

# Differentiable Photon Mapping using Generalized Path Gradients

JIANKAI XING, Key Laboratory of Pervasive Computing, Ministry of Education, Department of Computer Science and Technology, Tsinghua University, China

ZENGYU LI, Key Laboratory of Pervasive Computing, Ministry of Education, Department of Computer Science and Technology, Tsinghua University, China

FUJUN LUAN, Adobe Research, USA

KUN XU\*, Key Laboratory of Pervasive Computing, Ministry of Education, Department of Computer Science and Technology, Tsinghua University, China

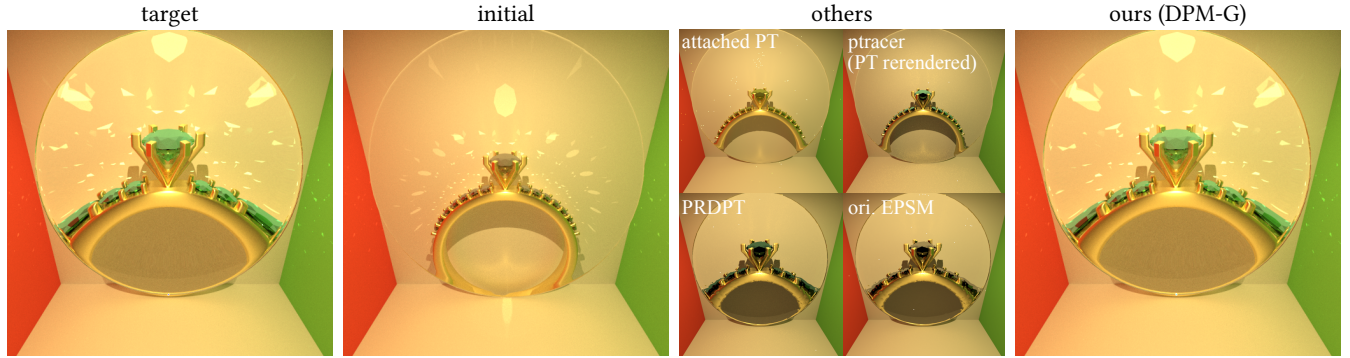


Fig. 1. Visual comparisons between our differentiable photon mapping method with several state-of-the-art physics-based differentiable rendering methods. The scene contains a ring behind a magnifying glass within the Cornell box. The optimizing parameters include the color of the diamond on the ring and the thickness of the glass. Initially, the magnifying glass is very thin and the goal of optimization is to thicken the magnifying glass and to modify the color of the diamond to match the magnified view of the ring in the target image. The comparison baseline methods include attached PT (a reimplemented simplified version of attached path replay backpropagation [Vicini et al. 2021]), ptracer [Nimier-David et al. 2019], Plateau-reduced Differentiable Path Tracing (PRDPT) [Fischer and Ritschel 2022], and the original extended path space manifolds based method (ori. EPSM) [Xing et al. 2023]. For better display quality, the displayed visual results of each method shown in the figure are re-rendered with 8192 spps using that method, except that path tracing is used for re-rendering the displayed visual results of ptracer (marked as 'PT rerendered') since ptracer is not able to sample the refraction paths in this scene.

Photon mapping is a fundamental and practical Monte Carlo rendering technique for efficiently simulating global illumination effects, especially for caustics and specular-diffuse-specular (SDS) paths. In this paper, we present the first differentiable rendering method for photon mapping. The core of our method is a newly introduced concept named *generalized path gradients*. Based on the extended path space manifolds (EPSMs) [Xing et al. 2023], the generalized path gradients define the derivatives of the vertex positions and color contributions of a path with respect to scene parameters under given geometric constraints. By formalizing photon mapping as a path sampling technique through vertex merging [Georgiev et al. 2012] and incorporating a smooth differentiable density estimation kernel, we enable the differentiation of the photon mapping algorithms based on the theoretical results of generalized path gradients.

Experiments demonstrate that our method is more effective than state-of-the-art physics-based differentiable rendering methods in inverse rendering applications involving difficult illumination paths, especially SDS paths.

\*Kun Xu is the corresponding author.

Authors' addresses: Jiankai Xing, xjk21@mails.tsinghua.edu.cn, Key Laboratory of Pervasive Computing, Ministry of Education, Department of Computer Science and Technology, Tsinghua University, Beijing, China; Zengyu Li, lizengyu21@mails.tsinghua.edu.cn, Key Laboratory of Pervasive Computing, Ministry of Education, Department of Computer Science and Technology, Tsinghua University, Beijing, China; Fujun Luan, fluan@adobe.com, Adobe Research, USA; Kun Xu, xukun@tsinghua.edu.cn, Key Laboratory of Pervasive Computing, Ministry of Education, Department of Computer Science and Technology, Tsinghua University, Beijing, China.

CCS Concepts: • **Computing methodologies** → **Ray tracing**.

Additional Key Words and Phrases: differentiable rendering, photon mapping

## ACM Reference Format:

Jiankai Xing, Zengyu Li, Fujun Luan, and Kun Xu. 2024. Differentiable Photon Mapping using Generalized Path Gradients. *ACM Trans. Graph.* 43, 6, Article 257 (December 2024), 14 pages. <https://doi.org/10.1145/3687958>

## 1 INTRODUCTION

Photorealistic rendering has been a central research field in computer graphics for decades, aiming to simulate physical light transport in complex scenes and faithfully reproduce realistic images. To solve the rendering equation [Kajiya 1986], various Monte Carlo global illumination algorithms have been proposed, such as path tracing, bidirectional path tracing, and photon mapping [Jensen 1996]. Unlike path tracing, photon mapping is known to be particularly effective at complex light transport effects such as caustics and specular-diffuse-specular (SDS) paths [Hachisuka et al. 2008].

In recent years, physics-based differentiable rendering has emerged as a powerful tool and attracted more attention in the graphics research community, enabling the optimization of scene parameters (geometry, materials, lighting) by differentiating the rendering process. This capability is crucial for inverse rendering applications,

where the goal is to reconstruct 3D scene properties from 2D images. Differentiable rendering has been successfully applied to various algorithms, such as differentiable path tracing [Li et al. 2018], enabling high-quality reconstruction of scene attributes.

However, existing differentiable rendering methods primarily focus on path tracing and its variants. While effective for many scenarios, these methods struggle with complex light transport effects that the photon mapping technique handles efficiently, such as the aforementioned caustics and SDS paths. The absence of a differentiable photon mapping algorithm has prevented this technique from being applied in scenes where these complex lighting effects are important (e.g., computational caustics design).

Motivated by this gap, we propose a newly introduced concept, *generalized path gradients*, which builds on recent work extended path space manifolds (EPSMs) [Xing et al. 2023]. This concept defines the derivatives of the vertex positions and color contributions of a path with respect to scene parameters under certain geometric constraints (e.g., half-vector). By formalizing photon mapping as a path sampling technique [Georgiev et al. 2012; Hachisuka et al. 2012] and incorporating a smooth differentiable density estimation kernel, we enable the differentiation of the photon mapping algorithms based on the theoretical results of generalized path gradients. Our approach bridges the gap between differentiable rendering and photon mapping, allowing for the efficient and accurate handling of complex illumination paths. Our experiments demonstrate that our method outperforms the state-of-the-art physics-based differentiable rendering methods in various scenarios, especially the ones with SDS paths.

Concretely, our contributions include:

- We propose *generalized path gradients*, a novel concept that defines both contribution derivatives and geometric derivatives of a path under given geometric constraints. We also show that our generalized path gradients offer a unified theoretical framework that many existing gradient formulations can be viewed as specialized cases of our formulation;
- We develop a *differentiable photon mapping* algorithm based on the theoretical results of generalized path gradients, which are demonstrated to be effective and robust in inverse rendering tasks involving complex illumination effects through experiments.

The code of our method is available at <https://www.github.com/jkxing/DPMG>.

## 2 RELATED WORK

### 2.1 Photon Mapping

Photon mapping (PM) [Jensen 1996] is a two-pass rendering algorithm that has emerged as a fundamental and practical technique in computer graphics for efficiently simulating the global illumination effects, especially caustics and specular-diffuse-specular (SDS) paths [Hachisuka et al. 2008]. In the first pass, a photon map is built by tracing a number of photon paths as they interact with surfaces in a scene, and the second pass traces camera rays and queries the photon map to render the final image through kernel density estimation.

Progressive photon mapping (PPM) [Hachisuka et al. 2008] provides a memory-friendly improvement over photon mapping by switching to a progressive refinement of photon statistics using multiple photon tracing passes, which combines the result from several photon maps that ensure consistent convergence to the correct radiance value (i.e., bias goes to zero in the limit). Stochastic progressive photon mapping (SPPM) [Hachisuka and Jensen 2009] further extends the PPM algorithm and generates new gathering points by the distributed ray tracing pass to render effects such as glossy reflections, depth of field, or motion blur. Most recently, more advanced estimators in photon density estimation framework were developed such as photon beams [Jarosz et al. 2011, 2008], planes and volumes [Bitterli and Jarosz 2017; Deng et al. 2019], offering efficient algorithms for simulating complex lighting transport effects. Besides forward rendering, there is work discussing the derivatives of photon mapping [Hachisuka et al. 2010], which is focused on the error estimation framework. In contrast, we differentiate the photon mapping algorithm to enable computing gradients of rendering output with respect to arbitrary scene parameters and use it for inverse rendering tasks.

Vertex connection and merging (VCM) [Georgiev et al. 2012] and unified path sampling (UPS) [Hachisuka et al. 2012] concurrently proposed a novel reformulation of photon mapping that is compatible with the path integral formulation [Veach 1997] of light transport. Specifically, VCM reconsiders photon mapping as a *vertex merging* technique, extending the standard PM sample to be a conditionally accepted regular path, while UPS unified Monte Carlo path integration and photon density estimation into a general formulation. In this work, we also employ their formulation, facilitating the differentiation of photon mapping in the path space formulation.

### 2.2 Differentiable Rendering

Recent advancements in physics-based differentiable rendering have focused on enhancing the capabilities of path tracers to handle global illumination effects through light transport simulation [Bangaru et al. 2020; Li et al. 2018; Loubet et al. 2019; Vicini et al. 2021, 2022; Zeltner et al. 2021; Zhang et al. 2020, 2021]. The differentiation process mainly involves estimating two main components: interior integrals derived from differentiating the integrands associated with forward-rendering models, and boundary integrals determined over the discontinuities present in those integrands. Reparameterization techniques have been explored to avoid computing boundary integrals [Bangaru et al. 2020; Loubet et al. 2019; Xu et al. 2023], and recent studies have investigated the influence of different sampling strategies on Monte Carlo estimations [Vicini et al. 2021; Zeltner et al. 2021].

Contemporary differentiable rendering methods often face limitations in inverse rendering tasks, particularly in aiding global and long-range optimization. While some techniques, such as those by Xing et al. [2022], have explored screen-space derivatives for assessing pixel color and position changes with scene parameters, they may not handle complex global illumination effects effectively. Fischer and Ritschel [2022] proposed a method that convolves the

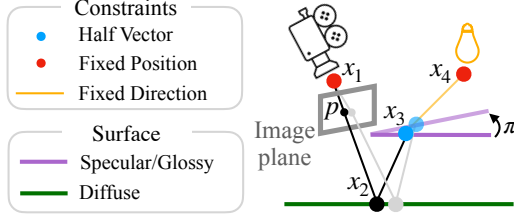


Fig. 2. An example of EPSM. When we perturb scene parameters  $\pi$ , path geometry will change under the constraint of the EPSM. Note that the perturbation of parameters should actually be infinitesimal since we can only ensure the geometric derivatives of path exist in its neighborhood.

rendering function with a kernel to capture long-range relationships, but its effectiveness diminishes with an increasing number of scene parameters.

