Warped-Area Reparameterization of Differential Path Integrals

PEIYU XU, University of California, Irvine, USA
SAI BANGARU, MIT CSAIL, USA
TZU-MAO LI, University of California, San Diego, USA
SHUANG ZHAO, University of California, Irvine, USA

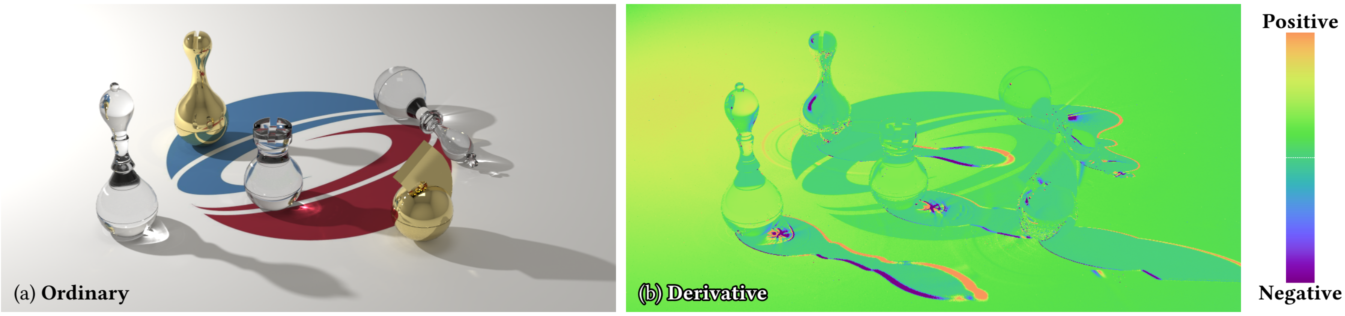


Fig. 1. We introduce the formulation of *reparameterized differential path integrals* for physics-based differentiable rendering. Our formulation can be efficiently estimated using advanced methods like bidirectional path tracing without requiring explicit sampling of discontinuity boundaries. In this example, we show several glass and metal chess pieces lit by an area light. The derivatives (obtained with our *bidirectional* estimator) are w.r.t. the position of the light.

Physics-based differentiable rendering is becoming increasingly crucial for tasks in inverse rendering and machine learning pipelines. To address discontinuities caused by geometric boundaries and occlusion, two classes of methods have been proposed: 1) the edge-sampling methods that directly sample light paths at the scene discontinuity boundaries, which require nontrivial data structures and precomputation to select the edges, and 2) the reparameterization methods that avoid discontinuity sampling but are currently limited to hemispherical integrals and unidirectional path tracing.

We introduce a new mathematical formulation that enjoys the benefits of both classes of methods. Unlike previous reparameterization work that focused on hemispherical integral, we derive the reparameterization in the path space. As a result, to estimate derivatives using our formulation, we can apply advanced Monte Carlo rendering methods, such as bidirectional path tracing, while avoiding explicit sampling of discontinuity boundaries. We show differentiable rendering and inverse rendering results to demonstrate the effectiveness of our method.

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

Additional Key Words and Phrases: Differentiable rendering, differential path integral, warped-area reparameterization

ACM Reference Format:
Peiyu Xu, Sai Bangaru, Tzu-Mao Li, and Shuang Zhao. 2023. Warped-Area Reparameterization of Differential Path Integrals. *ACM Trans. Graph.* 42, 6, Article 213 (December 2023), 18 pages. https://doi.org/10.1145/3618330

Authors' addresses: Peiyu Xu, peiyux3@uci.edu, University of California, Irvine, USA; Sai Bangaru, sbangaru@mit.edu, MIT CSAIL, USA; Tzu-Mao Li, tzli@ucsd.edu, University of California, San Diego, USA; Shuang Zhao, shz@ics.uci.edu, University of California, Irvine, USA.

0730-0301/2023/12-ART213
https://doi.org/10.1145/3618330

1 INTRODUCTION

Physics-based differentiable rendering is the task of numerically computing derivatives of radiometric measurements with respect to arbitrary scene parameters such as object shapes and optical properties. Such scene derivatives not only can enable gradient-based optimization for solving inverse rendering problems (e.g., [Azinović et al. 2019; Luan et al. 2021]), but also are a key ingredient for integrating physics-based rendering into probabilistic-inferences and machine-learning pipelines (e.g., [Che et al. 2020]).

A key challenge for developing general-purpose differentiable rendering techniques is the differentiation with respect to scene geometries (such as the pose of an object or the position of a mesh vertex). This is because such geometries affect visibility and, if not handled properly, can lead to severely biased derivative estimates—which has been demonstrated by many prior works (e.g., [Li et al. 2018; Loubet et al. 2019; Zhang et al. 2019]).

To address this problem, two categories of techniques have been introduced. The first category *directly samples discontinuity boundaries* [Li et al. 2018; Zhang et al. 2019, 2020, 2021b], and the state of the art is Zhang et al.'s [2020] *differential path integral* formulation which tracks and handles discontinuities at the path level. The second category, on the other hand, *reparameterizes rendering integrals* to avoid explicit handling of discontinuities altogether [Loubet et al. 2019; Bangaru et al. 2020], with the state of the art being Bangaru et al.'s [2020] *warped-area reparameterization.*

In practice, Zhang et al.'s differential path integrals offer the flexibility to develop advanced Monte Carlo estimators, such as bidirectional path tracing, but require nontrivial data structures and precomputation [Yan et al. 2022] to efficiently sample discontinuity boundaries. Bangaru et al.'s warped-area reparameterization, on the other hand, enjoys the convenience of not needing explicit tracking of discontinuities but has been limited to hemispherical integrals and unidirectional path tracing.

In this paper, we introduce a new formulation that enjoys the benefits of both these methods. Specifically, we adopt Bangaru et al.'s [2020] technique to reparameterize Zhang et al.'s [2020] differential path integrals. Our resulting formulation can be estimated using advanced methods (e.g., bidirectional path tracing) without explicit sampling of discontinuity boundaries.

Concretely, our contributions include:

- Introducing a new formulation for *differential one-bounce light transport* by applying Bangaru et al.'s [2020] warped-area method on top of Zhang et al.'s [2020] material-form reparameterization (§3).
- Establishing the formulation of *reparameterized differential path integrals* by generalizing the one-bounce result in §3 to handle full path integrals (§4).
- Developing Monte Carlo estimators leveraging unidirectional and bidirectional path sampling methods for our reparameterized differential path integrals (§5).

We validate our technique by comparing derivative estimates obtained using finite difference (FD) and our methods with high sample counts (Figure 11). Further, we show differentiable rendering (Figures 12, 13, 15 and 18) and inverse rendering (Figures 12–17) results to demonstrate the effectiveness of our technique.

2 RELATED WORK

Recently, great progresses have been made in the field of physics-based differentiable rendering. In what follows, we discuss most relevant techniques. For a more comprehensive overview, please refer to online course materials, e.g., [Zhao et al. 2020].

Handling discontinuities. A main challenge toward 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, two classes of techniques have been introduced. The first class directly samples discontinuity boundaries. Specifically, Li et al. [2018] introduced Monte Carlo edge sampling—the first technique in this class—by differentiating Kajiya's [1986] rendering equation. Zhang et al. [2019] later generalized this technique to differentiate the radiative transfer equation [Chandrasekhar 1960] for volumetric light transport. Further, Zhang et al. [2020; 2021b] have recently introduced the formulation of *differential path integrals* where discontinuities are tracked and handled at the path level, leading to significantly better performance.

The second class of methods reparameterize boundary integrals and avoid explicit handling of discontinuity boundaries altogether. The first approach of this category was biased and introduced by Loubet et al. [2019]. This method was later extended by Bangaru et al.'s [2020] *warped-area reparameterization*—which is capable of offering unbiased and consistent derivative estimates.

In §3, we will derive a new formulation by applying Bangaru et al.'s warped-area technique on top of Zhang et al.'s material-form reparameterization to differentiate one-bounce light transport. This result will then be generalized in §4 to produce *reparameterized differential path integrals*—the main result of this paper.

Efficient sampling and differentiation. Previously, differentiable rendering algorithms have been largely repurposing Monte Carlo methods developed for forward rendering to sample light paths. Recently, several sampling methods have been developed specifically for differentiable rendering [Zeltner et al. 2021; Zhang et al. 2021a; Nimier-David et al. 2022; Yan et al. 2022; Yu et al. 2022]. Our technique is largely complementary to these methods, although we consider developing new Monte Carlo techniques specific to our formulation an important topic for future research.

Additionally, several approaches have been proposed to allow differentiable renderers scaling out to complex scenes with large numbers of parameters [Nimier-David et al. 2020; Vicini et al. 2021]. Our mathematical formulation is largely orthogonal to these techniques. Practically, our *unidirectional* estimator is compatible with these techniques. On the other hand, how these methods can be used with bidirectional path-sampling techniques (including our *bidirectional* estimator) remains an open problem.

3 DIFFERENTIATING ONE-BOUNCE LIGHT TRANSPORT WITH EVOLVING EMITTERS

We introduce in this section a simple form of our technique that differentiates *one-bounce light transport with evolving area lights.*

Specifically, we revisit the *material-form* reparameterization introduced by Zhang et al. [2020] in §3.1 before introducing our method that further reparameterizes the material-form result using the *warped-area* technique developed by Bangaru et al. [2020] in §3.2.

Problem specification. Under one-bounce light transport (aka. *direct illumination*), the scattered (i.e., reflected or refracted) radiance leaving a surface point $\boldsymbol{y}$ toward some point $\boldsymbol{y}'$ can be expressed as an integral over the surface $\mathcal{M}_\mathrm{e}$ of all emitters:

$$I_\mathrm{dir}(\boldsymbol{y}\to\boldsymbol{y}') = \int_{\mathcal{M}_\mathrm{e}} L_\mathrm{e}(\boldsymbol{x}\to\boldsymbol{y})\, f_\mathrm{s}(\boldsymbol{x}\to\boldsymbol{y}\to\boldsymbol{y}')\, G(\boldsymbol{x}\leftrightarrow\boldsymbol{y})\, \mathrm{d}A(\boldsymbol{x}), \quad (1)$$

where L_e denotes the **source emission**, f_s indicates the **bidirectional scattering distribution function** (BSDF), G is the (visibility-aware) **geometric term**, and A is the area measure.

In what follows, we focus on the configuration where the emitter surface $\mathcal{M}_\mathrm{e}$ evolves with some parameter $\theta \in \mathbb{R}$ while all other surfaces—as well as $\boldsymbol{y}$ and $\boldsymbol{y}'$ on the right-hand side of Eq. (1)—are static (i.e., independent of θ). Under this setting, we consider the problem of differentiating the outgoing radiance I_dir with respect to θ. We summarize commonly used symbols and their definitions in Table 1.

3.1 Preliminary: Material-Form Reparameterization

To facilitate the differentiation of I_dir, Zhang et al. [2020] propose to rewrite the integral as one over some fixed **reference surface** $\mathcal{B}_\mathrm{e}$. This can be achieved using a differentiable one-to-one mapping (aka. a **deformation**) $\mathsf{X}(\cdot,\theta)$ that transforms the fixed reference surface $\mathcal{B}_\mathrm{e}$ to the evolving one $\mathcal{M}_\mathrm{e}(\theta)$. To distinguish points on the fixed $\mathcal{B}_\mathrm{e}$ and the evolving $\mathcal{M}_\mathrm{e}(\theta)$, we call any $\boldsymbol{p} \in \mathcal{B}_\mathrm{e}$ a **material point** and $\boldsymbol{x} \in \mathcal{M}_\mathrm{e}(\theta)$ a **spatial point**.

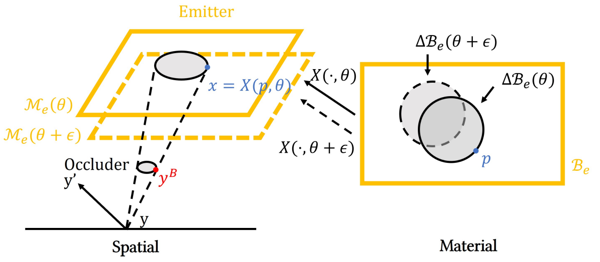

Emitter
$\mathcal{M}_e(\theta)$
$\mathcal{M}_e(\theta+\epsilon)$
$x = X(p,\theta)$
$X(\cdot,\theta)$
$X(\cdot,\theta+\epsilon)$
Occluder
y'
y^B
y
Spatial
$\Delta\mathcal{B}_e(\theta+\epsilon)$
$\Delta\mathcal{B}_e(\theta)$
p
$\mathcal{B}_e$
Material

Fig. 2. **Material-form reparameterization:** This example involves a simple configuration with a fixed occluder and an emitter over an evolving surface $\mathcal{M}_e(\theta)$. Leveraging Zhang et al.'s [2020] material-form reparameterization, one can use a constant reference surface $\mathcal{B}_e$ coupled with a differentiable motion X such that, for any θ, $\mathsf{X}(\cdot,\theta)$ is a differentiable one-to-one mapping (i.e., a deformation) from $\mathcal{B}_e$ to $\mathcal{M}_e(\theta)$. We note that, although the reference surface $\mathcal{B}_e$ is fixed, the discontinuity curves $\Delta\mathcal{B}_e$ capturing visibility boundaries with respect to the surface point $\boldsymbol{y}$ generally depends on the parameter θ.

Applying a change of variable $\boldsymbol{x} := \mathsf{X}(\boldsymbol{p},\theta)$ to the original surface integral (1) produces a *material-form*[1] variant:

$$I_{\mathrm{dir}} = \int_{\mathcal{B}_e} \underbrace{L_e(\boldsymbol{x}\to\boldsymbol{y})\, f_s(\boldsymbol{x}\to\boldsymbol{y}\to\boldsymbol{y}')\, G(\boldsymbol{x}\leftrightarrow\boldsymbol{y})\, J(\boldsymbol{p},\theta)}_{=:\,F_{\mathrm{dir}}(\boldsymbol{p})}\, \mathrm{d}A(\boldsymbol{p}), \tag{2}$$

where

$$J(\boldsymbol{p},\theta) := \left\| \frac{\mathrm{d}A(\boldsymbol{x})}{\mathrm{d}A(\boldsymbol{p})} \right\|, \tag{3}$$

is the Jacobian determinant resulting from the change of variable. In Eqs. (2) and (3), we consider $\boldsymbol{x} = \mathsf{X}(\boldsymbol{p},\theta)$ a function of the material point $\boldsymbol{p} \in \mathcal{B}_e$ and the parameter θ. Also, we omit the dependency of the integrand F_{dir} on the parameter θ for notational simplicity.

Fundamentally, the material-form reparameterization has moved the derivative contribution from the evolution of the domain $\mathcal{M}_e$ into the new integrand F_{dir}—whose derivative is easier to compute due to fewer types of discontinuities that requires handling.

Choice of reference surfaces. When estimating derivatives at $\theta = \theta_0$ (for some fixed θ_0), the reference surface is usually set to $\mathcal{B}_e = \mathcal{M}_e(\theta_0)$. Under this configuration, the mapping $\mathsf{X}(\cdot,\theta_0)$ becomes the identity map, causing the corresponding Jacobian determinant $J(\boldsymbol{p},\theta_0)$ to reduce to one. We note that, the derivative of $J(\boldsymbol{p},\theta)$ with respect to θ is generally nonzero—even at $\theta = \theta_0$.

Differential direct-illumination integral. In general, differentiating Eq. (2) using Reynolds transport theorem [1903; 1973] produces *material-form differential integrals*:

$$\frac{\mathrm{d}I_{\mathrm{dir}}}{\mathrm{d}\theta} = \overset{\textbf{interior}}{\boxed{\int_{\mathcal{B}_e} \frac{\mathrm{d}F_{\mathrm{dir}}(\boldsymbol{p})}{\mathrm{d}\theta}\, \mathrm{d}A(\boldsymbol{p})}} + \overset{\textbf{boundary}}{\boxed{\int_{\Delta\mathcal{B}_e} \Delta F_{\mathrm{dir}}(\boldsymbol{p})\, V(\boldsymbol{p})\, \mathrm{d}\ell(\boldsymbol{p})}}, \tag{4}$$

where ℓ denotes the curve-length measure, and the *interior* component is a surface integral identical to Eq. (1) except for having the integrand F_{dir} differentiated. The *boundary* component in Eq. (4)—which we will define and discuss in §3.1.1 and §3.1.2 below—is a line integral that is unique to differentiable rendering.

Table 1. Commonly used symbols in §3. The right-most column indicates θ-dependency.

Symbol	**Definition**	**θ-dep.**
$\mathcal{M}_e$	Emitter surface evolving with some parameter θ	Yes
$\mathcal{B}_e$	Reference surface	**No**
$\mathsf{X}(\cdot,\theta)$	Differentiable one-to-one mapping (aka. deformation) from $\mathcal{B}_e$ to $\mathcal{M}_e(\theta)$	Yes
I_{dir}	Radiance $L(\boldsymbol{y}\to\boldsymbol{y}')$ with fixed $\boldsymbol{y}$ and $\boldsymbol{y}'$	Yes
F_{dir}	Integrand of the material-form integral (2)	Yes
$\Delta\mathcal{B}_e$	Curves comprising jump discontinuities of F_{dir}	Yes
$\mathcal{B}_e^{\mathrm{wa}}$	Regions on the reference surface $\mathcal{B}_e$ comprising points $\boldsymbol{p}$ such that $\mathsf{X}(\boldsymbol{p},\theta)$ is visible to $\boldsymbol{y}$	Yes
$\partial\mathcal{B}_e^{\mathrm{wa}}$	Boundary of $\mathcal{B}_e^{\mathrm{wa}}$ (and a superset of $\Delta\mathcal{B}_e$)	Yes
$\boldsymbol{n}^{\partial}$	Unit-normal field associated with $\Delta\mathcal{B}_e$	Yes
$\boldsymbol{v}^{\partial}$	Velocity field over $\Delta\mathcal{B}_e$ (and later extended to $\partial\mathcal{B}_e^{\mathrm{wa}}$)	Yes
$\boldsymbol{v}^{\mathrm{dis}}$	Discontinuous velocity field over $\mathcal{B}_e$	Yes
$\boldsymbol{v}$	Continuous velocity field obtained by convolving $\boldsymbol{v}^{\mathrm{dis}}$	Yes
w	Spatially varying kernel for convolving $\boldsymbol{v}^{\mathrm{dis}}$	**No**

3.1.1 *Discontinuities.*

The *boundary* integral in Eq. (4) is over a set of curves $\Delta\mathcal{B}_e$ comprising jump-discontinuity points of the integrand F_{dir} (with respect to $\boldsymbol{p}$). We note that, although the reference surface $\mathcal{B}_e$ is fixed, the discontinuity curves $\Delta\mathcal{B}_e$ typically depend on the parameter θ. Also, thanks to the material-form reparameterization, the discontinuity curves do not contain the topological boundary $\partial\mathcal{B}_e$ of the reference surface $\mathcal{B}_e$ (when $\mathcal{B}_e$ is open) since $\partial\mathcal{B}_e$ does not depend on the parameter θ. Additionally, for any discontinuity point $\boldsymbol{p} \in \Delta\mathcal{B}_e$, ΔF_{dir} captures the difference in F_{dir} across the boundary.

