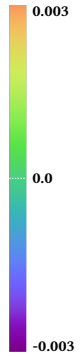
Path-Space Differentiable Rendering

CHENG ZHANG, University of California, Irvine
BAILEY MILLER, Carnegie Mellon University
KAI YAN, University of California, Irvine
IOANNIS GKIOULEKAS, Carnegie Mellon University
SHUANG ZHAO, University of California, Irvine



Original **Derivative** with respect to sun location

Fig. 1. We introduce **path-space differentiable rendering**, a new theoretical framework to estimate derivatives of radiometric measurements with respect to arbitrary scene parameters (e.g., material properties and object geometries). By directly differentiating full path integrals, we derive the *differential path integral* framework, enabling the design of new unbiased Monte Carlo methods capable of efficiently estimating derivatives in virtual scenes with complex geometry and light transport effects. This example shows a dinning room scene lit by the sun from outside the window. On the right, we show the corresponding derivative image with respect to the vertical location of the sun. (Please use Adobe Acrobat to view the teaser images to see them animated.)

Physics-based differentiable rendering, the estimation of derivatives of radiometric measures with respect to arbitrary scene parameters, has a diverse array of applications from solving analysis-by-synthesis problems to training machine learning pipelines incorporating forward rendering processes. Unfortunately, general-purpose differentiable rendering remains challenging due to the lack of efficient estimators as well as the need to identify and handle complex discontinuities such as visibility boundaries.

In this paper, we show how path integrals can be differentiated with respect to arbitrary differentiable changes of a scene. We provide a detailed theoretical analysis of this process and establish new differentiable rendering formulations based on the resulting differential path integrals. Our path-space differentiable rendering formulation allows the design of new Monte Carlo estimators that offer significantly better efficiency than state-of-the-art methods in handling complex geometric discontinuities and light transport phenomena such as caustics.

We validate our method by comparing our derivative estimates to those generated using the finite-difference method. To demonstrate the effectiveness of our technique, we compare inverse-rendering performance with a few state-of-the-art differentiable rendering methods.

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

Additional Key Words and Phrases: Differentiable rendering, path integral, Monte Carlo rendering

ACM Reference Format:
Cheng Zhang, Bailey Miller, Kai Yan, Ioannis Gkioulekas, and Shuang Zhao. 2020. Path-Space Differentiable Rendering. *ACM Trans. Graph.* 39, 4, Article 143 (July 2020), 19 pages. https://doi.org/10.1145/3386569.3392383

Authors' addresses: Cheng Zhang, University of California, Irvine, chengz20@uci.edu; Bailey Miller, Carnegie Mellon University, baileymark.miller@gmail.com; Kai Yan, University of California, Irvine, kyan8@uci.edu; Ioannis Gkioulekas, Carnegie Mellon University, igkioule@andrew.cmu.edu; Shuang Zhao, University of California, Irvine, shz@ics.uci.edu.

0730-0301/2020/7-ART143 $15.00
https://doi.org/10.1145/3386569.3392383

1 INTRODUCTION

Physics-based light transport simulation, a core research topic in computer graphics since the field's inception, focus on numerically estimating radiometric sensor responses in fully specified virtual scenes. Previous research efforts have led to mature *forward rendering* algorithms that can efficiently and accurately simulate light transport in virtual environments with high complexities.

Differentiable rendering computes the derivatives of radiometric measurements with respect to differential changes of such environments. These techniques can enable, for example: (i) *gradient-based optimization* when solving *inverse-rendering* problems; and (ii) efficient integration of physics-based light transport simulation in *machine learning* and *probabilistic inference* pipelines.

Unfortunately, unlike forward rendering, differentiable rendering remains challenging. One key challenge is the lack of efficient Monte Carlo estimation techniques. In the forward case, the *path integral formulation* introduced by Veach [1997] has opened the door to the design of sophisticated rendering algorithms including bidirectional path tracing and many Markov-Chain Monte Carlo (MCMC) rendering techniques (e.g., [Jakob and Marschner 2012; Pauly et al. 2000; Veach and Guibas 1997]). Similar formulations are lacking for differentiable rendering, causing state-of-the-art techniques [Li et al. 2018a; Loubet et al. 2019; Zhang et al. 2019] to rely on unidirectional path tracing, which has been demonstrated to be inefficient in handling complex light transport effects such as indirect-dominated illumination and caustics.

Another challenge, which is unique to differentiable rendering, is the need to handle discontinuities via edge or boundary integrals. Previously, this was handled by either tracing expensive "side paths" [Li et al. 2018a; Zhang et al. 2019] or using approximate reparameterizations that introduce bias [Loubet et al. 2019]. To our knowledge, there does not exist any prior solution to this problem that is both efficient and unbiased.

In this paper, we address both challenges by (i) introducing the *differential path integral* formulation, the differentiable-rendering counterpart of the *path integral* for forward rendering; and (ii) providing comprehensive theoretical and empirical analysis.

Concretely, our contributions include:

- The differentiation of full path integrals with respect to arbitrary scene parameterizations (§5.1), resulting in our *differential path integrals* comprised of completely separated *interior* and *boundary* components that can be estimated independently using different Monte Carlo estimators.
- A reparameterization of the path integral (§5.2) that minimizes the types of discontinuities to be handled by the *boundary* integral.
- New unbiased Monte Carlo methods that estimate, respectively, the *interior* and *boundary* components of our *differential path integrals* (§6). Our technique greatly outperforms previous methods for complex scene geometries and light transport effects.

To facilitate the derivation of our main results in §5 and §6, we utilize mathematical tools from continuum and fluid mechanics [Cermelli et al. 2005; Gurtin 1981], which we briefly review in §3, for their generality and rigor. Additionally, as a warm-up, we apply these tools to differentiate direct-illumination integrals in §4.

To validate our theory and algorithms, we compare our derivative estimates with those produced using finite differences (Figures 12 and 13). To demonstrate the effectiveness of our method, we compare (i) derivative images generated with our technique and state-of-the-art approaches (Figures 14 and 17); and (ii) inverse-rendering performance using gradients estimated with these methods (Figures 15, 16, and 18).

2 RELATED WORK

Path-space rendering. Veach [1997] introduced the *path integral* formulation arising from recursively expanding the rendering equation [Kajiya 1986]. This formulation expresses radiometric measurements as high-dimensional integrals (instead of solutions to integral equations), enabling the development of many new Monte Carlo estimators (e.g., [Jakob and Marschner 2012; Veach and Guibas 1995, 1997]) that are capable of efficiently simulating challenging effects such as indirect illumination and near-specular transport. In this paper, we introduce a *differential path integral* formulation for differentiable rendering.

Derivatives for rendering. Analytical derivatives have been used in forward rendering to compute pixel footprints [Igehy 1999], handle specular light paths [Chen and Arvo 2000; Jakob and Marschner 2012], use Hamiltonian Monte Carlo to sample paths [Li et al. 2015], and enable interactive editing of single-scattering albedo [Hašan and Ramamoorthi 2013]. Arvo [1994] presented an analytical method for calculating the gradients of irradiance in diffuse scenes. Ramamoorthi et al. [2007] introduced a first-order analysis of light transport, focusing on effects such as soft shadows. All these derivatives are specialized for certain types of light transport effects, and most of them neglect geometric discontinuities.

Physics-based differentiable rendering. Differentiable rendering for specific light transport effects has been used to solve analysis-by-synthesis problems in volumetric scattering [Gkioulekas et al. 2016, 2013], 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].

A main challenge towards developing general-purpose differentiable rendering engines has been the differentiation with respect to scene geometry, which generally requires calculating additional boundary integrals. To address this problem, Li et al. [2018a] introduced a Monte Carlo edge-sampling method that provides unbiased estimates of these boundary integrals. This technique was then generalized by Zhang et al. [2019] to handle volumetric light transport. Concurrently, Loubet et al. [2019] proposed a reparameterization-based method to avoid computing boundary integrals altogether, at the cost of introducing bias. Despite their ability to differentiate with respect to arbitrary scene parameterizations, all these methods are obtained by differentiating the rendering equation [Kajiya 1986] (and the radiative transfer equation [Chandrasekhar 1960]), and rely on unidirectional path tracing for derivative estimations, which can be inefficient when handling complex scenes.

Derivatives for vision. Having derivatives of rendered images allows physics-based rendering to be efficiently integrated into deep learning pipelines (e.g., as the decoder of an auto-encoder architecture [Che et al. 2018]). Many recent works utilize various forms of rendering losses to regularize the training and improve generalization of neural network models [Che et al. 2018; Kato et al. 2018; Li et al. 2018b; Meka et al. 2018; Sengupta et al. 2018; Wu et al. 2017]. The renderers used in most of these works make restrictive simplifications such as single-bounce illumination [Loper and Black 2014].

Automatic differentiation. Automatic differentiation allows the derivative of a function specified by a computer program to be evaluated numerically. These techniques have been widely used in machine learning and statistical inference [Griewank and Walther

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

Symbol	Definition
f	measurement contribution
Ω	path space
$\partial\Omega$	boundary path space
μ	area-product measure
μ'	differential area-product measure
$\mathcal{M}(\pi)$	evolving surface
$\mathcal{B}$	reference configuration
$\boldsymbol{p}$	material point
X	motion
P	reference map
$\hat{\boldsymbol{x}}$	(local or global) surface parameterization
$\boldsymbol{n}(\boldsymbol{x})$	unit normal of a surface at $\boldsymbol{x}$
$\boldsymbol{n}_{\partial\mathcal{A}}(\boldsymbol{x})$	unit normal of a curve $\partial\mathcal{A}$ at $\boldsymbol{x}$
$V(\boldsymbol{x})$	scalar normal velocity of a surface at $\boldsymbol{x}$
$V_{\partial\mathcal{A}}(\boldsymbol{x})$	scalar normal velocity of a curve at $\boldsymbol{x}$
$\Delta\mathcal{M}[\varphi](\pi)$	discontinuity curves wrt. φ in $\mathcal{M}(\pi)$
$\overline{\partial\mathcal{M}}[\varphi](\pi)$	extended boundary of $\mathcal{M}(\pi)$
$()^{\bullet}$	scene derivative
$()^{\square}$	normal scene derivative

2008; McClelland et al. 1986; Wengert 1964] to obtain gradients of complex functions (e.g., neural networks). Most general-purpose differentiable rendering techniques, including ours, utilize automated differentiation. On the other hand, our main theory and algorithms are orthogonal to the choice of automated differentiation method and can benefit greatly from efficient implementations (e.g., [Nimier-David et al. 2019]).

3 PRELIMINARIES

A main objective of this paper is to differentiate full path integrals, which we review in §3.1, with respect to arbitrary differential changes of the scene. As a path integral is comprised of nested surface integrals, its differentiation largely boils down to calculating derivatives of surface integrals. To this end, we utilize mathematical tools from continuum and fluid mechanics to (i) express the evolution of surfaces; and (ii) differentiate integrals over evolving surfaces. We provide a brief recap of these preliminaries in §3.2, §3.3, and the supplemental document.

Table 1 summarizes commonly used symbols and their definitions.

3.1 The Path-Integral Formulation

We will be focusing on Veach's *path-integral formulation* [1997] of light transport, which has been widely used in physics-based rendering, where a radiometric measurement I is expressed as a **path integral** of the form:

$$I = \int_\Omega f(\bar{\boldsymbol{x}})\,\mathrm{d}\mu(\bar{\boldsymbol{x}}). \tag{1}$$

In this integral, each **light path** $\bar{\boldsymbol{x}} = (\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$ with $N \geq 1$ is an ordered sequence of points $\boldsymbol{x}_n \in \mathcal{M}$ for $n = 0, \ldots, N$, where $\mathcal{M}$ is the union of all object surfaces in the scene. The integral is performed over the **path space** $\Omega := \bigcup_{N=1}^{\infty} \mathcal{M}^{N+1}$, and with respect to the **area-product measure** μ defined as:

$$\mathrm{d}\mu(\bar{\boldsymbol{x}}) := \prod_{n=0}^{N} \mathrm{d}A(\boldsymbol{x}_n), \tag{2}$$

where A is the surface-area measure. We will additionally be considering **order-N path integrals** of the form:

$$I_N = \int_{\Omega_N} f(\bar{\boldsymbol{x}})\,\mathrm{d}\mu(\bar{\boldsymbol{x}}), \tag{3}$$

where the integration domain $\Omega_N := \mathcal{M}^{N+1}$ is constrained to only light paths[1] with N segments and $(N+1)$ vertices. Then, it follows that the full path integral equals $I = \sum_{N=1}^{\infty} I_N$.

In both integrals (1) and (3), the **measurement contribution function** f captures the amount of radiant power carried by individual light paths, and equals

$$f(\bar{\boldsymbol{x}}) = \left(\prod_{n=0}^{N-1} g(\boldsymbol{x}_{n+1};\, \boldsymbol{x}_{n-1}, \boldsymbol{x}_n)\right) W_\mathrm{e}(\boldsymbol{x}_N \to \boldsymbol{x}_{N-1}). \tag{4}$$

In this equation, W_e is the **sensor importance** (response), and

$$g(\boldsymbol{x}_{n+1};\, \boldsymbol{x}_{n-1}, \boldsymbol{x}_n) := f_\mathrm{s}(\boldsymbol{x}_{n-1} \to \boldsymbol{x}_n \to \boldsymbol{x}_{n+1})\, G(\boldsymbol{x}_n \leftrightarrow \boldsymbol{x}_{n+1}), \tag{5}$$

for $0 \leq n < N$. In Eq. (5), f_s is the **bidirectional scattering distribution function (BSDF)** when $n > 0$, and the **source emission** when $n = 0$ (i.e., $f_\mathrm{s}(\boldsymbol{x}_{-1} \to \boldsymbol{x}_0 \to \boldsymbol{x}_1) := L_\mathrm{e}(\boldsymbol{x}_0 \to \boldsymbol{x}_1)$ with $\boldsymbol{x}_{-1}$ being a dummy variable); and G is the **geometric term**:

$$G(\boldsymbol{x}_n \leftrightarrow \boldsymbol{x}_{n+1}) = \mathbb{V}(\boldsymbol{x}_n \leftrightarrow \boldsymbol{x}_{n+1})\, G_0(\boldsymbol{x}_n \leftrightarrow \boldsymbol{x}_{n+1}), \tag{6}$$

where $\mathbb{V}$ is the **mutual visibility function** (which equals one if $\boldsymbol{x}_n$ and $\boldsymbol{x}_{n+1}$ are mutually visible and zero otherwise), and G_0 is the **visibility-free** component of G:

$$G_0(\boldsymbol{x}_n \leftrightarrow \boldsymbol{x}_{n+1}) := \frac{|\boldsymbol{n}(\boldsymbol{x}_n) \cdot \boldsymbol{\omega}_n|\, |\boldsymbol{n}(\boldsymbol{x}_{n+1}) \cdot -\boldsymbol{\omega}_n|}{\|\boldsymbol{x}_{n+1} - \boldsymbol{x}_n\|^2}. \tag{7}$$

In Eq. (7), $\boldsymbol{\omega}_n = \boldsymbol{x}_n \to \boldsymbol{x}_{n+1} := (\boldsymbol{x}_{n+1} - \boldsymbol{x}_n)/\|\boldsymbol{x}_{n+1} - \boldsymbol{x}_n\|$ is the unit vector pointing from $\boldsymbol{x}_n$ toward $\boldsymbol{x}_{n+1}$; $\boldsymbol{n}$ is the unit-normal field; and "$\cdot$" indicates vector inner product.

The path-integral formulation has been generalized to also incorporate volumetric light transport [Pauly et al. 2000], but we omit these details as we will be focusing on the surface-only case in the rest of this paper.

3.2 Surface Evolution and Scene Derivatives

We now briefly review some concepts developed in continuum and fluid mechanics, which we will use to mathematically describe the differentiable evolution of surfaces through the three-dimensional Euclidean space $\mathbb{R}^3$. For more details, please refer to the supplemental document and Gurtin [1981].

Evolving surfaces. Let $\mathcal{B}$ be some abstract 2D manifold that we call the **reference configuration**. A **deformation**[2] of $\mathcal{B}$ is a smooth and one-to-one function that maps $\mathcal{B}$ to some regular surface $\mathcal{M}$.

We focus on parametric families of deformations and call each such family a **motion** of $\mathcal{B}$, which, formally, is a class C^3 function X defined on $\mathcal{B} \times \mathbb{R}$ such that, for any fixed parameter value $\pi \in \mathbb{R}$, $\mathsf{X}(\cdot, \pi)$ is a deformation of $\mathcal{B}$.[3]

[1] We hyperlink many keywords to their definitions.
[2] Deformations in classic continuum mechanics operate on volumes. In this paper, we use definitions adapted for surfaces by Cermelli et al. [2005] in fluid mechanics.
[3] In physics, the parameter π is typically used to represent time; in our case, on the contrary, π is an abstract parameter controlling the global scene geometry.

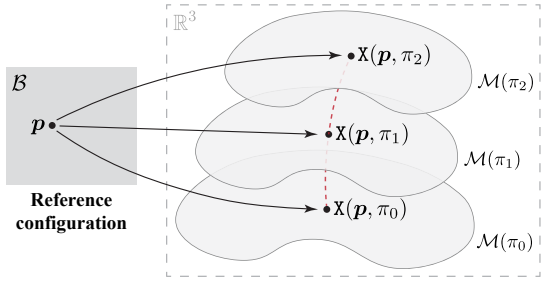


Fig. 2. **Deformation, motion, and evolving surface:** A motion X, for each $\pi \in \mathbb{R}$, provides a deformation $\mathsf{X}(\cdot, \pi)$ that maps a reference configuration $\mathcal{B}$ continuously to a regular surface $\mathcal{M}(\pi) \subset \mathbb{R}^3$. The red curve illustrates the trajectory of a single point (shown as the red dashed curve) as the images of a material point $\boldsymbol{p} \in \mathcal{B}$.

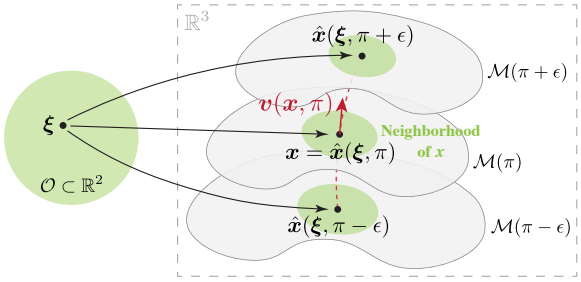


Fig. 3. **Local velocity:** Provided a surface parameterization $\hat{\boldsymbol{x}}$, for any π and $\boldsymbol{x} \in \mathcal{M}(\pi)$, assume $\boldsymbol{x}$ to have local coordinates $\boldsymbol{\xi}$. Then, with $\boldsymbol{\xi}$ fixed, $\hat{\boldsymbol{x}}(\boldsymbol{\xi}, \cdot)$ produces the local trajectory of $\boldsymbol{x}$ near π (illustrated as the red dashed curve), whose derivative with respect to π gives the corresponding local velocity $\boldsymbol{v}(\boldsymbol{x}, \pi)$.