Recently, Xing et al. [2023] introduced extended path space manifolds (EPSMs), defined in the combined space of path vertices and scene parameters. By enforcing geometric constraints, EPSMs enable tracking specific illumination effects and corresponding paths after perturbations in scene parameters. This approach provides path geometric derivatives that measure how path vertices change with respect to scene parameters under constraints. The effectiveness and robustness of this method have been demonstrated in inverse rendering applications involving complex illumination effects, such as specular reflections and caustics. However, they do not support derivatives of path contribution and cannot be used to optimize color related scene parameters like albedo map.

### 2.3 Path Similarity in Forward Rendering

Efficient sampling of paths is a broadly invested topic in forward rendering. An important problem in the field is to find a path similar to an existing path. Shift mapping [Kettunen et al. 2015] links two paths originating from adjacent pixels for variance reduction in gradient domain path tracing or ReSTIR-based methods [Lin\* et al. 2022]. Besides, for efficient path sampling, path space manifolds [Jakob and Marschner 2012] are proposed to sample specular chain light paths when endpoints are perturbed. Given an existing path, they build mathematical constraints to compute the derivatives of middle path vertices with respect to the positions of endpoints, and the derivatives are then used to adjust the paths. Note that these methods use shift mapping or derivatives only for forward rendering.

## 3 BACKGROUND

### 3.1 Extended Path Space Manifolds

Extended path space manifolds (EPSMs) [Xing et al. 2023] are introduced to handle difficult light paths especially specular paths in differentiable path tracing. They can be used to define how path geometry will change when we infinitesimally perturb scene parameters. An illustration of one EPSM is given in Fig. 2. Next, we will briefly review the definition of EPSMs, the derivation of the geometric derivatives, and how to construct EPSMs.

**Definition.** Considering a path with  $k$  vertices  $\bar{x} = [x_1, x_2, \dots, x_k]$  starting from the eye position  $x_1$  and ending at a point  $x_k$  on the light source, and scene parameters of interest  $\pi = [\pi_1, \pi_2, \dots, \pi_m]$ , the EPSM is defined as an implicit mapping from the scene parameters  $\pi$  to the paths  $\bar{x}$  by introducing a stack of 2D valued constraint functions:

$$C(\bar{x}, \pi) = [c_1(\bar{x}, \pi), \dots, c_k(\bar{x}, \pi)] = \mathbf{0}. \quad (1)$$

**Geometric derivatives.** The *Implicit Function Theorem* ensures that there exists a function  $\bar{x} = \bar{x}(\pi)$  within the neighborhood of an existing path if the Jacobian matrix of  $C$  with respect to  $\bar{x}$  is invertible. The derivatives of  $\bar{x}$  with respect to scene parameters  $\pi$  could be computed through implicit partial differentiation of the constraint functions:

$$\frac{\partial \bar{x}}{\partial \pi} = - \left( \frac{\partial C}{\partial \bar{x}} \right)^{-1} \cdot \frac{\partial C}{\partial \pi}. \quad (2)$$

We refer to the above derivatives as *geometric derivatives*.

**Types of EPSMs and constraint functions.** In theory, we could define an EPSM and compute the derivatives as long as the Jacobian matrix  $\partial C / \partial \bar{x}$  in Eq 2 is invertible, which requires us to carefully design the constraint functions and set the number of constraint functions equal to the number of path vertices ( $k$ ). Xing et al. [2023] have introduced 3 types of EPSMs for effectively handling difficult light paths, including general EPSMs for reflection, refraction, and highlights, caustics EPSMs for caustics effects, and shadow EPSMs for shadows in direct illumination, respectively. Each type of EPSM uses a combination of differentiable constraint functions, including:

- (1) the *half-vector constraint* enforces the half direction of incoming and outgoing ray directions of a specific vertex  $x_i$  to be unchanged in the local frame (see the blue dot in Fig. 2):

$$T_\pi \left( h(\overrightarrow{x_i x_{i-1}}, \overrightarrow{x_i x_{i+1}}, \pi) \right) = \mathbf{const}, \quad (3)$$

where  $h(\cdot)$  is the unit half-vector and  $T_\pi(\cdot)$  transforms the vector from global frame to local frame;

- (2) the *fixed position constraint* enforces a vertex position  $x_i$  to be locally fixed (see the red dot in Fig. 2):

$$\mathbf{w}(x_i, \pi) = \mathbf{const}, \quad (4)$$

where  $\mathbf{w}(\cdot)$  computes the local coordinates of vertex on the surface, i.e., barycentric coordinates of  $x_i$  to its belonging triangle in mesh-based representation;

- (3) the *fixed direction constraint* enforces a ray direction  $\overrightarrow{x_i x_{i+1}}$  to be locally unchanged (see the orange line in Fig. 2):

$$T_\pi \left( \overrightarrow{x_i x_{i+1}} \right) = \mathbf{const}. \quad (5)$$

- (4) the *colinear constraint* enforces two ray directions to be always colinear.

The first 3 constraints are used in our method while the last is not. More details can be found in the original paper [Xing et al. 2023].

### 3.2 EPSM for Inverse Rendering

One limitation of using differentiable rendering method in solving inverse rendering tasks is it requires a good initialization. To address this problem, Xing et al. [2022] proposed a novel loss function by employing approximated optimal transport [Cuturi 2013; Feydy

et al. 2019] to find a one-on-one dense matching between pixels in the rendered image and the target image. It further proposed a method to compute derivatives of screen space pixel position with respect to scene parameters, but it could be only used for computing derivatives of simple light paths in differentiable rasterization.

To extend the method to handle complex light paths, Xing et al. [2023] proposed *geometric derivatives* and a path-pixel matching based loss function. In path tracing framework, the 2D screen space intersection point  $\mathbf{p}$  between the eye ray  $\overrightarrow{\mathbf{x}_1\mathbf{x}_2}$  and the image plane is recorded:

$$\mathbf{p} = P(\overline{\mathbf{x}}, \pi) = P(\overrightarrow{\mathbf{x}_1\mathbf{x}_2}, \pi), \quad (6)$$

where  $P$  denotes the function of ray-plane intersection. Its derivatives can be directly computed from the geometric derivatives through the chain rule:

$$\frac{\partial \mathbf{p}}{\partial \pi} = \frac{\partial P}{\partial \overline{\mathbf{x}}} \cdot \frac{\partial \overline{\mathbf{x}}}{\partial \pi} + \frac{\partial P}{\partial \pi}. \quad (7)$$

We refer to it as the *screen-space positional derivatives*.

For loss function, Xing et al. [2023] proposed a path-pixel matching adapted from the pixel-pixel matching method from Xing et al. [2022]. The matching provides a target pixel position for each intersection point  $\mathbf{p}$ . Then, the loss function is defined as the L2 distance between the screen space intersection point  $\mathbf{p}$  and the position of the matched pixel in the target image  $\mathbf{p}_{\text{target}}$  and its derivatives are:

$$L(\pi) = (\mathbf{p} - \mathbf{p}_{\text{target}})^2, \quad \frac{\partial L}{\partial \pi} = 2(\mathbf{p} - \mathbf{p}_{\text{target}}) \cdot \frac{\partial \mathbf{p}}{\partial \pi}. \quad (8)$$

Thus, the scene parameters  $\pi$  can be optimized through iterative backpropagation by minimizing the above loss function.

Though EPSM can handle complex illumination effects, it has some limitations. First, it only considers geometric changes of path, ignoring derivatives of the path contribution. Second, the paths used for computing derivatives are sampled from a normal path tracing framework, some complex light paths like SDS paths are hard to sample in such a framework.

Next, we will enhance EPSMs by introducing contribution derivatives and generalized path gradients in Sec. 4, and use them to handle differentiation of photon mapping in Sec. 5.

## 4 GENERALIZED PATH GRADIENTS

Xing et al. [2023] provided geometric derivatives for EPSMs, i.e., telling how path geometry changes with respect to infinitesimally perturbed scene parameters. However, they did not derive how the contribution of the path change with respect to scene parameters. Such *contribution derivatives* are also important and useful. On the one hand, without contribution derivatives, it will be impossible to optimize parameters like albedo map. On the other hand, including contribution derivatives could lead to a more comprehensive theory of EPSMs. So we enhance EPSMs by further deriving the contribution derivatives in its framework, and refer the results as *generalized path gradients*.

Since the path contribution is a product of multiple terms, before deriving the contribution derivatives, we first look at a more general case on deriving derivatives of a general product function.

### 4.1 Derivatives of a Product Function

**4.1.1 Formulation.** Given a path  $\overline{\mathbf{x}} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k]$  and scene parameters of interest  $\pi = [\pi_1, \pi_2, \dots, \pi_m]$  under the constraints of an EPSM  $C(\overline{\mathbf{x}}, \pi) = \mathbf{0}$ , suppose we have a function  $F(\overline{\mathbf{x}}, \pi)$  which could be written as the product of multiple separate functions  $F_i(\overline{\mathbf{x}}, \pi)$  ( $1 \leq i \leq n$ ):

$$F(\overline{\mathbf{x}}, \pi) = \prod_{i=1}^n F_i(\overline{\mathbf{x}}, \pi). \quad (9)$$

Within the manifold space of EPSM, the path  $\overline{\mathbf{x}}$  can be uniquely determined from scene parameters  $\pi$  in its neighborhood. In other words, the path itself is also a function of scene parameters:  $\overline{\mathbf{x}} = \overline{\mathbf{x}}(\pi)$ . Hence, the above product function could also be written as:

$$F^c(\pi) = F(\overline{\mathbf{x}}(\pi), \pi) = \prod_{i=1}^n F_i^c(\pi), \quad (10)$$

where each separate function is:

$$F_i^c(\pi) = F_i(\overline{\mathbf{x}}(\pi), \pi) = F_i(\overline{\mathbf{x}}, \pi). \quad (11)$$

Note that, to avoid denotation confusion, throughout the entire paper, we will add a superscript  $(\cdot)^c$  to a function when we represent the function as a function of only scene parameters  $\pi$ .

**4.1.2 Computation of the derivatives.** The derivatives of the product function with respect to scene parameters  $\partial F^c / \partial \pi$  can be computed by differentiating Eq. 10 using the *product rule*<sup>1</sup>:

$$\frac{\partial F^c}{\partial \pi} = \sum_{i=1}^n \left( \frac{\partial F_i^c}{\partial \pi} \cdot \prod_{j \neq i} F_j^c(\pi) \right) = F^c(\pi) \cdot \sum_{i=1}^n \left( \frac{\partial F_i^c / \partial \pi}{F_i^c(\pi)} \right), \quad (12)$$

where the derivatives of each separate function can be computed by applying the chain rule to Eq. 11:

$$\frac{\partial F_i^c}{\partial \pi} = \frac{\partial F_i(\overline{\mathbf{x}}(\pi), \pi)}{\partial \pi} = \frac{\partial F_i}{\partial \pi} + \frac{\partial F_i}{\partial \overline{\mathbf{x}}} \cdot \frac{\partial \overline{\mathbf{x}}}{\partial \pi}, \quad (13)$$

where the geometric derivatives  $\partial \overline{\mathbf{x}} / \partial \pi$  are computed using Eq. 2 through implicit partial differentiation of the constraint functions of the EPSM.

