Efficient Differentiation of Pixel Reconstruction Filters for Path-Space Differentiable Rendering

ZIHAN YU, University of California, Irvine, USA
CHENG ZHANG, University of California, Irvine, USA and Meta Reality Labs, USA
DEREK NOWROUZEZAHRAI, McGill University, Canada
ZHAO DONG, Meta Reality Labs, USA
SHUANG ZHAO, University of California, Irvine, USA

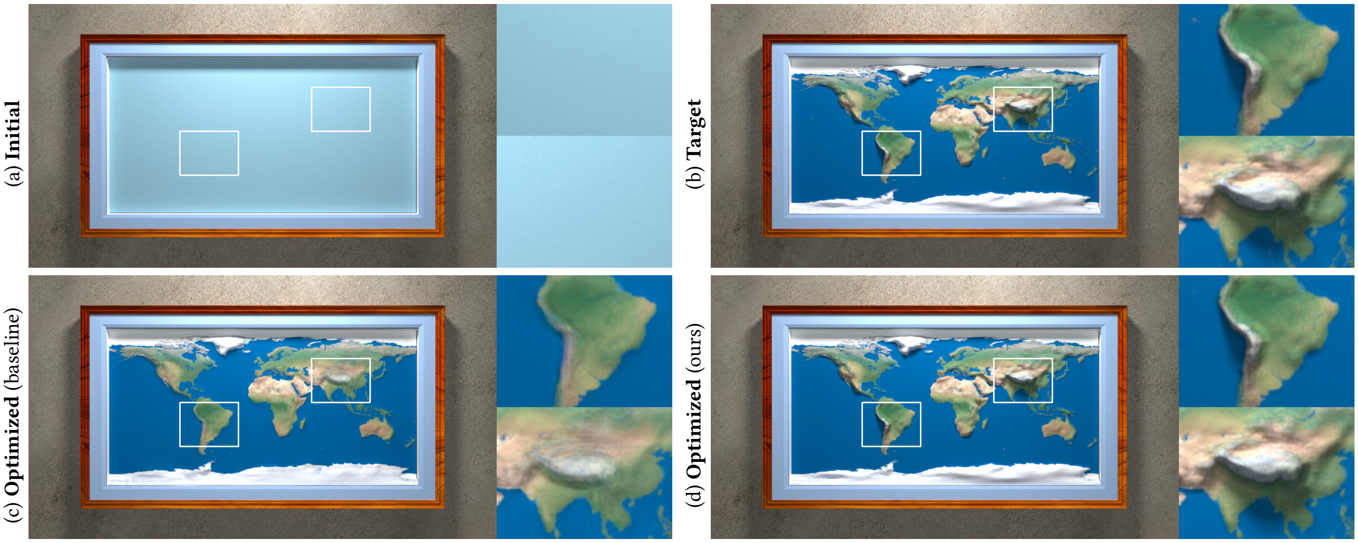


Fig. 1. We accurately and efficiently differentiate pixel reconstruction filters in the path-space formulation [Zhang et al. 2020, 2021b]. Our sampling method is applicable to challenging inverse-rendering problems, with a variety of pixel reconstruction filters. Above, we optimize the world map relief with multiple target images using the tent filter: initialized with a flat shape with constant reflectance (a), the inverse-rendering optimization with our technique (d) successfully recovers the detailed target geometry and texture (b). Without our technique, on the contrary, the optimization fails to recover the geometric details (c) due to high-variance gradient estimates. *(Please use Acrobat and click on (d) to see the optimization process animated.)*

Pixel reconstruction filters play an important role in physics-based rendering and have been thoroughly studied. In physics-based differentiable rendering, however, the proper treatment of pixel filters remains largely under-explored. We present a new technique to efficiently differentiate pixel reconstruction filters based on the path-space formulation. Specifically, we formulate the pixel *boundary* integral that models discontinuities in pixel filters and introduce new antithetic sampling methods that support differentiable path sampling methods, such as adjoint particle tracing and bidirectional path tracing. We demonstrate both the need and efficacy of antithetic sampling when estimating this integral, and we evaluate its effectiveness across several differentiable- and inverse-rendering settings.

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

Additional Key Words and Phrases: Differentiable rendering, differential path integral, pixel reconstruction filter

ACM Reference Format:
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, Article 191 (December 2022), 16 pages. https://doi.org/10.1145/3550454.3555500

1 INTRODUCTION

Pixel intensities in physics-based rendering are not typically formulated as individual radiance values but rather image-plane integrals of the radiance field, modulated by pixel *reconstruction filters*. This mimics physical camera sensors, allowing alias-free rendered images (i.e., with proper sampling rates) and complex sensor effects.

Consequently, the treatment of pixel reconstruction filters has been an essential component of forward rendering. Unidirectional path tracing initiates light paths by importance sampling “camera

Authors’ addresses: Zihan Yu, zihay19@uci.edu, University of California, Irvine, USA; Cheng Zhang, zhangchengee@gmail.com, University of California, Irvine, USA and Meta Reality Labs, USA; Derek Nowrouzezahrai, derek.nowrouzezahrai@mcgill.ca, McGill University, Canada; Zhao Dong, zhaodong@fb.com, Meta Reality Labs, USA; Shuang Zhao, shz@ics.uci.edu, University of California, Irvine, USA.

0730-0301/2022/12-ART191
https://doi.org/10.1145/3550454.3555500

rays" according the reconstruction filters. Adjoint methods, on the other hand, trace light transport paths from emitters and evaluate pixel filters when directly connecting a path vertex to the camera with next-event estimation. Bidirectional methods such as bidirectional path tracing (BDPT) leverage both aforementioned strategies.

In physics-based *differentiable* rendering, the treatment of pixel reconstruction filters has typically been approached by repurposing techniques developed in forward rendering. Unfortunately, while adequate for static geometry, this can lead to high variance in derivative estimates (w.r.t. changes in geometry) with dynamic geometry.

Our work systematically studies the treatment of pixel reconstruction filters in differentiable rendering, and under the differentiable path-space paradigm [Zhang et al. 2020, 2021b]. We present a path-space method that formulates the derivative of a pixel's intensity as the sum of *interior* and *boundary* path integrals. We focus on the ubiquitous perspective pinhole camera model and develop new Monte Carlo methods to efficiently handle derivatives and discontinuities emerging from pixel reconstruction filters, e.g., due to geometric changes in the scene. Our technique is general, supporting a wide array of pixel filters (e.g., tent, box, anisotropic Gaussian), and this under both surface- and volume-based light transport.

Concretely, our contributions include:

- a new antithetic sampling method to efficiently estimate *interior* integrals (§5), applicable to primal and adjoint path sampling (i.e., tracing light transport paths from cameras and emitters),
- a new pixel *boundary* integral that explicitly treats reconstruction filters (e.g., box) with jump discontinuities (§6), and
- efficient Monte Carlo sampling methods to estimate this integral.

We demonstrate the effectiveness of our method, comparing gradient image estimates with and without our sampling strategy in Figures 12 and 13. Moreover, we compare inverse-rendering performance using synthetic examples in Figures 1, 14 and 15.

2 RELATED WORK

We review works in forward- and differentiable-rendering related to pixel reconstruction and path-space differentiable rendering.

BSDF sampling. Most physics-based forward rendering methods rely on Monte Carlo light path sampling. *Local sampling* methods, e.g., unidirectional path tracing and adjoint particle tracing, construct light paths incrementally by importance sampling vertices according to local bidirectional scattering distribution functions (BSDFs). Many suitable BSDF models exist, including widely-adopted microfacet BSDFs (e.g., [Cook and Torrance 1982; Ward 1992; Schlick 1994; van Ginneken et al. 1998; Kelemen and Szirmay-Kalos 2001; Walter et al. 2007; Heitz et al. 2016; Bagher et al. 2016; Lee et al. 2018; Xie and Hanrahan 2018]) with effective importance sampling schemes [Walter et al. 2007; Heitz and d'Eon 2014; Heitz et al. 2016].

Adaptive sampling and reconstruction. Many adaptive sampling method have been developed to reduce variance in Monte Carlo forward rendering. By analyzing light transport processes, these techniques can reason about per-pixel sampling rates and reconstruction filters suited to the underlying dynamics, e.g., using local frequency analyses [Durand et al. 2005; Egan et al. 2009; Soler et al. 2009; Bagher et al. 2012; Mehta et al. 2012] or covariance matrices [Belcour et al. 2013, 2014, 2017]. We refer readers to a comprehensive survey of these methods [Zwicker et al. 2015]. Our work is complementary to these approaches and, in principle, can be applied to differentiate reconstruction filters obtained in these methods.

Physics-based differentiable rendering. With numerous applications in computer graphics, vision, and computational imaging, physics-based differentiable rendering treats the numerical estimation of derivatives of forward-rendering results (or rendering losses) with respect to arbitrary scene parameters, such as object geometries and/or optical material properties. Generally, physics-based differentiable rendering amounts to estimating (i) *interior* integrals given by differentiating the integrands of corresponding forward-rendering integrals; and (ii) *boundary* ones defined over discontinuities in those integrands.

The *interior* integrals are usually estimated by repurposing forward rendering sampling strategies. Zeltner et al. [2021] have studied how various parameterizations (e.g., "attached" and "detached") affect the performance of Monte Carlo estimation of the *interior* integral. Zhang et al. [2021a] additionally introduced BSDF antithetic sampling to more efficiently handle glossy materials.

The *boundary* integrals are unique to differentiable rendering. Recent works have shown that the *boundary* integrals can be estimated by Monte Carlo sampling object silhouettes [Li et al. 2018; Zhang et al. 2019]. Alternatively, they can be avoided altogether by reparameterizing rendering integrals [Loubet et al. 2019; Bangaru et al. 2020]. Moreover, a differential path integral formulation [Zhang et al. 2020, 2021b] that expresses both the *interior* and the *boundary* components as full path integrals, allows for the development of sophisticated Monte Carlo estimators for both components (i.e., beyond unidirectional path tracing). Yan et al. [2022] devised a guiding-based method to efficiently estimate *boundary* path integrals. Work on effective sampling and variance reduction for differentiable rendering remains nascent compared to analogous explorations in forward rendering.

Our work focuses on the largely under-explored problem of differentiating pixel reconstruction filters in a differential path integral formulation. While most closely related to BSDF antithetic sampling [Zhang et al. 2021a], our method differs in several notable and significant ways, which we discuss in §5.

3 PRELIMINARIES

We briefly review the mathematical formulations of generalized path integral [Veach 1997; Pauly et al. 2000] in §3.1 and its differential counterpart [Zhang et al. 2021b] in §3.2.

3.1 Generalized Path Integral

The response $I \in \mathbb{R}$ of a radiometric detector can be expressed as a **generalized path integral** of the form:

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

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), Ω is the **path space**, f is the **measurement contribution function**, and μ the Lebesgue measure on Ω.

Veach [1997] has shown that, for surface-only light transport (treating only interfacial reflection and refraction), the path space[1] is given by $\Omega = \cup_{N=1}^{\infty} \mathcal{M}^{N+1}$, where $\mathcal{M}$ is the union of all object surfaces, and μ is the area-product measure. Pauly et al. [2000] extend this formulation to treat volumetric light transport based on radiative transfer theory [Chandrasekhar 1960]. We briefly review this generalized formulation, below.

Path space and measure. Let $\mathcal{V} \subset \mathbb{R}^3$ be a volume that encapsulates the scene and $\mathcal{M} \subset \mathcal{V}$ be the union of all object surfaces (and media interfaces) in the scene. Then, the path space is defined as

$$\Omega = \bigcup_{N \geq 1} \mathcal{V}^{N+1}. \tag{2}$$

For a light transport path $\bar{\boldsymbol{x}} = (\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_N) \in \Omega$ with $(N+1)$ vertices and N segments, the measure μ satisfies

$$\mathrm{d}\mu(\bar{\boldsymbol{x}}) = \prod_{n=0}^{N} \begin{cases} \mathrm{d}A(\boldsymbol{x}_n), & (\boldsymbol{x}_n \in \mathcal{M}) \\ \mathrm{d}V(\boldsymbol{x}_n), & (\boldsymbol{x}_n \in \mathcal{V} \setminus \mathcal{M}) \end{cases} \tag{3}$$

where $\mathrm{d}A$ and $\mathrm{d}V$ denote the surface-area and volume measures.

Measurement contribution. Provided light path $\bar{\boldsymbol{x}} = (\boldsymbol{x}_0, \ldots, \boldsymbol{x}_N)$, its measurement contribution is the product of per-vertex and per-segment contributions:

$$f(\bar{\boldsymbol{x}}) := \left[\prod_{n=0}^{N} f_{\mathrm{v}}(\boldsymbol{x}_{n-1} \to \boldsymbol{x}_n \to \boldsymbol{x}_{n+1})\right] \left[\prod_{n=1}^{N} G(\boldsymbol{x}_{n-1} \leftrightarrow \boldsymbol{x}_n)\right]. \tag{4}$$

In this equation, the per-vertex contribution is defined as

$$f_{\mathrm{v}}(\boldsymbol{x}_{n-1} \to \boldsymbol{x}_n \to \boldsymbol{x}_{n+1}) := \\ \begin{cases} f_{\mathrm{s}}(\boldsymbol{x}_{n-1} \to \boldsymbol{x}_n \to \boldsymbol{x}_{n+1}), & (0 < n < N \text{ and } \boldsymbol{x}_n \in \mathcal{M}) \\ \sigma_{\mathrm{s}}(\boldsymbol{x}_n)\, f_{\mathrm{p}}(\boldsymbol{x}_{n-1} \to \boldsymbol{x}_n \to \boldsymbol{x}_{n+1}), & (0 < n < N \text{ and } \boldsymbol{x}_n \in \mathcal{V}_0) \\ L_{\mathrm{e}}(\boldsymbol{x}_0 \to \boldsymbol{x}_1), & (n = 0) \\ W_{\mathrm{e}}(\boldsymbol{x}_{N-1} \to \boldsymbol{x}_N), & (n = N) \end{cases} \tag{5}$$

where f_{s} is the **bidirectional scattering distribution function** (BSDF), f_{p} denotes the **single-scattering phase function**, σ_{s} is the **scattering coefficient**, and L_{e} and W_{e} capture the **source emission** and **detector importance** (or response). The per-segment contribution equals the **generalized geometric term**,

$$G(\boldsymbol{x} \leftrightarrow \boldsymbol{y}) := \tau(\boldsymbol{x} \leftrightarrow \boldsymbol{y})\ \underbrace{\mathbb{V}(\boldsymbol{x} \leftrightarrow \boldsymbol{y}) \frac{D_{\boldsymbol{x}}(\boldsymbol{y})\, D_{\boldsymbol{y}}(\boldsymbol{x})}{\|\boldsymbol{x} - \boldsymbol{y}\|^2}}_{=:\, G_0(\boldsymbol{x} \leftrightarrow \boldsymbol{y})}, \tag{6}$$

where $\mathbb{V}$ is the **mutual visibility function**. For any $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{V}$,

$$D_{\boldsymbol{x}}(\boldsymbol{y}) := \begin{cases} \left|\boldsymbol{n}(\boldsymbol{x}) \cdot \overrightarrow{\boldsymbol{xy}}\right|, & (\boldsymbol{x} \in \mathcal{M}) \\ 1, & (\boldsymbol{x} \in \mathcal{V}_0) \end{cases} \tag{7}$$

with $\boldsymbol{n}(\boldsymbol{x})$ being the (unit-length) surface normal at $\boldsymbol{x}$, and $\overrightarrow{\boldsymbol{xy}} := (\boldsymbol{y} - \boldsymbol{x})/\|\boldsymbol{y} - \boldsymbol{x}\|$. Further, $\tau(\boldsymbol{x} \leftrightarrow \boldsymbol{y})$ indicates the **transmittance** between $\boldsymbol{x}$ and $\boldsymbol{y}$ that equals

$$\tau(\boldsymbol{x} \leftrightarrow \boldsymbol{y}) = \exp\left[-\int_{\overline{\boldsymbol{xy}}} \sigma_{\mathrm{t}}(\boldsymbol{x}')\, \mathrm{d}\ell(\boldsymbol{x}')\right], \tag{8}$$

where σ_{t} is the **extinction coefficient**; $\overline{\boldsymbol{xy}}$ denotes the line segment connecting $\boldsymbol{x}$ and $\boldsymbol{y}$; and ℓ is the curve-length measure.

3.2 Generalized Differential Path Integral

Zhang et al. [2020; 2021b] have recently shown that derivatives of generalized path integrals can be expressed as *generalized differential path integrals*, a result we briefly recapitulate, below.