In this paper, we assume the emitted radiance L_e and BSDF f_s to be C^0-continuous. Under this assumption, the discontinuities emerge solely from the mutual visibility between $\boldsymbol{x}$ and $\boldsymbol{y}$—a factor of the geometric term G in Eqs. (1) and (2). Precisely, for any $\theta \in \mathbb{R}$, the discontinuity curves $\Delta\mathcal{B}_e$ consist of material points $\boldsymbol{p}$ whose spatial representations $\boldsymbol{x} = \mathsf{X}(\boldsymbol{p},\theta)$ are jump discontinuities of the mutual visibility $\mathbb{V}(\boldsymbol{x}\leftrightarrow\boldsymbol{y})$ (with $\boldsymbol{y}$ fixed), as illustrated in Figure 2. Further, we assume without loss of generality that the curve normal $\boldsymbol{n}^{\partial}(\boldsymbol{p})$ points toward the occluded side for all $\boldsymbol{p}$ (see Figure 3). Then, it holds that

$$\Delta F_{\mathrm{dir}}(\boldsymbol{p}) = F_{\mathrm{dir}}(\boldsymbol{p}), \tag{5}$$

where ΔF_{dir} denotes the difference in F_{dir} across a discontinuties boundary

3.1.2 *Scalar normal velocity.*

In Eq. (4), another key component of the *boundary* integral is the **scalar normal velocity** V capturing how fast the discontinuity boundary $\Delta\mathcal{B}_e$ evolves (with respect to the parameter θ) along the curve normal $\boldsymbol{n}^{\partial}$. Precisely, this term is defined as

$$V(\boldsymbol{p}) = \boldsymbol{n}^{\partial}(\boldsymbol{p}) \cdot \boldsymbol{v}^{\partial}(\boldsymbol{p}), \tag{6}$$

[1] The term "material form" originates in continuum mechanics where reparameterizations like Eq. (2) are common.

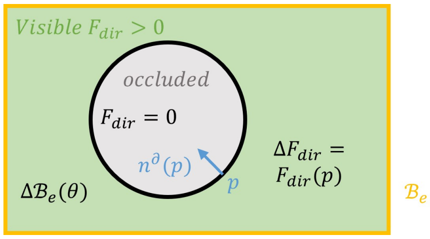


Fig. 3. In this paper, we assume without loss of generality that, at any point $\boldsymbol{p}$ on the discontinuity boundary $\Delta\mathcal{B}_\mathrm{e}$, the curve normal $\boldsymbol{n}^\partial(\boldsymbol{p})$ always points toward the occluded side of the boundary. This implies that $\Delta F_\mathrm{dir}(\boldsymbol{p}) = F_\mathrm{dir}(\boldsymbol{p})$.

where $\boldsymbol{v}^\partial(\boldsymbol{p})$ is the (vector-valued) **boundary velocity** indicating the change rate (with respect to θ) of the discontinuity point $\boldsymbol{p}$.

In practice, according to Zhang et al. [2020], $\boldsymbol{v}^\partial(\boldsymbol{p})$ can be computed as follows. For any discontinuity point $\boldsymbol{p} \in \Delta\mathcal{B}_\mathrm{e}$, the (spatial) line segment $\overline{\boldsymbol{yx}}$ with $\boldsymbol{x} = \mathsf{X}(\boldsymbol{p}, \theta)$ must intersect some occluder at one point $\boldsymbol{y}^\mathrm{B}$ (see the red point in Figure 2). Let

$$\boldsymbol{z} = \mathsf{rayIntersect}(\boldsymbol{y} \to \boldsymbol{y}^\mathrm{B},\ \mathcal{M}_\mathrm{e}) \tag{7}$$

be the intersection between the ray $\boldsymbol{y} \to \boldsymbol{y}^\mathrm{B}$ and the emitter surface $\mathcal{M}_\mathrm{e}$. Although $\boldsymbol{z}$ and $\boldsymbol{x}$ are both points on the emitter, their derivatives are different: $\mathrm{d}\boldsymbol{z}/\mathrm{d}\theta$ is given by differentiating the ray-intersection computation in Eq. (7); $\mathrm{d}\boldsymbol{x}/\mathrm{d}\theta$, on the other hand, is obtained by differentiating the deformation $\mathsf{X}(\boldsymbol{p}, \theta)$.

Lastly, with the intersection point $\boldsymbol{z}$ computed, we define the boundary velocity at $\boldsymbol{p}$ as

$$\boldsymbol{v}^\partial(\boldsymbol{p}) = \frac{\mathrm{d}}{\mathrm{d}\theta}\mathsf{X}^{-1}(\boldsymbol{z}, \theta), \tag{8}$$

where $\mathsf{X}^{-1}(\cdot, \theta)$ is the inverse of the mapping $\mathsf{X}(\cdot, \theta)$ and transforms the emitter surface $\mathcal{M}_\mathrm{e}(\theta)$ back to the reference $\mathcal{B}_\mathrm{e}$.

3.2 Warped-Area Reparameterization of Boundary Line Integrals

Bangaru et al. [2020] have proposed to rewrite *boundary* integrals as *interior* ones using the divergence theorem. Let Ω be a closed 2D domain and $\partial\Omega$ be the domain boundary associated with outward normal field $\boldsymbol{n}^\partial$. Then, the divergence theorem states that, for any continuous vector-valued function $\boldsymbol{F}$, it holds that

$$\int_{\partial\Omega} \boldsymbol{F} \cdot \boldsymbol{n}^\partial \,\mathrm{d}\ell = \int_\Omega \nabla \cdot \boldsymbol{F} \,\mathrm{d}A. \tag{9}$$

By letting $\boldsymbol{F}(\boldsymbol{p}) \coloneqq F_\mathrm{dir}(\boldsymbol{p})\, \boldsymbol{v}^\partial(\boldsymbol{p})$ in Eq. (9), we aim to reparameterize the *boundary* component of Eq. (4) via

$$\overset{\textbf{boundary}}{\boxed{\int_{\Delta\mathcal{B}_\mathrm{e}} \left(F_\mathrm{dir}\, \boldsymbol{v}^\partial\right) \cdot \boldsymbol{n}^\partial \,\mathrm{d}\ell}} = \overset{\textbf{interior}}{\boxed{\int_{\mathcal{B}_\mathrm{e}^\mathrm{wa}} \nabla \cdot (F_\mathrm{dir}\, \boldsymbol{v}) \,\mathrm{d}A}}, \tag{10}$$

where $\mathcal{B}_\mathrm{e}^\mathrm{wa}$ is some 2D region within the reference surface $\mathcal{B}_\mathrm{e}$ (see Figure 5). To realize this reparameterization, we still need to overcome the following obstacles:

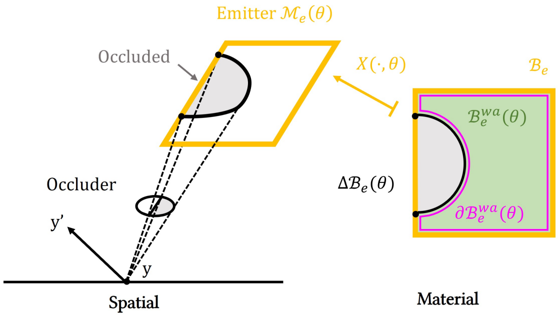

Emitter $\mathcal{M}_e(\theta)$
Occluded
$X(\cdot, \theta)$
$\mathcal{B}_e$
$\mathcal{B}_e^{wa}(\theta)$
Occluder
$\Delta\mathcal{B}_e(\theta)$
$\partial\mathcal{B}_e^{wa}(\theta)$
y'
y
Spatial
Material

Fig. 4. **Open curves:** The discontinuity curves $\Delta\mathcal{B}_\mathrm{e}$ may be open, making it nontrivial to apply the divergence theorem in Eq. (9). This can happen, as illustrated in this example, when the visibility boundaries are clipped by the edge of the emitter surface. To address this problem, we define $\mathcal{B}_\mathrm{e}^\mathrm{wa} \subseteq \mathcal{B}_\mathrm{e}$ (the green region on the right) as the material representation of the spatial region visible to $\boldsymbol{y}$ (the green region on the left). Then, the boundary $\partial\mathcal{B}_\mathrm{e}^\mathrm{wa}$ of $\mathcal{B}_\mathrm{e}^\mathrm{wa}$ (illustrated in magenta) is a superset of the discontinuity curves $\Delta\mathcal{B}_\mathrm{e}$.

- The discontinuity curves $\Delta\mathcal{B}_\mathrm{e}$ may not be closed (see Figure 4), making it nontrivial to define the domain $\mathcal{B}_\mathrm{e}^\mathrm{wa}$ of the reparameterized *interior* integral.
- The normal velocity $\boldsymbol{v}^\partial$ is defined on the discontinuity boundaries $\Delta\mathcal{B}_\mathrm{e}$ only and needs to be extended *continuously* to some vector field $\boldsymbol{v}$ over the interior of the domain $\mathcal{B}_\mathrm{e}^\mathrm{wa}$.

To address these problems, we adopt Bangaru et al.'s [2020] scheme by introducing a two-stage process:

S.1 Extending the discontinuity curves $\Delta\mathcal{B}_\mathrm{e}$ to a set of closed curves $\partial\mathcal{B}_\mathrm{e}^\mathrm{wa} \supset \Delta\mathcal{B}_\mathrm{e}$ with a well-defined interior $\mathcal{B}_\mathrm{e}^\mathrm{wa}$. This will allow us to treat the left-hand side of Eq. (10) as an integral over $\partial\mathcal{B}_\mathrm{e}^\mathrm{wa}$ (where the integrand is set to zero for all points $\boldsymbol{p} \in \partial\mathcal{B}_\mathrm{e}^\mathrm{wa} \setminus \Delta\mathcal{B}_\mathrm{e}$).

S.2 Smoothly interpolating $\boldsymbol{v}^\partial$ from the boundary $\partial\mathcal{B}_\mathrm{e}^\mathrm{wa}$ to the interior $\mathcal{B}_\mathrm{e}^\mathrm{wa}$, making the right-hand side of Eq. (10) well defined.

After establishing Eq. (10), we will be able to rewrite the full material-form differential integral (4) as one *interior* integral:

$$\frac{\mathrm{d}I_\mathrm{dir}}{\mathrm{d}\theta} = \overset{\textbf{interior}}{\boxed{\int_{\mathcal{B}_\mathrm{e}} \left(\frac{\mathrm{d}F_\mathrm{dir}(\boldsymbol{p})}{\mathrm{d}\theta} + [\nabla \cdot (F_\mathrm{dir}\, \boldsymbol{v})](\boldsymbol{p}) \right) \mathrm{d}A(\boldsymbol{p})}}. \tag{11}$$

In the following, we detail Stages **S.1** and **S.2**, respectively, in §3.2.1 and §3.2.2 before completing the derivation of Eq. (11) in §3.2.3.

3.2.1 Extending discontinuity boundaries. As presented in §3.1.1, the discontinuity curves $\Delta\mathcal{B}_\mathrm{e}$ emerge from visibility boundaries of $\boldsymbol{x}$ when $\boldsymbol{y}$ is fixed. Let $\mathcal{B}_\mathrm{e}^\mathrm{wa}$ be a subset of the reference surface $\mathcal{B}_\mathrm{e}$ containing all material points $\boldsymbol{p}$ whose spatial representations $\boldsymbol{x} = \mathsf{X}(\boldsymbol{p}, \theta)$ are visible to $\boldsymbol{y}$. That is,

$$\mathcal{B}_\mathrm{e}^\mathrm{wa} \coloneqq \{\boldsymbol{p} \in \mathcal{B}_\mathrm{e} \;:\; \mathbb{V}(\mathsf{X}(\boldsymbol{p}, \theta) \leftrightarrow \boldsymbol{y}) = 1\}. \tag{12}$$

Based on this definition, as illustrated in Figures 4 and 5, the boundary $\partial\mathcal{B}_\mathrm{e}^\mathrm{wa}$ of $\mathcal{B}_\mathrm{e}^\mathrm{wa}$ consists of material points $\boldsymbol{p}$ whose spatial

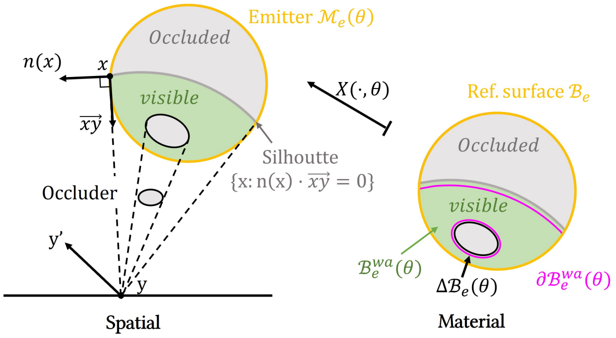

Emitter $\mathcal{M}_e(\theta)$
Occluded
$n(x)$
x
visible
$\overrightarrow{xy}$
$X(\cdot,\theta)$
Ref. surface $\mathcal{B}_e$
Silhoutte
$\{x\colon n(x)\cdot\overrightarrow{xy}=0\}$
Occluder
y'
y
Occluded
visible
$\mathcal{B}_e^{wa}(\theta)$
$\Delta\mathcal{B}_e(\theta)$
$\partial\mathcal{B}_e^{wa}(\theta)$
Spatial
Material

Fig. 5. We define $\mathcal{B}_\mathrm{e}^\mathrm{wa} \subset \mathcal{B}_\mathrm{e}$ as the set of all material points $\boldsymbol{p}$ whose spatial representations $\boldsymbol{x} = \mathsf{X}(\boldsymbol{p}, \theta)$ are visible to $\boldsymbol{y}$. This example shows a spherical emitter with no topological boundary. In this case, the boundary $\partial\mathcal{B}_\mathrm{e}^\mathrm{wa}$ (illustrated in magenta) equals the union of visibility boundary $\Delta\mathcal{B}_\mathrm{e}$ (illustrated in black) caused by the small occluder and the silhouette (illustrated in gray) of the sphere with respect to $\boldsymbol{y}$.

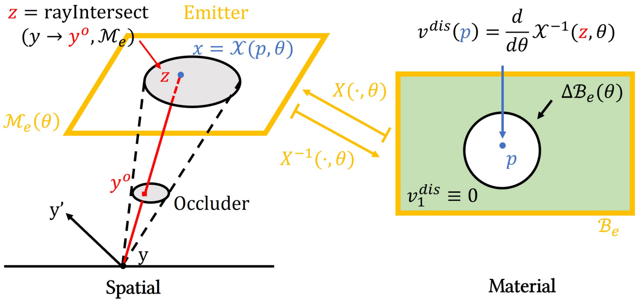


Fig. 6. **Discontinuous velocity $\boldsymbol{v}^\mathrm{dis}$:** For each material point $\boldsymbol{p} \in \mathcal{B}_\mathrm{e}$, if its spatial representation $\boldsymbol{x} = \mathsf{X}(\boldsymbol{p}, \theta)$ is visible to $\boldsymbol{y}$, we set $\boldsymbol{v}^\mathrm{dis}(\boldsymbol{p}) = \boldsymbol{0}$. Otherwise, there must exist an intersection $\boldsymbol{y}^\mathrm{O}$ (illustrated as the red point) between the open line segment $\overline{\boldsymbol{yx}}$ and some occluder. We compute $\boldsymbol{z} = \mathrm{rayIntersect}(\boldsymbol{y} \to \boldsymbol{y}^\mathrm{O}, \mathcal{M}_\mathrm{e})$ and set the velocity $\boldsymbol{v}^\mathrm{dis}(\boldsymbol{p}) = \frac{\mathrm{d}}{\mathrm{d}\theta}\mathsf{X}^{-1}(\boldsymbol{z}, \theta)$.

representations $\boldsymbol{x} = \mathsf{X}(\boldsymbol{p}, \theta)$ belong to one of the following categories:

C.1 Topological boundary of the emitter surface $\mathcal{M}_\mathrm{e}(\theta)$;

C.2 Silhouette of the emitter (with respect to $\boldsymbol{y}$) comprising spatial points $\boldsymbol{x}$ satisfying $\boldsymbol{n}(\boldsymbol{x}) \cdot \overrightarrow{\boldsymbol{xy}} = 0$;

C.3 Visibility boundaries resulting from occlusion.

It follows that the discontinuity boundary $\Delta\mathcal{B}_\mathrm{e}$—which corresponds only points from the last category (**C.3**)—is a subset of $\partial\mathcal{B}_\mathrm{e}^\mathrm{wa}$. This allows us to rewrite the left-hand side of Eq. (10) as

$$\int_{\partial\mathcal{B}_\mathrm{e}^\mathrm{wa}} \left(F_\mathrm{dir}(\boldsymbol{p})\, \boldsymbol{v}^\partial(\boldsymbol{p})\right) \cdot \boldsymbol{n}^\partial(\boldsymbol{p})\, \mathrm{d}\ell(\boldsymbol{p}), \tag{13}$$

where the normal velocity $\boldsymbol{v}^\partial(\boldsymbol{p})$ is set to zero for all $\boldsymbol{p}$ beyond the ordinary discontinuity boundaries $\Delta\mathcal{B}_\mathrm{e}$.

3.2.2 Interpolating $\boldsymbol{v}^\partial$. Our objective of this stage is to specify a continuous velocity field $\boldsymbol{v}$ in the interior of the domain $\mathcal{B}_\mathrm{e}^\mathrm{wa}$ such that $\boldsymbol{v}$ agrees with $\boldsymbol{v}^\partial$ on the domain boundary $\partial\mathcal{B}_\mathrm{e}^\mathrm{wa}$. To this end, we adopt Bangaru et al.'s [2020] warped-area approach and employ a two-step process: we first define a discontinuous velocity field $\boldsymbol{v}^\mathrm{dis}$ and then smooth it to make it continuous. In the following, we detail each of the two steps.

Step 1. In the first step, we define a discontinuous velocity field $\boldsymbol{v}^\mathrm{dis}$ over the entire reference surface $\mathcal{B}_\mathrm{e}$ (see Figure 6).

