far}}, \boldsymbol{\omega})$ ▷ side path
14: **end if**
15: $\Delta L \leftarrow T(\boldsymbol{x}^{\text{near}}, \boldsymbol{x})\,\Delta L$
16: **if** $D^- < D^+$ **then**
17: $\Delta L \leftarrow -\Delta L$
18: **end if**
19: **else** ▷ (53)
20: Calculate $\boldsymbol{n}^-$ and $\boldsymbol{n}^+$
21: $\Delta L \leftarrow L_{\text{s}}^{\text{e}}(\boldsymbol{x}_0, \boldsymbol{\omega} \mid \boldsymbol{n}^-) - L_{\text{s}}^{\text{e}}(\boldsymbol{x}_0, \boldsymbol{\omega} \mid \boldsymbol{n}^+)$
22: Draw $\boldsymbol{\omega}'$ with probability $p(\boldsymbol{\omega}')$
23: $\Delta f_{\text{s}} \leftarrow f_{\text{s}}(\boldsymbol{x}_0, -\boldsymbol{\omega}', \boldsymbol{\omega} \mid \boldsymbol{n}^-) - f_{\text{s}}(\boldsymbol{x}_0, -\boldsymbol{\omega}', \boldsymbol{\omega} \mid \boldsymbol{n}^+)$
24: $\Delta L \leftarrow \Delta L + L_{\text{s}}(\boldsymbol{x}, \boldsymbol{\omega}')\,\Delta f_{\text{s}}/p(\boldsymbol{\omega}')$ ▷ side path
25: **end if**
26: **return** ΔL
27: **end function**

Algorithm 4 Monte Carlo estimation of L^{ins}

1: **function** INSCATRADIANCE($\boldsymbol{x}$, $\boldsymbol{\omega}$)
2: **if** $\boldsymbol{x} \in \Omega \setminus \partial\Omega$ **then** ▷ (5)
3: Draw $\boldsymbol{\omega}' \sim f_{\text{p}}(\boldsymbol{x}, -\boldsymbol{\omega}', \boldsymbol{\omega})$
4: $L^{\text{ins}} \leftarrow L(\boldsymbol{x}, \boldsymbol{\omega}')$ ▷ side path
5: **else** ▷ (20)
6: Draw $\boldsymbol{\omega}' \sim f_{\text{p}}(\boldsymbol{x} + \epsilon\,\boldsymbol{\omega}, -\boldsymbol{\omega}', \boldsymbol{\omega})$ ▷ $\epsilon \to 0^+$
7: **if** $\langle \boldsymbol{n}(\boldsymbol{x}), \boldsymbol{\omega}' \rangle > 0$ **then**
8: $L^{\text{ins}} \leftarrow L(\boldsymbol{x}, \boldsymbol{\omega}')$ ▷ side path
9: **else**
10: $L^{\text{ins}} \leftarrow L_{\text{s}}(\boldsymbol{x}, \boldsymbol{\omega}')$ ▷ side path
11: **end if**
12: **end if**
13: **return** L^{ins}
14: **end function**