The correctness of applying the product rule in Eq. 12 requires that  $F_i^c$  is smooth over the space of scene parameters  $\pi$ , i.e.,  $\partial F_i^c / \partial \pi$  exists. The correctness of applying the chain rule in Eq. 13 requires that  $F_i$  is smooth over the full combined space of paths  $\overline{\mathbf{x}}$  and scene parameters  $\pi$ , i.e., both of  $\partial F_i / \partial \overline{\mathbf{x}}$  and  $\partial F_i / \partial \pi$  exist. Next, we will demonstrate how to derive the contribution derivatives in rendering and ensure their correctness based on the theoretical results above.

### 4.2 Contribution Derivatives and Generalized Path Gradients

**4.2.1 Formulation of the contribution.** the contribution function (or called throughput) of a path  $\overline{\mathbf{x}} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k]$  could be written as:

$$I(\overline{\mathbf{x}}, \pi) = L_e(\overline{\mathbf{x}}, \pi) \prod_{i=1}^{k-1} G_i(\overline{\mathbf{x}}, \pi) \prod_{i=2}^{k-1} \rho_i(\overline{\mathbf{x}}, \pi), \quad (14)$$

where  $L_e(\overline{\mathbf{x}}, \pi) = L_e(\mathbf{x}_1, \pi)$  is the radiance emitted from the light source,  $G_i(\overline{\mathbf{x}}, \pi) = G(\mathbf{x}_i, \mathbf{x}_{i+1}, \pi) = |\cos \theta_{i,i+1} \cos \theta_{i+1,i}| / \|\mathbf{x}_i - \mathbf{x}_{i+1}\|^2$

<sup>1</sup>  $(abc)' = (a)'bc + (b)'ac + (c)'ab = abc \cdot \left( \frac{a'}{a} + \frac{b'}{b} + \frac{c'}{c} \right)$



denotes the geometry factor term, and  $\rho_i(\bar{\mathbf{x}}, \pi) = \rho(\mathbf{x}_{i-1}, \mathbf{x}_i, \mathbf{x}_{i+1}, \pi)$  denotes the BSDF term.

Our goal is to compute the contribution derivatives  $\partial I^c / \partial \pi$ , i.e., tracking how the contribution of the path changes with scene parameters. Please recall that we add a superscript  $(\cdot)^c$  to a function when representing it as a function of only scene parameters ( $I^c(\pi) = I(\bar{\mathbf{x}}(\pi), \pi) = I(\bar{\mathbf{x}}, \pi)$ ).

Notice that the contribution function  $I$  is perfectly in the form of a product function. Hence, once the derivatives of each separate function is obtained, we can use the product rule in Eq. 12 to compute the contribution derivatives  $\partial I^c / \partial \pi$ . For the light radiance function  $L_e$ , geometry factor term functions  $G_i$ , and low-frequency BSDFs (i.e., Lambertian BRDFs or low glossy BRDFs), they are smooth functions so that their derivatives can be computed using the chain rule in Eq. 13. For pure specular (or highly specular BSDFs), they are discontinuous functions (or functions with very large gradients) in the full path space so that the chain rule cannot be applied. We present a decomposition scheme to address this problem.

**4.2.2 BSDF decomposition.** Let's first look at the chain rule again:

$$\frac{\partial \rho^c}{\partial \pi} = \frac{\partial \rho}{\partial \pi} + \frac{\partial \rho}{\partial \bar{\mathbf{x}}} \cdot \frac{\partial \bar{\mathbf{x}}}{\partial \pi}, \quad (15)$$

However, it is not always achievable for BSDFs. This is because the BSDF function  $\rho$  might be discontinuous (and hence undifferentiable) or differentiating it would be numerically unstable when pure specular or near specular materials are used. Imagine a mirror reflection case, the BSDF value could suddenly drop to zero when keeping the incident ray direction fixed while slightly changing the reflected ray direction. In such cases,  $\partial \rho / \partial \bar{\mathbf{x}}$  does not exist or is numerically unstable to compute.

We observe that although contribution function  $\rho$  might be discontinuous in the full space of paths,  $\rho^c$  could be continuous (and differentiable) in some low dimensional manifold space (i.e., the space satisfying the constraints of EPSM  $\mathbf{C}(\bar{\mathbf{x}}, \pi) = \mathbf{0}$ ). Taking the example of mirror reflection again, when the reflected ray direction is changed, the half-vector constraint in the EPSM would always ensure the half-vector is fixed, hence leading to a continuous and smooth change of BSDF values.

The observation motivates us to seek an approach to perform differentiation that takes advantage of the constraint functions. Specifically, we perform the following decomposition to BSDF functions:

$$\rho^c(\pi) = \rho(\bar{\mathbf{x}}, \pi) = g_0(\bar{\mathbf{x}}, \pi) + \sum_{j=1}^n g_j(\bar{\mathbf{x}}, \pi) h_j(\bar{\mathbf{x}}, \pi), \quad (16)$$

where each  $g_j(\bar{\mathbf{x}}, \pi)$  ( $0 \leq j \leq n$ ) is a smooth function in the full path space, while each  $h_j(\bar{\mathbf{x}}, \pi)$  ( $1 \leq j \leq n$ ) could be a discontinuous function in the full path space but it is required to be a *constant-valued function* in the manifold space we construct. The value of  $n$  depends on the complexity of BSDFs and usually  $n = 1$  is enough. Details on how to construct  $g_j$  and  $h_j$  will be explained later.

Since  $h_j(\cdot)$  is constant within the manifold space, the BSDF function in Eq. 16 could be simplified to:

$$\rho^c(\pi) = \rho(\bar{\mathbf{x}}, \pi) = g_0(\bar{\mathbf{x}}, \pi) + \sum_{j=1}^n g_j(\bar{\mathbf{x}}, \pi) h_j, \quad (17)$$

where  $h_j$  denotes the constant value of the function  $h_j(\cdot)$ . After that, the derivatives of the BSDF function could be computed by differentiating Eq. 17:

$$\frac{\partial \rho^c}{\partial \pi} = \left( \frac{\partial g_0}{\partial \pi} + \frac{\partial g_0}{\partial \bar{\mathbf{x}}} \cdot \frac{\partial \bar{\mathbf{x}}}{\partial \pi} \right) + \sum_{j=1}^n \left( \frac{\partial g_j}{\partial \pi} + \frac{\partial g_j}{\partial \bar{\mathbf{x}}} \cdot \frac{\partial \bar{\mathbf{x}}}{\partial \pi} \right) \cdot h_j, \quad (18)$$

where the derivatives ( $\partial g_j / \partial \pi$  and  $\partial g_j / \partial \bar{\mathbf{x}}$ ) can be easily computed since each  $g_j(\cdot)$  is required to be a smooth function.

**4.2.3 An example for BSDF decomposition.** We show how to decompose a typical highly specular microfacet BRDF model [Cook and Torrance 1982]. It could be represented by:

$$\rho(\mathbf{i}, \mathbf{o}) = k_d + k_s M(\mathbf{i}, \mathbf{o}) D(\mathbf{h}), \quad (19)$$

where  $\mathbf{h}$  is the half vector,  $D$  is the normal distribution function, and  $M$  is a smooth function that combines shadowing and Fresnel terms.

The above BRDF could be decomposed as  $\rho(\cdot) = g_0(\cdot) + g_1(\cdot) h_1(\cdot)$  where  $g_0(\cdot) = k_d$ ,  $g_1(\cdot) = M(\mathbf{i}, \mathbf{o})$ , and  $h_1(\cdot) = D(\mathbf{h})$ . Notice that the half vector constraint in the EPSM will enforce the half vector  $\mathbf{h}$  fixed within the manifold space hence leading to a constant valued function  $D(\mathbf{h})$ .

**4.2.4 Generalized path gradients.** We define the generalized path gradients as the combination of the contribution derivatives, geometric derivatives, and the screen space positional derivatives:

$$\left[ \frac{\partial I^c}{\partial \pi}, \frac{\partial \bar{\mathbf{x}}}{\partial \pi}, \frac{\partial \mathbf{p}}{\partial \pi} \right]. \quad (20)$$

Note that the screen space positional derivatives  $\partial \mathbf{p} / \partial \pi$  are directly computed from the geometric derivatives (Eq. 7). We include the screen space positional derivatives within the generalized path gradients since they are directly used in optimization. We call our gradient formulation to be 'generalized' since previous gradient formulations can be viewed as specialized cases of our formulation. We will explain it in the next subsection.

### 4.3 Relationship to Previous Gradient Formulations

Recall that an EPSM is valid as long as the Jacobian matrix Eq. 2 is invertible. Hence, users can freely design their own EPSM by introducing their own constraint functions. Some of the existing gradient formulations can be also viewed as specialized cases of our generalized path gradients by designing specialized EPSMs, as illustrated in Fig. 3.

**4.3.1 Detached Path Replay Backpropagation (PRB).** Detached PRB [Vicini et al. 2021] uses the same pseudorandom numbers to replay the path and keeps each ray direction unchanged in the world space when scene parameters perturb. As illustrated in Fig. 3(a), it can be viewed as using a specialized EPSM with the following constraints:

- (1) enforce the *fixed position constraint* to the eye position  $\mathbf{x}_1$ ;
- (2) for each consequent ray direction  $\overrightarrow{\mathbf{x}_i \mathbf{x}_{i+1}}$  ( $1 \leq i \leq k-1$ ), enforce it to be unchanged in the world space. Note that it is slightly different from the *fixed direction constraint* which enforces a ray direction to be unchanged in the local frame.

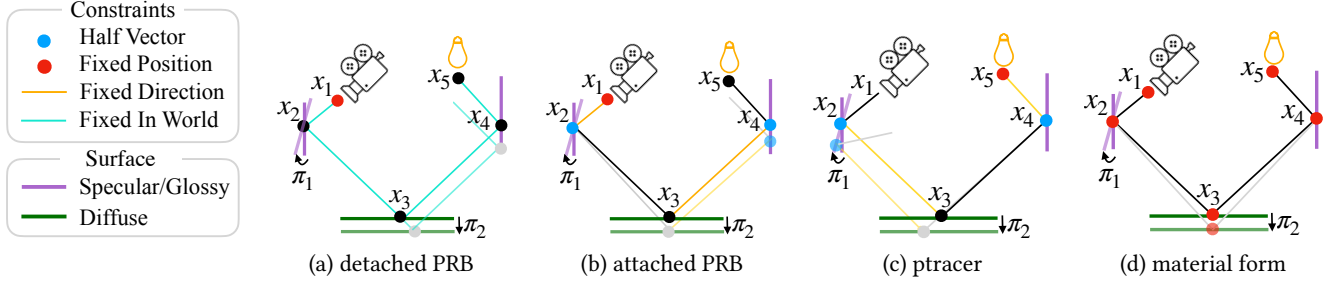


Fig. 3. Existing differentiable rendering methods viewed as specialized EPSMs. (a) detached Path Replay Backpropagation (detached PRB) [Vicini et al. 2021]; (b) attached Path Replay Backpropagation (attached PRB) [Vicini et al. 2021; Zeltner et al. 2021]; (c) ptracer [Nimier-David et al. 2019]; (d) material form [Xu et al. 2023; Zhang et al. 2020].