For any material point $\boldsymbol{p}$, when its spatial representation $\boldsymbol{x} = \mathsf{X}(\boldsymbol{p}, \theta)$ is invisible to $\boldsymbol{y}$ or resides on a visibility boundary (i.e., $\boldsymbol{p} \notin \mathcal{B}_\mathrm{e}^\mathrm{wa}$), we define $\boldsymbol{v}^\mathrm{dis}(\boldsymbol{p})$ in a similar fashion as the boundary velocity $\boldsymbol{v}^\partial(\boldsymbol{p})$ described in §3.1.2. Precisely, let $\boldsymbol{y}^\mathrm{O}$ be an intersection of the open line segment $\overline{\boldsymbol{yx}}$ and the scene geometry. Then, by substituting $\boldsymbol{y}^\mathrm{B}$ with $\boldsymbol{y}^\mathrm{O}$ in Eq. (7), we define $\boldsymbol{v}^\mathrm{dis}(\boldsymbol{p})$ using Eq. (8). Based on this construction, it is easy to verify that $\boldsymbol{v}^\mathrm{dis}(\boldsymbol{p}) = \boldsymbol{v}^\partial(\boldsymbol{p})$ for any point $\boldsymbol{p}$ on the boundary $\partial\mathcal{B}_\mathrm{e}^\mathrm{wa}$.

On the other hand, when $\boldsymbol{x}$ is visible to $\boldsymbol{y}$ (i.e., $\boldsymbol{p} \in \mathcal{B}_\mathrm{e}^\mathrm{wa}$), there is no occluder between $\boldsymbol{x}$ and $\boldsymbol{y}$. In this case, we simply set $\boldsymbol{v}^\mathrm{dis}(\boldsymbol{p}) = \boldsymbol{0}$.

Step 2. With the discontinuous velocity field $\boldsymbol{v}^\mathrm{dis}$ defined, we construct the continuous velocity field $\boldsymbol{v}$ by smoothing $\boldsymbol{v}^\mathrm{dis}$. Specifically, for any $\boldsymbol{p}$, we set

$$\boldsymbol{v}(\boldsymbol{p}) := \frac{\int_{\mathcal{B}_\mathrm{e}} \mathrm{w}(\boldsymbol{q}; \boldsymbol{p})\, \boldsymbol{v}^\mathrm{dis}(\boldsymbol{q})\, \mathrm{d}A(\boldsymbol{q})}{\int_{\mathcal{B}_\mathrm{e}} \mathrm{w}(\boldsymbol{q}; \boldsymbol{p})\, \mathrm{d}A(\boldsymbol{q})}, \tag{14}$$

where w is a **spatially varying kernel**.

To ensure that the resulting $\boldsymbol{v}$ agrees with $\boldsymbol{v}^\mathrm{dis}$ (and, in turn, $\boldsymbol{v}^\partial$) at the boundary $\partial\mathcal{B}_\mathrm{e}^\mathrm{wa}$, we need $\mathrm{w}(\cdot, \boldsymbol{p})$ to behave like *Dirac delta distributions* for all $\boldsymbol{p} \in \partial\mathcal{B}_\mathrm{e}^\mathrm{wa}$ on the boundary, while having a smooth falloff when $\boldsymbol{p}$ moves away from the boundary. We follow Bangaru et al. [2020] and use:

$$\mathrm{w}(\boldsymbol{q}; \boldsymbol{p}) := (\mathrm{D}(\boldsymbol{q}; \boldsymbol{p}) + \mathrm{B}(\boldsymbol{q}))^{-1}, \tag{15}$$

where $\mathrm{D}(\boldsymbol{q}; \boldsymbol{p})$ is the **distance function** that must approach zero when $\boldsymbol{q}$ approaches $\boldsymbol{p}$, and $\mathrm{B}(\boldsymbol{q})$ is the **boundary-test function** that should approach zero when $\boldsymbol{q}$ approach the boundary $\partial\mathcal{B}_\mathrm{e}^\mathrm{wa}$. We note that, since the kernel w is used solely for smoothing $\boldsymbol{v}^\mathrm{dis}$, we consider w independent of the parameter θ (i.e., "detached").

In the following, we express our choices of the distance and the boundary-test functions.

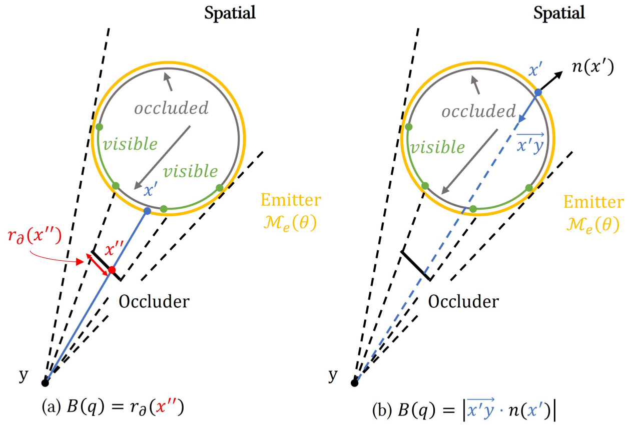
Distance function. Bangaru et al. [2020] have introduced a distance function, but we found it non-robust—which we will demonstrate in §6.2. To address this problem, we choose a different distance function by letting

$$\mathrm{D}(\boldsymbol{q}; \boldsymbol{p}) := \left(\frac{1}{\sigma}\left(1 - \exp\left(-\frac{r^2}{\sigma}\right)\right)\right)^a, \tag{16}$$

where

$$r := \|\boldsymbol{q} - \boldsymbol{p}\|/r_0, \quad \sigma := \sigma_0/r_0, \tag{17}$$

with $r_0 := \|\boldsymbol{x} - \boldsymbol{y}\|$ being a normalization factor that makes r and σ invariant of global scaling of the scene. In Eqs. (16) and (17), a and σ_0 are user-specified hyperparameters—for which we use $a = 3$ and $\sigma_0 = 0.006$ in practice which work well in all our experiments.



(a) $B(q) = r_\partial(x'')$

(b) $B(q) = |\overrightarrow{x'y} \cdot n(x')|$

Fig. 7. **Boundary-test function** B: For any material point $\boldsymbol{q}$, we set $\mathrm{B}(\boldsymbol{q})$ based on its spatial representation $\boldsymbol{x}' = \mathsf{X}(\boldsymbol{q}, \theta)$. In (a), we set $\mathrm{B}(\boldsymbol{q})$ to $r^\partial(\boldsymbol{x}'')$—the shortest distance from $\boldsymbol{x}''$ to the edge of the occluder. In (b), $\boldsymbol{x}'$ resides on the emitter surface that has no topological boundary. In this case, we set $\mathrm{B}(\boldsymbol{q})$ to $\left|\overrightarrow{\boldsymbol{x}'\boldsymbol{y}} \cdot \boldsymbol{n}(\boldsymbol{x}')\right|$.

Boundary-test function. The last remaining ingredient for completing the kernel (15) and, in turn, the convolution in Eq. (14) is the boundary-test function B. For all $\boldsymbol{q}$ on the boundary $\partial\mathcal{B}_\mathrm{e}^\mathrm{wa}$, we set $\mathrm{B}(\boldsymbol{q}) = 0$. For each $\boldsymbol{q} \in \mathcal{B}_\mathrm{e}^\mathrm{wa}$, we have $\boldsymbol{v}^\mathrm{dis}(\boldsymbol{q}) \equiv \boldsymbol{0}$ (as discussed in Step 1 above), and the value of $\mathrm{w}(\boldsymbol{q}; \boldsymbol{p})$ does not matter (as long as it is finite). Thus, we set $\mathrm{B}(\boldsymbol{q}) = 1$.

We now focus on the last case where $\boldsymbol{q} \in \mathcal{B}_\mathrm{e} \setminus (\mathcal{B}_\mathrm{e}^\mathrm{wa} \cup \partial\mathcal{B}_\mathrm{e}^\mathrm{wa})$. According to the definition of $\mathcal{B}_\mathrm{e}^\mathrm{wa}$ in Eq. (12), this implies that the spatial representation $\boldsymbol{x}' = \mathsf{X}(\boldsymbol{q}, \theta)$ is (strictly) invisible to $\boldsymbol{y}$ (i.e., $\mathbb{V}(\boldsymbol{x}' \leftrightarrow \boldsymbol{y}) = 0$). In this case, to facilitate the specification of $\mathrm{B}(\boldsymbol{q})$, we first define a **boundary-distance function** B^dist for any spatial point $\boldsymbol{x}$ as

$$\mathrm{B}^\mathrm{dist}(\boldsymbol{x}) \coloneqq \min\left(\left|\overrightarrow{\boldsymbol{x}\boldsymbol{y}} \cdot \boldsymbol{n}(\boldsymbol{x})\right|,\ r^\partial(\boldsymbol{x})\right), \tag{18}$$

where $\overrightarrow{\boldsymbol{x}\boldsymbol{y}}$ denotes the unit vector pointing from $\boldsymbol{x}$ to $\boldsymbol{y}$, and $r^\partial(\boldsymbol{x})$ equals the shortest distance from $\boldsymbol{x}$ to an open boundary of the surface that contains $\boldsymbol{x}$ and $+\infty$ if the surface containing $\boldsymbol{x}$ is closed (and, thus, has no topological boundary). Based on this construction, it is easy to verify that, when $\boldsymbol{x}$ approaches a visibility boundary with respect to $\boldsymbol{y}$, $\mathrm{B}^\mathrm{dist}(\boldsymbol{x})$ approaches zero.

With the boundary-distance function B^dist specified, we define the boundary-test function $\mathrm{B}(\boldsymbol{q})$ based on the surface normal $\boldsymbol{n}(\boldsymbol{x}')$ at the spatial representation $\boldsymbol{x}' = \mathsf{X}(\boldsymbol{q}, \theta)$ as follows.

- When $\boldsymbol{n}(\boldsymbol{x}') \cdot \overrightarrow{\boldsymbol{x}'\boldsymbol{y}} > 0$, $\boldsymbol{x}'$ lies on a "front-facing" surface with respect to $\boldsymbol{y}$ (see Figure 7-a), and the open line segment $\overline{\boldsymbol{y}\boldsymbol{x}'}$ must intersect an occluder at $\boldsymbol{x}''$. We set $\mathrm{B}(\boldsymbol{q}) = \mathrm{B}^\mathrm{dist}(\boldsymbol{x}'')$.
- When $\boldsymbol{n}(\boldsymbol{x}') \cdot \overrightarrow{\boldsymbol{x}'\boldsymbol{y}} < 0$, $\boldsymbol{x}'$ is located on a "back-facing" surface with respect to $\boldsymbol{y}$ (see Figure 7-b). We set $\mathrm{B}(\boldsymbol{q}) = \mathrm{B}^\mathrm{dist}(\boldsymbol{x}')$.

3.2.3 Completing the derivation.

After constructing the continuous vector field $\boldsymbol{v}$ in §3.2.2, the right-hand side of Eq. (10) becomes well defined.

Lastly, according to the definition of the domain $\mathcal{B}_\mathrm{e}^\mathrm{wa}$ in Eq. (12), the spatial representation $\boldsymbol{x} = \mathsf{X}(\boldsymbol{p}, \theta)$ of any material point $\boldsymbol{p} \notin \mathcal{B}_\mathrm{e}^\mathrm{wa}$ must be invisible to $\boldsymbol{y}$ (i.e., $\mathbb{V}(\boldsymbol{x} \leftrightarrow \boldsymbol{y}) = 0$). Since the mutual visibility $\mathbb{V}(\boldsymbol{x} \leftrightarrow \boldsymbol{y})$ is a factor of $F_\mathrm{dir}(\boldsymbol{p})$, it follows that that $F_\mathrm{dir}(\boldsymbol{p}) = 0$ and, in turn,

$$[\nabla \cdot (F_\mathrm{dir}\, \boldsymbol{v})](\boldsymbol{p}) = 0, \quad \text{for all } \boldsymbol{p} \notin \mathcal{B}_\mathrm{e}^\mathrm{wa}. \tag{19}$$

This allows us to extend the domain of the integral on the right-hand side of Eq. (10) from $\mathcal{B}_\mathrm{e}^\mathrm{wa}$ to the full reference surface $\mathcal{B}_\mathrm{e}$:

$$\int_{\mathcal{B}_\mathrm{e}^\mathrm{wa}} \nabla \cdot (F_\mathrm{dir}\, \boldsymbol{v})\, \mathrm{d}A = \int_{\mathcal{B}_\mathrm{e}} \nabla \cdot (F_\mathrm{dir}\, \boldsymbol{v})\, \mathrm{d}A, \tag{20}$$

which, when substituted into Eq. (4), yields Eq. (11)—our main result for this section.

Discussion. The original warped-area method has been applied to the *boundary* integral over spherical curves (obtained by differentiating spherical rendering integrals). We will discuss the relation of our result and Bangaru et al.'s [2020] warped-area method in Appendix A.

4 WARPED-AREA REPARAMETERIZATION OF DIFFERENTIAL PATH INTEGRALS

In what follows, we first provide a brief recap of *differential path integrals* [Zhang et al. 2020, 2021b] in §4.1 and then introduce the general form of our technique—the main result of this paper—in §4.2.

We summarize the important symbols used in this section in Table 2.

4.1 Preliminary: Differential Path Integrals

4.1.1 Path integrals for forward rendering.

At the foundation of many, if not most, advanced forward rendering techniques is the formulation that expresses the response I of a radiometric detector as a **path integral** [Veach 1997]:

$$I = \int_\Omega f(\bar{x})\, \mathrm{d}\mu(\bar{x}), \tag{21}$$

where $\bar{x} = (\boldsymbol{x}_0, \ldots, \boldsymbol{x}_N)$ denotes a **light transport path** (with $\boldsymbol{x}_0$ on an emitter and $\boldsymbol{x}_N$ on a detector), $\Omega = \bigcup_{N=1}^\infty \mathcal{M}^{N+1}$ is the **path space** (with $\mathcal{M}$ is the union of all object surfaces), and μ is the corresponding area-product measure.

The integrand of the path integral is the **measurement contribution function** f given by

$$f(\bar{x}) \coloneqq L_\mathrm{e}(\boldsymbol{x}_0 \to \boldsymbol{x}_1)\, W_\mathrm{e}(\boldsymbol{x}_{N-1} \to \boldsymbol{x}_N) \left[\prod_{n=1}^{N-1} f_\mathrm{s}(\boldsymbol{x}_{n-1} \to \boldsymbol{x}_n \to \boldsymbol{x}_{n+1})\right] \left[\prod_{n=1}^{N} G(\boldsymbol{x}_{n-1} \leftrightarrow \boldsymbol{x}_n)\right], \tag{22}$$

where L_e and W_e are the source emission and **detector importance** (or response), f_s is the bidirectional scattering distribution function (BSDF), and G is the geometric term.

Material-form path integrals. When the scene geometry $\mathcal{M}$ evolves with some parameter θ, using the material-form reparameterization described in §3.1, one can capture the evolution of $\mathcal{M}$ using a fixed reference surface $\mathcal{B}$ coupled with a deformation $\mathsf{X}(\cdot, \theta)$ that maps $\mathcal{B}$ to $\mathcal{M}(\theta)$ for any θ.

Let $\bar{\boldsymbol{p}} = (\boldsymbol{p}_0, \ldots, \boldsymbol{p}_N)$ be a **material light path** with each vertex $\boldsymbol{p}_n$ on a reference surface $\mathcal{B}$. Applying the mapping $\mathsf{X}(\cdot, \theta)$ to each vertex of $\bar{\boldsymbol{p}}$ produces a spatial light path $\bar{\boldsymbol{x}} = (\boldsymbol{x}_0, \ldots, \boldsymbol{x}_N)$ where $\boldsymbol{x}_n = \mathsf{X}(\boldsymbol{p}_n, \theta)$ for all $n = 0, 1, \ldots, N$.

Then, by changing the variable of integration from spatial light paths $\bar{\boldsymbol{x}}$ to material ones $\bar{\boldsymbol{p}}$, the path integral of Eq. (21) can be rewritten in material form as:

$$I = \int_{\hat{\Omega}} \hat{f}(\bar{\boldsymbol{p}})\, \mathrm{d}\mu(\bar{\boldsymbol{p}}), \tag{23}$$

where the domain of integration is the **material path space**

$$\hat{\Omega} := \bigcup_{N=1}^{\infty} \mathcal{B}^{N+1}, \tag{24}$$

comprising material paths (of finite lengths). Additionally, the integrand of Eq. (23) is the **material measurement contribution** $\hat{f}$ that equals the product of the original measurement contribution f defined in Eq. (22) and the ratio of the two integral elements $\mathrm{d}\mu(\bar{\boldsymbol{x}})/\mathrm{d}\mu(\bar{\boldsymbol{p}}) {=} \prod_{n=0}^{N} J(\boldsymbol{p}_n, \theta)$:

$$\hat{f}(\bar{\boldsymbol{p}}) := f(\bar{\boldsymbol{x}}) \prod_{n=0}^{N} J(\boldsymbol{p}_n, \theta)\,, \tag{25}$$

where J follows the definition in Eq. (3).

4.1.2 Differential path integrals. Zhang et al. [2020] have derived using Reynolds transport theorem [1903] derivatives of the material-form path integral (23) with respect to arbitrary parameter θ. The result can generally be expressed as **material-form differential path integrals**:

$$\frac{\mathrm{d}I}{\mathrm{d}\theta} = \overset{\text{interior}}{\boxed{\int_{\hat{\Omega}} \frac{\mathrm{d}\hat{f}(\bar{\boldsymbol{p}})}{\mathrm{d}\theta}\, \mathrm{d}\mu(\bar{\boldsymbol{p}})}} + \overset{\text{boundary}}{\boxed{\int_{\partial\hat{\Omega}} \Delta\hat{f}_K(\bar{\boldsymbol{p}})\, V_K(\boldsymbol{p}_K)\, \mathrm{d}\dot{\mu}(\bar{\boldsymbol{p}})}}\,, \tag{26}$$

where the *interior* and *boundary* components are, respectively, over continuous regions of $\hat{f}$ and the discontinuous boundaries separating these regions. In what follows, we explain the *boundary* component in more details.

Domain of integration, discontinuity curves. The *boundary* integral in Eq. (26) is over the **material boundary path space** $\partial\hat{\Omega}$. They contain paths where one of the path segment goes through the scene boundary (see Figure 8 left). The space equals the union of disjoint subspaces $\partial\hat{\Omega}_{N,K}$:

$$\partial\hat{\Omega} = \bigcup_{N=1}^{\infty} \bigcup_{K=0}^{N-1} \partial\hat{\Omega}_{N,K}. \tag{27}$$

Each subspace $\partial\hat{\Omega}_{N,K}$ comprises **material boundary paths** $\bar{\boldsymbol{p}} = (\boldsymbol{p}_0, \boldsymbol{p}_1, \ldots, \boldsymbol{p}_N)$ where $\boldsymbol{p}_n \in \mathcal{B}$ for all $n \neq K$ and $\boldsymbol{p}_K$ is constrained over a set of curves $\Delta\mathcal{B}_K \subset \mathcal{B}$ such that $\boldsymbol{p}_K$'s spatial representation $\boldsymbol{x}_K = \mathsf{X}(\boldsymbol{p}_K, \theta)$ resides on a visibility boundary with respect to $\boldsymbol{x}_{K+1}$. We call the spatial line segment $\overline{\boldsymbol{x}_K\, \boldsymbol{x}_{K+1}}$—which intersects the scene geometry $\mathcal{M}(\theta)$ at a single point $\boldsymbol{x}^{\mathrm{B}}$—a **boundary segment** (see Figure 8).

