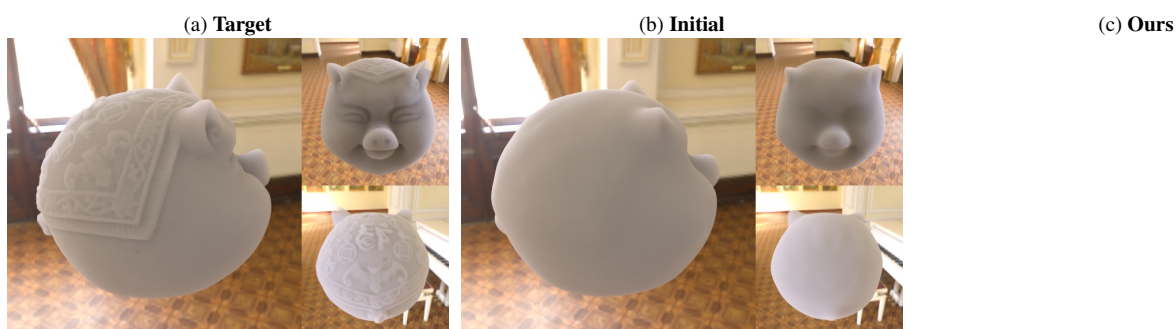


# Efficient Path-Space Differentiable Volume Rendering With Respect To Shapes

Z. Yu<sup>1,2</sup>, C. Zhang<sup>2</sup>, O. Maury<sup>2</sup>, C. Hery<sup>2</sup>, Z. Dong<sup>2</sup>, S. Zhao<sup>1</sup>

<sup>1</sup>University of California, Irvine

<sup>2</sup>Meta Reality Labs Research



**Figure 1:** We introduce a new technique for the differentiable rendering of translucent objects with respect to their shapes. Compared to state-of-the-art formulations [ZWZ\*19, ZYZ21], our technique enjoys the advantages of not requiring silhouette detection or specifying change rates inside translucent objects, and allows efficient shape optimization for translucent objects. In this example, using 40 images of a translucent pig (with three shown), we optimize the shape of this object. (Please use Adobe Acrobat and click column (c) to view an animation of the optimization process.)

## Abstract

Differentiable rendering of translucent objects with respect to their shapes has been a long-standing problem. State-of-the-art methods require detecting object silhouettes or specifying change rates inside translucent objects—both of which can be expensive for translucent objects with complex shapes.

In this paper, we address this problem for translucent objects with no refractive or reflective boundaries. By reparameterizing interior components of differential path integrals, our new formulation does not require change rates to be specified in the interior of objects. Further, we introduce new Monte Carlo estimators based on this formulation that do not require explicit detection of object silhouettes.

## 1. Introduction

Translucent materials such as marble, wax, and human skin are ubiquitous in the real world and crucial to many applications, such as computational fabrication, remote sensing, and biomedical imaging. Therefore, simulating the appearances, or *forward rendering*, of these materials has been an active research topic in computer graphics, leading to mature algorithms capable of efficiently and accurately rendering translucent materials in complex virtual environments.

*Differentiable rendering* concerns estimating derivatives of radiometric measurements (e.g., radiance) with respect to differential changes of a scene. These techniques have a wide range of appli-

cations by facilitating, for instance, the use of gradient-based optimization for solving *inverse-rendering* problems.

Recently, great progress has been made in the differentiable rendering of translucent materials. Mathematically, differential radiative transfer [ZWZ\*19] differentiates the widely adopted radiative transfer framework [Cha60]. Also, generalized differential path integrals [ZYZ21] present an alternative formulation based on the path-integral formulation [PKK00]. Both formulations offer the same level of physical accuracy and the generality of differentiation with respect to arbitrary scene parameters.

On the other hand, efficient differentiation with respect to shapes of translucent objects remains a challenge. Although the aforemen-

tioned mathematical formulations both support such geometric differentiations, the resulting algorithms require, respectively, detecting object silhouettes and specifying changing rates in the interior of translucent objects—both of which can be expensive for objects with complex geometries.

In this paper, we take a first step to address this problem and focus on differentiating shapes of translucent objects without refractive or reflective boundaries. Our technique supports full volumetric light transport (e.g., multiple scattering) and scales better for translucent objects with complex geometries than previous methods [ZWZ\*19, ZYZ21].

Concretely, our contributions include:

- Devising a reparameterized form of (the interior component of) Zhang et al.’s [ZYZ21] generalized differential path integrals (§4). Our formulation only requires change rates (with respect to shapes of object boundaries) to be evaluated on object boundaries.
- Introducing unbiased Monte Carlo estimators for our reparameterized path integrals (§5).

To validate our theory and algorithms, we compare our derivative estimates with those produced using finite differences (Figure 7). We further demonstrate the effectiveness of our method using differentiable rendering (Figures 8 and 9) and inverse rendering (Figures 10 and 11) examples.

## 2. Related Work

**Forward volume rendering.** Monte Carlo methods have been the “gold standard” for simulating photon and neutron transport in complex environments [SG69]. In computer graphics, volume path tracing (e.g., [KVH84, CPP\*05]) is capable of producing unbiased and consistent estimates of radiometric measures. Further, based on the path-integral formulation [Vea97, PKK00], more advanced bidirectional [LW96] and Markov-Chain Monte Carlo (e.g., [PKK00, KSKAC02, JM12]) methods have been introduced to allow efficient simulation of challenging light-transport effects such as glossy interreflection and caustics.

For a comprehensive survey on Monte Carlo volume rendering techniques, we refer to Novak et al.’s survey [NGHJ18].

**Differentiable volume rendering.** Specialized differentiable volume rendering has been used to solve analysis-by-synthesis problems in volumetric scattering [GZB\*13], prefiltering of high-resolution volumes [ZWDR16], and fabrication of translucent materials [SRB\*19]. All these methods compute derivatives with respect to specific material properties like optical density.

For general-purpose differentiable volume rendering, Che et al. [CLZ\*20] developed a system [CLZ\*20] capable of computing derivatives with respect to optical material and local normal properties. Recently, efficient tracking methods [NDMKJ22], differentiation techniques [NDSRJ20, VSJ21], and differentiable rendering systems [NDVZJ19, JSRV22] have been introduced to allow differentiable volume rendering to scale to complex scenes with millions of parameters. Vicini et al. [VJK21] introduced a non-exponential transmittance model that improves the volumetric

representation of opaque surfaces and their reconstruction using differentiable rendering. Unfortunately, when applied to volume rendering, these methods have difficulties differentiating with respect to shapes as they neglect discontinuities of optical densities that evolve with boundary shapes.

To overcome this challenge, Zhang et al. [ZWZ\*19] adopted Monte Carlo edge sampling [LADL18] (originally developed for differentiable rendering of surfaces) and introduced the first differentiable volumetric path tracer capable of differentiating with respect to shapes of translucent objects. This technique—which we refer to as “DTRT”—requires detection of object silhouettes and can be prohibitively expensive for complex scenes.

Later, Zhang et al. [ZYZ21] introduced a technique—which we refer to as “PSDR”—by generalizing the formulation of differential path integrals [ZMY\*20] to volumetric light transport. Offering the same level of generality as DTRT, PSDR allows the development of new Monte Carlo methods with better efficiency by not requiring expensive silhouette detection. On the other hand, when differentiating with respect to shapes of translucent objects, PSDR requires specifying change rates in the interior of these objects, which can be expensive. In §4, we will address this problem by introducing a reparameterized formulation that does not involve change rates inside translucent objects.

Recently, Deng et al. [DLW\*22] extended PSDR to support bidirectional subsurface scattering distribution functions (BSSRDFs)—an approximated formulation of volumetric light transport. Our technique, on the other hand, uses the full radiative transfer model [Cha60] and is orthogonal to Deng et al.’s approach.

## 3. Preliminaries

In this paper, we tackle differentiable rendering of translucent objects with no refractive interfaces or reflective boundaries. Further, we focus on differentiation with respect to shapes of such objects.

In the following, we review the related mathematical formulations: generalized path integral (§3.1) and its differential expression (§3.2). Then, we formally specify our problem (§3.3).

### 3.1. Generalized Path Integral

The formulation of *generalized path integrals* [Vea97, PKK00] has been the mathematical foundation of many modern Monte Carlo rendering techniques such as bidirectional path tracing [VG95, LW96] and Metropolis light transport [VG97].

Under this formulation, the response  $I \in \mathbb{R}$  of a radiometric detector is expressed as a **generalized path integral** [Vea97, PKK00] of the form:

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

where  $\bar{\mathbf{x}} = (\mathbf{x}_0, \dots, \mathbf{x}_N)$  denotes a **light transport path** (with  $\mathbf{x}_0$  on a light source and  $\mathbf{x}_N$  on a detector). Additionally,  $\Omega$  is the **path space** with measure  $\mu$ , and  $f$  is the **measurement contribution function**. In the following, we summarize the definitions of these terms.

