Monte Carlo Estimators for Differential Light Transport

TIZIAN ZELTNER, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
SÉBASTIEN SPEIERER, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
ILIYAN GEORGIEV, Autodesk, United Kingdom
WENZEL JAKOB, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland

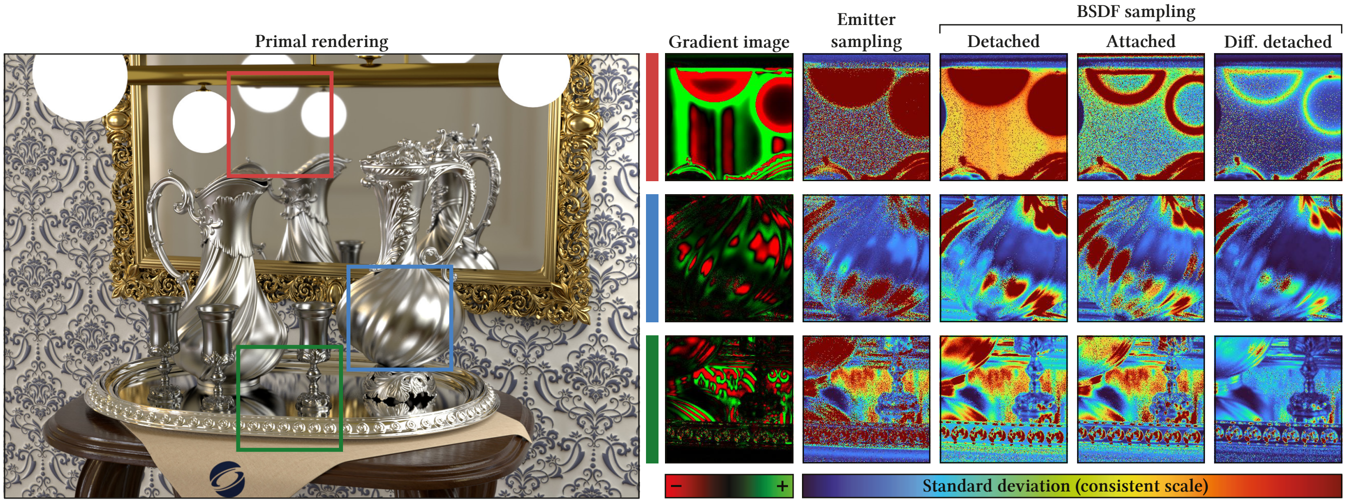

Fig. 1. Differentiable rendering of a scene featuring specular interreflection between metallic surfaces of varying roughness. We differentiate the image with respect to the combined roughness of all objects, which produces the gradients shown in the first column with insets. A disconcertingly large number of differential estimators can solve this problem, albeit with drastically different statistical efficiency: the following four columns highlight the standard deviation of emitter sampling and three material-based strategies. An overview of the exhaustive set of combinations (21 methods) and results for an additional four estimators are provided in the supplemental material, which also contains uncropped images. The objective of our work is to provide intuition on how to navigate the large design space of differential Monte Carlo estimators.

Physically based differentiable rendering algorithms propagate derivatives through realistic light transport simulations and have applications in diverse areas including inverse reconstruction and machine learning. Recent progress has led to unbiased methods that can simultaneously compute derivatives with respect to millions of parameters. At the same time, elementary properties of these methods remain poorly understood.

Current algorithms for differentiable rendering are constructed by mechanically differentiating a given primal algorithm. While convenient, such an approach is simplistic because it leaves no room for improvement. Differentiation produces major changes in the integrals that occur throughout the rendering process, which indicates that the primal and differential algorithms should be decoupled so that the latter can suitably adapt.