Material-form reparameterization. To facilitate the differentiation of a generalized path integral (1) when the scene geometry depends on the parameter θ, Zhang et al. propose to apply a change of variable to the ordinary path integral so that the new domain of integration becomes independent of the scene parameter θ. This can be achieved by parameterizing the scene-encapsulating volume $\mathcal{V}(\theta)$ using a **motion** X such that $\mathsf{X}(\cdot, \theta)$ is a differentiable bijection that maps some fixed **reference volume** $\mathcal{B}_{\mathcal{V}} \subset \mathbb{R}^3$ to $\mathcal{V}(\theta)$. Additionally, when restricted to some fixed **reference surface** $\mathcal{B}_{\mathcal{M}} \subset \mathcal{B}_{\mathcal{V}}$, the mapping $\mathsf{X}(\cdot, \theta)$ becomes a bijection from $\mathcal{B}_{\mathcal{M}}$ to the object surfaces and media interfaces $\mathcal{M}(\theta)$. Lastly, any $\boldsymbol{x} \in \mathcal{V}(\theta)$ and $\boldsymbol{p} \in \mathcal{B}_{\mathcal{V}}$ are called a **spatial point** and a **material point**, respectively.

Let the **material path space** $\hat{\Omega}$ be the set of all finite-length paths over the reference volume:

$$\hat{\Omega} := \bigcup_{N=1}^{\infty} \mathcal{B}_{\mathcal{V}}^{N+1}, \tag{9}$$

whose elements $\bar{\boldsymbol{p}} \in \hat{\Omega}$ are termed as **material light paths**.

Then, for any θ, the global parameterization $\mathsf{X}(\cdot, \theta)$ induces a path-wise bijection $\bar{\mathsf{X}}(\cdot, \theta)$ that maps a material path $\bar{\boldsymbol{p}} = (\boldsymbol{p}_0, \ldots, \boldsymbol{p}_N) \in \hat{\Omega}$ to an ordinary one $\bar{\mathsf{X}}(\bar{\boldsymbol{p}}, \theta) := (\mathsf{X}(\boldsymbol{p}_0, \theta), \ldots, \mathsf{X}(\boldsymbol{p}_N, \theta)) \in \Omega(\theta)$. Applying the change of variable given by $\bar{\mathsf{X}}(\cdot, \theta)$ to Eq. (1) gives the **material-form generalized path integral**:

$$I = \int_{\hat{\Omega}} \hat{f}(\bar{\boldsymbol{p}})\, \mathrm{d}\mu(\bar{\boldsymbol{p}}), \tag{10}$$

where the domain of integration $\hat{\Omega}$ is independent of the scene parameter θ. In Eq. (10), $\hat{f}$ is the **material measurement contribution** provided by the original measurement contribution of Eq. (4) and Jacobian determinants J capturing this change of variable:

$$\hat{f}(\bar{\boldsymbol{p}}) := f(\bar{\boldsymbol{x}}) \prod_{n=0}^{N} J(\boldsymbol{p}_n), \tag{11}$$

where $\bar{\boldsymbol{x}} = \bar{\mathsf{X}}(\bar{\boldsymbol{p}}, \theta)$, $\boldsymbol{x}_n = \mathsf{X}(\boldsymbol{p}_n, \theta)$, and, for any material point $\boldsymbol{p}$,

$$J(\boldsymbol{p}) := \begin{cases} \|\mathrm{d}A(\mathsf{X}(\boldsymbol{p}, \theta))/\mathrm{d}A(\boldsymbol{p})\|, & (\boldsymbol{p} \in \mathcal{B}_{\mathcal{M}}) \\ \|\mathrm{d}V(\mathsf{X}(\boldsymbol{p}, \theta))/\mathrm{d}V(\boldsymbol{p})\|. & (\boldsymbol{p} \in \mathcal{B}_{\mathcal{V}} \setminus \mathcal{B}_{\mathcal{M}}) \end{cases} \tag{12}$$

Choice of reference configuration. In practice, when estimating derivatives at $\theta = \theta_0$ (for some fixed θ_0), the reference volume can be set to $\mathcal{B}_{\mathcal{V}} = \mathcal{V}(\theta_0)$ and the reference surface to $\mathcal{B}_{\mathcal{M}} = \mathcal{M}(\theta_0)$.

Under this configuration, at $\theta = \theta_0$, the mapping $\mathsf{X}(\cdot, \theta_0)$ becomes the identity map, causing the Jacobian determinants $\|\mathrm{d}\boldsymbol{x}_n/\mathrm{d}\boldsymbol{p}_n\|$ defined in Eq. (12) to reduce to one. Further, the path space $\Omega(\theta_0)$ coincides with the material path space $\hat{\Omega}$.

We note that, when θ controls scene geometry, the derivative of $\|\mathrm{d}\boldsymbol{x}_n/\mathrm{d}\boldsymbol{p}_n\|$ with respect to θ is generally nonzero—even at $\theta = \theta_0$.

[1] We hyperlink keywords to their definitions.

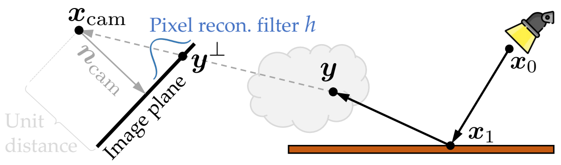


Fig. 2. A **perspective pinhole camera** is specified by its center of projection $\boldsymbol{x}_{\text{cam}} \in \mathcal{V}$, axis of projection $\boldsymbol{n}_{\text{cam}} \in \mathbb{S}^2$, and per-pixel reconstruction filters h defined over an image plane. We treat a light transport path as $(\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{N-1}, \boldsymbol{y})$ with $\boldsymbol{y}$ being an arbitrary (surface or volume) vertex that encodes the response of a pinhole camera via Eq. (14).

Generalized differential path integral. In general, the derivative of Eq. (10) have been proven [Zhang et al. 2020, 2021b] to equal

$$\frac{\mathrm{d}I}{\mathrm{d}\theta} = \overset{\text{interior}}{\boxed{\int_{\hat{\Omega}} \frac{\mathrm{d}\hat{f}(\bar{\boldsymbol{p}})}{\mathrm{d}\theta}\,\mathrm{d}\mu(\bar{\boldsymbol{p}})}} + \overset{\text{boundary}}{\boxed{\int_{\partial\hat{\Omega}} \Delta\hat{f}_K(\bar{\boldsymbol{p}})\, V_{\Delta\mathcal{B}_K}(\boldsymbol{p}_K)\,\mathrm{d}\dot{\mu}(\bar{\boldsymbol{p}})}}\,. \tag{13}$$

This result is called the (material-form) **differential generalized path integral**. In Eq. (13), the *interior* component is over the same material path space $\hat{\Omega}$ as the material-form generalized path integral in Eq. (10). The *boundary* integral, on the other hand, is over the **material boundary path space** $\partial\hat{\Omega}$ that is unique to differentiable rendering and generally depends on the scene parameter θ. At a high level, $\partial\hat{\Omega}$ comprises **material boundary paths**: for each $\bar{\boldsymbol{p}} = (\boldsymbol{p}_0, \ldots, \boldsymbol{p}_N) \in \partial\hat{\Omega}$ there exists exactly one **boundary segment** $\overline{\boldsymbol{p}_{K-1}\,\boldsymbol{p}_K}$ such that $\boldsymbol{p}_K$ lies on a discontinuity boundary w.r.t. $\boldsymbol{p}_{K-1}$.

4 FORMULATING PERSPECTIVE PINHOLE CAMERAS

We focus on the commonly-used *perspective pinhole camera* model, parameterized by a point $\boldsymbol{x}_{\text{cam}} \in \mathcal{V}$ for the **center of projection** and a direction $\boldsymbol{n}_{\text{cam}} \in \mathbb{S}^2$ for its **axis of projection**. Additionally, each pixel has a **pixel reconstruction filter** h specified over the *image plane* perpendicular to $\boldsymbol{n}_{\text{cam}}$ at unit distance from $\boldsymbol{x}_{\text{cam}}$.

Zhang et al. [2021b] propose an encoding of perspective pinhole cameras using the detector importance function W_{e} as:

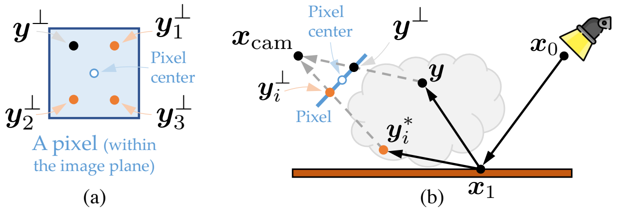
$$W_{\text{e}}(\boldsymbol{x}_{N-1} \to \boldsymbol{y}) = f_{\text{v}}(\boldsymbol{x}_{N-1} \to \boldsymbol{y} \to \boldsymbol{x}_{\text{cam}})\, W_{\text{e}}^{\text{pinhole}}(\boldsymbol{y}), \tag{14}$$

where, for all $\boldsymbol{y} \in \mathcal{V}$,

$$W_{\text{e}}^{\text{pinhole}}(\boldsymbol{y}) := \frac{G(\boldsymbol{y} \leftrightarrow \boldsymbol{x}_{\text{cam}})\, h(\boldsymbol{y}^{\perp})}{(\boldsymbol{n}_{\text{cam}} \cdot \overrightarrow{\boldsymbol{x}_{\text{cam}}\,\boldsymbol{y}})^3}\,, \tag{15}$$

where G is the generalized geometric term of Eq. (6), $\boldsymbol{y}^{\perp}$ is the projection of $\boldsymbol{y}$ onto the image plane, and $\overrightarrow{\boldsymbol{x}_{\text{cam}}\,\boldsymbol{y}}$ is the unit vector from $\boldsymbol{x}_{\text{cam}}$ to $\boldsymbol{y}$. In Eqs. (14) and (15), we rename the vertex $\boldsymbol{x}_N$ as $\boldsymbol{y}$ to emphasize that it is directly "connected" to the pinhole camera.

Under this formulation, as illustrated in Figure 2, a light path takes the form of $\bar{\boldsymbol{x}} = (\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{N-1}, \boldsymbol{y})$ where $\boldsymbol{y} \in \mathcal{V}$ is an arbitrary surface or volume vertex that encodes the response of the pinhole camera at $\boldsymbol{x}_{\text{cam}}$. This effectively treats the entire scene volume $\mathcal{V}$ as a virtual detector and, thus, does not require restricting the last vertex of each light path to $\boldsymbol{x}_{\text{cam}}$ (which would introduce additional Dirac delta functions to the measurement contribution f).


$\boldsymbol{y}^{\perp}$
$\boldsymbol{y}_1^{\perp}$
Pixel center
$\boldsymbol{y}_2^{\perp}$
$\boldsymbol{y}_3^{\perp}$
A pixel (within the image plane)
$\boldsymbol{x}_{\text{cam}}$
Pixel center
$\boldsymbol{y}^{\perp}$
$\boldsymbol{x}_0$
$\boldsymbol{y}$
$\boldsymbol{y}_i^{\perp}$
Pixel
$\boldsymbol{y}_i^*$
$\boldsymbol{x}_1$

(a) (b)

Fig. 3. **Pixel-level antithetic sampling (interior):** (a) Correlated sets of light paths are generated so that their intersections $\boldsymbol{y}^{\perp}$, $\boldsymbol{y}_1^{\perp}$, $\boldsymbol{y}_2^{\perp}$ and $\boldsymbol{y}_3^{\perp}$ with the image plane are symmetric around the pixel center. (b) Given an ordinary path $\bar{\boldsymbol{x}} = (\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{y})$, we construct its antithesis $\bar{\boldsymbol{x}}_i^*$ by replacing only the last vertex with $\boldsymbol{y}_i^*$, yielding $\bar{\boldsymbol{x}}^* = (\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{y}_i^*)$ for $i = 1, 2, 3$.

As the scene evolves with a parameter θ, the camera's position $\boldsymbol{x}_{\text{cam}}$, axis $\boldsymbol{n}_{\text{cam}}$, and pixel reconstruction filters h can also generally be treated as functions of θ. We treat both $\boldsymbol{x}_{\text{cam}}$ and $\boldsymbol{n}_{\text{cam}}$ as being differentiable with respect to θ. Moreover, we assume pixel reconstruction filters h are also differentiable, except at a set of zero-measure jump discontinuities (e.g., pixel boundaries).

Provided Eq. (14), for any material light path $\bar{\boldsymbol{p}} = (\boldsymbol{p}_0, \ldots, \boldsymbol{p}_{N-1}, \boldsymbol{q})$, the material-form generalized path integral (10) can be rewritten as

$$\boxed{I = \int_{\hat{\Omega}} \underbrace{\hat{f}_0(\bar{\boldsymbol{p}})\, W_{\text{e}}^{\text{pinhole}}(\boldsymbol{y})\, J(\boldsymbol{q})}_{=\hat{f}(\bar{\boldsymbol{p}})}\, \mathrm{d}\mu(\bar{\boldsymbol{p}}),} \tag{16}$$

where $\boldsymbol{y} = \mathsf{X}(\boldsymbol{q}, \theta)$; $J(\boldsymbol{q})$ is defined in Eq. (12); $\hat{f}_0(\bar{\boldsymbol{p}})$ is a factor of the material measurement contribution $\hat{f}(\bar{\boldsymbol{p}})$; and $\boldsymbol{q}$ is the spatial counterpart of $\boldsymbol{y}$ (that is "connected" to the pinhole camera at $\boldsymbol{x}_{\text{cam}}$).

As described in §3.2, differentiating Eq. (16) with respect to a scene parameter θ yields a generalized differential path integral (13) where the *interior* component requires estimating derivatives $\mathrm{d}h/\mathrm{d}\theta$ of the pixel reconstruction filter h, and the *boundary* component needs to capture jump discontinuities of h. Previously, both aspects have remained largely under-explored. In the rest of this paper, we introduce new Monte Carlo methods to efficiently estimate both the *interior* (§5) and the *boundary* components (§6).

Discussion. Besides what we will present in §5 and §6, the differentiation of Eq. (16) can be treated using several existing methods such as applying attached sampling [Zeltner et al. 2021] to the pixel reconstruction filter h and differentiating through filter normalization [Vicini et al. 2022]. Unfortunately, these methods suffer from many problems, which we will demonstrate in Appendix A.

5 ESTIMATING THE INTERIOR COMPONENT

When differentiating radiometric measurements of a perspective pinhole camera given by Eq. (16) with respect to a scene parameter θ, the *interior* component of the resulting derivative has the form

$$\int_{\hat{\Omega}} \frac{\mathrm{d}}{\mathrm{d}\theta} \left(\hat{f}_0(\bar{\boldsymbol{p}})\, W_{\text{e}}^{\text{pinhole}}(\boldsymbol{y})\, J(\boldsymbol{q}) \right) \mathrm{d}\mu(\bar{\boldsymbol{p}}), \tag{17}$$

where $W_{\text{e}}^{\text{pinhole}}$ is defined in Eq. (15) and involves the pixel reconstruction filter $h(\boldsymbol{y}^{\perp})$. When the parameter θ affects object geometry that is directly visible to the camera, for any spatial point $\boldsymbol{y}$ associated with the object, $h(\boldsymbol{y}^\perp)$ depends on θ—even if the filter h itself is constant. This is because, under the material-form parameterization in §3.2, we have $\boldsymbol{y} = \mathsf{X}(\boldsymbol{q}, \theta)$ for some fixed material point $\boldsymbol{q}$.

To efficiently estimate this integral when the derivative ${}^{\mathrm{d}h}/_{\mathrm{d}\theta}$ exhibits point symmetry with respect to the pixel center—which is typically the case in practice—Zhang et al. [2021b] proposed leveraging pixel-level antithetic sampling. The basic idea is to generate pairs of material paths $\bar{\boldsymbol{p}} = (\boldsymbol{p}_{1,0}, \boldsymbol{p}_{1,1}, \ldots, \boldsymbol{q})$ and $\bar{\boldsymbol{p}}^* = (\boldsymbol{p}_{2,0}, \boldsymbol{p}_{2,1}, \ldots, \boldsymbol{q}^*)$ such that image-plane projections of $\boldsymbol{y} = \mathsf{X}(\boldsymbol{q}, \theta)$ and $\boldsymbol{y}^* = \mathsf{X}(\boldsymbol{q}^*, \theta)$ are point-symmetric. This can be achieved by (i) sampling image-plane location $\boldsymbol{y}_1^\perp$ based on the reconstruction filter h and setting $\boldsymbol{y}_2^\perp$ as the point reflection of $\boldsymbol{y}_1^\perp$; (ii) tracing two camera rays through $\boldsymbol{y}_1^\perp$ and $\boldsymbol{y}_2^\perp$ to obtain spatial points $\boldsymbol{y}$ and $\boldsymbol{y}^*$ (which in turn yield the corresponding material points $\boldsymbol{q}$ and $\boldsymbol{q}^*$); and (iii) constructing the remaining vertices of paths $\bar{\boldsymbol{p}}$ and $\bar{\boldsymbol{p}}*$ with standard methods, like path tracing (with correlated random samples).

Although the aforementioned process works adequately in some cases, it suffers from several major problems: first, it has a high computational overhead since two full paths $\bar{\boldsymbol{p}}$ and $\bar{\boldsymbol{p}}^*$ have to be constructed; second, coupling an ordinary path with only one antithesis (that exploits point symmetry) can sometimes produces high variance at edges; and, finally, the process does not handle an important path-sampling scheme for adjoint particle tracing (APT) and BDPT, where next-event estimation is applied at $\boldsymbol{q}$ (with a shadow ray towards $\boldsymbol{x}_{\mathrm{cam}}$).

