

Differentiable Rendering for Specular Materials via Projected Specular Manifolds

RUIZENG LI, University of Science and Technology of China, China and Laoshan Laboratory, China

PEIQI WANG, University of Science and Technology of China, China

BEIBEI WANG*, School of Intelligence Science and Technology, Nanjing University, China

LIGANG LIU*, University of Science and Technology of China, China and Laoshan Laboratory, China

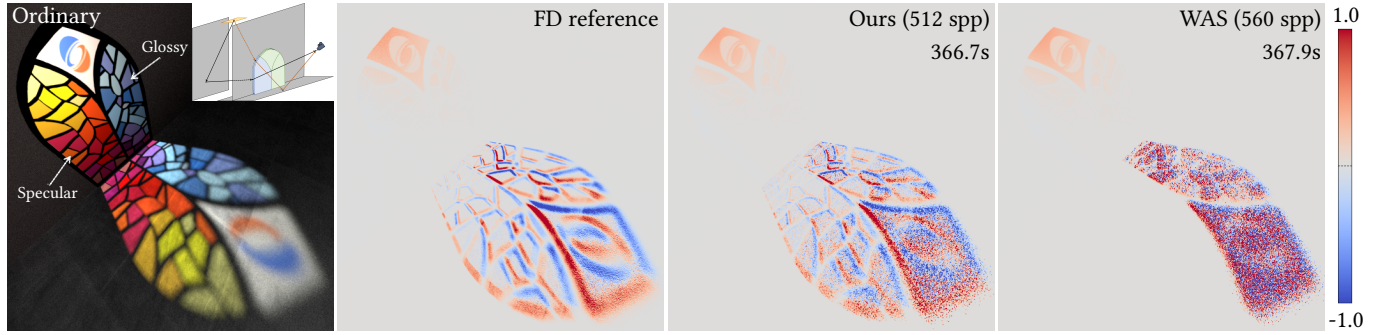


Fig. 1. We introduce the projected specular manifold to enable derivative computation with specular BSDFs containing Dirac delta functions. We then apply the warped-area reparameterization and propose a direct-interpolation-based method to construct the velocity field, which is practical and efficient. Our method can handle the boundary integral caused by discontinuity curves under specular constraints with low additional cost, and can be generalized to glossy BSDFs to reduce derivative noise. In this example, we show a glass window with specular (left) and glossy (right) pieces. The derivatives are computed with respect to the position of the light outside the window.

Physics-based differentiable rendering computes derivatives of radiometric measures with respect to scene parameters. A key challenge lies in handling scene discontinuities from geometries, materials, and other factors. These discontinuities lead to a unique boundary integral defined on discontinuity curves. While occlusion-induced discontinuities have been well studied, those caused by materials, especially specular materials, remain largely unexplored. Specular bidirectional scattering distribution functions (BSDFs) contain Dirac delta distributions arising from the laws of perfect reflection or refraction, which constrain light path geometry and restrict the path integral to the specular manifold. The specular manifold is defined by geometric constraints and lacks explicit location, making the evaluation of discontinuity curves challenging. To address this, our key insight is that we can locate the specular manifold without explicit computation by projecting it onto a subregion of a specular object, which we call the *projected specular manifold (PSM)*. After projection, while exact locations of discontinuity curves remain unknown, we can adopt the warped-area reparameterization to convert the boundary integral into an interior integral over the surface, avoiding the need to precisely locate these curves. The key challenge of this reparameterization is constructing an additional velocity field. For this, our

key observation is that for a specular path, the distances from vertices to manifold boundaries and associated velocities are known. Therefore, we propose a simple but practical *direct-interpolation-based method* to construct the velocity. Furthermore, we extend the warped-area sampling to handle visibility boundaries with specular BSDFs and generalize our method to glossy materials. Our computed derivatives match references with minor additional computational cost, and can successfully drive gradient-based optimization tasks like shape and position estimation. Our implementation is available at <https://github.com/Rhythm25/PSM>.

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

Additional Key Words and Phrases: Differentiable rendering, specular manifold, discontinuity, delta function

ACM Reference Format:

Ruizeng Li, Peiqi Wang, Beibei Wang, and Ligang Liu. 2026. Differentiable Rendering for Specular Materials via Projected Specular Manifolds. *ACM Trans. Graph.* 45, 6, Article 198 (December 2026), 17 pages. <https://doi.org/10.1145/3842514>

1 INTRODUCTION

Physics-based differentiable rendering computes the derivatives of a rendered image with respect to scene parameters. These derivatives can be used in gradient-based optimization for inverse rendering applications. A key challenge in differentiable rendering is handling scene discontinuities, which may arise from geometries, materials, light sources, and other factors. Many existing approaches [Li et al. 2018; Xu et al. 2023; Zhang et al. 2020] focus on discontinuities caused by occlusions, showing that differentiating radiance requires computing an interior integral and a boundary integral. While occlusion-induced discontinuities have been widely studied,

*Corresponding authors.

Authors' Contact Information: Ruizeng Li, University of Science and Technology of China, China and Laoshan Laboratory, China, pb17061297@mail.ustc.edu.cn; Peiqi Wang, University of Science and Technology of China, China, pecky@mail.ustc.edu.cn; Beibei Wang, School of Intelligence Science and Technology, Nanjing University, China, beibei.wang@nju.edu.cn; Ligang Liu, University of Science and Technology of China, China and Laoshan Laboratory, China, lgliu@ustc.edu.cn.



This work is licensed under a Creative Commons Attribution 4.0 International License.
© 2026 Copyright held by the owner/author(s).
ACM 1557-7368/2026/12-ART198
<https://doi.org/10.1145/3842514>

discontinuities caused by materials, such as specular materials, are often neglected. Specular materials commonly appear on surfaces like perfectly smooth water and metals. Their bidirectional scattering distribution function (BSDF) contains a Dirac delta distribution (referred to simply as the delta function), which introduces discontinuities. Our goal is to compute derivatives for specular BSDFs and to handle discontinuities caused by delta functions.

The specular BSDF contains a delta function, which scatters light into a discrete set of directions. As a result, light paths are constrained to satisfy geometric laws of perfect reflection or perfect refraction. Previous work [Jakob and Marschner 2012] has shown that such constraints restrict the path integral to the specular manifold, which is the space of specular paths satisfying these geometric laws. In differentiable rendering, some existing works [Vicini et al. 2021; Xing et al. 2023, 2024; Zeltner et al. 2021] perform differentiation with specular BSDFs by enforcing that light paths satisfy these constraints during differentiation. However, their methods can only evaluate the interior integral and cannot compute the boundary integral, which handles discontinuities arising from scene geometry. As a result, computing the boundary integral for specular BSDFs remains unexplored.

Since the path integral is restricted to the specular manifold, computing the boundary integral requires evaluating discontinuity curves on the specular manifold. Two types of such curves need to be considered: visibility boundaries, and curves arising from object boundaries or normal discontinuities. We refer to the second type as manifold boundaries. Unfortunately, the specular manifold cannot be located explicitly, making it difficult to compute its exact region, which in turn challenges the computation of the boundary integral. To overcome this challenge, our key insight is that the specular manifold is a two-dimensional manifold, and we can define it by projecting it onto a subregion of a specular object, which we refer to as the *projected specular manifold* (PSM). Given this definition, while the exact location of the PSM remains unknown, we can apply the warped-area reparameterization [Bangaru et al. 2020; Wu et al. 2025; Xu et al. 2023] to the projected object. As a result, the boundary integral is transformed into an interior integral on the surface, enabling its computation without the need of precisely locating the discontinuity curves on the PSM.

To adopt the warped-area reparameterization, the key challenge is to construct an additional smooth velocity field defined on the PSM. Prior methods design this velocity field for visibility boundaries using warped-area sampling (WAS) [Bangaru et al. 2020; Xu et al. 2023] or walk-on-spheres (WoS) approach [Wu et al. 2025]. However, these methods are difficult to adapt for manifold boundaries: extending the WAS requires further interpolation, which leads to slower convergence and makes it impractical. Meanwhile, the WoS method is difficult to apply under specular constraints. Therefore, we propose an efficient and practical approach based on *direct interpolation*. Our key observation is that, for a specular path, both the distances from each vertex to discontinuity curves on its object and the corresponding velocities are known. This differs from the case of visibility boundaries, allowing us to avoid sampling-based interpolation of velocities. Consequently, we directly interpolate the velocity field using these distances for manifold boundaries. Furthermore, we generalize the WAS method to handle visibility boundaries

for specular materials. Additionally, we extend our method to glossy materials, where differentiation on the PSM effectively reduces derivative noise.

We demonstrate the effectiveness of our method through various experiments of differentiable rendering and inverse optimization tasks. Our computed derivatives can match the finite difference (FD) reference well and can successfully drive gradient-based optimization tasks like shape and position estimation. In summary, our main contributions are:

- We propose the projected specular manifold to enable computation of the boundary integral for specular BSDFs.
- We propose an efficient direct-interpolation-based method to construct the velocity field for applying the warped-area reparameterization method to the PSM.
- We generalize the WAS method to handle visibility boundaries on the PSM, and further extend our method to glossy BSDFs.

2 RELATED WORK

In this section, we review the most relevant prior works on handling discontinuities and specular objects in differentiable rendering.

Differentiable rendering with discontinuities. In differentiable rendering, computing derivatives typically requires handling two terms: an interior integral and a boundary integral. The interior integral computes the integral of the differentiated integrand for forward rendering, while the boundary integral is defined on the discontinuous boundaries, accounting for scene discontinuities like occlusions. Previous methods for addressing the boundary integral can be mainly divided into two categories: boundary sampling and reparameterization.

The boundary sampling method directly samples on the boundaries for Monte Carlo estimation. Li et al. [2018] propose edge sampling to explicitly sample geometric edges to handle the discontinuities. Zhang et al. [2019] extend edge sampling to support the radiative transfer equation (RTE). Zhang et al. [2020] introduce the path space differentiable rendering (PSDR) and evaluate the boundary integral in boundary path space. PSDR is later extended to handle volumetric light transport [Zhang et al. 2021b], time-gated rendering [Wu et al. 2021], transient rendering [Yi et al. 2021], and discontinuous pixel reconstruction filters [Yu et al. 2022]. Several works have enhanced the sampling efficiency of edge sampling and PSDR approaches [Soroka et al. 2025; Xu et al. 2024; Yan et al. 2022; Zhang et al. 2023]. The boundary sampling method requires additional sampling on discontinuous curves to evaluate the boundary integral, which is time-consuming and often relies on additional guiding data structures to improve sampling efficiency.

The reparameterization method converts the boundary integral to an interior integral defined on the surfaces, eliminating the need for extra Monte Carlo integration on the boundaries. Loubet et al. [2019] apply changes of variables to the boundary integral with rotations, leading to biased results. Bangaru et al. [2020] use the divergence theorem to transform the integral and propose the WAS approach to get unbiased results. Xu et al. [2023] generalize the WAS method to path integrals, which supports unidirectional and bidirectional path-sampling techniques. Their method is biased but consistent. Further, the WoS algorithm is introduced to improve

the warped-area reparameterization [Wu et al. 2025], resulting in unbiased estimations.

In addition, many works focus on designing shader programs to automatically detect discontinuities [Bangaru et al. 2023; Belhe et al. 2025; Michel et al. 2024; Yang et al. 2022]. Zhou et al. [2021] present vectorization to achieve differentiable visibility with automatic differentiation or finite differences. Discontinuities in height fields [Tong et al. 2023] and signed distance fields (SDFs) [Bangaru et al. 2022; Vicini et al. 2022; Wang et al. 2024] have also been explored.

However, prior boundary sampling and reparameterization methods generally assume continuous BSDFs and do not support specular BSDFs. Moreover, the discontinuities considered in previous works are mostly step discontinuities, whose derivatives are delta functions. The differentiation with specular BSDFs involving delta functions has not been thoroughly explored. Our work extends the warped-area reparameterization method to support specular objects, which involve delta functions in the integrals.

Differentiation with specular BSDFs. Zhang et al. [2021a] point out that differentiable rendering with near-specular surfaces often suffers from high variance, as the importance sampling strategy for forward rendering may be not suited for BSDF derivatives. They propose antithetic sampling to improve sampling efficiency. Belhe et al. [2024] introduce an importance sampling method for BRDF derivatives that handles a wide range of complex BRDFs, including highly specular materials. For complex illumination like caustics, path guiding is incorporated by Fan et al. [2024] to enable efficient derivative computation. Multiple importance sampling for differentiable rendering has also been achieved via control variates [Lu et al. 2025]. However, these approaches mainly focus on the interior integral and do not handle perfectly specular surfaces.

Zeltner et al. [2021] propose two sampling methods in differentiable rendering: detached sampling and attached sampling. Attached sampling can handle specular surfaces with delta functions and has been used in efficient backpropagation algorithms [Nimier-David et al. 2020; Vicini et al. 2021]. However, it falls back to detached sampling near geometric boundaries, introducing bias. Xing et al. [2023] introduce the extended path space manifold (EPSM) to support differentiation on specular manifolds, which is later enhanced by differentiable photon mapping (DPM) [Xing et al. 2024]. However, they do not compute the boundary integral for geometric discontinuities and instead rely on optimal transport [Xing et al. 2022] to drive geometric changes during optimization. In contrast, our work presents a method for computing the boundary integral on specular manifolds. Zhang et al. [2025] propose many-worlds derivatives, which can handle specular objects. But their method is mainly designed for optimization and is implemented on occupancy and orientation fields. Hence, we consider it orthogonal to our work.

3 PRELIMINARIES

In this section, we review the preliminaries for differentiating light transport integrals. Taking a simple case of direct illumination, we introduce material-form [Zhang et al. 2020] and warped-area reparameterizations [Bangaru et al. 2020; Wu et al. 2025; Xu et al. 2023] in Sec. 3.1 and Sec. 3.2, respectively. In Sec. 3.3, we revisit the

Table 1. Commonly used symbols in this paper.

Symbol	Definition
$\mathcal{M}_e, \mathcal{M}_s, \mathcal{M}_d$	Emitter, specular (glossy), and non-specular (glossy) objects
$\mathcal{B}_e, \mathcal{B}_s, \mathcal{B}_d$	Reference surfaces
$\mathcal{H}$	Local hemisphere
$\mathbf{x}, \mathbf{x}_i$	Spatial points
$\mathbf{x}^c$	Constrained spatial point
$\mathbf{p}, \mathbf{p}_i$	Material points
$\mathbf{q}, \mathbf{q}_i$	Sampled material points
$\mathbf{h}, \mathbf{h}_0$	Local half-vector, $\mathbf{h}_0 = (0, 0, 1)$
$X(\cdot, \theta)$	One-to-one mapping from reference surfaces to original surfaces
F_{dir}, F	Contributions of a light path
$\hat{F}_{\text{dir}}, \hat{F}$	Contributions of a light path excluding delta functions or NDFs
δ, D	Delta function and NDF
I_{dir}, I	Radiance
θ	Scene parameter
$\mathcal{S}$	Material-form specular manifold
$\mathcal{S}_p$	Projected specular manifold (PSM)
$\mathcal{S}_p^{\text{wa}}$	Unoccluded region of the PSM
$\Delta \mathcal{S}_p$	Discontinuity curves on the PSM
$\Delta \mathcal{S}_p^s, \Delta \mathcal{S}_p^e$	Manifold boundaries caused by different objects
$\partial \mathcal{S}_p^{\text{wa}}$	Visibility boundaries on the PSM
$\mathbf{v}_s, \mathbf{v}_e$	Velocities on the PSM from different objects
$\mathbf{v}_m$	Interpolated velocity for manifold boundaries
$\mathbf{v}$	Full velocity considering visibility boundaries
$\mathbf{v}^\partial$	Velocity on discontinuity boundaries
w	Weight function for interpolating $\mathbf{v}_m$
$\mathbf{w}$	Spatially varying kernel for velocity interpolation at occlusion boundaries

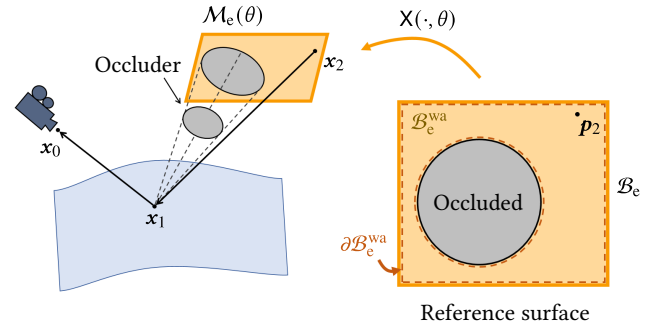


Fig. 2. Left: one-bounce light transport with an occluder and an emitter $\mathcal{M}_e(\theta)$. Right: after applying the material-form reparameterization to $\mathcal{M}_e$ with an one-to-one mapping $X(\cdot, \theta)$, the integration domain is converted to the reference surface $\mathcal{B}_e$. $\mathcal{B}_e$ is static, while the visibility boundaries still depends on θ .

definition of the EPSM. Table 1 presents a list of commonly used symbols and their definitions.