This leads to a large space of possibilities: consider that even the most basic Monte Carlo path tracer already involves several design choices concerning the techniques for sampling materials and emitters, and their combination, e.g. via multiple importance sampling (MIS). Differentiation causes a veritable explosion of this decision tree: should we differentiate only the estimator, or also the sampling technique? Should MIS be applied before or after differentiation? Are specialized derivative sampling strategies of any use? How should visibility-related discontinuities be handled when millions of parameters are differentiated simultaneously? In this paper, we provide a taxonomy and analysis of different estimators for differential light transport to provide intuition about these and related questions.

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

Additional Key Words and Phrases: differentiable rendering, inverse rendering, differentiating visibility, radiative backpropagation

ACM Reference Format:
Tizian Zeltner, Sébastien Speierer, Iliyan Georgiev, and Wenzel Jakob. 2021. Monte Carlo Estimators for Differential Light Transport. *ACM Trans. Graph.* 40, 4, Article 78 (August 2021), 16 pages. https://doi.org/10.1145/3450626.3459807

Authors' addresses: Tizian Zeltner, tizian.zeltner@epfl.ch, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland; Sébastien Speierer, sebastien.speierer@epfl.ch, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland; Iliyan Georgiev, iliyan.georgiev@autodesk.com, Autodesk, London, United Kingdom; Wenzel Jakob, wenzel.jakob@epfl.ch, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland.

1 INTRODUCTION

The inverse analysis of images formed by visible light and other electromagnetic radiation is a central problem in many scientific and engineering disciplines. For example, a CT scanner images a region of space using angularly spaced measurements, but the resulting data is of little use without a reconstruction technique that can invert the process of X-ray absorption to reveal the interior. Structured light techniques analyze photographs of objects under carefully designed illumination patterns to infer their three-dimensional shape.

Such tailored methods are highly effective within their design scope, but they can fail when central assumptions are violated. For example, CT reconstruction normally assumes absorptive materials and tends to produce severe artifacts when the specimen contains metal fragments that are highly reflective to X-rays. Structured light techniques encounter issues when specular parts of an object refocus illumination patterns onto unintended surface regions. Rather than addressing these specific flaws, our goal is to study a universal mathematical framework that has the potential to improve the quality of solutions in these and similar challenging inversion tasks in the future.

In a general, image formation is the result of the complex interplay of shape, illumination, and materials, in which indirect effects like shadowing and interreflection couple distant parts of the scene: a bright spot on a surface could, e.g., be explained by texture or shape variation, illumination from a light source, or focused reflection from another object. Resolving this ambiguity requires multiple observations and reconstruction techniques that account for the interconnected nature of light transport and scattering.

In this article, we study the mathematical principles of *differentiable rendering*, which formulates the inversion process as a gradient-based optimization task defined on a high-dimensional domain with millions of *scene parameters* specifying illumination, shapes, and materials. Scene parameter derivatives of the rendered image encode cues to unravel the radiative coupling, and they also provide an important direction of steepest ascent in this high-dimensional space, making them an invaluable asset for solving inverse problems involving light. Differentiability is also required when rendering occurs as part of a larger differentiable calculation, such as a neural autoencoder or generative adversarial network.

So far, the creation of differentiable rendering algorithms has followed a fairly rigid sequence of steps: derivatives are first moved into light transport integrals solved by a standard method (e.g., path tracing), possibly with extra steps to handle visibility-related discontinuities. Subsequent differentiation of the integrand involves the standard rules of calculus and can be performed by hand, or using software-based techniques for *automatic differentiation* (AD).

While this approach generally works, we observe that differentiation fundamentally changes the nature of the underlying integrals. A scene parameter can be *sensitive* in the sense that a small perturbation of its value would lead to a significant positive or negative change in the value of the integrand that affects the rendered image and optimization objective. Monte Carlo theory then tells us that a low-variance gradient estimator should place a proportional number of samples into this region. However, this type of adaptation is simply impossible when the differential rendering algorithm is rigidly created from its primal counterpart. In the worst case, the sensitive region could even be zero-valued and discarded during primal integration, in which case the differential algorithm is biased.