Provided a motion X, each deformation $\mathsf{X}(\cdot, \pi)$ maps the reference $\mathcal{B}$ to the **evolving surface** $\mathcal{M}(\pi) \coloneqq \{\mathsf{X}(\boldsymbol{p}, \pi) : \boldsymbol{p} \in \mathcal{B}\} \subset \mathbb{R}^3$, as illustrated in Figure 2. As deformations are assumed one-to-one, $\mathsf{X}(\cdot, \pi)$ has an inverse $\mathsf{P}(\cdot, \pi) : \mathcal{M}(\pi) \mapsto \mathcal{B}$ that is called a **reference map** and transforms the evolving surface $\mathcal{M}(\pi)$ back to the reference $\mathcal{B}$.

Lastly, we call the set $\mathcal{T} \coloneqq \{(\boldsymbol{x}, \pi) : \boldsymbol{x} \in \mathcal{M}(\pi),\ \pi \in \mathbb{R}\} \subset \mathbb{R}^4$ the **trajectory** of $\mathcal{B}$ resulting from the motion X. We note that, given a reference $\mathcal{B}$, there may exist (infinitely) many X leading to identical trajectories.

Material and spatial representations. Following the convention in continuum mechanics, we introduce the following terminology:

- We call $\boldsymbol{p} \in \mathcal{B}$ a **material point** and $\boldsymbol{x} \in \mathcal{M}(\pi)$ a **spatial point**. Given a motion X and $\pi \in \mathbb{R}$, the deformation $\mathsf{X}(\cdot, \pi)$ establishes a continuous and one-to-one mapping from material points to spatial ones.
- For a given motion X, a **material field** is a function of the material point and parameter π with domain $\mathcal{B} \times \mathbb{R}$; a **spatial field**, on the contrary, is a function defined on the trajectory $\mathcal{T}$.

Throughout the paper, we will often encounter situations where a quantity can be defined with respect to either the evolving surface $\mathcal{M}(\pi)$, or the reference $\mathcal{B}$; we will use the terms "*spatial*" and "*material*", respectively, to distinguish between these definitions.

Additionally, the deformation $\mathsf{X}(\cdot, \pi)$ can be used as a change of variables to convert surface integrals defined over the evolving surface $\mathcal{M}(\pi)$ to surface integrals over the reference $\mathcal{B}$ (and vice-versa). To distinguish between the two parameterizations, in the following we will refer to such integrals as **spatial-form integrals** and **material-form integrals**. We will use the same terminology for integrals whose domain is defined based on the evolving surface $\mathcal{M}(\pi)$, or the reference $\mathcal{B}$—for instance, when the path space Ω is defined with respect to an evolving surface $\mathcal{M}(\pi)$, we will refer to the path integral of Eq. (1) as the **spatial-form path integral**.

Surface parameterizations. In order to define derivatives of an evolving surface $\mathcal{M}(\pi)$ with respect to the parameter π, we will need to first have available a parameterization of $\mathcal{M}(\pi)$.

We can parameterize the evolving surface $\mathcal{M}(\pi)$ *locally* based on the corresponding trajectory $\mathcal{T}$: A **local parameterization** near fixed $(\boldsymbol{x}, \pi) \in \mathcal{T}$ takes the form[4] of $\hat{\boldsymbol{x}}(\boldsymbol{\xi}, \pi')$ that, for some open $\mathcal{O} \subset \mathbb{R}^2$, satisfies the following conditions:

- $\hat{\boldsymbol{x}}(\mathcal{O}, \pi) \subset \mathcal{M}(\pi)$ is a neighborhood of $\boldsymbol{x}$.
- For each fixed π' near π, $\hat{\boldsymbol{x}}(\cdot, \pi')$ is a smooth and one-to-one function that maps $\mathcal{O} \subset \mathbb{R}^2$ to some open subset of $\mathcal{M}(\pi')$. For each fixed $\boldsymbol{\xi} \in \mathcal{O}$, $\hat{\boldsymbol{x}}(\boldsymbol{\xi}, \cdot)$ is smooth near π.

Please refer to the supplemental document for example local parameterizations of a few simple evolving surfaces.

Alternatively, when we have available a motion X for the evolving surface $\mathcal{M}(\pi)$, a **global parameterization**, which is effectively a local one that remains constant for all $(\boldsymbol{x}, \pi) \in \mathcal{T}$, can be induced via $\hat{\boldsymbol{x}}(\boldsymbol{\xi}, \pi) \coloneqq \mathsf{X}(\boldsymbol{p}(\boldsymbol{\xi}), \pi)$ where $\boldsymbol{p} : \mathcal{O} \mapsto \mathcal{B}$ is a smooth and one-to-one mapping that parameterizes the corresponding reference $\mathcal{B}$.

Velocities. Given any spatial point $\boldsymbol{x} \in \mathcal{M}(\pi)$ and a surface parameterization $\hat{\boldsymbol{x}}$ that is either local to $(\boldsymbol{x}, \pi)$ or global, there exists exactly one $\boldsymbol{\xi} \in \mathcal{O}$ satisfying $\hat{\boldsymbol{x}}(\boldsymbol{\xi}, \pi) = \boldsymbol{x}$. We call $\boldsymbol{\xi}$ the **local coordinates** of $\boldsymbol{x}$ and use it to define the **local velocity** of $\boldsymbol{x}$ as

$$\boldsymbol{v}(\boldsymbol{x}, \pi) = \left.\frac{\partial \hat{\boldsymbol{x}}(\boldsymbol{\xi}, \pi')}{\partial \pi'}\right|_{\pi' = \pi}, \tag{8}$$

as illustrated in Figure 3.

Assuming the evolving surface $\mathcal{M}(\pi)$ to be oriented by a spatial unit-normal field $\boldsymbol{n}(\boldsymbol{x}, \pi)$, it is well-known that the **scalar normal velocity** $V = \boldsymbol{v} \cdot \boldsymbol{n}$ is *parameterization-independent*, namely, it does not depend on the choice of surface parameterization $\hat{\boldsymbol{x}}$ [Grinfeld 2013]. On the contrary, the **local tangential velocity** $\boldsymbol{v}_{\tan} = \boldsymbol{v} - V\boldsymbol{n}$ is *parameterization-dependent*.

We now consider an evolving curve $\partial\mathcal{A}(\pi) \subset \mathcal{M}(\pi)$ with a unit-normal field $\boldsymbol{n}_{\partial\mathcal{A}}(\boldsymbol{x}, \pi)$ defined to be tangent to $\mathcal{M}(\pi)$ and normal to $\partial\mathcal{A}(\pi)$. We use "$\boldsymbol{n}_{\partial\mathcal{A}}$" to indicate the unit-normal field over a curve $\partial\mathcal{A}(\pi)$ and "$\boldsymbol{n}$" to denote that over some surface.

The **local velocity** $\boldsymbol{v}_{\partial\mathcal{A}}$ of $\partial\mathcal{A}(\pi)$ can be defined similarly to Eq. (8), and $V_{\partial\mathcal{A}} = \boldsymbol{v}_{\partial\mathcal{A}} \cdot \boldsymbol{n}_{\partial\mathcal{A}}$ indicates the **scalar normal velocity** of $\partial\mathcal{A}(\pi)$, which is parameterization-independent.

[4] Strictly, a local parameterization should be expressed as $\hat{\boldsymbol{x}}(\boldsymbol{\xi}, \pi'; \boldsymbol{x}, \pi)$. We omit the dependency on $\boldsymbol{x}$ and π for easier readability.

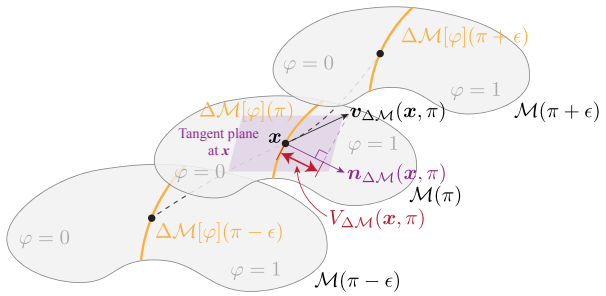


Fig. 4. **Evolution of discontinuity curves:** This example illustrates an evolving surface $\mathcal{M}(\pi)$ on which a binary-value scalar spatial field $\varphi(\boldsymbol{x},\pi)$ is defined. For some fixed $\pi\in\mathbb{R}$, assume $\varphi(\boldsymbol{x},\pi)$ to have discontinuity curve $\Delta\mathcal{M}[\varphi](\pi)\subset\mathcal{M}(\pi)$ with respect to $\boldsymbol{x}$ (the orange curves). Assume $\boldsymbol{x}$ to have local velocity $\boldsymbol{v}_{\Delta\mathcal{M}}(\boldsymbol{x},\pi)$ resulting from some surface parameterization that ensures $\boldsymbol{x}$ to stay on $\Delta\mathcal{M}[\varphi]$ near π. Then, the parameterization-independent scalar normal velocity $V_{\Delta\mathcal{M}}(\boldsymbol{x},\pi)$, which is marked red, equals $\boldsymbol{v}_{\Delta\mathcal{M}}(\boldsymbol{x},\pi)\cdot\boldsymbol{n}_{\Delta\mathcal{M}}(\boldsymbol{x},\pi)$ with $\boldsymbol{n}_{\Delta\mathcal{M}}(\boldsymbol{x},\pi)$ being the curve normal (the purple arrow) that resides within the tangent plane at $\boldsymbol{x}$ (in light purple).

Scene derivatives. For a scalar spatial field $\varphi(\boldsymbol{x},\pi)$ on $\mathcal{M}(\pi)$, its **scene derivative** (which is parameterization-dependent) and **normal scene derivative** (which is parameterization-independent) are

$$\dot{\varphi}(\boldsymbol{x},\pi)=\frac{\partial}{\partial\pi'}\varphi(\hat{\boldsymbol{x}}(\boldsymbol{\xi},\pi'),\pi')\bigg|_{\pi'=\pi}, \tag{9}$$

$$\overset{\square}{\varphi}=\dot{\varphi}-\boldsymbol{v}_{\text{tan}}\cdot\text{grad}_{\mathcal{M}}(\varphi), \tag{10}$$

respectively, where $\boldsymbol{\xi}$ is the local coordinates of $\boldsymbol{x}$, and $\text{grad}_{\mathcal{M}}(\varphi)$ denotes the surface gradient of φ.

When the surface parameterization $\hat{\boldsymbol{x}}$ produces zero tangential velocity at some $\boldsymbol{x}\in\mathcal{M}(\pi)$, i.e., $\boldsymbol{v}_{\text{tan}}(\boldsymbol{x},\pi)=\boldsymbol{0}$, the normal scene derivative $(\varphi)^{\square}$ reduces to the scene derivative $\dot{\varphi}$.

3.3 Differentiating Surface Integrals

The path integral formulation of Eq. (1) effectively reduces physics-based rendering into a problem of evaluating surface integrals. Consequently, central to deriving differentiable rendering is the problem of differentiating these integrals. Based on the concepts described in §3.2, we will utilize a so-called *transport relation*[5] that originated in fluid mechanics [Cermelli et al. 2005].

We consider an evolving surface $\mathcal{M}(\pi)$ oriented by a unit-normal field $\boldsymbol{n}(\boldsymbol{x},\pi)$. Let $\varphi(\boldsymbol{x},\pi)$ be a scalar spatial field defined on $\mathcal{M}(\pi)$. Assume that, for each fixed π, $\varphi(\boldsymbol{x},\pi)$ is C^0-continuous[6] with respect to $\boldsymbol{x}$ except along a zero-measure set of **discontinuity curves** $\Delta\mathcal{M}[\varphi](\pi)\subset\mathcal{M}(\pi)$ that consists of jump discontinuity points of $\varphi(\boldsymbol{x},\pi)$ and evolve continuously (see Figure 4). We define the **extended boundary** of $\mathcal{M}(\pi)$ with respect to φ, which we denote as $\overline{\partial\mathcal{M}}[\varphi](\pi)$, as the union of the boundary $\partial\mathcal{M}(\pi)$ and the discontinuity curves $\Delta\mathcal{M}[\varphi](\pi)$. When the scalar spatial field $\varphi(\boldsymbol{x},\pi)$ is clear from the context, we omit "$[\varphi]$" and write $\Delta\mathcal{M}(\pi)$ and $\overline{\partial\mathcal{M}}(\pi)$ for notational convenience.

[5]Here the term "transport" refers to *transport phenomena* that are studied by many sub-fields of physics such as continuum mechanics and thermodynamics.
[6]In the rest of this paper, we omit C^0 and use "continuous" to indicate C^0 continuity.

Cermelli et al. [2005] have shown that the derivative of the integral of $\varphi(\boldsymbol{x},\pi)$ over an evolving surface $\mathcal{M}(\pi)$ involves an *interior* and a *boundary* term:

$$\frac{\mathrm{d}}{\mathrm{d}\pi}\int_{\mathcal{M}}\varphi\,\mathrm{d}A=\underbrace{\int_{\mathcal{M}}\left(\overset{\square}{\varphi}-\varphi\,\kappa\,V\right)\mathrm{d}A}_{\text{interior}}+\underbrace{\int_{\overline{\partial\mathcal{M}}}\Delta\varphi\,V_{\overline{\partial\mathcal{M}}}\,\mathrm{d}\ell}_{\text{boundary}}, \tag{11}$$

where $\mathrm{d}A$ and $\mathrm{d}\ell$ are the surface-area and curve-length measures, respectively; κ is the **total curvature** (that is, the sum of the principal curvatures); V and $V_{\overline{\partial\mathcal{M}}}$ are the scalar normal velocity of $\mathcal{M}(\pi)$ and that of its extended boundary $\overline{\partial\mathcal{M}}(\pi)$, respectively. Additionally,

$$\Delta\varphi(\boldsymbol{x},\pi):=\begin{cases}\varphi(\boldsymbol{x},\pi), & \text{for } \boldsymbol{x}\in\partial\mathcal{M}(\pi)\\ \varphi^{-}(\boldsymbol{x},\pi)-\varphi^{+}(\boldsymbol{x},\pi), & \text{for } \boldsymbol{x}\in\Delta\mathcal{M}(\pi)\end{cases} \tag{12}$$

where $\varphi^{-}(\boldsymbol{x},\pi)$ and $\varphi^{+}(\boldsymbol{x},\pi)$, respectively, denote the one-sided limits of $\varphi(\boldsymbol{x},\pi)$ when approaching $\boldsymbol{x}$ from $-\boldsymbol{n}_{\Delta\mathcal{M}}(\boldsymbol{x},\pi)$ and $\boldsymbol{n}_{\Delta\mathcal{M}}(\boldsymbol{x},\pi)$.

In a special case where the surface $\mathcal{M}$ is independent of π and, thus, exhibits no motion, it holds that $V=0$ and $(\varphi)^{\square}=\dot{\varphi}$. We can then simplify Eq. (11) to the standard Reynolds [1903] transport relation used by Zhang et al. [2019]:

$$\tfrac{\mathrm{d}}{\mathrm{d}\pi}\textstyle\int_{\mathcal{M}}\varphi\,\mathrm{d}A=\int_{\mathcal{M}}\dot{\varphi}\,\mathrm{d}A+\int_{\Delta\mathcal{M}}\Delta\varphi\,V_{\Delta\mathcal{M}}\,\mathrm{d}\ell. \tag{13}$$

Notice that, although $V=0$, $V_{\Delta\mathcal{M}}$ may be nonzero as the discontinuity curves $\Delta\mathcal{M}(\pi)$ can still depend on the parameter π.

In physics-based rendering, a virtual scene is generally parameterized with a set of (mutually independent) **scene parameters** $\boldsymbol{\pi}=\{\pi_1,\pi_2,\ldots\}$, where each π_j is associated with a motion x_j of the scene geometry. In the rest of this paper, we tackle the problem of calculating partial derivatives of radiometric measurements I given by path integrals with respect to individual $\pi\in\boldsymbol{\pi}$.

4 DIFFERENTIAL DIRECT ILLUMINATION

As a warm-up before developing our general theory in §5, we first go over the case of direct illumination (i.e., one-bounce light transport). As we will see, results obtained from this section generalize nicely to the case of full path integrals. Therefore, we can use the direct illumination case to develop intuition about the general case.

We consider a simple scene configuration with one static object with surface $\mathcal{M}_{\text{obj}}$ lit by a light source defined on an evolving surface $\mathcal{L}(\pi)$. Then, given two points $\boldsymbol{y},\ \boldsymbol{y}'\in\mathcal{M}_{\text{obj}}$, the reflected radiance exiting $\boldsymbol{y}$ toward $\boldsymbol{y}'$ resulting from direct illumination equals:

$$I_{\text{direct}}=\int_{\mathcal{L}}\underbrace{L_{\text{e}}(\boldsymbol{x}\to\boldsymbol{y})\,f_{\text{s}}(\boldsymbol{x}\to\boldsymbol{y}\to\boldsymbol{y}')\,G(\boldsymbol{x}\leftrightarrow\boldsymbol{y})}_{=:f_{\text{direct}}(\boldsymbol{x})}\,\mathrm{d}A(\boldsymbol{x}), \tag{14}$$

where G is the geometric term. Our goal is to derive the derivative of I_{direct} with respect to π.

To simplify our derivation, we make two assumptions:

A.1 For all $\boldsymbol{x}\in\mathcal{L}(\pi)$, there exists a surface parameterization such that $\boldsymbol{x}$ has zero tangential velocity.

A.2 $L_{\text{e}}(\boldsymbol{x}\to\boldsymbol{y})\,f_{\text{s}}(\boldsymbol{x}\to\boldsymbol{y}\to\boldsymbol{y}')$ is continuous with respect to $\boldsymbol{x}$ in the interior of $\mathcal{L}(\pi)$ when $\boldsymbol{y},\ \boldsymbol{y}'\in\mathcal{M}_{\text{obj}}$ are fixed.

Based on these assumptions, applying the transport relation of Eq. (11) to the surface integral of Eq. (14) produces:

$$\frac{\partial I_{\text{direct}}}{\partial \pi} = \underbrace{\int_{\mathcal{L}} \left[\dot{f}_{\text{direct}} - f_{\text{direct}}\,\kappa\, V\right] \mathrm{d}A}_{\text{interior}} + \underbrace{\int_{\overline{\partial \mathcal{L}}} \Delta f_{\text{direct}}\, V_{\overline{\partial \mathcal{L}}}\, \mathrm{d}\ell}_{\text{boundary}}, \tag{15}$$

where:

- $\dot{f}_{\text{direct}}$ is the scene derivative of f_{direct} and equals the normal scene derivative $(f_{\text{direct}})^{\square}$ under Assumption A.1.
- κ is the total curvature.
- $\overline{\partial \mathcal{L}}(\pi)$ denotes the extended boundary of $\mathcal{L}(\pi)$ comprised of the boundary $\partial \mathcal{L}(\pi)$ and the discontinuity curves $\Delta \mathcal{L}[f_{\text{direct}}](\pi)$.
- V and $V_{\overline{\partial \mathcal{L}}}$ are the scalar normal velocity of $\mathcal{L}(\pi)$ and that of $\overline{\partial \mathcal{L}}(\pi)$, respectively.
- Lastly, $\Delta f_{\text{direct}}(\boldsymbol{x})$ follows the definition in Eq. (12) and indicates the difference in $f_{\text{direct}}(\boldsymbol{x})$ across the discontinuity curves $\Delta \mathcal{L}(\pi)$. Under Assumption A.2, this term equals

$$\Delta f_{\text{direct}}(\boldsymbol{x}) = L_{\text{e}}(\boldsymbol{x} \to \boldsymbol{y})\, f_{\text{s}}(\boldsymbol{x} \to \boldsymbol{y} \to \boldsymbol{y}')\, \Delta G(\boldsymbol{x} \leftrightarrow \boldsymbol{y}). \tag{16}$$

Implications of assumptions. Our derivation of Eq. (15) relies on two key assumptions (i.e., A.1 and A.2). In what follows, we discuss the implications of these assumptions.