To address these problems, we introduce a new antithetic sampling method for pixel reconstruction filters. Provided an *ordinary* path $\bar{\boldsymbol{p}} = (\boldsymbol{p}_0, \ldots, \boldsymbol{p}_{N-1}, \boldsymbol{q})$, we deterministically construct three *antithetic* paths $\bar{\boldsymbol{p}}_i^* = (\boldsymbol{p}_0, \ldots, \boldsymbol{p}_{N-1}, \boldsymbol{q}_i^*)$ for $i = 1, 2, 3$. The ordinary and antithetic paths are identical except for the last vertices $\boldsymbol{q}$ and $\boldsymbol{q}_i^*$: their image-plane projections of $\boldsymbol{y} = \mathsf{X}(\boldsymbol{q}, \theta)$ and $\boldsymbol{y}_i^* = \mathsf{X}(\boldsymbol{q}_i^*, \theta)$ are symmetric about the pixel center (see Figure 3.) We demonstrate the advantage of this four-point pattern in Figures 4 and 16.

Let $\mathrm{pdf}(\bar{\boldsymbol{p}})$ be the probability density of a material path $\bar{\boldsymbol{p}}$ sampled using standard techniques such as unidirectional and bidirectional path tracing as well as adjoint particle tracing. When the mapping (induced by aforementioned construction) between the original path $\bar{\boldsymbol{p}}$ and the antithetic one $\bar{\boldsymbol{p}}_i^*$ is *one-to-one* (for each $i = 1, 2, 3$), we can express the probability density $\mathrm{pdf}_i^*(\bar{\boldsymbol{p}}_i^*)$ of antithetic path $\bar{\boldsymbol{p}}_i^*$ analytically based on $\mathrm{pdf}(\bar{\boldsymbol{p}})$. We will discuss the construction of $\bar{\boldsymbol{p}}_i^*$ and the calculation of its probability density $\mathrm{pdf}_i^*(\bar{\boldsymbol{p}}_i^*)$ in §5.1.

We combine the contributions of both the ordinary path $\bar{\boldsymbol{p}}$ and its antitheses $\bar{\boldsymbol{p}}_i^*$using multiple importance sampling (MIS):

$$\boxed{w(\bar{\boldsymbol{p}})\,\frac{\frac{\mathrm{d}}{\mathrm{d}\theta}\hat{f}(\bar{\boldsymbol{p}})}{\mathrm{pdf}(\bar{\boldsymbol{p}})} + \sum_i w_i^*(\bar{\boldsymbol{p}}_i^*)\,\frac{\frac{\mathrm{d}}{\mathrm{d}\theta}\hat{f}(\bar{\boldsymbol{p}}_i^*)}{\mathrm{pdf}_i^*(\bar{\boldsymbol{p}}_i^*)},} \tag{18}$$

where w and w^* are the MIS weighting functions which we set using the balanced heuristics [Veach 1997]: $w(\bar{\boldsymbol{p}}) = \frac{\mathrm{pdf}(\bar{\boldsymbol{p}})}{\mathrm{pdf}(\bar{\boldsymbol{p}}) + \sum_j \mathrm{pdf}_j^*(\bar{\boldsymbol{p}})}$ and $w_i^*(\bar{\boldsymbol{p}}) = \frac{\mathrm{pdf}_i^*(\bar{\boldsymbol{p}})}{\mathrm{pdf}(\bar{\boldsymbol{p}}) + \sum_j \mathrm{pdf}_j^*(\bar{\boldsymbol{p}})}$, for any material path $\bar{\boldsymbol{p}}$ and $i = 1, 2, 3$.

Below, we discuss how antithetic path $\bar{\boldsymbol{p}}_i*$ can be constructed in a one-to-one fashion given the ordinary $\bar{\boldsymbol{p}}$ for each i. We detail how to estimate contributions of antithetic paths in Appendix B.

5.1 Constructing Antithetic Paths

Provided an ordinary path $\bar{\boldsymbol{p}} = (\boldsymbol{p}_0, \ldots, \boldsymbol{p}_{N-1}, \boldsymbol{q})$, we discuss the construction of an antithetic path of the form $\bar{\boldsymbol{p}}^* = (\boldsymbol{p}_0, \ldots, \boldsymbol{p}_{N-1}, \boldsymbol{q}^*)$ that induces a bijective mapping from $\bar{\boldsymbol{p}}$ to $\bar{\boldsymbol{p}}^*$. Note that this process will be applied to obtain each antithetic path $\bar{\boldsymbol{p}}_i^*$ for $i = 1, 2, 3$.

Since $\bar{\boldsymbol{p}}$ and $\bar{\boldsymbol{p}}^*$ only differ in their last vertices, the problem amounts to constructing the last vertex $\boldsymbol{q}^*$ of the antithetic path $\bar{\boldsymbol{p}}^*$ in a one-to-one fashion. Below, we discuss how to construct $\boldsymbol{q}^*$ based on the ordinary path $\bar{\boldsymbol{p}}$ and, more specifically, its last vertex $\boldsymbol{q}$.

Surface vertex. When $\boldsymbol{q}$ is a surface vertex (i.e., $\boldsymbol{q} \in \mathcal{B}_{\mathcal{M}}$), as illustrated in Figure 5-a, $\boldsymbol{q}^*$ can be constructed by:

(1) Finding the image-plane projection $\boldsymbol{y}_1^\perp$ of $\boldsymbol{y} = \mathsf{X}(\boldsymbol{p}, \theta)$;

(2) Obtaining the point $\boldsymbol{y}_2^\perp$ by mirroring $\boldsymbol{y}_1^\perp$ (based on antithetic sampling pattern);

(3) Tracing a camera ray through the image-plane at $\boldsymbol{y}_2^\perp$ to obtain the first surface intersection $\boldsymbol{y}^*$ (ignoring all media without refractive interfaces);

(4) Letting $\boldsymbol{q}^* = \mathsf{X}^{-1}(\boldsymbol{y}^*, \theta)$.

Since the mapping between image-plane locations $\boldsymbol{y}_1^\perp$ and $\boldsymbol{y}_2^\perp$ is one-to-one, so is the mapping between $\boldsymbol{q}$ and $\boldsymbol{q}^*$.

To obtain the probability density $\mathrm{pdf}^*(\bar{\boldsymbol{p}}^*)$ we rely on the relation

$$\mathrm{pdf}^*(\bar{\boldsymbol{p}}^*) = \mathrm{pdf}(\bar{\boldsymbol{p}}) \left\| \frac{\mathrm{d}A(\boldsymbol{q})}{\mathrm{d}A(\boldsymbol{q}^*)} \right\|. \tag{19}$$

We now derive the ratio $\|{}^{\mathrm{d}A(\boldsymbol{q})}/_{\mathrm{d}A(\boldsymbol{q}^*)}\|$ on the right-hand side of Eq. (19) due to the effective change of variable from $\boldsymbol{q}$ to $\boldsymbol{q}^*$. Let $\phi_{\boldsymbol{y}}$ and $\phi_{\boldsymbol{y}}^*$ denote, respectively, the angles from the camera's axis of projection $\boldsymbol{n}_{\mathrm{cam}}$ to directions $\overrightarrow{\boldsymbol{x}_{\mathrm{cam}}\,\boldsymbol{y}}$ and $\overrightarrow{\boldsymbol{x}_{\mathrm{cam}}\,\boldsymbol{y}^*}$. Then, as shown by Lehtinen et al. [2013], it holds that

$$\left\| \frac{\mathrm{d}A(\boldsymbol{y})}{\mathrm{d}A(\boldsymbol{y}^*)} \right\| = \frac{G_0(\boldsymbol{y}^* \leftrightarrow \boldsymbol{x}_{\mathrm{cam}}) \cos^3\phi_{\boldsymbol{y}}}{G_0(\boldsymbol{y} \leftrightarrow \boldsymbol{x}_{\mathrm{cam}}) \cos^3\phi_{\boldsymbol{y}}^*}, \tag{20}$$

where G_0 is the (standard) geometric term defined in Eq. (6). Thus,

$$\left\| \frac{\mathrm{d}A(\boldsymbol{q})}{\mathrm{d}A(\boldsymbol{q}^*)} \right\| = \frac{G_0(\boldsymbol{y}^* \leftrightarrow \boldsymbol{x}_{\mathrm{cam}}) \cos^3\phi_{\boldsymbol{y}}}{G_0(\boldsymbol{y} \leftrightarrow \boldsymbol{x}_{\mathrm{cam}}) \cos^3\phi_{\boldsymbol{y}}^*} \, \frac{J(\boldsymbol{q}^*)}{J(\boldsymbol{q})}. \tag{21}$$

In practice, as discussed in §3, when estimating derivatives at $\theta = \theta_0$ with the reference surface set to $\mathcal{B}_{\mathcal{M}} = \mathcal{M}(\theta_0)$, both $\mathsf{X}(\cdot, \theta_0)$ and $\mathsf{X}^{-1}(\cdot, \theta_0)$ become identity maps. This causes $\boldsymbol{q}^*$ to coincide with $\boldsymbol{y}^*$, and $\boldsymbol{q}$ with $\boldsymbol{y}$. Further, ${}^{J(\boldsymbol{q}^*)}/_{J(\boldsymbol{q})}$ in Eq. (21) reduces to one.

Volume vertex. When $\boldsymbol{q}$ is a volume vertex (i.e., $\boldsymbol{q} \in \mathcal{B}_{\mathcal{V}} \setminus \mathcal{B}_{\mathcal{M}}$), we construct the vertex $\boldsymbol{q}^*$ using a three-step process identical to the surface case except for the third step where we select the spatial point $\boldsymbol{y}^*$ along the ray $\boldsymbol{x}_{\mathrm{cam}} \to \boldsymbol{y}_2^\perp$ such that the transmittance between $\boldsymbol{x}_{\mathrm{cam}}$ and $\boldsymbol{y}^*$ matches that between $\boldsymbol{x}_{\mathrm{cam}}$ and $\boldsymbol{y}$ (see Figure 5-b): that is, $\tau(\boldsymbol{x}_{\mathrm{cam}} \leftrightarrow \boldsymbol{y}^*) = \tau(\boldsymbol{x}_{\mathrm{cam}} \leftrightarrow \boldsymbol{y})$.

Similar to Eq. (19), the probability density $\mathrm{pdf}^*(\bar{\boldsymbol{p}}^*)$ satisfies

$$\mathrm{pdf}^*(\bar{\boldsymbol{p}}^*) = \mathrm{pdf}(\bar{\boldsymbol{p}}) \left\| \frac{\mathrm{d}V(\boldsymbol{q})}{\mathrm{d}V(\boldsymbol{q}^*)} \right\|. \tag{22}$$

Let $r = \|\boldsymbol{y} - \boldsymbol{x}_{\mathrm{cam}}\|$ and $r^* = \|\boldsymbol{y}^* - \boldsymbol{x}_{\mathrm{cam}}\|$. Then, we can verify that:

$$\begin{aligned} {}^{\mathrm{d}V(\boldsymbol{y})}/_{r^2} &= \cos^3\phi_{\boldsymbol{y}}\,\mathrm{d}A(\boldsymbol{y}_1^\perp)\,\mathrm{d}r, \\ {}^{\mathrm{d}V(\boldsymbol{y}^*)}/_{(r^*)^2} &= \cos^3\phi_{\boldsymbol{y}}^*\,\mathrm{d}A(\boldsymbol{y}_2^\perp)\,\mathrm{d}r^*. \end{aligned} \tag{23}$$

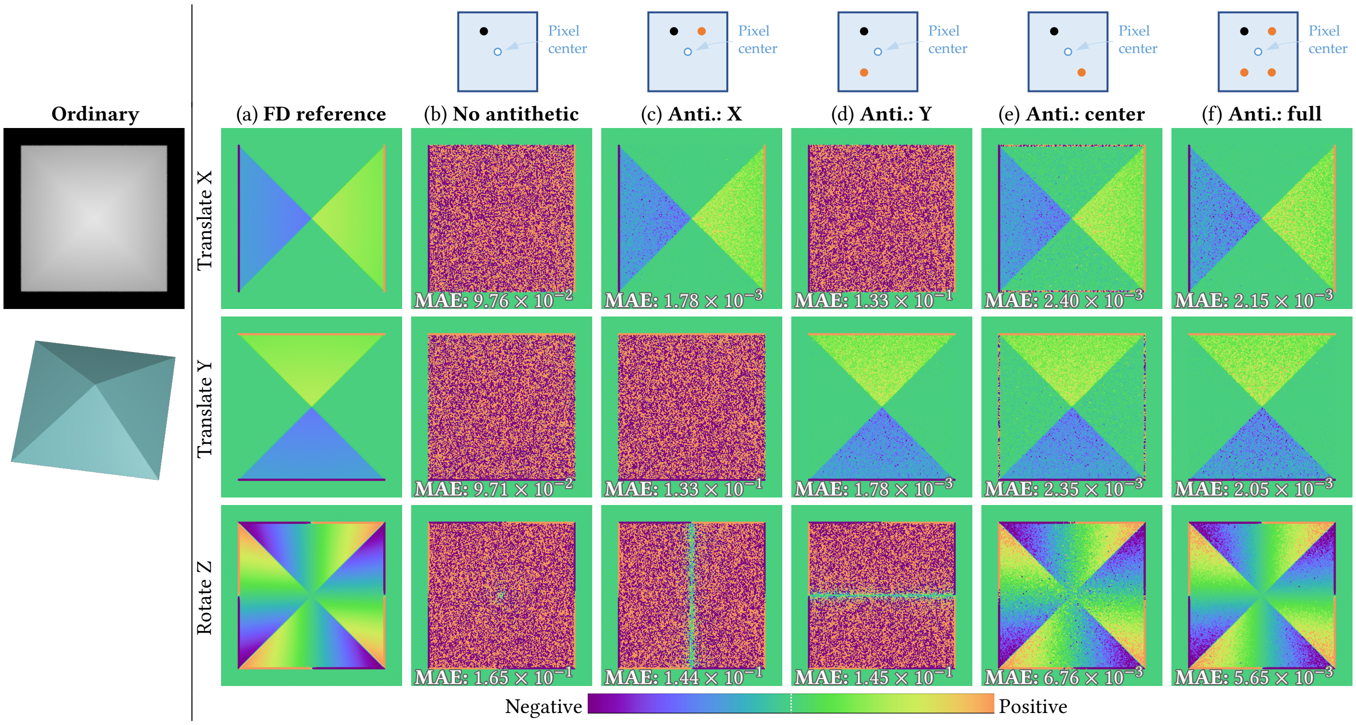

Pixel center
Pixel center
Pixel center
Pixel center
Pixel center
Ordinary
(a) FD reference
(b) No antithetic
(c) Anti.: X
(d) Anti.: Y
(e) Anti.: center
(f) Anti.: full
Translate X
MAE: 9.76×10^{-2}
MAE: 1.78×10^{-3}
MAE: 1.33×10^{-1}
MAE: 2.40×10^{-3}
MAE: 2.15×10^{-3}
Translate Y
MAE: 9.71×10^{-2}
MAE: 1.33×10^{-1}
MAE: 1.78×10^{-3}
MAE: 2.35×10^{-3}
MAE: 2.05×10^{-3}
Rotate Z
MAE: 1.65×10^{-1}
MAE: 1.44×10^{-1}
MAE: 1.45×10^{-1}
MAE: 6.76×10^{-3}
MAE: 5.65×10^{-3}
Negative
Positive

Fig. 4. **Antithetic sampling pattern**: We evaluate the effectiveness of four different antithetic sampling patterns (c–f) using derivative images of a pyramid-like object viewed from the top using the tent pixel reconstruction filter (31). The derivatives are estimated with respect to translations along the X and Y axes (that are within the image plane), and rotation about the Z axis (that is perpendicular to the image plane), respectively. All derivative images in (b–f), whose accuracy are measured using mean absolute error (MAE), are generated in equal time. Previously, Zhang et al. [2021b] proposed to use the two-point pattern in (e). Although this works adequately in many cases, we found the four-point pattern in (f) to offer better robustness. We note that the derivative estimates in (b–f) are all unbiased and will eventually converge to the correct solution with sufficiently many samples.