**Path space and measure.** With the presence of translucent objects (without refractive/reflective boundaries), the path space can be defined as

$$\Omega = \bigcup_{N \geq 1} (\mathcal{V} \cup \mathcal{M})^{N+1}, \quad (2)$$

where  $\mathcal{V}$  and  $\mathcal{M}$  denote, respectively, the interior of all translucent objects and the surfaces of all opaque objects. For a light transport path  $\bar{\mathbf{x}} = (\mathbf{x}_0, \dots, \mathbf{x}_N) \in \Omega$ , the measure  $\mu$  satisfies

$$d\mu(\bar{\mathbf{x}}) = \prod_{n=0}^N \begin{cases} dA(\mathbf{x}_n), & (\mathbf{x}_n \in \mathcal{M}) \\ dV(\mathbf{x}_n), & (\mathbf{x}_n \in \mathcal{V}) \end{cases} \quad (3)$$

where  $dA$  and  $dV$  denote, respectively, the surface-area and volume measures.

**Measurement contribution.** Given a light path  $\bar{\mathbf{x}} = (\mathbf{x}_0, \dots, \mathbf{x}_N)$ , its measurement contribution  $f(\bar{\mathbf{x}})$  equals the product of per-vertex terms  $f_v$  and per-segment ones  $G$ :

$$f(\bar{\mathbf{x}}) := \left[ \prod_{n=0}^N f_v(\mathbf{x}_{n-1} \rightarrow \mathbf{x}_n \rightarrow \mathbf{x}_{n+1}) \right] \left[ \prod_{n=1}^N G(\mathbf{x}_{n-1} \leftrightarrow \mathbf{x}_n) \right]. \quad (4)$$

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

$$f_v(\mathbf{x}_{n-1} \rightarrow \mathbf{x}_n \rightarrow \mathbf{x}_{n+1}) := \begin{cases} f_s(\mathbf{x}_{n-1} \rightarrow \mathbf{x}_n \rightarrow \mathbf{x}_{n+1}), & (0 < n < N, \mathbf{x}_n \in \mathcal{M}) \\ \sigma_s(\mathbf{x}_n) f_p(\mathbf{x}_{n-1} \rightarrow \mathbf{x}_n \rightarrow \mathbf{x}_{n+1}), & (0 < n < N, \mathbf{x}_n \in \mathcal{V}) \\ L_e(\mathbf{x}_0 \rightarrow \mathbf{x}_1), & (n = 0) \\ W_e(\mathbf{x}_{N-1} \rightarrow \mathbf{x}_N), & (n = N) \end{cases} \quad (5)$$

where  $f_s$  is the **bidirectional scattering distribution function** (BSDF),  $f_p$  denotes the **single-scattering phase function**,  $\sigma_s$  is the **scattering coefficient**, and  $L_e$  and  $W_e$  capture the **source emission** and **detector importance** (or response). Additionally, the per-segment **generalized geometric term** is given by:

$$G(\mathbf{x} \leftrightarrow \mathbf{y}) := T(\mathbf{x} \leftrightarrow \mathbf{y}) \nabla(\mathbf{x} \leftrightarrow \mathbf{y}) \frac{D_{\mathbf{x}}(\mathbf{y}) D_{\mathbf{y}}(\mathbf{x})}{\|\mathbf{x} - \mathbf{y}\|^2}, \quad (6)$$

where, for any points  $\mathbf{x}, \mathbf{y} \in \mathcal{V} \cup \mathcal{M}$ ,

$$D_{\mathbf{x}}(\mathbf{y}) := \begin{cases} |\mathbf{n}(\mathbf{x}) \cdot \vec{\mathbf{x}\mathbf{y}}|, & (\mathbf{x} \in \mathcal{M}) \\ 1, & (\mathbf{x} \in \mathcal{V}) \end{cases} \quad (7)$$

with  $\mathbf{n}(\mathbf{x})$  being the (unit-length) surface normal at  $\mathbf{x}$ , and  $\vec{\mathbf{x}\mathbf{y}} := (\mathbf{y} - \mathbf{x}) / \|\mathbf{y} - \mathbf{x}\|$  is the unit direction from  $\mathbf{x}$  to  $\mathbf{y}$ .

Lastly, in Eq. (6),  $\nabla(\mathbf{x} \leftrightarrow \mathbf{y})$  denotes the **mutual visibility function**, and  $T(\mathbf{x} \leftrightarrow \mathbf{y})$  indicates the **transmittance** between  $\mathbf{x}$  and  $\mathbf{y}$  that equals

$$T(\mathbf{x} \leftrightarrow \mathbf{y}) = \exp \left[ -\int_{\vec{\mathbf{x}\mathbf{y}}} \sigma_t d\ell \right], \quad (8)$$

where  $\sigma_t$  is the **extinction coefficient** (or optical density);  $\vec{\mathbf{x}\mathbf{y}}$  denotes the line segment connecting  $\mathbf{x}$  and  $\mathbf{y}$ ; and  $\ell$  is the curve-length measure.

### 3.2. Generalized Differential Path Integral

Zhang et al. [ZYZZ1] have recently derived derivatives of generalized path integrals as *generalized differential path integrals*—a result we briefly revisit in the following.

**Material-form reparameterization.** When the scene geometry (e.g., object location) depends on some parameter  $\theta$ , Zhang et al. proposed to apply a change of variable to the ordinary path integral so that the new domain of integration becomes independent of  $\theta$ , simplifying differentiations with respect to this parameter.

To this end, one can parameterize the scene geometry using a **motion**  $\mathbf{X}$  such that  $\mathbf{X}(\cdot, \theta)$  is a differentiable bijection that maps some fixed **reference volume**  $\mathcal{B}_{\mathcal{V}}$  and **surface**  $\mathcal{B}_{\mathcal{M}}$  to, respectively, the interior  $\mathcal{V}(\theta)$  of translucent objects and the surfaces  $\mathcal{M}(\theta)$  of opaque objects. Lastly, we call any  $\mathbf{x} \in \mathcal{V}(\theta) \cup \mathcal{M}(\theta)$  a **spatial point** and any  $\mathbf{p} \in \mathcal{B}_{\mathcal{V}} \cup \mathcal{B}_{\mathcal{M}}$  a **material point**.

Then, applying the change of variable given by the mapping  $\mathbf{X}(\cdot, \theta)$  to each vertex of a light path allows Eq. (1) to be rewritten as the **material-form generalized path integral**:

$$I = \int_{\hat{\Omega}} \hat{f}(\bar{\mathbf{p}}) d\mu(\bar{\mathbf{p}}), \quad (9)$$

where the domain of integration is the **material path space**  $\hat{\Omega}$  given by

$$\hat{\Omega} := \bigcup_{N \geq 1} (\mathcal{B}_{\mathcal{V}} \cup \mathcal{B}_{\mathcal{M}})^{N+1}, \quad (10)$$

which comprises **material light paths**  $\bar{\mathbf{p}} = (\mathbf{p}_0, \dots, \mathbf{p}_N)$ .

In Eq. (9),  $\hat{f}$  is the **material measurement contribution** provided by the original measurement contribution of Eq. (4) and Jacobian determinants  $J$  capturing this change of variable:

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

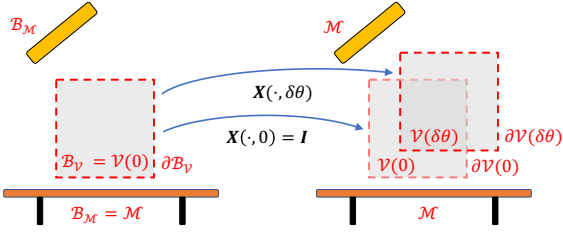
where  $\bar{\mathbf{x}} = (\mathbf{x}_0, \dots, \mathbf{x}_n)$  with  $\mathbf{x}_n = \mathbf{X}(\mathbf{p}_n, \theta)$ , and

$$J(\mathbf{p}) := \begin{cases} \|dA(\mathbf{X}(\mathbf{p}, \theta)) / dA(\mathbf{p})\|, & (\mathbf{p} \in \mathcal{B}_{\mathcal{M}}) \\ \|dV(\mathbf{X}(\mathbf{p}, \theta)) / dV(\mathbf{p})\|, & (\mathbf{p} \in \mathcal{B}_{\mathcal{V}}) \end{cases} \quad (12)$$

**Generalized differential path integral.** In general, the derivative of Eq. (9) have been proven [ZMY\*20, ZYZ21] to equal

$$\frac{dI}{d\theta} = \int_{\hat{\Omega}} \frac{d\hat{f}(\bar{\mathbf{p}})}{d\theta} d\mu(\bar{\mathbf{p}}) + \int_{\Delta\hat{\Omega}} \Delta\hat{f}(\bar{\mathbf{p}}) v_{\perp}(\mathbf{p}_K) d\mu(\bar{\mathbf{p}}). \quad (13)$$

This result is called the (material-form) **generalized differential path integral**. In Eq. (13), the interior component is over the same material path space  $\hat{\Omega}$  as the material-form generalized path integral in Eq. (9). In contrast, the boundary integral is over the **material boundary path space**  $\Delta\hat{\Omega}$  comprised of **material boundary paths**. These paths are identical to the ordinary ones except for having one **boundary segment**  $\overline{\mathbf{p}_{K-1} \mathbf{p}_K}$  such that the spatial representations  $\mathbf{x}_{K-1} = \mathbf{X}(\mathbf{p}_{K-1}, \theta)$  and  $\mathbf{x}_K = \mathbf{X}(\mathbf{p}_K, \theta)$  of the endpoints reside on visibility boundaries with respect to each other. Additionally,  $\Delta\hat{f}$  denotes the difference in material measurement contribution  $\hat{f}$  across the discontinuity boundary (at  $\mathbf{p}_K$ ), and  $v_{\perp}(\mathbf{p}_K)$  is the normal change rate (with respect to  $\theta$ ) of the discontinuity boundary. We refer the readers to the work by Zhang et al. [ZYZZ1] for the precise definitions of the boundary component.