Our first assumption (A.1) is that, for all $\boldsymbol{x} \in \mathcal{L}(\pi)$, there exists a surface parameterization $\hat{\boldsymbol{x}}$ that produces zero tangential velocity. In general, writing down such $\hat{\boldsymbol{x}}$ requires solving differential equations. On the other hand, we note that, to calculate the scene derivative $\dot{f}_{\text{direct}}$ at some fixed π, explicitly expressing the surface parameterization $\hat{\boldsymbol{x}}$ is unnecessary: As long as the local velocity $\boldsymbol{v}(\boldsymbol{x}, \pi)$ and the local change rate of the surface normal $\boldsymbol{n}(\boldsymbol{x}, \pi)$ can be numerically evaluated at π, so can $\dot{f}_{\text{direct}}$. When $\mathcal{M}(\pi)$ allows ray intersection to be computed analytically, which is usually the case in practice, these quantities can be computed numerically by differentiating the ray tracing process (see Eq. (19) of the work by Zhang et al. [2019] for more details).

Alternatively, when $\mathcal{L}$ is independent of the scene parameter π, Assumption A.1 is satisfied trivially. As we will show in §4.1, this can be achieved by leveraging material-form rendering integrals.

Our second assumption (A.2) of $(L_{\text{e}}\, f_{\text{s}})$ being continuous with respect to $\boldsymbol{x}$ typically implies the emitted radiance $L_{\text{e}}(\boldsymbol{x} \to \boldsymbol{y})$ and the BSDF $f_{\text{s}}(\boldsymbol{x} \to \boldsymbol{y} \to \boldsymbol{y}')$ to both be continuous (with $\boldsymbol{y}$ and $\boldsymbol{y}'$ fixed). This requires the absence of zero-measure (e.g., point and directional) sources and ideal specular surfaces such as perfect reflectors and smooth dielectric interfaces, which are also assumed by prior works [Li et al. 2018a; Loubet et al. 2019; Zhang et al. 2019].

We notice that L_{e} and f_{s} can be discontinuous with respect to $\boldsymbol{x}$ in the tangent plane of $\boldsymbol{y}$ (i.e., for $\boldsymbol{x}$ with $\boldsymbol{n}(\boldsymbol{y}) \cdot (\boldsymbol{y} \to \boldsymbol{x}) = 0$), violating Assumption A.2. Fortunately, when modulated with the cosine factor $|\boldsymbol{n}(\boldsymbol{y}) \cdot (\boldsymbol{y} \to \boldsymbol{x})|$ from the visibility-free geometry term G_0, these discontinuities usually vanish.[7]

[7] With shading normals deviating greatly from geometric ones, discontinuities within tangent planes may not vanish. We neglect this case for cleaner derivations, and our theory can be easily generalized by making the extended boundary curves $\overline{\partial \mathcal{L}}(\pi)$ to also contain spatial points within the tangent plane of $\boldsymbol{y}$ (that is, all $\boldsymbol{x} \in \mathcal{L}(\pi)$ with $\boldsymbol{n}(\boldsymbol{y}) \cdot (\boldsymbol{y} \to \boldsymbol{x}) = 0$).

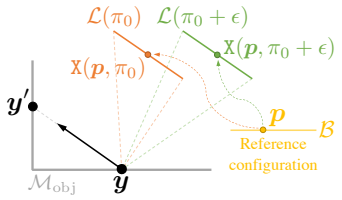


Fig. 5. Our **material** direct-illumination integral of Eq. (17) is over the π-independent reference configuration $\mathcal{B}$. If the light source $\mathcal{L}$ is not occluded when viewed from $\boldsymbol{x}$, the *boundary* integral vanishes as $\Delta \mathcal{B}[\hat{f}_{\text{direct}}](\pi) = \emptyset$.

Sources of discontinuities. In Eq. (15), $\overline{\partial \mathcal{L}}(\pi)$ is determined by the boundary of $\mathcal{L}(\pi)$, as well as the discontinuity curves $\Delta \mathcal{L}[f_{\text{direct}}](\pi)$ comprised of jump discontinuity points of $f_{\text{direct}}(\boldsymbol{x})$. Under Assumption A.2, these discontinuities are entirely due to geometric term G. Thus, $\Delta \mathcal{L}[f_{\text{direct}}](\pi) = \Delta \mathcal{L}[G(\cdot \leftrightarrow \boldsymbol{y})](\pi)$.

Specifically, a *discontinuous surface normal* can lead to sudden changes of the term $|\boldsymbol{n}(\boldsymbol{y}) \cdot (\boldsymbol{y} \to \boldsymbol{x})|$ with fixed $\boldsymbol{x} \in \mathcal{M}_{\text{obj}}$. Additionally, *visibility boundaries* correspond to discontinuities of the mutual visibility function $\mathbb{V}$. Geometrically, the discontinuities due to surface normals are categorized by Zhang et al. [2019] as *sharp edges*, whereas those due to visibility correspond to *silhouette edges*.

4.1 Material-Form Integrals

Prior works [Loubet et al. 2019; Zhang et al. 2019] have shown that the *boundary* term in Eq. (15) can be very costly to estimate in complex scenes. In what follows, we introduce a reformulation of the surface integral (14) such that, after differentiation, we can ignore the boundary $\partial \mathcal{L}(\pi)$ of the evolving surface $\mathcal{L}(\pi)$.

We assume $\mathcal{L}(\pi)$ to have a global parameterization induced from some motion X. Then, as shown in Figure 5, we can reformulate Eq. (14) to be over the corresponding reference configuration $\mathcal{B}$, yielding the following reparameterized integral:

$$I_{\text{direct}} = \int_{\mathcal{B}} \hat{f}_{\text{direct}}(\boldsymbol{p})\, \mathrm{d}A(\boldsymbol{p}), \tag{17}$$

where $\hat{f}_{\text{direct}}(\boldsymbol{p}) \coloneqq f_{\text{direct}}(\boldsymbol{x})\, J(\boldsymbol{p})$ with $\boldsymbol{x} = \mathsf{X}(\boldsymbol{p}, \pi)$ and

$$J(\boldsymbol{p}) = \left|\mathrm{d}A(\boldsymbol{x}) / \mathrm{d}A(\boldsymbol{p})\right|, \tag{18}$$

being the Jacobian determinant for the change of variables from spatial point $\boldsymbol{x}$ to its material representation $\boldsymbol{p}$. Following the terminology described in §3.2, we call Eqs. (14) and (17) spatial-form and material-form direct-illumination integrals, respectively.

Despite the similarity between the spatial-form (14) and the material-form (17) integrals, the latter enjoys a key advantage that it can be differentiated using the Reynolds transport relation (13) because its domain of integration $\mathcal{B}$ is independent of π, yielding:

$$\frac{\partial I_{\text{direct}}}{\partial \pi} = \underbrace{\int_{\mathcal{B}} (\hat{f}_{\text{direct}})^{\cdot}\, \mathrm{d}A}_{\text{interior}} + \underbrace{\int_{\Delta \mathcal{B}} \Delta \hat{f}_{\text{direct}}\, V_{\Delta \mathcal{B}}\, \mathrm{d}\ell}_{\text{boundary}}, \tag{19}$$

where $\Delta \mathcal{B}[\hat{f}_{\text{direct}}](\pi)$ contains the discontinuity curves of $\hat{f}_{\text{direct}}(\boldsymbol{p})$.

Practical advantages. Estimating $\partial I_{\text{direct}}/\partial\pi$ using the material-form integral of Eq. (19) offers a number of advantages:

- Compared to the spatial form (15), the material form allows the boundary curves $\partial\mathcal{B}$ to be excluded from the *boundary* component, reducing the computational cost for estimating this term.
- Compared to the solid-angle-integral formulation used by prior works [Li et al. 2018a; Zhang et al. 2019], the material form generalizes more easily to the full path integrals (§5.2), enabling the design of more sophisticated Monte Carlo estimators (§6).

5 DIFFERENTIAL PATH INTEGRALS

We now generalize the analysis in §4 to establish the *differential path integral* framework. Our objective is to differentiate radiometric measurements I depicted as path integrals (1) with the path space Ω defined over evolving surfaces $\mathcal{M}(\pi)$. That is, we aim to calculate $\partial I/\partial\pi$ where $\Omega(\pi) = \bigcup_{N=1}^{\infty} \mathcal{M}(\pi)^{N+1}$.

Preview. The transport relations of Eqs. (11) and (13) presented in §3.3 have shown that the derivative of a surface integral consists of an *interior* and a *boundary* components. We will show in §5.1 that the derivative $\partial I/\partial\pi$ can be expressed in a similar fashion as the sum of (i) an *interior* path integral over the original path space, and (ii) a *boundary* integral over the *boundary path space* comprised of *boundary light paths* (see Figure 6-b). This result, which we will refer to as the *spatial-form differential path integral*, will be shown in Eq. (29).

Additionally, by applying the reparameterization introduced in §4.1 to the *spatial-form differential path integral*, we will derive its *material-form* counterpart (36) in §5.2. Similar to the direct illumination case, this reparameterization simplifies both the *interior* integral (by having non-evolving surfaces) and the *boundary* one (by minimizing the type of discontinuities).

5.1 Spatial-form Differential Path Integral

We start with differentiating, with respect to π, measurements I_N defined as the spatial-form order-N path integral of Eq. (3) for some fixed $N \geq 1$. The derivatives of the spatial-form full path integral I will then follow from the relationship $I = \sum_{N=1}^{\infty} I_N$.

Recursive expression of path integrals. To derive $\partial I_N/\partial\pi$, we first rewrite Eq. (3) recursively. To this end, we define

$$h_N(\boldsymbol{x}_N;\, \boldsymbol{x}_{N-1}) \coloneqq W_e(\boldsymbol{x}_N \to \boldsymbol{x}_{N-1}), \tag{20}$$

and, for $0 \leq n < N$,

$$h_n(\boldsymbol{x}_n;\, \boldsymbol{x}_{n-1}) \coloneqq \int_{\mathcal{M}} g(\boldsymbol{x}_{n+1};\, \boldsymbol{x}_{n-1}, \boldsymbol{x}_n)\, h_{n+1}(\boldsymbol{x}_{n+1};\, \boldsymbol{x}_n)\, \mathrm{d}A(\boldsymbol{x}_{n+1}), \tag{21}$$

where g is defined in Eq. (5). Then, it is easy to verify that

$$h_0(\boldsymbol{x}_0) = \int_{\mathcal{M}^N} f(\bar{\boldsymbol{x}}) \prod_{n=1}^{N} \mathrm{d}A(\boldsymbol{x}_n), \tag{22}$$

$$I_N = \int_{\mathcal{M}} h_0(\boldsymbol{x}_0)\, \mathrm{d}A(\boldsymbol{x}_0). \tag{23}$$

Given Eq. (23), calculating $\partial I_N/\partial\pi$ reduces to differentiating h_n. Similar to our handling of direct illumination in §4, we assume the absence of zero-measure sources or ideal specular surfaces. This ensures that, for all $0 \leq n \leq N$, h_n is continuous at all interior points $\boldsymbol{x}_n \in \mathcal{M}(\pi) \setminus \partial\mathcal{M}(\pi)$. Then, using a surface parameterization that

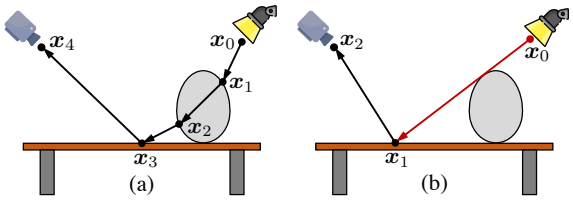


Fig. 6. **Boundary light paths:** Unlike their regular counterparts (a), each boundary light path (b) has one of its vertices constrained on a curve. In this example, the path $(\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{x}_2) \in \partial\Omega_{2,1}$ has its vertex $\boldsymbol{x}_1 \in \overline{\partial\mathcal{M}}_1(\pi)$ on the silhouette with respect to $\boldsymbol{x}_0$.

produces zero tangential velocity for all $\boldsymbol{x} \in \mathcal{M}(\pi)$, we can apply the transport relation of Eq. (11) to express the scene derivative of h_n as:

$$\dot{h}_n = \int_{\mathcal{M}} \left[(h_{n+1}\, g)^{\cdot} - h_{n+1}\, g\, \kappa\, V\right] \mathrm{d}A + \int_{\overline{\partial\mathcal{M}}_{n+1}} h_{n+1}\, \Delta g\, V_{\overline{\partial\mathcal{M}}_{n+1}}\, \mathrm{d}\ell, \tag{24}$$

where:

- $(h_{n+1}\, g)^{\cdot}$ is the scene derivative of $(h_{n+1}\, g)$ and equals the normal scene derivative $(h_{n+1}\, g)^{\square}$ under Assumption **A.1**.
- $\overline{\partial\mathcal{M}}_{n+1}(\pi) \coloneqq \partial\mathcal{M}(\pi) \cup \Delta\mathcal{M}[g(\cdot;\, \boldsymbol{x}_{n-1}, \boldsymbol{x}_n)](\pi)$ denotes the extended boundary of $\mathcal{M}(\pi)$ comprised of the boundary $\partial\mathcal{M}(\pi)$ and the discontinuity curves $\Delta\mathcal{M}(\pi)$ with respect to the function g (when $\boldsymbol{x}_{n-1}$ and $\boldsymbol{x}_n$ are fixed).
- V and $V_{\overline{\partial\mathcal{M}}_{n+1}}$ are the scalar normal velocity of $\mathcal{M}(\pi)$ and that of $\overline{\partial\mathcal{M}}_{n+1}(\pi)$, respectively.
- Δg follows Eq. (12) and indicates the difference in g across the discontinuity curves. We note that Δh_{n+1} is not needed here, as h_{n+1} is assumed to be continuous in the interior of $\mathcal{M}(\pi)$.

Differentiating order-N path integrals. With Eqs. (21) and (24) at hand, we can now differentiate I_N. To this end, we use the expression of I_N in Eq. (23), and repeatedly expand h_n and $\dot{h}_n$ for $n = 0, 1, \ldots, N-1$, resulting in:

$$\begin{aligned}\partial I_N/\partial\pi = &\int_{\Omega_N} \left[\dot{f}(\bar{\boldsymbol{x}}) - f(\bar{\boldsymbol{x}}) \textstyle\sum_{K=0}^{N} \kappa(\boldsymbol{x}_K)\, V(\boldsymbol{x}_K)\right] \mathrm{d}\mu(\bar{\boldsymbol{x}}) + \\ &\textstyle\sum_{K=0}^{N} \left[\int_{\partial\Omega_{N,K}} \Delta f_K(\bar{\boldsymbol{x}})\, V_{\overline{\partial\mathcal{M}}_K}(\boldsymbol{x}_K)\, \mathrm{d}\mu'_{N,K}(\bar{\boldsymbol{x}})\right],\end{aligned} \tag{25}$$

where

$$\partial\Omega_{N,K} \coloneqq \mathcal{M}(\pi)^K \times \overline{\partial\mathcal{M}}_K(\pi) \times \mathcal{M}(\pi)^{N-K}, \tag{26}$$

$$\mathrm{d}\mu'_{N,K}(\bar{\boldsymbol{x}}) \coloneqq \mathrm{d}\ell(\boldsymbol{x}_K) \prod_{\substack{0 \leq n \leq N \\ n \neq K}} \mathrm{d}A(\boldsymbol{x}_n), \tag{27}$$

$$\Delta f_K(\bar{\boldsymbol{x}}) = f(\bar{\boldsymbol{x}})\, \Delta g(\boldsymbol{x}_K;\, \boldsymbol{x}_{K-2}, \boldsymbol{x}_{K-1}) / g(\boldsymbol{x}_K;\, \boldsymbol{x}_{K-2}, \boldsymbol{x}_{K-1}). \tag{28}$$

As a base case, as h_0 is assumed continuous, i.e., $\Delta\mathcal{M}[h_0](\pi) = \emptyset$, it holds that $\overline{\partial\mathcal{M}}_0(\pi) = \partial\mathcal{M}(\pi)$ and $\Delta f_0(\bar{\boldsymbol{x}}) = f(\bar{\boldsymbol{x}})$. Please see Appendix A for a full derivation of this result.

Completing the derivation. Finally, as $I = \sum_{N=1}^{\infty} I_N$, it holds that $\partial I/\partial\pi = \sum_{N=1}^{\infty} \partial I_N/\partial\pi$. Thus, we can sum up Eq. (25) for all $N \geq 1$ into a single expression of $\partial I/\partial\pi$ as follows.

Spatial-form differential path integral

For a radiometric measurement I given by a spatial-form path integral, its derivative with respect to scene parameter π can be expressed as a **spatial-form differential path integral**:

$$\frac{\partial I}{\partial \pi} = \underbrace{\int_\Omega \left[\dot{f}(\bar{\boldsymbol{x}}) - f(\bar{\boldsymbol{x}}) \sum_{K=0}^{N} \kappa(\boldsymbol{x}_K)\, V(\boldsymbol{x}_K) \right] \mathrm{d}\mu(\bar{\boldsymbol{x}})}_{\text{interior}} + \underbrace{\int_{\partial\Omega} \Delta f_K(\bar{\boldsymbol{x}})\, V_{\overline{\partial \mathcal{M}}_K}(\boldsymbol{x}_K)\, \mathrm{d}\mu'(\bar{\boldsymbol{x}})}_{\text{boundary}}, \tag{29}$$

where $\partial\Omega \subset \Omega$ is the **boundary path space** defined as:

$$\partial\Omega := \bigcup_{N=1}^{\infty} \bigcup_{K=0}^{N} \partial\Omega_{N,K}, \tag{30}$$

and μ' is the **differential area-product measure** given by:

$$\mu'(D) := \bigcup_{N=1}^{\infty} \bigcup_{K=0}^{N} \mu'_{N,K}\left(D \cap \partial\Omega_{N,K}\right), \tag{31}$$

for any $D \subset \partial\Omega$. We call light paths of this kind as **boundary light paths**, and the function Δf_K as the **boundary contribution function**. We also distinguish the segment $\boldsymbol{x}_{K-1}\,\boldsymbol{x}_K$ as the **boundary segment** of boundary path $\bar{\boldsymbol{x}}$.

Analogous to the transport relation of Eq. (11), our spatial-form differential path integral formulation involves a *interior* and a *boundary* term: The *interior* term is a path integral over the same path space Ω and with the same area-product measure μ as the original path integral. By contrast, the *boundary* term is a path integral over the boundary path space $\partial\Omega$ and uses the differential area-product measure μ'. This is the space of light paths $\bar{\boldsymbol{x}} = (\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$ with $N \geq 1$ and one of the vertices $\boldsymbol{x}_K$ being constrained to be on the curves given by $\overline{\partial\mathcal{M}}_K(\pi)$ (see Figure 6).