Similar to $\Delta\mathcal{B}_{\mathrm{e}}$ for one-bounce light transport (see §3), the discontinuity curves $\Delta\mathcal{B}_K$ generally depend on the parameter θ. In fact, $\Delta\mathcal{B}_{\mathrm{e}}$ are essentially $\Delta\mathcal{B}_0$ (with $N = 2$).

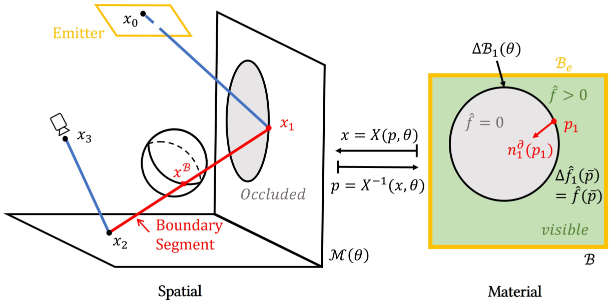


Fig. 8. The spatial representation of a material boundary path that is an element of the subspace $\partial\hat{\Omega}_{3,1}$ and has the boundary segment $\overline{\boldsymbol{x}_1\, \boldsymbol{x}_2}$ that intersects the scene at exactly one point $\boldsymbol{x}^{\mathrm{B}}$. Every path form this subspace contain four vertices $\boldsymbol{p}_0, \boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3$ with $\boldsymbol{p}_1$ constrained to a set of curves $\Delta\mathcal{B}_1$ comprising points whose spatial representations beside on visibility boundaries with respect to $\boldsymbol{x}_2$. To simplify the illustration, we only show part of the reference surface $\mathcal{B}$ that corresponds to the vertical plane containing the constrained $\boldsymbol{x}_1$.

Lastly, the measure $\dot{\mu}$ associated with the material boundary path space $\partial\hat{\Omega}$ satisfies that

$$\mathrm{d}\dot{\mu}(\bar{\boldsymbol{p}}) = \mathrm{d}\ell(\boldsymbol{p}_K) \prod_{n \neq K} \mathrm{d}A(\boldsymbol{p}_n). \tag{28}$$

Difference in measurement contribution. The term $\Delta\hat{f}_K(\bar{\boldsymbol{p}})$ captures the difference in material measurement contribution $\hat{f}$ when the vertex $\boldsymbol{p}_K$ crosses the discontinuity boundary $\Delta\mathcal{B}_K$ (with all the other vertices fixed).

As stated in §3.1.1, we assume without loss of generality that the boundary $\Delta\mathcal{B}_K$'s normal $\boldsymbol{n}_K^{\partial}$ points toward the occluded side (as shown on the right of Figure 8). Then, it holds that

$$\Delta\hat{f}_K(\bar{\boldsymbol{p}}) = \hat{f}(\bar{\boldsymbol{p}}). \tag{29}$$

Boundary velocity. The term $V_K(\boldsymbol{p}_K)$ is the scalar normal velocity defined as

$$V_K(\boldsymbol{p}_K) = \boldsymbol{v}_K^{\partial}(\boldsymbol{p}_K) \cdot \boldsymbol{n}_K^{\partial}(\boldsymbol{p}_K), \tag{30}$$

where $\boldsymbol{v}_K^{\partial}(\boldsymbol{p}_K)$ is the boundary velocity of the discontinuity point $\boldsymbol{p}_K$ and can be computed using the process described by Eqs. (7) and (8) in §3.1.2.

In this case, the spatial point $\boldsymbol{z}$ (that coincides $\boldsymbol{x}_K$) in Eq. (7) should be computed by intersecting the ray $\boldsymbol{x}_{K+1} \to \boldsymbol{x}^{\mathrm{B}}$ and the scene geometry $\mathcal{M}(\theta)$. When differentiating this ray-intersection process, the derivatives $\mathrm{d}\boldsymbol{x}_{K+1}/\mathrm{d}\theta$ and $\mathrm{d}\boldsymbol{x}^{\mathrm{B}}/\mathrm{d}\theta$ are computed by differentiating the mappings $\mathsf{X}(\boldsymbol{p}_{K+1}, \theta)$ and $\mathsf{X}(\boldsymbol{p}^{\mathrm{B}}, \theta)$, respectively. For the latter, $\boldsymbol{p}^{\mathrm{B}} := \mathrm{detach}[\mathsf{X}^{-1}(\boldsymbol{x}^{\mathrm{B}}, \theta)]$ is the "detached" material representation of the point $\boldsymbol{x}^{\mathrm{B}}$.

4.2 Warped-Area Reparameterization of Boundary Path Integrals

We now generalize our one-bounce result derived in §3.2 to reparameterize the *boundary* component of the material-form differential path integrals in Eq. (26).

Derivation outline. To this end, our derivation will involve the following three main steps:

Table 2. Commonly used symbols in §4. The right-most column indicates θ-dependency.

Symbol	Definition	θ-dep.
$\mathcal{M}$	The union of all surfaces in the scene	Yes
$\mathcal{B}$	Reference surface	**No**
$\mathsf{X}(\cdot,\theta)$	Differentiable one-to-one mapping (aka. deformation) transforming $\mathcal{B}$ to $\mathcal{M}(\theta)$	Yes
$\hat{\Omega}$	material path space defined in Eq. (24)	**No**
$\partial\hat{\Omega}$	material boundary path space defined in Eq. (27)	Yes
$\hat{f}$	material measurement distribution defined in Eq. (25)	Yes
$\hat{f}_K$	Equals $\hat{f}$ but treats the vertex $\boldsymbol{p}_K$ as the only variable (and considers all the other vertices fixed)	Yes
$\Delta\mathcal{B}_K$	Jump discontinuities of $\hat{f}_K$ with respect to $\boldsymbol{p}_K$	Yes
$\mathcal{B}_K^{\mathrm{wa}}$	Regions on the reference surface $\mathcal{B}$ comprising points $\boldsymbol{p}_K$ such that $\mathsf{X}(\boldsymbol{p}_K,\theta)$ is visible to $\boldsymbol{x}_{K+1}$	Yes
$\boldsymbol{n}_K^{\partial}$	Unit-normal field associated with $\Delta\mathcal{B}_K$	Yes
$\boldsymbol{v}_K^{\partial}$	Boundary velocity over $\Delta\mathcal{B}_K$	Yes
$\boldsymbol{v}_K$	Continuous velocity field over $\mathcal{B}$ that agrees with $\boldsymbol{v}_K^{\partial}$ on the boundary $\Delta\mathcal{B}_K$	Yes

(1) Expanding the *boundary* path integral in Eq. (26) as nested integrals of individual path vertices $\{\boldsymbol{p}_n : n = 0, 1, \ldots, N\}$. This allows us to isolate the integral with respect to $\boldsymbol{p}_K$ over discontinuity curves $\Delta\mathcal{B}_K$.

(2) Reparameterizing the isolated line integral as an integral over the reference surface $\mathcal{B}$ using the warped-area method discussed in §3.2.

(3) Rewriting the nested integrals with the reparameterized surface integral with respect to $\boldsymbol{p}_K$ as a new *interior* path integral which, in turn, is merged with the *interior* component of Eq. (26) to produce our *reparameterized differential path integral* in Eq. (38).

In the following, we detail each of these steps.

Step 1. We first express the *boundary* path integral as a sum of integrals over the subspaces $\partial\hat{\Omega}_{N,K}$ defined below Eq. (27):

$$\int_{\partial\hat{\Omega}} \hat{f}(\bar{\boldsymbol{p}})\, V_K(\boldsymbol{p}_K)\, \mathrm{d}\dot{\mu}(\bar{\boldsymbol{p}}) = \sum_{N=1}^{\infty} \sum_{K=0}^{N-1} \int_{\partial\hat{\Omega}_{N,K}} \hat{f}(\bar{\boldsymbol{p}})\, V_K(\boldsymbol{p}_K)\, \mathrm{d}\dot{\mu}(\bar{\boldsymbol{p}}). \tag{31}$$

For all N, K, the integral on the right-hand side of Eq. (31) can be further rewritten as nested integrals of individual vertices of the material path $\bar{\boldsymbol{p}} = (\boldsymbol{p}_0, \ldots, \boldsymbol{p}_N)$:

$$\begin{aligned} &\int_{\partial\hat{\Omega}_{N,K}} \hat{f}(\bar{\boldsymbol{p}})\, V_K(\boldsymbol{p}_K)\, \mathrm{d}\dot{\mu}(\bar{\boldsymbol{p}}) \\ &= \int_{\mathcal{B}^N} \left(\int_{\Delta\mathcal{B}_K} \hat{f}(\bar{\boldsymbol{p}})\, V_K(\boldsymbol{p}_K)\, \mathrm{d}\ell(\boldsymbol{p}_K) \right) \prod_{n \neq K} \mathrm{d}A(\boldsymbol{p}_n). \end{aligned} \tag{32}$$

On the right-hand side of this equation, the inner integral—which depends on $\boldsymbol{p}_n$ for all $n \neq K$—is with respect to the vertex $\boldsymbol{p}_K$ and over the discontinuity curves $\Delta\mathcal{B}_K$:

$$\begin{aligned} I(\boldsymbol{p}_n : n \neq K) &:= \int_{\Delta\mathcal{B}_K} \hat{f}(\bar{\boldsymbol{p}})\, V_K(\boldsymbol{p}_K)\, \mathrm{d}\ell(\boldsymbol{p}_K) \\ &= \int_{\Delta\mathcal{B}_K} \left(\hat{f}(\bar{\boldsymbol{p}})\, \boldsymbol{v}_K^{\partial}(\boldsymbol{p}_K) \right) \cdot \boldsymbol{n}_K^{\partial}(\boldsymbol{p}_K)\, \mathrm{d}\ell(\boldsymbol{p}_K). \end{aligned} \tag{33}$$

Step 2. Eq. (33) is essentially identical to the one-bounce *boundary* integral on the left-hand side of Eq. (10). Therefore, we can apply our warped-area reparameterization given by Eq. (10) to rewrite Eq. (33) as a surface integral:

$$\boxed{I(\boldsymbol{p}_n : n \neq K) = \int_{\mathcal{B}_K^{\mathrm{wa}}} \left[\nabla \cdot \left(\hat{f}_K\, \boldsymbol{v}_K \right) \right] (\boldsymbol{p}_K)\, \mathrm{d}A(\boldsymbol{p}_K).} \tag{34}$$

In this equation:

- The domain of integration $\mathcal{B}_K^{\mathrm{wa}}$ follows the definition in Eq. (12) and comprises all material points $\boldsymbol{p}$ whose spatial representations $\boldsymbol{x}_K = \mathsf{X}(\boldsymbol{p}_K, \theta)$ are visible to the (spatial) vertex $\boldsymbol{x}_{K+1} = \mathsf{X}(\boldsymbol{p}_{K+1}, \theta)$. That is, $\mathcal{B}_K^{\mathrm{wa}} = \{\boldsymbol{p}_K \in \mathcal{B} : \mathbb{V}(\boldsymbol{x}_K \leftrightarrow \boldsymbol{x}_{K+1}) = 1\}$.
- The vector field $\boldsymbol{v}_K$ is obtained using the process presented in §3.2.2 (that smoothly extends the velocity $\boldsymbol{v}_K^{\partial}(\boldsymbol{p}_K)$ defined on the boundary $\Delta\mathcal{B}_K$ into the interior of $\mathcal{B}_K^{\mathrm{wa}}$).

Additionally, the term $\hat{f}_K$ in Eq. (34) equals the material measurement contribution $\hat{f}$ but considers $\boldsymbol{p}_K$ the only variable and all the other vertices $\{\boldsymbol{p}_n : n \neq K\}$ constants. Specifically, let $\hat{g}_K(\boldsymbol{p}_K)$ be the product of all factors of $\hat{f}$ that depend on $\boldsymbol{p}_K$:[2]

$$\begin{aligned} \hat{g}_K(\boldsymbol{p}_K) := {} & G(\boldsymbol{x}_{K-1} \leftrightarrow \boldsymbol{x}_K)\, G(\boldsymbol{x}_K \leftrightarrow \boldsymbol{x}_{K+1}) \\ & f_{\mathrm{s}}(\boldsymbol{x}_{K-2} \to \boldsymbol{x}_{K-1} \to \boldsymbol{x}_K)\, f_{\mathrm{s}}(\boldsymbol{x}_{K-1} \to \boldsymbol{x}_K \to \boldsymbol{x}_{K+1}) \\ & f_{\mathrm{s}}(\boldsymbol{x}_K \to \boldsymbol{x}_{K+1} \to \boldsymbol{x}_{K+2})\, J(\boldsymbol{p}_K), \end{aligned} \tag{35}$$

where $\boldsymbol{x}_n = \mathsf{X}(\boldsymbol{p}_n, \theta)$ is spatial representation of material point $\boldsymbol{p}_n$ for $n = K-1, K, K+1$. Then, it holds that

$$\hat{f}_K(\bar{\boldsymbol{p}}) = \hat{g}_K(\boldsymbol{p}_K)\, \mathrm{detach}\left(\frac{\hat{f}(\bar{\boldsymbol{p}})}{\hat{g}_K(\boldsymbol{p}_K)} \right). \tag{36}$$

We note that $\hat{f}_K(\bar{\boldsymbol{p}})$ will be differentiated only with respect to the point $\boldsymbol{p}_K$ (and not the scene parameter θ) when evaluating the divergence $\nabla \cdot (\hat{f}_K\, \boldsymbol{v}_K)$—which we will discuss in §5.1.

Step 3. As shown in §3.2.3, the divergence $[\nabla \cdot (\hat{f}_K\, \boldsymbol{v}_K)](\boldsymbol{p}_K) = 0$ for all $\boldsymbol{p}_K \notin \mathcal{B}_K^{\mathrm{wa}}$ because the mutual visibility $\mathbb{V}(\boldsymbol{x}_K \leftrightarrow \boldsymbol{x}_{K+1}) = 0$ by definition. This allows us to extend the domain of the reparameterized surface integral in Eq. (34) from $\mathcal{B}_K^{\mathrm{wa}}$ to the full reference surface $\mathcal{B}$. Then, substituting the extended Eq. (34) into Eqs. (31)

[2] We omit the dependency of $\hat{g}_K$ on the points $\boldsymbol{p}_{K-1}$ and $\boldsymbol{p}_{K+1}$ (that are considered constants) for notational simplicity.

and (32) produces

$$\int_{\partial\hat{\Omega}} \hat{f}(\bar{\boldsymbol{p}})\, V_K(\boldsymbol{p}_K)\, \mathrm{d}\dot{\mu}(\bar{\boldsymbol{p}})$$
$$= \sum_{N=1}^{\infty} \sum_{K=0}^{N-1} \int_{\mathcal{B}^{N+1}} \left[\nabla \cdot \left(\hat{f}_K\, \boldsymbol{v}_K\right)\right](\boldsymbol{p}_K)\, \mathrm{d}\mu(\bar{\boldsymbol{p}}) \tag{37}$$
$$= \int_{\hat{\Omega}} \left[\sum_{K=0}^{N-1} \left[\nabla \cdot \left(\hat{f}_K\, \boldsymbol{v}_K\right)\right](\boldsymbol{p}_K)\right] \mathrm{d}\mu(\bar{\boldsymbol{p}}),$$

where the second equality is obtained by: (i) exchanging inner summation and the integration; and (ii) rewriting the sum of integrals over $\mathcal{B}^{N+1}$ for all N as one integral over the material path space $\partial\hat{\Omega}$.

Lastly, substituting Eq. (37) into the material-form differential path integrals (26) yields the main result of this paper—the **reparameterized differential path integral**:

$$\frac{\mathrm{d}I}{\mathrm{d}\theta} = \overset{\textbf{interior}}{\boxed{\int_{\hat{\Omega}} \left(\frac{\mathrm{d}\hat{f}(\bar{\boldsymbol{p}})}{\mathrm{d}\theta} + \sum_{K=0}^{N-1} \left[\nabla \cdot \left(\hat{f}_K\, \boldsymbol{v}_K\right)\right](\boldsymbol{p}_K)\right) \mathrm{d}\mu(\bar{\boldsymbol{p}})}}, \tag{38}$$

where N indicates the length (i.e., number of segments) of the material light path $\bar{\boldsymbol{p}}$.

5 OUR MONTE CARLO ESTIMATORS

We now introduce Monte Carlo estimators for our reparameterized differential path integral expressed in Eq. (38). Specifically, we first discuss in §5.1 the estimation of the divergence term $\nabla \cdot (\hat{f}_K\, \boldsymbol{v}_K)$. Then, in §5.2, we introduce our path-space Monte Carlo estimators and how do we cache values along paths to increase computational efficiency.

5.1 Estimating Divergence

A key component of our reparameterized differential path integral of Eq. (38) is the divergence

$$\nabla \cdot \left(\hat{f}_K\, \boldsymbol{v}_K\right) = \left(\nabla \hat{f}_K\right) \cdot \boldsymbol{v}_K + \hat{f}_K\, (\nabla \cdot \boldsymbol{v}_K), \tag{39}$$

at some vertex $\boldsymbol{p}_K$ of a given material path $\bar{\boldsymbol{p}} = (\boldsymbol{p}_0, \ldots, \boldsymbol{p}_N)$. In the following, we discuss the estimation of $\hat{f}_K$ and $\nabla\hat{f}_K$ in §5.1.1 as well as $\boldsymbol{v}_K$ and $\nabla \cdot \boldsymbol{v}_K$ in §5.1.2.

5.1.1 Measurement contribution. The right-hand side of Eq. (39) involves the term $\hat{f}_K$ and its gradient $\nabla\hat{f}_K$. We recall that, as defined in Eq. (36), $\hat{f}_K(\bar{\boldsymbol{p}})$ equals the material measurement contribution $\hat{f}(\bar{\boldsymbol{p}})$ except for treating $\boldsymbol{p}_K$ as the only variable and all other vertices $\{\boldsymbol{p}_n \,:\, n \neq K\}$ as constants. Therefore, we have $\hat{f}_K(\bar{\boldsymbol{p}}) = \hat{f}(\bar{\boldsymbol{p}})$, and:

$$\left[\nabla \hat{f}_K\right](\bar{\boldsymbol{p}}) = \left[\frac{\partial \hat{f}_K}{\partial \boldsymbol{p}_K}\right](\bar{\boldsymbol{p}}) = [\nabla \hat{g}_K](\boldsymbol{p}_K)\, \frac{\hat{f}(\bar{\boldsymbol{p}})}{\hat{g}_K(\boldsymbol{p}_K)}, \tag{40}$$

where $\hat{g}_K$ is defined in Eq. (35) and the gradient $\nabla\hat{g}_K := \partial\hat{g}_K/\partial\boldsymbol{p}_K$ can be computed using automatic differentiation.