**Figure 2: Material-form reparameterization:** Zhang et al. [ZYZ21] proposed to reparameterize surfaces  $\mathcal{M}(\theta)$  of opaque objects and volumes  $\mathcal{V}(\theta)$  of translucent objects that evolve with some scene parameter  $\theta$ . This can be achieved using a differentiable bijection  $X(\cdot, \theta)$  that maps some  $\theta$ -independent reference surface  $\mathcal{B}_{\mathcal{M}} = \mathcal{M}(0)$  and volume  $\mathcal{B}_{\mathcal{V}} = \mathcal{V}(0)$  to the evolving  $\mathcal{M}(\theta)$  and  $\mathcal{V}(\theta)$ , respectively.

### 3.3. Problem Specification

We focus on differentiable renderings of translucent objects with respect to their shapes expressed with some parameter  $\theta$  and make the following assumptions:

- S.1** The translucent objects contain no refractive or reflective boundaries (i.e., with “null” BSDFs)—which ensures that these objects cause no visibility discontinuities. Further, there are no visibility boundaries (such as cast or volume shadows) that depend on the parameter  $\theta$ .
- S.2** The parameter  $\theta$  controls only shapes of translucent objects (but not their material properties such as extinction and scattering coefficients).

We note that, Our derivatives use one parameter  $\theta \in \mathbb{R}$  for simplicity and can be generalized for multiple parameters  $\boldsymbol{\theta} \in \mathbb{R}^m$  easily. In §4, we will discuss why Assumption **S.1** is needed as well as how it can be relaxed. When solving practical inverse-rendering problems, our technique can be used to compute partial derivatives with respect to object shapes. Other partial derivatives with respect to, for example, material optical properties can be computed using other existing methods such as path-replay backpropagation [VSJ21].

For notational simplicity, we assume without loss of generality that all differentiations with respect to  $\theta$  are evaluated at  $\theta = 0$ , and define

$$d_{\theta} h := [dh/d\theta]_{\theta=0}, \quad \partial_{\theta} h := [\partial h/\partial \theta]_{\theta=0}, \quad (14)$$

as total and partial derivatives of  $h$  (for any scalar- or vector-valued function  $h$ ).

Under the aforementioned settings, the boundary component of the generalized differential path integral (13) vanishes, yielding

$$d_{\theta} I = \int_{\Omega} (d_{\theta} \hat{f}(\bar{\mathbf{p}})) d\mu(\bar{\mathbf{p}}), \quad (15)$$

with  $\hat{f}$  defined in Eq. (11).

**Reference configuration.** As illustrated in Figure 2, similar to prior works [ZYZ21], we set the reference volume to  $\mathcal{B}_{\mathcal{V}} = \mathcal{V}(0)$

and the reference surface to  $\mathcal{B}_{\mathcal{M}} = \mathcal{M}(0)$ . This causes the mapping  $X(\cdot, 0)$  to reduce to the identity map and the Jacobian determinants  $\|\mathbf{dx}_n/d\mathbf{p}_n\|$  defined in Eq. (12) to equal one. Further, the path space  $\Omega(0)$  coincides with the material path space  $\Omega$ .

We note that, when  $\theta$  controls scene geometry, the derivative of  $\|\mathbf{dx}_n/d\mathbf{p}_n\|$  with respect to  $\theta$  is generally nonzero—even at  $\theta = 0$ .

**Challenges.** Evaluating  $d_{\theta} \hat{f}(\bar{\mathbf{p}})$  for any path  $\bar{\mathbf{p}} = (\mathbf{p}_0, \dots, \mathbf{p}_N)$  requires specifying and differentiating the mapping  $X(\cdot, \theta)$  at each vertex  $\mathbf{p}_n$ . When optimizing the shapes of translucent objects expressed using triangle meshes, the scene parameter  $\theta$  controls the positions of mesh vertices. In this case, the mapping  $X(\cdot, \theta)$  is not uniquely determined for  $\mathbf{p}_n$  in the interior of the reference volume (i.e.,  $\mathbf{p}_n \in \mathcal{B}_{\mathcal{V}}$ ), and the choice of the mapping  $X(\mathbf{p}_n, \theta)$  does not affect the resulting derivative  $d_{\theta} I$ —as long as it is continuous.

Previously, Zhang et al. [ZYZ21] used tetrahedral meshes to interpolate  $X(\cdot, \theta)$  defined on the boundaries of translucent objects into their interiors. Despite offering high generality (by being capable of handling, for example, visibility discontinuities and refractive interfaces), this method introduces high computational overhead by requiring frequent tetrahedralizations and point-in-tetrahedron lookups.

To address this problem for translucent objects with no refractive/reflective boundaries, we will introduce a mathematical formulation that does not require evaluating  $X(\cdot, \theta)$  inside objects in §4 and new Monte Carlo estimators for this formulation in §5.

## 4. Reparameterized Interior Differential Path Integrals

Based on the aforementioned assumptions (**S.1** and **S.2**), we now introduce a reparameterized form of the interior differential path integral expressed in Eq. (15). The key idea is to convert the volume integral to a boundary area integral using the divergence theorem. By doing so, the terms related to the mapping  $X(\cdot, \theta)$  are moved from the interior to the boundary. This provides computational benefits as the mapping  $X(\cdot, \theta)$  does not need to be evaluated in the interior of the volume, avoiding the need for the expensive tetrahedralization. In the context of surface-based differentiable rendering, Bangaru et al. [BLD20], on the contrary, applied the divergence theorem in the opposite way to avoid dedicated data structure for edge sampling.

To facilitate our derivations, provided some reference volume  $\mathcal{B}_{\mathcal{V}}$  and motion  $X$ , we define the **velocity**  $\mathbf{v}$  as a vector field over  $\mathcal{B}_{\mathcal{V}}$  given by

$$\mathbf{v}(\mathbf{p}) := d_{\theta} X(\mathbf{p}, \theta). \quad (16)$$

Based on the velocity  $\mathbf{v}$ , we further define the **normal velocity**  $v_{\perp}$  as a scalar field over the boundary  $\partial\mathcal{B}_{\mathcal{V}}$  of the reference volume  $\mathcal{B}_{\mathcal{V}}$ :

$$v_{\perp}(\mathbf{p}) := \mathbf{v}(\mathbf{p}) \cdot \mathbf{n}(\mathbf{p}), \quad (17)$$

where  $\mathbf{n}$  denotes the outward unit-normal field associated with  $\partial\mathcal{B}_{\mathcal{V}}$ , and “ $\cdot$ ” is the dot-product operator.

### 4.1. Simple Volume Integrals

Before deriving our general result for material-form differential path integral defined in Eq. (15), we first consider the following

material-form differential volume integral:

$$\int_{\mathcal{B}_V} d_\theta \hat{h}(\mathbf{p}, \theta) dV(\mathbf{p}), \quad (18)$$

where  $\hat{h}$  is the given by some scalar-valued differentiable function  $h$  evaluated at  $\mathbf{x} = \mathbf{X}(\mathbf{p}, \theta)$ , and jacobian determinant  $J(\mathbf{p}, \theta) = \|\mathbf{dV}(\mathbf{x})/\mathbf{dV}(\mathbf{p})\|$  defined in Eq. (12) capturing the change of variable:

$$\hat{h}(\mathbf{p}, \theta) := h(\mathbf{x}, \theta) J(\mathbf{p}, \theta), \quad (19)$$

Based on the chain rule, it holds that

$$\int_{\mathcal{B}_V} d_\theta \hat{h} dV = \int_{\mathcal{B}_V} (\nabla \hat{h} \cdot \mathbf{v} + \partial_\theta \hat{h}) dV, \quad (20)$$

where  $\mathbf{v}$  is the velocity defined in Eq. (16). According to integration by parts, it holds that

$$\int_{\mathcal{B}_V} \mathbf{v} \cdot \nabla \hat{h} dV = \int_{\mathcal{B}_V} \nabla \cdot (\hat{h} \mathbf{v}) dA - \int_{\mathcal{B}_V} \hat{h} \nabla \cdot \mathbf{v} dV, \quad (21)$$

where “ $\nabla \cdot$ ” is the divergence operator. Then, applying the divergence theorem to first term on the right-hand side of Eq. (21) yields

$$\int_{\mathcal{B}_V} \mathbf{v} \cdot \nabla \hat{h} dV = \int_{\partial \mathcal{B}_V} \hat{h} v_\perp dA - \int_{\mathcal{B}_V} \hat{h} \nabla \cdot \mathbf{v} dV, \quad (22)$$

Substituting Eq. (22) back into Eq. (20) yields

$$\int_{\mathcal{B}_V} d_\theta \hat{h} dV = \int_{\partial \mathcal{B}_V} \hat{h} v_\perp dA + \int_{\mathcal{B}_V} (\partial_\theta \hat{h} - \hat{h} \nabla \cdot \mathbf{v}) dV, \quad (23)$$

where  $v_\perp$  is the scalar-valued normal velocity defined in Eq. (17).