Sources of discontinuities. In Eqs. (24) and (25), for $0 \leq n < N$, $\overline{\partial\mathcal{M}}_{n+1}(\pi)$ is determined by any boundaries of $\mathcal{M}(\pi)$, as well as the discontinuity curves of $g(\boldsymbol{x}_{n+1};\, \boldsymbol{x}_{n-1}, \boldsymbol{x})$ with respect to $\boldsymbol{x}_{n+1}$, when holding $\boldsymbol{x}_{n-1}$ and $\boldsymbol{x}_n$ fixed. Similar to the direct-illumination case, these discontinuities generally arise from *sharp* and *silhouette* edges.

Reciprocity. It is easy to verify that the *interior* term of our differential path integral (29) is reciprocal.[8] The *boundary* term, on the other hand, is not. This is primarily due to our "unidirectional" definition of $\overline{\partial\mathcal{M}}_n(\pi)$: for all $k > 0$, it contains the discontinuity points of g with respect to $\boldsymbol{x}_n$, with $\boldsymbol{x}_{n-2}$ and $\boldsymbol{x}_{n-1}$ fixed. If we defined $\overline{\partial\mathcal{M}}_n(\pi)$ by considering discontinuities with $\boldsymbol{x}_{n+1}$ and $\boldsymbol{x}_{n+2}$ fixed, we would obtain another *boundary* term that takes a different form but is mathematically equivalent.

As we will discuss in §6, the *boundary* term not providing reciprocity has little impact on the design of efficient Monte Carlo estimators. Thus, we consider the derivation of reciprocal *boundary* terms as future work.

[8] The presence of refractive interfaces and shading normals can break the reciprocity of BSDFs and, thus, that of measurement contribution f and its scene derivative $\dot{f}$. Fortunately, the handling of such asymmetry by introducing additional correction terms [Veach 1997] generalizes naturally to differentiable rendering.

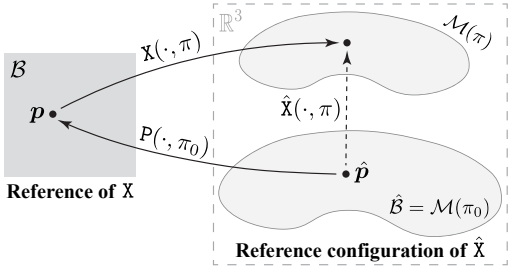


Fig. 7. **Alternative parameterization:** Given a motion X and a fixed $\pi_0 \in \mathbb{R}$, let P be the reference map of X. Then, composing $\mathsf{P}(\cdot, \pi_0)$ and $\mathsf{X}(\cdot, \pi)$ yields a new motion $\hat{\boldsymbol{x}}(\hat{\boldsymbol{p}}, \pi) = \mathsf{X}(\mathsf{P}(\hat{\boldsymbol{p}}, \pi_0), \pi)$ that reduces to the identity map at $\pi = \pi_0$.

5.2 Material-Form Path Integrals

We now generalize our material-form direct-illumination integrals in Eqs. (17) and (19) to material-form full path integrals. Similar to the direct illumination case, our material-form differential path integral minimizes the contribution of the *boundary* term and improves the effectiveness of reusing path samples when jointly estimating the original radiometric measurements I (e.g., the original image) and the corresponding derivatives $\partial I/\partial\pi$ (e.g., the derivative images). Additionally, minimizing the contribution of the *boundary* term reduces the computational overhead required for estimating this term, which previously required tracing expensive "side paths" to obtain unbiased estimations [Li et al. 2018a; Zhang et al. 2019].

Assume that evolving surfaces $\mathcal{M}(\pi)$ can be parameterized globally using some motion X with a reference configuration $\mathcal{B}$. By substituting each $\boldsymbol{x}_n \in \mathcal{M}(\pi)$ using $\mathsf{X}(\boldsymbol{p}_n, \pi)$ with material point $\boldsymbol{p}_n$, the original spatial-form path integral of Eq. (1) can be expressed as another **material-form path integral**:

$$I = \int_{\hat{\Omega}} \hat{f}(\bar{\boldsymbol{p}})\, \mathrm{d}\mu(\bar{\boldsymbol{p}}), \tag{32}$$

where $\hat{\Omega} := \bigcup_{N=1}^{\infty} \mathcal{B}^{N+1}$ is the **material path space** independent of π. In Eq. (32), for a **material light path** $\bar{\boldsymbol{p}} = (\boldsymbol{p}_0, \ldots, \boldsymbol{p}_N) \in \hat{\Omega}$, its **material measurement contribution function** $\hat{f}$, which can be obtained by modifying Eqs. (4–7), equals:

$$\hat{f}(\bar{\boldsymbol{p}}) = \left(\prod_{n=0}^{N-1} \hat{g}(\boldsymbol{p}_{n+1};\, \boldsymbol{p}_{n-1}, \boldsymbol{p}_n) \right) \hat{W}_\mathrm{e}(\boldsymbol{p}_N \to \boldsymbol{p}_{N-1}), \tag{33}$$

where

$$\hat{W}_\mathrm{e}(\boldsymbol{p}_N \to \boldsymbol{p}_{N-1}) := J(\boldsymbol{p}_N)\, W_\mathrm{e}(\boldsymbol{x}_N \to \boldsymbol{x}_{N-1}), \tag{34}$$

and, for $0 \leq n < N$,

$$\hat{g}(\boldsymbol{p}_{n+1};\, \boldsymbol{p}_{n-1}, \boldsymbol{p}_n) := \underbrace{\hat{f}_\mathrm{s}(\boldsymbol{p}_{n-1} \to \boldsymbol{p}_n \to \boldsymbol{p}_{n+1})}_{:= J(\boldsymbol{p}_n)\, f_\mathrm{s}(\boldsymbol{x}_{n-1} \to \boldsymbol{x}_n \to \boldsymbol{x}_{n+1})} G(\boldsymbol{x}_n \leftrightarrow \boldsymbol{x}_{n+1}). \tag{35}$$

In Eqs. (34) and (35), J is the Jacobian determinant given by Eq. (18), and $\boldsymbol{x}_n = \mathsf{X}(\boldsymbol{p}_n, \pi)$ is, for all n, the spatial representation of the material point $\boldsymbol{p}_n$.

Material-form differential path integral

Assuming the Jacobian determinant J to be continuous,[a] differentiating the material-form path integral of Eq. (32) produces the following **material-form differential path integral**:

$$\frac{\partial I}{\partial \pi} = \underbrace{\int_{\hat{\Omega}} (\hat{f})^{\cdot}(\bar{\boldsymbol{p}})\,\mathrm{d}\mu(\bar{\boldsymbol{p}})}_{\text{interior}} + \underbrace{\int_{\partial\hat{\Omega}} \Delta\hat{f}_K(\bar{\boldsymbol{p}})\, V_{\Delta\mathcal{B}_K}(\boldsymbol{p}_K)\,\mathrm{d}\mu'(\bar{\boldsymbol{p}})}_{\text{boundary}}, \tag{36}$$

where:

- The scene derivative $(\hat{f})^{\cdot}$ of the material measurement contribution $\hat{f}$ of Eq. (33) is obtained using the global paramterization of $\mathcal{M}(\pi)$ induced by the motion X.
- The **material boundary path** $\bar{\boldsymbol{p}}$, its **material boundary contribution** $\Delta\hat{f}_K(\bar{\boldsymbol{p}})$, and the **material boundary path space** $\partial\hat{\Omega}$ are, respectively, defined in a similar fashion as $\bar{\boldsymbol{x}}$, $\Delta f_K(\bar{\boldsymbol{x}})$, and $\partial\Omega$ in Eqs. (28) and (30).
- Lastly, $\Delta\mathcal{B}_K(\pi) \coloneqq \Delta\mathcal{B}[\hat{g}(\cdot; \boldsymbol{p}_{K-2}, \boldsymbol{p}_{K-1})](\pi)$ comprises the discontinuity curves of $\hat{g}$ with respect to $\boldsymbol{p}_K$, when $\boldsymbol{p}_{K-2}$ and $\boldsymbol{p}_{K-1}$ are fixed. We call $\boldsymbol{p}_{K-1}\,\boldsymbol{p}_K$ a **material boundary segment**.

[a] Eq. (36) also holds when the discontinuity curves $\Delta\mathcal{B}[J]$ are independent of π.

Comparison of spatial-form and material-form. Similar to the direct illumination case, estimating the partial derivative $\partial I/\partial\pi$ of radiometric measurements I using our material-form differential path integral offers the advantage of not having to include $\partial\mathcal{B}$, which is independent of π, in the *boundary* term. Furthermore, by leveraging proper parameterizations, the material form allows its *boundary* term to involve even fewer types of discontinuities, as we discuss below.

On the other hand, our material-form integrals of Eqs. (19) require as input a pre-determined global parameterization induced from some motion X. For certain applications, such as when the evolving scene geometry $\mathcal{M}(\pi)$ is expressed *implicitly*, such global parameterizations may be difficult to obtain. In these cases, it may be necessary to resort to the spatial form of Eqs. (15) and (29), which only impose the following requirements on $\mathcal{M}(\pi)$: (i) it must allow ray tracing (i.e., ray-surface intersection computation) to be performed in a *differentiable* fashion; and (ii) it must allow the sampling of points on the surface (which can be done using particle-based methods [Witkin and Heckbert 1994] for implicit surfaces). The second requirement is for the Monte Carlo estimation algorithms we introduce in §6.

Alternative parameterizations. In practice, when calculating $\partial I/\partial\pi$ at some fixed π_0, we can define another motion $\hat{\mathsf{X}}$ that is *local* to π_0 and has the reference $\hat{\mathcal{B}} = \mathcal{M}(\pi_0)$. Let P be the reference map of the motion X. For any fixed $\pi' \in \mathbb{R}$, let

$$\hat{\mathsf{X}}(\cdot, \pi) = \mathsf{X}(\cdot, \pi) \circ \mathsf{P}(\cdot, \pi_0). \tag{37}$$

Namely, $\hat{\mathsf{X}}(\hat{\boldsymbol{p}}, \pi) = \mathsf{X}(\mathsf{P}(\hat{\boldsymbol{p}}, \pi_0), \pi)$ for all $\hat{\boldsymbol{p}} \in \hat{\mathcal{B}}$ (see Figure 7). Then, $\hat{\mathsf{X}}(\cdot, \pi)$ reduces to the identity map when $\pi = \pi_0$. Thus, when using the locally defined motion $\hat{\mathsf{X}}$ for the material-form differential path integral of Eq. (36), the Jacobian determinants J from Eq. (18) become one, allowing the path integral to be efficiently estimated using previously developed path sampling methods. Notice that the scene derivatives $\dot{J}$ generally remain nonzero.

Another possibility is to have the reference $\hat{\mathcal{B}}$ set to $[0,1)^2$ and the motion $\hat{\mathsf{X}}(\hat{\boldsymbol{p}}, \pi)$ determined by the sampling process of points $\boldsymbol{x}$ on the surface $\mathcal{M}(\pi)$, with $\hat{\boldsymbol{p}}$ being the random numbers. Then, the resulting material path space $\hat{\Omega}$ essentially becomes the *primary sample space* used by many Markov-Chain Monte Carlo (MCMC) rendering algorithms (e.g., [Kelemen et al. 2002]). We opt to use the parameterization of Eq. (37) in this paper due to the advantage of having unit-valued Jacobian determinants J.

Sources of discontinuities. As discussed in §5.1, for the spatial-form differential path integral (29), discontinuities of $g(\boldsymbol{x}_{n+1}; \boldsymbol{x}_{n-1}, \boldsymbol{x}_n)$ arise from those in surface normal and visibility. This remains the case for $\hat{g}(\boldsymbol{p}_{n+1}; \boldsymbol{p}_{n-1}, \boldsymbol{p}_n)$ in the material-form differential path integral (36).

Fortunately, since the integral domain $\mathcal{B}$ is independent of the scene parameter π, many of the jump discontinuity points of g with respect to $\boldsymbol{p}_{n+1}$ no longer moves with π. This causes the normal velocity $V_{\Delta\mathcal{B}_{n+1}}$ at these points to vanish, allowing them to be omitted from the *boundary* integral.

In practice, when $\mathcal{M}(\pi)$ and $\mathcal{B}$ are depicted using polygonal meshes, the motion X or its local variant $\hat{\mathsf{X}}$ given by Eq. (37) usually maps face edges of $\mathcal{B}$ to those of $\mathcal{M}(\pi)$. In this case, although surface normal can be discontinuous across face edges in $\mathcal{B}$, these edges do not have to be included in $\Delta\mathcal{B}[\hat{f}](\pi)$ as they are π-independent. Therefore, the only type of discontinuity needed to be handled with *boundary* integrals is visibility-related, i.e., the *silhouette edges*.

We note that the Jacobian determinant J given by Eq. (18) is usually constant within each face of a polygonal mesh but discontinuous across the face boundaries. Fortunately, because the face boundaries of the reference $\mathcal{B}$ are independent of π, our material-form differential path integral of Eq. (36) still holds, and the discontinuity curves $\Delta\mathcal{B}_n(\pi)$ do not need to include the face boundaries.

6 MONTE CARLO ESTIMATION OF DIFFERENTIAL PATH INTEGRALS

Our differential path integral formulations of Eq. (29) and (36) facilitate the design of efficient Monte Carlo methods for estimating derivatives of radiometric measurements with respect to arbitrary scene parameters π. We focus our derivations on the material form, but they can be easily generalized to handle the spatial form as well.

Terminology. In the rest of this section, to simplify terminology, we omit explicitly specifying that we use path integrals (32) and differential path integrals (36) in their **material forms**.

Preview. Since the *interior* term integrates over the original path space, it can be estimated by adapting existing path sampling techniques such as unidirectional and bidirectional path tracing, which we will discuss in §6.1.

The *boundary* path integral, on the contrary, operates over the boundary path space. We will introduce in §6.2–§6.4 a new Monte

Carlo estimator for this term. Our estimator works in a *multi-directional* fashion and constructs a boundary light path starting with its boundary segment. Then, two original light paths are sampled to connect one endpoint of the boundary segment to the light source and the other to the detector.

6.1 Estimating the Interior Integral

Thanks to the similarity between the original path integral (32) the *interior* component of our differential path integral (36), the latter can be estimated using previously developed path sampling methods. Specifically, we draw light paths $\bar{\boldsymbol{p}}$ from the path space $\hat{\Omega}$ by applying conventional path sampling methods to the reference configuration $\mathcal{B}$. This allows us to estimate radiometric measurement I and its derivative $\partial I/\partial\pi$ jointly by reusing the path samples.

In practice, we adopt two commonly used algorithms, unidirectional and bidirectional path tracing, to obtain unbiased estimates of the *interior* integral. Algorithm 1 outlines the unidirectional variant of our algorithm. For notational clarity, we omit multiple importance sampling (MIS) in this algorithm.

In this algorithm, we calculate the scene derivatives based the relation between spatial and material points: $\boldsymbol{x}_n = \mathsf{X}(\boldsymbol{p}_n, \pi)$ for $n = 0, 1, 2$. For instance, the scene derivatives of the BSDF equals

$$[f_\mathrm{s}(\boldsymbol{x}_0 \to \boldsymbol{x}_1 \to \boldsymbol{x}_2)]^\bullet = \frac{\partial}{\partial\pi} f_\mathrm{s}(\mathsf{X}(\boldsymbol{p}_0, \pi) \to \mathsf{X}(\boldsymbol{p}_1, \pi) \to \mathsf{X}(\boldsymbol{p}_2, \pi)), \tag{38}$$

with the material points $\boldsymbol{p}_0$, $\boldsymbol{p}_1$, and $\boldsymbol{p}_2$ fixed.

Proper use of automatic differentiation. In practice, all the scene derivatives from Lines 3, 8, and 15 of Algorithm 1 can be computed numerically using automatic differentiation (autodiff) techniques. However, precaution is needed when applying autodiff to existing path tracing implementations: When differentiating α_direct in Line 10, for instance, we should ensure that the differentiation involves the full representation of the source emission L_e, the surface BSDF f_s, and the geometric term G. Traditional path tracers usually have certain components, such as the cosine factors from G, omitted, as they are canceled out by the PDF term $\mathbb{P}_\mathrm{direct}$. Directly applying automatic differentiation to such implementations can produce incorrect derivative estimates. These risks can be alleviated by designing differentiable renderers in a way that allows completely separating the computation of the measurement contribution and PDF terms, and *detaching* the latter from the automatic differentiation process.

6.2 Multi-Directional Form of the Boundary Integral

An important distinction between our differential path integral formulations of Eqs. (29) and (36) and prior works [Li et al. 2018a; Loubet et al. 2019; Zhang et al. 2019] is the complete *decoupling* of the *boundary* integral from its *interior* counterpart. Leveraging this flexibility, we introduce a new unbiased Monte Carlo method to estimate the *boundary* integral in the rest of this section.

We consider a material boundary path $\bar{\boldsymbol{p}} = (\boldsymbol{p}_0, \boldsymbol{p}_1, \ldots, \boldsymbol{p}_N) \in \partial\hat{\Omega}$, with $\boldsymbol{x}_n := \mathsf{X}(\boldsymbol{p}_n, \pi)$ being the corresponding spatial points on the evolving surface $\mathcal{M}(\pi)$. As discussed in §5.2, we assume that one vertex of the boundary segment of $\bar{\boldsymbol{p}}$ is a jump discontinuity point of the visibility function when the other vertex is fixed. This creates

ALGORITHM 1: Estimating the *interior* integral of Eq. (36) using unidirectional path tracing

```
1  MaterialDifferentiablePathTracing(x_1, x_2)
   Input: Two spatial points x_1, x_2
   Output: L(x_1 → x_2) and its scene derivative L̇
2  begin
3      (L, L̇) ← (L_e(x_1 → x_2), [L_e(x_1 → x_2)]•);
4      (T, Ṫ) ← (1, 0);                                  // path throughput
5      while true do
           /* Direct illumination (light sampling)                    */
6          Draw p_0 ~ P_light(p_0);                       // area measure
7          x_0 ← X(p_0, π);
8          Compute α_direct and α̇_direct as, respectively, the value and
             scene derivative of:
             L_e(x_0 → x_1) f_s(x_0 → x_1 → x_2) G(x_0 ↔ x_1) J(p_0);
9          L ← L + T α_direct / P_light(p_0);
10         L̇ ← L̇ + (T α̇_direct + Ṫ α_direct) / P_light(p_0);
           /* Indirect illumination (BSDF sampling)                   */
11         Draw ω_i ~ P_bsdf(ω_i);                    // solid-angle measure
12         x_0 ← rayTrace(x_1, ω_i);
13         if x_0 is valid then                          // Ray tracing hits
14             p_0 ← P(x_0, π);
15             Compute α_indirect and α̇_indirect as, respectively, the
                 value and scene derivative of:
                 f_s(x_0 → x_1 → x_2) G(x_0 ↔ x_1) J(p_0);
               /* Convert probability to area measure                 */
16             q ← P_bsdf(ω_i) |n(x_0, π) · −ω_i| / ‖x_0 − x_1‖^2;
               /* Update throughputs                                  */
17             T ← T α_indirect / q; Ṫ ← (T α̇_indirect + Ṫ α_indirect) / q;
               /* Continue the path                                   */
18             x_2 ← x_1; x_1 ← x_0;
19         else                                          // Ray tracing misses
20             break;
21         end
22     end
23     return (L, L̇)
24 end
```

complications when building the boundary path: Constructing its boundary segment, for instance, could require sampling one of its vertices from the silhouette edges viewed from the other. Unfortunately, identifying silhouette edges can be costly [Loubet et al. 2019], making it difficult for prior unbiased methods [Li et al. 2018a; Zhang et al. 2019] to handle scenes with complex geometries.