Recent work by Nimier-David et al. [2019] proposed a method termed *radiative backpropagation* (RB), which casts differentiable rendering into the form of an *adjoint* (i.e., reversed) transport problem that propagates derivative "radiation" from sensors towards objects with differentiable parameters. Their formulation decouples the primal and differential estimators and provides the starting point for our investigation of the latter.

This decoupling brings considerable additional freedom but also reveals that elementary aspects of differentiable rendering remain poorly understood. In this paper, we investigate the following choices that guide the design of differential transport estimators:

- Estimators that apply importance sampling often do so using the inversion method, which involves a mapping to transform uniform variates to the target distribution. When creating the differential estimator, this mapping could remain unchanged, or it could be differentiated along with the integrand. We refer to these respectively as *detached* and *attached* strategies. The former produce static samples, while the latter capture the infinitesimal motion of samples with respect to parameter changes.
- Sampling strategies are almost never used alone, but in combination with others via the framework of *multiple importance sampling* (MIS). Once more, the primal MIS weights could be used as-is or differentiated to track infinitesimal changes.
- Sampling strategies are designed to approximate the shape of the associated integrand, but this property may no longer hold following differentiation regardless of whether attached or detached strategies are used. In such cases, it may be possible to design tailored strategies that match this new integrand, which we refer to as *differential strategies*. We propose one such strategy for the commonly used family of microfacet models. Our analysis demonstrates clear benefits of specialized differential strategies, while the trade-offs between detached and attached sampling remain more nuanced and problem-dependent.
- Visibility-related discontinuities require careful treatment to avoid bias in computed gradients. We explain how recent techniques that are designed to sidestep this issue [Loubet et al. 2019; Bangaru et al. 2020] can be adapted to the adjoint framework of RB, enabling efficient and unbiased geometric optimization.
- Not all options are compatible with each other: some combinations of attached/detached MIS and attached/detached sampling strategies yield biased estimators, and attached sampling strategies can interfere with techniques to handle discontinuous integrals. We show how sampling strategies can simultaneously be attached yet behave correctly in the presence of discontinuities.
- Finally, rendering algorithms frequently take discrete random decisions including path termination via Russian roulette and sampling of multi-lobed BSDFs. We show that these steps should *never* be differentiated, as this would severely bias their result.

The remainder of this paper provides a taxonomy of differential estimators based on this bewilderingly large set of possibilities.

2 PRIOR WORK AND BACKGROUND

Differential light transport in other fields. Derivatives of Monte Carlo simulations that are highly related to recent work on differentiable rendering have been used to model the criticality of nuclear reactors [Lux and Koblinger 1990], and to perform inverse modeling of tissue [Hayakawa et al. 2001]. These types of methods are named *Perturbation Monte Carlo* or *Differential Monte Carlo.*

Derivatives in rendering. Analytic and approximate ray-space derivatives are part of the standard graphics pipeline that computes texture-space footprints to filter texture lookups via mip-mapping and elliptically weighted averages [Heckbert 1989]. Such footprints can be further propagated following interaction with smooth [Igehy 1999] and rough [Belcour et al. 2013] materials. Other uses of derivative information in rendering include gradient-domain rendering [Hua et al. 2019], adaptive sampling and reconstruction [Ramamoorthi et al. 2007], and the interpolation of local solutions of diffuse [Ward and Heckbert 1992] and non-diffuse [Křivánek et al. 2005] global illumination.

Light paths with specular and near-specular interactions present many challenges in rendering algorithms. Path derivatives enable efficient search [Mitchell and Hanrahan 1992; Jakob and Marschner 2012; Zeltner et al. 2020] and extrapolation [Chen and Arvo 2000] of specular path configurations.

Differentiable rendering in computer vision. Inverse rendering is a standard problem in computer vision, where a considerable body of prior work has investigated ways of differentiating the process of image formation. Indirect effects like shadows, interreflection, and depth of field have historically played a lesser role during this process, and related works investigating differentiable rasterization of meshes and volumes thus mainly focus on primary visibility [Loper and Black 2014; Rhodin et al. 2015; Kato et al. 2018; Liu et al. 2019; Petersen et al. 2019; Laine et al. 2020].