**4.3.2 Attached Path Replay Backpropagation.** The attached PRB [Vicini et al. 2021; Zeltner et al. 2021] is introduced to address the problem of handling pure specular materials in the detached PRB method. Attached PRB also replays the paths using the same pseudorandom numbers, as done in the detached mode. However, it samples paths in a lower dimensional primary path space, which essentially ensures specular reflections/refractions are correctly handled. As illustrated in Fig. 3(b), it is equivalent to the following specialized EPSM:

- (1) enforce the *fixed position constraint* to the eye position  $\mathbf{x}_1$ ;
- (2) enforce the *fixed direction constraint* to the view ray  $\overrightarrow{\mathbf{x}_1\mathbf{x}_2}$ ;
- (3) for each middle vertex  $\mathbf{x}_i$ : if  $\mathbf{x}_i$  is specular, enforce the *half-vector constraint* to it; otherwise, for vertex  $\mathbf{x}_i$ ,
  - if  $i = k - 1$  and the connection to light source is through next event estimation, enforce the *fixed position constraint* to light source  $\mathbf{x}_k$ ;
  - otherwise, enforce the *fixed direction constraint* to ray  $\overrightarrow{\mathbf{x}_i\mathbf{x}_{i+1}}$ .

**4.3.3 Ptracer.** The ptracer [Nimier-David et al. 2019] is generally similar to the attached PRB except that it uses light tracing instead of path tracing. As illustrated in Fig. 3(c), it can be viewed as the following specialized EPSM:

- (1) enforce the *fixed position constraint* to the light source endpoint  $\mathbf{x}_k$ ;
- (2) enforce the *fixed direction constraint* to the light ray  $\overrightarrow{\mathbf{x}_k\mathbf{x}_{k-1}}$ ;
- (3) for each middle vertex  $\mathbf{x}_i$ : if  $\mathbf{x}_i$  is specular, enforce the *half-vector constraint* to it; otherwise, for vertex  $\mathbf{x}_i$ ,
  - if  $i = 2$  and the connection to eye point is through next event estimation, enforce the *fixed position constraint* to eye point  $\mathbf{x}_1$ ;
  - otherwise, enforce the *fixed direction constraint* to ray  $\overrightarrow{\mathbf{x}_i\mathbf{x}_{i-1}}$ .

**4.3.4 Material Form.** The material form [Xu et al. 2023; Zhang et al. 2020] parameterizes paths in a material space. It simply corresponds to an EPSM where all path vertices are enforced by a *fixed position constraint*. Fig. 3(d) gives an illustration.

**4.3.5 Summary.** As analyzed, the above previous gradient formulations can all be viewed as specialized cases of our formulation by designing specialized EPSMs. These formulations can correctly

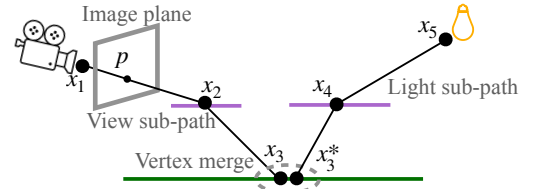


Fig. 4. Illustration of a photon mapping path. The path is composed of a view sub-path  $[\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3]$  and a light sub-path  $[\mathbf{x}_5, \mathbf{x}_4, \mathbf{x}_3^*]$ . Vertex merging happens between  $\mathbf{x}_3$  and  $\mathbf{x}_3^*$ . The screen-space intersection point of the camera ray and the image plane is noted as  $p$ .

compute derivatives in many scenes but still have some limitations. For example, detached PRB and material form are not able to handle contribution derivatives of pure specular paths because they do not employ the half-vector constraint. On the other hand, attached-based methods like attached PRB and ptracer do not ensure the light paths always link camera point and light source, so they may struggle with handling high-frequency effects caused by light transport like caustics. Besides, the above gradient formulations only provide contribution derivatives, while the original EPSM paper [Xing et al. 2023] only provides geometric derivatives. In contrast, our generalized path gradients provide both geometric and contribution derivatives, offering a more unified theoretical framework.

## 5 DIFFERENTIABLE PHOTON MAPPING

Photon mapping is a two-pass algorithm. In the first pass, a photon map is built by tracing photons from the light sources to interact with the scene which stop at diffuse surfaces. From a path tracing aspect, photons can be viewed as light sub-paths sampled from the light source [Georgiev et al. 2012; Hachisuka et al. 2012]. In the second pass, view sub-paths are traced from the eye and query the photon map through kernel-based density estimation. Variants of photon mapping such as SPPM [Hachisuka and Jensen 2009] may reverse the order of the two passes.

As illustrated in Fig. 4, we follow previous works [Georgiev et al. 2012; Hachisuka et al. 2012] to view photon mapping as a path sampling technique, where density estimation between the photon and density estimation location can be intuitively viewed as vertex

merging. Formally, considering a photon  $\mathbf{x}_s^*$  and its tracing history as a light sub-path  $[\mathbf{x}_k, \mathbf{x}_{k-1}, \dots, \mathbf{x}_{s+1}, \mathbf{x}_s^*]$ , and a density estimation location  $\mathbf{x}_s$  and the corresponding view sub-path  $[\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_s]$  traced from the eye, the photon mapping path is defined as [Georgiev et al. 2012; Hachisuka et al. 2012]:

$$\bar{\mathbf{x}} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_s, \mathbf{x}_s^*, \mathbf{x}_{s+1}, \dots, \mathbf{x}_k], \quad (21)$$

which contains  $k + 1$  vertices. The relationship between density estimation position  $\mathbf{x}_s$  and photon position  $\mathbf{x}_s^*$  is vertex merging.

An EPSM can be naturally extended for a photon mapping path. Since one more vertex is included for vertex merging, we need to add an additional constraint function. Below, we first explain how to compute the contribution derivatives for photon mapping paths under the constraints of EPSMs in Sec. 5.1. Then, we explain how to construct EPSMs for photon mapping paths in Sec. 5.2.

## 5.1 Contribution Derivatives for Photon Mapping Paths

**5.1.1 Differentiable density estimation kernel.** In order to make the whole process of photon mapping differentiable, we require a smooth and differentiable density estimation kernel. For this purpose, we use the following kernel function [Hachisuka et al. 2010]:

$$K_r(d) = \begin{cases} \frac{7}{2\pi} (1 - 6d_r^5 + 15d_r^4 - 10d_r^3) & \text{if } d_r \leq 1, \\ 0 & \text{otherwise.} \end{cases} \quad (22)$$

where  $d_r = d/r$  is the relative distance.

**5.1.2 Contribution function of photon mapping.** Considering a photon mapping path  $\bar{\mathbf{x}} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_s, \mathbf{x}_s^*, \mathbf{x}_{s+1}, \dots, \mathbf{x}_k]$  and scene parameters of interest  $\pi$  under the constraints of an EPSM, its contribution to the final rendered image can be computed by:

$$I(\bar{\mathbf{x}}, \pi) = L_e(\bar{\mathbf{x}}, \pi) \left( \prod_{i=1}^{k-1} G_i(\bar{\mathbf{x}}, \pi) \right) \cdot \left( \prod_{i=2}^{k-1} \rho_i(\bar{\mathbf{x}}, \pi) \right) \cdot K_r(\bar{\mathbf{x}}, \pi), \quad (23)$$

where  $L_e$  is the radiance emitted from the light source,  $G_i$  is the geometric factor,  $\rho_i$  is the BSDF function, and  $K_r(\bar{\mathbf{x}}, \pi) = K_r(|\mathbf{x}_s - \mathbf{x}_s^*|)$  is the smooth density estimation function. The geometric factors and BSDF functions are defined in the same way as in Eq. 14, but care must be taken for vertices near  $\mathbf{x}_s$  and  $\mathbf{x}_s^*$ , such as:  $G_s(\bar{\mathbf{x}}, \pi) = G(\mathbf{x}_s^*, \mathbf{x}_{s+1}, \pi)$ ,  $\rho_s(\bar{\mathbf{x}}, \pi) = \rho(\overrightarrow{\mathbf{x}_s \mathbf{x}_{s-1}}, \overrightarrow{\mathbf{x}_s^* \mathbf{x}_{s+1}}, \pi)$ , and  $\rho_{s+1}(\bar{\mathbf{x}}, \pi) = \rho(\overrightarrow{\mathbf{x}_{s+1} \mathbf{x}_s^*}, \overrightarrow{\mathbf{x}_{s+1} \mathbf{x}_{s+2}}, \pi)$ .

Since the above contribution function is a product function, its derivatives with respect to scene parameters can be computed in the same way as in Sec. 4.1 and Sec. 4.2. Note that BSDF decomposition is still required for specular BSDFs.

## 5.2 EPSMs for Differentiable Photon Mapping

To adapt EPSMs with photon mapping paths, as we mentioned before,  $k + 1$  constraints are required. Specifically, we have designed three new EPSMs for differentiable photon mapping to handle different inverse rendering tasks, as illustrated in Fig. 5. All the three EPSMs employ the same constraints on the light sub-path  $[\mathbf{x}_k, \dots, \mathbf{x}_{s+1}, \mathbf{x}_s^*]$ :

- (1) enforce the *fixed position constraint* to the light source vertex  $\mathbf{x}_k$ ;
- (2) enforce the *fixed direction constraint* to the light ray  $\overrightarrow{\mathbf{x}_k \mathbf{x}_{k-1}}$ ;

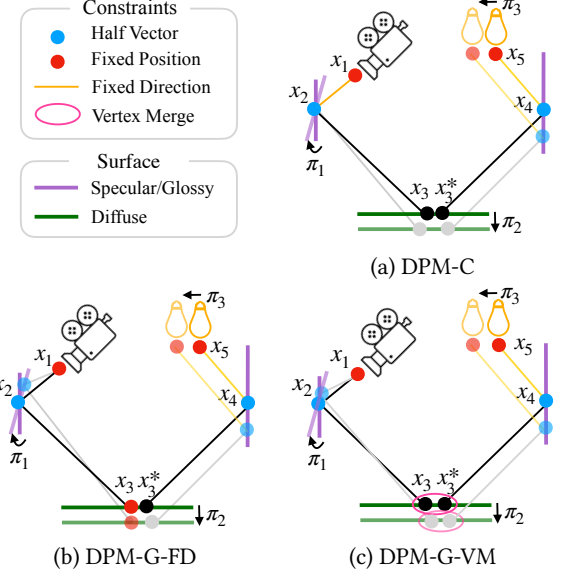


Fig. 5. The three EPSMs designed for differentiable photon mapping.

- (3) for each middle vertex  $\mathbf{x}_i$  ( $s + 1 \leq i \leq k - 1$ ): if it is specular, enforce the *half vector constraint* to it; otherwise, enforce the *fixed direction constraint* to ray  $\overrightarrow{\mathbf{x}_i \mathbf{x}_{i-1}}$ .

The 3 newly introduced EPSMs differ in how to set constraints on the view sub-path  $[\mathbf{x}_1, \dots, \mathbf{x}_s]$ . According to the algorithm of photon mapping, the middle vertices of the view sub-path  $\mathbf{x}_i$  ( $2 \leq i \leq s - 1$ ) are all specular and the density estimation vertex  $\mathbf{x}_s$  is diffuse.