To overcome this challenge without sacrificing unbiasedness, we propose building the boundary light paths $\bar{\boldsymbol{p}}$ in a *multi-directional* manner.[9] For notational convenience, we rename path vertices so that $\bar{\boldsymbol{p}} = (\boldsymbol{p}_s^\mathrm{S}, \ldots, \boldsymbol{p}_0^\mathrm{S}, \boldsymbol{p}_0^\mathrm{D}, \ldots, \boldsymbol{p}_t^\mathrm{D})$, and accordingly for the points $\boldsymbol{x}_n^\mathrm{S}$ and $\boldsymbol{x}_n^\mathrm{D}$ for all n. This lets us decompose the boundary light path $\bar{\boldsymbol{p}}$ into its boundary segment $\boldsymbol{p}_0^\mathrm{S}\,\boldsymbol{p}_0^\mathrm{D}$, preceded by a **source subpath** $\bar{\boldsymbol{p}}^\mathrm{S} := (\boldsymbol{p}_s^\mathrm{S}, \ldots, \boldsymbol{p}_1^\mathrm{S})$ connecting $\boldsymbol{p}_0^\mathrm{S}$ to the light source, and succeeded

[9] Anderson et al. [2017] presented a tri-directional path tracing algorithm that is conceptually similar to our approach. Their method, however, estimates the original path integral while ours focuses on the *boundary* term of our differential path integral.

by an **detector subpath** $\bar{\boldsymbol{p}}^{\mathrm{D}} := (\boldsymbol{p}_1^{\mathrm{D}}, \ldots, \boldsymbol{p}_t^{\mathrm{D}})$ connecting $\boldsymbol{p}_0^{\mathrm{D}}$ to the detector (i.e., the eye), respectively.

The main idea of our multi-directional sampling of a boundary light path is to first construct its boundary segment and then build the source and detector subpaths. To this end, we first rewrite the *boundary* term from Eq. (36), such that the contributions from these three parts are decoupled, as:

$$\int_{\partial\hat{\Omega}} \hat{f}^{\mathrm{S}}\, \hat{f}^{\mathrm{B}}\, \hat{f}^{\mathrm{D}}\, \mathrm{d}\mu', \tag{39}$$

where the terms

$$\hat{f}^{\mathrm{B}} := \Delta G(\boldsymbol{x}_0^{\mathrm{S}} \leftrightarrow \boldsymbol{x}_0^{\mathrm{D}})\, V_{\Delta\mathcal{B}}, \tag{40}$$

$$\hat{f}^{\mathrm{S}} := \hat{f}_{\mathrm{s}}(\boldsymbol{p}_1^{\mathrm{S}} \to \boldsymbol{p}_0^{\mathrm{S}} \to \boldsymbol{p}_0^{\mathrm{D}}) \prod_{n=1}^{s} \hat{f}_{\mathrm{s}}(\boldsymbol{p}_{n+1}^{\mathrm{S}} \to \boldsymbol{p}_n^{\mathrm{S}} \to \boldsymbol{p}_{n-1}^{\mathrm{S}})\, G(\boldsymbol{x}_{n-1}^{\mathrm{S}} \leftrightarrow \boldsymbol{x}_n^{\mathrm{S}}), \tag{41}$$

$$\hat{f}^{\mathrm{D}} := \hat{f}_{\mathrm{s}}(\boldsymbol{p}_0^{\mathrm{S}} \to \boldsymbol{p}_0^{\mathrm{D}} \to \boldsymbol{p}_1^{\mathrm{D}}) \prod_{n=1}^{t} \hat{f}_{\mathrm{s}}(\boldsymbol{p}_{n-1}^{\mathrm{D}} \to \boldsymbol{p}_n^{\mathrm{D}} \to \boldsymbol{p}_{n+1}^{\mathrm{D}})\, G(\boldsymbol{x}_{n-1}^{\mathrm{D}} \leftrightarrow \boldsymbol{x}_n^{\mathrm{D}}), \tag{42}$$

capture the contributions of the boundary segment $\boldsymbol{p}_0^{\mathrm{S}}\, \boldsymbol{p}_0^{\mathrm{D}}$, the source subpath $\bar{\boldsymbol{p}}^{\mathrm{S}}$ (given $\boldsymbol{p}_0^{\mathrm{S}}$), and the detector subpath $\bar{\boldsymbol{p}}^{\mathrm{D}}$ (given $\boldsymbol{p}_0^{\mathrm{D}}$) to the boundary contribution function, respectively. In Eqs. (40–42), $V_{\Delta\mathcal{B}}$ is the scalar normal velocity of $\boldsymbol{p}_0^{\mathrm{D}}$ with $\boldsymbol{p}_0^{\mathrm{S}}$ fixed; and $\hat{f}_{\mathrm{s}}$ follows the definition in Eq. (35).

By further separating the boundary path space and differential area-product measure in Eq. (39), we obtain the *multi-directional form* of the *boundary* integral from Eq. (36) as:

$$\int_{\mathcal{B}} \int_{\Delta\mathcal{B}} \left[\int_{\hat{\Omega}} \hat{f}^{\mathrm{S}}\, \mathrm{d}\mu(\bar{\boldsymbol{p}}^{\mathrm{S}})\right] \hat{f}^{\mathrm{B}} \left[\int_{\hat{\Omega}} \hat{f}^{\mathrm{D}}\, \mathrm{d}\mu(\bar{\boldsymbol{p}}^{\mathrm{D}})\right] \mathrm{d}\ell(\boldsymbol{p}_0^{\mathrm{D}})\, \mathrm{d}A(\boldsymbol{p}_0^{\mathrm{S}}). \tag{43}$$

6.3 Multi-Directional Sampling of Boundary Paths

We now present our Monte Carlo solution for estimating the multi-directional *boundary* integral (43). Our unbiased algorithm samples boundary paths by drawing the boundary segment $\boldsymbol{p}_0^{\mathrm{S}}\, \boldsymbol{p}_0^{\mathrm{D}}$ first followed by the source subpath $\bar{\boldsymbol{p}}^{\mathrm{S}}$ (given $\boldsymbol{p}_0^{\mathrm{S}}$) and detector subpath $\bar{\boldsymbol{p}}^{\mathrm{D}}$ (given $\boldsymbol{p}_0^{\mathrm{D}}$). In what follows, we provide a detail description of this process.

As stated in §6.2, we would like to sample boundary segments without performing explicit searches for silhouette edges. For $\boldsymbol{p}_0^{\mathrm{S}}\, \boldsymbol{p}_0^{\mathrm{D}}$ to be a boundary segment, its spatial counterpart $\boldsymbol{x}_0^{\mathrm{S}}\, \boldsymbol{x}_0^{\mathrm{D}}$ with $\boldsymbol{x}_0^{\mathrm{S}} = \mathsf{X}(\boldsymbol{p}_0^{\mathrm{S}}, \pi)$ and $\boldsymbol{x}_0^{\mathrm{D}} = \mathsf{X}(\boldsymbol{p}_0^{\mathrm{D}}, \pi)$ must intersect the evolving surface $\mathcal{M}(\pi)$ at exactly one point $\boldsymbol{x}^{\mathrm{B}}$ besides the endpoints.[10] This point is not a vertex of the path, but simply corresponds to a point on the silhouette of $\mathcal{M}(\pi)$ when viewed from $\boldsymbol{x}_0^{\mathrm{S}}$ or $\boldsymbol{x}_0^{\mathrm{D}}$.

Change of variables. To sample the boundary segment $\boldsymbol{p}_0^{\mathrm{S}}\, \boldsymbol{p}_0^{\mathrm{D}}$, we first perform a change of variables from $\boldsymbol{p}_0^{\mathrm{S}}$ and $\boldsymbol{p}_0^{\mathrm{D}}$ to $\boldsymbol{x}^{\mathrm{B}}$ and $\boldsymbol{\omega}^{\mathrm{B}} := \boldsymbol{x}_0^{\mathrm{S}} \to \boldsymbol{x}_0^{\mathrm{D}}$. It is easy to verify that the equations controlling this change of variables are

$$\boldsymbol{p}_0^{\mathrm{S}} = \mathsf{P}(\boldsymbol{x}_0^{\mathrm{S}}, \pi), \qquad \boldsymbol{p}_0^{\mathrm{D}} = \mathsf{P}(\boldsymbol{x}_0^{\mathrm{D}}, \pi), \tag{44}$$

[10] When Assumption A.2 is relaxed to allow jump discontinuities of the source emission L_{e} and the BSDF f_{s} within tangent planes (as discussed in footnote 7), a spatial boundary segment can also reside within the tangent plane of one of its endpoints.

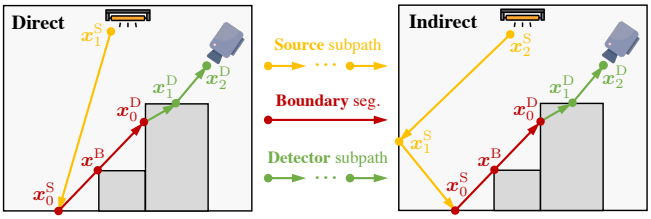


Fig. 8. **Multi-directional sampling of boundary paths:** To sample a boundary light path, we start with sampling the boundary segment $\boldsymbol{p}_0^{\mathrm{S}}\, \boldsymbol{p}_0^{\mathrm{D}}$ followed by a source subpath $\bar{\boldsymbol{p}}^{\mathrm{S}}$ and a detector subpath $\bar{\boldsymbol{p}}^{\mathrm{D}}$. To obtain $\boldsymbol{p}_0^{\mathrm{S}}\, \boldsymbol{p}_0^{\mathrm{D}}$, we start with sampling some $\boldsymbol{x}^{\mathrm{B}} \in \mathcal{M}(\pi)$ with $\boldsymbol{\omega}^{\mathrm{B}} \in \mathbb{S}^2$ and performing ray tracing to obtain $\boldsymbol{x}_0^{\mathrm{S}}$ and $\boldsymbol{x}_0^{\mathrm{D}}$, which in turn determine $\boldsymbol{p}_0^{\mathrm{S}}$ and $\boldsymbol{p}_0^{\mathrm{D}}$ via Eq. (44), respectively. The point $\boldsymbol{x}^{\mathrm{B}}$ itself is not a vertex of the sampled path. In the figure, we illustrate two *spatial* boundary paths with $\boldsymbol{x}_n^* = \mathsf{X}(\boldsymbol{p}_n^*, \pi)$ with $* \in \{\mathrm{S}, \mathrm{D}\}$. The arrows indicate the direction of the flow of light (and are independent of the sampling of the subpaths).

where $\boldsymbol{x}_0^{\mathrm{S}} = \mathrm{rayTrace}(\boldsymbol{x}^{\mathrm{B}}, -\boldsymbol{\omega}^{\mathrm{B}})$, $\boldsymbol{x}_0^{\mathrm{D}} = \mathrm{rayTrace}(\boldsymbol{x}^{\mathrm{B}}, \boldsymbol{\omega}^{\mathrm{B}})$, and P is the reference map of the motion X that transforms a material point to its spatial representation.

Based on this change of variables, we then rewrite the multi-directional *boundary* integral (43) as

$$\iint \underbrace{\left[\int_{\hat{\Omega}} \hat{f}^{\mathrm{S}}\, \mathrm{d}\mu(\bar{\boldsymbol{p}}^{\mathrm{S}})\right] \left[\hat{f}^{\mathrm{B}}\, J^{\mathrm{B}}(\boldsymbol{x}^{\mathrm{B}}, \boldsymbol{\omega}^{\mathrm{B}})\right] \left[\int_{\hat{\Omega}} \hat{f}^{\mathrm{D}}\, \mathrm{d}\mu(\bar{\boldsymbol{p}}^{\mathrm{D}})\right]}_{=: F^{\mathrm{B}}(\boldsymbol{x}^{\mathrm{B}}, \boldsymbol{\omega}^{\mathrm{B}})} \mathrm{d}\boldsymbol{\omega}^{\mathrm{B}}\, \mathrm{d}\boldsymbol{x}^{\mathrm{B}}, \tag{45}$$

where

$$J^{\mathrm{B}}(\boldsymbol{x}^{\mathrm{B}}, \boldsymbol{\omega}^{\mathrm{B}}) = \left|\frac{\mathrm{d}A(\boldsymbol{x}_0^{\mathrm{S}})\, \mathrm{d}\ell(\boldsymbol{x}_0^{\mathrm{D}})}{\mathrm{d}\boldsymbol{x}^{\mathrm{B}}\, \mathrm{d}\boldsymbol{\omega}^{\mathrm{B}}}\right| \left|\frac{\mathrm{d}A(\boldsymbol{p}_0^{\mathrm{S}})\, \mathrm{d}\ell(\boldsymbol{p}_0^{\mathrm{D}})}{\mathrm{d}A(\boldsymbol{x}_0^{\mathrm{S}})\, \mathrm{d}\ell(\boldsymbol{x}_0^{\mathrm{D}})}\right|, \tag{46}$$

is the product of two Jacobian determinants: the former captures the change of variables from $(\boldsymbol{x}_0^{\mathrm{S}}, \boldsymbol{x}_0^{\mathrm{D}})$ to $(\boldsymbol{x}^{\mathrm{B}}, \boldsymbol{\omega}^{\mathrm{B}})$, and the latter from material points $(\boldsymbol{p}_0^{\mathrm{S}}, \boldsymbol{p}_0^{\mathrm{D}})$ to spatial ones $(\boldsymbol{x}_0^{\mathrm{S}}, \boldsymbol{x}_0^{\mathrm{D}})$.

Monte Carlo estimator. As depicted in Figure 8, Monte Carlo estimation of Eq. (45) boils down to: (i) sampling $\boldsymbol{x}^{\mathrm{B}}$, $\boldsymbol{\omega}^{\mathrm{B}}$ and converting them to the boundary segment $\boldsymbol{p}_0^{\mathrm{S}}\, \boldsymbol{p}_0^{\mathrm{D}}$ using Eq. (44); (ii) building the two subpaths $\bar{\boldsymbol{p}}^{\mathrm{S}}$ and $\bar{\boldsymbol{p}}^{\mathrm{D}}$, which are original light paths themselves, using existing path sampling methods. With the full boundary path available, the corresponding single-sample estimator becomes:

$$\frac{\hat{f}^{\mathrm{S}}}{\mathbb{P}(\bar{\boldsymbol{p}}^{\mathrm{S}} \mid \boldsymbol{x}^{\mathrm{B}}, \boldsymbol{\omega}^{\mathrm{B}})} \frac{\hat{f}^{\mathrm{B}}\, J^{\mathrm{B}}(\boldsymbol{x}^{\mathrm{B}}, \boldsymbol{\omega}^{\mathrm{B}})}{\mathbb{P}(\boldsymbol{x}^{\mathrm{B}}, \boldsymbol{\omega}^{\mathrm{B}})} \frac{\hat{f}^{\mathrm{D}}}{\mathbb{P}(\bar{\boldsymbol{p}}^{\mathrm{D}} \mid \boldsymbol{x}^{\mathrm{B}}, \boldsymbol{\omega}^{\mathrm{B}})}, \tag{47}$$

where $\mathbb{P}(\boldsymbol{x}^{\mathrm{B}}, \boldsymbol{\omega}^{\mathrm{B}})$ denotes the joint probability density for sampling $\boldsymbol{x}^{\mathrm{B}}$ and $\boldsymbol{\omega}^{\mathrm{B}}$; $\mathbb{P}(\bar{\boldsymbol{p}}^{\mathrm{S}} \mid \boldsymbol{x}^{\mathrm{B}}, \boldsymbol{\omega}^{\mathrm{B}})$ and $\mathbb{P}(\bar{\boldsymbol{p}}^{\mathrm{D}} \mid \boldsymbol{x}^{\mathrm{B}}, \boldsymbol{\omega}^{\mathrm{B}})$ are, respectively, the conditional probability densities for sampling the two subpaths $\bar{\boldsymbol{p}}^{\mathrm{S}}$ and $\bar{\boldsymbol{p}}^{\mathrm{D}}$, given $\boldsymbol{x}^{\mathrm{B}}$ and $\boldsymbol{\omega}^{\mathrm{B}}$ (which in turn determine the boundary segment $\boldsymbol{p}_0^{\mathrm{S}}\, \boldsymbol{p}_0^{\mathrm{D}}$).

We summarize the estimation of the *boundary* integral (45) using Eq. (47) in Algorithm 2 (and will discuss the separation of direct and indirect paths in §6.4). To realize this Monte Carlo estimator, what is left now is to determine how to sample $\boldsymbol{x}^{\mathrm{B}}$ and $\boldsymbol{\omega}^{\mathrm{B}}$ (Line 3), which in turn requires to determine their integral domains, as well as how to compute the first Jacobian in Eq. (46).

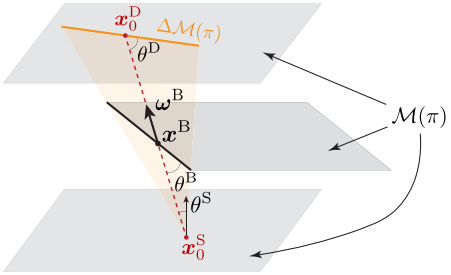


Fig. 9. We apply a **change of variable** from $(\boldsymbol{x}_0^{\mathrm{S}}, \boldsymbol{x}_0^{\mathrm{D}})$ to $(\boldsymbol{x}^{\mathrm{B}}, \boldsymbol{\omega}^{\mathrm{B}})$ for sampling spatial boundary segments. This figure illustrates the quantities needed to calculate the corresponding Jacobian determinant of Eq. (48) that is derived in Appendix B.

Jacobian determinant. When the scene geometry $\mathcal{M}(\pi)$ is specified using polygonal meshes, the integral domain of $\boldsymbol{x}^{\mathrm{B}}$ is the union of all face edges $\mathcal{E}(\pi)$ of the mesh. Further, $\mathrm{d}\boldsymbol{x}^{\mathrm{B}}$ and $\mathrm{d}\boldsymbol{\omega}^{\mathrm{B}}$, respectively, equal the curve-length and the solid-angle measures. This is because, for a boundary segment to touch a polygonal mesh at a single point $\boldsymbol{x}^{\mathrm{B}}$, this point must lie on the edge of a polygonal face. For each $\boldsymbol{x}^{\mathrm{B}} \in \mathcal{E}(\pi)$, the direction $\boldsymbol{\omega}^{\mathrm{B}}$ needs to satisfy the following two conditions:

- First, $\mathrm{rayTrace}(\boldsymbol{x}^{\mathrm{B}}, -\boldsymbol{\omega}^{\mathrm{B}})$ and $\mathrm{rayTrace}(\boldsymbol{x}^{\mathrm{B}}, \boldsymbol{\omega}^{\mathrm{B}})$ should successfully intersect $\mathcal{M}(\pi)$ at some $\boldsymbol{x}_0^{\mathrm{S}}$ and $\boldsymbol{x}_0^{\mathrm{D}}$, respectively.
- Second, the segment $\boldsymbol{x}_0^{\mathrm{S}}\,\boldsymbol{x}_0^{\mathrm{D}}$ should not penetrate $\mathcal{M}(\pi)$. Specifically, as shown in Figure 10, if the face edge containing $\boldsymbol{x}^{\mathrm{B}}$ is shared by two faces with normal vectors $\boldsymbol{n}$ and $\boldsymbol{n}'$, we need $(\boldsymbol{\omega}^{\mathrm{B}} \cdot \boldsymbol{n})\,(\boldsymbol{\omega}^{\mathrm{B}} \cdot \boldsymbol{n}') < 0$.