As discussed in §3.3, under the configuration of reference volume  $\mathcal{B}_V = \mathcal{V}(0)$ , the mapping  $\mathbf{X}(\cdot, 0)$  reduces to the identity map, and  $J(\mathbf{p}, 0)$  equals one.

$$\hat{h}(\mathbf{p}, 0) = h(\mathbf{p}, 0) J(\mathbf{p}, 0) = h(\mathbf{p}, 0). \quad (24)$$

**Simple case.** When  $h(\mathbf{x}, \theta) = h(\mathbf{x})$  does not depend on  $\theta$  directly, it holds that

$$\partial_\theta \hat{h}(\mathbf{p}, 0) = h(\mathbf{p}, 0) \partial_\theta J(\mathbf{p}, 0), \quad (25)$$

Substituting Eqs. (24) and (25) into Eq. (23) gives

$$\dot{I} = \int_{\partial \mathcal{B}_V} h v_\perp dA + \int_{\mathcal{B}_V} h (\partial_\theta J - \nabla \cdot \mathbf{v}) dV, \quad (26)$$

where, as shown in Eq. (2) from Chapter III, Section 10 of the book by Gurtin [Gur81], it holds that

$$\partial_\theta J(\mathbf{p}, \theta) = J(\mathbf{p}, \theta) (\nabla \cdot \mathbf{v})(\mathbf{p}), \quad (27)$$

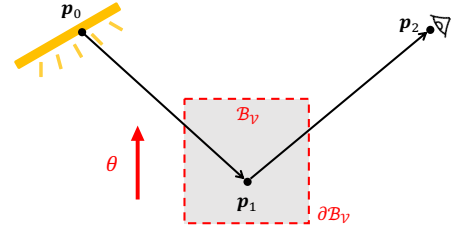
for all  $\mathbf{p} \in \mathcal{B}_V$ . Under the configuration of reference volume  $\mathcal{B}_V = \mathcal{V}(0)$ ,  $J(\mathbf{p}, 0)$  equals one, resulting in

$$\partial_\theta J(\mathbf{p}, 0) = (\nabla \cdot \mathbf{v})(\mathbf{p}), \quad (28)$$

This relation is a key ingredient for deriving the well-known Reynolds transport theorem [Rey03].

Given Eq. (28), the volume integral in Eq. (26) vanishes, allowing Eq. (18) to be rewritten as an integral over the boundary  $\partial \mathcal{B}_V$ :

$$\dot{I} = \int_{\partial \mathcal{B}_V} h v_\perp dA. \quad (29)$$



**Figure 3:** When a translucent object with “null” interfaces (illustrated as the gray rectangle) moves with the parameter  $\theta$ , the measurement contribution  $f(\bar{\mathbf{p}})$  can depend on  $\theta$ —even with the path  $\bar{\mathbf{p}}$  fixed.

**General case.** When the integrand  $h(\mathbf{x}, \theta)$  directly depends on the parameter  $\theta$ , we have

$$\partial_\theta \hat{h}(\mathbf{p}, 0) = h(\mathbf{p}, 0) \partial_\theta J(\mathbf{p}, 0) + \partial_\theta h(\mathbf{p}, 0). \quad (30)$$

By handling the first term on the right-hand side in a similar fashion as the simple case, we obtain a reparameterized form of Eq. (18):

$$\dot{I} = \int_{\mathcal{B}_V}^{\text{interior}} \partial_\theta h(\mathbf{p}, \theta) dV(\mathbf{p}) + \int_{\partial \mathcal{B}_V}^{\text{boundary}} h(\mathbf{p}, 0) v_\perp(\mathbf{p}) dA(\mathbf{p}). \quad (31)$$

Despite involving one extra boundary component than Eq. (18), the reparameterized Eq. (31) enjoys the key advantage of not requiring the mapping  $\mathbf{X}(\cdot, \theta)$  or the velocity  $\mathbf{v}$  to be calculated in the interior of the reference volume  $\mathcal{B}_V$ . This is because its interior component  $\int_{\mathcal{B}_V} \partial_\theta h(\mathbf{p}, \theta) dV(\mathbf{p})$  only takes *partial derivative* with respect to the parameter  $\theta$  and does not involve the mapping  $\mathbf{X}(\cdot, \theta)$ .

## 4.2. Volume Path Integrals

We now leverage the results derived in §4.1 to differentiate integrals over material paths  $\bar{\mathbf{p}} := (\mathbf{p}_0, \dots, \mathbf{p}_N)$  comprised of  $(N+1)$  volume vertices  $\mathbf{p}_0, \dots, \mathbf{p}_N \in \mathcal{B}_V$  (for some fixed  $N$ ):

$$\dot{I} = \int_{\hat{\Omega}_N} d_\theta \left( h(\bar{\mathbf{x}}, \theta) \prod_{n=0}^N J(\mathbf{p}_n, \theta) \right) d\mu(\bar{\mathbf{p}}), \quad (32)$$

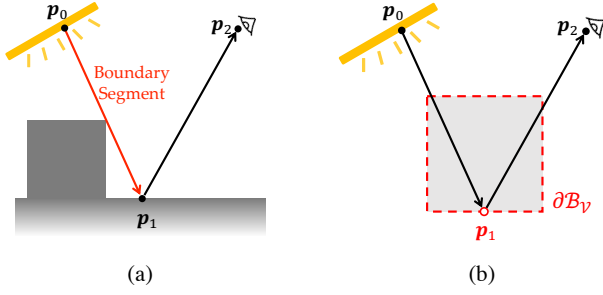
where  $\hat{\Omega}_N := \mathcal{B}_V^{N+1}$ ,  $\bar{\mathbf{x}} := (\mathbf{x}_0, \dots, \mathbf{x}_N)$  is the spatial description of the path  $\bar{\mathbf{p}}$  with  $\mathbf{x}_n = \mathbf{X}(\mathbf{p}_n, \theta)$ , and  $\mu$  is defined in Eq. (3).

Based on the derivations detailed in Appendix A, we show that the derivative  $\dot{I}$  in Eq.(32) can be expressed as

$$\dot{I} = \int_{\hat{\Omega}_N}^{\text{interior}} \partial_\theta h(\bar{\mathbf{p}}, \theta) d\mu(\bar{\mathbf{p}}) + \int_{\partial \hat{\Omega}_N}^{\text{boundary}} h(\bar{\mathbf{p}}) v_\perp(\mathbf{p}_N) d\mu_N(\bar{\mathbf{p}}), \quad (33)$$

where the boundary domain  $\partial \hat{\Omega}_N$  and the associated measure  $\mu_N$  are defined, respectively, in Eqs. (46) and (47) in Appendix A.

Similar to Eq. (31), the interior component on the right-hand side of Eq. (33) does not involve the motion  $\mathbf{X}$  or the velocity  $\mathbf{v}$ .



**Figure 4: Boundary path spaces:** (a) Zhang et al. [ZMY\*20] differential path integrals involve a boundary path space  $\Delta\hat{\Omega}$  comprised of paths with exactly one boundary segment. (b) Our volume boundary path space  $\partial\hat{\Omega}$  consists of paths each with exactly one volume boundary vertex ( $p_1$  in this example).

**Handling general differential path integrals.** Although Eq. (33) is defined for paths comprised of  $(N + 1)$  volume vertices only (i.e.,  $p_n \in \mathcal{B}_V$  for all  $n$ ), this result can be generalized easily to the general case of Eq. (15) where a path can also contain surface vertices (i.e.,  $p_n \in \mathcal{B}_M$  for some  $n$ ). Specifically, by letting  $h$  be the measurement contribution function  $f$  and pre-integrating all surface vertices, we obtain the main result of this section:

$$\int_{\hat{\Omega}} d_{\theta} \hat{f}(\bar{p}) d\mu(\bar{p}) = \int_{\hat{\Omega}} \overset{\text{interior}}{\partial_{\theta} f(\bar{p})} d\mu(\bar{p}) + \int_{\partial\hat{\Omega}} \overset{\text{boundary}}{f(\bar{p}) v_{\perp}(\mathbf{p}_K)} d\mu(\bar{p}), \quad (34)$$

where  $\partial\hat{\Omega}$  is the **volume boundary path space** comprised of **volume boundary paths**  $\bar{p} = (p_0, \dots, p_N)$  containing exactly one **volume boundary vertex**  $p_K$  residing on the boundary  $\partial\mathcal{B}_V$  of the reference volume for some  $0 < K < N$ . For paths with two or more vertices constrained on the boundary, they have zero measure and make no contribution to the first-order derivatives. Further, the differential measure  $\mu$  satisfies

$$d\mu(\bar{p}) = dA(\mathbf{p}_K) \prod_{n \neq K} \begin{cases} dV(\mathbf{p}_n), & (\mathbf{p}_n \in \mathcal{B}_V) \\ dA(\mathbf{p}_n). & (\mathbf{p}_n \in \mathcal{B}_M) \end{cases} \quad (35)$$

**Discussion.** In Eq. (34), the path contribution  $f(\bar{p})$  can vary with the parameter  $\theta$ , even with the path  $\bar{p}$  fixed. This is mainly because the transmittance  $T(\mathbf{p}_n \leftrightarrow \mathbf{p}_{n+1})$ —which is a factor of  $f$ —can depend on  $\theta$  when a segment  $\overline{\mathbf{p}_n \mathbf{p}_{n+1}}$  of the path crosses the (“null”) interface of some translucent object (as illustrated in Figure 3). Such dependency is captured by the interior component.