5.1.2 Continuous velocity. The remaining terms on the right-hand side of Eq. (39) are the continuous velocity $\boldsymbol{v}_K(\boldsymbol{p}_K)$ and its divergence $[\nabla \cdot \boldsymbol{v}_K](\boldsymbol{p}_K)$ (given the neighboring vertex $\boldsymbol{p}_{K+1}$ that is considered fixed). We outline this process in Algorithm 1.

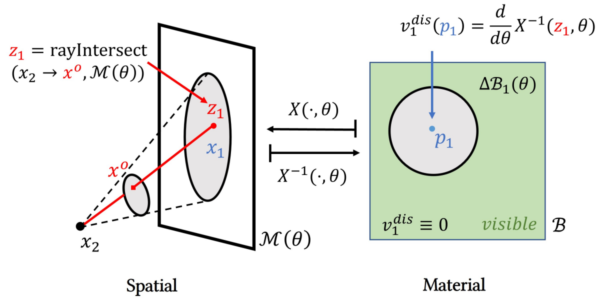


Fig. 9. **Discontinuous velocity $\boldsymbol{v}_1^{\mathrm{dis}}$**: Let $\boldsymbol{p}_2 \in \mathcal{B}$ be a fixed material point $\boldsymbol{p}_2 \in \mathcal{B}$ with the spatial representation $\boldsymbol{x}_2 = \mathsf{X}(\boldsymbol{p}_2, \theta)$. For each point $\boldsymbol{p}_1$, we set the discontinuous velocity $\boldsymbol{v}_1^{\mathrm{dis}}(\boldsymbol{p}_1)$ to zero if $\boldsymbol{p}_1$'s the spatial counterpart $\boldsymbol{x}_1 = \mathsf{X}(\boldsymbol{p}_1, \theta)$ is visible to $\boldsymbol{x}_2$. Otherwise, there must exist an intersection $\boldsymbol{x}^{\mathrm{O}}$ (illustrated as the red square) between the open line segment $\overline{\boldsymbol{x}_1\boldsymbol{x}_2}$ and some occluder. We compute $\boldsymbol{z}_1 = \mathrm{rayIntersect}(\boldsymbol{x}_2 \to \boldsymbol{x}^{\mathrm{O}}, \mathcal{M})$ and set the velocity as $\boldsymbol{v}_1^{\mathrm{dis}}(\boldsymbol{p}_1) = \frac{\mathrm{d}}{\mathrm{d}\theta} \mathsf{X}^{-1}(\boldsymbol{z}_1, \theta)$.

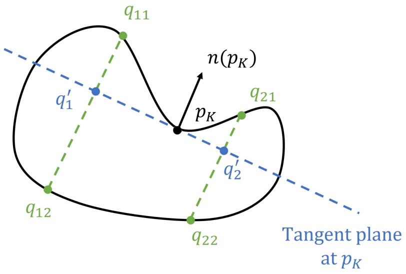


Fig. 10. We sample points $\{\boldsymbol{q}_{ij}\}$ near a point $\boldsymbol{p}_K$ by drawing points $\boldsymbol{q}_i'$ in the tangent plane of $\boldsymbol{p}_K$ (from a Gaussian distribution centered at $\boldsymbol{p}_K$) and finding all projections of $\boldsymbol{q}_i'$ on the reference $\mathcal{B}$ along the surface normal $\boldsymbol{n}(\boldsymbol{p}_K)$.

According to the definition in Eq. (14), we express the continuous velocity $\boldsymbol{v}_K(\boldsymbol{p}_K)$ as

$$\boldsymbol{v}_K(\boldsymbol{p}_K) = \tilde{\boldsymbol{v}}_K(\boldsymbol{p}_K) / W(\boldsymbol{p}_K)\,. \tag{41}$$

On the right-hand of this equation, the denominator is the *normalization term* W given by

$$W(\boldsymbol{p}_K) := \int_{\mathcal{B}} \mathrm{w}(\boldsymbol{q}; \boldsymbol{p}_K)\, \mathrm{d}A(\boldsymbol{q}), \tag{42}$$

and the numerator is the *convolved velocity* $\tilde{\boldsymbol{v}}_K$ defined as

$$\tilde{\boldsymbol{v}}_K(\boldsymbol{p}_K) := \int_{\mathcal{B}} \mathrm{w}(\boldsymbol{q}; \boldsymbol{p}_K)\, \boldsymbol{v}_K^{\mathrm{dis}}(\boldsymbol{q})\, \mathrm{d}A(\boldsymbol{q}). \tag{43}$$

In Eq. (43), $\boldsymbol{v}_K^{\mathrm{dis}}$ is the discontinuous vector field that agrees with the boundary velocity $\boldsymbol{v}_K^{\partial}$ and can be constructed (conditioned on the vertex $\boldsymbol{p}_{K+1}$) using the process described in Step 1 of §3.2.2 (see Figure 9).

Sampling $\boldsymbol{q}$. Estimating the continuous velocity $\boldsymbol{v}_K$ and its divergence $(\nabla \cdot \boldsymbol{v}_K)$ based on Eqs. (42) and (43) requires sampling material points $\boldsymbol{q}$ from the reference surface $\mathcal{B}$.

ALGORITHM 1: Consistent estimator for the velocity $\boldsymbol{v}_K$ and its divergence $\nabla \cdot \boldsymbol{v}_K$

```
Estimate_Velocity($\boldsymbol{p}_K, \boldsymbol{p}_{K+1}$)
Input: Two vertices $\boldsymbol{p}_K$ and $\boldsymbol{p}_{K+1}$ of a material path
Output: $\boldsymbol{v}_K(\boldsymbol{p}_K)$ and $[\nabla \cdot \boldsymbol{v}_K](\boldsymbol{p}_K)$
begin
    /* Estimate $W$ and $\tilde{\boldsymbol{v}}_K$ (unbiased and consistent)   */
    $W \leftarrow 0$; $\nabla W \leftarrow \mathbf{0}$;
    $\tilde{\boldsymbol{v}}_K \leftarrow \mathbf{0}$; $\nabla \cdot \tilde{\boldsymbol{v}}_K \leftarrow 0$;
    for $i = 1$ to $N_q$ do
        Sample point $\boldsymbol{q}'_i$ from the tangent plane of $\boldsymbol{p}_K$ with
          probability $\mathrm{pdf}(\boldsymbol{q}'_i)$;
        Compute projections $\{\boldsymbol{q}_{ij} : j = 1, 2, \ldots\}$ of $\boldsymbol{q}'_i$ on the
          reference surface $\mathcal{B}$;
        foreach projection $\boldsymbol{q}_{ij}$ of $\boldsymbol{q}'_i$ do
            $\mathrm{w} \leftarrow \mathrm{w}(\boldsymbol{q}_{ij}; \boldsymbol{p}_K)$;
            $\nabla \mathrm{w} \leftarrow [\nabla \mathrm{w}(\cdot; \boldsymbol{p}_K)](\boldsymbol{q}_{ij})$;
            $\mathrm{pdf} \leftarrow \mathrm{pdf}(\boldsymbol{q}'_i)\, |\boldsymbol{n}(\boldsymbol{p}_K) \cdot \boldsymbol{n}(\boldsymbol{q}_{ij})|$;
            Compute $\boldsymbol{v}^{\mathrm{dis}} = \boldsymbol{v}^{\mathrm{dis}}_K(\boldsymbol{q}_{ij})$ and $\nabla \cdot \boldsymbol{v}^{\mathrm{dis}}$ (using AD)
             conditioned on $\boldsymbol{p}_{K+1}$ ;              // §3.2.2, Step 1
            $W$ += $\mathrm{w}/\mathrm{pdf}$ ;                          // Eq. (44)
            $\nabla W$ += $\nabla \mathrm{w}/\mathrm{pdf}$ ;                       // Eq. (60)
            $\tilde{\boldsymbol{v}}_K$ += $(\mathrm{w}\, \boldsymbol{v}^{\mathrm{dis}})/\mathrm{pdf}$ ;                   // Eq. (45)
            $\nabla \cdot \tilde{\boldsymbol{v}}_K$ += $(\nabla \mathrm{w} \cdot \boldsymbol{v}^{\mathrm{dis}} + \mathrm{w}\, (\nabla \cdot \boldsymbol{v}^{\mathrm{dis}}))/\mathrm{pdf}$ ; // Eq. (61)
        end
    end
    /* Estimate $\boldsymbol{v}_K$ and $\nabla \cdot \boldsymbol{v}_K$ (biased but consistent) */
    $\boldsymbol{v}_K \leftarrow \tilde{\boldsymbol{v}}_K / W$ ;                            // Eq. (47)
    $\nabla \cdot \boldsymbol{v}_K \leftarrow (\nabla \cdot \tilde{\boldsymbol{v}}_K)/W - (\nabla W \cdot \tilde{\boldsymbol{v}}_K)/W^2$ ;      // Eq. (62)
    return $\boldsymbol{v}_K, \nabla \cdot \boldsymbol{v}_K$;
end
```

To this end, we employ a process similar to the sampling of outgoing locations for *bidirectional subsurface scattering distribution functions* (BSSRDFs) as follows. As illustrated in Figure 10, we first draw a point $\boldsymbol{q}'$ inside the tangent plane of $\boldsymbol{p}_K$ from a 2D (isotropic) Gaussian distribution centered at $\boldsymbol{p}_K$. Then, we project $\boldsymbol{q}'$ onto the reference surface $\mathcal{B}$ along the surface normal $\boldsymbol{n}(\boldsymbol{p}_K)$ and return all resulting projections $\mathcal{B} \cap \{\boldsymbol{q}' + t\, \boldsymbol{n}(\boldsymbol{p}_K) \ : \ t \in \mathbb{R}\}$ as samples of $\boldsymbol{q}$.

Estimating $\boldsymbol{v}_K$. Leveraging the same point sampling method described above, we estimate the *normalization term* W and the *convolved velocity* $\tilde{\boldsymbol{v}}_K$ in an unbiased and consistent fashion as follows.

First, we draw N_q samples $\{\boldsymbol{q}'_i \ : \ i = 1, 2, \ldots, N_q\}$ of $\boldsymbol{q}$ from the tangent plane of $\boldsymbol{p}_K$. Let the probability for drawing $\boldsymbol{q}'_i$ be $\mathrm{pdf}(\boldsymbol{q}'_i)$. Then, unbiased estimators of W and $\tilde{\boldsymbol{v}}_K$ can be obtained via

$$\left\langle W(\boldsymbol{p}_K) \right\rangle = \sum_{i=1}^{N_q} \sum_j \frac{\mathrm{w}(\boldsymbol{q}_{ij}; \boldsymbol{p}_K)}{\mathrm{pdf}(\boldsymbol{q}_{ij})}, \tag{44}$$

$$\left\langle \tilde{\boldsymbol{v}}_K(\boldsymbol{p}_K) \right\rangle = \sum_{i=1}^{N_q} \sum_j \frac{\mathrm{w}(\boldsymbol{q}_{ij}; \boldsymbol{p}_K)\, \boldsymbol{v}^{\mathrm{dis}}_K(\boldsymbol{q}_{ij})}{\mathrm{pdf}(\boldsymbol{q}_{ij})}, \tag{45}$$

where $\{\boldsymbol{q}_{ij} \ : \ j = 1, 2, \ldots\}$ are the projections of $\boldsymbol{q}'_i$ on $\mathcal{B}$, and

$$\mathrm{pdf}(\boldsymbol{q}_{ij}) = \mathrm{pdf}(\boldsymbol{q}'_i)\, \left|\boldsymbol{n}(\boldsymbol{p}_K) \cdot \boldsymbol{n}(\boldsymbol{q}_{ij})\right|. \tag{46}$$

ALGORITHM 2: Our *unidirectional* estimator for the reparameterized differential path integral of Eq. (38).

```
Path_Tracing()
// We use $\boldsymbol{x}^{\mathrm{D}}_n$ and $\boldsymbol{x}^{\mathrm{S}}_n$ to denote, respectively, the
   spatial representations of $\boldsymbol{p}^{\mathrm{D}}_n$ and $\boldsymbol{p}^{\mathrm{S}}_n$ for all $n$
begin
    /* Initialize detector subpath                                */
    Sample camera ray $\boldsymbol{p}^{\mathrm{D}}_0 \to \boldsymbol{p}^{\mathrm{D}}_1$ with probability pdf;
    $\hat{f} \leftarrow W_{\mathrm{e}}(\boldsymbol{x}^{\mathrm{D}}_1 \to \boldsymbol{x}^{\mathrm{D}}_0)\, G(\boldsymbol{x}^{\mathrm{D}}_1 \leftrightarrow \boldsymbol{x}^{\mathrm{D}}_0)\, J(\boldsymbol{p}^{\mathrm{D}}_0)\, J(\boldsymbol{p}^{\mathrm{D}}_1)$;
    $H^{\mathrm{D}} \leftarrow 0$;   $\mathrm{d}I \leftarrow 0$;
    for $n = 1, 2, \ldots$ do
        /* Sample light vertex $\boldsymbol{p}^{\mathrm{S}}_n$                                 */
        Sample $\boldsymbol{p}^{\mathrm{S}}_n$ on the emitter with probability $\mathrm{pdf}^{\mathrm{NEE}}_n$;
        /* Compute $\hat{f}^{\mathrm{NEE}} = \hat{f}(\boldsymbol{p}^{\mathrm{S}}_n, \boldsymbol{p}^{\mathrm{D}}_n, \ldots, \boldsymbol{p}^{\mathrm{D}}_0)$ and $\mathrm{d}\hat{f}^{\mathrm{NEE}}/\mathrm{d}\theta$   */
        $\hat{f}^{\mathrm{NEE}} \leftarrow \hat{f} \cdot f_{\mathrm{s}}(\boldsymbol{x}^{\mathrm{S}}_n \to \boldsymbol{x}^{\mathrm{D}}_n \to \boldsymbol{x}^{\mathrm{D}}_{n-1})\, G(\boldsymbol{x}^{\mathrm{S}}_n \leftrightarrow \boldsymbol{x}^{\mathrm{D}}_n)\, J(\boldsymbol{p}^{\mathrm{S}}_n)$;
        $\hat{f}^{\mathrm{NEE}}$ *= $L_{\mathrm{e}}(\boldsymbol{x}^{\mathrm{S}}_n \to \boldsymbol{x}^{\mathrm{D}}_n)$;
        Compute $\mathrm{d}\hat{f}^{\mathrm{NEE}} := \mathrm{d}\hat{f}^{\mathrm{NEE}}/\mathrm{d}\theta$ using AD;
        /* Compute the sum of divergences via Eq. (49) */
        $\Sigma_{\mathrm{div}} \leftarrow \hat{f}^{\mathrm{NEE}} \left(H^{\mathrm{D}} + h(\boldsymbol{p}^{\mathrm{D}}_n; \boldsymbol{p}^{\mathrm{S}}_n, \boldsymbol{p}^{\mathrm{D}}_{n-1}) + h(\boldsymbol{p}^{\mathrm{S}}_n; \boldsymbol{p}^{\mathrm{D}}_n)\right)$;
        /* Accumulate path contribution                            */
        $\mathrm{d}I$ += $(\mathrm{d}\hat{f}^{\mathrm{NEE}} + \Sigma_{\mathrm{div}})/(\mathrm{pdf} \cdot \mathrm{pdf}^{\mathrm{NEE}}_n)$;
        /* Sample next vertex $\boldsymbol{p}^{\mathrm{D}}_{n+1}$                              */
        Sample $\boldsymbol{p}^{\mathrm{D}}_{n+1}$ with probability $\mathrm{pdf}_{n+1}$;
        pdf *= $\mathrm{pdf}_{n+1}$;
        /* Update measurement contribution                         */
        $\hat{f}$ *= $f_{\mathrm{s}}(\boldsymbol{x}^{\mathrm{D}}_{n+1} \to \boldsymbol{x}^{\mathrm{D}}_n \to \boldsymbol{x}^{\mathrm{D}}_{n-1})\, G(\boldsymbol{x}^{\mathrm{D}}_{n+1} \leftrightarrow \boldsymbol{x}^{\mathrm{D}}_n)\, J(\boldsymbol{p}^{\mathrm{D}}_{n+1})$;
        /* Update pre-fix sum based on Eq. (50)                     */
        $H^{\mathrm{D}}$ += $h(\boldsymbol{p}^{\mathrm{D}}_n; \boldsymbol{p}^{\mathrm{D}}_{n+1}, \boldsymbol{p}^{\mathrm{D}}_{n-1})$;
    end
    return $\mathrm{d}I$;
end
```

Lastly, given Eq. (41), a biased-but-consistent estimator of the continuous velocity $\boldsymbol{v}_K$ can be obtained using the unbiased estimations of the *normalization term* W and the *convolved velocity* $\tilde{\boldsymbol{v}}_K$ given by Eqs. (44) and (45), respectively:

$$\left\langle \boldsymbol{v}_K(\boldsymbol{p}_K) \right\rangle = \left\langle \tilde{\boldsymbol{v}}_K(\boldsymbol{p}_K) \right\rangle / \left\langle W(\boldsymbol{p}_K) \right\rangle . \tag{47}$$

Further, the estimation of the divergence $\nabla \cdot \boldsymbol{v}_K$ can be accomplished in a similar fashion using unbiased estimates of the gradient $\nabla \mathrm{w}$ of the *normalization term* W and the divergence $\nabla \cdot \tilde{\boldsymbol{v}}_K$ of the *convolved velocity* $\tilde{\boldsymbol{v}}_K$. We provide more details of this process in Appendix B.

Debiasing. Bangaru et al. [2020] have introduced a process based on Russian roulette to debias estimators of $\boldsymbol{v}_K(\boldsymbol{p}_K)$ and $[\nabla \cdot \boldsymbol{v}_K](\boldsymbol{p}_K)$. On the other hand, achieving full unbiasedness requires the distance and boundary-test functions to satisfy specific conditions, which can be nontrivial to obtain (please see their work for more details). We consider the debiasing of our estimators a future research topic.