Differentiable rendering in computer graphics. Physically based rendering algorithms [Pharr et al. 2016] account for indirect effects, which involves Monte Carlo sampling of integrals of the form

$$I(\boldsymbol{\pi}) = \int_{\mathcal{X}} f(\mathbf{x}, \boldsymbol{\pi})\, \mathrm{d}\mathbf{x}, \tag{1}$$

where $\boldsymbol{\pi}$ refers to a set of *scene parameters*. The domain $\mathcal{X}$ typically consists of light paths $(\mathbf{x}_0, \ldots, \mathbf{x}_n)$ connecting a light source to a sensor via intermediate scattering events $\mathbf{x}_1, \ldots, \mathbf{x}_{n-1}$. In this work, we are concerned with individual (hemi-)spherical integrals that may reference nested integrals, hence we set $\mathcal{X} = S^2$. We have not investigated path-space methods [Zhang et al. 2020], though it is likely that many of our observations will generalize.

Physically based differentiable rendering algorithms [Li et al. 2018; Azinović et al. 2019; Nimier-David et al. 2019] estimate the partial derivative of the above integral with respect to π:[1]

$$\partial_\pi I(\pi) = \partial_\pi \left[\int_{\mathcal{X}} f(\mathbf{x}, \pi)\, \mathrm{d}\mathbf{x} \right], \tag{2}$$

where we use a shorthand notation $\partial_\pi := \partial/\partial_\pi$. To reduce clutter, we also omit the dependence of I and $\partial_\pi I$ on π.

Indirect effects are especially important when optimizing materials like participating media that are characterized by significant multiple scattering [Gkioulekas et al. 2013, 2016; Zhao et al. 2016; Che et al. 2018; Zhang et al. 2019]. Initial work on physically based differentiable rendering relied on *forward-mode differentiation* to propagate an infinitesimal perturbation through the simulation, requiring a separate run for each parameter of interest. Later techniques applied *reverse-mode differentiation* [Li et al. 2018; Nimier-David et al. 2019] to compute derivatives with respect to all scene parameters at once.

Reverse-mode differentiation is a widely used tool [Griewank and Walther 2008] that greatly improves the efficiency when many derivatives are desired, but it also introduces its own set of problems: derivative evaluation now requires access to intermediate steps of the primal computation, and this sequence of accesses furthermore occurs in reverse order compared to the original program execution. Program reversal is impractical without at least some temporary storage of primal variables, and the size of this scratch space tends to be overwhelming in the context of rendering.

The *radiative backpropagation* (RB) method [Nimier-David et al. 2020] addresses this issue by observing that the derivative program effectively solves a separate type of transport problem where derivative "radiation" that corresponds to the derivative of the objective in pixel space is "emitted" from the camera, "scatters" from scene objects, and is eventually "received" by differentiable scene objects that now take the role of the sensor. Instead of being constrained by the inflexibility and memory overheads of automatically differentiating a primal algorithm in reverse mode, one can thus create differential algorithms that directly solve this modified transport problem. Our work builds on this idea and leverages the decoupled nature of primal and differential phases to evolve the latter.

In a concurrent publication, Vicini et al. [2021] propose algorithms that can evaluate attached and detached differential estimators studied in this article using linear time and constant space complexity.