It follows that we can express the first Jacobian on the right-hand side of Eq. (46) as

$$\left|\frac{\mathrm{d}A(\boldsymbol{x}_0^{\mathrm{S}})\,\mathrm{d}\ell(\boldsymbol{x}_0^{\mathrm{D}})}{\mathrm{d}\ell(\boldsymbol{x}^{\mathrm{B}})\,\mathrm{d}\sigma(\boldsymbol{\omega}^{\mathrm{B}})}\right| = \left\|\boldsymbol{x}_0^{\mathrm{D}} - \boldsymbol{x}_0^{\mathrm{S}}\right\| \left\|\boldsymbol{x}^{\mathrm{B}} - \boldsymbol{x}_0^{\mathrm{S}}\right\| \frac{\sin\theta^{\mathrm{B}}}{\sin\theta^{\mathrm{D}}\,|\cos\theta^{\mathrm{S}}|}, \tag{48}$$

where θ^{B} and θ^{D} are the angles between $\boldsymbol{\omega}^{\mathrm{B}}$ and, respectively, the face edge at $\boldsymbol{x}^{\mathrm{B}}$ and the visibility boundary at $\boldsymbol{x}_0^{\mathrm{D}}$; and θ^{S} is the angle between $\boldsymbol{\omega}^{\mathrm{B}}$ and the surface normal at $\boldsymbol{x}_0^{\mathrm{S}}$ (see Figure 9). We provide a derivation of this result in Appendix B.

When using the global parameterization induced from the motion $\hat{\mathsf{X}}$ of Eq. (37) for some fixed $\pi_0 \in \mathbb{R}$, both $\hat{\mathsf{X}}(\cdot, \pi_0)$ and $\hat{\mathsf{P}}(\cdot, \pi_0)$ reduce to identity maps. Then, $\boldsymbol{p}_n^{\mathrm{S}} = \boldsymbol{x}_n^{\mathrm{S}}$ and $\boldsymbol{p}_n^{\mathrm{D}} = \boldsymbol{x}_n^{\mathrm{D}}$ for all n, the second term on the RHS of Eq. (46) reduces to one, and $J^{\mathrm{B}}(\boldsymbol{x}^{\mathrm{B}}, \boldsymbol{\omega}^{\mathrm{B}})$ equals Eq. (48).

6.4 Next-Event Estimation and Importance Sampling

Next-event estimation. To improve the efficiency of boundary-path sampling, we adopt next-event estimation (NEE), a technique widely used by forward rendering algorithms, as follows. We consider **direct boundary paths** $\bar{\boldsymbol{p}} \in \bigcup_{N=2}^{\infty} \partial\hat{\Omega}_{N,1}$, in analogy with *direct-illumination paths* in path tracing. Then, for each such path $\bar{\boldsymbol{p}} = (\boldsymbol{p}_0, \boldsymbol{p}_1, \ldots)$, the boundary segment coincides with the first segment $\boldsymbol{p}_0\,\boldsymbol{p}_1$ (i.e., $\boldsymbol{p}_0^{\mathrm{S}} = \boldsymbol{p}_0$ and $\boldsymbol{p}_0^{\mathrm{D}} = \boldsymbol{p}_1$), as shown in Figure 8. Accordingly,

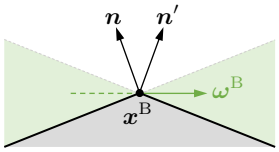


Fig. 10. **Sampling the boundary segment:** When $\boldsymbol{x}^{\mathrm{B}}$ lies on an edge of a polygonal mesh that is shared by two faces with normals $\boldsymbol{n}$ and $\boldsymbol{n}'$, the direction $\boldsymbol{\omega}^{\mathrm{B}}$ needs to satisfy $(\boldsymbol{\omega}^{\mathrm{B}} \cdot \boldsymbol{n})\,(\boldsymbol{\omega}^{\mathrm{B}} \cdot \boldsymbol{n}') < 0$ (i.e., the green region) for the resulting segment $\boldsymbol{x}_0^{\mathrm{S}}\,\boldsymbol{x}_0^{\mathrm{D}}$ to be a (spatial) boundary segment.

their contribution to the *boundary* integral (45) equals

$$\iint L_{\mathrm{e}}(\boldsymbol{x}_0^{\mathrm{S}} \to \boldsymbol{x}_0^{\mathrm{D}}) \left[\hat{f}^{\mathrm{B}}\, J^{\mathrm{B}}(\boldsymbol{x}^{\mathrm{B}}, \boldsymbol{\omega}^{\mathrm{B}})\right] \left[\int_{\hat{\Omega}} \hat{f}^{\mathrm{D}}\, \mathrm{d}\mu(\bar{\boldsymbol{p}}^{\mathrm{D}})\right] \mathrm{d}\boldsymbol{\omega}^{\mathrm{B}}\, \mathrm{d}\boldsymbol{x}^{\mathrm{B}}, \tag{49}$$

Because $\boldsymbol{x}_0^{\mathrm{S}} = \mathrm{rayTrace}(\boldsymbol{x}^{\mathrm{B}}, -\boldsymbol{\omega}^{\mathrm{B}})$ needs to lie on a light source, we restrict the sampling of $\boldsymbol{\omega}^{\mathrm{B}}$ to satisfy this condition. When $\mathcal{M}(\pi)$ is expressed as polygonal meshes, for instance, we draw $\boldsymbol{\omega}^{\mathrm{B}}$ by sampling a point on the light source.

In practice, this separation of direct and indirect boundary paths can be implemented by using different probability densities in Algorithm 2 and sampling the source subpaths accordingly (Lines 9 and 11).

Grid-based importance sampling. A naive way to sample the spatial point $\boldsymbol{x}^{\mathrm{B}}$ and direction $\boldsymbol{\omega}^{\mathrm{B}}$ (Line 3 of Algorithm 2) is by uniformly drawing $\boldsymbol{x}^{\mathrm{B}}$ followed by $\boldsymbol{\omega}^{\mathrm{B}}$. Unfortunately, this can results in estimates of high variance, due to the complexity of the integrand $F^{\mathrm{B}}(\boldsymbol{x}^{\mathrm{B}}, \boldsymbol{\omega}^{\mathrm{B}})$ of Eq. (45). Instead, we would like to sample $\boldsymbol{x}^{\mathrm{B}}$ and $\boldsymbol{\omega}^{\mathrm{B}}$ jointly, with a probability density proportional to the integrand:

ALGORITHM 2: Multi-directional estimation of the *boundary* integral of Eq. (45)

```
1  EstimateBoundaryIntegral(ℙ, direct)
   Input: Probability density ℙ for sampling the boundary segment,
          and a boolean direct for next-event estimation (NEE)
2  begin
       /* Sample boundary segment                                   */
3      Draw (x^B, ω^B) ~ ℙ(x^B, ω^B);
4      x_0^S ← rayTrace(x^B, −ω^B); x_0^D ← rayTrace(x^B, ω^B);
5      if x_0^S and x_0^D are both valid then // Both ray tracings hit
6          T^B ← f̂^B J^B(x^B, ω^B)/ℙ(x^B, ω^B);
7          p_0^S ← P(x_0^S, π); p_0^D ← P(x_0^D, π);          // Eq. (44)
           /* Sample subpaths                                       */
8          if direct then                                  // For direct path
9              T^S ← L_e(x_0^S → x_0^D);
10         else                                          // For indirect path
11             T^S ← EstimateSourcePath(p_0^S; p_0^D);
12         end
13         T^D ← EstimateDetectorPath(p_0^D; p_0^S);
14         return T^S T^B T^D;                                  // Eq. (47)
15     else
16         return 0;
17     end
18 end
```

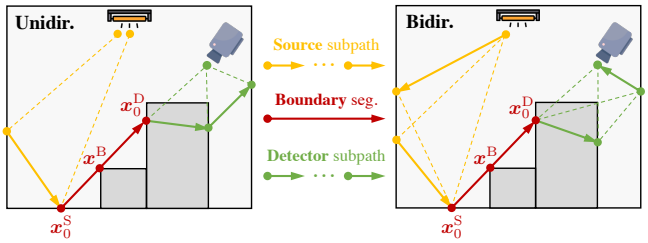


Fig. 11. **Estimating the multi-directional *boundary* integral** of Eq. (45): Our *unidirectional* algorithm (I.1) samples the source and detector subpaths using unidirectional tracing with next-event estimation (NEE). The *bidirectional* variant (I.2), on the contrary, uses bidirectional path tracing (BDPT) to construct both subpaths.

$\mathbb{P}(\boldsymbol{x}^{\mathrm{B}}, \boldsymbol{\omega}^{\mathrm{B}}) \propto F^{\mathrm{B}}(\boldsymbol{x}^{\mathrm{B}}, \boldsymbol{\omega}^{\mathrm{B}})$. Although this probability is difficult to compute analytically, we note that the total dimensionality of the domains of $\boldsymbol{x}^{\mathrm{B}}$ and $\boldsymbol{\omega}^{\mathrm{B}}$ is only three (e.g., when $\mathcal{M}(\pi)$ is expressed as polygonal meshes, $\boldsymbol{x}^{\mathrm{B}}$ belongs to a 1D manifold of mesh face edges, while $\boldsymbol{\omega}^{\mathrm{B}}$ lies on a subset of the 2D sphere of directions). We take advantage of this low dimensionality to develop a simple method for importance sampling $\boldsymbol{x}^{\mathrm{B}}$ and $\boldsymbol{\omega}^{\mathrm{B}}$ as follows.

We discretize $\mathbb{P}(\boldsymbol{x}^{\mathrm{B}}, \boldsymbol{\omega}^{\mathrm{B}})$ as a regular 3D grid that is precomputed before the rendering process. Inspired from standard path guiding using photon maps [Jensen 1995], we start the preprocessing with generating a *photon* map (by tracing photons carrying radiance information from the light source) and an *importon* map (by tracing importons carrying importance information from the detector). Then, we integrate $F^{\mathrm{B}}(\boldsymbol{x}^{\mathrm{B}}, \boldsymbol{\omega}^{\mathrm{B}})$ within each cell C_i by uniformly sampling $\boldsymbol{x}^{\mathrm{B}}$ and $\boldsymbol{\omega}^{\mathrm{B}}$:

$$\mathbb{P}_i = \int_{C_i} F^{\mathrm{B}}(\boldsymbol{x}^{\mathrm{B}}, \boldsymbol{\omega}^{\mathrm{B}}) \, \mathrm{d}\boldsymbol{\omega}^{\mathrm{B}} \, \mathrm{d}\boldsymbol{x}^{\mathrm{B}}, \tag{50}$$

which provides a piecewise constant representation of $\mathbb{P}(\boldsymbol{x}^{\mathrm{B}}, \boldsymbol{\omega}^{\mathrm{B}})$.

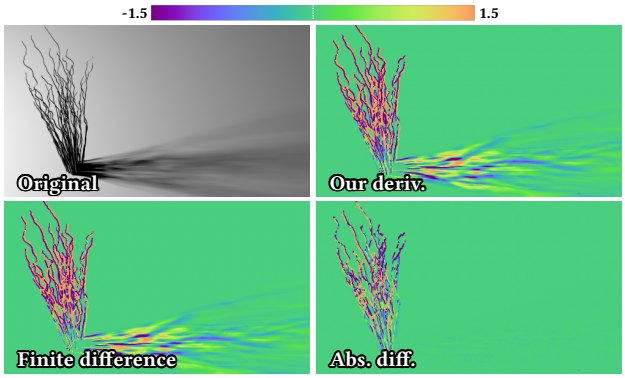


Fig. 12. **Validation** of our *unidirectional* algorithm (I.1). In this example, the derivative is computed with respect to the rotation angle of the branches around the vertical axis. Derivatives estimated with our method closely match those obtained using the finite-difference (FD) approach with the main difference due to the FD bias.

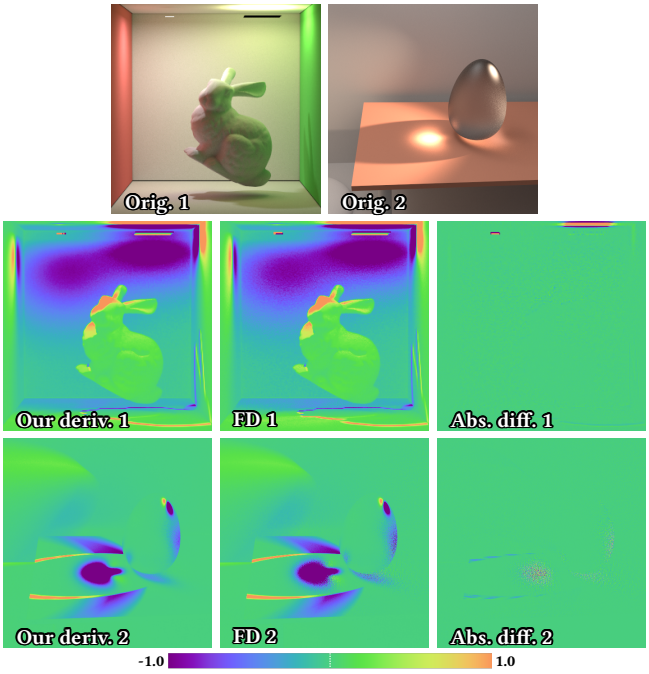


Fig. 13. **Validation** of our *bidirectional* algorithm (I.2). In the top example, the differentiation is with respect to the horizontal displacement of both area light sources. In the bottom example, the derivatives are computed with respect to the vertical location of a spot light. Both examples involve light transport effects that are challenging for unidirectional methods. Derivatives estimated with our method closely match those obtained using the finite-difference (FD) approach with the main difference due to the FD bias.

To approximate the two integrals in $F^{\mathrm{B}}(\boldsymbol{x}^{\mathrm{B}}, \boldsymbol{\omega}^{\mathrm{B}})$, we leverage kernel density estimation using the pre-generated photon and importon maps. Specifically, given $\boldsymbol{x}_0^{\mathrm{S}}$ and $\boldsymbol{x}_0^{\mathrm{D}}$, by performing a nearest-neighbor (NN) search in the photon map around $\boldsymbol{x}_0^{\mathrm{S}}$, we can approximate the contribution of the source subpath as

$$\int_{\hat{\Omega}} \hat{f}^{\mathrm{S}} \, \mathrm{d}\mu \approx \frac{1}{A} \sum_p \hat{f}_{\mathrm{s}}(\boldsymbol{x}_0^{\mathrm{S}}, \boldsymbol{\omega}_p, \boldsymbol{\omega}^{\mathrm{B}}) \, \Phi_p, \tag{51}$$

where A is the surface area of the search neighborhood, Φ_p denotes the power of the p-th photon in the neighborhood, and $\boldsymbol{\omega}_p$ is the photon's incident direction. A similar estimate can be formed using the importon map for the contribution $\int_{\hat{\Omega}} \hat{f}^{\mathrm{D}} \, \mathrm{d}\mu$ of the detector subpath.

We emphasize that, even though our estimate $\mathbb{P}(\boldsymbol{x}^{\mathrm{B}}, \boldsymbol{\omega}^{\mathrm{B}})$ is biased, the resulting estimator of the *boundary* integral (45) remains unbiased, as $\mathbb{P}(\boldsymbol{x}^{\mathrm{B}}, \boldsymbol{\omega}^{\mathrm{B}})$ is only used to importance sample $\boldsymbol{x}^{\mathrm{B}}$ and $\boldsymbol{\omega}^{\mathrm{B}}$. In practice, we precompute two probability densities $\mathbb{P}_{\mathrm{direct}}(\boldsymbol{x}^{\mathrm{B}}, \boldsymbol{\omega}^{\mathrm{B}})$ and $\mathbb{P}_{\mathrm{indirect}}(\boldsymbol{x}^{\mathrm{B}}, \boldsymbol{\omega}^{\mathrm{B}})$ for importance sampling the direct and the indirect boundary paths, respectively.

7 RESULTS

To evaluate the effectiveness of our technique, we implement two path-space algorithms based on the Monte Carlo methods introduced in §6 as follows.

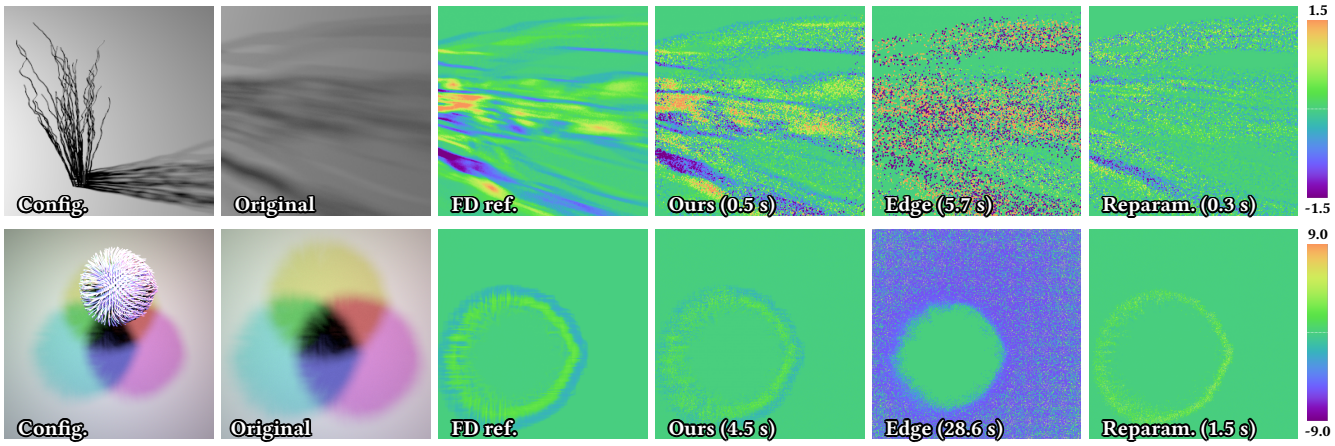


Fig. 14. **Evaluation** of the effectiveness of our *unidirectional* algorithm (**I.1**). Images in the left column visualize the overall scene configurations. All derivative images (other than the finite-difference reference) are generated under *equal sample*. Our method runs much faster than edge sampling and produces much cleaner derivative images.

I.1 Our **unidirectional algorithm** uses differentiable unidirectional path tracing (Algorithm 1) for the *interior* integral and a multi-directional estimator (Algorithm 2) with next-event estimation and grid-based importance sampling for the *boundary* integral. When building the source and detector subpaths, it uses a unidirectional scheme (that starts from $\boldsymbol{p}_0^{\mathrm{S}}$ and $\boldsymbol{p}_0^{\mathrm{D}}$ for the two subpaths, respectively), as illustrated on the left of Figure 11.