**5.2.1 EPSMs without screen-space positional derivatives (short as DPM-C).** For this EPSM, the constraints of view sub-paths are designed as:

- (1) enforce the *fixed position constraint* to the eye position  $\mathbf{x}_1$ ;
- (2) enforce the *fixed direction constraint* to the eye ray  $\overrightarrow{\mathbf{x}_1 \mathbf{x}_2}$ ;
- (3) enforce the *half-vector constraint* to all middle vertices  $\mathbf{x}_i$  ( $2 \leq i \leq s - 1$ ). Note that all these vertices are specular.

Since this EPSM enforces the eye ray  $\overrightarrow{\mathbf{x}_1 \mathbf{x}_2}$  to be fixed, leading to fixed screen space intersection location ( $\mathbf{p}$ ) and zero-valued screen-space positional derivatives ( $\partial \mathbf{p} / \partial \pi \equiv 0$ ). This makes the optimization process rely solely on the contribution derivatives (see Eq. 25). Note that the geometric derivatives are still very useful in the intermediate derivation process to obtain the correct contribution derivatives.

This EPSM will be suitable for inverse rendering tasks where screen-space derivatives are not so useful, such as cases when the initial and target object locations are relatively close or when optimal transport based matching between initial and target images is not so accurate. We refer to this EPSM (and our different photon mapping method based on it) as *DPM-C* for short.

**5.2.2 EPSMs with full generalized path gradients (short as DPM-G).** To deal with inverse rendering tasks when the initial and target locations of illumination effects of interest are far away in the screen space, the screen-space derivatives are useful and should be included. We further introduced two EPSMs which do not fix the eye ray for

this purpose. These two EPSMs allow changeable eye ray  $\overrightarrow{x_1 x_2}$  and could make use of both screen-space and contribution derivatives in optimization. Specifically, one of them is designed by additionally including the following constraints of view sub-paths:

- (1) enforce the *fixed position constraint* to the eye position  $x_1$  and to the density estimation vertex  $x_s$ ;
- (2) enforce the *half-vector constraint* to all middle vertices  $x_i$  ( $2 \leq i \leq s-1$ ).

The main feature of this EPSM is it enforces the density estimation vertex  $x_s$  to be locally fixed. We refer to it as *fixed density EPSM*.

The other EPSM is designed with the following additional constraints:

- (1) enforce the *fixed position constraint* to the eye position  $x_1$ ;
- (2) enforce the *half-vector constraint* to all middle vertices  $x_i$  ( $2 \leq i \leq s-1$ );
- (3) enforce the photon position  $x_s^*$  to be unchanged in the local frame of the density estimation location  $x_s$ :

$$T_{x_s}(x_s^*) = \mathbf{const}, \quad (24)$$

where  $T_{x_s}(\cdot)$  transforms a position from world space to the local frame of  $x_s$ . We refer to this newly introduced constraint as *vertex merging constraint*.

The *vertex merging constraint* intuitively keeps the relative distance between photon  $x_s^*$  and density estimation location  $x_s$ . Hence, it helps to directly propagate geometric gradients from the view sub-path to the light sub-path, and is rather suitable for optimizing location of caustics. We refer to this EPSM as *vertex merging EPSM*.

To avoid manually choosing a specific strategy, we further propose a hybrid strategy: for each path, we build two EPSMs and use the averaged contribution and screen-space positional derivatives computed from the two EPSMs for optimization. We refer to this strategy (and our different photon mapping method based on it) as *DPM-G* for short.

**5.2.3 Which one to use?** Generally speaking, DPM-C (Sec. 5.2.1), which provides zero-valued screen-space positional derivatives, is more suitable for such inverse rendering tasks:

- (1) only contribution-related scene parameters like albedo map are being optimized;
- (2) the initial and target locations of the objects (or rendering effects) being optimized are already close to each other so that pixel-to-pixel matching is not needed;
- (3) while an accurate pixel-to-pixel matching cannot be obtained. For example, when the goal is to optimize for caustic effects with a desired shape on a wall while initially the wall has a constant color.

In contrast, DPM-G (Sec. 5.2.2) is more suitable for tasks when the initial and target locations of the objects (or rendering effects) are distant from each other and we can find an accurate pixel-to-pixel matching between them.

### 5.3 Optimization

Given a scene with initial scene parameters and a target image, a typical inverse rendering task is to optimize the scene parameters

in order to minimize the difference between the rendered image and the target image.

For DPM-C, since its screen-space positional derivatives are zero-valued by definition, only contribution derivatives can be used in optimization. Hence, we choose commonly used L2 loss for optimization.

For DPM-G, i.e., the EPSMs with full generalized path gradients, we follow Xing et al. [2023] to find a dense path-to-pixel matching using approximated optimal transport. But we add a color difference term to the original loss function (Eq. 8) in order to make use of our contribution derivatives:

$$L(\pi) = \lambda_1 (\mathbf{p} - \mathbf{p}_{\text{target}})^2 + \lambda_2 (I^c(\pi) - I_{\text{target}})^2, \quad (25)$$

$$\frac{\partial L}{\partial \pi} = 2\lambda_1 (\mathbf{p} - \mathbf{p}_{\text{target}}) \cdot \frac{\partial \mathbf{p}}{\partial \pi} + 2\lambda_2 (I^c(\pi) - I_{\text{target}}) \cdot \frac{\partial I^c}{\partial \pi},$$

where  $I_{\text{target}}$  is the color value of the matched pixel. The weights are empirically set as  $\lambda_1 = \lambda_2 = 0.5$  following previous work [Xing et al. 2022]. Notice that the loss derivatives can be directly computed from our generalized path gradients.

## 6 IMPLEMENTATION

We demonstrate our differentiable framework on SPPM [Hachisuka and Jensen 2009; Pharr et al. 2016]. We choose SPPM due to its ease of implementation and its constant memory usage.

At each optimization iteration, we run SPPM for 16 passes. In each pass, for forward rendering, we sample 1 eye sub-path on each pixel (i.e., 1 spp) and sample  $16 \times$  photons (or light sub-paths) compared to the number of eye sub-paths. For backward rendering and differentiable computations, an ideal case is to obtain all photon mapping paths by merging eye sub-paths with nearby light sub-paths and constructing one EPSM for each photon mapping path. For better efficiency, we do not use all photon mapping paths, instead, we only construct EPSMs from a few of the paths. Specifically, we randomly sample  $1/16$  pixels, and sample one photon mapping path from each sampled pixel.

We implement our entire method on GPU based on the open source code provided by the original EPSM paper [Xing et al. 2023], with mixed-use of MITSUBA 3 [Jakob et al. 2022b,a], PyTORCH [Paszke et al. 2019] and LUISARENDER [Zheng et al. 2022]. Typically, forward rendering and path related computations are implemented using MITSUBA 3. Optimal transport, solving the linear system for the geometric derivatives (Eq. 2) are implemented using PyTORCH. We implement a hash grid for efficiently query nearby density estimation points for each photon on LUISARENDER, due to the flexibility of LUISARENDER in supporting user-written customized GPU kernels, while writing such data structures on MITSUBA 3 or PyTORCH seems difficult. Data transmission between different frameworks is efficient because the underlying data in GPU memory does not need to be moved or duplicated.

## 7 EXPERIMENTS

All experiments are performed on a PC with an NVIDIA RTX 3090 GPU (24G memory). By default, we set the rendering resolution as  $512 \times 512$ , and the number of optimization iterations as 1000. We use the Adam optimizer [Kingma and Ba 2015] for backpropagation.

One iteration of optimization typically takes about 2.2-5.4 seconds for DPM-C, and takes about 3.0 - 8.0 seconds for DPM-G. The time includes both forward rendering and backpropagation.

## 7.1 Analysis and Evaluation

**7.1.1 Visualization of the gradients.** As shown in Fig. 6, to verify the correctness of our gradient formulation, we visualize the contribution derivatives computed from our DPM-C method and compare them to the ground truth derivatives computed from finite differences. We use DPM-C since it ensures zero-valued screen-space positional derivatives so that its contribution derivatives are comparable to those of existing gradient formulations. The tested scene is composed of a small area light, a floor, and a curved mirror-like reflector with low roughness. The contribution derivatives are computed with respect to the rotation angle of the reflector along the Y-axis. Notice the reflection highlights on the reflector and the caustics on the floor.

From the results, we could find that our gradients match well with the ground truth using 256 spps (1 spp/pass  $\times$  256 passes), while the gradients of *detached PRB* [Vicini et al. 2021] and *ptracer* [Nimier-David et al. 2019] still exhibit large differences with the ground truth. Also, notice that *ptracer* converges relatively faster on the caustics but slower on the reflector, while *detached PRB* converges faster on the reflector but slower on the caustics. In contrast, our method converges well in both areas.

**7.1.2 Variants of differentiable photon mappings.** As mentioned in Sec. 5.2, we have offered two variants of differentiable photon mapping methods DPM-C and DPM-G. In DPM-G, we use a hybrid of two EPSMs *fixed density* and *vertex merging*. We evaluate DPM-C, DPM-G and using DPM-G with only *fixed density* (short as DPM-G (fd-only)) or *vertex merging* (short as DPM-G (vm-only)) on three scenes to find the scope of each variant. Visual results are given in Fig. 7.

In the scene FISHBOWL (Fig. 7 (a)), the goal is to optimize the geometry (i.e., a height field) of the water surface from the caustic pattern on the background. The camera is outside the fishbowl so that the caustics paths are in the form of **ESDSL**. Notice that DPM-C is able to basically recover the caustic pattern while a few noises still exhibit. In contrast, the three other method variants (DPM-G, DPM-G (fd-only) and DPM-G (vm-only)) cannot achieve a good recovery. Because the background in the initial image is generally clear, we can only get a relatively poor matching with optimal transport, making the screen-space positional derivatives in these methods less useful.

However, DPM-C is less successful in scenes EGG (Fig. 7 (b)) and GLASSBOX (Fig. 7 (c)). Both scenes exhibit long-range relationships between the initial and target images, e.g., the caustics in scene EGG and the objects in scene GLASSBOX. Since DPM-C does not use screen-space positional derivatives in optimization, it is not able to deal with such long-range relationships.

In contrast, such long-range relationships can be captured well with screen-space positional derivatives and matching based loss function. In particular, DPM-G (vm-only) captures the long-range relationships between caustics (Fig. 7 (b)) well due to the inclusion

of the *vertex merging constraint* and DPM-G (fd-only) captures long-range relationships between objects (Fig. 7 (c)) well since it enforces the density estimation point to be fixed, while their combination DPM-G is able to handle both cases well.

## 7.2 Comparisons and Results

**7.2.1 Scene configurations.** We have collected 7 scene configurations for testing and comparisons. All these scenes contain complex illumination effects, such as reflections, refractions, caustics, etc.

Scene RING (Fig. 1) contains a ring object behind a magnifying glass within the Cornell box. The optimizing parameters include the color of the diamond on the ring and the thickness of the glass. Initially, the magnifying glass is very thin and the goal is to thicken the magnifying glass and to modify the color of the diamond to match the magnified view of the ring.

Scenes WINEUP and WINELOW (Fig. 8 (a) and (b)) use the same geometry and lighting, and optimize for the same scene parameters. They differ in the camera location. In each scene, there are two glasses of liquid and a point light, and the optimizing parameter is the color of the liquid. In scene WINEUP, the camera faces towards the liquid with refraction effects. In scene WINELOW, the camera faces toward the bottom of the wineglass with caustic effects.