Geometric discontinuities. The integrand f is generally riddled with discontinuities in the incident radiance (e.g. due to geometric edges), and the position of these discontinuities furthermore depends on the scene parameters π that are to be differentiated. Differentiation under the integral sign is invalid under these conditions and produces biased estimators. Instead one must apply the Reynolds transport theorem [Zhang et al. 2019, 2020], which introduces an additional boundary correction term:

$$\partial_\pi I = \int_{\mathcal{X}} \partial_\pi f(\mathbf{x}, \pi)\, \mathrm{d}\mathbf{x} + \oint_{\partial\mathcal{X}(\pi)} f(\mathbf{x}, \pi)\ \langle \partial_\pi \mathbf{x}, \hat{\mathbf{n}} \rangle\, \mathrm{d}\mathbf{x}. \tag{3}$$

Here, $\hat{\mathbf{n}}$ denotes the normal direction at $\mathbf{x} \in \partial\mathcal{X}(\pi)$. Li et al. [2018] proposed the first method to correctly account for this effect by importance sampling the set of silhouettes edges observed from a given scene location. However, existing data structures for this sampling step exhibit poor scaling as the geometric complexity grows. Zhang et al. [2020] consider a higher-dimensional path space, which provides access to additional edge sampling strategies.

Loubet et al. [2019] observed that the presence of discontinuities is not problematic in itself: bias arises only due to the dependence of their position on scene parameters $\boldsymbol{\pi}$. They propose a *change*

[1] For conceptual and notational simplicity, we take the derivative with respect to a single parameter π. However, our final algorithms will evaluate *all* derivatives at once.

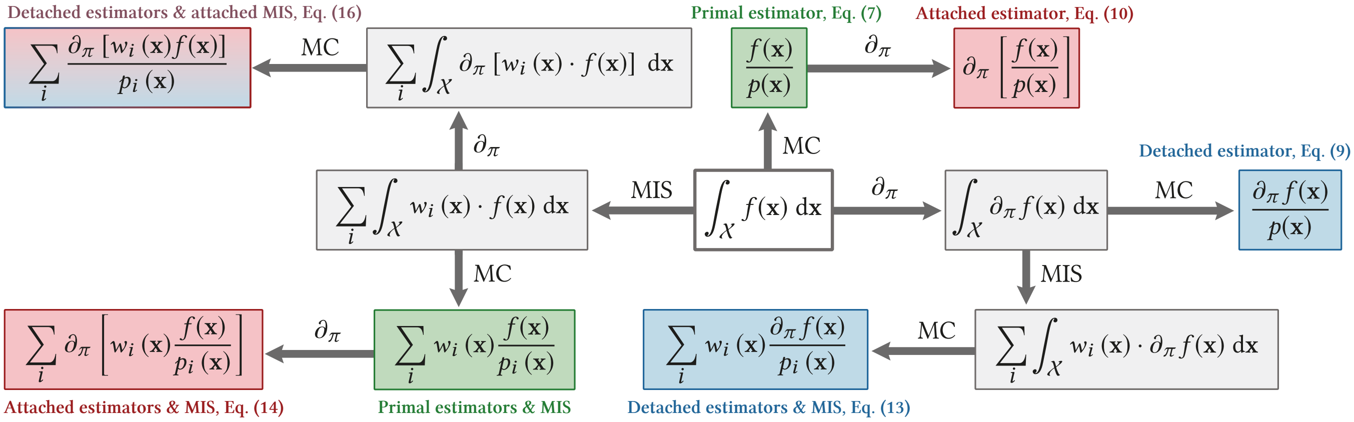
Detached estimators & attached MIS, Eq. (16)
$\sum_i \frac{\partial_\pi [w_i(\mathbf{x}) f(\mathbf{x})]}{p_i(\mathbf{x})}$
MC
$\sum_i \int_\mathcal{X} \partial_\pi [w_i(\mathbf{x}) \cdot f(\mathbf{x})] \, \mathrm{d}\mathbf{x}$
Primal estimator, Eq. (7)
$\frac{f(\mathbf{x})}{p(\mathbf{x})}$
∂_π
Attached estimator, Eq. (10)
$\partial_\pi \left[\frac{f(\mathbf{x})}{p(\mathbf{x})}\right]$
∂_π
MC
Detached estimator, Eq. (9)
$\sum_i \int_\mathcal{X} w_i(\mathbf{x}) \cdot f(\mathbf{x}) \, \mathrm{d}\mathbf{x}$
MIS
$\int_\mathcal{X} f(\mathbf{x}) \, \mathrm{d}\mathbf{x}$
∂_π
$\int_\mathcal{X} \partial_\pi f(\mathbf{x}) \, \mathrm{d}\mathbf{x}$
MC
$\frac{\partial_\pi f(\mathbf{x})}{p(\mathbf{x})}$
MC
MIS
$\sum_i \partial_\pi \left[w_i(\mathbf{x}) \frac{f(\mathbf{x})}{p_i(\mathbf{x})}\right]$
∂_π
$\sum_i w_i(\mathbf{x}) \frac{f(\mathbf{x})}{p_i(\mathbf{x})}$
$\sum_i w_i(\mathbf{x}) \frac{\partial_\pi f(\mathbf{x})}{p_i(\mathbf{x})}$
MC
$\sum_i \int_\mathcal{X} w_i(\mathbf{x}) \cdot \partial_\pi f(\mathbf{x}) \, \mathrm{d}\mathbf{x}$
Attached estimators & MIS, Eq. (14)
Primal estimators & MIS
Detached estimators & MIS, Eq. (13)