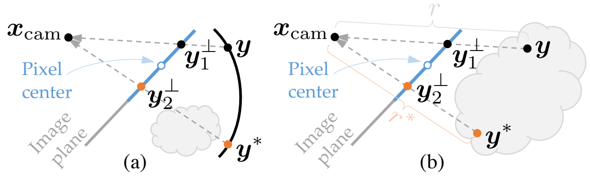

$\boldsymbol{x}_{\mathrm{cam}}$
Pixel center
$\boldsymbol{y}_1^{\perp}$
$\boldsymbol{y}$
$\boldsymbol{y}_2^{\perp}$
Image plane
$\boldsymbol{y}^*$
(a)
r
$\boldsymbol{x}_{\mathrm{cam}}$
Pixel center
$\boldsymbol{y}_1^{\perp}$
$\boldsymbol{y}$
$\boldsymbol{y}_2^{\perp}$
r^*
Image plane
$\boldsymbol{y}^*$
(b)

Fig. 5. **Construction of antithetic path:** Given ordinary light path with the last vertex $\boldsymbol{y}$, we build the antithetic path that is identical to the ordinary except for the last vertex $\boldsymbol{y}^*$. We construct $\boldsymbol{y}^*$ such that the mapping between $\boldsymbol{y}$ and $\boldsymbol{y}^*$ is one-to-one based on the type of vertex $\boldsymbol{y}$: (a) When $\boldsymbol{y}$ is a surface vertex (i.e., $\boldsymbol{y} \in \mathcal{M}$), we trace a camera ray $\boldsymbol{x}_{\mathrm{cam}} \to \boldsymbol{y}_2^{\perp}$ and set $\boldsymbol{y}^*$ as the first surface intersection while ignoring all medium (with index-matched interfaces); (b) When $\boldsymbol{y}$ is a volume vertex (i.e., $\boldsymbol{y} \in \mathcal{V} \setminus \mathcal{M}$), we set $\boldsymbol{y}^* = \boldsymbol{x}_{\mathrm{cam}} + r^* \, \overrightarrow{\boldsymbol{x}_{\mathrm{cam}} \, \boldsymbol{y}_2^{\perp}}$ with some $r^* > 0$ such that $\tau(\boldsymbol{x}_{\mathrm{cam}} \leftrightarrow \boldsymbol{y}) = \tau(\boldsymbol{x}_{\mathrm{cam}} \leftrightarrow \boldsymbol{y}^*)$.

Our construction of the vertex $\boldsymbol{y}^*$ implies that $\mathrm{d}A(\boldsymbol{y}_1^{\perp}) = \mathrm{d}A(\boldsymbol{y}_2^{\perp})$ and ${}^{\mathrm{d}r^*}\!/_{\mathrm{d}r} = {}^{\sigma_{\mathrm{t}}(\boldsymbol{y})}\!/_{\sigma_{\mathrm{t}}(\boldsymbol{y}^*)}$. It follows that

$$\left\| \frac{\mathrm{d}V(\boldsymbol{q})}{\mathrm{d}V(\boldsymbol{q}^*)} \right\| = \left\| \frac{\mathrm{d}V(\boldsymbol{y})}{\mathrm{d}V(\boldsymbol{y}^*)} \right\| \frac{J(\boldsymbol{q}^*)}{J(\boldsymbol{q})} = \frac{\cos^3\phi_{\boldsymbol{y}} \, r^2 \, \sigma_{\mathrm{t}}(\boldsymbol{y}^*)}{\cos^3\phi_{\boldsymbol{y}}^* \, (r^*)^2 \, \sigma_{\mathrm{t}}(\boldsymbol{y})} \frac{J(\boldsymbol{q}^*)}{J(\boldsymbol{q})}. \tag{24}$$

Similar to the surface case, when estimating derivatives at $\theta = \theta_0$ with the reference volume set to $\mathcal{B}_{\mathcal{V}} = \mathcal{V}(\theta_0)$, the factor ${}^{J(\boldsymbol{q}^*)}\!/_{J(\boldsymbol{q})}$ in Eq. (24) reduces to one.

5.2 Discussion

Failed constructions. Occasionally, for an ordinary path $\bar{\boldsymbol{p}}$, the construction of its antitheses $\bar{\boldsymbol{p}}_i^*$ (§5.1) can fail when: (i) the camera ray for the antithetic path does not intersect any scene surface for the surface case; or (ii) there is no point $\boldsymbol{y}^*$ along the ray satisfying the transmittance constraint $\tau(\boldsymbol{x}_{\mathrm{cam}} \leftrightarrow \boldsymbol{y}^*) = \tau(\boldsymbol{x}_{\mathrm{cam}} \leftrightarrow \boldsymbol{y})$. When the construction fails, we simply consider the antithetic path $\bar{\boldsymbol{p}}^*$ to be nonexistent and set its contribution to zero.

Similarly, when calculating the probability density $\mathrm{pdf}_i^*(\bar{\boldsymbol{p}})$ for some path $\bar{\boldsymbol{p}}$, there may not exist an ordinary path whose i-th antithesis equals $\bar{\boldsymbol{p}}$. In this case, we set $\mathrm{pdf}_i^*(\bar{\boldsymbol{p}}) = 0$.

Equal-transmittance vs. equal-distance. When $\boldsymbol{y}$ is a volume vertex (Figure 5-b), its antithesis $\boldsymbol{y}^*$ can also be constructed in a *equal-distance* fashion by setting $r^* = r$ rather than requiring $\tau(\boldsymbol{x}_{\mathrm{cam}} \leftrightarrow \boldsymbol{y}^*)$ to match $\tau(\boldsymbol{x}_{\mathrm{cam}} \leftrightarrow \boldsymbol{y})$. This, however, tends to increase variance (see Figure 6). On the other hand, when the aforementioned equal-transmittance construction is difficult, e.g., in heterogeneous media, the equal-distance variant can be used instead.

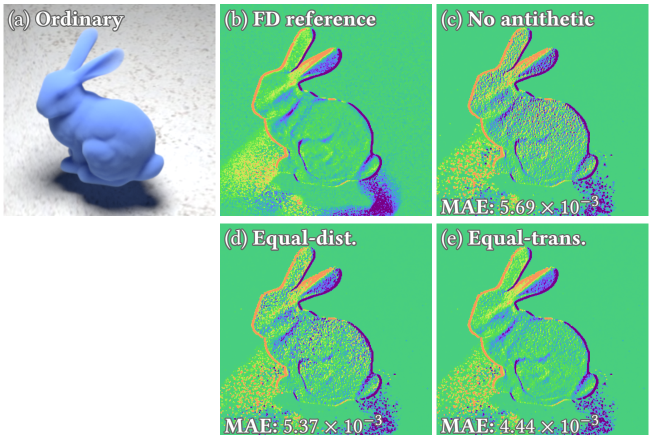


Fig. 6. **Equal-time comparison** of derivate images (visualized using the same color map as Figure 4) estimated with volumetric adjoint particle tracing (APT) with: (c) no antithetic sampling; (d) our *equal-distance* antithetic sampling with $r^* = r$; and (e) our *equal-transmittance* antithetic sampling with $\tau(\boldsymbol{x}_{\text{cam}} \leftrightarrow \boldsymbol{y}^*) = \tau(\boldsymbol{x}_{\text{cam}} \leftrightarrow \boldsymbol{y})$. This example contains a homogeneous translucent bunny (without refractive interfaces) lit by an area light and uses the tent reconstruction filter (31). The derivatives are computed with respect to the horizontal translation of the bunny.

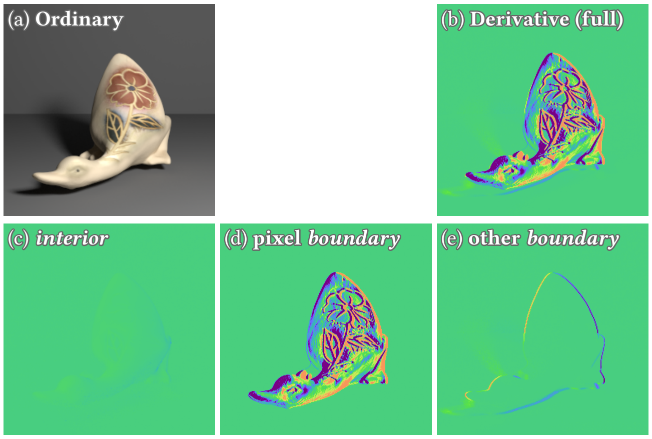


Fig. 7. **Derivative component visualization:** (a) Forward rendering of a duck model [Dong et al. 2014] using a box reconstruction filter (32). (b) Corresponding derivative image with respect to horizontal translation of the duck (visualized with the same color map as Figure 4). The full derivative equals the sum of the *interior* (c), pixel *boundary* (d), and other *boundary* components (e). We discuss the estimation of (d) in §6.

Handling multiple pixels. When the supports of reconstruction filters of neighboring pixels overlap, a light path $\bar{\boldsymbol{p}}$ can "intersect" (i.e., contribute to) multiple pixels. Here, for each intersecting pixel j, we construct one antithetic path $\bar{\boldsymbol{p}}_j^*$ using the same ordinary path $\bar{\boldsymbol{p}}$, and use our MIS estimator of Eq. (18) with $\bar{\boldsymbol{p}}$ and $\bar{\boldsymbol{p}}_j^*$ for this pixel.

Relation with BSDF antithetic sampling. Our technique is related to the BSDF sampling method introduced by Zhang et al. [2021a]. Besides being applied to pixel reconstruction filters as opposed to BSDFs, our method differs from Zhang et al.'s in the following ways. Firstly, we leverage a four-point sampling pattern in the image plane, and Zhang et al. use a two-point pattern in the half-way vector space. Secondly, our technique considers volume light transport while Zhang et al.'s approach is restricted to the surface-only case. Lastly, as we will demonstrate in §6, our antithetic sampling also benefits the estimation of *boundary* integrals.

6 ESTIMATING THE BOUNDARY COMPONENT

We now discuss the estimation of the *boundary* integral emerging from the differentiation of Eq. (16). Previously, Zhang et al. [2021b] have assumed the pixel reconstruction filter h to be continuous. We relax this assumption by considering jump discontinuity points of h and focus on the following **pixel *boundary* integral**:

$$\int_{\Delta\mathcal{B}_{\mathcal{V}}[h]} \underbrace{\left(\int_{\hat{\Omega}_0} \hat{f}_0(\bar{\boldsymbol{p}}_+)\,\mathrm{d}\mu(\bar{\boldsymbol{p}})\right) \Delta W_{\mathrm{e}}^{\text{pinhole}}(\boldsymbol{y})\, v(\boldsymbol{q})\, J(\boldsymbol{q})\,\mathrm{d}\boldsymbol{q}}_{=:\,F_{\mathrm{B}}(\boldsymbol{q})}, \qquad (25)$$

where:

- $\boldsymbol{y} = \mathsf{X}(\boldsymbol{q}, \theta)$, J follows Eq. (12), and
$$\mathrm{d}\boldsymbol{q} = \begin{cases} \mathrm{d}\ell(\boldsymbol{q}), & (\boldsymbol{q} \in \mathcal{B}_{\mathcal{M}}) \\ \mathrm{d}A(\boldsymbol{q}), & (\boldsymbol{q} \in \mathcal{B}_{\mathcal{V}} \setminus \mathcal{B}_{\mathcal{M}}) \end{cases} \qquad (26)$$
with ℓ being the curve-length measure;
- $\bar{\boldsymbol{p}}_+ := (\boldsymbol{p}_0, \ldots, \boldsymbol{p}_{N-1}, \boldsymbol{q})$ indicates the path obtained by appending the vertex $\boldsymbol{q}$ to $\bar{\boldsymbol{p}} = (\boldsymbol{p}_0, \ldots, \boldsymbol{p}_{N-1})$;
- $\hat{\Omega}_0 := \cup_{N=0}^{\infty} \mathcal{B}_{\mathcal{V}}^{N+1}$ denotes the space of material paths with at least one vertex (as opposed to the material path space $\hat{\Omega}$ comprised of paths with at least two vertices);
- $\Delta\mathcal{B}_{\mathcal{V}}[h]$ denotes all jump discontinuity points of h (with respect to the material point $\boldsymbol{q}$);
- $v(\boldsymbol{q})$ captures the scalar change rate (with respect to the scene parameter θ) of the discontinuity boundary along the normal direction (we will provide more details later);

Additionally,

$$\Delta W_{\mathrm{e}}^{\text{pinhole}}(\boldsymbol{y}) = \frac{G(\boldsymbol{y} \leftrightarrow \boldsymbol{x}_{\text{cam}})\, \Delta h(\boldsymbol{y}^{\perp})}{(\boldsymbol{n}_{\text{cam}} \cdot \overrightarrow{\boldsymbol{x}_{\text{cam}}\,\boldsymbol{y}})^3}, \qquad (27)$$

where Δh is the difference of the pixel reconstruction filter h across the discontinuity boundary. We note that the pixel *boundary* integral (25) only integrates over a subset of the full boundary path space $\partial\hat{\Omega}$ (see Figure 7 for per-component derivative visualizations).

We focus on estimating the pixel *boundary* integral (25), below. Contributions from other forms of boundary paths can be handled using previous methods like multi-directional sampling [Zhang et al. 2020, 2021b] and primary-sample-space guiding [Yan et al. 2022].

Eq. (25) behaves in a similar fashion as the so-called primary *boundary* integral that captures the discontinuities of the generalized geometric term $G(\boldsymbol{y} \leftrightarrow \boldsymbol{x}_{\text{cam}})$ emerging from object silhouettes

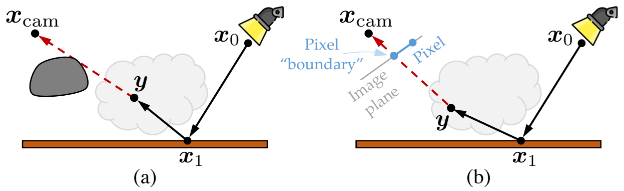


Fig. 8. Similarity between primary (a) and pixel *boundary* integrals (b): The former emerges from object silhouettes, and the latter from jump discontinuities of pixel reconstruction filters.

with respect to the camera (see Figure 8). Both integrals essentially capture discontinuities of the $W_e^{\text{pinhole}}(\boldsymbol{y})$ term defined in Eq. (15).

To efficiently estimate the pixel *boundary* integral (25), we first rewrite the outer integral as one over image-plane locations $\boldsymbol{y}^{\perp}$. Conceptually, this is similar to the handling of the primary *boundary* (Figure 8-a) by integrating over image-plane projections of object silhouettes [Li et al. 2018; Zhang et al. 2019, 2020, 2021b].

Specifically, as illustrated in Figure 9, we rewrite Eq. (25) as

$$\boxed{\int_{\Delta\mathcal{I}[h]}\left(F_{\text{B}}(\boldsymbol{q}_{\text{s}})\left\|\frac{\mathrm{d}\ell(\boldsymbol{q}_{\text{s}})}{\mathrm{d}\ell(\boldsymbol{y}^{\perp})}\right\| + \int_0^D F_{\text{B}}(\boldsymbol{q}_{\text{v}})\left\|\frac{\mathrm{d}A(\boldsymbol{q}_{\text{v}})}{\mathrm{d}r\,\mathrm{d}\ell(\boldsymbol{y}^{\perp})}\right\|\mathrm{d}r\right)\mathrm{d}\ell(\boldsymbol{y}^{\perp}),} \tag{28}$$

where:

- $\Delta\mathcal{I}[h]$ are curves comprised of jump discontinuities of the pixel reconstruction filter h with respect to image-plane points $\boldsymbol{y}^{\perp}$;
- $D \in \mathbb{R}_{>0}$ denotes the distance from the pinhole camera $\boldsymbol{x}_{\text{cam}}$ to the first surface intersection along the direction $\boldsymbol{\omega}_{\boldsymbol{y}} := \overrightarrow{\boldsymbol{x}_{\text{cam}}\,\boldsymbol{y}^{\perp}}$;
- $\boldsymbol{y}_{\text{s}} := \boldsymbol{x}_{\text{cam}} + D\,\boldsymbol{\omega}_{\boldsymbol{y}}$, and $\boldsymbol{q}_{\text{s}} = \mathsf{X}^{-1}(\boldsymbol{y}_{\text{s}}, \theta)$ that are surface vertices;
- $\boldsymbol{y}_{\text{v}} := \boldsymbol{x}_{\text{cam}} + r\,\boldsymbol{\omega}_{\boldsymbol{y}}$, and $\boldsymbol{q}_{\text{v}} = \mathsf{X}^{-1}(\boldsymbol{y}_{\text{v}}, \theta)$ that are volume vertices;

F_{B} is the integrand of Eq. (25) and can be estimated using standard forward-rendering methods like unidirectional path tracing. Additionally, the change-of-variable ratios are

$$\left\|\frac{\mathrm{d}\ell(\boldsymbol{q}_{\text{s}})}{\mathrm{d}\ell(\boldsymbol{y}^{\perp})}\right\| = \left\|\frac{\mathrm{d}\ell(\boldsymbol{y}_{\text{s}})}{\mathrm{d}\ell(\boldsymbol{y}^{\perp})}\right\|\left\|\frac{\mathrm{d}\ell(\boldsymbol{q}_{\text{s}})}{\mathrm{d}\ell(\boldsymbol{y}_{\text{s}})}\right\| = \frac{D\sin\beta^{\perp}}{\cos^{-1}\phi_{\boldsymbol{y}}\,\sin\beta_{\text{s}}}\left\|\frac{\mathrm{d}\ell(\boldsymbol{q}_{\text{s}})}{\mathrm{d}\ell(\boldsymbol{y}_{\text{s}})}\right\|, \tag{29}$$

and

$$\left\|\frac{\mathrm{d}A(\boldsymbol{q}_{\text{v}})}{\mathrm{d}r\,\mathrm{d}\ell(\boldsymbol{y}^{\perp})}\right\| = \left\|\frac{\mathrm{d}A(\boldsymbol{y}_{\text{v}})}{\mathrm{d}r\,\mathrm{d}\ell(\boldsymbol{y}^{\perp})}\right\|\left\|\frac{\mathrm{d}A(\boldsymbol{q}_{\text{v}})}{\mathrm{d}A(\boldsymbol{y}_{\text{v}})}\right\| = \frac{r\sin\beta^{\perp}}{\cos^{-1}\phi_{\boldsymbol{y}}}\left\|\frac{\mathrm{d}A(\boldsymbol{q}_{\text{v}})}{\mathrm{d}A(\boldsymbol{y}_{\text{v}})}\right\|, \tag{30}$$

where $\cos^{-1}\phi_{\boldsymbol{y}}$ is the distance from image-plane point $\boldsymbol{y}^{\perp}$ to $\boldsymbol{x}_{\text{cam}}$.