I.2 Our **bidirectional algorithm** estimates the *interior* integral using bidirectional path tracing (that is, the bidirectional counterpart of Algorithm 1). For the *boundary* integral, the source and detector subpaths are both sampled in a bidirectional fashion, as shown on the right of Figure 11. The source subpath, for instance, is constructed by sampling from both $\boldsymbol{p}_0^{\mathrm{S}}$ and a light source.

7.1 Validation

To validate the correctness of our derivations and implementations, we compare radiance derivatives estimated with our method to those obtained using the finite difference (FD) method.

We validate our *unidirectional* algorithm (**I.1**) in Figure 12 using a test scene that contains an object comprised of many branches. We compute the derivative images with respect to the rotation angle of the object.

Our *bidirectional* algorithm (**I.2**) is validated in Figure 13. The first example in this figure contains a Cornell-box-like scene lit by two area light sources with the right one facing upward. The derivative images are computed with respect to horizontal displacements of both lights. The second example is modeled after the well known scene created by Veach [1997] for demonstrating the effectiveness of bidirectional path tracing (BDPT). This scene involves a large floor lamp, a small spot light, and a glass egg on a table, and we use a camera setting to focus on the egg. We compute the derivatives with respect to the vertical displacement of the spot light.

In both figures, our results match those generated by the finite-difference method in *much longer* time. The small differences are due to the bias introduced by applying finite difference. This bias can be reduced by using smaller spacing but at the cost of significantly higher Monte Carlo noise. Our technique is capable of producing much cleaner and unbiased derivative estimation.

7.2 Evaluations

Thanks to our differential path integral formulations, our Monte Carlo algorithms (**I.1** and **I.2**) are capable of handling efficiently complex geometric discontinuities and light transport effects. In what follows, we evaluate the effectiveness of our method on both aspects. We compare our results to those generated using the (unbiased) edge-sampling method [Li et al. 2018a; Zhang et al. 2019] and the (biased) reparameterization method [Loubet et al. 2019]. We use two kinds of configurations for these comparisons: (i) scenes with complex geometry and occlusion; and (ii) those with light transport effects that are known to make unidirectional methods inefficient (e.g., caustics).

Complex geometry. Previously, sampling points from *silhouette edges* of a surface point (i.e., *edge sampling*) was generally required to obtain unbiased derivative estimates [Li et al. 2018a; Zhang et al. 2019] with respect to the scene geometry. This process, however, can be highly expensive for scenes with complex geometries. Another solution is to trade unbiasedness for computational efficiency by applying a local reparameterization [Loubet et al. 2019]. This method relies on a number of simplifying assumptions that can be violated in scenes with complex motions, making the resulting derivatives too biased for inverse rendering applications.

We use two test scenes with complex geometry to evaluate the performance of our *unidirectional* algorithm (**I.1**) as follows. The **branches** scene, which has been used in Figure 12, contains several branches lit by a small area source, causing complex visibility changes that can be observed from the shadows on the ground;

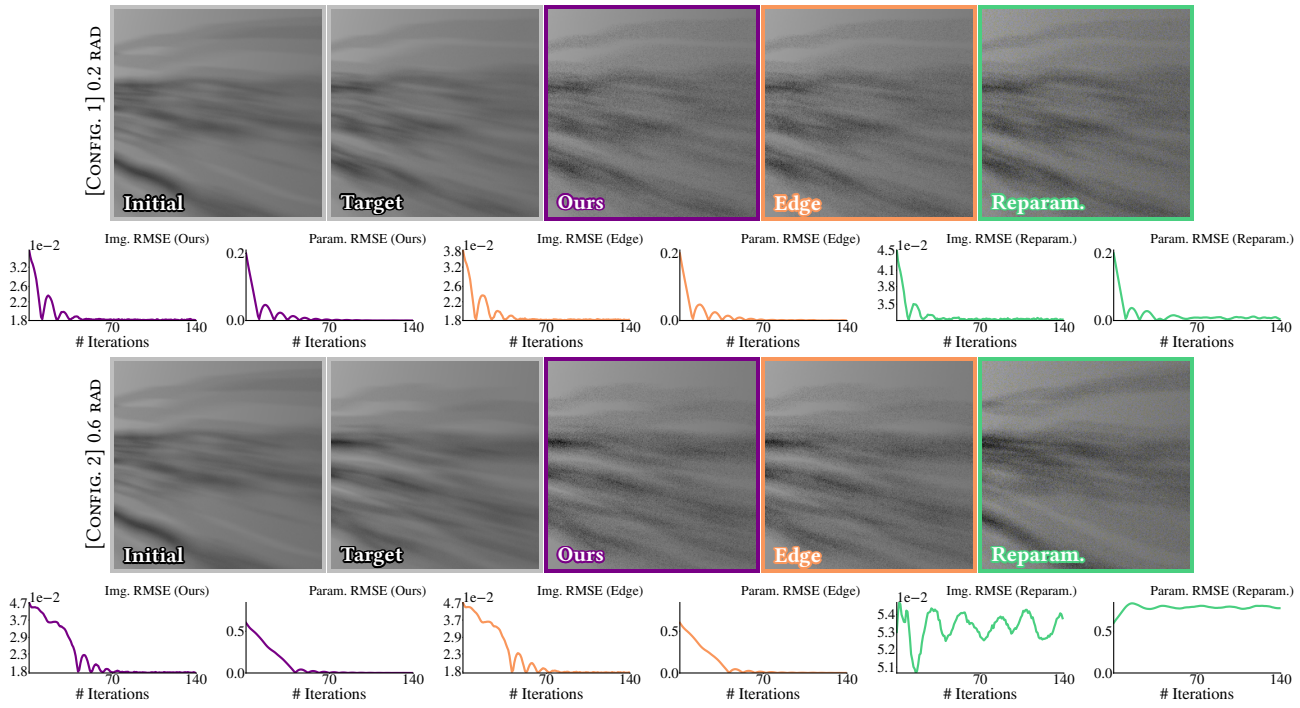


Fig. 15. **Inverse rendering comparisons** using the **branches** scene with the object's rotation angle being optimized. All methods are configured to have *equal sample* count per pixel. We show rendered images produced by each method after the last iteration with ours marked in purple, edge sampling in orange, and reparameterization in green. Notice that these images are noisy as we use low sample count during optimization. The image and parameter RMSE plots are color-coded the same way, and the latter is not used for optimization. Our *unidirectional* implementation runs much faster than edge sampling while preserving unbiasedness. The reparameterization-based method fails to converge to the correct solution under the second configuration due to its bias.

and the differentiation is with respect to the rotation angle of the branches around the vertical axis. The **puffer ball** scene involves a highly-detailed mesh generated via physics-based simulation [Zheng and James 2012]. This model contains over one million faces and is illuminated by three area emitters of red, green, and blue colors, creating the colored shadows on the ground. For each light, we use a single parameter to control its size and intensity such that the total power remains constant. The derivative images are computed with respect to the parameter controlling the red light (which casts a blue shadow).

We show in Figure 14 *equal-sample* comparisons[11] of derivative images computed by our method as well as the state-of-the-art edge sampling [Li et al. 2018a; Zhang et al. 2019] and reparameterization [Loubet et al. 2019] methods. For both scenes, our results closely matches the references generated using the finite-difference method. Edge sampling, despite being unbiased, struggled to produce clean results. Compared to edge sampling, our method is both *faster* and provides derivative estimates with *much lower noise*. The reparameterization method, on the other hand, generates clean results but with high bias.

[11] We use CPU-based implementations of both our algorithms (**I.1** and **I.2**) and the edge-sampling ones [Li et al. 2018a; Zhang et al. 2019]. The reparameterization method [Loubet et al. 2019], on the other hand, replies on a GPU-based implementation. Due to this architectural difference, we opt for equal-sample instead of equal-time comparisons, as the former are more representative of different methods' relative performance.

Additionally, we show inverse rendering comparisons using the same test scenes. We use the Adam method [Kingma and Ba 2014] implemented in PyTorch for the optimizations. In each comparison, we use derivative images generated at *equal sample* with prior methods and ours. To ensure fairness, we fix all inverse-rendering parameters other than the derivative images such as initial state and learning rate. Please refer to Table 2 for performance statistics and the supplemental material for animated versions of these results.

Figure 15 shows inverse rendering results using the **branches** scene with two settings that have identical initial configurations but different targets that are, respectively, 0.2 and 0.6 radian from the initial. Under the first setting, all methods including the biased reparameterization method, manage to converge to the global optimum; under the second setting, on the other hand, the reparameterization approach fails converge properly due to its high bias. Under both settings, our method runs significantly faster than edge sampling while producing much cleaner derivatives.

In Figure 16, we show inverse-rendering processes of the *puffer ball* scene. Due to the very high face count, edge sampling produces too much noise for the optimization to converge properly. Our technique again produces clean and unbiased derivative estimates, allowing the optimization to converge easily.

Complex light transport effect. Another major benefit of our theory is to allow the *interior* term (and subpath contributions in the

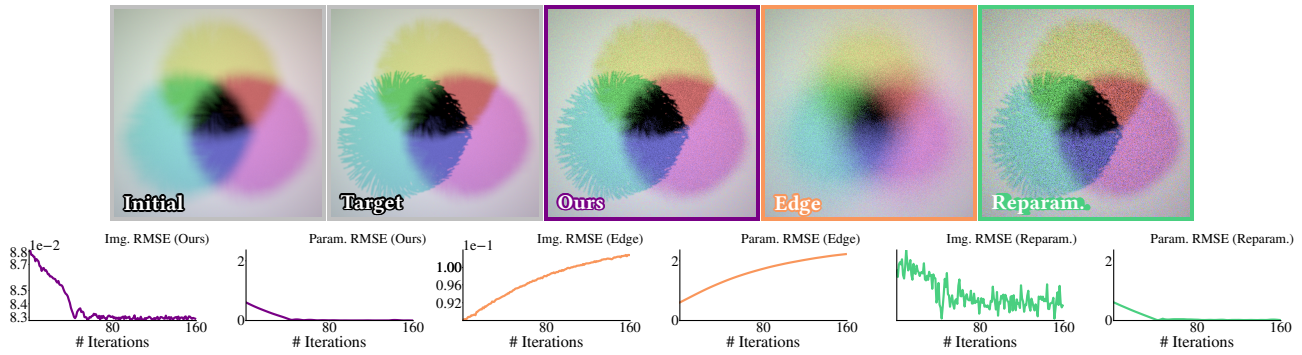


Fig. 16. **Inverse rendering comparisons** using the **puffer ball** scene with the light source sizes being optimized. All methods are configured to use equal sample count per pixel, and the visualization scheme follows that of Figure 15. Because of the high geometric complexity of this scene, edge sampling suffers from very high noise and fails to converge.

boundary term) to be estimated using sophisticated methods such as bidirectional path tracing. We use the **Veach egg** scene to evaluate the performance of our *bidirectional* algorithm (**I.2**). This scene remains largely identical to the one used for validation in Figure 13, except for using a lower roughness for the glass egg.

In Figure 17, we show derivatives with respect to the vertical displacement of the spot light estimated using our *bidirectional* algorithm (**I.2**), *unidirectional* algorithm (**I.1**), edge sampling, and biased reparameterization. All results (other than the finite-difference reference) are generated under *equal sample* per pixel. Our *bidirectional* algorithm (**I.2**) outperforms the others significantly by producing clean derivatives estimates in the caustics area.

We further demonstrate the advantage of our *bidirectional* algorithm (**I.2**) using an inverse-rendering setup where the position of the spot light and the refractive index of the glass egg are optimized jointly. We compare the performance of our *unidirectional* (**I.1**) and *bidirectional* (**I.2**) methods as well as edge sampling. We adjust the sample count so that each iteration takes roughly *equal time* for all methods. We do not include the reparameterization method [Loubet et al. 2019] for this comparison as its implementation does not support derivatives with respect to refractive indices.

As shown in Figure 18, gradients estimated with edge sampling are too noisy for the optimization to converge properly. Those produced by our *unidirectional* algorithm (**I.1**) have higher quality but are still noisy, preventing the optimization from finding to the exact solution. The *bidirectional* variant (**I.2**), on the other hand, produce significantly cleaner gradient estimates that allow the optimization to converge smoothly to the global optimum.

7.3 Additional Inverse-Rendering Results

We now provide two extra inverse-rendering results generated using our *bidirectional* algorithm (**I.2**).

Figure 19 shows an example where a glass mug is lit from the inside by a small area light, creating complex caustics patterns on the table below. We jointly optimize the orientation and roughness of the mug as well as the placement of the small area light. Figure 20 contains a silver ring illuminated by four area lights with different colors. We optimize the cross-sectional shape of the ring, which is parameterized by 100 free variables, to match the caustics pattern in the target image.

In both examples, small perturbations of the scene geometry can yield much more significant changes in the images. Our method is capable of producing low-noise derivative estimations, allowing the inverse-rendering optimizations in both examples to converge smoothly. Please refer to the supplemental material for animated versions of these results.

Table 2. Performance statistics for the inverse-rendering comparisons in Figures 15, 16, and 18. The "*time*" numbers indicate average computation time (in seconds) per iteration, including the overhead (shown in parentheses) for precomputing importance-sampling grid (discussed in §6.4). The "*RMSE*" numbers measure the differences between estimated derivatives and the corresponding groundtruth (calculated under initial configurations shared by all methods). The experiments are conducted on a workstation equipped with an octa-core Intel i7-7820X CPU and an Nvidia Titan RTX graphics card.

Scene	**Branches**		**Puffer ball**		**Veach egg**	
# param./# iter.	1/140		3/160		3/200	
	time	**RMSE**	**time**	**RMSE**	**time**	**RMSE**
Our unidir.	0.5 (0.1)	0.52	4.5 (2.0)	0.09	19.7 (0.2)	9.73
Our bidir.	—	—	—	—	19.7 (0.2)	2.43
Edge	5.7	3.60	28.6	1.16	19.7	112
Reparam.	0.3	0.57	1.5	0.08	—	—

8 DISCUSSION AND CONCLUSION

Limitations and future work. Our derivations in §4 and §5 have focused on the surface-only case, and generalizing them to handle volumetric light transport governed by radiative transfer [Chandrasekhar 1960] will be an important future direction. Additionally, although our material-form reformulation theoretically captures primary-sample-space rendering, the algorithms we realized in §6 have largely focused on the path space. Thus, developing advanced Monte Carlo estimators that work in the primary sample space is an interesting future topic. Furthermore, the original path integral

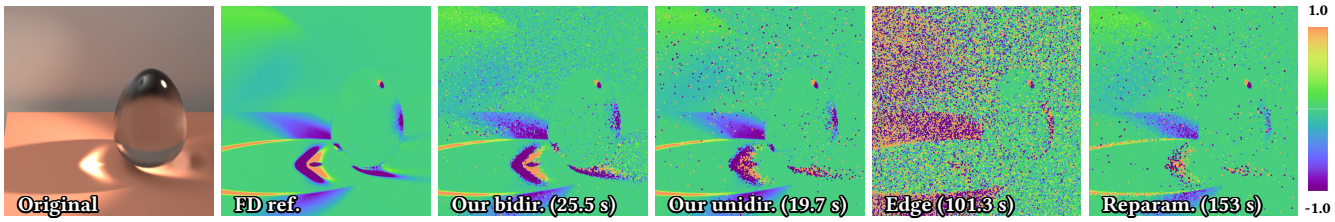


Fig. 17. **Evaluation** of the effectiveness of our *bidirectional* algorithm (I.2) using the **Veach egg** scene. All derivative images (other than the finite-difference reference) are generated under *equal sample*. Previous methods all rely on unidirectional path tracing, which works poorly in this example. Our *bidirectional* method, on the other hand, utilizes bidirectional path tracing and produces much cleaner results.

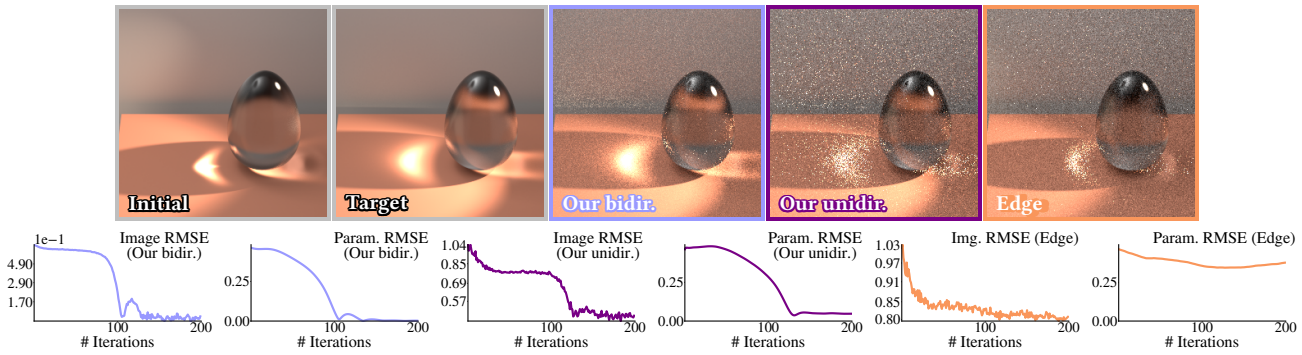


Fig. 18. **Inverse rendering comparison** using the **Veach egg** scene with the spot light's location and the glass egg's refractive index optimized jointly. The optimizations are configured so that each iteration takes *equal time* for all methods, and the visualization scheme follows that of Figure 15. Edge sampling produces too much noise for the optimization to converge properly. Our *unidirectional* algorithm (I.1) offers gradient estimates that are much cleaner but still too noisy for a convergence to the exact solution. Our the *bidirectional* method (I.2), on the contrary, allows the optimization to converge smoothly to the correct solution.

formulation has been the foundation of many Markov-Chain Monte Carlo (MCMC) rendering algorithms. Therefore, introducing new MCMC techniques based on our theory will enable differentiable rendering for even more challenging situations.

Conclusion. In this paper, we introduced the theoretical framework of *differential path integral* (in spatial and material forms) for physics-based differentiable rendering. We showed that the derivative of a path integral (with respect to arbitrary differential change of the scene) equals the sum of completely separated *interior* and *boundary* components expressed as path integrals over the original and boundary path spaces, respectively. This path-integral expression allows the design of new Monte Carlo estimators for the *interior* and *boundary* integrals. Specifically, based on our material-form formulation, we adapted unidirectional and bidirectional path tracing for the *interior* integral, and developed a multi-directional method to estimate the *boundary* component without explicitly searching for silhouettes. We demonstrated the effectiveness of our Monte Carlo methods via a few derivative-estimation and inverse-rendering examples.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their constructive comments. This work was supported by NSF grants 1730147, 1900849 and 1900927, as well as a gift from the AWS Cloud Credits for Research program.

REFERENCES

Luke Anderson, Tzu-Mao Li, Jaakko Lehtinen, and Frédo Durand. 2017. Aether: an embedded domain specific sampling language for Monte Carlo rendering. *ACM Trans. Graph.* 36, 4 (2017), 99:1–99:16.

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)*.

Paolo Cermelli, Eliot Fried, and Morton E Gurtin. 2005. Transport relations for surface integrals arising in the formulation of balance laws for evolving fluid interfaces. *Journal of Fluid Mechanics* 544 (2005), 339–351.

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

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.

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.

Andreas Griewank and Andrea Walther. 2008. *Evaluating derivatives: principles and techniques of algorithmic differentiation*. Vol. 105. Siam.