(a) Object boundaries (b) Normal boundaries (c) Visibility boundaries

Fig. 3. Types of geometric boundaries: (a) object, (b) normal, and (c) visibility.

We first revisit the case of direct illumination (i.e., one-bounce light transport). We consider a simple scene with a surface $\mathcal{M}$ and a light source $\mathcal{M}_e$, where the latter depends on the parameter θ . Then given two static points $\mathbf{x}_0$ and $\mathbf{x}_1$, the radiance leaving $\mathbf{x}_1$ toward $\mathbf{x}_0$ can be calculated as (Fig. 2):

$$I_{\text{dir}}(\mathbf{x}_1 \rightarrow \mathbf{x}_0) = \int_{\mathcal{M}_e(\theta)} L_e(\mathbf{x}_1, \mathbf{x}_2) f_s(\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2) G(\mathbf{x}_1, \mathbf{x}_2) dA(\mathbf{x}_2), \quad (1)$$

where L_e is the source emission, f_s is the BSDF, and G is the geometry term. To compute the derivatives of I_{dir} with respect to θ , we introduce material-form and warped-area reparameterizations.

Types of discontinuity curves. Previous works [Zhang et al. 2020, 2019] assume that $L_e(\mathbf{x}_1, \mathbf{x}_2) f_s(\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2)$ is continuous with respect to $\mathbf{x}_2$ in the interior of $\mathcal{M}_e(\theta)$. Consequently, all discontinuities arise from geometry and can be classified into three categories (Fig. 3): (i) object boundaries; (ii) normal boundaries where the surface normals are discontinuous; (iii) visibility boundaries caused by occlusion.

3.1 Material-Form Reparameterization

In Eqn. (1), the integration domain $\mathcal{M}_e$ depends on θ . Consequently, when differentiating the integral, the object boundaries $\partial\mathcal{M}_e(\theta)$ and normal boundaries on $\mathcal{M}_e$ must be taken into account. Zhang et al. [2020] introduce the material-form reparameterization to eliminate them from the boundary integral. They propose finding a fixed **reference surface** $\mathcal{B}_e$ and defining a differentiable one-to-one mapping $X(\cdot, \theta)$ that maps $\mathcal{B}_e$ to $\mathcal{M}_e(\theta)$. Following their definition, we refer to $\mathbf{p} \in \mathcal{B}_e$ as a **material point** and $\mathbf{x} \in \mathcal{M}_e(\theta)$ as a **spatial point** to distinguish points on different surfaces. Then Eqn. (1) can be transformed to an integral over $\mathcal{B}_e$:

$$I_{\text{dir}} = \int_{\mathcal{B}_e} \underbrace{L_e(\mathbf{x}_1, \mathbf{x}_2) f_s(\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2) G(\mathbf{x}_1, \mathbf{x}_2) J(\mathbf{p}_2, \theta)}_{=: F_{\text{dir}}(\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \theta)} dA(\mathbf{p}_2), \quad (2)$$

where $\mathbf{x}_2 = X(\mathbf{p}_2, \theta)$. J is the Jacobian determinant derived from X :

$$J(\mathbf{p}, \theta) := \left\| \frac{dA(\mathbf{x})}{dA(\mathbf{p})} \right\| = \left\| \frac{dA(X(\mathbf{p}, \theta))}{dA(\mathbf{p})} \right\|. \quad (3)$$

For brevity, we write $F_{\text{dir}}(\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \theta)$ as $F_{\text{dir}}(\mathbf{p})$ with $\mathbf{p} = \mathbf{p}_2 \in \mathcal{B}_e$. Then the derivative of the material-form integral can be obtained via Reynolds transport theorem [Cermelli et al. 2005; Reynolds 1969]:

$$\frac{\partial I_{\text{dir}}}{\partial \theta} = \underbrace{\int_{\mathcal{B}_e} \frac{\partial F_{\text{dir}}(\mathbf{p})}{\partial \theta} dA(\mathbf{p})}_{\text{interior}} + \underbrace{\int_{\partial\mathcal{B}_e} \Delta F_{\text{dir}}(\mathbf{p}) V_{\Delta}(\mathbf{p}) dl(\mathbf{p})}_{\text{boundary}}. \quad (4)$$

$\Delta\mathcal{B}_e$ contains the discontinuity curves of F_{dir} within $\mathcal{B}_e$. Since $\mathcal{B}_e$ is independent of θ , its object and normal boundaries are not included in $\Delta\mathcal{B}_e$. Furthermore, under the assumption that L_e and f_s are continuous, discontinuities are only caused by occlusions between $\mathbf{x}_1$ and $\mathbf{x}_2$ (Fig. 2). For the integrand of the boundary integral, by assuming that the curve normal $\mathbf{n}^\partial$ on discontinuity boundaries points toward the occluded side, we have:

$$\begin{aligned} \Delta F_{\text{dir}}(\mathbf{p}) &= F_{\text{dir}}(\mathbf{p}), \\ V_{\Delta}(\mathbf{p}) &= \mathbf{n}^\partial(\mathbf{p}) \cdot \mathbf{v}^\partial(\mathbf{p}), \end{aligned} \quad (5)$$

where $\mathbf{v}^\partial(\mathbf{p})$ is the velocity of the boundary point $\mathbf{p}$ regarding θ .

3.2 Warped-Area Reparameterization

To calculate the boundary integral in Eqn. (4), the warped-area reparameterization [Bangaru et al. 2020; Wu et al. 2025; Xu et al. 2023] is proposed to convert the line integral on discontinuity curves to an area integral on $\mathcal{B}_e$. By using the divergence theorem, the boundary integral can be transformed as

$$\int_{\Delta\mathcal{B}_e} (\Delta F_{\text{dir}} \mathbf{v}^\partial \cdot \mathbf{n}^\partial)(\mathbf{p}) dl(\mathbf{p}) = \int_{\mathcal{B}_e^{\text{wa}}} \nabla \cdot (F_{\text{dir}} \mathbf{v})(\mathbf{p}) dA(\mathbf{p}), \quad (6)$$

where $\mathcal{B}_e^{\text{wa}} \subseteq \mathcal{B}_e$ is the region visible from $\mathbf{x}_1$ (see Fig. 2). $\mathbf{v}$ is a smooth velocity field with $\mathbf{v}(\mathbf{p}) = \mathbf{v}^\partial(\mathbf{p})$ for $\mathbf{p} \in \partial\mathcal{B}_e^{\text{wa}}$. Interpolating $\mathbf{v}$ is the key challenge of the warped-area reparameterization.

Xu et al. [2023] employ a two-step process to construct the velocity field $\mathbf{v}$. They first define a discontinuous velocity field $\mathbf{v}^{\text{dis}}$ on $\mathcal{B}_e$ such that $\mathbf{v}^{\text{dis}}(\mathbf{p}) = \mathbf{v}^\partial(\mathbf{p})$ for $\mathbf{p} \in \partial\mathcal{B}_e^{\text{wa}}$. Then they compute $\mathbf{v}$ by interpolating $\mathbf{v}^{\text{dis}}$ smoothly with a **spatially varying kernel** w :

$$\begin{aligned} \mathbf{v}(\mathbf{p}) &= \frac{\int_{\mathcal{B}_e} w(\mathbf{q}; \mathbf{p}) \mathbf{v}^{\text{dis}}(\mathbf{q}) dA(\mathbf{q})}{\int_{\mathcal{B}_e} w(\mathbf{q}; \mathbf{p}) dA(\mathbf{q})}, \\ w(\mathbf{q}; \mathbf{p}) &= \frac{1}{d(\mathbf{q}; \mathbf{p}) + B(\mathbf{q})}, \end{aligned} \quad (7)$$

where $d(\mathbf{q}; \mathbf{p})$ is the **distance function** that tends to 0 as $\mathbf{q}$ approaches $\mathbf{p}$, and $B(\mathbf{q})$ is the **boundary-test function** that satisfies $B(\mathbf{q}) \rightarrow 0$ as $\mathbf{q} \rightarrow \partial\mathcal{B}_e^{\text{wa}}$. In practice, $\mathbf{v}$ is estimated via Monte Carlo sampling. For each $\mathbf{p} \in \mathcal{B}_e$, they sample a set of points $\mathbf{q}$ around $\mathbf{p}$ and compute $\mathbf{v}$ by a Monte Carlo estimator. Since the expectation $\mathbb{E}[1/f] \neq 1/\mathbb{E}[f]$, the resulting estimate is biased.

Finally, for $\mathbf{p} \notin \mathcal{B}_e^{\text{wa}}$, we have $\nabla \cdot (F_{\text{dir}} \mathbf{v})(\mathbf{p}) = 0$. Then the integral in Eqn. (6) can be extended from $\mathcal{B}_e^{\text{wa}}$ to $\mathcal{B}_e$. The full derivative of I_{dir} can be calculated as:

$$\frac{\partial I_{\text{dir}}}{\partial \theta} = \underbrace{\int_{\mathcal{B}_e} \left(\frac{\partial F_{\text{dir}}(\mathbf{p})}{\partial \theta} + \nabla \cdot (F_{\text{dir}} \mathbf{v})(\mathbf{p}) \right) dA(\mathbf{p})}_{\text{interior}}. \quad (8)$$

3.3 Extended Path Space Manifolds

Xing et al. [2023] define EPSMs to calculate derivatives with difficult light transport problems. A length- n light path can be expressed as $\bar{\mathbf{x}} = \mathbf{x}_0 \mathbf{x}_1 \dots \mathbf{x}_n$, where $\mathbf{x}_0$ is the position of the eye (camera), $\mathbf{x}_i$ is the path vertex on surface for $1 \leq i \leq n-1$, and $\mathbf{x}_n$ is on the light. They find that $n+1$ constraint functions, which matches the number of path vertices, can be incorporated during differentiation. With

the constraint functions $C = [c_1(\bar{x}, \theta), c_2(\bar{x}, \theta), \dots, c_{n+1}(\bar{x}, \theta)]$, the EPSM is defined as

$$\{(\bar{x}, \theta) | \bar{x} \in \mathbb{R}^{2(n+1)}, \theta \in \mathbb{R}^m, C(\bar{x}, \theta) = \mathbf{0}\}. \quad (9)$$

The geometric derivative of the light path can be computed under constraints:

$$\frac{\partial \bar{x}}{\partial \theta} = - \left(\frac{\partial C}{\partial \bar{x}} \right)^{-1} \cdot \frac{\partial C}{\partial \theta}. \quad (10)$$

Constraint functions. Xing et al. [2023] introduce several types of constraint functions.

- (1) *Fixed position constraint.* For the camera point x_0 or point light, this constraint is a 3D constraint that fixes the position of x_0 . Xing et al. [2024] propose that the material-form reparameterization is also a 2D fixed position constraint. Assuming $\mathcal{M}_e(\theta)$ is a triangle mesh, the one-to-one mapping X can be defined by preserving the local barycentric coordinates from p to x on the corresponding triangles containing them.
- (2) *Half-vector constraint.* It fixes the local half-vector of the incoming and outgoing ray directions:

$$\mathbf{h}(x_0, x_1, x_2, \theta) = \mathbf{h}(\overrightarrow{x_1 x_0}, \overrightarrow{x_1 x_2}, \theta) = \mathbf{const}, \quad (11)$$

where $\mathbf{h}$ is the local half-vector at x_1 . Since $\mathbf{h}$ is a normalized vector, this constraint is actually two-dimensional.

EPSMs for specular materials. We apply the half-vector constraint at x_1 and let

$$x_2 = H(x_0, x_1, \mathbf{h}, \theta) \quad (12)$$

where H denotes the intersection point of $\mathcal{M}_e$ and the reflected or refracted ray determined by x_0, x_1 and $\mathbf{h}$. Then performing differentiation under the half-vector constraint corresponds to the geometric constraint arising from the delta function. However, we notice that for a light path with $n + 1$ vertices, only $n + 1$ constraints can be defined. This implies that if we apply the half-vector constraint at x_1 , the material-form reparameterization (i.e., the fixed position constraint) cannot be used. Consequently, the object and normal boundaries of $\mathcal{M}_e$ need to be considered in the boundary integral. But the computation of such boundary integrals has not been addressed in previous works.

4 DIFFERENTIATING INTEGRALS WITH SPECULAR BSDFS

In this section, we propose our method for differentiating integrals involving specular BSDFs with delta functions. In Sec. 4.1, we start from two-bounce light transport as our simple case, which differs from previous approaches that start from direct illumination, and propose the definition of our PSM. To compute the boundary integral on PSM, we propose a direct-interpolation-based method in Sec. 4.2. Then we handle the visibility boundaries on PSM in Sec. 4.3, and generalize our method to path integrals in Sec. 4.4.

One-bounce light transport. We assume that x_1 locates on a specular surface $\mathcal{M}_s$ (Fig. 4) and the BSDF f_s contains a delta function. For BSDFs that contain multiple delta functions, such as dielectric BSDFs, we handle them separately in the integration. Furthermore, we assume that discontinuities in f_s only arise from delta functions,

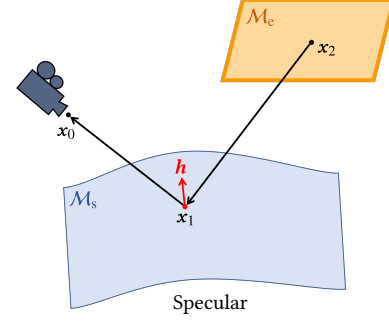


Fig. 4. One-bounce light transport with a specular surface. The light path $x_0 x_1 x_2$ is constrained by the local half-vector $\mathbf{h}$.