When estimating derivatives at $\theta = \theta_0$ with aforementioned configurations of reference surface and volume, the ratios $\|\mathrm{d}\ell(\boldsymbol{q}_{\text{s}})/\mathrm{d}\ell(\boldsymbol{y}_{\text{s}})\|$ and $\|\mathrm{d}A(\boldsymbol{q}_{\text{v}})/\mathrm{d}A(\boldsymbol{y}_{\text{v}})\|$ in Eqs. (29) and (30) reduce to one.

Calculating scalar change rates. We discuss the computation of the scalar change rate $v(\boldsymbol{q})$ in the formulation of Eq. (28), a key component in the integrand $F_{\text{B}}(\boldsymbol{q})$ of the pixel *boundary* integral.

Given a image-plane location $\boldsymbol{y}^{\perp}$ that is a jump discontinuity point of the pixel reconstruction filter h, we trace a camera ray $\boldsymbol{x}_{\text{cam}} \to \boldsymbol{y}^{\perp}$ and compute the intersection distance $D \in \mathbb{R}_{>0}$ and location $\boldsymbol{y}_{\text{s}} \in \mathcal{M}$ in a *differentiable* fashion. The spatial points $\boldsymbol{x}_{\text{cam}}$ and $\boldsymbol{y}^{\perp}$ both depend on the scene parameter θ via the material-form parameterization discussed in §3.2.

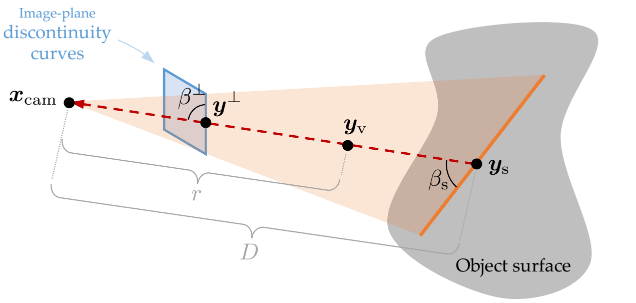


Fig. 9. The pixel boundary integral in Eq. (25) involves an outer integral over jump discontinuities of the pixel reconstruction filter h with respect to material point $\boldsymbol{q}$. The triangle and curve marked in orange illustrate (the spatial counterpart of) some such points. We rewrite the outer integral as (i) a line integral over discontinuity curves $\Delta\mathcal{I}[h]$ of h with respect to image-plane position $\boldsymbol{y}^{\perp}$ (blue rectangle) and (ii) another integral over the line segment (dashed red line) connecting the camera $\boldsymbol{x}_{\text{cam}}$ and the (first) surface intersection $\boldsymbol{y}_{\text{s}}$ of the camera ray $\boldsymbol{x}_{\text{cam}} \to \boldsymbol{y}^{\perp}$.

Let $\boldsymbol{q}_{\text{s}} = \mathsf{X}^{-1}(\boldsymbol{y}_{\text{s}}, \theta)$ and note that, despite being a material point, $\boldsymbol{q}_{\text{s}}$ still depends on θ in general as its spatial counterpart $\boldsymbol{y}_{\text{s}}$ is obtained by differentiable ray intersection. Provided $\boldsymbol{q}_{\text{s}}$, the scalar change rate is $v(\boldsymbol{q}_{\text{s}}) = \boldsymbol{n}(\boldsymbol{q}_{\text{s}}) \cdot (\mathrm{d}\boldsymbol{q}_{\text{s}}/\mathrm{d}\theta)$, where $\boldsymbol{n}(\boldsymbol{q}_{\text{s}}) \in \mathbb{S}^2$ is the unit normal of the discontinuity curve at $\boldsymbol{q}_{\text{s}}$. This curve resides on the reference surface $\mathcal{B}_{\mathcal{M}}$ and, when estimating derivatives at $\theta = \theta_0$ with $\mathcal{B}_{\mathcal{M}} = \mathcal{M}(\theta_0)$, coincides with the orange line in Figure 9.

Let $\boldsymbol{y}_{\text{v}} = (1-\xi)\,\boldsymbol{x}_{\text{cam}} + \xi\,\boldsymbol{y}_{\text{s}}$ with $\xi = r/D$ treated as a constant. Similar to $\boldsymbol{q}_{\text{s}}$, $\boldsymbol{q}_{\text{v}} = \mathsf{X}^{-1}(\boldsymbol{y}_{\text{v}}, \theta)$ is a spatial point depending on the scene parameter θ. Then, the scalar change rate is $v(\boldsymbol{q}_{\text{v}}) = \boldsymbol{n}(\boldsymbol{q}_{\text{v}}) \cdot (\mathrm{d}\boldsymbol{q}_{\text{v}}/\mathrm{d}\theta)$, where $\boldsymbol{n}(\boldsymbol{q}_{\text{v}}) \in \mathbb{S}^2$ denotes the unit normal of the discontinuity surface at $\boldsymbol{q}_{\text{v}}$. When estimating derivatives at $\theta = \theta_0$ with the reference volume setting to $\mathcal{B}_{\mathcal{V}} = \mathcal{V}(\theta_0)$, this surface coincides with the orange triangle illustrated in Figure 9.

Antithetic sampling on pixel discontinuities. Although the Eq. (28) do not involve derivatives of the measurement contribution $\hat{f}$, we make a key observation that the estimation of this term can still benefit significantly from pixel-level antithetic sampling if the image-plane discontinuity boundary $\Delta\mathcal{I}[h]$ exhibit the same types of symmetries as the pixel reconstruction filter itself. In practice, this is the case for many common used (e.g., box) filters.

Specifically, as shown in Figure 10, we sample image-plane points $\boldsymbol{y}^{\perp}$, $\boldsymbol{y}_1^{\perp}$, $\boldsymbol{y}_2^{\perp}$ and $\boldsymbol{y}_3^{\perp}$ from discontinuity curves $\Delta\mathcal{I}[h]$ following the same antithetic sampling pattern for the *interior* integral. Then, we construct the corresponding path vertices $\boldsymbol{y}$, $\boldsymbol{y}_1^*$, $\boldsymbol{y}_2^*$ and $\boldsymbol{y}_3^*$—which, in turn, gives full boundary paths $\bar{\boldsymbol{p}}$, $\bar{\boldsymbol{p}}_1^*$, $\bar{\boldsymbol{p}}_2^*$ and $\bar{\boldsymbol{p}}_3^*$—in a similar fashion as the *interior* case discussed in §5.1.

Since $\boldsymbol{y}^{\perp}$ and $\boldsymbol{y}_3^{\perp}$ are point symmetric, the normals of $\Delta\mathcal{I}[h]$ at $\boldsymbol{y}^{\perp}$ and $\boldsymbol{y}_3^{\perp}$ are in opposite directions. This implies that, when the scene geometry to vary slowly within the pixel, $\Delta W_e^{\text{pinhole}}(\boldsymbol{y})\,v(\boldsymbol{q}) \approx -\Delta W_e^{\text{pinhole}}(\boldsymbol{y}_3^*)\,v(\boldsymbol{q}_3^*)$, where $\boldsymbol{q}$ and $\boldsymbol{q}_3^*$ are the spatial representations of $\boldsymbol{y}$ and $\boldsymbol{y}_3^*$, respectively. It follows that the contributions of

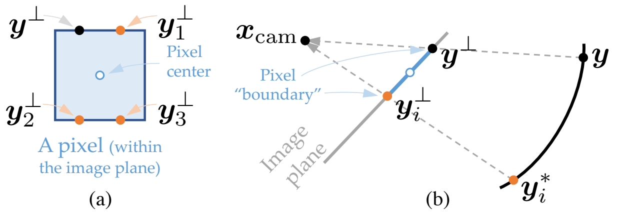


Fig. 10. **Pixel-level antithetic sampling (boundary):** (a) We draw image-plane points $\boldsymbol{y}^\perp$, $\boldsymbol{y}_1^\perp$, $\boldsymbol{y}_2^\perp$ and $\boldsymbol{y}_3^\perp$ from discontinuity curves of the pixel reconstruction filter following the same antithetic sampling pattern as the *interior* case illustrated in Figure 3. (b) For each $i = 1, 2, 3$, we obtain $\boldsymbol{y}_i^\perp$ on the image plane based on $\boldsymbol{y}^\perp$ and trace a camera ray through $\boldsymbol{y}_i^\perp$ to obtain the vertex $\boldsymbol{y}_i^*$ in a similar fashion as the process illustrated in Figure 5.

corresponding *boundary* paths $\bar{\boldsymbol{p}}$ and $\bar{\boldsymbol{p}}_3^*$ (and, similarly, the paths $\bar{\boldsymbol{p}}_1^*$ and $\bar{\boldsymbol{p}}_2^*$) to largely cancel each other out, producing lowered variance.

7 RESULTS

We developed a CPU-based differentiable renderer that utilizes Enzyme [Moses and Churavy 2020] for reverse-mode automatic differentiation (AD). Contrary to most AD libraries that work at the source code (e.g., C++) level, Enzyme operates at the LLVM level taking arbitrary code as input in LLVM Intermediate Representation and code generating gradient routines of its functions.

Specifically, we use LLVM to compile our C++ source code (that contains Enzyme-specific intrinsics) into LLVM's Intermediate Representation (IR; Figure 11). Then, we use Enzyme to automatically differentiate the LLVM IR and, in turn, compile the output into executables using LLVM. Since the differentiation occurs at the LLVM level and is integrated with LLVM's powerful optimization pipeline, the resulting system offers high performance and scalability.

Moreover, we develop a slightly different pipeline (also based on Enzyme) to generate gradient images with respect to one scene parameter. We use this pipeline *exclusively* for debugging and validation purposes, i.e., not for solving inverse-rendering problems.

7.1 Validation and Evaluation

Filter settings. To evaluate our technique, we use two pixel reconstruction filters: *tent* and *box*. Assuming square pixels $\mathcal{P}$ in the image plane $\mathcal{I}$ with edge length d (and surface area d^2), we have

$$h_{\text{tent}}(\boldsymbol{y}^\perp) := \frac{1}{d^2} \max\left(1 - \frac{|y_s^\perp|}{d}, 0\right) \max\left(1 - \frac{|y_t^\perp|}{d}, 0\right), \tag{31}$$

$$h_{\text{box}}(\boldsymbol{y}^\perp) := \frac{\mathbb{1}\left[-d/2 < y_s^\perp < d/2 \quad \text{and} \quad -d/2 < y_t^\perp < d/2\right]}{d^2}, \tag{32}$$

where $\mathbb{1}[\cdot]$ is the indicator function. Additionally, $y_s^\perp := (\boldsymbol{y}^\perp - \boldsymbol{c}) \cdot \boldsymbol{s}$ and $y_t^\perp := (\boldsymbol{y}^\perp - \boldsymbol{c}) \cdot \boldsymbol{t}$ where $\boldsymbol{c} \in \mathcal{I}$ denotes the center point of the pixel, and $\boldsymbol{s}, \boldsymbol{t} \in \mathbb{S}^2$ indicate the horizontal and vertical axes of the image plane, respectively. We intentionally let the supports of the tent filter from neighboring pixels to overlap, allowing the sum $\sum_i h_{\text{tent}}^{(i)}$ over all pixels to be constant across the image plane.

Given Eqs. (31) and (32), it is easy to verify that both filters integrate to one (on the image plane) and have discontinuities $\Delta\mathcal{I}[h_{\text{tent}}] = \emptyset$ and $\Delta\mathcal{I}[h_{\text{box}}] = \partial\mathcal{P}$, respectively.

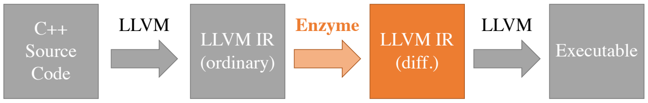


Fig. 11. Our system utilizes the Enzyme [Moses and Churavy 2020] automatic differentiation framework that operates at the LLVM level by generating differentiable versions of LLVM Intermediate Representations (IR).

Table 1. Performance statistics for our inverse-rendering results. The "DR" numbers indicate per-iteration computation time (conducted on a workstation with an AMD Ryzen 3950X 16-core CPU) for estimating image-loss gradients using our method; and "post." captures the cost for updating mesh vertex positions using the large-step method [Nicolet et al. 2021].

Scene		# Target	# Param.	# Iter.	DR	Post.
World map	(Fig. 1)	13	6 496 256	300	20.4s	2.3s
Bunny 1	(Fig. 14)	20	29 844	60	2.0s	0.2s
Kitty 2	(Fig. 14)	20	300 123	200	11.0s	2.0s
Terrain	(Fig. 15)	5	196 608	200	1.1s	0.6s
Pig 2	(Fig. 15)	20	599 733	200	2.7s	5.0s

Differentiable-rendering comparisons. We now evaluate our technique using differentiable-rendering results.

In Figure 12, we show results using the *tent* reconstruction filter (31). Since this filter is continuous (i.e., has no jump discontinuities), the pixel *boundary* integral vanishes. We apply our pixel-level antithetic sampling for the *interior* integral. Further, we use adjoint particle tracing (APT) and bidirectional path tracing (BDPT) to sample light transport paths. As discussed in §5, previous pixel-level antithetic sampling [Zhang et al. 2021b] is designed mostly for unidirectional path tracing and does not fully support methods like APT and BDPT that trace shadow rays toward the camera.

The "*earth*" and "*bust*" examples in Figure 12 contain, respectively, a textured diffuse earth object and a homogeneous translucent bust. Both examples are rendered using APT. Additionally, the "*kitty 1*" result involves a kitty object [Zhou et al. 2016] inside a Cornell box lit by an area light facing toward the ceiling, creating a indirect-illumination-dominated situation. The "*glass ball*" example contains a rough glass sphere lit by a small area light. Since both "*kitty 1*" and "*glass ball*" involve light-transport configurations known to be challenging for unidirectional methods, we render both examples using BDPT.

In all four examples, derivatives obtained with our method agree with references computed using finite differences (in much longer time). Meanwhile, our results are significantly cleaner than the baseline ones computed in equal time without our antithetic sampling.

In Figure 13, we show results using the *box* filter (32) that is not supported by previous path-space methods [Zhang et al. 2021b] since they require reconstruction filters to be continuous. As the box filter is constant in the interior, we apply our antithetic technique to the pixel *boundary* integral only.