Scene IMAGE (Fig. 8(c)) is adapted from Mitsuba's documentation [mit [n. d.]] with one modification. From left to right, there are directional lights with a Bayer pattern, a camera viewing towards right, a glass slab, and a receiving plane. The goal is to optimize the geometry (height field) of the glass slab to match a desired caustic pattern. Note that originally the camera is between the glass slab and the receiving plane in Mitsuba's documentation, but we move the camera to the location between the lights and the glass slab to increase the difficulty of recovery, i.e., the light paths are changed from **EDS+L** to **ES+DS+L**.

In scene EGG (Fig. 9(a)), the goal is to recover the height of the light source through the caustics on the table. Scene GLASSBOX (Fig. 9(b)) contains multiple objects within a glass box, and the goal is to optimize the 2D positions and the colors of objects through refractions. Scene MIXED (Fig. 9(c)) is composed of a textured Armadillo, a glass sphere, and a glass egg. The optimizing parameters include the position of the sphere, the roughness of the egg, and the albedo map of the Armadillo.

**7.2.2 Visual Comparisons.** We compare our method with several state-of-the-art physically based differentiable rendering methods on the 7 collected scenes. The baseline methods include: attached path tracing (attached PT), ptracer [Nimier-David et al. 2019], Plateau-reduced Differentiable Path Tracing (PRDPT) [Fischer and Ritschel 2022], and the original EPSM method (ori. EPSM) [Xing et al. 2023]. Since the code of attached path replay backpropagation (attached PRB) [Vicini et al. 2021] is not provided by the authors, we implement a simplified version of attached PRB named *attached path tracing (attached PT)* and compare our method with this simplified version. Attached PT is desired to produce the same results as attached PRB but requires more memories. For all scenes and for all methods being compared, in the optimization process, the rendering uses a resolution of  $512 \times 512$  and a sampling rate of 16 spps.



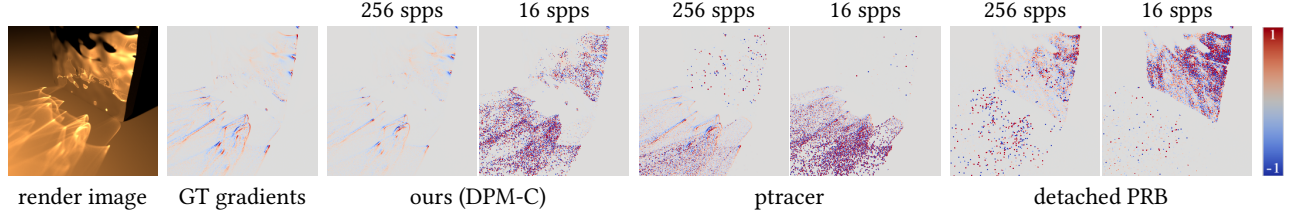


Fig. 6. Visualization of gradients.

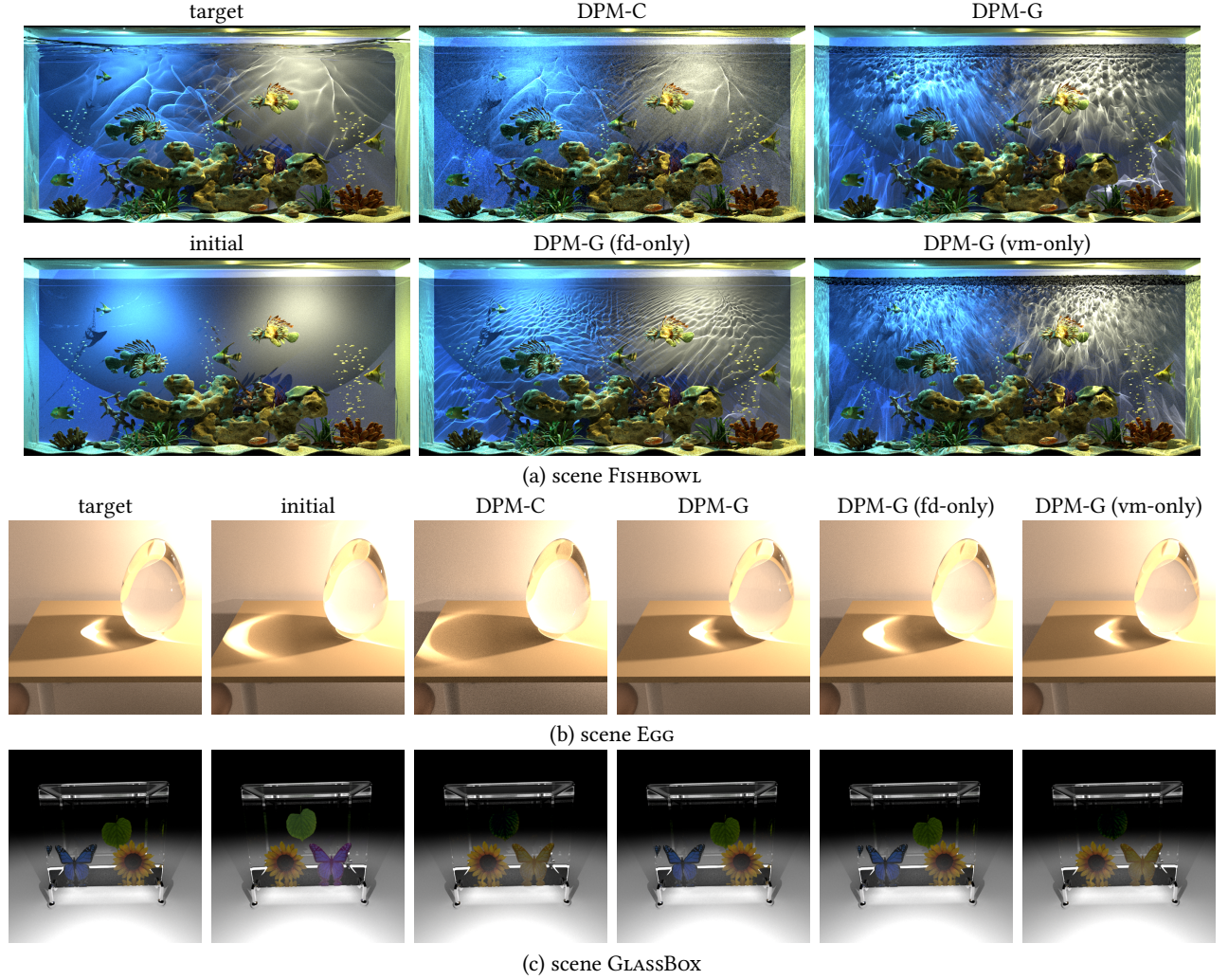


Fig. 7. Evaluations on variants of differentiable photon mapping methods.

According to the guidelines in Sec. 5.2.3, we use DPM-C for scenes WINEUP, WINELOW and *Image*, and use DPM-G for the other scenes (RING, EGG, MIXED, and GLASSBOX).

Fig. 8 shows visual comparisons between our DPM-C method with the baseline approaches. As shown in Fig. 8 (a) and (b), attached PT is able to recover the refractions in scene WINEUP while failing

to recover the caustics in scene WINEUP. Interestingly, ptracer in the reverse case: it is able to basically recover the caustics while failing to recover the the refractions. This is because attached PT utilizes a path tracing style integrator so that it is hard to sample caustics paths ( $\mathbf{ES}^*\mathbf{DS}+\mathbf{L}$ ) while ptracer uses a light tracing style integrator so that it is hard to sample refraction paths ( $\mathbf{ES}+\mathbf{L}$ ).

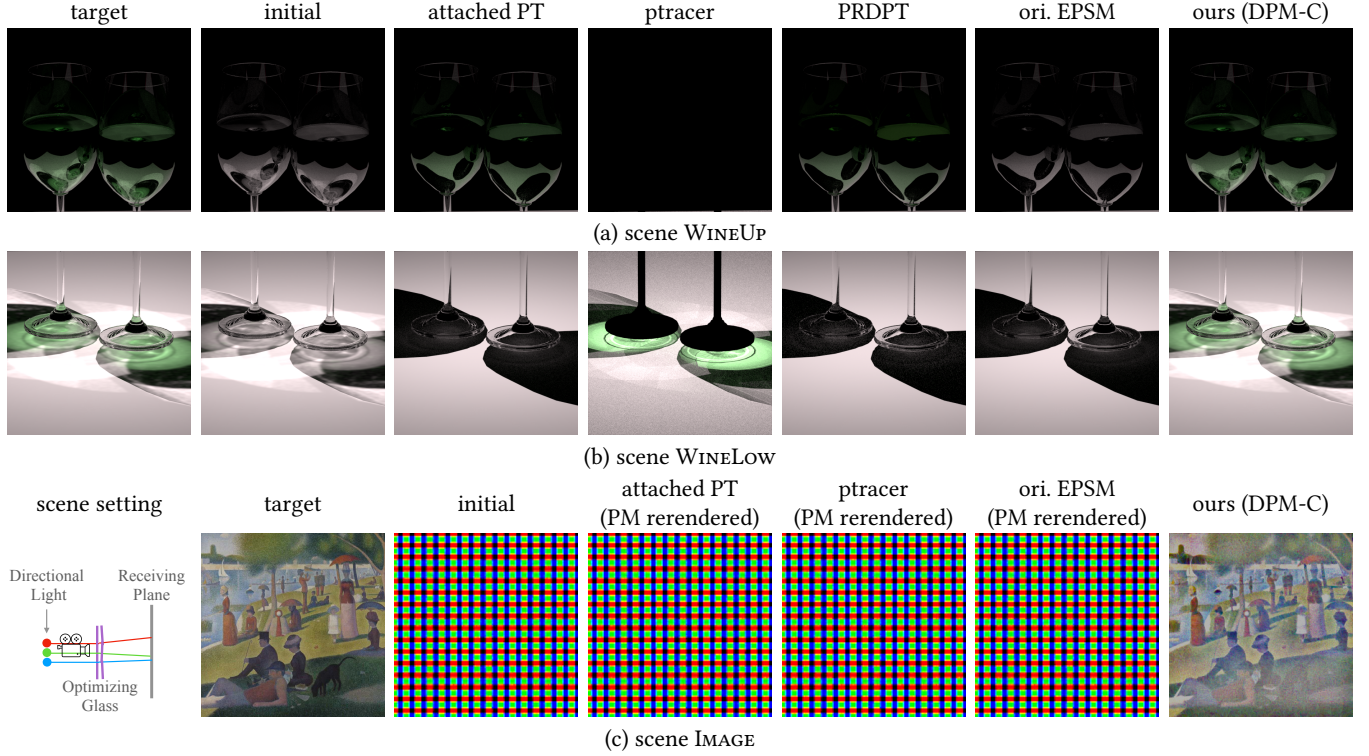


Fig. 8. Visual comparisons between our method (DPM-C) and the baseline methods, including attached PT, ptracer [Nimier-David et al. 2019], Plateau-reduced Differentiable Path Tracing (PRDPT) [Fischer and Ritschel 2022], the original EPSM method (ori. EPSM) [Xing et al. 2023]. For better display quality, in scene IMAGE, photon mapping is used for re-rendering the displayed visual results for attached PT, ptracer, and ori. EPSM, respectively, since those methods are not able to sample SDS paths (marked as ‘PM rerendered’); in other cases, the displayed visual results of each method shown in the figure are re-rendered with 8192 spp using that method.