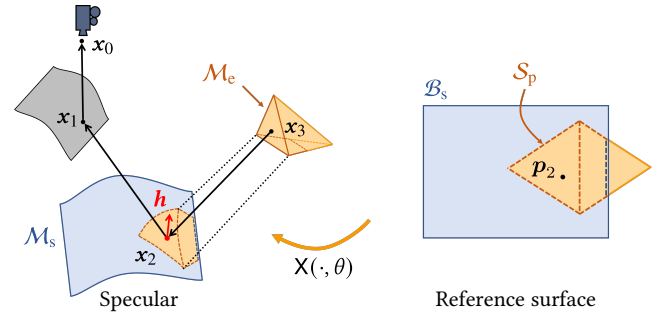


Fig. 5. Left: two-bounce light transport as our simple case with the specular surface $\mathcal{M}_s$. The integrand is discontinuous on $\mathcal{M}_s$ while being nonzero only on a subregion. Right: the PSM $\mathcal{S}_p$ is defined on the reference surface $\mathcal{B}_s$ by projecting $\mathcal{M}_e$ onto $\mathcal{B}_s$ under geometric constraints.

and that the light sources are continuous. Then the material-form integral in Eqn. (2) can be written as:

$$\begin{aligned} I_{\text{dir}} &= \int_{\mathcal{B}_e} F_{\text{dir}}(x_0, x_1, x_2, \theta) dA(p_2) \\ &= \int_{\mathcal{B}_e} \delta(\mathbf{h}(x_0, x_1, x_2, \theta) - \mathbf{h}_0) \widehat{F}_{\text{dir}}(x_0, x_1, x_2, \theta) dA(p_2) \\ &= \widehat{F}_{\text{dir}}(x_0, x_1, x_2^c, \theta), \end{aligned} \quad (13)$$

where $x_2^c = H(x_0, x_1, \mathbf{h}_0, \theta)$ with the function H define in Eqn. (12), p_2^c is the corresponding material point, and $\mathbf{h}_0 = (0, 0, 1)^T$. Due to the presence of the delta function, the derivative can be computed directly as the single integral degenerates.

However, since the delta function introduces *high-order discontinuities*¹, $I_{\text{dir}}(x_1 \rightarrow x_0)$ can be discontinuous for $x_1 \in \mathcal{M}_s$ (Fig. 5). Then for the integral of I_{dir} over $\mathcal{M}_s$, I_{dir} introduces a unique type of discontinuity curve arising from the delta function. Therefore, we begin with two-bounce light transport as our simple case to handle the discontinuities caused by specular BSDFs.

4.1 Differentiating Two-bounce Light Transport

As shown in Fig. 5, we consider the light path $x_0 x_1 x_2 x_3$ with x_0 and x_1 fixed. x_2 locates on a specular object $\mathcal{M}_s$ and x_3 is on the

¹The Dirac delta function δ in $\mathbb{R}^n$ belongs to the Sobolev space $H^{-s}(\mathbb{R}^n)$ for $s > n/2$ [Adams and Fournier 2003].

emitter $\mathcal{M}_e$. The two-layer material-form integral for two-bounce light transport can be written as:

$$I(\mathbf{x}_1 \rightarrow \mathbf{x}_0) = \int_{\mathcal{B}_s} \int_{\mathcal{B}_e} F(\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \theta) dA(\mathbf{p}_3) dA(\mathbf{p}_2), \quad (14)$$

where F represents the contribution of the light path. Due to the presence of the delta function, we get

$$\begin{aligned} I &= \int_{\mathcal{B}_s} \int_{\mathcal{B}_e} \delta(\mathbf{h} - \mathbf{h}_0) \widehat{F}(\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \theta) dA(\mathbf{p}_3) dA(\mathbf{p}_2) \\ &= \int_S \widehat{F}_s(\mathbf{x}_0, \mathbf{x}_1, \mathbf{p}, \theta) dA(\mathbf{p}). \end{aligned} \quad (15)$$

The original double integral is reduced to a single integral on the **material-form specular manifold** $\mathcal{S}$:

$$\begin{aligned} \mathcal{S} = \mathcal{S}(\mathbf{h}_0, \theta; \mathbf{x}_1) &:= \{\mathbf{p}_2 \in \mathcal{B}_s, \mathbf{p}_3 \in \mathcal{B}_e \mid \text{Corresponding spatial} \\ &\quad \text{points } (\mathbf{x}_2, \mathbf{x}_3) \text{ satisfy } \mathbf{h}(\overrightarrow{\mathbf{x}_2\mathbf{x}_1}, \overrightarrow{\mathbf{x}_2\mathbf{x}_3}, \theta) = \mathbf{h}_0\}, \end{aligned} \quad (16)$$

and the function $\widehat{F}_s$ accounts for the integrand with the Jacobian determinant included. Then we can get the derivative of I :

$$\frac{\partial I}{\partial \theta} = \underbrace{\int_S \frac{\partial \widehat{F}_s}{\partial \theta} dA(\mathbf{p})}_{\text{interior}} + \underbrace{\int_{\Delta S} \Delta \widehat{F}_s V_{\Delta} dl(\mathbf{p})}_{\text{boundary}}, \quad (17)$$

where ΔS contains the discontinuity curves of $\widehat{F}_s$.

Challenge of locating ΔS . In Eqn. (17), the derivative is not directly computable, since $\mathcal{S}$ is defined as an abstract manifold and not represented explicitly. Computing the boundary integral requires sampling on ΔS and then finding the corresponding valid light paths on the specular manifold. Although some previous manifold exploration methods [Jakob and Marschner 2012] can find valid specular paths on the specular manifold, they cannot guarantee that such paths lie exactly on the discontinuity curves. Therefore, they cannot be directly applied to locate ΔS .

Key insight. To compute the integrals in Eqn. (17), we first need to make the integration domains computable. We observe that $\mathcal{S}$ is a 2D manifold, which is formed by two 2D surfaces $\mathcal{B}_s$ and $\mathcal{B}_e$ under a two-dimensional constraint. Therefore, we can define $\mathcal{S}$ as a subregion of $\mathcal{B}_s$ or $\mathcal{B}_e$. We propose our **projected specular manifold** (PSM), which is defined by projecting $\mathcal{M}_e$ on $\mathcal{B}_s$ under half-vector constraint:

$$\begin{aligned} \mathcal{S}_p = \mathcal{S}_p(\mathbf{h}_0, \theta; \mathbf{x}_1) &:= \{\mathbf{p}_2 \in \mathcal{B}_s \mid \exists \mathbf{x}_3 \in \mathcal{M}_e, \\ &\quad \text{s.t. } \mathbf{h}(\overrightarrow{\mathbf{x}_2\mathbf{x}_1}, \overrightarrow{\mathbf{x}_2\mathbf{x}_3}, \theta) = \mathbf{h}_0 \text{ with } \mathbf{x}_2 = \mathbf{X}(\mathbf{p}_2, \theta)\}. \end{aligned} \quad (18)$$

Our formulation. With the PSM, we have

$$I = \int_S \widehat{F}_s(\mathbf{x}_0, \mathbf{x}_1, \mathbf{p}, \theta) dA(\mathbf{p}) = \int_{\mathcal{B}_s} \widehat{F}(\mathbf{x}_0, \mathbf{x}_1, \mathbf{p}_2, \theta) dA(\mathbf{p}_2). \quad (19)$$

Here for brevity, we use the same symbol $\widehat{F}$ defined as:

$$\widehat{F}(\mathbf{x}_0, \mathbf{x}_1, \mathbf{p}_2, \theta) = \begin{cases} \widehat{F}(\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3^c, \theta), & \mathbf{p}_2 \in \mathcal{S}_p \subseteq \mathcal{B}_s, \\ 0, & \mathbf{p}_2 \in \mathcal{B}_s \setminus \mathcal{S}_p. \end{cases} \quad (20)$$

Note that the Jacobian factor from Eqn. (15) to Eqn. (19) reduces to unity. Then we can transform the integration domain in Eqn. (17) to $\mathcal{B}_s$, obtaining the derivative:

$$\frac{\partial I}{\partial \theta} = \underbrace{\int_{\mathcal{B}_s} \frac{\partial \widehat{F}}{\partial \theta} dA(\mathbf{p}_2)}_{\text{interior}} + \underbrace{\int_{\Delta S_p} \Delta \widehat{F} V_{\Delta} dl(\mathbf{p}_2)}_{\text{boundary}}. \quad (21)$$

The interior integral can be computed using automatic differentiation (AD). For the boundary integral, we need to consider the discontinuity curves of $\widehat{F}$ on $\mathcal{B}_s$. In addition to visibility boundaries, object boundaries and normal boundaries of $\mathcal{B}_s$ and $\mathcal{B}_e$ also need to be considered. We refer to these two types of boundaries as the **manifold boundaries**, and use $\Delta \mathcal{S}_p^s$ and $\Delta \mathcal{S}_p^e$ to represent the manifold boundaries on $\mathcal{S}_p$ formed by $\mathcal{B}_s$ and $\mathcal{M}_e$, respectively. Further, we represent the visibility boundaries as $\partial \mathcal{S}_p^{\text{wa}}$, where

$$\mathcal{S}_p^{\text{wa}} := \{\mathbf{p}_2 \in \mathcal{S}_p \mid \overrightarrow{\mathbf{x}_2\mathbf{x}_1} \text{ and } \overrightarrow{\mathbf{x}_2\mathbf{x}_3} \text{ are both visible}\}. \quad (22)$$

Then $\Delta \mathcal{S}_p = \Delta \mathcal{S}_p^s \cup \Delta \mathcal{S}_p^e \cup \partial \mathcal{S}_p^{\text{wa}}$. The boundaries $\Delta \mathcal{S}_p^e$ and $\partial \mathcal{S}_p^{\text{wa}}$ cannot be eliminated by the material-form reparameterization, as they are determined by multiple objects under geometric constraints. In the following sections, we introduce our solution for computing the boundary integral.

4.2 Boundary Integral on Manifold Boundaries

We first assume that $\Delta \mathcal{S}_p$ only contains manifold boundaries, and leave visibility boundaries to Sec. 4.3. Then $\Delta \mathcal{S}_p = \Delta \mathcal{S}_p^s \cup \Delta \mathcal{S}_p^e$. Furthermore, as shown in Fig. 5, we observe that $\mathcal{S}_p$ is partitioned by discontinuity boundaries into several subregions, and $\widehat{F}$ is continuous in the interior of each subregion. The integral over $\mathcal{S}_p$ can be expressed as the sum of integrals over these subregions. Therefore, without loss of generality, we assume that $\mathcal{S}_p$ is one of these subregions. Then we have $\widehat{F}(\mathbf{p}) = 0$ for $\mathbf{p} \in \mathcal{B}_s \setminus \mathcal{S}_p$ and $\Delta \widehat{F} = \widehat{F}$.

4.2.1 Warped-area reparameterization. Existing methods for computing the boundary integral mainly include boundary sampling [Li et al. 2018; Soroka et al. 2025; Yan et al. 2022; Zhang et al. 2020, 2023] and reparameterization [Bangaru et al. 2020; Loubet et al. 2019; Wu et al. 2025; Xu et al. 2023]. The former requires additional sampling on discontinuity boundaries. But sampling valid boundary paths under geometric constraints is challenging in our case, and some approaches rely on guiding data structures. Therefore, we choose to adopt the warped-area reparameterization method to convert the boundary integral into the interior integral. We start by applying the divergence theorem:

$$\int_{\Delta \mathcal{S}_p} \Delta \widehat{F} V_{\Delta} dl(\mathbf{p}_2) = \int_{\mathcal{S}_p} \nabla \cdot (\widehat{F} \mathbf{v}_m) dA(\mathbf{p}_2). \quad (23)$$

The variable $\mathbf{v}_m$ denotes a smooth velocity field on $\mathcal{S}_p$ such that $\mathbf{v}_m(\mathbf{p}) = \mathbf{v}^{\partial}(\mathbf{p})$ for $\mathbf{p} \in \Delta \mathcal{S}_p$.

The key for computing Eqn. (23) is to construct the velocity field $\mathbf{v}_m$. As shown in Fig. 6, $\Delta \mathcal{S}_p$ consists of $\Delta \mathcal{S}_p^s$ and $\Delta \mathcal{S}_p^e$. $\mathbf{v}_m$ needs to

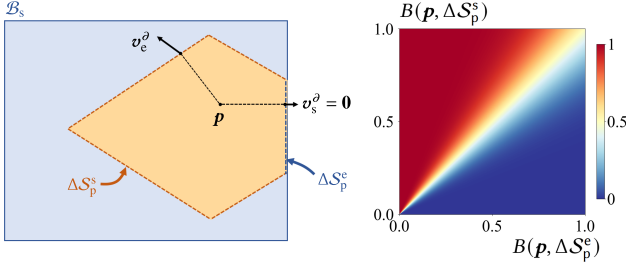


Fig. 6. Left: interpolating the velocity field on S_p requires considering different velocities on boundaries arising from different objects. Right: heatmap of our weight function w varying with $B(\mathbf{p}, \Delta S_p^s)$ and $B(\mathbf{p}, \Delta S_p^e)$ at $k = 6$.

be smooth² and satisfy:

$$\begin{aligned} \mathbf{v}_m(\mathbf{p}) &= \mathbf{v}_s^o(\mathbf{p}) = \mathbf{0} \text{ on } \Delta S_p^s, \\ \mathbf{v}_m(\mathbf{p}) &= \mathbf{v}_e^o(\mathbf{p}) \text{ on } \Delta S_p^e, \end{aligned} \quad (24)$$

where $\mathbf{v}_e^o$ is the velocity on $\mathcal{B}_s$ computed under geometric constraints like Eqn. (10). More details are provided in the supplemental material. Then we need to design $\mathbf{v}_m$ to satisfy the above conditions.

Sampling-based solution. A straightforward approach to computing the velocity field $\mathbf{v}_m$ is to extend the previous WAS method [Bangaru et al. 2020; Xu et al. 2023]. We adapt this framework to our PSM to obtain a *sampling-based solution*. Specifically, we sample auxiliary rays around a valid light path, trace them under the specular constraints, and compute the corresponding velocity field via Monte Carlo estimation. However, we observe that the velocity computation further involves interpolation using velocities and weights from multiple objects. This added complexity leads to slower convergence, which is validated in Fig. 15. The detailed formulation is provided in the supplemental material.

Moreover, other methods such as WoS [Wu et al. 2025] are difficult to apply in our case, because we cannot easily know the location of ΔS_p , and thus cannot obtain the distance from $\mathbf{p} \in \mathcal{B}_s$ to ΔS_p . Hence, our aim is to find a more efficient and practical method.

Our solution. We observe that the WAS method relies on Monte Carlo estimation because information about the occluder cannot be directly accessed by a valid light path. Specifically, for a light path that does not hit the occluder, the distance from the path to the occluder and the corresponding velocity are unknown. Consequently, it needs to sample around the light path to find the occluder. However, in our case, the velocity fields from different objects on the PSM are already known, which motivates us to design a more direct way to interpolate $\mathbf{v}_m$. Therefore, we introduce a *direct-interpolation-based solution*.

4.2.2 Direct-interpolation-based solution for constructing velocity. We propose a more practical and efficient way to obtain $\mathbf{v}_m(\mathbf{p})$ by *direct interpolation* instead of relying on Monte Carlo estimation:

$$\mathbf{v}_m(\mathbf{p}) := w(\mathbf{p})\mathbf{v}_e(\mathbf{p}) + (1 - w(\mathbf{p}))\mathbf{v}_s(\mathbf{p}), \quad (25)$$

² $\mathbf{v}_m$ needs to be differentiable almost everywhere except on a set of measure zero.

where w is the **weight function** satisfying:

$$\begin{aligned} w(\mathbf{p}) &= 0 \text{ on } \Delta S_p^s, \\ w(\mathbf{p}) &= 1 \text{ on } \Delta S_p^e. \end{aligned} \quad (26)$$

$\mathbf{v}_s = \mathbf{0}$ and $\mathbf{v}_e$ are velocity fields on $\mathcal{B}_s$, where $\mathbf{v}_e$ results from mapping the velocity field on $\mathcal{M}_e$ to $\mathcal{B}_s$ under geometric constraints, and $\mathbf{v}_e = \mathbf{v}_e^o$ on ΔS_p^e .