In this figure, the "*wheel*" example shows a disc with the color wheel pattern. The "*pig 1*" and "*duck*" results use textured duck [Dong

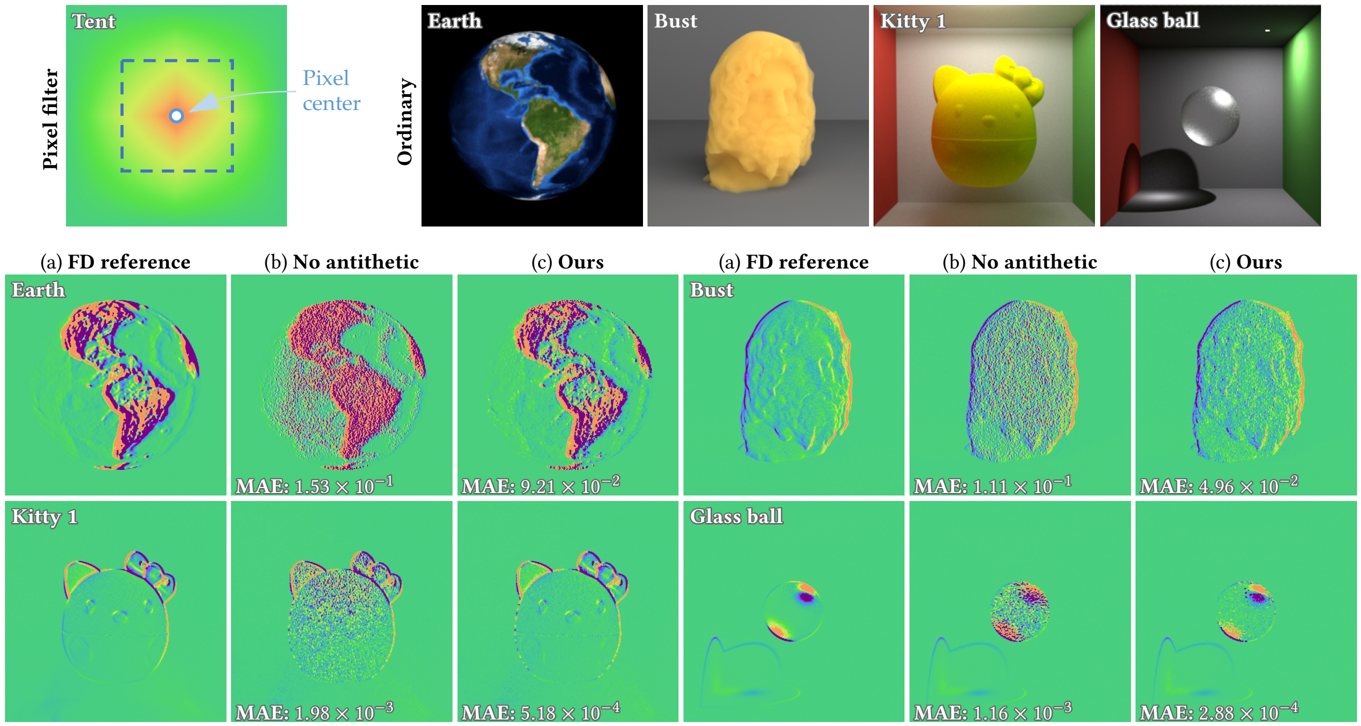

Tent
Pixel filter
Pixel center
Ordinary
Earth
Bust
Kitty 1
Glass ball
(a) FD reference
(b) No antithetic
(c) Ours
(a) FD reference
(b) No antithetic
(c) Ours
Earth
MAE: 1.53×10^{-1}
MAE: 9.21×10^{-2}
Bust
MAE: 1.11×10^{-1}
MAE: 4.96×10^{-2}
Kitty 1
MAE: 1.98×10^{-3}
MAE: 5.18×10^{-4}
Glass ball
MAE: 1.16×10^{-3}
MAE: 2.88×10^{-4}

Fig. 12. **Antithetic sampling (*interior*):** To demonstrate the effectiveness of our pixel-level antithetic sampling of *interior* integrals (§5), we show comparisons of derivative estimates (visualized using the same color map as Figure 4) with respect to object translations. Results in (b) and (c) are generated at equal time without and with our antithetic sampling, respectively. The references in (a) are generated using finite differences in much longer time. The *earth* and the *bust* examples are generated with volumetric adjoint particle tracing (APT); the *kitty* and the *glass ball* use bidirectional path tracing (BDPT). All examples use the tent pixel reconstruction filter (31).

et al. 2014] and pig [Zhou et al. 2016] objects. The "*jumpy dumpy*" example contains an object filled with homogeneous participating medium. We use unidirectional path tracing to estimate the *interior* integrals.

For all examples, derivative estimates obtained using our method agree well with the finite-difference references while containing significantly lower variance than the baseline results.

7.2 Inverse-Rendering Results

To further demonstrate the effectiveness of our technique, we compare inverse-rendering results with and without our pixel-level antithetic sampling. For each comparison, we use identical optimization configurations including initial states, learning rates, and optimizers. Additionally, we set the sample counts so that each inverse-rendering iteration (with and without our technique) takes approximately equal time. We use the large-step method [Nicolet et al. 2021] for robust updates of object shapes.

Please see Table 1 for performance statistics and the supplemental material for animated versions of these results.

Figure 14 shows inverse-rendering results using the tent reconstruction filter (31). The "*bunny 1*" example includes a translucent bunny under area lighting. The "*kitty 2*" result uses a configuration similar to that in Figure 12 with a Cornell box containing a kitty and an area light facing the ceiling. We render the *interior* components of the two examples using unidirectional and bidirectional path tracing, respectively. For both examples, we optimize the object shapes by minimizing the image loss. The reduced variance offered by our antithetic sampling technique has allowed both optimizations to converge more quickly, resulting in reconstructed geometries with lower error (measured by Chamfer distances [Barrow et al. 1977]).

Figure 15 contains results using the box reconstruction filter (32). The "*terrain*" example includes a terrain-like glossy surface. The "*pig 2*" result contains a pig model [Zhou et al. 2016] with detailed structures on the back. Using highly smooth initial shapes, we optimize the object geometries by minimizing image losses. For both examples, our technique allows the inverse-rendering optimizations to converge nicely. Without our sampling, on the other hand, the resulting gradients are very noisy, which can cause the mesh to self-intersect (even with the large-step method [Nicolet et al. 2021] applied).

Lastly, using the "*pig 2*" scene, we demonstrate in Figure 16 the advantage of our four-point antithetic sampling pattern (i.e., using $\boldsymbol{y}^\perp$, $\boldsymbol{y}_1^\perp$, $\boldsymbol{y}_2^\perp$, and $\boldsymbol{y}_3^\perp$ illustrated in Figure 10) over the two-point one (i.e., using only $\boldsymbol{y}^\perp$ and $\boldsymbol{y}_3^\perp$) for inverse rendering. Since the pig model's detailed surface geometry produces many "edges" in rendered images, the four-point pattern—which is more robust at these edges—allows the inverse-rendering optimization (at equal time per iteration) to converge more quickly.

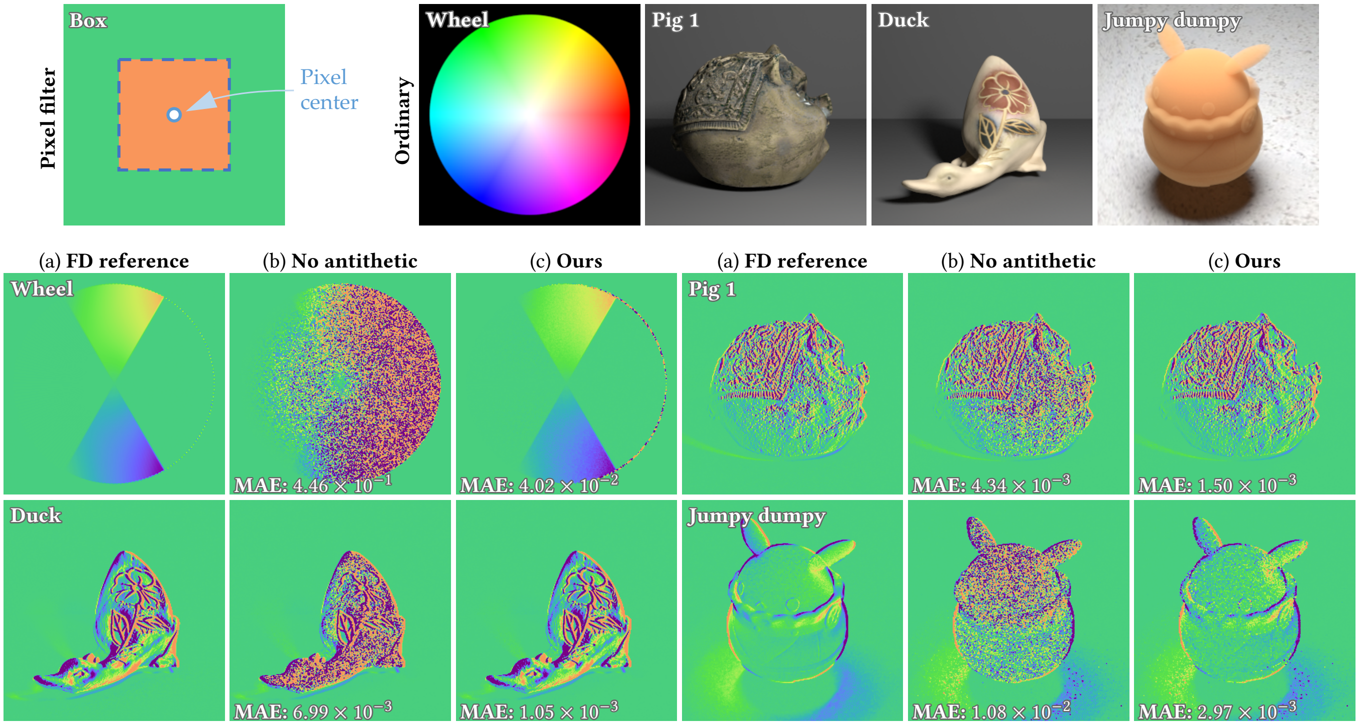


Fig. 13. **Antithetic sampling (*boundary*):** To demonstrate the effectiveness of our pixel-level antithetic sampling of *boundary* integrals (§6), we show comparisons of derivative estimates (visualized using the same color map as Figure 4) with respect to object rotation (a) and translations (b–d). Results in (b) and (c) are generated in equal time without and with our antithetic sampling, respectively. The references in (a) are generated using finite differences in much longer time. The *interior* components of all results are estimated using unidirectional path tracing. All examples use the box pixel reconstruction filter (32).

Please refer to the supplemental material for more examples.

7.3 Additional Experiments

Tent vs. box filters. Strictly, derivatives obtained with tent and box filters are distinct. On the other hand, the differences are typically at sub-pixel levels and quite subtle.

For applications where both tent and box filters can be used, we observed that the box one typically has lower variance at equal time, despite requiring an extra pixel *boundary* integral estimate. This is due to two reasons: first, the box filter of Eq. (31) is constant in the interior of a pixel, allowing the *interior* component to be estimated using relatively few samples due to its smooth and low contribution (as shown in Figure 7-c); second, compared with the *interior* integral that requires differentiating full measurement contributions, the pixel *boundary* integral (25) is significantly less expensive to estimate because its integrand F_B only has one component—the change rate $v(\boldsymbol{q})$—that requires differentiation.

We demonstrate this using two examples in Figure 17. The "*pig 3*" result has a Cornell box containing a pig model. The "*jumpy dumpy*" example uses the same scene as the one in Figure 13. In both cases, using the box filter leads to notably lower variance.

Anisotropic Gaussian filters. Another type of filters commonly used in adaptive sampling and reconstruction (e.g., [Mehta et al. 2012]) is axis-aligned anisotropic Gaussians that take the form of

$$h_\mathrm{gauss}(\boldsymbol{y}^\perp) = \frac{1}{2\pi d^2 \alpha_\mathrm{s}\, \alpha_\mathrm{t}} \exp\left(-\frac{(y_\mathrm{s}^\perp)^2}{2d^2\alpha_\mathrm{s}^2} - \frac{(y_\mathrm{t}^\perp)^2}{2d^2\alpha_\mathrm{t}^2}\right), \tag{33}$$

where $y_\mathrm{s}^\perp$ and $y_\mathrm{t}^\perp$ are defined the same way as in Eq. (31), and $\alpha_\mathrm{s}, \alpha_\mathrm{t} > 0$ are the filter's standard deviations along the image plane's horizontal and vertical axes (i.e., $\boldsymbol{s}$ and $\boldsymbol{t}$), respectively.

Our antithetic sampling technique also benefits anisotropic Gaussian filters, which we demonstrate in Figure 18 using three examples. All these example share the same scene containing a translucent bunny (similar to "*bunny 1*" in Figure 14) but use different per-pixel filter parameters α_s and α_t visualized below the corresponding ordinary renderings. Further, the first example ("*config 1*") uses smoothly varying parameters increasing from the center toward the edge of the image; the second result ("*config 2*") has checker-board patterns for both parameters; and the third example ("*config 3*") utilizes depth-driven parameter maps, mimicking a depth-of-field effect. Thanks to our antithetic sampling, for all examples, our results enjoy significantly lower noise compared with the baseline ones.

8 DISCUSSION AND CONCLUSION

Limitations and future work. When handling volumetric light transport, our method works best for homogeneous participating media where equal-transmittance constructions can be performed

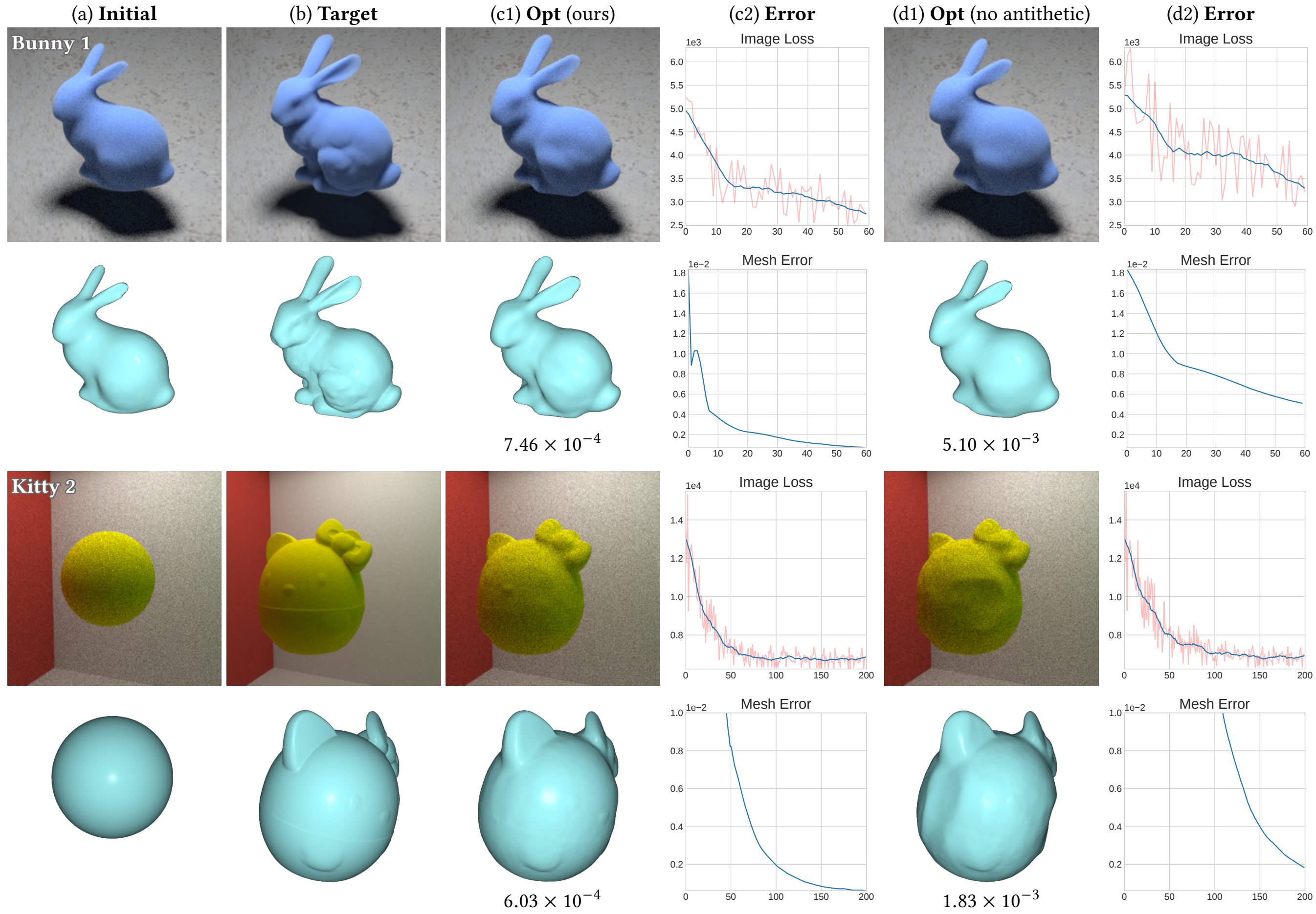


Fig. 14. **Antithetic sampling (*interior*):** We show inverse-rendering comparisons of derivatives estimated with (c) and without (d) our *interior* antithetic sampling (presented in §5). All examples use the tent pixel reconstruction filter (31). The mesh error numbers shown under visualized geometries in (c1) and (d1) as well as plotted in (c2) and (d2) indicate the Chamfer distances [Barrow et al. 1977] between the reconstructed and groundtruth geometries (normalized so that the GT has a unit bounding box). We use this information only for evaluation (and not for optimization).

efficiently. Better handling heterogeneous media where transmittance cannot be easily estimated or inverted is worth exploring. Further, our technique uses pixel reconstruction filers that depend only on image-plane positions; investigating other types of filers, such as bilateral filters that take intensities into account, is an interesting future topic. Lastly, since our technique focuses on pixel filers, combining it with BSDF antithetic sampling [Zhang et al. 2021a] will be useful for future differentiable rendering systems.

Conclusion. We systematically studied the treatment, and importance, of pixel reconstruction filters for path-space differentiable rendering. Specifically, we devised the pixel *boundary* integral to capture jump discontinuities of pixel filters. Furthermore, we introduced a new Monte Carlo antithetic sampling method to efficiently estimate both the *interior* and *boundary* integrals. Our technique supports both surface and volume (differentiable) light transport and is applicable to both primal and adjoint path sampling scenarios.