Scene IMAGE in Fig. 8(c) is a very difficult task since it involves SDS paths ( $\mathbf{ES} + \mathbf{DS} + \mathbf{L}$ ). All baseline approaches cannot successfully reconstruct the height map of the glass slab to match the desired caustic pattern. It is simply because the baseline approaches are all based on path tracing or light tracing and cannot successfully sample the SDS paths in forward rendering. Notice that the light sources are directional lights but not area lights. In contrast, our method is able to recover the caustic pattern to some extent.

Fig. 1 and Fig. 9 show the visual comparison between our DPM-G method with the baselines. Notice that the baseline approaches cannot recover the scene parameters of interest in most cases.

The failures of baseline approaches in recovering the scene parameters are due to the following reasons. First, since attached PT and ptracer only rely on contribution derivatives for optimization and no geometric derivatives are involved, it would be difficult for them to handle long-range relationships between the initial and target images. For example, the movement of caustics in scene EGG (Fig. 9 (a)), the movement of refracted objects in scene GLASSBOX (Fig. 9 (b)), and the magnification effect of the ring in scene RING (Fig. 1). Second, while the original EPSM method is able to capture long-range relationships, it lacks contribution derivatives so that it fails to recover color changes, such as the liquid color in scene WINEUP (Fig. 8 (a)), and Armadillo’s albedo map in scene MIXED

(Fig. 9 (c)). Third, since existing approaches use path tracing or light tracing as their forward renderer, they cannot handle certain types of paths, such as SDS paths, as demonstrated by scene IMAGE in Fig. 8(c). Finally, while PRDPT is able to handle refractions (scene WINEUP in Fig. 8(a)) and capture long-range relationships to some extent, we notice that its optimization process is rather unstable (see scenes EGG and GLASSBOX); furthermore, due to the increased sampling variance in a higher dimensional space, it cannot handle optimization tasks with a relatively large number of optimizing parameters, such as the height map of the glass slab in scene IMAGE, and Armadillo’s albedo map in scene MIXED.

In contrast, due to the effectiveness of our generalized path gradients and the ability of photon mapping in handling difficult paths such as SDS paths, our method is able to successfully recover the scene parameters for all these inverse rendering tasks.

Table 1 provides a comprehensive summary of the abilities of baseline approaches in handling the tested scenes. Fig. 10 shows the error curves of the optimization process. Notice that our method converges quickly on all tested scenes. The animation of the optimization process can be found in the supplemental video.



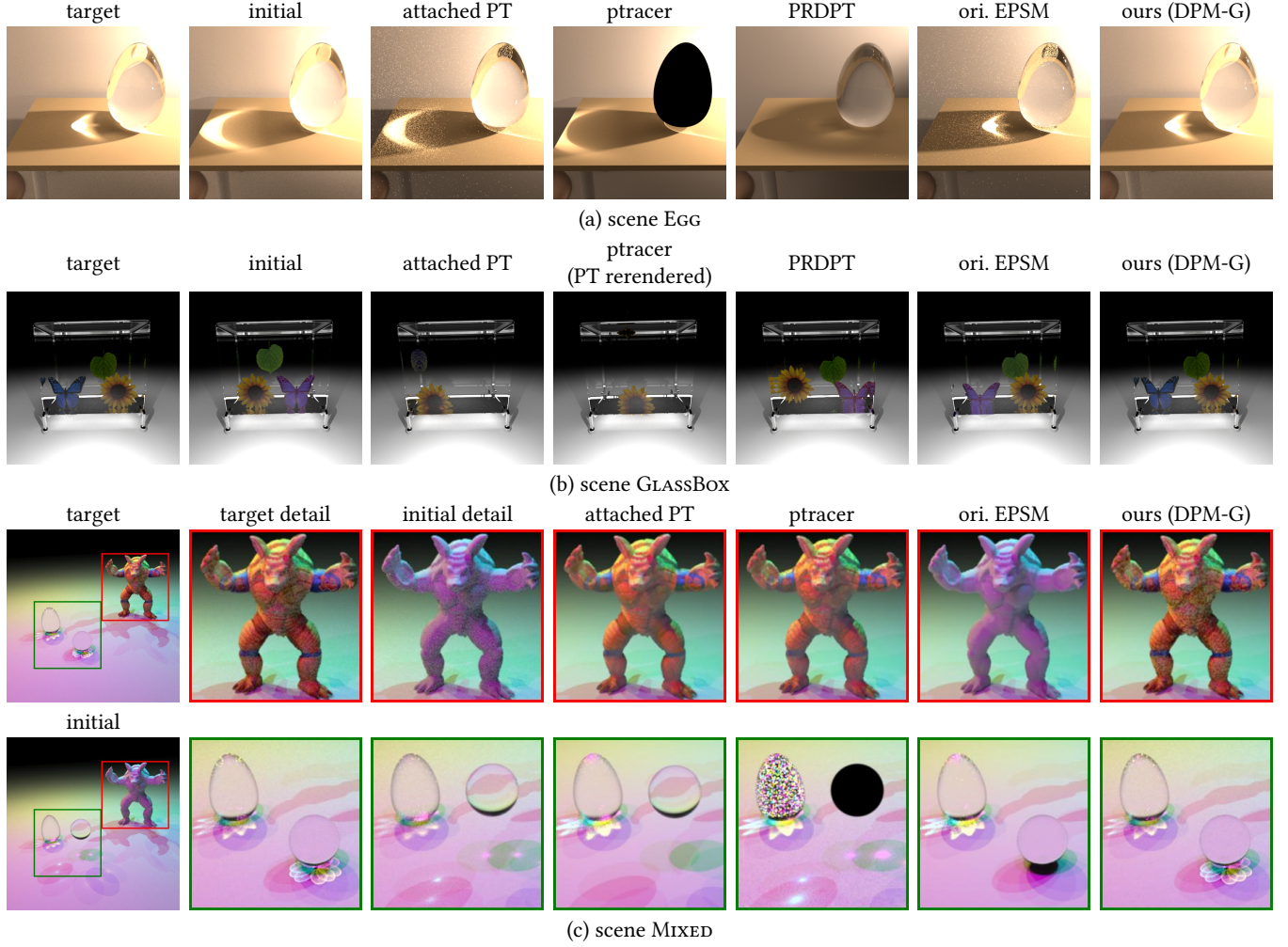


Fig. 9. Visual comparisons between our method (DPM-G) and the baseline methods, including attached PT, ptracer [Nimier-David et al. 2019], Plateau-reduced Differentiable Path Tracing (PRDPT) [Fischer and Ritschel 2022], and the original EPSM method (ori. EPSM) [Xing et al. 2023]. For better display quality, in scene GLASSBox, path tracing with 8192 spps is used for re-rendering the displayed visual results for ptracer (marked as ‘PT rerendered’); in other cases, the displayed visual results of each method shown in the figure is re-rendered with 8192 spps using that method.

**7.2.3 Additional comparisons with detached PRB.** We additionally conducted experiments using the detached path replay backpropagation method (detached PRB) [Zhang et al. 2023]. To enhance its ability in handling long-range relationships, we follow the comparison settings in Xing et al. [2023] to employ a multi-scale scheme to detached PRB. The results are given in Fig. 11. The results generated using our DPM-G method variant are also provided for comparison. Notice that detached PRB cannot handle such inverse rendering tasks with long-range relationships, even equipped with a multi-scale scheme.

## 8 CONCLUSION

In this paper, we have introduced the concept and mathematical derivations of the generalized path gradients for the extended path space manifolds (EPSMs) [Xing et al. 2023]. This concept defines

both contribution derivatives and geometric derivatives of a path with respect to scene parameters under the geometric constraints of EPSMs. We also show that many existing gradient formulations can be viewed as specialized cases of ours. Then, by formalizing photon mapping as a path sampling technique through vertex merging [Georgiev et al. 2012; Hachisuka et al. 2012], we further propose a differentiable photon mapping method based on the theoretical results of generalized path gradients. Experiments and comparisons show that our method is more effective and robust than existing physics-based differentiable rendering methods in inverse rendering tasks with complex illumination effects.

While our method has shown promising results in many inverse rendering tasks involving complex illumination effects, there is still room for improvement. First, geometry discontinuity (i.e., visibility change) is not considered in our method. Researchers have provided

Table 1. A comprehensive summary of the abilities of baseline approaches in handling the tested scenes.

scene	optimizing parameters (dimension)	attached PT	ptracer	PRDPT	ori. EPSM	ours
WINEUP Fig. 8 (a)	color (3)	✓	unable to sample ES+L paths	✓	not support color derivatives	✓ (DPM-C)
WINELOW Fig. 8 (b)	color (3)	unable to sample ES*DS+L paths	✓	unable to sample ES*DS+L paths	unable to sample ES*DS+L paths	✓ (DPM-C)
IMAGE Fig. 8 (c)	heightmap (128x128)	unable to sample ES+DS+L paths	unable to sample ES+DS+L paths	too many parameters	unable to sample ES+DS+L paths	✓ (DPM-C)
EGG Fig. 9 (a)	translation (1)	unable to perform long range optimization	unable to perform long range optimization	unstable optimization	✓	✓ (DPM-G)
GLASSBOX Fig. 9 (b)	translation (2) color (3)	unable to perform long range optimization	unable to sample ESDSL paths	too many parameters	not support color derivatives	✓ (DPM-G)
MIXED Fig. 9 (c)	translation (1) material (1) albedo map (256x256)	unable to perform long range optimization	unable to perform long range optimization	too many parameters	not support color derivatives	✓ (DPM-G)
RING Fig. 1	scale (1) color (3)	unable to perform long range optimization	unable to perform long range optimization	unstable optimization	not support color derivatives	✓ (DPM-G)

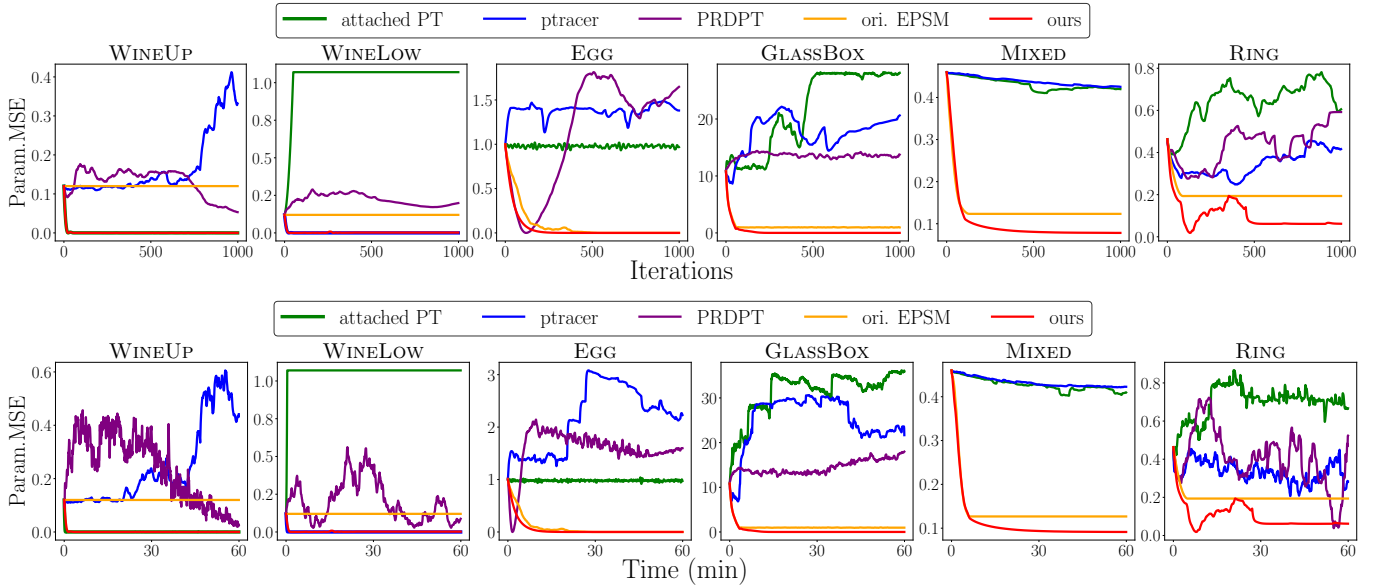


Fig. 10. Error curves. We show how the errors change during the optimization process in both equal iteration comparison and equal time comparison. We provide the MSE between optimized and target scene parameters of interest.