5.2 Path-Space Estimators

Our reparameterized differential path integral of Eq. (38) allows the development of path-space Monte Carlo estimators of the form:

$$\left\langle \frac{\mathrm{d}I}{\mathrm{d}\theta} \right\rangle = \frac{1}{\mathrm{pdf}(\bar{\boldsymbol{p}})} \left(\frac{\mathrm{d}\hat{f}(\bar{\boldsymbol{p}})}{\mathrm{d}\theta} + \sum_{K=0}^{N-1} \left[\nabla \cdot \left(\hat{f}_K \, \boldsymbol{v}_K \right) \right] (\boldsymbol{p}_K) \right), \tag{48}$$

where $\mathrm{pdf}(\bar{\boldsymbol{p}})$ denotes any probability density for sampling the material light path $\bar{\boldsymbol{p}} = (\boldsymbol{p}_0, \ldots, \boldsymbol{p}_N)$, and the divergence term $\nabla \cdot (\hat{f}_K \, \boldsymbol{v}_K)$ is computed using estimators presented in §5.1.

We introduce an *unidirectional* (§5.2.1) and a *bidirectional* estimator (§5.2.2) that, respectively, apply unidirectional path tracing (PT) with next-event estimation (NEE) and bidirectional path tracing (BDPT) to sample the material path $\bar{\boldsymbol{p}}$.

Time complexity of naïve implementation. Given a material path $\bar{\boldsymbol{p}}$ with N segments, evaluating Eq. (48) naïvely takes $O(N^2)$ time since computing the divergence $\nabla \cdot (\hat{f}_K \, \boldsymbol{v}_K)$—which involves computing and differentiating material measurement contribution $\hat{f}_K(\bar{\boldsymbol{p}})$ (with respect to $\boldsymbol{p}_K$)—takes $O(N)$ time for each K.

The time complexity becomes even higher when using PT with NEE and BDPT: For the former, each NEE effectively produces one full path, causing the cost for tracing one path to be $O(N^3)$; For the latter, since $O(N^2)$ paths are created[3] using one pair of source and detector subpaths, the total complexity becomes $O(N^4)$.

In what follows, we address this challenge by developing new techniques to estimate the divergence $\nabla \cdot (\hat{f}_K \, \boldsymbol{v}_K)$ efficiently for our *unidirectional* and *bidirectional* estimators.

5.2.1 *Our unidirectional estimator.*

We now detail our *unidirectional* estimator—which we outline in Algorithm 2.

For any material path $\bar{\boldsymbol{p}} = (\boldsymbol{p}_0, \ldots, \boldsymbol{p}_N)$, according to Eq. (40) and the fact that $\hat{f}_K(\bar{\boldsymbol{p}}) = \hat{f}(\bar{\boldsymbol{p}})$, we can rewrite the sum of divergence $\Sigma_{\mathrm{div}}(\bar{\boldsymbol{p}}) := \sum_K [\nabla \cdot (\hat{f}_K \, \boldsymbol{v}_K)](\boldsymbol{p}_K)$ by factoring out the material measurement contribution $\hat{f}(\bar{\boldsymbol{p}})$. That is,

$$\begin{aligned} \Sigma_{\mathrm{div}}(\bar{\boldsymbol{p}}) &:= \sum_{K=0}^{N-1} \left[\nabla \cdot (\hat{f}_K \, \boldsymbol{v}_K) \right] (\boldsymbol{p}_K) \\ &= \hat{f}(\bar{\boldsymbol{p}}) \sum_{K=0}^{N-1} \underbrace{\left(\frac{[\nabla \hat{g}_K](\boldsymbol{p}_K)}{\hat{g}_K(\boldsymbol{p}_K)} \cdot \boldsymbol{v}_K(\boldsymbol{p}_K) + [\nabla \cdot \boldsymbol{v}_K] \, (\boldsymbol{p}_K) \right)}_{=: \, h(\boldsymbol{p}_K; \, \boldsymbol{p}_{K-1}, \boldsymbol{p}_{K+1})}. \end{aligned} \tag{49}$$

In this equation, the term $\hat{g}_K$—which is a function of $\boldsymbol{p}_K$ conditioned on $\boldsymbol{p}_{K-1}$ and $\boldsymbol{p}_{K+1}$—is defined in Eq. (35), and the velocity $\boldsymbol{v}_K(\boldsymbol{p}_K)$ and its divergence $[\nabla \cdot \boldsymbol{v}_K](\boldsymbol{p}_K)$ can be estimated using Algorithm 1 from §5.1.2.

We now make an important observation that Eq. (49) can be computed in $O(N)$ time since the term $h(\boldsymbol{p}_K; \boldsymbol{p}_{K-1}, \boldsymbol{p}_{K+1})$ on the right-hand side is essentially a function of one point $\boldsymbol{p}_K$ and, thus, can be computed in $O(1)$ time.

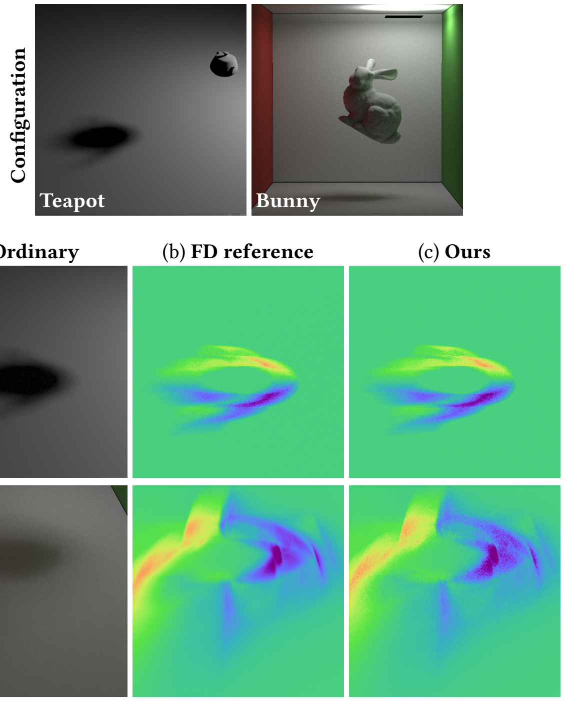


Fig. 11. We **validate** our techniques by comparing derivative images (visualized using the same color mapping as Figure 1) estimated by our method (c) to finite-difference (FD) references (b). Our "teapot" and "bunny" results are generated, respectively, using our *unidirectional* and *bidirectional* estimators. The derivatives are visualized using the same color map as Figure 1.

Based on Eq. (49), we introduce a *unidirectional* estimator that uses next-event estimation (NEE) that works as follows. When tracing a detector subpath $(\boldsymbol{p}_0^{\mathrm{D}}, \boldsymbol{p}_1^{\mathrm{D}}, \ldots)$, we maintain the prefix sum

$$H_n^{\mathrm{D}} := \sum_{m=1}^{n-1} h\left(\boldsymbol{p}_m^{\mathrm{D}}; \boldsymbol{p}_{m+1}^{\mathrm{D}}, \boldsymbol{p}_{m-1}^{\mathrm{D}}\right) = H_{n-1}^{\mathrm{D}} + h\left(\boldsymbol{p}_{n-1}^{\mathrm{D}}; \boldsymbol{p}_n^{\mathrm{D}}, \boldsymbol{p}_{n-2}^{\mathrm{D}}\right), \tag{50}$$

for each vertex $\boldsymbol{p}_n^{\mathrm{D}}$. After performing NEE at this vertex using a light sample point $\boldsymbol{p}_n^{\mathrm{S}}$, we obtain a full path $\bar{\boldsymbol{p}}_n := (\boldsymbol{p}_n^{\mathrm{S}}, \boldsymbol{p}_n^{\mathrm{D}}, \ldots, \boldsymbol{p}_1^{\mathrm{D}}, \boldsymbol{p}_0^{\mathrm{D}})$. Leveraging the prefix sum H_n, we can compute the sum of divergences $\Sigma_{\mathrm{div}}(\bar{\boldsymbol{p}}_n)$ defined in Eq. (49) using $O(1)$ time via:

$$\Sigma_{\mathrm{div}}(\bar{\boldsymbol{p}}_n) = \hat{f}(\bar{\boldsymbol{p}}_n) \left(H_n^{\mathrm{D}} + h\left(\boldsymbol{p}_n^{\mathrm{D}}; \boldsymbol{p}_n^{\mathrm{S}}, \boldsymbol{p}_{n-1}^{\mathrm{D}}\right) + h\left(\boldsymbol{p}_n^{\mathrm{S}}; \boldsymbol{p}_n^{\mathrm{D}}\right) \right), \tag{51}$$

where the material measurement contribution $\hat{f}(\bar{\boldsymbol{p}})$ is computed in $O(1)$ the same way as (forward-rendering) unidirectional path tracing (see Algorithm 2).

5.2.2 *Our bidirectional estimator.*

In addition to the *unidirectional* estimator, we also introduce a *bidirectional* one capable of handling challenging light-transport effects such as caustics.

Our *bidirectional* estimator samples paths the same way as forward-rendering bidirectional path tracing (BDPT) [Veach and Guibas 1995; Lafortune and Willems 1996]. Specifically, it generates a *source subpath* $\bar{\boldsymbol{p}}^{\mathrm{S}} = (\boldsymbol{p}_0^{\mathrm{S}}, \boldsymbol{p}_1^{\mathrm{S}}, \ldots)$ and a *detector subpath* $\bar{\boldsymbol{p}}^{\mathrm{D}} = (\boldsymbol{p}_0^{\mathrm{D}}, \boldsymbol{p}_1^{\mathrm{D}}, \ldots)$.

[3] In this paper, we focus on standard bidirectional path tracing (BDPT) that connects all pairs of vertices between the source and the detector subpaths.

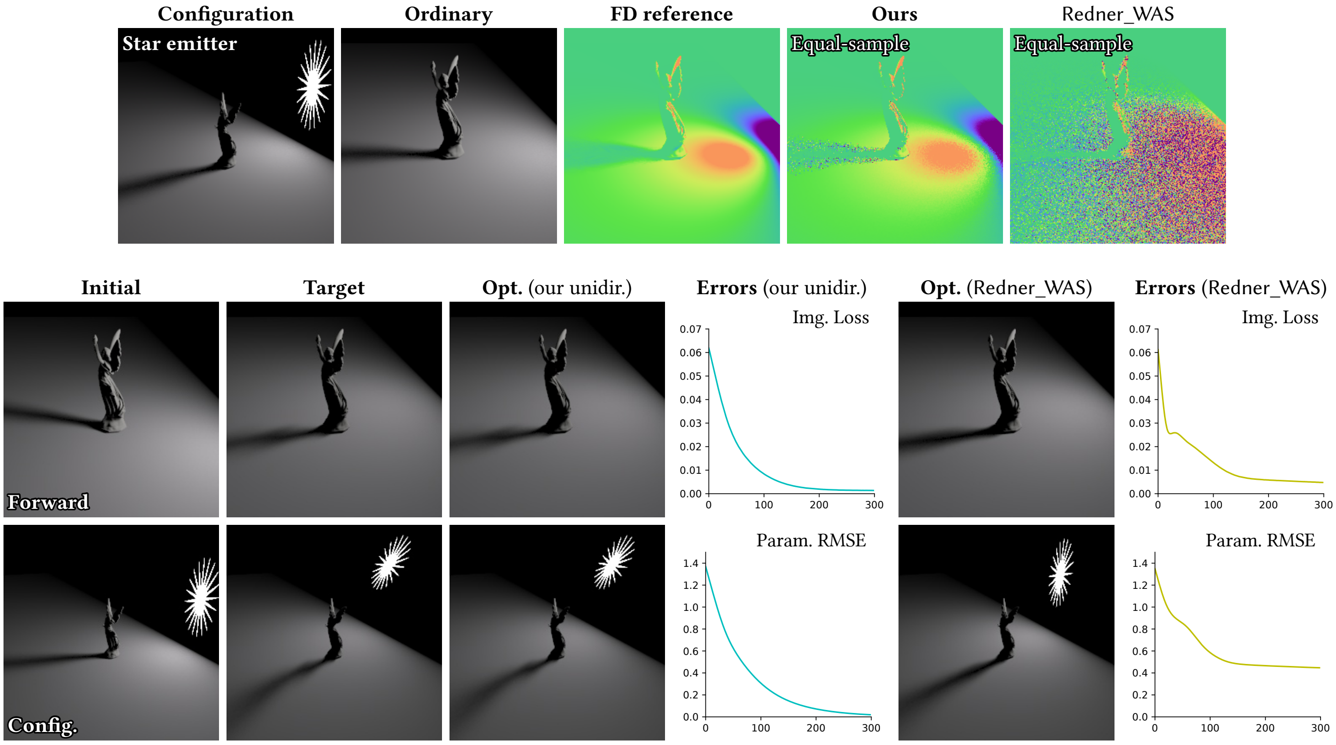


Fig. 12. **Equal-sample comparisons** between our *unidirectional* estimator with Redner_WAS [Bangaru et al. 2020]. This example involves a lucy lit by a star-shaped area light. (top) We compare derivatives with respect to the horizontal displacement of the light; (bottom) We compare inverse-rendering results where we infer the location and the pose of the light by minimizing the L^2 loss between the target and rendered images shown in the row marked as "forward". The configuration visualizations and parameter RMSE plots in the bottom row are used only for evaluation (and not for optimization).

Table 3. Performance statistics for our inverse-rendering results where "time" measures per-iteration optimization time (in seconds) on an Amazon EC2 `c6a.8xlarge` instance with 32 VCPUs. Additionally, "# img." indicates the number of images used for inverse rendering, and "# param." the number of scene parameters being optimized.

Scene		Time	# img.	# param.
Star emitter	(Fig. 12)	3.8	1	3
Chess	(Fig. 13)	4.5	40	30 000
Bunny in glass	(Fig. 14)	22.5	50	20 000
Lamp	(Fig. 15)	7.0	40	30 000
Dododo	(Fig. 16)	8.4	40	30 000
Caustics	(Fig. 17)	21.3	1	12

Then, by connecting each $\boldsymbol{p}_m^{\mathrm{S}}$ and $\boldsymbol{p}_n^{\mathrm{D}}$ (for any m and n), we obtain a complete path sample $\bar{\boldsymbol{p}}_{m,n} \coloneqq (\boldsymbol{p}_0^{\mathrm{S}}, \ldots, \boldsymbol{p}_m^{\mathrm{S}}, \boldsymbol{p}_n^{\mathrm{D}}, \ldots, \boldsymbol{p}_0^{\mathrm{D}})$.

For efficient computation of the sum $\sum_K \nabla \cdot (\hat{f}_K \boldsymbol{v}_K)$ for each path $\bar{\boldsymbol{p}}_{m,n}$, we extend the process described in §5.2.1 by maintaining two prefix sums $(H_m^{\mathrm{S}} : m = 0, 1, \ldots)$ and $(H_n^{\mathrm{D}} : n = 0, 1, \ldots)$ where the former equals

$$H_m^{\mathrm{S}} = H_{m-1}^{\mathrm{S}} + h\left(\boldsymbol{p}_{m-1}^{\mathrm{S}}; \boldsymbol{p}_{m-2}^{\mathrm{S}}, \boldsymbol{p}_m^{\mathrm{S}}\right), \tag{52}$$

and the latter follows Eq. (50).

With both pre-fix sums, we can compute the sum of divergences in Eq. (49) in $O(1)$ time using

$$\Sigma_{\text{div}}(\bar{\boldsymbol{p}}_{m,n}) = \hat{f}(\bar{\boldsymbol{p}}_{m,n}) \cdot \left(H_n^{\mathrm{D}} + H_m^{\mathrm{S}} + h\left(\boldsymbol{p}_m^{\mathrm{S}}; \boldsymbol{p}_{m-1}^{\mathrm{S}}, \boldsymbol{p}_n^{\mathrm{D}}\right) + h\left(\boldsymbol{p}_n^{\mathrm{D}}; \boldsymbol{p}_m^{\mathrm{S}}, \boldsymbol{p}_{n-1}^{\mathrm{D}}\right)\right), \tag{53}$$

resulting in the following bidirectional estimator

$$\left\langle \frac{\mathrm{d}I}{\mathrm{d}\theta} \right\rangle_{\text{bidir}} = \sum_{m,n} \frac{w_{\text{MIS}}(\bar{\boldsymbol{p}}_{m,n})}{\text{pdf}(\bar{\boldsymbol{p}}_{m,n})} \left(\frac{\mathrm{d}}{\mathrm{d}\theta} \hat{f}(\bar{\boldsymbol{p}}_{m,n}) + \Sigma_{\text{div}}(\bar{\boldsymbol{p}}_{m,n}) \right). \tag{54}$$

In this equation, for each m and n, the material measurement contribution $\hat{f}(\bar{\boldsymbol{p}}_{m,n})$, the probability density $\text{pdf}(\bar{\boldsymbol{p}}_{m,n})$, and the multiple importance sampling (MIS) weight $w_{\text{MIS}}(\bar{\boldsymbol{p}}_{m,n})$ can all be computed in $O(1)$ time (with $O(N)$ precomputations) the same way as forward-rendering BDPT.

6 RESULTS

We implement our *unidirectional* and *bidirectional* estimators (introduced in §5.2.1 and §5.2.2, respectively) based on the differentiable renderer released by Yu et al. [2022]. This system is CPU-based and uses the Enzyme automatic differentiation framework [Moses and Churavy 2020].