We evaluated the effectiveness of our technique by comparing against baseline methods on several differentiable-rendering and inverse-rendering experiments.

ACKNOWLEDGMENTS

Zihan Yu and Cheng Zhang were partially supported by NSF grant 1900927 and Meta Reality Labs.

REFERENCES

Mahdi M. Bagher, John M. Snyder, and Derek Nowrouzezahrai. 2016. A non-parametric factor microfacet model for isotropic BRDFs. *ACM Trans. Graph.* 35, 5 (2016), 159:1–159:16.

Mahdi Mohammad Bagher, Cyril Soler, Kartic Subr, Laurent Belcour, and Nicolas Holzschuch. 2012. Interactive rendering of acquired materials on dynamic geometry using frequency analysis. *IEEE Transactions on Visualization and Computer Graphics* 19, 5 (2012), 749–761.

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.

Harry G Barrow, Jay M Tenenbaum, Robert C Bolles, and Helen C Wolf. 1977. Parametric correspondence and chamfer matching: Two new techniques for image matching. In *Proceedings: Image Understanding Workshop.* 21–27.

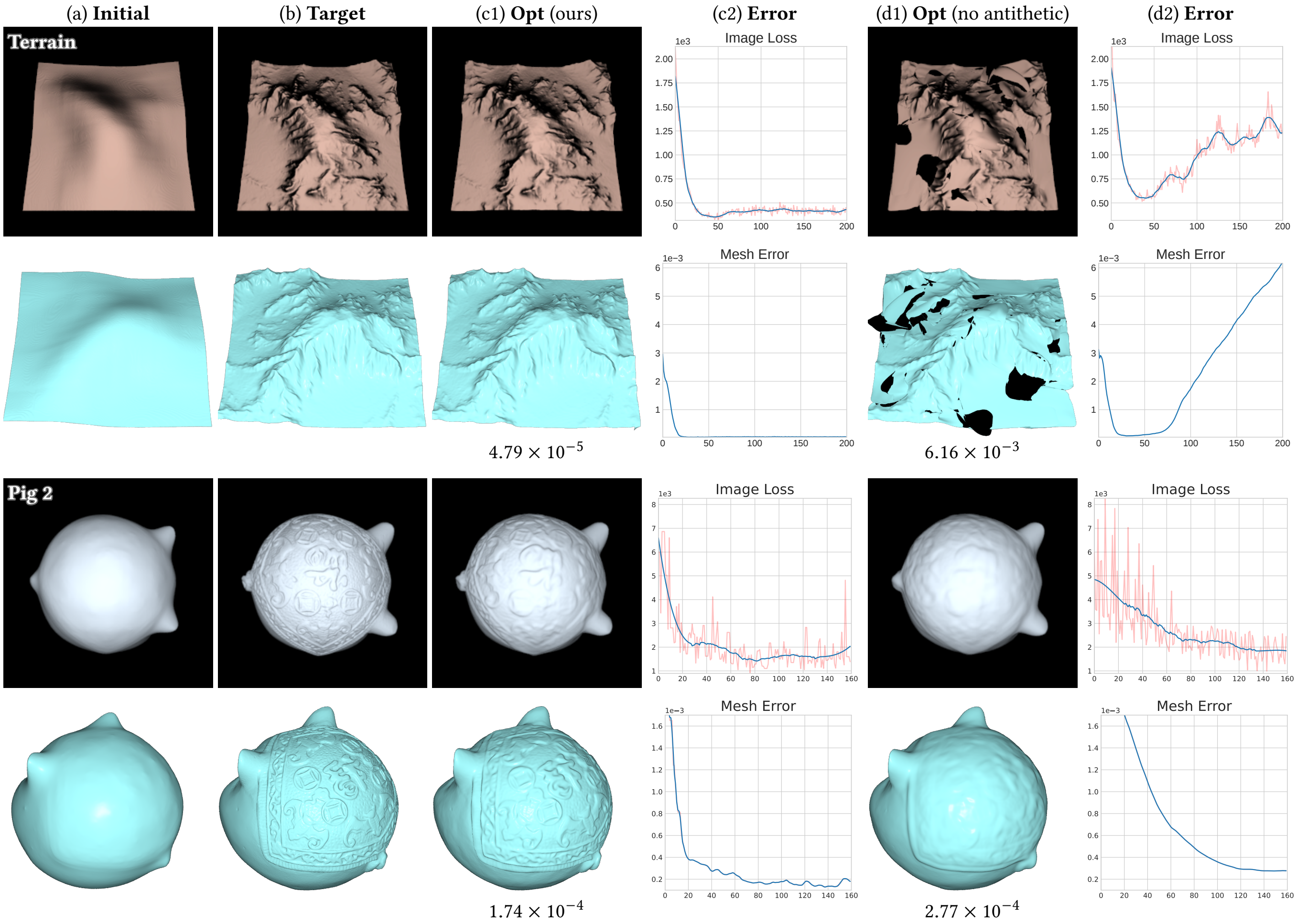


Fig. 15. **Antithetic sampling (*boundary*):** We show inverse-rendering comparisons of derivatives estimated with (c) and without (d) our *boundary* antithetic sampling (discussed in §6). All examples use the box pixel reconstruction filter (32) unsupported by previous path-space methods [Zhang et al. 2021b]. The mesh error numbers shown under visualized geometries in (c1) and (d1) as well as plotted in (c2) and (d2) indicate the Chamfer distances [Barrow et al. 1977] between the reconstructed and groundtruth geometries. We use this information for evaluation only.

Laurent Belcour, Kavita Bala, and Cyril Soler. 2014. A local frequency analysis of light scattering and absorption. *ACM Trans. Graph.* 33, 5 (2014), 163:1–163:17.

Laurent Belcour, Cyril Soler, Kartic Subr, Nicolas Holzschuch, and Fredo Durand. 2013. 5D covariance tracing for efficient defocus and motion blur. *ACM Trans. Graph.* 32, 3 (2013), 31:1–31:18.

Laurent Belcour, Ling-Qi Yan, Ravi Ramamoorthi, and Derek Nowrouzezahrai. 2017. Antialiasing complex global illumination effects in path-space. *ACM Trans. Graph.* 36, 1 (2017), 9:1–9:13.

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

Robert L Cook and Kenneth E. Torrance. 1982. A reflectance model for computer graphics. *ACM Transactions on Graphics (ToG)* 1, 1 (1982), 7–24.

Yue Dong, Guojun Chen, Pieter Peers, Jiawan Zhang, and Xin Tong. 2014. Appearance-from-Motion: Recovering Spatially Varying Surface Reflectance under Unknown Lighting. *ACM Trans. Graph.* 33, 6 (2014), 193:1–193:12.

Frédo Durand, Nicolas Holzschuch, Cyril Soler, Eric Chan, and François X. Sillion. 2005. A frequency analysis of light transport. *ACM Trans. Graph.* 24, 3 (2005), 1115–1126.

Kevin Egan, Yu-Ting Tseng, Nicolas Holzschuch, Frédo Durand, and Ravi Ramamoorthi. 2009. Frequency analysis and sheared reconstruction for rendering motion blur. In *ACM SIGGRAPH 2009 Papers (SIGGRAPH '09)*. Article 93, 13 pages.

Eric Heitz and Eugene d'Eon. 2014. Importance sampling microfacet-based BSDFs using the distribution of visible normals. In *Computer Graphics Forum*, Vol. 33. Wiley Online Library, 103–112.

Eric Heitz, Johannes Hanika, Eugene d'Eon, and Carsten Dachsbacher. 2016. Multiple-scattering microfacet BSDFs with the Smith model. *ACM Trans. Graph.* 35, 4 (2016), 58:1–58:14.

Csaba Kelemen and Laszlo Szirmay-Kalos. 2001. A microfacet based coupled specular-matte BRDF model with importance sampling. In *Eurographics short presentations*, Vol. 2. 4.

Markus Kettunen, Eugene D'Eon, Jacopo Pantaleoni, and Jan Novák. 2021. An unbiased ray-marching transmittance estimator. *ACM Trans. Graph.* 40, 4 (2021), 137:1–137:20.

Joo Ho Lee, Adrian Jarabo, Daniel S. Jeon, Diego Gutierrez, and Min H. Kim. 2018. Practical multiple scattering for rough surfaces. *ACM Trans. Graph.* 37, 6 (2018), 275:1–275:12.

Jaakko Lehtinen, Tero Karras, Samuli Laine, Miika Aittala, Frédo Durand, and Timo Aila. 2013. Gradient-Domain Metropolis Light Transport. *ACM Trans. Graph.* 32, 4 (2013), 95:1–95:12.

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 Transactions on Graphics (TOG)* 38, 6 (2019), 1–14.

Soham Uday Mehta, Brandon Wang, and Ravi Ramamoorthi. 2012. Axis-aligned filtering for interactive sampled soft shadows. *ACM Trans. Graph.* 31, 6 (2012), 163:1–163:10.

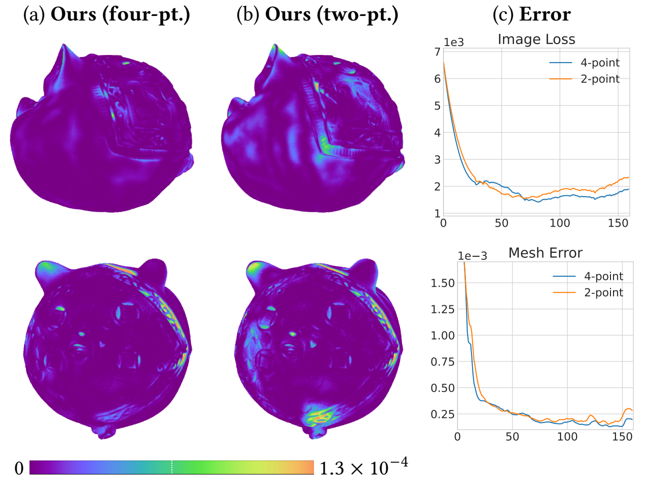


Fig. 16. **Two-point vs. four-point patterns:** Our four-point antithetic sampling pattern can produce lower variance at object edges, leading to more accurate reconstructions. In columns (a) and (b), using the "*pig 2*" scene, we visualize the distance from each surface point on the reconstructed meshes to its projection on the groundtruth geometry under two camera views. The mesh errors plotted in (c) are for evaluation only.

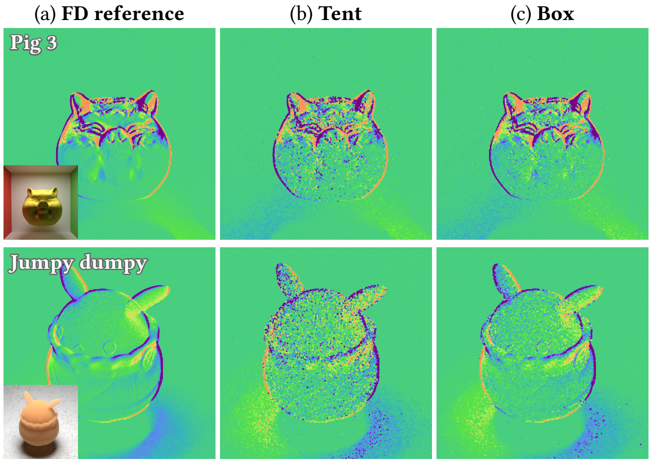


Fig. 17. **Tent vs. box filters:** We compare derivative images using tent (b) and box (c) filters. The results in (b) and (c) are generated in equal time using our pixel-level antithetic sampling. The references in (a) are generated using finite differences in significantly longer time. Despite requiring to estimate one more term (i.e., the pixel *boundary* integral), derivative estimates using the box filter have lower variance. The ordinary renderings are shown as insets in (a).

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.

Baptiste Nicolet, Alec Jacobson, and Wenzel Jakob. 2021. Large Steps in Inverse Rendering of Geometry. *ACM Trans. Graph.* 40, 6 (2021), 248:1–248:13.

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

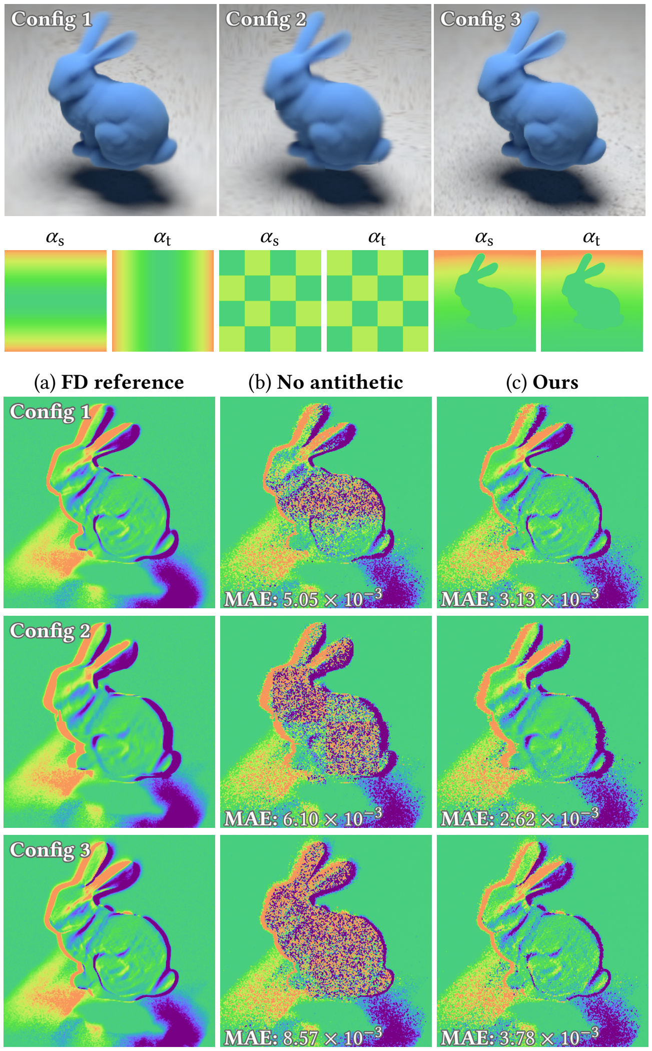

Config 1
Config 2
Config 3
α_s
α_t
α_s
α_t
α_s
α_t
(a) FD reference
(b) No antithetic
(c) Ours
Config 1
MAE: 5.05×10^{-3}
MAE: 3.13×10^{-3}
Config 2
MAE: 6.10×10^{-3}
MAE: 2.62×10^{-3}
Config 3
MAE: 8.57×10^{-3}
MAE: 3.78×10^{-3}

Fig. 18. **Anisotropic Gaussian filters:** Besides the tent and box filters, our technique can also benefit others that exhibit point symmetry, such as anisotropic Gaussians (33). In this figure, we show three examples sharing a scene similar to "*bunny 1*" in Figure 14 but with three distinct filter configurations. Below each ordinary rendering on the top, we visualize the corresponding per-pixel anisotropic Gaussian parameters α_s and α_t. The derivative images in (b) and (c) and generated in equal time, and the FD references in (a) are created in much longer time.

Christophe Schlick. 1994. An inexpensive BRDF model for physically-based rendering. In *Computer graphics forum*, Vol. 13. Wiley Online Library, 233–246.

Cyril Soler, Kartic Subr, Frédo Durand, Nicolas Holzschuch, and François Sillion. 2009. Fourier depth of field. *ACM Trans. Graph.* 28, 2 (2009), 18:1–18:12.

Bram van Ginneken, Marigo Stavridi, and Jan J Koenderink. 1998. Diffuse and specular reflectance from rough surfaces. *Applied optics* 37, 1 (1998), 130–139.

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

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

Bruce Walter, Stephen R Marschner, Hongsong Li, and Kenneth E Torrance. 2007. Microfacet models for refraction through rough surfaces. *Rendering techniques* 2007 (2007), 18th.

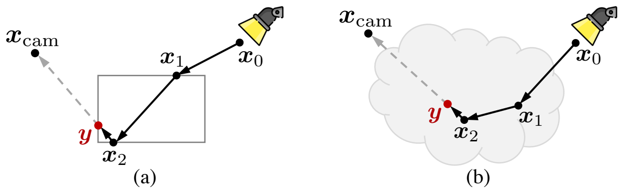


Fig. 19. **Problem of pixel-level attached sampling:** For any light path $\bar{x} = (x_0, \ldots, x_{N-1}, y)$, when the last vertex y is drawn by applying attached sampling to the pixel reconstruction filter, the change rate $^{\mathrm{d}y}/_{\mathrm{d}\theta}$ is determined by the sampling process and differentiable ray intersection. On the other hand, the change rates of other vertices $x_0, \ldots, x_{N-1}$ are given by the material-form parameterization (discussed in §3.2). When the distance between y and the neighboring vertex x_{N-1} (e.g., x_2 in this figure) approaches to zero, which can occur at object edges/corners (a) and in the interior of participating media (b), the change rates $^{\mathrm{d}y}/_{\mathrm{d}\theta}$ and $^{\mathrm{d}x_{N-1}}/_{\mathrm{d}\theta}$ remain different, leading to highly noisy derivative estimates.