Pavel Grinfeld. 2013. *Introduction to tensor analysis and the calculus of moving surfaces*. Springer.

Morton E Gurtin. 1981. *An introduction to continuum mechanics*. Academic press.

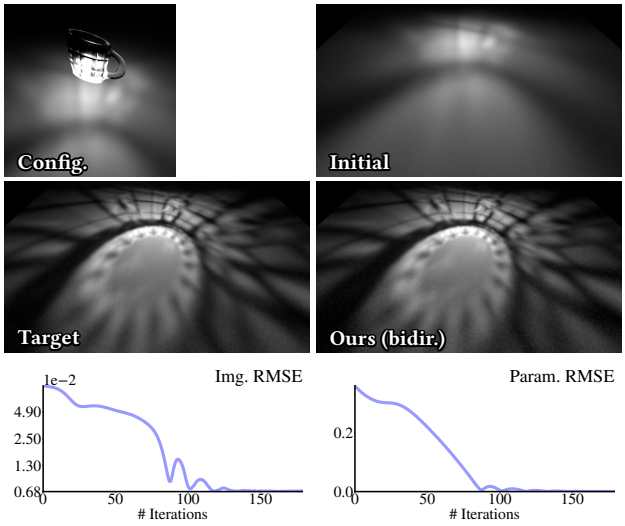


Fig. 19. **Inverse rendering result** where a mug with a small area source inside casts complex caustics on the ground, as visualized by the image marked with "Config". We jointly optimize three parameters: the orientation and roughness of a mug as well as the position of the light. Each iteration of the optimization takes 29.8 seconds (on a workstation equipped with an eight-core Intel i7-7820X CPU).

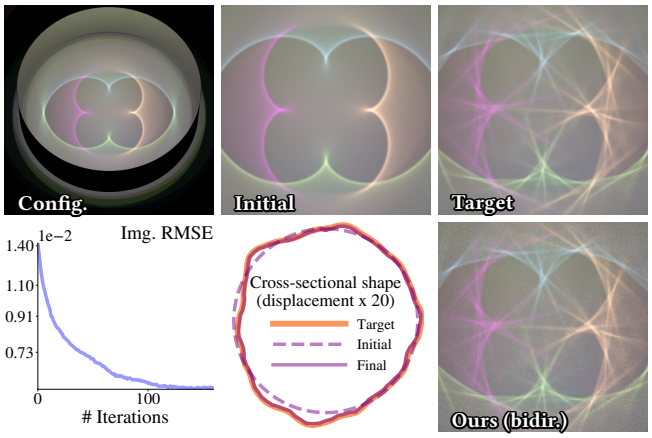


Fig. 20. **Shape optimization:** In this example, we optimize the cross-sectional shape of a ring parameterized with 100 variables to match the colored caustics on the ground. The image marked with "Config" visualizes the overall configuration, and the cross-sectional displacements are exaggerated by 20 times for visualization. Each iteration of this shape optimization takes 69.3 seconds.

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.

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.

Henrik Wann Jensen. 1995. Importance driven path tracing using the photon map. In *Rendering Techniques' 95*. Springer, 326–335.

James T. Kajiya. 1986. The Rendering Equation. *SIGGRAPH Comput. Graph.* 20, 4 (1986), 143–150.

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).

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.

Guillaume Loubet, Nicolas Holzschuch, and Wenzel Jakob. 2019. Reparameterizing discontinuous integrands for differentiable rendering. *ACM Trans. Graph.* 38, 6 (2019).

James L McClelland, David E Rumelhart, PDP Research Group, et al. 1986. Parallel distributed processing. *Explorations in the microstructure of cognition* 2 (1986), 216–271.

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.

Merlin Nimier-David, Delio Vicini, Tizian Zeltner, and Wenzel Jakob. 2019. Mitsuba 2: a retargetable forward and inverse renderer. *ACM Transactions on Graphics (TOG)* 38, 6 (2019), 203.

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

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.

Osborne Reynolds. 1903. *Papers on mechanical and physical subjects: the sub-mechanics of the universe*. Vol. 3. The University Press.

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.

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)*.

Eric Veach. 1997. *Robust Monte Carlo methods for light transport simulation*. Vol. 1610. Stanford University PhD thesis.

Eric Veach and Leonidas Guibas. 1995. Bidirectional estimators for light transport. In *Photorealistic Rendering Techniques*. Springer, 145–167.

Eric Veach and Leonidas J. Guibas. 1997. Metropolis Light Transport. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '97)*. ACM Press/Addison-Wesley Publishing Co., 65–76.

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.

Robert Edwin Wengert. 1964. A simple automatic derivative evaluation program. *Commun. ACM* 7, 8 (1964), 463–464.

Andrew P. Witkin and Paul S. Heckbert. 1994. Using particles to sample and control implicit surfaces. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '94)*. ACM, 269–277.

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.

Cheng Zhang, Lifan Wu, Changxi Zheng, Ioannis Gkioulekas, Ravi Ramamoorthi, and Shaung Zhao. 2019. A differential theory of radiative transfer. *ACM Trans. Graph.* 38, 6 (2019), 227:1–227:16.

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.
Changxi Zheng and Doug L. James. 2012. Energy-based self-collision culling for arbitrary mesh deformations. *ACM Trans. Graph.* 31, 4 (2012), 98:1–98:12.

A DERIVATION OF $\dot{I}_N$

We now derive $\partial I_N/\partial\pi$ in Eq. (25) using the recursive relations provided by Eqs. (21) and (24). Let

$$h_n^{(0)} \coloneqq \left[\prod_{n'=n+1}^{N} g(\boldsymbol{x}_{n'};\, \boldsymbol{x}_{n'-2}, \boldsymbol{x}_{n'-1})\right] W_\mathrm{e}(\boldsymbol{x}_N \to \boldsymbol{x}_{N-1}), \quad (52)$$

$$h_n^{(1)} \coloneqq \sum_{n'=n+1}^{N} \kappa(\boldsymbol{x}_{n'})\, V(\boldsymbol{x}_{n'}), \quad (53)$$

$$\Delta h_{n,n'}^{(0)} \coloneqq h_n^{(0)}\, {}_{\Delta g(\boldsymbol{x}_{n'};\, \boldsymbol{x}_{n'-2}, \boldsymbol{x}_{n'-1})}/_{g(\boldsymbol{x}_{n'};\, \boldsymbol{x}_{n'-2}, \boldsymbol{x}_{n'-1})}, \quad (54)$$

for $0 \le n < n' \le N$. We omit the dependencies of $h_n^{(0)}$, $h_n^{(1)}$, and $\Delta h_{n,n'}^{(0)}$ on $\boldsymbol{x}_{n+1}, \ldots, \boldsymbol{x}_N$ for notational convenience.

We now show that, for all $0 \le n < N$, it holds that

$$h_n(\boldsymbol{x}_n;\, \boldsymbol{x}_{n-1}) = \int_{\mathcal{M}^{N-n}} h_n^{(0)} \prod_{n'=n+1}^{N} \mathrm{d}A(\boldsymbol{x}_{n'}), \quad (55)$$

and

$$\dot{h}_n(\boldsymbol{x}_n;\, \boldsymbol{x}_{n-1}) = \int_{\mathcal{M}^{N-n}} \left[\left(h_n^{(0)}\right)^{\cdot} - h_n^{(0)} h_n^{(1)}\right] \prod_{n'=n+1}^{N} \mathrm{d}A(\boldsymbol{x}_{n'}) + \sum_{n'=n+1}^{N} \int \Delta h_{n,n'}^{(0)}\, V_{\overline{\partial\mathcal{M}_{n'}}}(\boldsymbol{x}_{n'})\, \mathrm{d}\ell(\boldsymbol{x}_{n'}) \prod_{\substack{n<i\le N \\ i\ne n'}} \mathrm{d}A(\boldsymbol{x}_i), \quad (56)$$

where the integral domain of the second term on the right-hand side, which is omitted for notational clarity, is $\mathcal{M}(\pi)$ for each $\boldsymbol{x}_i$ with $i \ne n'$ and $\overline{\partial\mathcal{M}}_{n'}(\pi)$, which depends on $\boldsymbol{x}_{n'-1}$, for $\boldsymbol{x}_{n'}$.

It is easy to verify that Eqs. (55) and (56) hold for $n = N-1$. We now show that, if they hold for some $0 < n < N$, then it is also the case for $n-1$. Let $g_{n-1} \coloneqq g(\boldsymbol{x}_n;\, \boldsymbol{x}_{n-2}, \boldsymbol{x}_{n-1})$ for all $0 < n \le N$. Then,

$$\begin{aligned} h_{n-1}(\boldsymbol{x}_{n-1};\, \boldsymbol{x}_{n-2}) &= \int_{\mathcal{M}} g_{n-1} \int_{\mathcal{M}^{N-n}} h_n^{(0)} \prod_{n'=n+1}^{N} \mathrm{d}A(\boldsymbol{x}_{n'})\, \mathrm{d}A(\boldsymbol{x}_n) \\ &= \int_{\mathcal{M}^{N-n+1}} h_{n-1}^{(0)} \prod_{n'=n}^{N} \mathrm{d}A(\boldsymbol{x}_{n'}), \end{aligned} \quad (57)$$

and

$$\begin{aligned} &\dot{h}_{n-1}(\boldsymbol{x}_{n-1};\, \boldsymbol{x}_{n-2}) \\ &= \int_{\mathcal{M}} \left[\dot{g}_{n-1}\, h_n + g_{n-1}(\dot{h}_n - h_n\, \kappa(\boldsymbol{x}_n)\, V(\boldsymbol{x}_n))\right] \mathrm{d}A(\boldsymbol{x}_n) \\ &\quad + \int_{\overline{\partial\mathcal{M}}_n} \Delta g_{n-1}\, h_n\, V_{\overline{\partial\mathcal{M}}_n}\, \mathrm{d}\ell(\boldsymbol{x}_n) \\ &= \int_{\mathcal{M}^{N-n+1}} \left\{\dot{g}_{n-1}\, h_n^{(0)} + g_{n-1}\left[\left(h_n^{(0)}\right)^{\cdot} - h_n^{(0)} h_{n-1}^{(1)}\right]\right\} \prod_{n'=k}^{N} \mathrm{d}A(\boldsymbol{x}_{n'}) \\ &\quad + \sum_{n'=n+1}^{N} \int g_{n-1}\, \Delta h_{n,n'}^{(0)}\, V_{\overline{\partial\mathcal{M}}_{n'}}(\boldsymbol{x}_{n'})\, \mathrm{d}\ell(\boldsymbol{x}_{n'}) \prod_{\substack{n\le i\le N \\ i\ne n'}} \mathrm{d}A(\boldsymbol{x}_i) \\ &\quad + \int \Delta g_{n-1}\, h_n^{(0)}\, V_{\overline{\partial\mathcal{M}}_n}\, \mathrm{d}\ell(\boldsymbol{x}_n) \prod_{n'=n+1}^{N} \mathrm{d}A(\boldsymbol{x}_{n'}) \\ &= \int_{\mathcal{M}^{N-n+1}} \left[\left(h_{n-1}^{(0)}\right)^{\cdot} - h_{n-1}^{(0)} h_{n-1}^{(1)}\right] \prod_{n'=n}^{N} \mathrm{d}A(\boldsymbol{x}_{n'}) \\ &\quad + \sum_{n'=n}^{N} \int \Delta h_{n-1,n'}^{(0)}\, V_{\overline{\partial\mathcal{M}}_{n'}}(\boldsymbol{x}_{n'})\, \mathrm{d}\ell(\boldsymbol{x}_{n'}) \prod_{\substack{n\le i\le N \\ i\ne n'}} \mathrm{d}A(\boldsymbol{x}_i). \end{aligned} \quad (58)$$

Thus, using mathematical induction, we know that Eqs. (55) and (56) hold for all $0 \le n < N$.

Notice that $h_0^{(0)} = f$ and $\Delta h_{0,n'}^{(0)} = \Delta f_{n'}$, where $\Delta f_{n'}$ follows the definition in Eq. (28). Letting $n = 0$ in Eq. (56) yields

$$\begin{aligned} \dot{h}_0(\boldsymbol{x}_0) = &\int_{\mathcal{M}^N} \left[\dot{f}(\bar{\boldsymbol{x}}) - f(\bar{\boldsymbol{x}}) \sum_{n'=1}^{N} \kappa(\boldsymbol{x}_{n'})\, V(\boldsymbol{x}_{n'})\right] \prod_{n'=1}^{N} \mathrm{d}A(\boldsymbol{x}_{n'}) \\ &+ \sum_{n'=1}^{N} \int \Delta f_{n'}(\bar{\boldsymbol{x}})\, V_{\overline{\partial\mathcal{M}}_{n'}}\, \mathrm{d}\ell(\boldsymbol{x}_{n'}) \prod_{\substack{0<i\le N \\ i\ne n'}} \mathrm{d}A(\boldsymbol{x}_i). \end{aligned} \quad (59)$$

Lastly, based on the assumption that h_0 is continuous in $\boldsymbol{x}_0$, Eq. (25) can be obtained by differentiating Eq. (23):

$$\begin{aligned} \frac{\partial I_N}{\partial\pi} &= \frac{\partial}{\partial\pi} \int_{\mathcal{M}} h_0(\boldsymbol{x}_0)\, \mathrm{d}A(\boldsymbol{x}_0) \\ &= \int_{\mathcal{M}} \left[\dot{h}_0(\boldsymbol{x}_0) - h_0(\boldsymbol{x}_0)\, \kappa(\boldsymbol{x}_0)\, V(\boldsymbol{x}_0)\right] \mathrm{d}A(\boldsymbol{x}_0) \\ &\quad + \int_{\overline{\partial\mathcal{M}}_0} h_0(\boldsymbol{x}_0)\, V_{\overline{\partial\mathcal{M}}_0}(\boldsymbol{x}_0)\, \mathrm{d}\ell(\boldsymbol{x}_0) \\ &= \int_{\Omega_N} \left[\dot{f}(\bar{\boldsymbol{x}}) - f(\bar{\boldsymbol{x}}) \sum_{K=0}^{N} \kappa(\boldsymbol{x}_K)\, V(\boldsymbol{x}_K)\right] \mathrm{d}\mu(\bar{\boldsymbol{x}}) \\ &\quad + \sum_{K=0}^{N} \int_{\Omega_{N,K}} \Delta f_K(\bar{\boldsymbol{x}})\, V_{\overline{\partial\mathcal{M}}_K}\, \mathrm{d}\mu'_{N,K}(\bar{\boldsymbol{x}}). \end{aligned} \quad (60)$$

B DERIVATION OF CHANGE-OF-VARIABLE RATIO

In what follows, we derive the change-of-variable ratio given by Eq. (48). Given some function $\varphi : \mathcal{M}(\pi)^2 \mapsto \mathbb{R}$, consider the following integral:

$$\int_{\mathcal{M}} \int_{\Delta\mathcal{M}[\mathbb{V}]} \varphi(\boldsymbol{x}_0^\mathrm{S}, \boldsymbol{x}_0^\mathrm{D})\, \mathrm{d}\ell(\boldsymbol{x}_0^\mathrm{D})\, \mathrm{d}A(\boldsymbol{x}_0^\mathrm{S}), \quad (61)$$

where $\Delta\mathcal{M}[\mathbb{V}](\pi)$ consists of the discontinuity curves of the mutual visibility function $\mathbb{V}$ with respect to $\boldsymbol{x}_0^\mathrm{D}$ when $\boldsymbol{x}_0^\mathrm{S}$ is fixed.

When $\mathcal{M}(\pi)$ is depicted using polygonal meshes, as illustrated in Figure 9, the interior of the segment $\boldsymbol{x}_0^\mathrm{S}\,\boldsymbol{x}_0^\mathrm{D}$ will always intersect $\mathcal{M}(\pi)$ at one point $\boldsymbol{x}^\mathrm{B}$ that belongs to a face edge. Let $\mathcal{E} \subset \mathcal{M}(\pi)$ denote the union of all face edges, we can apply a change of variable from $\boldsymbol{x}_0^\mathrm{D} \in \Delta\mathcal{M}[\mathbb{V}]$ to $\boldsymbol{x}^\mathrm{B} \in \mathcal{E}$ to Eq. (61), producing

$$\int_{\mathcal{M}} \int_{\mathcal{E}} \varphi(\boldsymbol{x}_0^\mathrm{S}, \boldsymbol{x}_0^\mathrm{D})\, \frac{\|\boldsymbol{x}_0^\mathrm{D} - \boldsymbol{x}_0^\mathrm{S}\|}{\|\boldsymbol{x}^\mathrm{B} - \boldsymbol{x}_0^\mathrm{S}\|}\, \frac{\sin\theta^\mathrm{B}}{\sin\theta^\mathrm{D}}\, \mathrm{d}\ell(\boldsymbol{x}^\mathrm{B})\, \mathrm{d}A(\boldsymbol{x}_0^\mathrm{S}), \quad (62)$$

where $\boldsymbol{x}_0^\mathrm{D} = \mathrm{rayTrace}(\boldsymbol{x}^\mathrm{B}, \boldsymbol{x}_0^\mathrm{S} \to \boldsymbol{x}^\mathrm{B})$, and the added terms result from the Jacobian determinant corresponding to this change of variable. Compared to Eq. (61), Eq. (62) enjoys a key advantage that the inner integral has a domain independent of $\boldsymbol{x}_0^\mathrm{S}$. This allows us to (i) exchange the ordering of the two integrals in Eq. (62), and (ii) apply another change of variable from $\boldsymbol{x}_0^\mathrm{S} \in \mathcal{M}(\pi)$ to $\boldsymbol{\omega}^\mathrm{B} \in \mathbb{S}^2$, yielding:

$$\begin{aligned} &\int_{\mathcal{E}} \int_{\mathcal{M}} \varphi(\boldsymbol{x}_0^\mathrm{S}, \boldsymbol{x}_0^\mathrm{D})\, \frac{\|\boldsymbol{x}_0^\mathrm{D} - \boldsymbol{x}_0^\mathrm{S}\|}{\|\boldsymbol{x}^\mathrm{B} - \boldsymbol{x}_0^\mathrm{S}\|}\, \frac{\sin\theta^\mathrm{B}}{\sin\theta^\mathrm{D}}\, \mathrm{d}A(\boldsymbol{x}_0^\mathrm{S})\, \mathrm{d}\ell(\boldsymbol{x}^\mathrm{B}) \\ &= \int_{\mathcal{E}} \int_{\mathbb{S}^2} \varphi(\boldsymbol{x}_0^\mathrm{S}, \boldsymbol{x}_0^\mathrm{D}) \underbrace{\frac{\|\boldsymbol{x}_0^\mathrm{D} - \boldsymbol{x}_0^\mathrm{S}\|}{\|\boldsymbol{x}^\mathrm{B} - \boldsymbol{x}_0^\mathrm{S}\|}\, \frac{\sin\theta^\mathrm{B}}{\sin\theta^\mathrm{D}}\, \frac{\|\boldsymbol{x}^\mathrm{B} - \boldsymbol{x}_0^\mathrm{S}\|^2}{|\cos\theta^\mathrm{S}|}}_{=\text{Eq. (48)}}\, \mathrm{d}\sigma(\boldsymbol{\omega}^\mathrm{B})\, \mathrm{d}\ell(\boldsymbol{x}^\mathrm{B}), \end{aligned} \quad (63)$$

where σ is the solid-angle measure.