Weight function. An intuitive observation is that w needs to be designed based on the distances from $\mathbf{p}$ to ΔS_p^s and ΔS_p^e . Although such exact distances are unknown, we find that a function can be designed to measure how far $\mathbf{p}$ is from ΔS_p^s and ΔS_p^e . We choose to rely on the **boundary-test function** used in Eqn. (7). Then we define the weight function w as:

$$w(\mathbf{p}) := \frac{B^k(\mathbf{p}, \Delta S_p^s)}{B^k(\mathbf{p}, \Delta S_p^s) + B^k(\mathbf{p}, \Delta S_p^e)}, \quad (27)$$

where k is a user-defined parameter. w satisfies the conditions in Eqn. (26). We set $k = 6$ which works well in practice. More types of weight functions and comparisons are provided in the supplemental material. B is the boundary-test function defined below.

Boundary-test function. For a material point $\mathbf{p}_2$ corresponding to a light path segment $\mathbf{x}_1\mathbf{x}_2(\mathbf{p}_2)\mathbf{x}_3(\mathbf{p}_2)$ and its material-form path $\mathbf{p}_1\mathbf{p}_2\mathbf{p}_3$, we define the boundary-test functions as:

$$\begin{aligned} B(\mathbf{p}_2, \Delta S_p^e) &:= \min(B_o(\mathbf{p}_2, \mathcal{B}_e), B_n(\mathbf{p}_2, \mathcal{B}_e), B_t(\mathbf{p}_2, \mathbf{p}_3, \mathcal{B}_e)), \\ B(\mathbf{p}_2, \Delta S_p^s) &:= \min(B_o(\mathbf{p}_2, \mathcal{B}_s), B_n(\mathbf{p}_2, \mathcal{B}_s), B_t(\mathbf{p}_1, \mathbf{p}_2, \mathcal{B}_s)), \end{aligned} \quad (28)$$

where B_o , B_n and B_t are boundary-test functions for different types of boundaries:

- (1) $B_o(\mathbf{p}, \mathcal{B})$ computes the shortest distance from $\mathbf{p}$ to the open boundary of $\mathcal{B}$ containing $\mathbf{p}$. If $\mathcal{B}$ is closed, B_o equals $+\infty$. For open triangle meshes, we precompute the shortest distance from each vertex to the open boundary and interpolate them for points inside a triangle during computation for efficiency.
- (2) $B_n(\mathbf{p}, \mathcal{B})$ calculates the shortest distance from $\mathbf{p}$ to the edges of $\mathcal{B}$ where the normal jumps discontinuously. If $\mathcal{B}$ has smooth normal, B_n equals $+\infty$.
- (3) $B_t(\mathbf{p}_i, \mathbf{p}_{i+1}, \mathcal{B})$ measures how close $\mathbf{p}_{i+1}$ is to the object boundary based on whether $\overrightarrow{\mathbf{p}_i\mathbf{p}_{i+1}}$ is tangent to the normal at $\mathbf{p}_{i+1}$:

$$B_t(\mathbf{p}_i, \mathbf{p}_{i+1}, \mathcal{B}) = |\overrightarrow{\mathbf{p}_i\mathbf{p}_{i+1}} \cdot \mathbf{n}(\mathbf{p}_{i+1})|. \quad (29)$$

Then it can be proved that $\mathbf{p} \in \Delta S_p$ is equivalent to $B(\mathbf{p}, \Delta S_p^e) \cdot B(\mathbf{p}, \Delta S_p^s) = 0$ (proof in the supplementary).

4.2.3 Full differentiation. Now we have the interpolated velocity field $\mathbf{v}_m$. With $\mathbf{v}_s = \mathbf{0}$, the integrand in Eqn. (23) becomes

$$\nabla \cdot (\widehat{F}\mathbf{v}_m) = \nabla \widehat{F} \cdot (w\mathbf{v}_e) + \widehat{F}\nabla w \cdot \mathbf{v}_e + \widehat{F}w\nabla \cdot \mathbf{v}_e, \quad (30)$$

where ∇w and $\nabla \cdot \mathbf{v}_e$ can be computed by AD. More details can be found in the supplemental material. Since $\widehat{F}(\mathbf{p}) = 0$ for $\mathbf{p} \in \mathcal{B}_s \setminus S_p$, we can extend the integration domain to $\mathcal{B}_s$ and get:

$$\int_{\Delta S_p} \Delta \widehat{F} V_{\Delta} dl(\mathbf{p}_2) = \int_{\mathcal{B}_s} \nabla \cdot (\widehat{F}\mathbf{v}_m) dA(\mathbf{p}_2), \quad (31)$$

which can be merged into the interior integral in Eqn. (17), obtaining our complete differentiation result. Algorithm 1 provides more

ALGORITHM 1: Compute the interior and manifold boundary integrals for a light path under the two-bounce case.

Input: A light path $\mathbf{x}_0\mathbf{x}_1\mathbf{x}_2\mathbf{x}_3$ with $\mathbf{x}_2$ on $\mathcal{M}_s$ and $\mathbf{x}_3$ on $\mathcal{M}_e$. $\mathbf{p}_2$ is the material point of $\mathbf{x}_2$ on $\mathcal{B}_s$.

Output: The derivative dI consisting of the interior and manifold boundary integrals.

```

1 begin
2   Compute the contribution  $\widehat{F}$  of the light path;
3   Compute  $d\widehat{F}/d\theta$  and  $\nabla\widehat{F}(\mathbf{p}_2)$  using AD;
4    $dI \leftarrow d\widehat{F}/d\theta$ ;
5   Compute  $B(\mathbf{p}_2, \Delta\mathcal{S}_p^s)$  and  $B(\mathbf{p}_2, \Delta\mathcal{S}_p^e)$ ;
6    $\mathbf{w} \leftarrow B^k(\mathbf{p}_2, \Delta\mathcal{S}_p^s) / (B^k(\mathbf{p}_2, \Delta\mathcal{S}_p^s) + B^k(\mathbf{p}_2, \Delta\mathcal{S}_p^e))$ ;
7   Compute  $\nabla\mathbf{w}(\mathbf{p}_2)$ ,  $\mathbf{v}_e(\mathbf{p}_2)$  and  $\nabla \cdot \mathbf{v}_e(\mathbf{p}_2)$  using AD;
8    $\mathbf{v}_m \leftarrow \mathbf{w}\mathbf{v}_e$ ;
9    $\nabla \cdot \mathbf{v}_m \leftarrow \nabla\mathbf{w} \cdot \mathbf{v}_e + \mathbf{w}\nabla \cdot \mathbf{v}_e$ ;
10   $dI \leftarrow dI + \nabla\widehat{F} \cdot \mathbf{v}_m + \widehat{F}\nabla \cdot \mathbf{v}_m$ ;
11  return  $dI$ ;
12 end

```

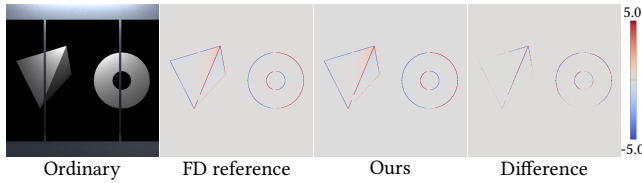


Fig. 7. We validate our method by comparing its computed derivatives under high (1024) samples per pixel (spp) with FD reference. We only show the derivatives on specular mirrors. Our method matches the FD reference well. Note that the observed difference comes from both the numerical error of the FD reference and the bias of our method.

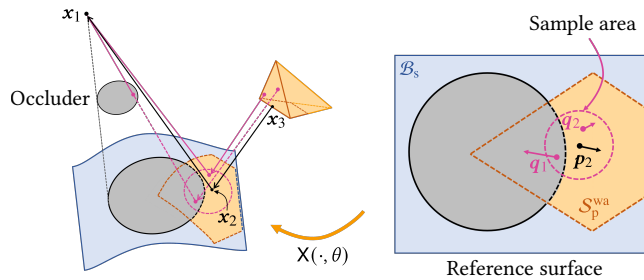


Fig. 8. Illustration of occlusion on the PSM. To compute the velocity of visibility boundaries on $\mathcal{S}_p$, we sample around $\mathbf{p}_2$ and trace rays under geometric constraints to locate the occluder.

details about the full differentiation. Our method matches the FD reference across different types of boundaries as shown in Fig. 7 and Sec. 6.

4.3 Boundary Integral on Visibility Boundaries

Now we handle the visibility boundaries $\partial\mathcal{S}_p^{\text{wa}}$ on the PSM. For this, we leverage the WAS method. By applying the divergence theorem,

we have:

$$\int_{\Delta\mathcal{S}_p} \Delta\widehat{F}V_{\Delta}dI(\mathbf{p}_2) = \int_{\mathcal{S}_p^{\text{wa}}} \nabla \cdot (\widehat{F}\mathbf{v})dA(\mathbf{p}_2). \quad (32)$$

The key difference from previous WAS work is that we need to compute the velocity of visibility boundaries on the PSM. Since there is already a velocity field $\mathbf{v}_m$ for manifold boundaries, we need to further interpolate the visibility boundary velocity and $\mathbf{v}_m$. We follow the two steps proposed by Xu et al. [2023] to design $\mathbf{v}$: defining a discontinuous velocity field $\mathbf{v}^{\text{dis}}$, and smoothing $\mathbf{v}^{\text{dis}}$.

Constructing $\mathbf{v}^{\text{dis}}$. We construct a discontinuous velocity field $\mathbf{v}^{\text{dis}}$ on $\mathcal{S}_p$ (Fig. 8):

$$\begin{aligned} \mathbf{v}^{\text{dis}}(\mathbf{p}) &= \mathbf{0} \text{ for } \mathbf{p} \in \mathcal{S}_p^{\text{wa}}, \\ \mathbf{v}^{\text{dis}}(\mathbf{p}) &= \mathbf{v}_{\text{vis}}(\mathbf{p}) - \mathbf{v}_m(\mathbf{p}) \text{ for } \mathbf{p} \in \mathcal{S}_p \setminus \mathcal{S}_p^{\text{wa}}, \end{aligned} \quad (33)$$

where $\mathbf{v}_{\text{vis}}$ is the velocity of the occluder on $\mathcal{S}_p$. We find that in this case, occlusion may occur both from $\mathbf{x}_1$ to $\mathcal{M}_s$ and from $\mathcal{M}_s$ to $\mathcal{M}_e$, which we represent as $\mathcal{S}_p^{\mathbf{x}_1\mathbf{x}_2}$ and $\mathcal{S}_p^{\mathbf{x}_2\mathbf{x}_3}$ respectively. Then $\mathcal{S}_p \setminus \mathcal{S}_p^{\text{wa}} = \mathcal{S}_p^{\mathbf{x}_1\mathbf{x}_2} \cup \mathcal{S}_p^{\mathbf{x}_2\mathbf{x}_3}$, and $\mathbf{v}_{\text{vis}}$ needs to be constructed with velocity fields $\mathbf{v}_{\text{vis}}^{\mathbf{x}_1\mathbf{x}_2}$ and $\mathbf{v}_{\text{vis}}^{\mathbf{x}_2\mathbf{x}_3}$. We choose to define $\mathbf{v}_{\text{vis}}$ using the velocity from the first occluder encountered by the occluded ray. The detailed computation is provided in the supplemental material.

Smoothing $\mathbf{v}^{\text{dis}}$. We smooth $\mathbf{v}^{\text{dis}}$ and get the full velocity as:

$$\mathbf{v}(\mathbf{p}) := \mathbf{v}_m(\mathbf{p}) + \frac{\int_{\mathcal{S}_p} \mathbf{w}(\mathbf{q}; \mathbf{p}) \mathbf{v}^{\text{dis}}(\mathbf{q}) dA(\mathbf{q})}{\int_{\mathcal{S}_p} \mathbf{w}(\mathbf{q}; \mathbf{p}) dA(\mathbf{q})}, \quad (34)$$

where $\mathbf{w}(\mathbf{q}; \mathbf{p})$ is a spatially varying kernel which behaves like a delta function for $\mathbf{p} \in \partial\mathcal{S}_p^{\text{wa}}$. We follow the design of $\mathbf{w}$ in Xu et al.'s [2023] work:

$$\mathbf{w}(\mathbf{q}; \mathbf{p}) := \frac{1}{d(\mathbf{q}; \mathbf{p}) + B_{\text{vis}}(\mathbf{q})}, \quad (35)$$

where $d(\mathbf{q}; \mathbf{p})$ is a non-Euclidean distance function defined as:

$$d(\mathbf{q}; \mathbf{p}) := \left(\frac{r_0}{\sigma_0} \left(1 - \exp \left(-\frac{\|\mathbf{q} - \mathbf{p}\|^2}{r_0\sigma_0} \right) \right) \right)^a, \quad (36)$$

where $r_0 = \|\mathbf{x}_1 - \mathbf{x}_2\|$ (Fig. 8), a and σ_0 are user-defined parameters. We follow the choices of Xu et al. [2023] to set $a = 3$ and $\sigma_0 = 0.006$. B_{vis} is the boundary-test function for visibility boundaries, which is also computed from the first occluded path segment.

Sampling on PSM. For a valid path corresponding to some point $\mathbf{p} \in \mathcal{S}_p$, we need to sample around $\mathbf{p}$ to compute Eqn. (34). We adopt the same sampling strategy as Xu et al.'s [2023] work. Please refer to their work for more details. Then as shown in Fig. 8, for a sampled point $\mathbf{q} \in \mathcal{B}_s$ around $\mathbf{p}$, we need to trace a reflected or refracted ray beginning at $\mathbf{q}$ to check whether $\mathbf{q}$ is located on $\mathcal{S}_p$ and $\mathcal{S}_p^{\text{wa}}$. We provide an algorithm about calculating $\mathbf{v}(\mathbf{p})$ under the two-bounce case in Algorithm 2.

4.4 Differential Path Integral with PSMs

Specular chain. Now we consider the light path $\mathbf{x}_{-1}\mathbf{x}_0 \dots \mathbf{x}_{n+1}$ which involves more specular objects as shown in Fig. 9. $\mathbf{x}_{-1}$ and

ALGORITHM 2: Construct the velocity $\mathbf{v}$ and $\nabla \cdot \mathbf{v}$.

Input: A light path segment $\mathbf{x}_0 \mathbf{x}_1 \mathbf{x}_2 \mathbf{x}_3$ with $\mathbf{x}_2$ on $\mathcal{M}_s$. $\mathbf{x}_3$ is on a non-specular object $\mathcal{M}_d$. $\mathbf{p}_2$ is the material point of $\mathbf{x}_2$ on the reference surface $\mathcal{B}_s$.