Gregory J Ward. 1992. Measuring and modeling anisotropic reflection. In *Proceedings of the 19th annual conference on Computer graphics and interactive techniques*. 265–272.

Feng Xie and Pat Hanrahan. 2018. Multiple scattering from distributions of specular V-grooves. *ACM Trans. Graph.* 37, 6 (2018), 276:1–2767:14.

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.

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.

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.

Zhiming Zhou, Guojun Chen, Yue Dong, David Wipf, Yong Yu, John Snyder, and Xin Tong. 2016. Sparse-as-Possible SVBRDF Acquisition. *ACM Trans. Graph.* 35, 6 (2016), 189:1–189:12.

Matthias Zwicker, Wojciech Jarosz, Jaakko Lehtinen, Bochang Moon, Ravi Ramamoorthi, Fabrice Rousselle, Pradeep Sen, Cyril Soler, and S-E Yoon. 2015. Recent advances in adaptive sampling and reconstruction for Monte Carlo rendering. *Computer graphics forum* 34, 2 (2015), 667–681.

A PROBLEMS OF ALTERNATIVE METHODS

We now discuss two alternative methods for differentiating pixel reconstruction filters and their limitations. Specifically, we will present attached sampling [Zeltner et al. 2021] in §A.1 and differentiating through filter normalization [Vicini et al. 2022] in §A.2.

A.1 Attached Sampling

Assuming that the pixel reconstruction filter can be perfectly importance sampled, the pixel intensity I can also be expressed in primary-sample-space as

$$I = \int_{[0,1)^2} c\, L_i(x_{cam}(\theta) \to y^{\perp}(\xi; \theta))\, \mathrm{d}\xi, \tag{34}$$

where c is a normalization factor, L_i the incident radiance that can in turn be estimated using path integrals (10), and $y^{\perp} : [0,1)^2 \mapsto \mathcal{I}$ a mapping that encodes the importance sampling (of the pixel reconstruction filter) and transforms a primary sample $\xi \sim U[0,1)^2$ to an image plane point $\mathcal{I}$. We note that, when the scene evolves with a parameter θ, the mapping $y^{\perp}$ generally depends on θ and, thus, needs to be differentiated when estimating derivatives of Eq. (34). This formulation is also known as *attached sampling* [Zeltner et al. 2021] (applied to the pixel reconstruction filter h) and has been used previously by Zhang et al. [2019].

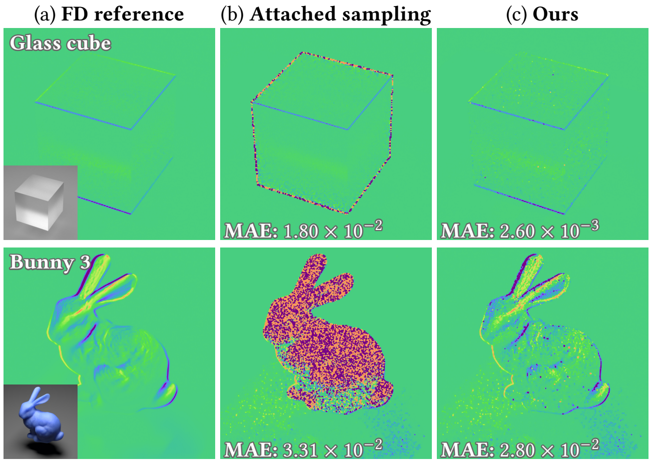


Fig. 20. **Equal-time comparison** of derivative images estimated with the vertex y drawn using pixel-level attached sampling (b) and our antithetic sampling (c). All results use the tent reconstruction filter (31) and are rendered with differentiable unidirectional path tracing. The ordinary renderings are shown as insets in (a).

Under the attached-sampling formulation of Eq. (34), for any light path $\bar{x} = (x_0, \ldots, x_{N-1}, y)$, the last vertex y is given by attached sampling (and ray intersection), and the remaining ones $x_0, \ldots, x_{N-1}$—which are used to estimate the incident radiance $L_i(x_{cam} \to y^{\perp})$ and its derivative—by the path-space method (§3.2). Thus, the change rates of y and $x_0, \ldots, x_{N-1}$ (with respect to θ) are calculated differently where the former is obtained by differentiating the sampling and ray intersection process:

$$\frac{\mathrm{d}y}{\mathrm{d}\theta} = \frac{\mathrm{d}}{\mathrm{d}\theta} \mathrm{rayIntersect}(x_{cam}(\theta) \to y^{\perp}(\xi; \theta)), \tag{35}$$

and the latter by differentiating the material-form parameterization:

$$\frac{\mathrm{d}x_n}{\mathrm{d}\theta} = \frac{\mathrm{d}}{\mathrm{d}\theta} \mathsf{X}(p_n, \theta), \tag{36}$$

for all $n = 0, 1, \ldots, N-1$ with p_n being a spatial point independent of the scene parameter θ. The discrepancy between Eqs. (35) and (36) becomes problematic when the vertex y and its neighbor x_{N-1} are very close to each other—which can occur at corners or in participating media (as illustrated in Figure 19).

Precisely, when the distance $\|y - x_{N-1}\|$ between y and x_{N-1} approaches zero while the difference $\|(^{\mathrm{d}y}/_{\mathrm{d}\theta}) - (^{\mathrm{d}x_{N-1}}/_{\mathrm{d}\theta})\|$ between their derivatives does not, the term $\frac{\mathrm{d}G(y \leftrightarrow x_{N-1})}{\mathrm{d}\theta} / G(y \leftrightarrow x_{N-1})$ diverges (i.e., goes to infinity). This term is a factor of the *interior* path integral when differentiating $L_i(x_{cam} \to y^{\perp})$ in Eq. (34).

We demonstrate this in Figure 20 using two examples. The "*glass cube*" result contains a rough glass cube lit by a large area light. Attached sampling leads to very high variance around the silhouette

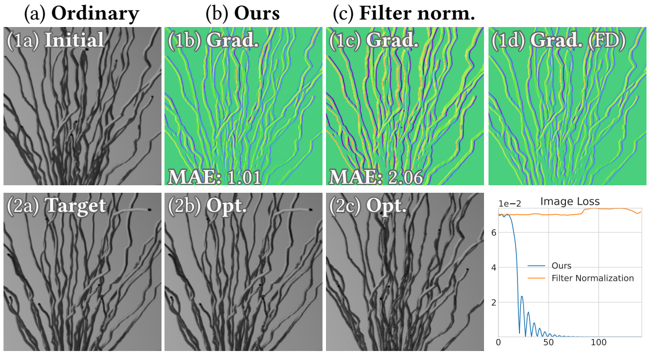


Fig. 21. **Problem of differentiating filter normalization:** Under the material-form parameterization, this approach produces high variance. This example contains a tree-like object. In the top row, we show derivative estimates (with respect to the rotation of the object about the vertial axis). Our result (1b) closely agrees with the finite-difference reference (1d) while the result obtained by differentiating filter normalization (1c) contains high bias. In the bottom row, we compare inverse-rendering results (with the rotation angle being optimized). Using our unbiased gradients, the optimization converges to the target nicely (2b). On the contrary, when using the biased gradients, the optimization diverges (2c).

of the cube. The "*bunny 3*" example contains a translucent bunny with a rough refractive interface. In this case, when the vertex $\boldsymbol{y}$ is on the interface and its neighbor $\boldsymbol{x}_{N-1}$ in the interior, the two vertices can be arbitrarily close, causing most pixels covering the bunny to suffer from high variance. In both examples, our method uses the same material-form parameterization for all vertices and, thus, produces significantly cleaner results.

A.2 Differentiating Filter Normalization

In a concurrent work, Vicini et al. [2022] proposed to approximate the derivative of a pixel value by differentiating the filter normalization process. Precisely, for a pixel with the reconstruction filter h, let $\boldsymbol{y}_1^\perp, \boldsymbol{y}_2^\perp, \ldots$ be image-plane locations sampled from h. Then, the derivative with respect to some scene parameter θ is approximated by differentiating

$$\sum_j \frac{h(\boldsymbol{y}_j^\perp)\, L_\mathrm{i}(\boldsymbol{x}_\mathrm{cam} \to \boldsymbol{y}_j^\perp)}{\sum_k h(\boldsymbol{y}_k^\perp)}. \tag{37}$$

Although this method works adequately under the formulation of warped-area sampling [Bangaru et al. 2020], when applied to the material-form parameterization (presented in §3.2), it produces high bias that can cause inverse-rendering optimizations to diverge. We demonstrate this using a synthetic example in Figure 21. Please see the supplement for an animated version of this result.

B THROUGHPUTS OF ANTITHETIC PATHS

To evaluate our MIS estimator of the *interior* integral in Eq. (18), a key ingredient is computing the *throughputs* of antithetic paths given by $\frac{\mathrm{d}}{\mathrm{d}\theta}\hat{f}(\bar{\boldsymbol{p}}_i^*)/\mathrm{pdf}_i^*(\bar{\boldsymbol{p}}_i^*)$ for $i = 1, 2, 3$.

Let $\bar{\boldsymbol{p}} = (\boldsymbol{p}_0, \ldots, \boldsymbol{p}_{N-1}, \boldsymbol{q})$ be an ordinary path sampled with probability $\mathrm{pdf}(\bar{\boldsymbol{p}})$ and $\bar{\boldsymbol{p}}^* = (\boldsymbol{p}_0, \ldots, \boldsymbol{p}_{N-1}, \boldsymbol{q}^*)$ be an antithesis of $\bar{\boldsymbol{p}}$ (with the subscript i omitted for notational simplicity). Then,

$$\frac{\hat{f}(\bar{\boldsymbol{p}}^*)}{\mathrm{pdf}^*(\bar{\boldsymbol{p}}^*)} = \frac{\hat{f}(\bar{\boldsymbol{p}})}{\mathrm{pdf}(\bar{\boldsymbol{p}})}\, \frac{\hat{f}(\bar{\boldsymbol{p}}^*)}{\hat{f}(\bar{\boldsymbol{p}})}\, \frac{\mathrm{pdf}(\bar{\boldsymbol{p}})}{\mathrm{pdf}^*(\bar{\boldsymbol{p}}^*)}. \tag{38}$$

We note that, as long as we treat all probability densities as constants (i.e., by "detaching" $\mathrm{pdf}(\bar{\boldsymbol{p}})$ and $\mathrm{pdf}^*(\bar{\boldsymbol{p}}^*)$), the derivative $\frac{\mathrm{d}}{\mathrm{d}\theta}\hat{f}(\bar{\boldsymbol{p}}^*)/\mathrm{pdf}^*(\bar{\boldsymbol{p}}^*)$ can be obtained by differentiable evaluation of Eq. (38).

On the right-hand side of Eq. (38), the first component is essentially the throughput of the ordinary path $\bar{\boldsymbol{p}}$ and can be obtained when constructing $\bar{\boldsymbol{p}}$. Since the antithetic path $\bar{\boldsymbol{p}}^*$ only differs from the ordinary one $\bar{\boldsymbol{p}}$ by the last vertex, the second component equals

$$\frac{\hat{f}(\bar{\boldsymbol{p}}^*)}{\hat{f}(\bar{\boldsymbol{p}})} = \frac{g(\boldsymbol{q}^*; \boldsymbol{p}_{N-1}, \boldsymbol{p}_{N-2})}{g(\boldsymbol{q}; \boldsymbol{p}_{N-1}, \boldsymbol{p}_{N-2})}, \tag{39}$$

where $W_\mathrm{e}^\mathrm{pinhole}$ is defined in Eq. (15). Further, for any material points $\boldsymbol{p}, \boldsymbol{p}', \boldsymbol{p}'' \in \mathcal{B}_\mathcal{V}$ with spatial counterparts $\boldsymbol{x} = \mathsf{X}(\boldsymbol{p}, \theta)$, $\boldsymbol{x}' = \mathsf{X}(\boldsymbol{p}', \theta)$, and $\boldsymbol{x}'' = \mathsf{X}(\boldsymbol{p}'', \theta)$:

$$g(\boldsymbol{p}; \boldsymbol{p}', \boldsymbol{p}'') := f_\mathrm{v}(\boldsymbol{x}'' \to \boldsymbol{x}' \to \boldsymbol{x})\, G(\boldsymbol{x}' \leftrightarrow \boldsymbol{x})\, J(\boldsymbol{p})\, W_\mathrm{e}^\mathrm{pinhole}(\boldsymbol{x}). \tag{40}$$

Lastly, as discussed in §5.1, the third component satisfies that

$$\frac{\mathrm{pdf}(\bar{\boldsymbol{p}})}{\mathrm{pdf}^*(\bar{\boldsymbol{p}}^*)} = \left\| \frac{\mathrm{d}\boldsymbol{q}^*}{\mathrm{d}\boldsymbol{q}} \right\|, \tag{41}$$

where the ratio $\|\mathrm{d}\boldsymbol{q}^*/\mathrm{d}\boldsymbol{q}\|$ follows Eqs. (21) and (24).

Transmittance ratio. Computing the ratio of measurement contributions provided by Eq. (39) in turn requires evaluating ratios in transmittances of the form $\tau(\boldsymbol{x}\leftrightarrow\boldsymbol{y}^*)/\tau(\boldsymbol{x}\leftrightarrow\boldsymbol{y})$ for $\boldsymbol{x} \in \{\boldsymbol{x}_{N-1}, \boldsymbol{x}_\mathrm{cam}\}$.

Let $s = \|\boldsymbol{x} - \boldsymbol{y}\|$, $s^* = \|\boldsymbol{x} - \boldsymbol{y}^*\|$, $\lambda(t) := \sigma_\mathrm{t}((1-t)\boldsymbol{x} + t\,\boldsymbol{y})$, and $\lambda^*(t) := \sigma_\mathrm{t}((1-t)\boldsymbol{x} + t\,\boldsymbol{y}^*)$, where σ_t denotes the extinction coefficient. Then, it holds that $\tau(\boldsymbol{x} \leftrightarrow \boldsymbol{y}) = \exp\left(-\int_0^1 s\,\lambda(t)\,\mathrm{d}t\right)$, and $\tau(\boldsymbol{x} \leftrightarrow \boldsymbol{y}^*) = \exp\left(-\int_0^1 s^*\,\lambda^*(t)\,\mathrm{d}t\right)$.

Therefore, we have

$$\frac{\tau(\boldsymbol{x} \leftrightarrow \boldsymbol{y}^*)}{\tau(\boldsymbol{x} \leftrightarrow \boldsymbol{y})} = \exp\left(\int_0^1 \Delta\lambda(t)\,\mathrm{d}t\right), \tag{42}$$

$$\frac{\mathrm{d}}{\mathrm{d}\theta}\frac{\tau(\boldsymbol{x} \leftrightarrow \boldsymbol{y}^*)}{\tau(\boldsymbol{x} \leftrightarrow \boldsymbol{y})} = \left(\int_0^1 \frac{\mathrm{d}}{\mathrm{d}\theta}\Delta\lambda(t)\,\mathrm{d}t\right) \frac{\tau(\boldsymbol{x} \leftrightarrow \boldsymbol{y}^*)}{\tau(\boldsymbol{x} \leftrightarrow \boldsymbol{y})}, \tag{43}$$

where $\Delta\lambda(t) := s\,\lambda(t) - s^*\,\lambda^*(t)$ for all $t \in [0, 1)$.

When the vertex $\boldsymbol{y}^*$ is obtained using equal-transmittance construction, Eq. (42) reduces to one for $\boldsymbol{x} = \boldsymbol{x}_\mathrm{cam}$. Its derivative given by Eq. (43), on the contrary, remains nonzero in general since the equality in transmittance is only enforced at $\theta = \theta_0$.

In practice, for homogeneous media, Eqs. (42) and (43) can be evaluated analytically. For heterogeneous media, the transmittance ratios need to be estimated numerically with, e.g., power-series estimators [Kettunen et al. 2021]:

$$\left\langle \frac{\tau(\boldsymbol{x} \leftrightarrow \boldsymbol{y}^*)}{\tau(\boldsymbol{x} \leftrightarrow \boldsymbol{y})} \right\rangle = \mathrm{e} \prod_{j=1}^{k} \Delta\lambda(t_j), \tag{44}$$

$$\left\langle \frac{\mathrm{d}}{\mathrm{d}\theta}\frac{\tau(\boldsymbol{x} \leftrightarrow \boldsymbol{y}^*)}{\tau(\boldsymbol{x} \leftrightarrow \boldsymbol{y})} \right\rangle = \left(\frac{\mathrm{d}}{\mathrm{d}\theta}\Delta\lambda(t_0)\right) \left\langle \frac{\tau(\boldsymbol{x} \leftrightarrow \boldsymbol{y}^*)}{\tau(\boldsymbol{x} \leftrightarrow \boldsymbol{y})} \right\rangle, \tag{45}$$

where e is Euler's number, $k \geq 0$ an integer drawn from $\mathrm{Pois}(1)$, and $t_j \sim U[0, 1)$ is sampled independently for all $j = 0, 1, \ldots, k$.