In what follows, we first validate our estimators using several differentiable-rendering experiments in §6.1. Then, we compare

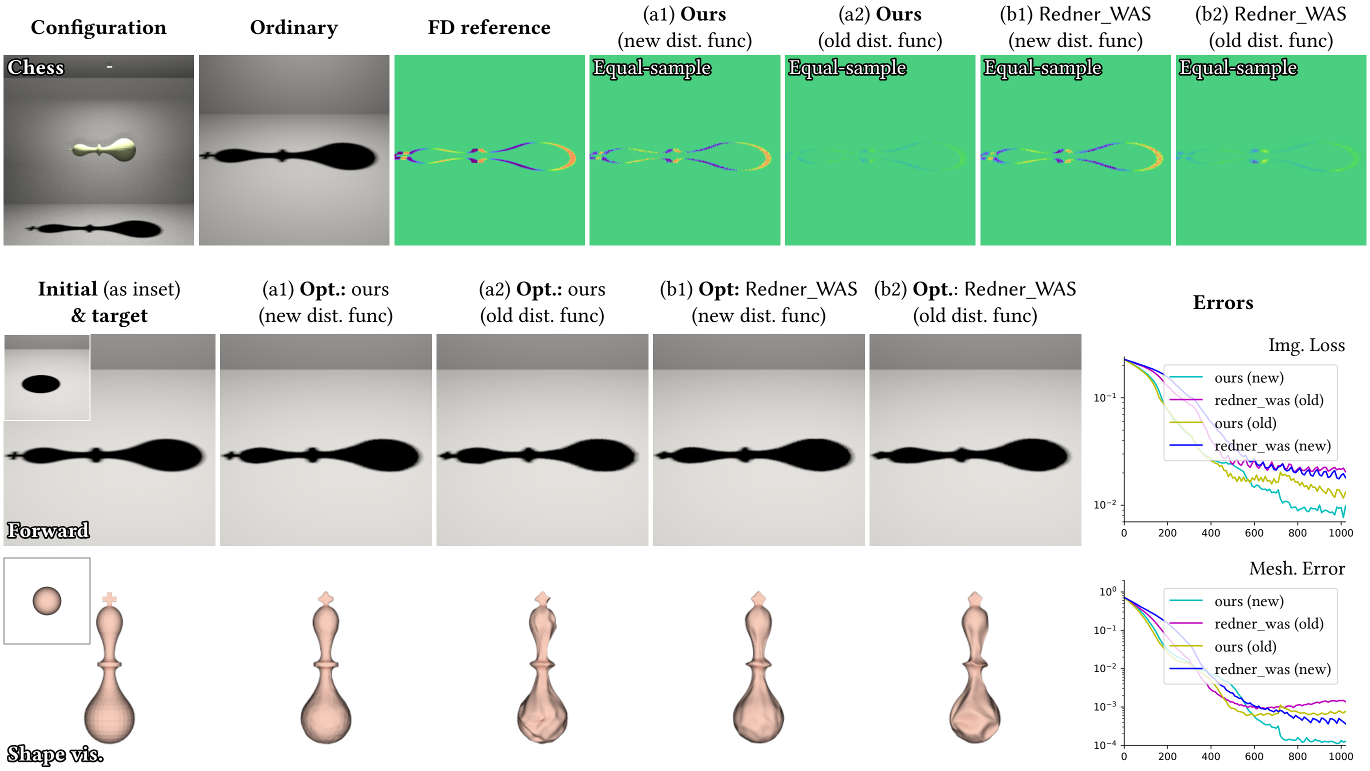


Fig. 13. We compare **differentiable- and inverse-rendering** results generated with: (a1) our *unidirectional* estimator with the new distance function in Eq. (16); (a2) our *unidirectional* estimator with Redner_WAS's distance function in Eq. (57); (b1, b2) Redner_WAS with the two distance functions, respectively. The differentiable-rendering results (top) are computed with respect to the displacement of the chess piece. For inverse rendering (bottom), we optimize the shape of the chess piece by only looking at its cast shadow on the ground. We use 40 images among which the object has varying known poses and show one in the row marked "forward". The mesh error information (plotted on the bottom right) is used for evaluation only.

our estimators to two main baselines—Bangaru et al.'s [2020] (indicated as Redner_WAS) and Zhang et al.'s [2020] (indicated as PSDR) methods—in §6.2 and §6.3, respectively. Lastly, we show additional inverse-rendering results in §6.4.

Please refer to Table 3 for performance statistics and the supplement for animated versions of our inverse-rendering results.

6.1 Validation

In Figure 11, we validate our technique by comparing derivative estimates generated using our method and finite differences (FD). The "Teapot" example shows the cast shadow of a teapot lit by an area light (with the configuration shown as inset), and the derivatives are with respect to the displacement of the teapot. The "Bunny" scene involves a diffuse bunny inside a Cornell box with an area light facing the ceiling, creating an indirect-illumination dominated configuration. For this example, we differentiate the ordinary rendering (a) with respect to the displacement of the bunny. In both examples, the derivative estimates produced by our estimators (c) closely match the FD references (b).

6.2 Comparisons with Redner_WAS

As discussed in Appendix A, although both our *unidirectional* estimator and Bangaru et al.'s [2020] method (indicated as Redner_WAS) apply unidirectional path tracing, the two methods differ significantly. We now demonstrate the practical advantages of our *unidirectional* estimator over Redner_WAS using several differentiable- and inverse-rendering examples.

We conduct all comparisons with Redner_WAS using *equal sample* (instead of equal time) due to the significant differences between the two codebases.

Material form. The first advantage of our estimator is the use of the material-form reparameterization (described in §3.1). This allows our warped-area reparameterization to handle fewer types of discontinuities compared with Redner_WAS, leading to overall less variance and bias.

To demonstrate this, we compare derivatives estimated with our *unidirectional* estimator and Redner_WAS on the top of Figure 12. This example uses a "Star emitter" scene involving a Lucy lit by a star-shaped area emitter under direct illumination. We estimate

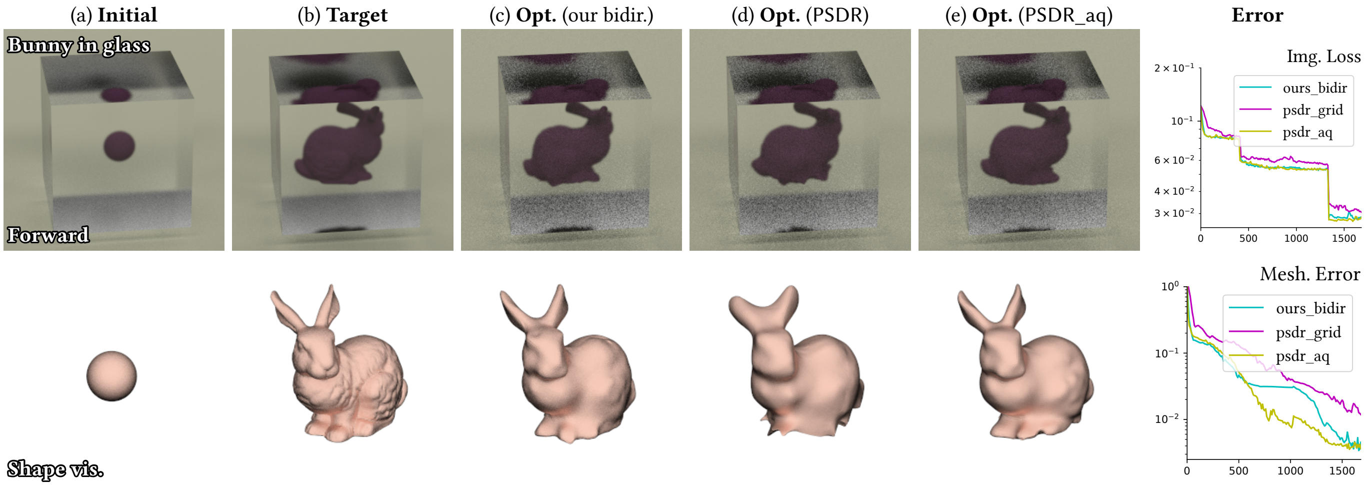


Fig. 14. We compare **inverse-rendering** performance of *bidirectional* estimators from our technique (c), PSDR [Zhang et al. 2020] (d), and PSDR_aq [Yan et al. 2022] (e). In this example, we optimize the shape of a diffuse object inside a rough-glass cube using 50 multi-view images (with one shown on the top). We use a multi-stage setup for the optimization, where each stage uses varying sample counts and learning rates. We visualize the initial, target, and optimized shapes on the bottom. Without guiding, our method outperforms PSDR and offers a similar level of performance as PSDR_aq (at equal time). The mesh error information (plotted on the bottom right) is used for evaluation only.

derivatives with respect to the horizontal displacement of the emitter. At equal sample, our *unidirectional* estimator produces derivatives with significantly less variance. This is becuase topological boundaries of the emitter—namely the edges of the star shape—do not need to be reparameterized by our method since they are fixed under the reference configuration. Redner_WAS, on the other hand, has to reparameterize these boundaries and, thus, produces higher variance.

Based on this configuration, we further show an inverse-rendering comparison on the bottom of Figure 12. For this example, we optimize the position and orientation of the star-shaped emitter without directly looking at it. Under identical optimization settings (including initializations and learning rates), our *unidirectional* estimator allows the optimization to converge nicely to the groundtruth. At equal sample, Redner_WAS suffers from much higher variance and causes the optimization to stuck.

Distance function. Another difference between our *unidirectional* and Redner_WAS—as detailed in Appendix A—is the choice of distance function D expressed in Eq. (16).

We demonstrate the effectiveness of our new distance function using differentiable- and inverse-rendering comparisons in Figure 13. This figure uses a "Chess" scene where a chess piece is lit by a small area light, resulting in a cast shadow on the ground.

The differentiable-rendering results indicate that, at equal time, our new distance function is crucial for fast convergence of derivative estimates.

For the inverse-rendering comparison, we use multiple shadow images with the object having varying known poses (with one shown in the figure) and optimize the shape (expressed as per-vertex positions) of the object. With our "new" distance function defined in Eq. (16), both our *unidirectional* estimator and Redner_WAS lead to reasonable reconstruction results. Our method slightly outperforms Redner_WAS in this case due to the use of material-form reparameterization. In contrast, with the "old" distance function introduced by Bangaru et al. [2020], both methods produce severely biased gradient estimates, yielding low-quality shape reconstructions.

6.3 Comparisons with PSDR

We now compare our *bidirectional* estimator with the bidirectional one introduced by Zhang et al. [2020] (indicated as PSDR) as well as an improved variant (indicated as PSDR_aq) by Yan et al. [2022]. When estimating the material-form differential path integrals, both methods handle the *boundary* component by directly sampling material boundary paths. To improve efficiency, both PSDR and PSDR_aq perform primary-sample-space guiding where the former relies on regular grids and the latter leverages adaptive gridding. For all comparisons, we set the sample counts for all methods so that they take approximately equal time to estimate the gradients.

In Figure 14, we show a "bunny in glass" scene that was originally modeled by Yan et al. [2022] and contains a diffuse bunny inside a rough-glass cube. Using 50 multi-view images, we optimize the shape of the bunny. Further, we configure all optimizations to use identical initializations (i.e., a sphere) and learning rates as well as approximately equal time per iteration.

The complexity of this scene causes the optimization using PSDR to converge slowly (as shown in Figure 14-d). On the other hand, PSDR_aq's guiding improves the rate of convergence considerably (see Figure 14-e). *Without using any precomputation or guiding*, our method is capable of offering a similar level of performance as PSDR_aq (see Figure 14-b).

Figure 15 shows another comparison with PSDR_aq. This example uses a "Lamp" scene where an object is lit by a desk lamp and

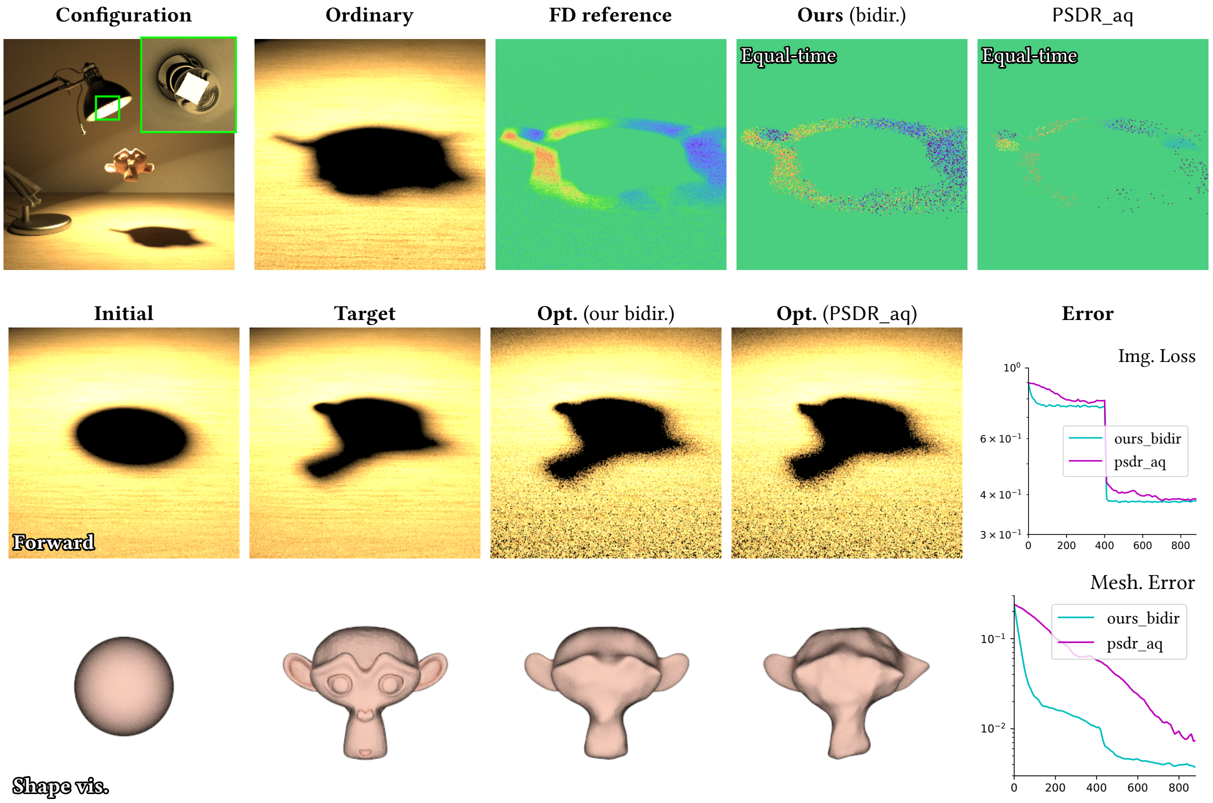


Fig. 15. We compare **differentiable- and inverse-rendering** performance of *bidirectional* estimators given by our technique and PSDR_aq [Yan et al. 2022]. The differentiable-rendering results (top) are computed with respect to the horizontal displacement of the object. For inverse rendering (bottom), we optimize the shape of the occluder by only looking at its cast shadow. We use 40 shadow images with the occluder having varying known poses (with one image shown). Without guiding, our method outperforms PSDR_aq at equal time. The mesh error information (plotted on the bottom right) is used for evaluation only.

casts a shadow on the desktop. Further, the area emitter is encapsulated within a glass bulb with low surface roughness, creating a very challenging situation for primary-sample-space guiding (since the high-value region is extremely small). This greatly reduces the effectiveness of guiding, causing PSDR_aq to produce more noisy derivative estimates, as demonstrated by the differentiable-rendering results (shown on the top of the figure).

For the inverse-rendering comparison (shown on the bottom), we use 40 images of the cast shadow with the object having multiple known poses (with one shown in the figure) and optimize the shape of the object. Using identical optimization configurations and equal time, our method outperforms PSDR_aq without the need of precomputation or guiding.

6.4 Additional Inverse-Rendering Results

We now show additional synthetic inverse-rendering results to demonstrate the usefulness of our *bidirectional* estimator.

Figure 16 uses a “Dodoco” scene that contains a diffuse object inside the Cornell box. The object is lit indirectly by a small area light facing the ceiling (similar to the “Bunny” scene in Figure 11). Using 40 images (with one shown in the figure), we optimize the shape of the object.

Additionally, Figure 17 use a “Caustics” scene with a glass bunny inside the Cornell box. We optimize the vertex positions of a cubical control cage by only looking at the ground.

Since both scenes involve complex light transport effects, Redner_WAS and our *unidirectional* estimator—both of which rely on unidirectional path tracing—perform poorly. On the contrary, our *bidirectional* estimator—which is enabled by our reparameterized differential path integral of Eq. (38)—offers the robustness and efficiency to simulate light transport in both scenes, allowing the optimizations to converge nicely.

7 DISCUSSION AND CONCLUSION

Limitations and future work. Our reparameterized differential path integral neglects volumetric light transport. Generalizing it to reparameterize Zhang et al.’s [2021b] generalized differential path integrals is an important future topic.

Also, our Monte Carlo estimators still mostly rely on unidirectional and bidirectional path tracing—techniques developed for forward rendering—to sample light paths. In the future, more efficient sampling schemes using, for instance, guiding, can be helpful.

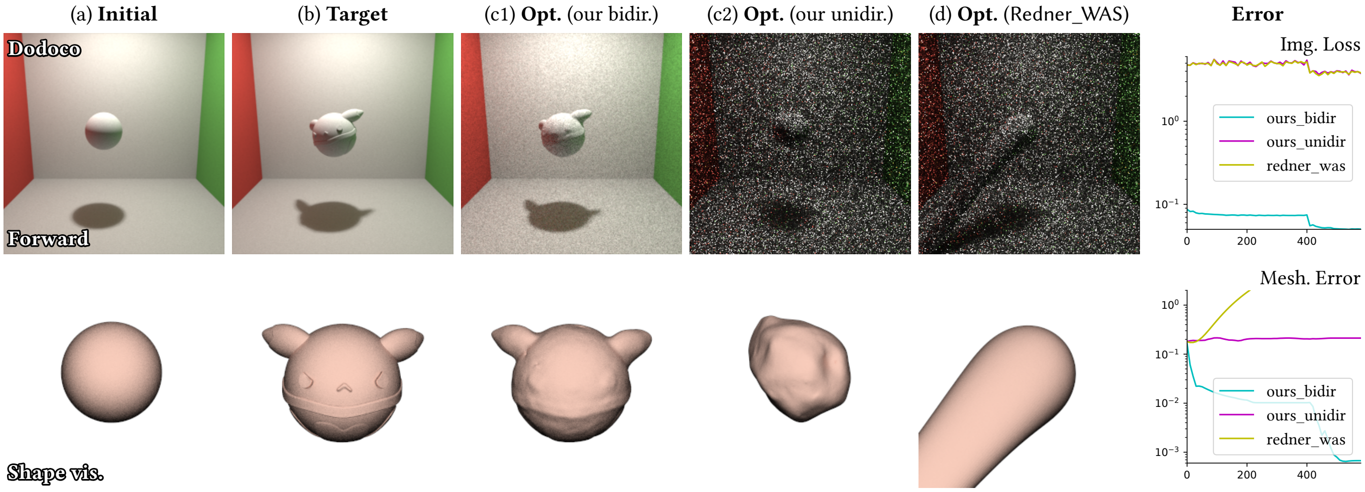


Fig. 16. We compare **inverse-rendering** performance of our *bidirectional* and *unidirectional* estimators as well as Redner_WAS [Bangaru et al. 2020]. To ensure fairness, we use equal time between our *bidirectional* and *unidirectional* estimators, and equal sample between our *unidirectional* estimator and Redner_WAS.

Lastly, adapting our method to support differentiable rendering of implicit geometries [Vicini et al. 2022; Bangaru et al. 2022] is worth investigating.

Conclusion. In this paper, we bridge the gap in physics-based differentiable rendering techniques between the warped-area sampling methods and the path-space methods, by introducing the formulation of reparameterized differential path integral to reparameterize boundary components of material-form differential path integrals. Our resulting formulation enjoys the advantages of both differential path integrals—by enabling advanced Monte Carlo estimators (such as bidirectional path tracing)—and warped-area reparameterization—by not requiring explicit handling of discontinuity boundaries.