Output: $\mathbf{v}(\mathbf{p}_2)$ and $\nabla \cdot \mathbf{v}(\mathbf{p}_2)$

```

1 begin
2    $W \leftarrow 0; \nabla W \leftarrow \mathbf{0};$ 
3    $\tilde{\mathbf{v}} \leftarrow \mathbf{0}; \nabla \cdot \tilde{\mathbf{v}} \leftarrow 0;$ 
4   Compute  $\mathbf{v}_m(\mathbf{p}_2), \nabla \cdot \mathbf{v}_m(\mathbf{p}_2);$ 
5   for  $i = 1$  to  $N_q$  do
6     Sample  $\mathbf{q}_i$  around  $\mathbf{p}_2$  with pdf and get  $\mathbf{y}_i = \mathcal{X}(\mathbf{q}_i, \theta);$ 
7     Get the next ray located at  $\mathbf{y}_i$  with specular constraint;
8     if the next ray cannot hit  $\mathcal{M}_d$  then
9       continue;
10    Get the intersection  $\mathbf{y}_{i,d}$  on  $\mathcal{M}_d$ ;
11    Compute  $\mathbf{v}_m(\mathbf{q}_i);$ 
12    if  $\mathbf{x}_0 \mathbf{x}_1 \mathbf{y}_{i,d}$  is occluded then
13      Compute  $\mathbf{v}_{vis}(\mathbf{q}_i), w$  and  $\nabla w;$ 
14       $W \leftarrow W + w/\text{pdf};$ 
15       $\nabla W \leftarrow \nabla W + \nabla w/\text{pdf};$ 
16       $\mathbf{v}^{\text{dis}} \leftarrow \mathbf{v}_{vis}(\mathbf{q}_i) - \mathbf{v}_m(\mathbf{q}_i);$ 
17       $\tilde{\mathbf{v}} \leftarrow \tilde{\mathbf{v}} + w\mathbf{v}^{\text{dis}}/\text{pdf};$ 
18       $\nabla \cdot \tilde{\mathbf{v}} \leftarrow \nabla \cdot \tilde{\mathbf{v}} + \nabla w \cdot \mathbf{v}^{\text{dis}}/\text{pdf};$ 
19  end
20   $\mathbf{v} \leftarrow \mathbf{v}_m(\mathbf{p}_2) + \tilde{\mathbf{v}}/W;$ 
21   $\nabla \cdot \mathbf{v} \leftarrow \nabla \cdot \mathbf{v}_m(\mathbf{p}_2) + \nabla \cdot \tilde{\mathbf{v}}/W - \nabla W \cdot \tilde{\mathbf{v}}/W^2;$ 
22  return  $\mathbf{v}, \nabla \cdot \mathbf{v};$ 
23 end

```

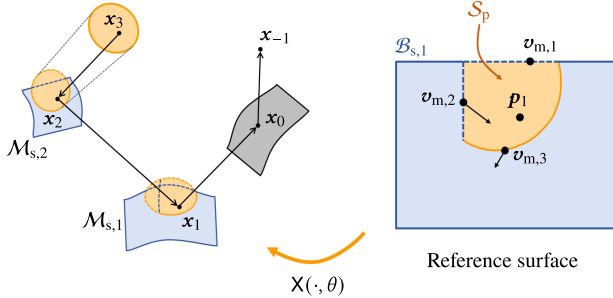


Fig. 9. For a specular chain, we define the PSM on the reference surface of the first specular surface. The boundary velocity is influenced by all objects in the chain.

$\mathbf{x}_0$ are fixed. $\mathbf{x}_1, \dots, \mathbf{x}_n$ lie on specular surfaces $\mathcal{M}_{s,i}$, $i = 1, 2, \dots, n$, respectively, and $\mathbf{x}_{n+1}$ lies on a non-specular object $\mathcal{M}_d$. We have:

$$\begin{aligned}
 I(\mathbf{x}_0 \rightarrow \mathbf{x}_{-1}) &= \int_{\mathcal{B}_{s,1}} \dots \int_{\mathcal{B}_{s,n}} \int_{\mathcal{B}_d} \left(\prod_{i=2}^{n+1} \delta(\mathbf{h}_i - \mathbf{h}_0) \right) \widehat{F} dA(\mathbf{p}_{n+1}) \dots dA(\mathbf{p}_1), \quad (37) \\
 &= \int_S \widehat{F}_s dA(\mathbf{p}_1),
 \end{aligned}$$

where $\widehat{F}$ and $\widehat{F}_s$ represent the contributions of the light path on their respective integral domains, excluding the delta functions.

The variable $\mathcal{B}_d$ denotes the reference surface of $\mathcal{M}_d$, and $\mathbf{h}_i$ is the local half-vector at $\mathbf{x}_i$. We find that the n delta functions reduce the multiple integral to a single integral over the two dimensional specular manifold. Then we choose to define the PSM on the first specular reference surface $\mathcal{B}_{s,1}$:

$$I(\mathbf{x}_0 \rightarrow \mathbf{x}_{-1}) = \int_{\mathcal{S}_p} \widehat{F} dA(\mathbf{p}_1) = \int_{\mathcal{B}_{s,1}} \widehat{F} dA(\mathbf{p}_1). \quad (38)$$

By applying our method, we can compute the boundary integral on manifold boundaries by interpolating the velocity field:

$$\begin{aligned}
 \mathbf{v}_m(\mathbf{p}) &:= \sum_{i=1}^{n+1} w_i(\mathbf{p}) \mathbf{v}_{m,i}(\mathbf{p}), \\
 w_i(\mathbf{p}) &= \begin{cases} 1 & \text{on } \Delta \mathcal{S}_p^i, \\ 0 & \text{on } \Delta \mathcal{S}_p^j \text{ for } j \neq i, \end{cases} \quad (39)
 \end{aligned}$$

where $\Delta \mathcal{S}_p^i$ is the manifold boundaries on $\mathcal{B}_{s,1}$ resulting from the boundaries of $\mathcal{M}_{s,i}$ (see Fig. 9). $\mathbf{v}_{m,i}$ is the velocity field on $\mathcal{B}_{s,1}$ mapped from $\mathcal{M}_{s,i}$. $\mathcal{M}_{s,n+1}$ denotes $\mathcal{M}_d$. We can extend our weight function as:

$$w_i(\mathbf{p}) := \prod_{\substack{1 \leq j \leq n+1 \\ j \neq i}} \frac{B^k(\mathbf{p}, \Delta \mathcal{S}_p^j)}{B^k(\mathbf{p}, \Delta \mathcal{S}_p^i) + B^k(\mathbf{p}, \Delta \mathcal{S}_p^j)}. \quad (40)$$

With visibility boundaries, the full velocity can be extended in the form of Eqn. (34).

PSM in path integral. For general path integrals, non-specular objects can separate the integration into individual specular path segments. We can define the corresponding PSM for each specular path segment. Our method applies to both unidirectional and bidirectional estimators. We design unidirectional and bidirectional estimators in a similar way as Xu et al.'s [2023] work, with the difference that we need to perform sampling on the PSM and additionally compute velocities for manifold boundaries. We refer to their work for more details about the estimators.

5 DIFFERENTIATING INTEGRALS WITH GLOSSY BSDFs

Our method is now capable of handling specular surfaces. In this section, we generalize our method to differentiate the integral with glossy surfaces.

5.1 Our Formulation

We consider the glossy BSDF with the form:

$$f_s(\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2) = D(\mathbf{h}) \widehat{f}_s(\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2), \quad (41)$$

where $\mathbf{h} = \mathbf{h}(\overrightarrow{\mathbf{x}_1 \mathbf{x}_0}, \overrightarrow{\mathbf{x}_1 \mathbf{x}_2})$ is the local half-vector at $\mathbf{x}_1$, $D(\mathbf{h})$ is a microfacet normal distribution function (NDF) like the Beckmann [Beckmann and Spizzichino 1963] and GGX [Trowbridge and Reitz 1975; Walter et al. 2007] models.

5.1.1 Two-bounce case. As shown in Fig. 10, we suppose that $\mathbf{x}_2$ lies on a glossy surface $\mathcal{M}_s$. With the BSDF f_s , the two-bounce material-form integral in Eqn. (14) can be written as:

$$I(\mathbf{x}_1 \rightarrow \mathbf{x}_0) = \int_{\mathcal{B}_s} \int_{\mathcal{B}_e} D(\mathbf{h}) \widehat{F} dA(\mathbf{p}_3) dA(\mathbf{p}_2), \quad (42)$$

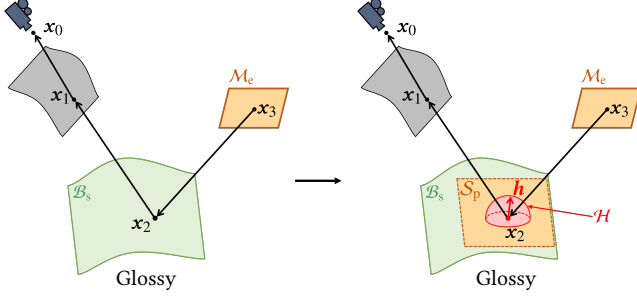


Fig. 10. Left: the two-bounce light transport with a glossy surface. Right: after applying the half-vector reparameterization and change of integration order, the integration domain of the two-bounce material-form integral can be converted to the local hemisphere $\mathcal{H}$ and PSM $\mathcal{S}_p$.

where we continue to use $\hat{F}$ to represent the contribution function after removing D . We propose that this integral can be converted to the PSM formulation by two operations: *half-vector reparameterization* and *change of integration order*.

Half-vector reparameterization. We first reparameterize the integral by performing a change of variables from the reference surface $\mathcal{B}_e$ to the static local hemisphere $\mathcal{H}$ of $\mathbf{h}$:

$$I = \int_{\mathcal{B}_s} \int_{\mathcal{H}} D(\mathbf{h}) \hat{F}_{\mathbf{h}} dA(\mathbf{h}) dA(\mathbf{p}_2), \quad (43)$$

where $dA(\mathbf{h})$ is the solid angle measure, and $\hat{F}_{\mathbf{h}}$ represents the contribution function after multiplying $\hat{F}$ by the corresponding Jacobian determinant. In this way, $\mathbf{x}_3$ is determined by the mapping $\mathbf{x}_3 = H(\mathbf{x}_1, \mathbf{x}_2, \mathbf{h}, \theta)$ as in Eqn. (12).

Change of integration order. In the next step, we reorder the integration as:

$$I = \int_{\mathcal{H}} \underbrace{\int_{\mathcal{B}_s} D(\mathbf{h}) \hat{F}_{\mathbf{h}} dA(\mathbf{p}_2)}_{=: F_{\text{int}}(\mathbf{x}_0, \mathbf{x}_1, \mathbf{h}, \theta)} dA(\mathbf{h}). \quad (44)$$

Note that this operation does not affect the result of the integral and the Monte Carlo estimate due to the Tonelli's theorem as the integrand is non-negative.

Our formulation. We observe that F_{int} is continuous on $\mathcal{H}$ (the details are provided in the supplemental material) and has a similar form as Eqn. (15). Then for each half-vector $\mathbf{h}$, we can compute the derivatives on the PSM $\mathcal{S}_p(\mathbf{h}) = \mathcal{S}_p(\mathbf{h}, \theta; \mathbf{x}_1)$ (Fig. 10):

$$\begin{aligned} \frac{\partial I}{\partial \theta} &= \int_{\mathcal{H}} \frac{\partial F_{\text{int}}}{\partial \theta} dA(\mathbf{h}), \\ \frac{\partial F_{\text{int}}}{\partial \theta} &= \underbrace{\int_{\mathcal{B}_s} D(\mathbf{h}) \frac{\partial \hat{F}_{\mathbf{h}}}{\partial \theta} dA(\mathbf{p}_2)}_{\text{interior}} + \underbrace{\int_{\Delta \mathcal{S}_p(\mathbf{h})} \Delta(D \hat{F}_{\mathbf{h}}) V_{\Delta} dl(\mathbf{p}_2)}_{\text{boundary}}, \end{aligned} \quad (45)$$

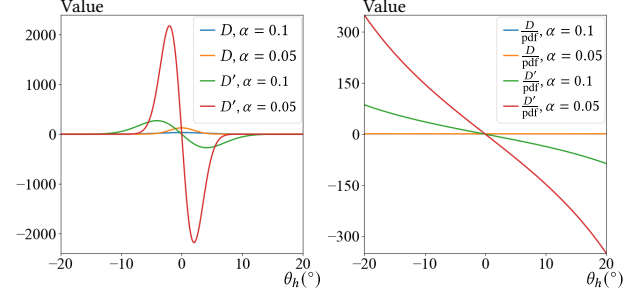


Fig. 11. Beckmann NDF D and its derivative D' with respect to the angle θ_h between the half-vector and the normal, with different roughness α . For small α , D' exhibits a greatly different shape compared to D , making D' not proportional to the importance sampling pdf of D , and therefore introducing large noise.

where $\Delta \mathcal{S}_p(\mathbf{h})$ is the PSM boundaries on $\mathcal{B}_s$ with half-vector $\mathbf{h}$. Then the boundary integral can be computed by our method with the full velocity $\mathbf{v}$ in Eqn. (34):

$$\int_{\Delta \mathcal{S}_p(\mathbf{h})} \Delta(D \hat{F}_{\mathbf{h}}) V_{\Delta} dl(\mathbf{p}_2) = \int_{\mathcal{B}_s} \nabla \cdot (D \hat{F}_{\mathbf{h}} \mathbf{v})(\mathbf{p}_2) dA(\mathbf{p}_2). \quad (46)$$

5.1.2 Long light path. Our method can be extended to support more glossy surfaces as in Eqn. (37):

$$I = \int_{\mathcal{H}_1} \dots \int_{\mathcal{H}_n} \int_{\mathcal{B}_s} D(\mathbf{h}_1) \dots D(\mathbf{h}_n) \hat{F}_{\mathbf{h}} dA(\mathbf{p}_2) dA(\mathbf{h}_n) \dots dA(\mathbf{h}_1). \quad (47)$$

In this way, we move the integral defined on the PSM to the inner layer and account for all boundaries on the PSM.

5.2 Noise Reduction by Differentiation on the PSM

Xing et al. [2024] propose that computing derivatives under half-vector constraints can reduce noise for glossy BSDFs. More specifically, if we compute the derivative of Eqn. (42), the derivative of D is required by the chain rule:

$$\frac{\partial D(\mathbf{h})}{\partial \theta} = \frac{\partial D}{\partial \mathbf{h}} \frac{\partial \mathbf{h}}{\partial x_2} \frac{\partial x_2}{\partial \theta} = \frac{\partial D}{\partial \mathbf{h}} \frac{\partial \mathbf{h}}{\partial x_2} \frac{\partial X(\mathbf{p}, \theta)}{\partial \theta}. \quad (48)$$

For nearly specular material, $\partial D / \partial \mathbf{h}$ differs significantly from D (Fig. 11), which leads to large variance of the Monte Carlo estimate. Since D is only related to $\mathbf{h}$, fixing $\mathbf{h}$ as the half-vector constraint makes $\partial D / \partial \mathbf{h}$ vanish, which can avoid the large noise caused by $\partial D / \partial \theta$. However, applying the half-vector constraint requires computing the boundary integral on the specular manifold, which is not solved by Xing et al. [2024]. For the boundary computed via our PSM, we have

$$\nabla \cdot (D \hat{F}_{\mathbf{h}} \mathbf{v})(\mathbf{p}_2) = D \nabla \cdot (\hat{F}_{\mathbf{h}} \mathbf{v})(\mathbf{p}_2), \quad (49)$$

where $\partial D / \partial \mathbf{h}$ also does not need to be computed. However, our velocity $\mathbf{v}$ can vary significantly because of weight functions, which introduces extra noise. Fortunately, this noise is concentrated in a small region because of the shapes of the weight functions (Fig. 8). Therefore, when differentiation is moved from reference surfaces to the PSM, the noise arising from $\partial D / \partial \mathbf{h}$ is transferred from the interior integral to the boundary integral and becomes more localized.