Fig. 2. A taxonomy of differential estimators. We illustrate key operations that can be applied to a "primal" integral (white box). These include Monte Carlo importance sampling, multiple importance sampling, and differentiation. Non-commutativity of these operations leads to a plethora of differential estimators that we study in this article. We omit the explicit dependence of f and p on π for brevity. Equation numbers refer to the corresponding locations in the text.

of variables based on a bijective spherical parameterization $R(\mathbf{x}, \pi)$ with Jacobian determinant $|J_R|$ that leaves the value of the integral unchanged, while freezing discontinuities in place. Following this change, the partial derivative can then be moved into the integral:

$$\partial_\pi I = \int_\mathcal{X} \partial_\pi \big[f(R(\mathbf{x}, \pi), \pi) \; |J_R(\mathbf{x}, \pi)| \big] \, \mathrm{d}\mathbf{x}. \tag{4}$$

Since the boundary correction is no longer needed, this integration only involves standard interior estimators. The specific change of variables proposed by Loubet et al. is approximate, however: it rotates the spherical domain and cannot counteract all silhouette motion unless it perfectly matches a spherical rotation as well.

Bangaru et al. [2020] observed that the divergence theorem can be applied to Equation (3), turning the troublesome boundary integral into an interior integral:

$$\partial_\pi I = \int_\mathcal{X} \partial_\pi f(\mathbf{x}, \pi) \, \mathrm{d}\mathbf{x} + \int_\mathcal{X} \nabla_\mathbf{x} \cdot \big(f(\mathbf{x}, \pi) \; V(\mathbf{x}, \pi)\big) \, \mathrm{d}\mathbf{x}, \tag{5}$$

where the *warp field* $V(\mathbf{x}, \pi)$ smoothly interpolates the boundary velocity $\partial_\pi \mathbf{x}$ from Equation (3). This formulation is ultimately shown to be equivalent to the change of variables approach of Loubet et al. In particular, the new divergence term directly corresponds to the derivative of the Jacobian in Equation (4), and there is a one-to-one correspondence between warp fields and parameterizations of integrals. An important contribution of Bangaru et al. [2020] is a novel warp field that smoothly tends to the correct velocity as one approaches a boundary. In the change of variables formulation, this can be interpreted as counteracting all boundary motion.

Neither of these techniques is readily usable in the framework of RB. We show how the warp field of Bangaru et al. can be used as a spherical parameterization that is queried as part of a memory-less reverse mode differentiation procedure. Interestingly, bias due to discontinuities can also arise from attached sampling strategies, which happens even when geometry is not part of the optimization process! We introduce a modified parameterization that addresses this problem.

3 DIFFERENTIAL ESTIMATORS

This section introduces several estimators for differential transport illustrated by the taxonomy in Figure 2. We analyze their properties and correctness, assuming for now that the underlying integrals are free of discontinuities. Section 5 will revisit the discontinuous case and discuss interactions that arise due to the choices made here.