We validated our technique and evaluated its effectiveness using several differentiable rendering and inverse rendering examples.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their constructive comments and Guangyan Cai, Wesley Chang, and Cheng Zhang for proofreading the paper. This project was partially funded by NSF grants 1900927 and 2105806.

REFERENCES

Dejan Azinović, Tzu-Mao Li, Anton Kaplanyan, and Matthias Nießner. 2019. Inverse Path Tracing for Joint Material and Lighting Estimation. In *Computer Vision and Pattern Recognition.*

Sai Bangaru, Michael Gharbi, Tzu-Mao Li, Fujun Luan, Kalyan Sunkavalli, Milos Hasan, Sai Bi, Zexiang Xu, Gilbert Bernstein, and Fredo Durand. 2022. Differentiable Rendering of Neural SDFs through Reparameterization. In *ACM SIGGRAPH Asia 2022 Conference Proceedings* (Daegu, Republic of Korea) *(SIGGRAPH Asia '22).* Association for Computing Machinery, New York, NY, USA, Article 22, 9 pages. https://doi.org/10.1145/3550469.3555397

Sai Praveen Bangaru, Tzu-Mao Li, and Frédo Durand. 2020. Unbiased Warped-Area Sampling for Differentiable Rendering. *ACM Trans. Graph.* 39, 6 (2020), 245:1–245:18.

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

Chengqian Che, Fujun Luan, Shuang Zhao, Kavita Bala, and Ioannis Gkioulekas. 2020. Towards Learning-based Inverse Subsurface Scattering. In *2020 IEEE International Conference on Computational Photography (ICCP).* IEEE, 1–12.

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

James T. Kajiya. 1986. The Rendering Equation. In *SIGGRAPH '86.* 143–150.

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

Tzu-Mao Li. 2023. redner: Differentiable rendering without approximation. https://github.com/BachiLi/redner.

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

Guillaume Loubet, Nicolas Holzschuch, and Wenzel Jakob. 2019. Reparameterizing discontinuous integrands for differentiable rendering. *ACM Trans. Graph.* 38, 6 (2019), 228:1–228:14.

Fujun Luan, Shuang Zhao, Kavita Bala, and Zhao Dong. 2021. Unified Shape and SVBRDF Recovery using Differentiable Monte Carlo Rendering. *Computer Graphics Forum* 40, 4 (2021), 101–113.

William Moses and Valentin Churavy. 2020. Instead of Rewriting Foreign Code for Machine Learning, Automatically Synthesize Fast Gradients. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 12472–12485.

Merlin Nimier-David, Thomas Müller, Alexander Keller, and Wenzel Jakob. 2022. Unbiased Inverse Volume Rendering with Differential Trackers. *ACM Trans. Graph.* 41, 4 (2022), 44:1–44:20.

Merlin Nimier-David, Sébastien Speierer, Benoît Ruiz, and Wenzel Jakob. 2020. Radiative backpropagation: an adjoint method for lightning-fast differentiable rendering. *ACM Trans. Graph.* 39, 4 (2020), 146:1–146:15.

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

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.

Delio Vicini, Sébastien Speierer, and Wenzel Jakob. 2021. Path Replay Backpropagation: Differentiating Light Paths Using Constant Memory and Linear Time. *ACM Trans. Graph.* 40, 4, Article 108 (2021), 108:1–108:14 pages.

Delio Vicini, Sébastien Speierer, and Wenzel Jakob. 2022. Differentiable Signed Distance Function Rendering. *ACM Trans. Graph.* 41, 4 (2022), 125:1–125:18.

Kai Yan, Christoph Lassner, Brian Budge, Zhao Dong, and Shuang Zhao. 2022. Efficient estimation of boundary integrals for path-space differentiable rendering. *ACM Trans. Graph.* 41, 4 (2022), 123:1–123:13.

Zihan Yu, Cheng Zhang, Derek Nowrouzezahrai, Zhao Dong, and Shuang Zhao. 2022. Efficient Differentiation of Pixel Reconstruction Filters for Path-Space Differentiable Rendering. *ACM Trans. Graph.* 41, 6 (2022), 191:1–191:16.

Tizian Zeltner, Sébastien Speierer, Iliyan Georgiev, and Wenzel Jakob. 2021. Monte Carlo estimators for differential light transport. *ACM Trans. Graph.* 40, 4 (2021), 78:1–78:16.

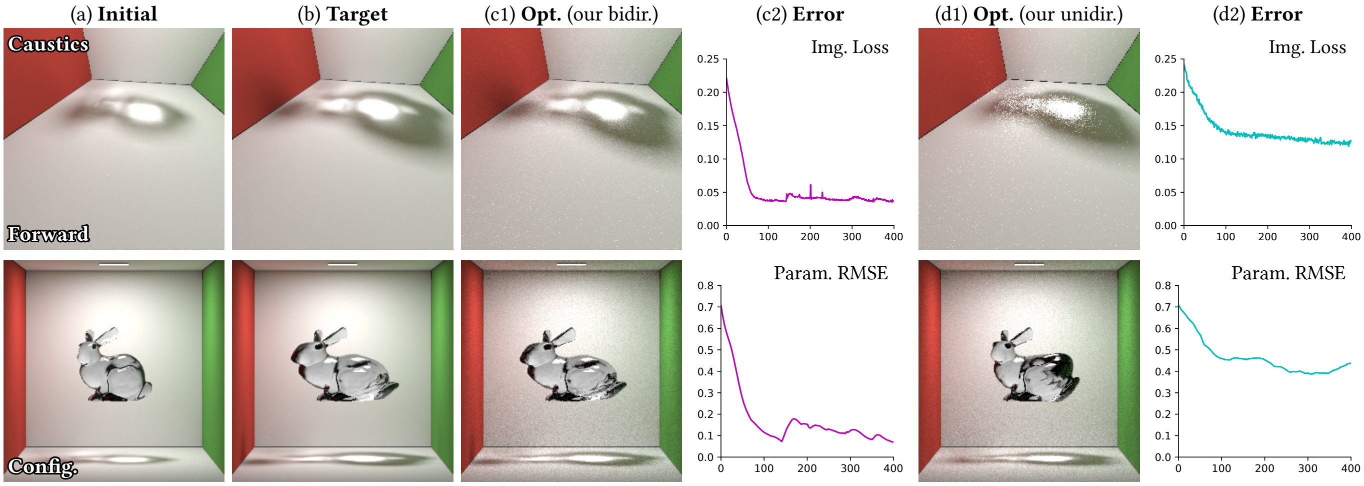


Fig. 17. We compare **inverse-rendering** performance between our *unidirectional* and *bidirectional* estimators. We adjust sample counts so that both estimators take approximately equal time per iteration.

Cheng Zhang, Zhao Dong, Michael Doggett, and Shuang Zhao. 2021a. Antithetic sampling for Monte Carlo differentiable rendering. *ACM Trans. Graph.* 40, 4 (2021), 77:1–77:12.

Cheng Zhang, Bailey Miller, Kai Yan, Ioannis Gkioulekas, and Shuang Zhao. 2020. Path-space differentiable rendering. *ACM Trans. Graph.* 39, 4 (2020), 143:1–143:19.

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

Cheng Zhang, Zihan Yu, and Shuang Zhao. 2021b. Path-space differentiable rendering of participating media. *ACM Trans. Graph.* 40, 4 (2021), 76:1–76:15.

Shuang Zhao, Wenzel Jakob, and Tzu-Mao Li. 2020. Physics-Based Differentiable Rendering: A Comprehensive Introduction. In *ACM SIGGRAPH 2020 Courses*. Article 14, 14:1–14:30 pages.

A RELATION WITH ORIGINAL WARPED-AREA REPARAMETERIZATION

In the following, we discuss the relation between our *unidirectional* method and Redner_WAS—the original warped-area technique introduced by Bangaru et al. [2020]. For simplicity, we use the one-bounce setting discussed in §3.2.

When expressing one-bounce scattered radiance leaving a surface point $\boldsymbol{y}$ in the direction $\boldsymbol{\omega}_\mathrm{o}$, Redner_WAS uses the spherical-integral formulation:

$$I_\mathrm{dir} = \int_{\mathbb{S}^2} \underbrace{L_\mathrm{e,i}(\boldsymbol{y}, \boldsymbol{\omega}_\mathrm{i})\, f_\mathrm{s}(\boldsymbol{y}, \boldsymbol{\omega}_\mathrm{i}, \boldsymbol{\omega}_\mathrm{o})\, |\boldsymbol{n}(\boldsymbol{y}) \cdot \boldsymbol{\omega}_\mathrm{i}|}_{=: F^\sigma_\mathrm{dir}(\boldsymbol{\omega}_\mathrm{i})} \,\mathrm{d}\sigma(\boldsymbol{\omega}_\mathrm{i}), \tag{55}$$

where $L_\mathrm{e,i}(\boldsymbol{y}, \boldsymbol{\omega}_\mathrm{i})$ indicates emitted radiance entering $\boldsymbol{y}$ from the incident direction $\boldsymbol{\omega}_\mathrm{i}$, $\boldsymbol{n}(\boldsymbol{y})$ denotes the (unit-length) surface normal at $\boldsymbol{y}$, and σ is the solid-angle measure.

Differentiating Eq. (55) with respect to some scene parameter θ yields

$$\frac{\mathrm{d}I_\mathrm{dir}}{\mathrm{d}\theta} = \overset{\text{interior}}{\boxed{\int_{\mathbb{S}^2} \frac{\mathrm{d}F^\sigma_\mathrm{dir}(\boldsymbol{\omega}_\mathrm{i})}{\mathrm{d}\theta}\,\mathrm{d}\sigma(\boldsymbol{\omega}_\mathrm{i})}} + \overset{\text{boundary}}{\boxed{\int_{\Delta\mathbb{S}^2} \Delta F^\sigma_\mathrm{dir}(\boldsymbol{\omega}_\mathrm{i})\, V(\boldsymbol{\omega}_\mathrm{i})\,\mathrm{d}\ell(\boldsymbol{\omega}_\mathrm{i})}}, \tag{56}$$

where:

- F^σ_dir is the integrand defined in Eq. (55);
- $\Delta\mathbb{S}^2$ are *spherical curves* comprising jump discontinuities of F^σ_dir;
- $V(\boldsymbol{\omega}_\mathrm{i}) = \boldsymbol{n}^\partial(\boldsymbol{\omega}_\mathrm{i}) \cdot \frac{\mathrm{d}\boldsymbol{\omega}_\mathrm{i}}{\mathrm{d}\theta}$ is the scalar velocity capturing how fast the curves $\Delta\mathbb{S}^2$ evolves with respect to the parameter θ.

Mathematically, Redner_WAS reparameterizes the *boundary* component of Eq. (56) that differs from our *boundary* integral in Eq. (4) in several significant ways as follows.

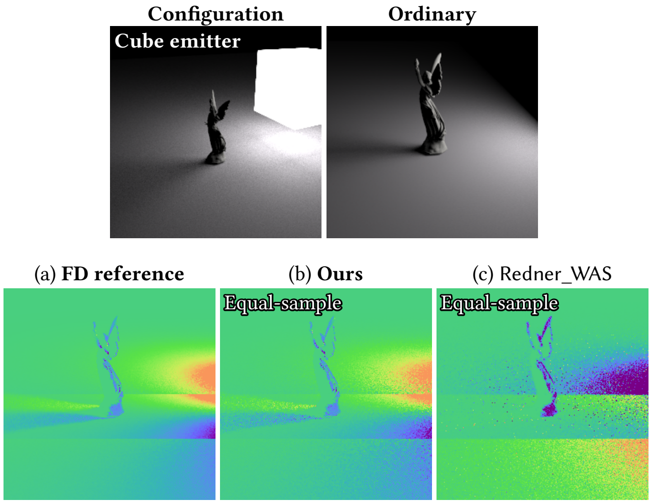
First, since the material-form reparameterization (§3.1) is not used in Eq. (56), the discontinuity curves $\Delta\mathbb{S}^2$ need to include topological boundaries and discontinuities of surface normals (see Figure 18) of the emitter surface. As demonstrated in Fig. 12, this makes their method much less efficient than ours when handling emitters with complex geometries.

Second, the scalar velocity $V(\boldsymbol{\omega}_\mathrm{i})$ is computed differently: (i) the curve normal $\boldsymbol{n}^\partial(\boldsymbol{\omega}_\mathrm{i})$ is defined with respect to spherical curves over $\mathbb{S}^2$ (as opposed to general curves over the reference surface $\mathcal{B}_\mathrm{e}$); and (ii) the velocity $\mathrm{d}\boldsymbol{\omega}_\mathrm{i}/\mathrm{d}\theta$ is computed with respect to direction $\boldsymbol{\omega}_\mathrm{i}$ and without differentiating the deformation's inverse $\mathsf{X}^{-1}(\cdot, \theta)$ as in Eq. (8).

At a more technical level, Redner_WAS uses a different distance function given by:

$$\mathrm{D}^\mathrm{was}\left(\boldsymbol{\omega}'_\mathrm{i}; \boldsymbol{\omega}_\mathrm{i}\right) \coloneqq \exp\left(1 - \left\langle \boldsymbol{\omega}_\mathrm{i}, \boldsymbol{\omega}'_\mathrm{i} \right\rangle\right) - 1. \tag{57}$$

When handling challenging scenes containing complicated geometries, small area lights, or intersecting discontinuity boundaries (with different velocities), the convergence of the consistent estimator $\langle \boldsymbol{v}_K \rangle$ of the continuous velocity $\boldsymbol{v}_K$ in Eq. (47) is known to be sensitive to the choice of distance functions [Bangaru et al. 2022]. In practice, our distance function expressed in Eq. (16) behaves well in all our experiments (including the example shown in Figure 13) with $N_{\boldsymbol{q}} = 8$ auxiliary samples. The one in Eq. (57) used by Redner_WAS, on the other hand, can lead to convergence issues even with significantly greater $N_{\boldsymbol{q}}$.



(a) **FD reference** (b) **Ours** (c) Redner_WAS

Fig. 18. **Equal-sample comparison** between our *unidirectional* estimator and Redner_WAS [Bangaru et al. 2020]. This example uses a scene with a lucy model lit by a cube-shaped area light. The formulation used by Redner_WAS requires the cube edges (where surface normals are discontinuous) to be handled by warped-area reparameterization. However, the public implementation of Redner_WAS [Li 2023] neglects this case and, thus, produces noisy and *highly biased* gradients (c). Our technique uses the material-form reparameterization (§3.1) and does not need to reparameterize the cube edges (as they are fixed on the reference surface). This allows our estimator to produce clean gradients (b) that closely match the reference (a).

B MONTE CARLO ESTIMATION OF PER-VERTEX DIVERGENCE

Provided a material path $\bar{\boldsymbol{p}} = (\boldsymbol{p}_0, \ldots, \boldsymbol{p}_N)$ and an index $0 \le K < N$, we now descirbe the estimation of the divergence $[\nabla \cdot \boldsymbol{v}_K](\boldsymbol{p}_K)$ of the continuous velocity $\boldsymbol{v}_K$ defined in Eq. (41).

Estimating ∇W and $\nabla \cdot \tilde{\boldsymbol{v}}_K$. It is easy verify that our kernel w defined in Eq. (15) and the discontinuous vector field $\boldsymbol{v}_K^{\mathrm{dis}}$ are differentiable everywhere except at their zero-measure jump discontinuities (that are fixed with respect to $\boldsymbol{p}_K$). Thus, provided Eqs. (42) and (43), we have

$$[\nabla W]\,(\boldsymbol{p}_K) = \int_{\mathcal{B}} \nabla \mathrm{w} \,\mathrm{d}A, \tag{58}$$

$$[\nabla \cdot \tilde{\boldsymbol{v}}_K]\,(\boldsymbol{p}_K) = \int_{\mathcal{B}} \left((\nabla \mathrm{w}) \cdot \boldsymbol{v}_K^{\mathrm{dis}} + \mathrm{w} \left(\nabla \cdot \boldsymbol{v}_K^{\mathrm{dis}} \right) \right) \mathrm{d}A. \tag{59}$$

Using the point-sampling process described in §5.1.2, we estimate ∇W and $\nabla \cdot \tilde{\boldsymbol{v}}_K$ in an unbiased and consistent fashion via

$$\left\langle [\nabla W]\,(\boldsymbol{p}_K) \right\rangle = \sum_{i=1}^{N_q} \sum_j \frac{\nabla \mathrm{w}(\boldsymbol{q}_{ij}; \boldsymbol{p}_K)}{\mathrm{pdf}(\boldsymbol{q}_{ij})}, \tag{60}$$

and

$$\left\langle [\nabla \cdot \tilde{\boldsymbol{v}}_K]\,(\boldsymbol{p}_K) \right\rangle = \sum_{i=1}^{N_q} \sum_j \frac{\nabla \mathrm{w}(\boldsymbol{q}_{ij}; \boldsymbol{p}_K) \cdot \boldsymbol{v}_K^{\mathrm{dis}}(\boldsymbol{q}_{ij}) + \mathrm{w}(\boldsymbol{q}_{ij}; \boldsymbol{p}_K) \left[\nabla \cdot \boldsymbol{v}_K^{\mathrm{dis}} \right] (\boldsymbol{q}_{ij})}{\mathrm{pdf}(\boldsymbol{q}_{ij})}, \tag{61}$$

where the gradient $\nabla \mathrm{w}$ of the kernel w and the divergence $\nabla \cdot \boldsymbol{v}_K^{\mathrm{dis}}$ of the discontinuous velocity $\boldsymbol{v}_K^{\mathrm{dis}}$ can both be computed using automatic differentiation.

Estimating $\nabla \cdot \boldsymbol{v}_K$. Finally, because

$$\nabla \cdot \boldsymbol{v}_K = (\nabla \cdot \tilde{\boldsymbol{v}}_K)/W - ((\nabla W) \cdot \tilde{\boldsymbol{v}}_K)/W^2,$$

we obtain a biased-but-consistent estimator of the divergence $\nabla \cdot \boldsymbol{v}_K$ using Eqs. (60) and (61):

$$\left\langle [\nabla \cdot \boldsymbol{v}_K](\boldsymbol{p}_K) \right\rangle = \frac{\left\langle [\nabla \cdot \tilde{\boldsymbol{v}}_K]\,(\boldsymbol{p}_K) \right\rangle}{\left\langle W(\boldsymbol{p}_K) \right\rangle} - \frac{\left\langle [\nabla W]\,(\boldsymbol{p}_K) \right\rangle \cdot \left\langle \tilde{\boldsymbol{v}}_K(\boldsymbol{p}_K) \right\rangle}{\left\langle W(\boldsymbol{p}_K) \right\rangle^2}, \tag{62}$$

where $\{\boldsymbol{q}_{ij}\}$ are samples of $\boldsymbol{q}$ drawn with the procedure explained in §5.1.2.