Additionally, the volume boundary path space  $\partial\hat{\Omega}$  is fundamentally different from the original boundary path space  $\Delta\hat{\Omega}$  in Eq. (13). As illustrated in Figure 4, the former contains paths with one vertex constrained on the boundary of some translucent object. The latter, on the other hand, consists of paths containing a pair of vertices constrained by a visibility boundary caused by some opaque surface.

**Continuity assumption.** Our derivation of the main result of Eq. (34) relies on the assumption (S.1) of having no reflective/refractive or (volume) shadow boundaries. This assumption ensures that material measurement contribution  $\hat{f}$  to be continuous in the interior of  $\mathcal{B}_V$ —which is required for applying the divergence theorem to obtain Eq. (22).

We note that our theory does not fundamentally rely on this assumption: When it is violated, one can make the volume boundary vertices to be constrained on the union of volume boundaries  $\partial\mathcal{B}_V$  as well as visibility (e.g., volume shadow) boundaries. However, since the “velocity” of visibility boundaries can be nontrivial to compute (without tetrahedralization), we consider the relaxation of Assumption S.1 an important topic for future research.

## 5. Monte Carlo Estimators

We now introduce new Monte Carlo methods to estimate Eq. (34). Since the interior component of Eq. (34) can be handled in similar fashion as that of the ordinary differential path integral of Eq. (13)—except for having the path vertices “detached”—we focus on the newly introduced boundary component:

$$i_{\text{bnd}} := \int_{\partial\hat{\Omega}} f(\bar{p}) v_{\perp}(\mathbf{p}_K) d\mu(\bar{p}). \quad (36)$$

By randomly sampling volume boundary paths  $\bar{p}$  (from the volume boundary path space  $\partial\hat{\Omega}$ ), we obtain the following (single-sample) Monte Carlo estimator of Eq. (36):

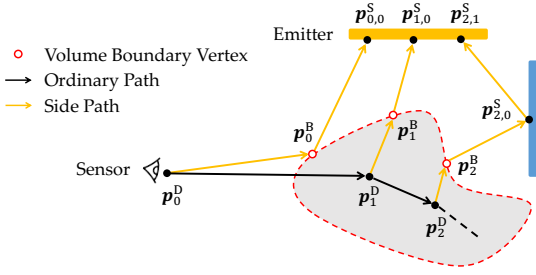
$$\langle i_{\text{bnd}} \rangle = f(\bar{p}) v_{\perp}(\mathbf{p}_K) / \text{pdf}(\bar{p}), \quad (37)$$

where  $\text{pdf}(\bar{p})$  denotes the probability density for sampling  $\bar{p}$ .

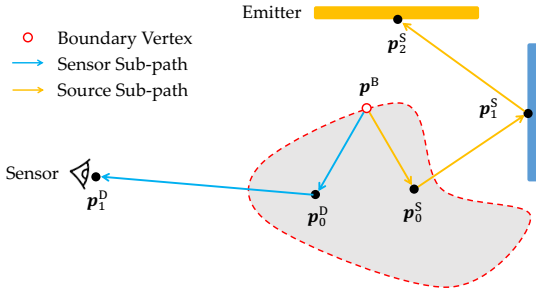
Based on this framework, we introduce a *unidirectional* and a *bidirectional* estimator—both of which support the full radiative transfer model [Cha60] including multiple scattering, anisotropic phase functions, and heterogeneity. Further, we combine these estimators using multiple importance sampling (MIS).

**Unidirectional estimator.** Starting with  $p_0^D$  on the detector, our unidirectional estimation first builds a “main” path  $(p_0^D, p_1^D, \dots)$  using (unidirectional) volume path tracing. Then, for each vertex  $p_n^D$ , the estimator construct a “side” path by explicitly drawing the volume boundary vertex  $p_n^B \in \partial\mathcal{B}_V$  (i.e., from the “null” boundary of a translucent object) and connecting this point to a light source via a subpath  $(p_{n,0}^S, \dots, p_{n,m_n}^S)$  for some  $m_n \geq 0$ . This gives a volume boundary path  $(p_{n,0}^S, \dots, p_{n,m_n}^S, p_n^B, p_n^D, \dots, p_0^D)$  where  $p_n^B$  corresponds to a volume scattering event happening exactly on the boundary (see Figure 5).

In practice, we sample the volume boundary vertex  $p_n^B$  by first drawing a direction  $\omega_n$  based on the BSDF or phase function at  $p_n^D$ . Then, we trace a ray from  $p_n^D$  in the direction  $\omega_n$  and let  $p_n^B$  be an intersection between this ray and all “null” interfaces  $\partial\mathcal{B}_V$ . Further, to avoid high computational overhead for generating “side” paths when the “main” one is long (i.e., with large numbers of vertices), we apply Russian roulette at each vertex  $p_n^D$  of the “main” path based on the throughput to stochastically skip the generation of “side” paths.



**Figure 5:** Our unidirectional sampling of volume boundary paths.



**Figure 6:** Our bidirectional sampling of volume boundary paths.

**Bidirectional estimator.** Additionally, we introduce a bidirectional estimator that samples volume boundary paths by first drawing the volume boundary vertex  $p^B$ . From this vertex, the algorithm then constructs two subpaths (using unidirectional sampling with next-event estimation) to connect  $p^B$  to the light source and the detector, respectively (see Figure 6). This process is conceptually similar to the multi-directional technique introduced by Zhang et al. [ZMY\*20] for sampling ordinary boundary paths.

**Multiple importance sampling.** Provided two volume boundary paths  $\bar{p}_1$  and  $\bar{p}_2$  sampled independently using our unidirectional and bidirectional estimators, respectively, we weight their contributions with (path-level) multiple importance sampling. Using the balanced heuristic [Vea97], we have

$$\langle j_{\text{bnd}} \rangle_{\text{mis}} = \frac{f(\bar{p}_1) v_{\perp}(\mathbf{p}_{K_1})}{\text{pdf}_1(\bar{p}_1) + \text{pdf}_2(\bar{p}_1)} + \frac{f(\bar{p}_2) v_{\perp}(\mathbf{p}_{K_2})}{\text{pdf}_1(\bar{p}_2) + \text{pdf}_2(\bar{p}_2)}, \quad (38)$$

where  $\text{pdf}_1$  and  $\text{pdf}_2$  denote the probabilities for sampling a path using our unidirectional and bidirectional methods, respectively.

In this equation, since the two volume boundary paths  $\bar{p}_1$  and  $\bar{p}_2$  can be sampled independently, they can have varying lengths and  $K_1$  does not have to equal  $K_2$ .

We note that, using this MIS estimator requires computing the probability densities  $\text{pdf}_1$  and  $\text{pdf}_2$ , which can be difficult when methods like delta tracking [WMHL65] are used to sample free-flight distances.

In this paper, we use the MIS estimator in Eq. (38) only for objects with homogeneous optical densities and the aforementioned bidirectional estimator for those with spatially varying albedo.

## 6. Results

We implement our technique (§4 and §5) on the CPU based on the system developed by Yu et al. [YZN\*22] that utilizes the Enzyme automatic differentiation framework [MC20].

### 6.1. Validation and Evaluation

We validate our technique in Figure 7 using the “Bunny 1” that contains a homogeneous translucent bunny under an area light. Derivatives (with respect to the horizontal position of the bunny) estimated by our method closely matches the reference given by finite differences (FD).

Additionally, in Figure 8, we evaluate our technique by comparing derivatives estimated with FD, our method, and Zhang et al.’s path-space method [ZYZZ1] (indicated as PSDR). This figure uses a “Bumpy ball” scene containing a homogeneous translucent bumpy ball under environmental illumination, and the derivatives are with respect to the horizontal position of the object. When the ball is rendered from a distance, our bidirectional method outperforms the unidirectional. When the ball is rendered close-up, our unidirectional estimator outperforms the bidirectional one, thanks to its pixel-level stratification. In both cases, our MIS estimator offers the best quality at equal time while PSDR suffers from high variance due to tetrahedralization and point-in-tetrahedron tests.

Lastly, we compare our technique with both PSDR and the differential radiative transfer method [ZWZ\*19] (indicated as “DTRT”) in Figure 9. This example includes a “Bunny 2” scene where a homogeneous translucent Bunny is lit by an area light with complex shape (as illustrated in “Configuration”). When estimating derivatives with respect to the horizontal position of the area light, both our method and PSDR benefit from the material-form parameterization (described in §3.2) and do not require detecting boundaries of the light. DTRT—which relies on the spherical integral formulation—requires explicit silhouette detection and, therefore, performs significantly worse than both PSDR and our method.