3.1 Detached sampling strategies

We begin with the most basic case that we refer to as the *detached strategy* for reasons that will become clear shortly when we contrast it to *attached* strategies. This approach corresponds to how one would ordinarily differentiate an integral with pencil and paper, i.e., without focusing on its eventual numerical evaluation. We simply move the partial derivative and differentiate under the integral sign:

$$\partial_\pi I = \partial_\pi \left[\int_\mathcal{X} f(\mathbf{x}, \pi) \, \mathrm{d}\mathbf{x} \right] = \int_\mathcal{X} \partial_\pi f(\mathbf{x}, \pi) \, \mathrm{d}\mathbf{x}. \tag{6}$$

This transformation is legal if the integral is free of discontinuities. It also holds when any present discontinuities are *static*, i.e., independent of the parameter π being differentiated.

Importance sampling. Transport integrals in computer graphics are almost exclusively evaluated using Monte Carlo estimators based on importance sampling, and we must therefore understand how this interacts with differentiation. We focus on the classical *inverse transform sampling* [Devroye 1986] which involves a diffeomorphism $T\colon \mathcal{U} \to \mathcal{X}$ that parameterizes the target domain $\mathcal{X}$ by the unit-hypercube $\mathcal{U} = [0, 1]^n$ of matching dimension n. The mapping $\mathbf{x} = T(\mathbf{u})$ is constructed from a target density $p(\mathbf{x})$ so that its Jacobian determinant satisfies $|J(\mathbf{u})| = p(\mathbf{x})^{-1}$. The reparameterized primal integral then takes the form

$$I = \int_\mathcal{U} f(T(\mathbf{u})) \, |J(\mathbf{u})| \mathrm{d}\mathbf{u} = \int_\mathcal{U} \frac{f(T(\mathbf{u}))}{p(T(\mathbf{u}))} \, \mathrm{d}\mathbf{u}. \tag{7}$$

If $p(\mathbf{x}) \approx f(\mathbf{x})$, the integrand is near unity, in which case Monte Carlo estimates of this integral are characterized by low variance.

The mapping transforms uniformly distributed points on $\mathcal{U}$ into samples on $\mathcal{X}$ with density $p(\mathbf{x})$.

Equation (7) is the expected value of a corresponding Monte Carlo estimator obtained by replacing the integration with evaluation of f/p with uniform samples $\mathbf{u} \in \mathcal{U}$. As a slight abuse of terminology we will also refer to these integrals as estimators. Their more conventional expressions can be found in Figure 2 together with references to the equation numbers in the text.

With this notation established, let us now return to differential estimators. The change of variables in Equation (7) can be straightforwardly applied to a differential integral $\int_{\mathcal{X}} \partial_\pi f(\mathbf{x})\,\mathrm{d}\mathbf{x}$:

$$\partial_\pi I = \int_{\mathcal{U}} \frac{\partial_\pi f(T(\mathbf{u}))}{p(T(\mathbf{u}))}\,\mathrm{d}\mathbf{u}. \tag{8}$$

In the case of rendering, the integrand f depends on the scene parameters $\pi \in \Pi$, and in primal estimators the density p and sampling technique T will generally also share this dependence to enable efficient scene-adaptive importance sampling. In the differential setting, prior work [Loubet et al. 2019] has handled this dependence by introducing another conceptual parameter variable π_0, whose value happens to match π, but that otherwise does not participate in the differentiation. In this case, the inverse transform warp and density can depend on π_0 to benefit from specialized primal sampling strategies, which finally gives us the expression of the *detached* estimator

$$\partial_\pi I = \int_{\mathcal{U}} \frac{\partial_\pi f(T(\mathbf{u}, \pi_0), \pi)}{p(T(\mathbf{u}, \pi_0), \pi_0)}\,\mathrm{d}\mathbf{u}. \tag{9}$$

While not considering part of the expression during differentiation may intuitively appear incorrect, the above expression remains a valid estimator as long as the primal strategy samples all positions where $\partial_\pi f \neq 0$ with nonzero probability. This requirement may be violated in practice and requires special precautions in differential rendering algorithms, e.g., by ensuring a minimum density even in zero-valued regions of the integrand. An example where this would be necessary is a spatially varying emitter with zero-valued regions that can potentially be "turned on" by the optimization process.