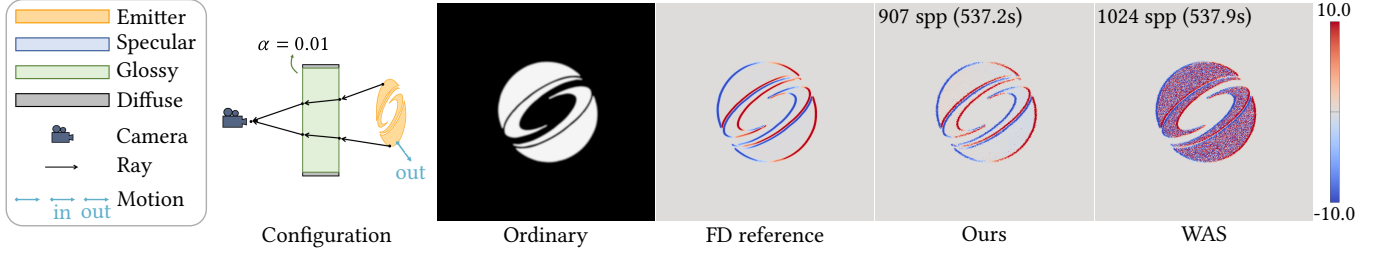


Fig. 12. Equal-time comparisons of our method against WAS under glossy BSDFs with roughness $\alpha = 0.01$. The WAS method struggles with high noise even at high spp due to the derivatives of the NDF, while our method converge well by performing differentiation on the PSM. For glossy BSDFs, we use the Beckmann model [Beckmann and Spizzichino 1963] throughout this paper. We provide a legend for the configurations used in this paper.

This results in effective noise reduction over the entire domain, as demonstrated in Fig. 12 and more results in Sec. 6.

6 RESULTS

We have implemented our method based on PS-DR-Enzyme by Xu et al. [2023], which is CPU-based and uses the Enzyme automatic differentiation framework [Moses and Churavy 2020; Moses et al. 2021, 2022]. All timings are measured on a single NVIDIA 3090 GPU and an AMD Ryzen 9 5950X CPU with 16 cores and 32 threads. The resolutions of images are set as 512×512 for derivative visualization results and 256×256 for optimization experiments considering efficiency, unless otherwise stated. For sampling visibility boundaries, we use 8 additional samples as WAS. We provide configurations to illustrate our scene with the legend in Fig. 12.

We compare our method against existing methods including WAS [Xu et al. 2023], projective sampling (Projective) [Zhang et al. 2023], DPM [Xing et al. 2024], and attached sampling (Attached) [Zeltner et al. 2021], using their released codes and default settings. For Attached, the code for attached path replay backpropagation [Vicini et al. 2021] has not been released. We therefore rely on the implementation from Zeltner et al. [2021] which only supports attached sampling under direct illumination.

6.1 Validation on Derivative Computation

Derivative computation. To validate the correctness and effectiveness of our approach, we first compare our computed derivatives with respect to scene geometries against FD reference and existing methods in Fig. 13. Due to differences between the codebases, we use equal sample rather than equal time. The performances of different method are provided in Table 2. Our method closely matches the FD reference for both specular and glossy BSDFs by effectively computing the boundary integral. In contrast, prior methods cannot correctly compute geometry derivatives. The WAS can only compute geometry derivatives for glossy BSDFs and fails under specular BSDFs. Projective returns some derivatives for short specular paths (e.g., first row of Fig. 13) because projective sampling is still permitted in the presence of delta functions, as stated in MITSUBA 3 [Jakob et al. 2022], combined with its derivative backpropagation. However, Projective becomes ineffective for long specular paths. DPM can compute derivatives for specular BSDFs but fails to handle geometric discontinuities, returning incorrect derivatives. Attached only supports direct illumination (the first row in Fig. 13) and also fails

to correctly handle geometric changes. In Table. 3, we provide a summary on abilities of different methods.

Equal-time comparison. We provide equal-time error comparisons between our method and existing approaches in Fig. 14. Although Projective, DPM, and Attached are GPU-based, while our method and WAS are CPU-based, we still perform an equal-time comparison directly. We report the MAEs computed against the FD reference. Although our method is computationally more expensive, it achieves the smallest error, whereas the existing methods, despite their lower computational cost, produce results with large errors. This is because our method correctly handles geometric discontinuities under specular materials, which is not achieved by other approaches.

Validation of our direct-interpolation-based method. We compare our direct-interpolation-based method against the sampling-based solution in Fig. 15. The sampling-based solution is biased but consistent, which requires a large number of auxiliary samples to converge to a relatively correct result, which is time-consuming and not practical. Our method is more efficient due to the direct interpolation. The slow convergence of sampling-based solution is caused by the interpolation form under the case of specular manifold. More details can be found in the supplemental material.

Comparison with antithetic sampling. Zhang et al. [2021a] propose the antithetic sampling to reduce noise caused by derivatives of glossy BSDFs. In Fig. 16, we compare our method with antithetic sampling under equal time. We only show the interior integral computed by the unidirectional path-space method (PS) [Zhang et al. 2020] with material-form reparameterization and antithetic sampling applied, which corresponds to our interior integral plus manifold boundary integral on PSM with a unidirectional estimator. Our method effectively reduces noise across the entire region, resulting in less noise compared to antithetic sampling. The reason is that although material-form reparameterization excludes object boundaries from the boundary integral, this operation introduces a fixed position constraint (see Sec. 3.3). Consequently, the path vertex varies with the parameter θ during differentiation, which requires the derivatives of the NDF. Our method performs differentiation on the PSM, eliminating the need to compute NDF derivatives in the interior integral. Meanwhile, the manifold boundary integral is concentrated near the manifold boundaries. Consequently, the overall noise is significantly reduced.

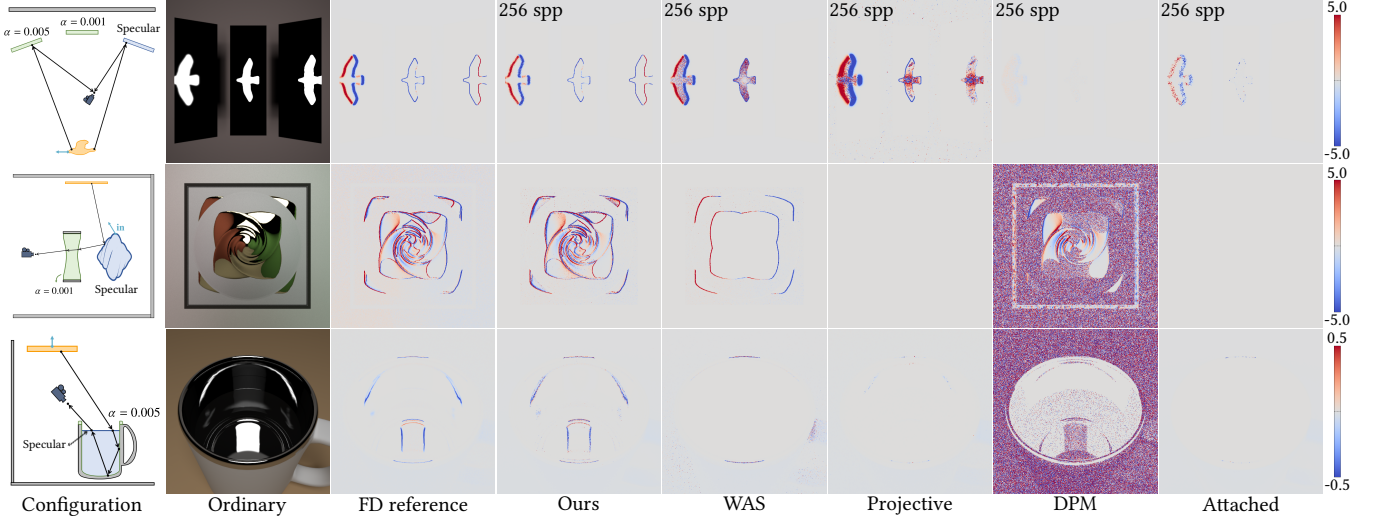


Fig. 13. Comparisons of derivatives between our method and prior works including WAS, Projective, DPM, and Attached under equal sample (256 spp). Our method can match the FD reference closely for both specular and glossy BSDFs, while other methods fail to compute derivatives correctly.

Table 2. We report the performances of different methods for both derivative computation and optimization. For optimization, we report the time and corresponding parameters of the first optimization stage, as we employ a multi-stage optimization strategy. The parameters include spp, maximum path length, and the vertex number of the optimized object (# param). Note that our method and WAS run on the CPU, while the others run on the GPU.

Scene	spp	Max path length	# param	Ours	WAS	Projective	DPM	Attached
Bird (Fig. 13)	256	2	-	221.4s	209.2s	12.4s	13.3s	0.53s
Flower (Fig. 13)	256	5	-	679.3s	541.6s	13.4s	134.1s	0.56s
Cup (Fig. 13)	256	6	-	541.1s	420.8s	26.1s	89.9s	0.54s
Squirrel (Fig. 14)	64	3	10000	9.5s	8.9s	0.3s	2.5s	0.14s
Bunny (Fig. 14)	64	4	20000	13.2s	11.0s	0.6s	4.3s	0.16s
Teddy (Fig. 14)	16	4	10000	2.6s	2.2s	0.9s	2.5s	0.05s
Chess (Fig. 18)	32	3	10000	16.8s	14.9s	0.7s	3.2s	0.11s

Table 3. Comparison of different methods on their abilities. Our method successfully handles geometric discontinuities in the presence of specular BSDFs, while other methods fail to correctly handle both specular BSDFs and geometric discontinuities.

Methods	Ours	WAS	Projective	DPM	Attached
Support specular BSDFs	✓	×	✓ (with large error)	✓	✓
Handle geometric discontinuities	✓	✓	✓	×	×

Besides, we also compare against the Projective method in Fig. 16. It exhibits less noise, but loses derivatives in small highlight regions caused by large geometric changes. The reason is that Projective uses a different reparameterization than WAS, and projects ordinary interior samples onto nearby boundaries to guide the sampling of the boundary integral. However, this projection operation tends to fail in these small regions. More discussions about the Projective method can be found in Sec. 6.4.

Combination with bidirectional estimator. Our method can be applied to the bidirectional path tracer. Fig. 17 provides a comparison of derivatives between our method and WAS using bidirectional path tracing (BDPT), with glossy BSDFs and equal time. WAS suffers

from the high noise caused by the derivative of the NDF. In contrast, our method produces significantly less noise by differentiating on the PSM, which avoids the computation of NDF derivative and reducing noise over the entire domain. Compared to WAS, our method introduces low additional computational cost.

Validation on backward-mode gradients. We validate the backward-mode gradients by comparing the gradients propagated back to the mesh vertices in Fig. 18. We use a specular mirror to reflect a light source, similar to the setup in Fig. 15. To compute the FD reference, we use a low-poly mesh for the light, set the image loss L to the sum of all pixel colors, and compute the FD derivatives of L with respect to each component of every vertex

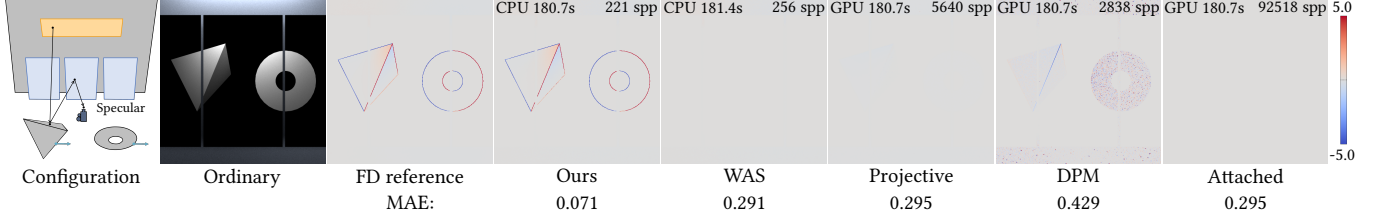


Fig. 14. Equal-time error comparisons between our method and existing approaches. Our method achieves the lowest MAE. In contrast, the other methods produce results with substantially larger errors even though they run faster. Note that Projective, DPM, and Attached are GPU-based, and the Attached method is restricted to direct illumination, which contributes to their shorter runtime.

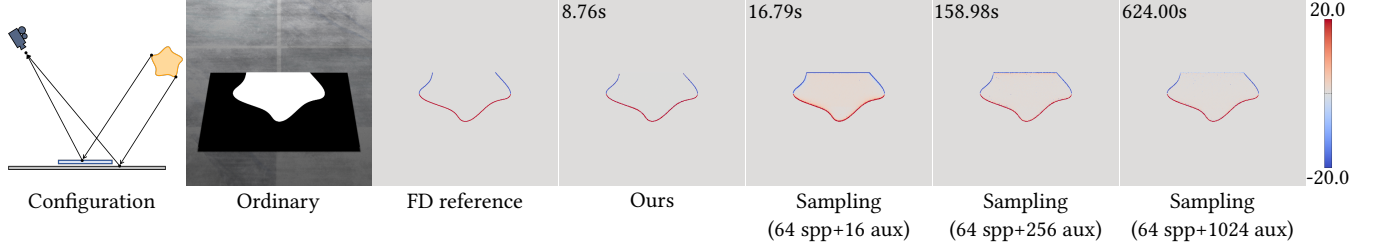


Fig. 15. Comparisons between our direct-interpolation-based solution and the sampling-based solution on the manifold boundary integral under 64 spp. The sampling-based solution requires auxiliary samples for the Monte Carlo estimation and converges slowly because of the interpolation form.

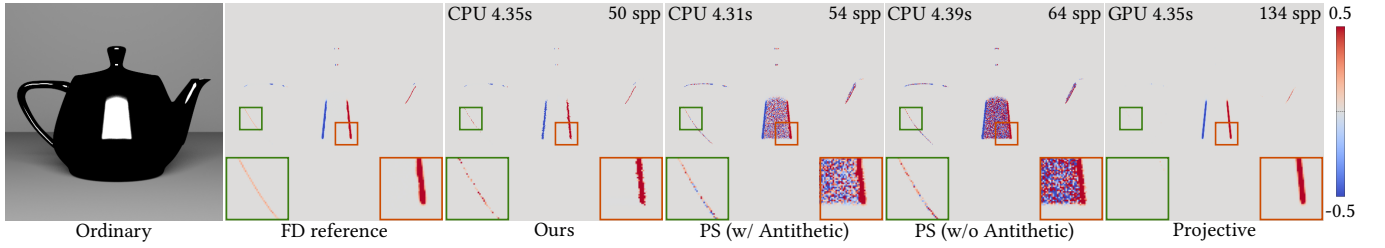


Fig. 16. We compare our method against antithetic sampling under equal time with a glossy BSDF of roughness $\alpha = 0.01$. We only show the interior integral of the PS method [Zhang et al. 2020], which is computed under material-form reparameterization with and without antithetic sampling. This corresponds to the interior integral plus the manifold boundary integral in our method. Our method outperforms antithetic sampling and has less noise. In addition, Projective is also included as a baseline for comparison. It produces less noise but struggles in small highlight regions.