### 6.2. Inverse-Rendering Results

We further demonstrate the usefulness of our method using several synthetic inverse-rendering examples. Please refer to Table 1 for performance statistics and the supplemental material for animated versions of these results.

In Figure 10, we compare inverse-rendering performance between our method and PSDR. The “Bumpy ball” example in this figure uses the same translucent ball as Figure 8 under area lighting, and we optimize the shape of the ball. Additionally, the “Cube” example involves a heterogeneous object under area lighting, and we jointly optimize the shape of the object and its spatially varying albedo. We compute derivatives with respect to albedo using path-replay backpropagation [VSJ21]. For both examples, we adjust sample counts so that our method and the PSDR baseline takes approximately the same time per iteration. Using identical optimization settings (including initializations and learning rates), our method produces significantly cleaner gradients (at equal time), allowing the optimizations to converge much faster.

Lastly, since translucent objects with “null” interfaces can also

**Table 1:** Inverse-rendering configurations and performance statistics. In this table, “**Trg.**” indicates the number of target images; “**Bat.**” is the size of mini-batches (i.e., the number of images rendered per iteration); “**Param.**” denotes the number of scene parameters being optimized; “**Iter.**” is the number of iterations; and “**DR**” indicates the time spent (on an AMD Ryzen 3950X CPU) per iteration on differentiable rendering using our method.

Scene	Trg.	Bat.	Param.	Iter.	DR
PIG (Fig. 1)	40	5	599,733	200	38.06s
BUMPY BALL (Fig. 10)	20	2	149,685	400	9.53s
CUBE (Fig. 10)	20	2	29,952	100	5.81s
LETTER Y (Fig. 11)	20	1	15,084	160	11.0s
KITTY (Fig. 11)	20	4	300,123	500	5.30s

be represented using regular 3D volumes (with spatially varying densities), we compare in Figure 11 mesh-based inverse rendering using our method and volume-based inverse rendering [NDMKJ22]. This figure involves two examples, “Letter Y” and “Kitty”, each containing a homogeneous translucent object with known single-scattering albedo under environmental illumination. When using our method, we optimize the shapes of the object boundaries. When using Nimier-David et al. [NDMKJ22]’s, in contrast, we optimize per-voxel optical densities. Using equal sample, our method converges faster and produces overall cleaner reconstructions.

We note that, volume-based representations enjoy a few advantages over mesh-based ones—such as the flexibility for supporting topology changes. Figure 11 is to show practical advantages of our method (and mesh-based representations) when applied to solid objects with constant topology.

## 7. Discussion and Conclusion

**Limitations and future work.** Our technique is currently limited to translucent objects without refractive or reflective boundaries. Generalizing it to support these interfaces is an important topic for future research. Additionally, our MIS estimator is only applicable to translucent objects with homogeneous optical densities. Extending it to support heterogeneous densities by leveraging, for instance, the null-scattering formulation [MGJ19] is worth exploring.

**Conclusion.** We devised a new formulation for differentiable rendering of translucent objects with respect to their shapes. By reparameterizing interior components of differential path integrals as a new class of boundary path integrals, our formulation does not require change rates to be specified in the interior of objects. Additionally, we developed new Monte Carlo estimators for the newly introduced boundary path integral.

We validated our technique by comparing derivatives estimated with finite-difference (FD) and ours methods. Further, we demonstrated the effectiveness of our method using a few differentiable rendering and inverse rendering results.

## References

- [BLD20] BANGARU S., LI T.-M., DURAND F.: Unbiased warped-area sampling for differentiable rendering. *ACM Trans. Graph.* 39, 6 (2020), 245:1–245:18. 4
- [Cha60] CHANDRASEKHAR S.: *Radiative transfer*. Courier Corporation, 1960. 1, 2, 6
- [CLZ\*20] CHE C., LUAN F., ZHAO S., BALA K., GKIOULEKAS I.: Towards learning-based inverse subsurface scattering. In *2020 IEEE International Conference on Computational Photography (ICCP)* (2020), IEEE, pp. 1–12. 2
- [CPP\*05] CEREZO E., PÉREZ F., PUEYO X., SERON F. J., SILLION F. X.: A survey on participating media rendering techniques. *The Visual Computer* 21, 5 (2005), 303–328. 2
- [DLW\*22] DENG X., LUAN F., WALTER B., BALA K., MARSCHNER S.: Reconstructing translucent objects using differentiable rendering. In *ACM SIGGRAPH 2022 Conference Proceedings* (2022), pp. 38:1–38:10. 2
- [Gur81] GURTIN M. E.: *An introduction to continuum mechanics*. Academic press, 1981. 5
- [GZB\*13] GKIOULEKAS I., ZHAO S., BALA K., ZICKLER T., LEVIN A.: Inverse volume rendering with material dictionaries. *ACM Trans. Graph.* 32, 6 (2013), 162:1–162:13. 2
- [JM12] JAKOB W., MARSCHNER S.: Manifold exploration: A markov chain monte carlo technique for rendering scenes with difficult specular transport. *ACM Trans. Graph.* 31, 4 (2012), 58:1–58:13. 2
- [JSRV22] JAKOB W., SPEIERER S., ROUSSEL N., VICINI D.: Dr.jit: A just-in-time compiler for differentiable rendering. *ACM Trans. Graph.* 41, 4 (2022), 124:1–124:19. 2
- [KSKAC02] KELEMEN C., SZIRMAI-KALOS L., ANTAL G., CSONKA F.: A simple and robust mutation strategy for the metropolis light transport algorithm. In *Computer Graphics Forum* (2002), vol. 21, Wiley Online Library, pp. 531–540. 2
- [KVH84] KAJIYA J. T., VON HERZEN B. P.: Ray tracing volume densities. *SIGGRAPH Comput. Graph.* 18, 3 (1984), 165–174. 2
- [LADL18] LI T.-M., AITTALA M., DURAND F., LEHTINEN J.: Differentiable monte carlo ray tracing through edge sampling. *ACM Trans. Graph.* 37, 6 (2018), 222:1–222:11. 2
- [LW96] LAFORTUNE E. P., WILLEMS Y. D.: *Rendering participating media with bidirectional path tracing*. In *Rendering techniques’ 96*. Springer, 1996, pp. 91–100. 2
- [MC20] MOSES W., CHURAVY V.: Instead of rewriting foreign code for machine learning, automatically synthesize fast gradients. In *Advances in Neural Information Processing Systems* (2020), Larochelle H., Ranzato M., Hadsell R., Balcan M. F., Lin H., (Eds.), vol. 33, Curran Associates, Inc., pp. 12472–12485. 7
- [MGJ19] MILLER B., GEORGIEV I., JAROSZ W.: A null-scattering path integral formulation of light transport. *ACM Trans. Graph.* 38, 4 (2019), 44:1–44:13. 8
- [NDMKJ22] NIMIER-DAVID M., MÜLLER T., KELLER A., JAKOB W.: Unbiased inverse volume rendering with differential trackers. *ACM Trans. Graph.* 41, 4 (2022), 44:1–44:20. 2, 8, 11
- [NDSRJ20] NIMIER-DAVID M., SPEIERER S., RUIZ B., JAKOB W.: Radiative backpropagation: an adjoint method for lightning-fast differentiable rendering. *ACM Trans. Graph.* 39, 4 (2020), 146:1–146:15. 2
- [NDVZJ19] NIMIER-DAVID M., VICINI D., ZELTNER T., JAKOB W.: Mitsuba 2: A retargetable forward and inverse renderer. *ACM Trans. Graph.* 38, 6 (2019), 203:1–203:17. 2
- [NGHJ18] NOVÁK J., GEORGIEV I., HANIKA J., JAROSZ W.: Monte Carlo methods for volumetric light transport simulation. In *Computer Graphics Forum* (2018), vol. 37, Wiley Online Library, pp. 551–576. 2
- [PKK00] PAULY M., KOLLIG T., KELLER A.: *Metropolis light transport for participating media*. In *Rendering Techniques 2000*. Springer, 2000, pp. 11–22. 1, 2