several effective solutions [Bangaru et al. 2020; Li et al. 2018; Xu et al. 2023] for this problem, and our method could be probably combined with those works in the future. Second, the success of our method relies on an accurate pixel-to-pixel matching with optimal transport. A sophisticated matching algorithm that could be more accurate and more efficient deserves future work. Finally, we currently offer multiple candidates of EPSMs and require users to manually select

the best one (i.e., DPM-C or DPM-G) for their tasks. It is worthwhile to develop new types of EPSMs that can be adapted to all inverse rendering tasks, or automatic strategies to select the most suitable EPSMs from candidates.

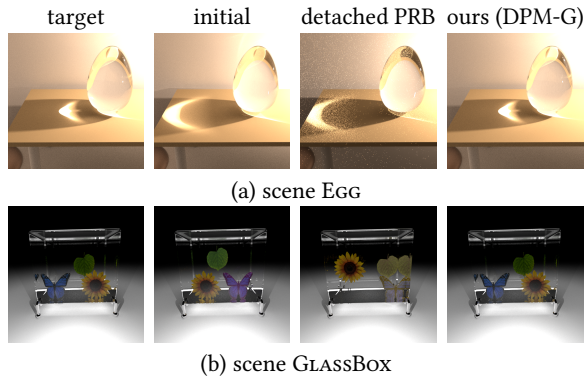


Fig. 11. Comparison with detached PRB [Zhang et al. 2023]. The parameters for handling geometry discontinuity are all set by its default configuration. The multi-scale scheme used for it is the same as the one used in Xing et al. [2023].

## ACKNOWLEDGMENTS

We would like to thank all reviewers for their helpful comments. This work is supported by the National Natural Science Foundation of China (Project Number: 61932003).

## REFERENCES

- [n. d.]. Caustics optimization - Mitsuba 3. [https://mitsuba.readthedocs.io/en/latest/src/inverse\\_rendering/caustics\\_optimization.html](https://mitsuba.readthedocs.io/en/latest/src/inverse_rendering/caustics_optimization.html)
- Sai Praveen Bangaru, Tzu-Mao Li, and Frédo Durand. 2020. Unbiased warped-area sampling for differentiable rendering. *ACM Transactions on Graphics (TOG)* 39, 6 (2020), 1–18.
- Benedikt Bitterli and Wojciech Jarosz. 2017. Beyond points and beams: Higher-dimensional photon samples for volumetric light transport. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 1–12.
- R. L. Cook and K. E. Torrance. 1982. A Reflectance Model for Computer Graphics. *ACM transactions on graphics* 1, 1 (jan 1982), 7–24. <https://doi.org/10.1145/357290.357293>
- Marco Cuturi. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems* 26 (2013).
- Xi Deng, Shaojie Jiao, Benedikt Bitterli, and Wojciech Jarosz. 2019. Photon surfaces for robust, unbiased volumetric density estimation. *ACM Transactions on Graphics* 38, 4 (2019).
- Jean Feydy, Thibault Séjourné, François-Xavier Vialard, Shun-ichi Amari, Alain Trounev, and Gabriel Peyré. 2019. Interpolating between Optimal Transport and MMD using Sinkhorn Divergences. In *The 22nd International Conference on Artificial Intelligence and Statistics*. 2681–2690.
- Michael Fischer and Tobias Ritschel. 2022. Plateau-reduced Differentiable Path Tracing. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022).
- Iliyan Georgiev, Jaroslav Krivánek, Tomáš Davidovič, and Philipp Slusallek. 2012. Light transport simulation with vertex connection and merging. *ACM transactions on graphics* 31, 6, Article 192 (nov 2012), 10 pages. <https://doi.org/10.1145/2366145.2366211>
- Toshiya Hachisuka, Wojciech Jarosz, and Henrik Wann Jensen. 2010. A progressive error estimation framework for photon density estimation. *ACM Trans. Graph.* 29, 6, Article 144 (dec 2010), 12 pages. <https://doi.org/10.1145/1882261.1866170>
- Toshiya Hachisuka and Henrik Wann Jensen. 2009. Stochastic progressive photon mapping. In *ACM SIGGRAPH Asia 2009 papers*. 1–8.
- Toshiya Hachisuka, Shinji Ogaki, and Henrik Wann Jensen. 2008. Progressive photon mapping. In *ACM SIGGRAPH Asia 2008 papers*. 1–8.
- Toshiya Hachisuka, Jacopo Pantaleoni, and Henrik Wann Jensen. 2012. A path space extension for robust light transport simulation. *ACM Trans. Graph.* 31, 6, Article 191 (nov 2012), 10 pages. <https://doi.org/10.1145/2366145.2366210>
- Wenzel Jakob and Steve Marschner. 2012. Manifold exploration: A markov chain monte carlo technique for rendering scenes with difficult specular transport. *ACM Transactions on Graphics (TOG)* 31, 4 (2012), 1–13.
- Wenzel Jakob, Sébastien Speierer, Nicolas Roussel, Merlin Nimier-David, Delio Vicini, Tizian Zeltner, Baptiste Nicolet, Miguel Crespo, Vincent Leroy, and Ziyi Zhang. 2022b. *Mitsuba 3 renderer*. <https://mitsuba-renderer.org>.
- Wenzel Jakob, Sébastien Speierer, Nicolas Roussel, and Delio Vicini. 2022a. DrJit: A Just-In-Time Compiler for Differentiable Rendering. *Transactions on Graphics (Proceedings of SIGGRAPH)* 41, 4 (July 2022). <https://doi.org/10.1145/3528223.3530099>
- Wojciech Jarosz, Derek Nowrouzezahrai, Iman Sadeghi, and Henrik Wann Jensen. 2011. A comprehensive theory of volumetric radiance estimation using photon points and beams. *ACM transactions on graphics (TOG)* 30, 1 (2011), 1–19.
- Wojciech Jarosz, Matthias Zwicker, and Henrik Wann Jensen. 2008. The beam radiance estimate for volumetric photon mapping. In *ACM SIGGRAPH 2008 classes*. 1–12.
- Henrik Wann Jensen. 1996. Global illumination using photon maps. In *Rendering Techniques '96: Proceedings of the Eurographics Workshop in Porto, Portugal, June 17–19, 1996*. 7. Springer, 21–30.
- James T. Kajiya. 1986. The rendering equation. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques - SIGGRAPH '86*. ACM Press. <https://doi.org/10.1145/15922.15902>
- Markus Kettunen, Marco Manzi, Miika Aittala, Jaakko Lehtinen, Frédo Durand, and Matthias Zwicker. 2015. Gradient-domain path tracing. *ACM Trans. Graph.* 34, 4, Article 123 (jul 2015), 13 pages. <https://doi.org/10.1145/2766997>
- Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). <http://arxiv.org/abs/1412.6980>
- Tzu-Mao Li, Miika Aittala, Frédo Durand, and Jaakko Lehtinen. 2018. Differentiable monte carlo ray tracing through edge sampling. *ACM Transactions on Graphics (TOG)* 37, 6 (2018), 1–11.
- Daqi Lin\*, Markus Kettunen\*, Benedikt Bitterli, Jacopo Pantaleoni, Cem Yuksel, and Chris Wyman. 2022. Generalized Resampled Importance Sampling: Foundations of ReSTIR. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2022)* 41, 4, Article 75 (07 2022), 23 pages. (\*Joint First Authors).
- Guillaume Loubet, Nicolas Holzschuch, and Wenzel Jakob. 2019. Reparameterizing discontinuous integrands for differentiable rendering. *ACM Transactions on Graphics (TOG)* 38, 6 (2019), 1–14.
- Merlin Nimier-David, Delio Vicini, Tizian Zeltner, and Wenzel Jakob. 2019. Mitsuba 2: A retargetable forward and inverse renderer. *ACM Transactions on Graphics (TOG)* 38, 6 (2019), 1–17.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems* 32. Curran Associates, Inc., 8024–8035. <http://papers.nips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>
- Matt Pharr, Wenzel Jakob, and Greg Humphreys. 2016. *Physically Based Rendering: From Theory to Implementation (3rd ed.)* (3rd ed.). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. 1266 pages.
- Eric Veach. 1997. *Robust Monte Carlo methods for light transport simulation*. Stanford University.
- Delio Vicini, Sébastien Speierer, and Wenzel Jakob. 2021. Path Replay Backpropagation: Differentiating Light Paths using Constant Memory and Linear Time. *Transactions on Graphics (Proceedings of SIGGRAPH)* 40, 4 (Aug. 2021), 108:1–108:14. <https://doi.org/10.1145/3450626.3459804>
- Delio Vicini, Sébastien Speierer, and Wenzel Jakob. 2022. Differentiable signed distance function rendering. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 1–18.
- 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*. 1–11.
- Jiankai Xing, Fujun Luan, Ling-Qi Yan, Xuejun Hu, Houde Qian, and Kun Xu. 2022. Differentiable Rendering using RGBXY Derivatives and Optimal Transport. *ACM Trans. Graph.* 41, 6, Article 189 (dec 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. <https://doi.org/10.1145/3618330>
- Tizian Zeltner, Sébastien Speierer, Iliyan Georgiev, and Wenzel Jakob. 2021. Monte Carlo estimators for differential light transport. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–16.
- Cheng Zhang, Bailey Miller, Kan Yan, Ioannis Gkioulekas, and Shuang Zhao. 2020. Path-space differentiable rendering. *ACM transactions on graphics* 39, 4 (2020).
- Cheng Zhang, Zihan Yu, and Shuang Zhao. 2021. Path-space differentiable rendering of participating media. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–15.
- Ziyi Zhang, Nicolas Roussel, and Wenzel Jakob. 2023. Projective Sampling for Differentiable Rendering of Geometry. *Transactions on Graphics (Proceedings of SIGGRAPH Asia)* 42, 6 (Dec. 2023). <https://doi.org/10.1145/3618385>
- Shaokun Zheng, Zhiqian Zhou, Xin Chen, Difei Yan, Chuyan Zhang, Yuefeng Geng, Yan Gu, and Kun Xu. 2022. LuisaRender: A High-Performance Rendering Framework with Layered and Unified Interfaces on Stream Architectures. *ACM Trans. Graph.* 41, 6, Article 232 (nov 2022), 19 pages.