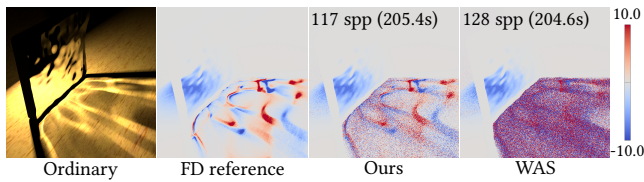


Fig. 17. Comparisons of our method and WAS with BDPT under equal time. We use a glossy BSDF with roughness $\alpha = 0.01$. Our method produces significantly less noise comparing with WAS.

on the light. For our method, we set the gradient dL/dI to identity and back-propagate it to the light source mesh vertices. Then we compare the derivatives at the vertices of the light. The results from our method show minimal error from the FD reference. This validates that although our method applies reparameterization to convert the boundary integral into an interior integral, it is a

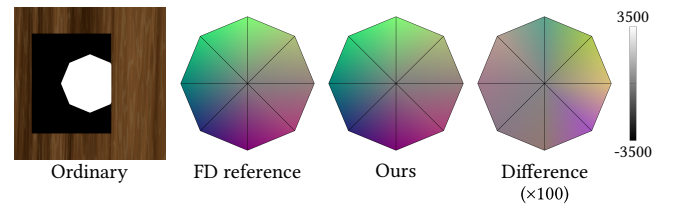


Fig. 18. Validations on backward-mode gradients. We compare the gradients propagated back to the low-poly octagonal light mesh vertices. For the x , y , and z components of the gradients at vertices, we apply a unified color mapping to convert them into RGB values and visualize them on the mesh. We use the barycentric interpolation across each triangle using the vertex colors. Our method achieves minimal errors compared to the FD reference.

mathematical reformulation that does not change the gradient. The gradient is expressed through a different representation, but is not

altered or redistributed. It therefore does not affect the use of the gradient in inverse rendering, which is further validated in Sec. 6.2.

6.2 Validation on Gradient-based Optimization

Gradient-based geometry optimization. In Fig. 19, we present a comparison of inverse rendering via gradient-based optimization. We optimize object shape to match target images rendered from 40 known poses. Mesh vertices are optimized using the method of Nicolet et al. [2021] with identical settings across all methods, including spp, learning rate, mesh vertex count, and the mean absolute error (MAE) loss function. Note that although the original DPM relies on the optimal transport method [Xing et al. 2022] to drive geometry optimization, we only use its computed derivatives within Nicolet et al.’s framework for a fair comparison. Our method successfully drives geometry optimization to match the target images with both specular and glossy BSDFs, using our computed derivatives. For other methods, in the first two rows with specular materials, only Projective can drive optimization, but its results fail to match the target. This is because WAS, DPM, and Attached cannot compute geometry derivatives under specular BSDFs, while the derivatives produced by Projective are incorrect. In the third row with glossy materials, WAS, Projective and DPM do not converge well due to the high noise introduced by glossy BSDF derivatives. The optimization process and the corresponding mesh shape are provided in the supplemental video. The optimization performances of different methods are provided in Table 2.

Optimization for occluder. In Fig. 20, we validate our method for visibility boundaries on PSM by performing shape optimization for an occluder. We compare with previous methods under equal sample and same settings as above. Our method can optimize the occluder’s shape under a specular surface, while other methods fail. Specifically, WAS does not support specular BSDFs. Projective returns highly inaccurate derivatives, leading to divergence. DPM and Attached fails to handle geometric discontinuities. Our approach generalizes WAS by sampling on the PSM, enabling it to handle visibility boundaries under specular BSDFs.

6.3 Additional Applications

We show additional results including derivative computations for point lights and the index of refraction (IOR). Prior methods typically do not compute derivatives with respect to point light position or the IOR of specular objects due to the presence of delta functions, and set them as non-differentiable parameters. We therefore present only the results of our method without comparison.

Delta function from point light. Our method computes derivatives in the presence of delta functions introduced by specular BSDFs, which produces a geometric half-vector constraint. Our method can be extended to handle delta functions from other sources, such as point lights. When both point lights and specular objects are present, we define our PSM by projecting the constrained manifold onto the first non-specular surface before the specular object. In Fig. 21, we use BDPT to render point light reflected by a specular object onto a diffuse surface. Based on the reflected pattern, we optimize the position of the point light. In this case, our PSM is

defined on the diffuse surface. Our method remains effective even under these extreme conditions.

Derivatives of the IOR. The IOR of a specular BSDF affects the location of the specular manifold. Consequently, changes in the IOR introduce the boundary integral that must be accounted for. In Fig. 22, we present an example of derivative computation and optimization with respect to the IOR. Specifically, we optimize the IOR of a convex lens to match a target image, demonstrating the effectiveness of our method.

6.4 Discussion and Limitations

Biasedness. While our direct-interpolation-based method is unbiased in theory, it introduces bias in practice, as $\mathbf{v}_m$ is ill-defined at the intersection of ΔS_p^s and ΔS_p^e . For points near the intersection, both $B(\mathbf{p}, \Delta S_p^s)$ and $B(\mathbf{p}, \Delta S_p^e)$ are small, which makes weight functions numerically unstable. More details about the error analysis are provided in the supplemental material. Developing unbiased methods based on boundary sampling or WoS is left for future work.

Choice of the object defining the PSM. In the two-bounce case, we choose to define the PSM on $\mathcal{B}_s$ which is the reference surface of $\mathcal{M}_s$. While defining PSM on $\mathcal{B}_e$ is also possible, our choice is due to the reason that computing $\mathbf{x}_3$ from $\mathbf{x}_1, \mathbf{x}_2$ is straightforward, which requires evaluating a ray-intersection function H in Eqn. (12). But obtaining $\mathbf{x}_2$ from $\mathbf{x}_1$ and $\mathbf{x}_3$ is implicit, involving more matrix operations when computing Eqn. (30). Besides, representing the manifold on $\mathcal{M}_e$ also faces difficulties with handling visibility boundaries, which requires finding the corresponding point on $\mathcal{M}_s$ given a sampled point on $\mathcal{M}_e$.

Discontinuity of the pixel reconstruction filter. Yu et al. [2022] show that when the pixel reconstruction filter is discontinuous (e.g., a box filter), differentiating the integral over the pixel results in a pixel boundary integral. In our case, if $\mathbf{x}_0$ is the camera position, we transform Eqn. (15) to an integral over a footprint $\mathcal{P}(\theta) \subseteq \mathcal{B}_s$ corresponding to a given pixel:

$$I = \int_{\mathcal{P}(\theta)} \widehat{F}_p(\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2^c, \theta) dA(\mathbf{p}_1), \quad (50)$$

where $\widehat{F}_p$ computes the corresponding contribution with the filter but excludes the delta functions. Then our method can be extended to $\mathcal{P}$, which requires interpolation using the distance from $\mathbf{p}_1$ to boundaries on $\mathcal{P}$. We add a boundary-test function $B_p(\mathbf{p}_0, \mathbf{p}_1, \mathcal{P})$ that measures the distance from $\mathbf{x}_0, \mathbf{x}_1$ to the pixel boundary. More details are provided in the supplemental material.

Advanced optimization methods. For geometry optimization, we only provide results based on the method of Nicolet et al. [2021]. Optimizing specular or transparent objects often requires more complex techniques, as the high-frequency reflection and transmission are hard to match the target image. Our method focuses on computing derivatives with specular BSDFs. These computed derivatives can then be used for more advanced optimization approaches like the optimal transport [Xing et al. 2022], enabling the solution of more complex inverse rendering tasks. We leave the integration with more advanced optimization methods for future work.

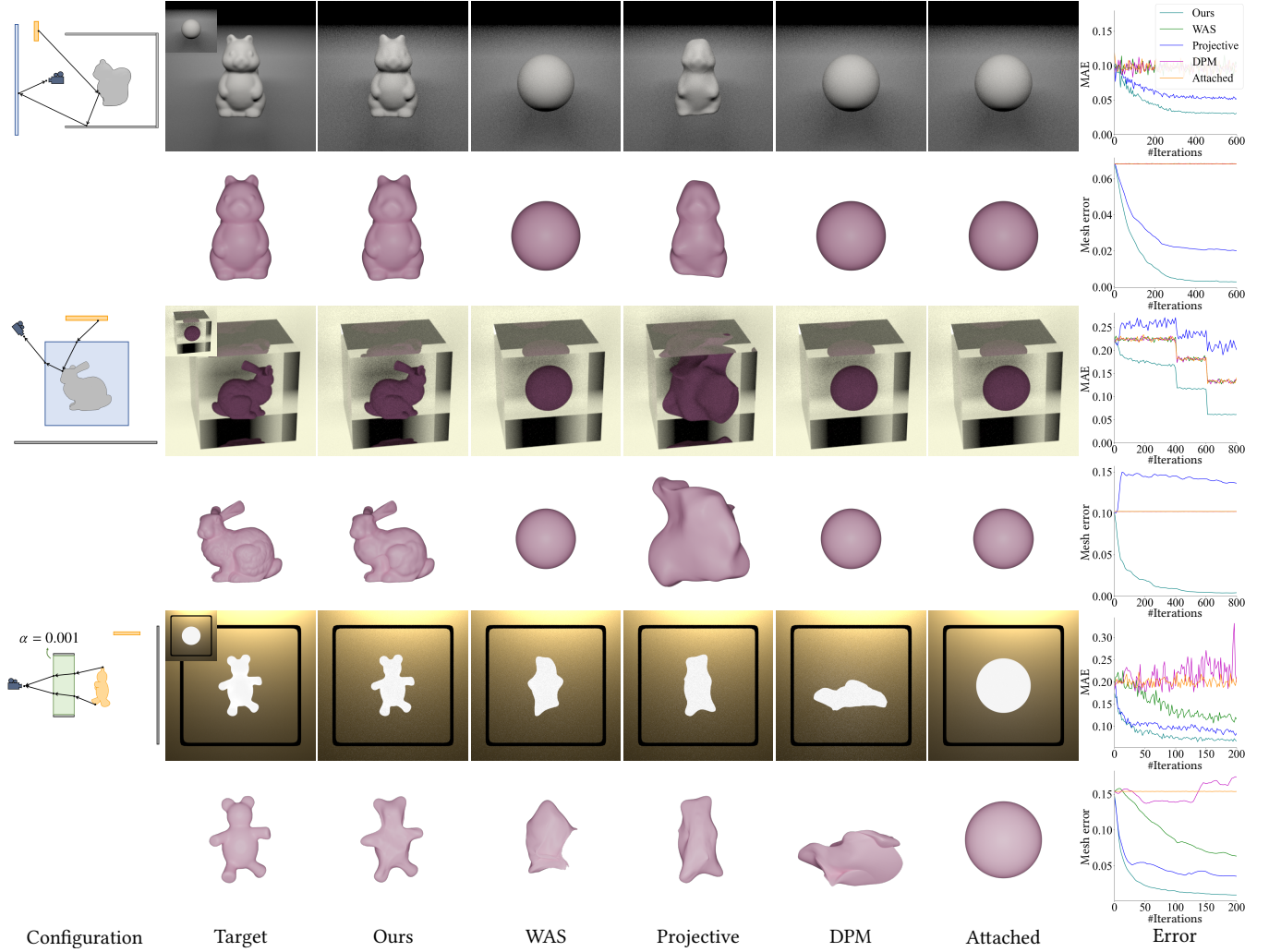


Fig. 19. Comparisons of gradient-based shape optimization between our method and previous works. We employ the method of Nicolet et al. [2021] to update mesh vertices with identical settings for all methods. We use images rendered with objects from 40 poses for targets. Our method drives optimization successfully to match targets, while other methods cannot converge well. Mesh error is computed as the Chamfer distance [Barrow et al. 1977], which is only used for evaluation.

Performances of the projective sampling. The Projective method performs poorly in most comparisons. The reason is that projective sampling’s performance is highly dependent on path length and whether the material is specular or glossy. Projective sampling projects samples onto nearby object boundaries and then employs a guiding data structure to sample the boundary integral. It performs relatively well on short glossy paths, as illustrated in the first row of Figs. 13 and 16. However, it struggles with longer paths or specular materials, since the increasing scattering events and sparse boundary integrand make it difficult for the projection to find valid samples. This is also evidenced by Wu et al. [2025].

Limitations. Our method has several limitations. First, our method is biased due to the interpolation of velocity fields. For complex scenes containing multiple tiny specular objects, the complexity

of manifold boundaries may cause our method to produce large errors. Second, the computation of velocity fields and divergences causes additional overhead. For short specular paths, we find that the overhead is minimal. But for long paths, the complexity of matrix operations increases (see the supplemental material for more details), which leads to more computational cost. Third, our method is limited to object surfaces and does not support volumetric rendering.

7 CONCLUSION

In this paper, we present a novel way to compute derivatives for specular BSDFs by considering the constrained specular manifold induced by the delta functions. At the core is the projected specular manifold (PSM), which enables the use of warped-area reparameterization to compute the boundary integral, avoiding the need for

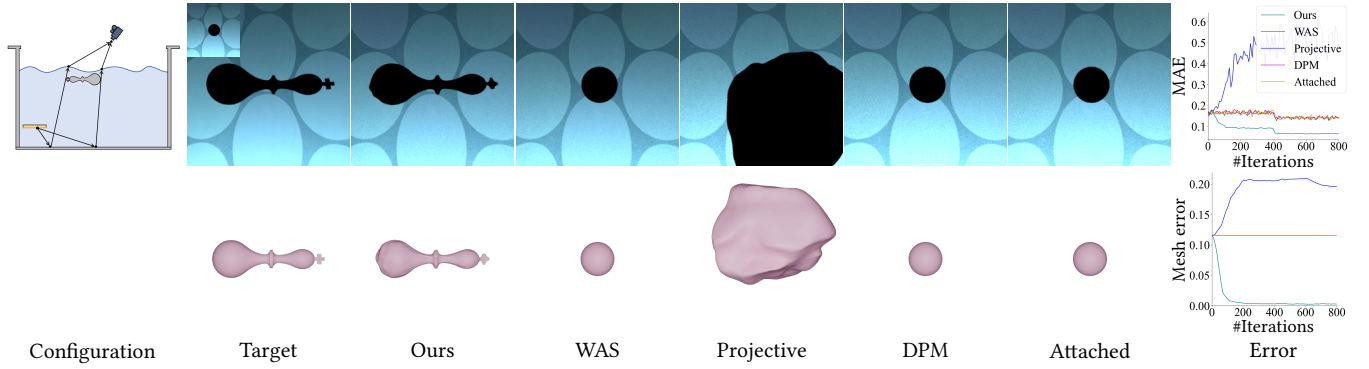


Fig. 20. Comparisons of shape optimization for the occluder. We limit the path length to 3 and optimize the occluder's geometry only based on the shape of the occluded region. The target images are rendered from 40 known poses of the occluder. Our method can handle visibility boundaries under specular BSDFs, while other methods fail. The mesh error is only used for evaluation.

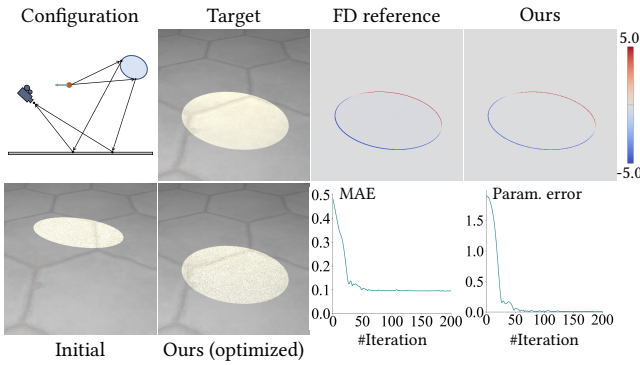


Fig. 21. Derivatives and inverse rendering results about point light with BDPT. In the first row, we compute derivatives with respect to the horizontal displacement of the point light. In the second row, we optimize the point light position based on its reflection pattern from a specular object onto a diffuse surface. The optimization is performed under single view. The parameter error is only used for evaluation.