- [Rey03] REYNOLDS O.: *Papers on mechanical and physical subjects: the sub-mechanics of the universe*, vol. 3. The University Press, 1903. 5
- [SG69] SPANIER J., GELBARD E. M.: *Monte Carlo principles and neutron transport problems*. The Addison-Wesley Publishing Company, 1969. 2
- [SRB\*19] SUMIN D., RITTIG T., BABAEI V., NINDEL T., WILKIE A., DIDYK P., BICKEL B., KRIVÁNEK J., MYSZKOWSKI K., WEYRICH T.: Geometry-aware scattering compensation for 3d printing. *ACM Trans. Graph.* 38, 4 (2019), 111:1–111:14. 2
- [Vea97] VEACH E.: *Robust Monte Carlo methods for light transport simulation*, vol. 1610. Stanford University PhD thesis, 1997. 2, 7
- [VG95] VEACH E., GUIBAS L.: Bidirectional estimators for light transport. In *Photorealistic Rendering Techniques*. Springer, 1995, pp. 145–167. 2
- [VG97] VEACH E., GUIBAS L. J.: Metropolis light transport. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques* (1997), SIGGRAPH '97, ACM Press/Addison-Wesley Publishing Co., p. 65–76. 2
- [VJK21] VICINI D., JAKOB W., KAPLANYAN A.: A non-exponential transmittance model for volumetric scene representations. *Transactions on Graphics (Proceedings of SIGGRAPH)* 40, 4 (Aug. 2021), 136:1–136:16. 2
- [VSI21] VICINI D., SPEIERER S., JAKOB W.: Path replay backpropagation: Differentiating light paths using constant memory and linear time. *ACM Trans. Graph.* 40, 4 (2021). 2, 4, 7
- [WMHL65] WOODCOCK E., MURPHY T., HEMMINGS P., LONGWORTH S.: Techniques used in the gem code for monte carlo neutronics calculations in reactors and other systems of complex geometry. In *Proc. Conf. Applications of Computing Methods to Reactor Problems* (1965), vol. 557. 7
- [YZN\*22] YU Z., ZHANG C., NOWROUZEZAHRAI D., DONG Z., ZHAO S.: Efficient differentiation of pixel reconstruction filters for path-space differentiable rendering. *ACM Trans. Graph.* 41, 6 (2022), 191:1–191:16. 7
- [ZMY\*20] ZHANG C., MILLER B., YAN K., GKIOULEKAS I., ZHAO S.: Path-space differentiable rendering. *ACM Trans. Graph.* 39, 4 (2020), 143:1–143:19. 2, 3, 6, 7
- [ZWDR16] ZHAO S., WU L., DURAND F., RAMAMOORTHI R.: Down-sampling scattering parameters for rendering anisotropic media. *ACM Trans. Graph.* 35, 6 (2016), 166:1–166:11. 2
- [ZW\*19] ZHANG C., WU L., ZHENG C., GKIOULEKAS I., RAMAMOORTHI R., ZHAO S.: A differential theory of radiative transfer. *ACM Trans. Graph.* 38, 6 (2019), 227:1–227:16. 1, 2, 7, 10
- [ZY21] ZHANG C., YU Z., ZHAO S.: Path-space differentiable rendering of participating media. *ACM Trans. Graph.* 40, 4 (2021), 76:1–76:15. 1, 2, 3, 4, 7, 10, 11

## Appendix A: Derivation of Eq. (33)

We now show how Eq. (33) can be derived from Eq. (32). When the function  $h$  does not directly depend on  $\theta$ , according to the chain rule, it holds that

$$d_{\theta} h(\bar{\mathbf{x}}) = \sum_{n=0}^N \underbrace{\left[ \frac{\partial h}{\partial \mathbf{x}_n} \right]_{\mathbf{x}_n = \mathbf{x}(\mathbf{p}_n, 0)}}_{=:(\nabla_n h)(\bar{\mathbf{p}})} \cdot \mathbf{v}(\mathbf{p}_n). \quad (39)$$

Further, according to the product rule, it holds that

$$\partial_{\theta} \prod_{n=0}^N J(\mathbf{p}_n, \theta) = \sum_{n=0}^N \partial_{\theta} J(\mathbf{p}_n, \theta) \underbrace{\prod_{m \neq n} J(\mathbf{p}_m, 0)}_{=1} = \sum_{n=0}^N \partial_{\theta} J(\mathbf{p}_n, \theta). \quad (40)$$

Provided Eqs. (39) and (40), we have

$$d_{\theta} \left( h(\bar{\mathbf{x}}) \prod_{n=0}^N J(\mathbf{p}_n, \theta) \right) = \sum_{n=0}^N ((\nabla_n h)(\bar{\mathbf{p}}) \cdot \mathbf{v}(\mathbf{p}_n) + h(\bar{\mathbf{p}}) \partial_{\theta} J(\mathbf{p}_n, \theta)). \quad (41)$$

According to Eqs. (18, 26, 28, 29), it holds that

$$\int_{\mathcal{B}_{\mathcal{V}}} ((\nabla_n h)(\bar{\mathbf{p}}) \cdot \mathbf{v}(\mathbf{p}_n) + h(\bar{\mathbf{p}}) \partial_{\theta} J(\mathbf{p}_n, \theta)) dV(\mathbf{p}_n) = \int_{\partial \mathcal{B}_{\mathcal{V}}} h(\bar{\mathbf{p}}) v_{\perp}(\mathbf{p}_n) dA(\mathbf{p}_n). \quad (42)$$

Given Eqs. (41) and (42), the interior integral in Eq. (32) can be rewritten as

$$\dot{I} = \sum_{K=0}^N \int_{\partial \hat{\Omega}_{N,K}} h(\bar{\mathbf{p}}) v_{\perp}(\mathbf{p}_K) d\hat{\mu}_{N,K}(\bar{\mathbf{p}}), \quad (43)$$

where  $\partial \hat{\Omega}_{N,K} := \mathcal{B}_{\mathcal{V}}^K \times \partial \mathcal{B}_{\mathcal{V}} \times \mathcal{B}_{\mathcal{V}}^{N-K}$ , and

$$d\hat{\mu}_{N,K}(\bar{\mathbf{p}}) := dA(\mathbf{p}_K) \prod_{n \neq K} dV(\mathbf{p}_n). \quad (44)$$

Notationally, Eq. (43) can be further simplified as a single boundary path integral of the form

$$\dot{I} = \int_{\partial \hat{\Omega}_N} h(\bar{\mathbf{p}}) v_{\perp}(\mathbf{p}_K) d\hat{\mu}_N(\bar{\mathbf{p}}), \quad (45)$$

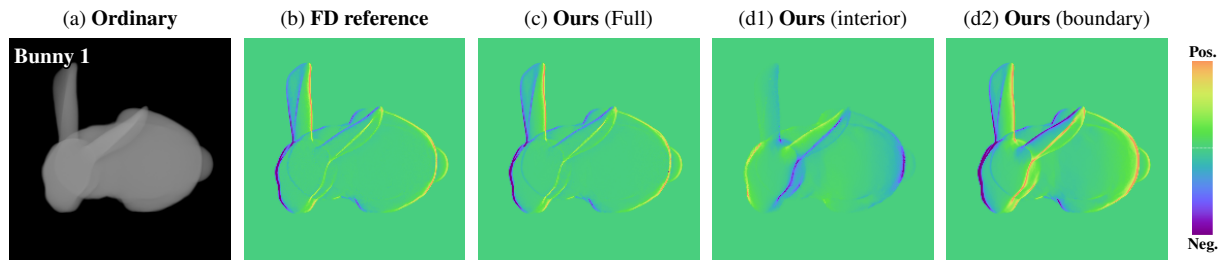
where

$$\partial \hat{\Omega}_N := \bigcup_{K=0}^N \partial \hat{\Omega}_{N,K}, \quad (46)$$

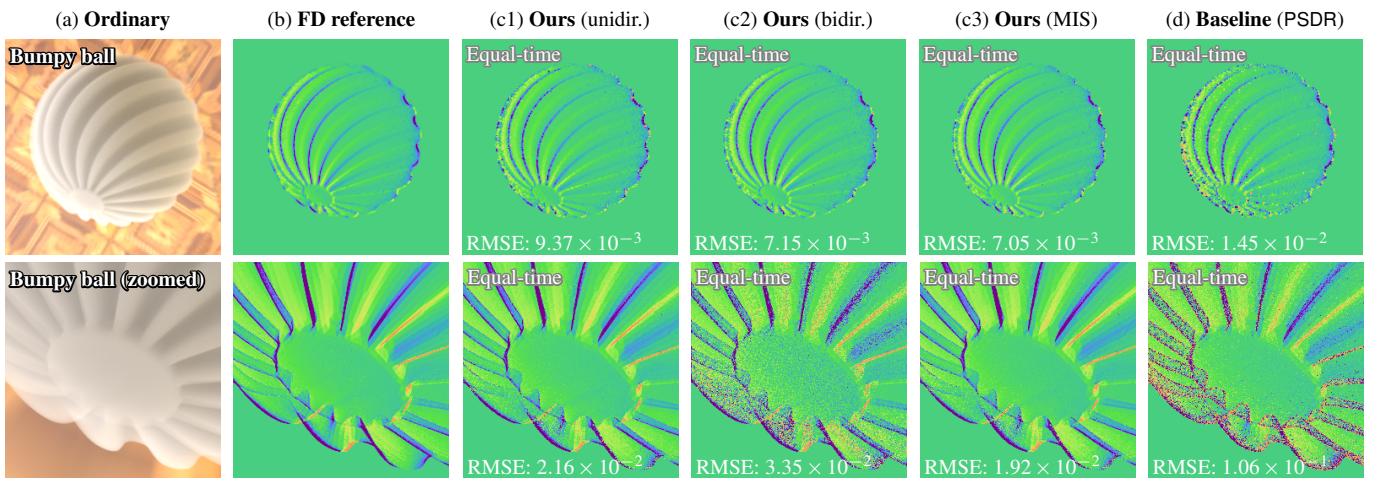
and, for any  $\mathcal{P} \subset \partial \hat{\Omega}$ ,

$$\hat{\mu}_N(\mathcal{P}) := \sum_{K=0}^N \hat{\mu}_{N,K}(\partial \hat{\Omega}_{N,K} \cap \mathcal{P}), \quad (47)$$