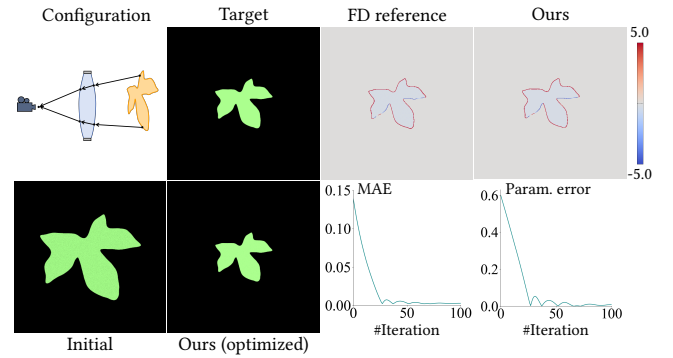


Fig. 22. Derivative computation and optimization for the IOR of the specular BSDF. We optimize the IOR of a convex lens to reduce the size of the light source in the rendered image, under the single view. The parameter error is only used for evaluation.

precisely locating the discontinuity curves on the specular manifold. To construct the velocity field required by warped-area reparameterization, we further propose a simple but effective direct interpolation method, utilizing the distances from vertices to discontinuity curves and the corresponding velocities. We then generalize the WAS method to the PSM for visibility boundaries and extend our method to glossy BSDFs. As a result, our method produces derivatives that match the reference closely while introducing low additional overhead.

In the future, generalizing our method to more complex discontinuities, such as those from discontinuous light sources or stylized functions [West and Mukherjee 2024], is an interesting direction. Complex specular scenes containing many small specular objects remain a challenge in rendering. Combining our method with specular path rendering techniques [Fan et al. 2025; Zeltner et al. 2020] is worth exploring.

Acknowledgments

We thank the reviewers for the valuable comments. This work is supported by Laoshan Laboratory (No. LSKJ202300305) and the National Natural Science Foundation of China (No. 62025207).

References

- Robert A Adams and John JF Fournier. 2003. *Sobolev spaces*. Vol. 140. Elsevier.
- Sai Praveen Bangaru, Michael Gharbi, Fujun Luan, Tzu-Mao Li, Kalyan Sunkavalli, Milos Hasan, Sai Bi, Zexiang Xu, Gilbert Bernstein, and Fredo Durand. 2022. Differentiable Rendering of Neural SDFs through Reparameterization. In *SIGGRAPH Asia 2022 Conference Papers* (Daegu, Republic of Korea) (SA '22). Association for Computing Machinery, New York, NY, USA, Article 5, 9 pages.
- Sai Praveen Bangaru, Tzu-Mao Li, and Frédo Durand. 2020. Unbiased warped-area sampling for differentiable rendering. *ACM Trans. Graph.* 39, 6, Article 245 (Nov. 2020), 18 pages.
- Sai Praveen Bangaru, Lifan Wu, Tzu-Mao Li, Jacob Munkberg, Gilbert Bernstein, Jonathan Ragan-Kelley, Frédo Durand, Aaron Lefohn, and Yong He. 2023. SLANG.D: Fast, Modular and Differentiable Shader Programming. *ACM Trans. Graph.* 42, 6, Article 264 (Dec. 2023), 28 pages.
- H. G. Barrow, J. M. Tenenbaum, R. C. Bolles, and H. C. Wolf. 1977. Parametric correspondence and chamfer matching: two new techniques for image matching. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence - Volume 2* (Cambridge, USA) (IJCAI'77). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 659–663.

- Petr Beckmann and Andre Spizzichino. 1963. The scattering of electromagnetic waves from rough surfaces. *Pergamon Press* (1963).
- Yash Belhe, Ishit Mehta, Wesley Chang, Iliyan Georgiev, Michael Gharbi, Ravi Ramamoorthi, and Tzu-Mao Li. 2025. Automatic Sampling for Discontinuities in Differentiable Shaders. *ACM Trans. Graph.* 44, 6, Article 209 (Dec. 2025), 19 pages.
- Yash Belhe, Bing Xu, Sai Praveen Bangaru, Ravi Ramamoorthi, and Tzu-Mao Li. 2024. Importance Sampling BRDF Derivatives. *ACM Trans. Graph.* 43, 3, Article 25 (April 2024), 21 pages.
- Paolo Cermelli, Eliot Fried, and Morton E. Gurtin. 2005. Transport relations for surface integrals arising in the formulation of balance laws for evolving fluid interfaces. *Journal of Fluid Mechanics* 544 (2005), 339–351.
- Zhimin Fan, Pengcheng Shi, Mufan Guo, Ruoyu Fu, Yanwen Guo, and Jie Guo. 2024. Conditional Mixture Path Guiding for Differentiable Rendering. *ACM Trans. Graph.* 43, 4, Article 48 (July 2024), 11 pages.
- Zhimin Fan, Chen Wang, Yiming Wang, Boxuan Li, Yuxuan Guo, Ling-Qi Yan, Yanwen Guo, and Jie Guo. 2025. Bernstein Bounds for Caustics. *ACM Trans. Graph.* 44, 4, Article 63 (July 2025), 15 pages.
- Wenzel Jakob and Steve Marschner. 2012. Manifold exploration: a Markov Chain Monte Carlo technique for rendering scenes with difficult specular transport. *ACM Trans. Graph.* 31, 4, Article 58 (July 2012), 13 pages.
- Wenzel Jakob, Sébastien Speierer, Nicolas Roussel, Merlin Nimier-David, Delio Vicini, Tizian Zeltner, Baptiste Nicolet, Miguel Crespo, Vincent Leroy, and Ziyi Zhang. 2022. *Mitsuba 3 renderer*. <https://mitsuba-renderer.org>.
- Tzu-Mao Li, Miika Aittala, Frédo Durand, and Jaakko Lehtinen. 2018. Differentiable Monte Carlo ray tracing through edge sampling. *ACM Trans. Graph.* 37, 6, Article 222 (Dec. 2018), 11 pages.
- Guillaume Loubet, Nicolas Holzschuch, and Wenzel Jakob. 2019. Reparameterizing discontinuous integrands for differentiable rendering. *ACM Trans. Graph.* 38, 6, Article 228 (Nov. 2019), 14 pages.
- Haolin Lu, Delio Vicini, Wesley Chang, and Tzu-Mao Li. 2025. Vector-Valued Monte Carlo Integration Using Ratio Control Variates. *ACM Trans. Graph.* 44, 4, Article 100 (July 2025), 16 pages.
- Jesse Michel, Kevin Mu, Xuanda Yang, Sai Praveen Bangaru, Elias Rojas Collins, Gilbert Bernstein, Jonathan Ragan-Kelley, Michael Carbin, and Tzu-Mao Li. 2024. Distributions for Compositionally Differentiating Parametric Discontinuities. *Proc. ACM Program. Lang.* 8, OOPSLA1, Article 126 (April 2024), 30 pages.
- 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.
- William S. Moses, Valentin Churavy, Ludger Paehler, Jan Hückelheim, Sri Hari Krishna Narayanan, Michel Schanen, and Johannes Doerfert. 2021. Reverse-Mode Automatic Differentiation and Optimization of GPU Kernels via Enzyme. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis* (St. Louis, Missouri) (SC '21). Association for Computing Machinery, New York, NY, USA, Article 61, 16 pages.
- William S. Moses, Sri Hari Krishna Narayanan, Ludger Paehler, Valentin Churavy, Michel Schanen, Jan Hückelheim, Johannes Doerfert, and Paul Hovland. 2022. Scalable Automatic Differentiation of Multiple Parallel Paradigms through Compiler Augmentation. In *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis* (Dallas, Texas) (SC '22). IEEE Press, Article 60, 18 pages.
- Baptiste Nicolet, Alec Jacobson, and Wenzel Jakob. 2021. Large steps in inverse rendering of geometry. *ACM Trans. Graph.* 40, 6, Article 248 (Dec. 2021), 13 pages.
- Merlin Nimier-David, Sébastien Speierer, Benoît Ruiz, and Wenzel Jakob. 2020. Radiative backpropagation: an adjoint method for lightning-fast differentiable rendering. *ACM Trans. Graph.* 39, 4, Article 146 (Aug. 2020), 15 pages.
- Osborne Reynolds. 1969. Papers on mechanical and physical subjects. *International Journal of Heat and Mass Transfer* 12, 2 (1969), 129–136.
- Mariia Soroka, Christoph Peters, and Steve Marschner. 2025. Quadric-Based Silhouette Sampling for Differentiable Rendering. *ACM Trans. Graph.* 44, 4, Article 81 (July 2025), 20 pages.
- Xiaochun Tong, Hsueh-Ti Derek Liu, Yotam Gingold, and Alec Jacobson. 2023. Differentiable Heightfield Path Tracing with Accelerated Discontinuities. In *ACM SIGGRAPH 2023 Conference Proceedings* (Los Angeles, CA, USA) (SIGGRAPH '23). Association for Computing Machinery, New York, NY, USA, Article 19, 9 pages.
- T. S. Trowbridge and K. P. Reitz. 1975. Average irregularity representation of a rough surface for ray reflection. *J. Opt. Soc. Am.* 65, 5 (May 1975), 531–536.
- Delio Vicini, Sébastien Speierer, and Wenzel Jakob. 2021. Path replay backpropagation: differentiating light paths using constant memory and linear time. *ACM Trans. Graph.* 40, 4, Article 108 (July 2021), 14 pages.
- Delio Vicini, Sébastien Speierer, and Wenzel Jakob. 2022. Differentiable signed distance function rendering. *ACM Trans. Graph.* 41, 4, Article 125 (July 2022), 18 pages.
- Bruce Walter, Steve Marschner, Hongsong Li, and Kenneth E. Torrance. 2007. Microfacet Models for Refraction through Rough Surfaces. In *Rendering Techniques*.
- Zichen Wang, Xi Deng, Ziyi Zhang, Wenzel Jakob, and Steve Marschner. 2024. A Simple Approach to Differentiable Rendering of SDFs. In *SIGGRAPH Asia 2024 Conference Papers* (Tokyo, Japan) (SA '24). Association for Computing Machinery, New York, NY, USA, Article 119, 11 pages.
- Rex West and Sayan Mukherjee. 2024. Stylized Rendering as a Function of Expectation. *ACM Trans. Graph.* 43, 4, Article 96 (July 2024), 19 pages.
- Lifan Wu, Guangyan Cai, Ravi Ramamoorthi, and Shuang Zhao. 2021. Differentiable time-gated rendering. *ACM Trans. Graph.* 40, 6, Article 287 (Dec. 2021), 16 pages.
- Lifan Wu, Nathan Morrical, Sai Praveen Bangaru, Rohan Sawhney, Shuang Zhao, Chris Wyman, Ravi Ramamoorthi, and Aaron Lefohn. 2025. Unbiased Differential Visibility Using Fixed-Step Walk-on-Spherical-Caps And Closest Silhouettes. *ACM Trans. Graph.* 44, 4, Article 79 (July 2025), 16 pages.
- Jiankai Xing, Xuejun Hu, Fujun Luan, Ling-Qi Yan, and Kun Xu. 2023. Extended Path Space Manifolds for Physically Based Differentiable Rendering. In *SIGGRAPH Asia 2023 Conference Papers* (Sydney, NSW, Australia) (SA '23). Association for Computing Machinery, New York, NY, USA, Article 30, 11 pages.
- Jiankai Xing, Zengyu Li, Fujun Luan, and Kun Xu. 2024. Differentiable Photon Mapping using Generalized Path Gradients. *ACM Trans. Graph.* 43, 6, Article 257 (Nov. 2024), 15 pages.
- Jiankai Xing, Fujun Luan, Ling-Qi Yan, Xuejun Hu, Houde Qian, and Kun Xu. 2022. Differentiable Rendering Using RGBX Derivatives and Optimal Transport. *ACM Trans. Graph.* 41, 6, Article 189 (Nov. 2022), 13 pages.
- Peiyu Xu, Sai Bangaru, Tzu-Mao Li, and Shuang Zhao. 2023. Warped-Area Reparameterization of Differential Path Integrals. *ACM Trans. Graph.* 42, 6, Article 213 (Dec. 2023), 18 pages.
- Peiyu Xu, Sai Bangaru, Tzu-Mao Li, and Shuang Zhao. 2024. Markov-Chain Monte Carlo Sampling of Visibility Boundaries for Differentiable Rendering. In *SIGGRAPH Asia 2024 Conference Papers* (Tokyo, Japan) (SA '24). Association for Computing Machinery, New York, NY, USA, Article 118, 11 pages.
- 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, Article 123 (July 2022), 13 pages.
- Yuting Yang, Connelly Barnes, Andrew Adams, and Adam Finkelstein. 2022. A δ : autodiff for discontinuous programs - applied to shaders. *ACM Trans. Graph.* 41, 4, Article 135 (July 2022), 24 pages.
- Shinyoung Yi, Donggun Kim, Kiseok Choi, Adrian Jarabo, Diego Gutierrez, and Min H. Kim. 2021. Differentiable transient rendering. *ACM Trans. Graph.* 40, 6, Article 286 (Dec. 2021), 11 pages.
- 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 (Nov. 2022), 16 pages.
- Tizian Zeltner, Iliyan Georgiev, and Wenzel Jakob. 2020. Specular manifold sampling for rendering high-frequency caustics and glints. *ACM Trans. Graph.* 39, 4, Article 149 (Aug. 2020), 15 pages.
- Tizian Zeltner, Sébastien Speierer, Iliyan Georgiev, and Wenzel Jakob. 2021. Monte Carlo estimators for differential light transport. *ACM Trans. Graph.* 40, 4, Article 78 (July 2021), 16 pages.
- Cheng Zhang, Zhao Dong, Michael Doggett, and Shuang Zhao. 2021a. Antithetic sampling for Monte Carlo differentiable rendering. *ACM Trans. Graph.* 40, 4, Article 77 (July 2021), 12 pages.
- Cheng Zhang, Bailey Miller, Kai Yan, Ioannis Gkioulekas, and Shuang Zhao. 2020. Path-space differentiable rendering. *ACM Trans. Graph.* 39, 4, Article 143 (Aug. 2020), 19 pages.
- Cheng Zhang, Lifan Wu, Changxi Zheng, Ioannis Gkioulekas, Ravi Ramamoorthi, and Shuang Zhao. 2019. A differential theory of radiative transfer. *ACM Trans. Graph.* 38, 6, Article 227 (Nov. 2019), 16 pages.
- Cheng Zhang, Zihan Yu, and Shuang Zhao. 2021b. Path-space differentiable rendering of participating media. *ACM Trans. Graph.* 40, 4, Article 76 (July 2021), 15 pages.
- Ziyi Zhang, Nicolas Roussel, and Wenzel Jakob. 2023. Projective Sampling for Differentiable Rendering of Geometry. *ACM Trans. Graph.* 42, 6, Article 212 (Dec. 2023), 14 pages.
- Ziyi Zhang, Nicolas Roussel, and Wenzel Jakob. 2025. Many-Worlds Inverse Rendering. *ACM Trans. Graph.* 45, 1, Article 8 (Oct. 2025), 17 pages.
- Yang Zhou, Lifan Wu, Ravi Ramamoorthi, and Ling-Qi Yan. 2021. Vectorization for Fast, Analytic, and Differentiable Visibility. *ACM Trans. Graph.* 40, 3, Article 27 (July 2021), 21 pages.