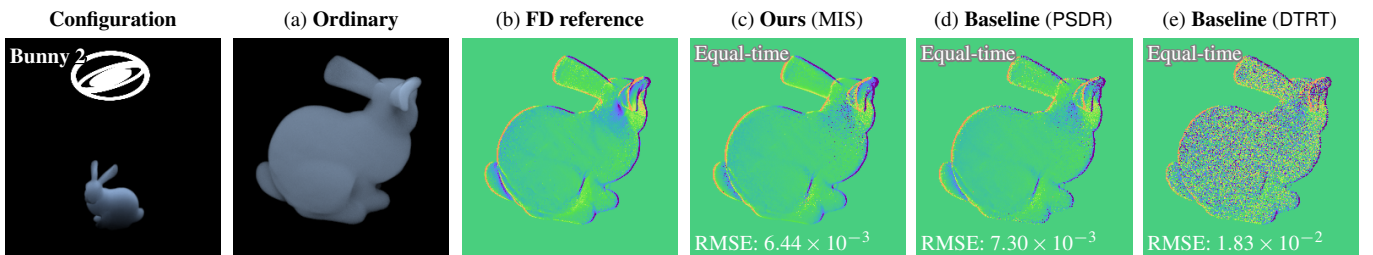
with  $\hat{\mu}_{N,K}$  defined in Eq. (44). When the integrand  $h(\bar{\mathbf{x}}, \theta)$  directly depends on the parameter  $\theta$ , similar to Eq. (31), one extra interior integral of the partial derivative  $\partial_{\theta} h$  needs to be introduced, yielding Eq. (33).



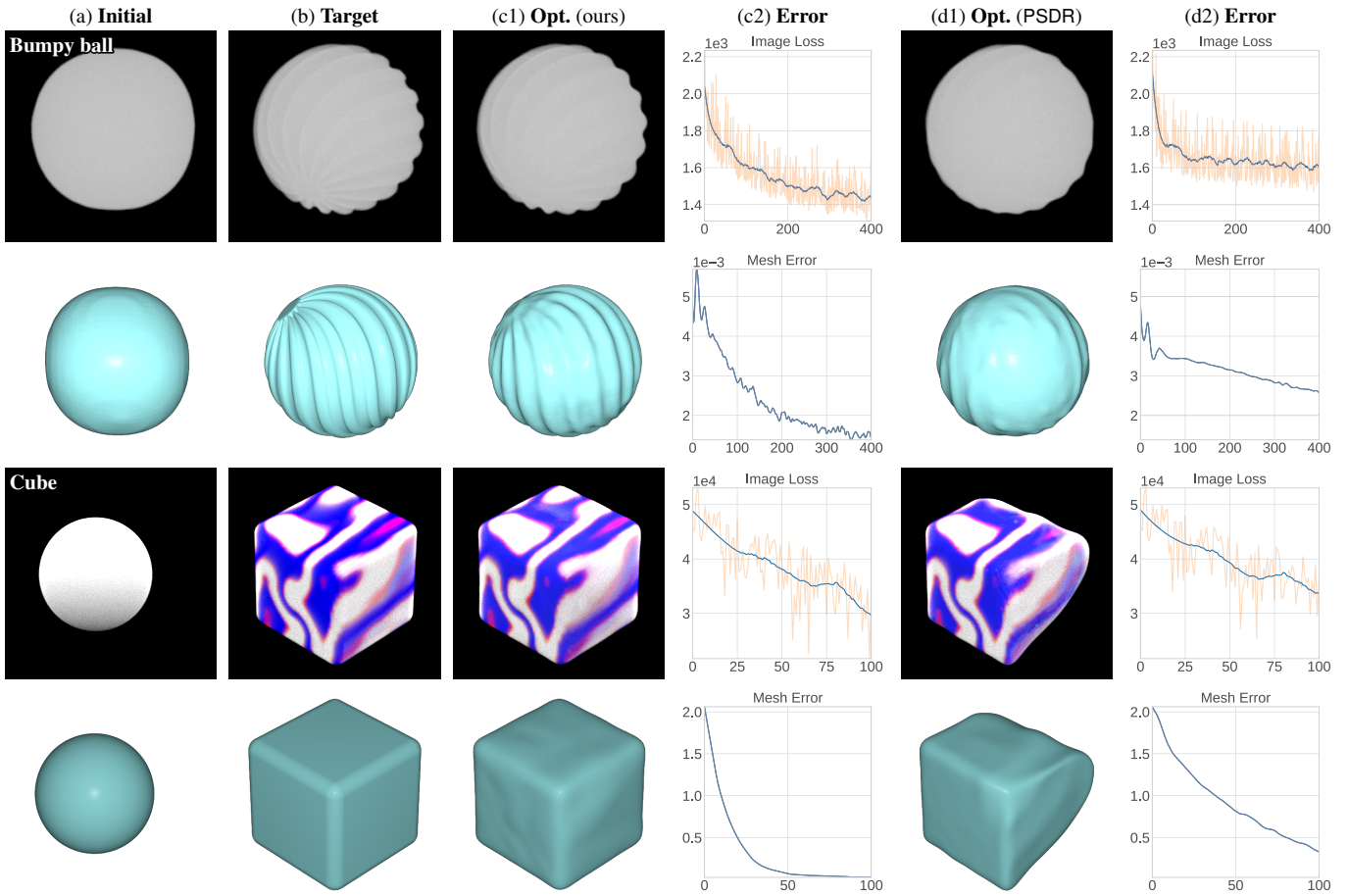
**Figure 7:** We *validate* our technique by comparing derivatives obtained using finite differences (b) and our method (c). Additionally, we visualize our interior (d1) and boundary (d2) components given by the corresponding terms in Eq. (34). Our full estimate (c) equals the sum of these two components.



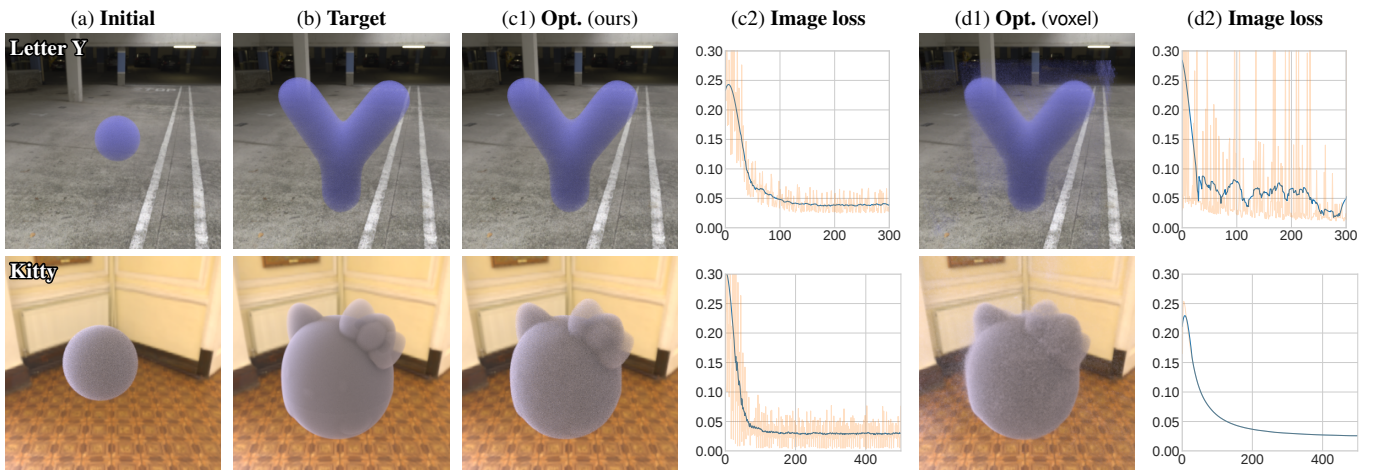
**Figure 8: Validation & evaluation:** We compare derivative estimates (visualized using the same color map as Figure 7) generated with our unidirectional (c1), bidirectional (c2), and MIS (c3) estimators to those obtained using finite differences (b) as well as Zhang et al.'s PSDR [ZYZ21] (d).



**Figure 9: Validation & evaluation:** We compare derivative estimates (visualized using the same color map as Figure 7) generated with our MIS estimator (c) to those obtained using finite differences (b), PSDR [ZYZ21] (d), and DTRT [ZWZ\*19] (e).



**Figure 10:** *Inverse-rendering* comparison between our method and PSDR [ZYZZ1]. For each example, we show images used for optimization on the top and shape visualizations on the bottom. The “Mesh Error” information is only used for evaluation (and not for optimization). We use 20 target images for both examples (with one shown).



**Figure 11:** *Inverse-rendering* comparison between our method (using mesh-based geometric representations) and Nimier-David et al.’s approach [NDMKJ22] (using voxel-based representations indicated as “voxel”). We use 20 target images for both examples (with one shown).