It is important to realize that a high-quality primal sampling strategy with $p \approx f$ is not necessarily also a good choice for the differential estimator of $\partial_\pi f$, as illustrated in Figure 3. As with standard (i.e., non-differential) Monte Carlo estimators, the effectiveness of a strategy depends on how well its sampling density p matches the integrand $\partial_\pi f$.

Detached sampling strategies also cannot be used when the integrand contains a Dirac delta function, which collapses the integration domain. This affects tasks like computing derivatives with respect to the surface normal of a mirror.

3.2 Attached sampling strategies

Many widely used sampling strategies depend on scene parameters. Examples from the context of physically-based rendering include:

(1) Sampling of directionally peaked distributions like microfacet models that depend on a roughness parameter.
(2) Directional sampling of environment maps proportionally to their textured intensity.

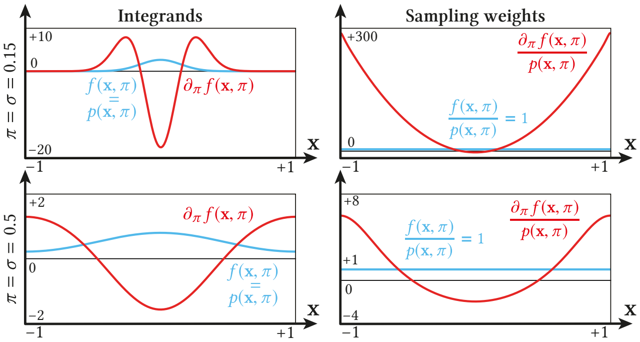

Fig. 3. 1D examples of *detached* samplers. **Top left**: f (blue) follows a wrapped normal distribution parameterized by standard deviation $\pi = \sigma$. Its derivative $\partial_\pi f$ (red) has a markedly different shape. **Top right**: The sampling density $p = f$ yields a zero-variance primal estimator $f/p = 1$ (blue), while the detached estimator of the derivative $\partial_\pi f/p$ (red) produces large sampling weights. **Bottom row**: The same experiment with a different value of π. The problem is less pronounced as $\partial_\pi f$ and p become more uniform.

(3) Any BSDF sampling method that has an implicit dependence on the local frame which is computed from the shading normals (and ultimately, the surface positions).

In this case, the generated samples conceptually move when we perturb the associated scene parameter π. The previously discussed strategy discarded these effects and was thus *detached* from this motion, motivating its name. We now turn to *attached* strategies that do account for the additional dependence. With this change, everything including the function f, the transformation T, and the division by the probability p are jointly differentiated.

$$\partial_\pi I = \int_{\mathcal{U}} \partial_\pi \left[\frac{f(T(\mathbf{u}, \pi), \pi)}{p(T(\mathbf{u}, \pi), \pi)} \right] \mathrm{d}\mathbf{u}. \tag{10}$$

It is interesting to note that the attached strategy will usually be produced *by default* when Monte Carlo sampling code is transparently differentiated using techniques for automatic differentiation. Consequently, attached sampling is also possible across more than one scattering event. In that case, the integration domain $\mathcal{U}$ and its counterpart $\mathcal{X}$ simply have higher dimension. Let us briefly consider the setting where $\dim \mathcal{U} = \dim \mathcal{X} = 1$ to better understand Equation (10). Applying the above derivatives then yields

$$\begin{aligned} \partial_\pi I = \int_{\mathcal{U}} \frac{1}{p(x,\pi)^2} \cdot \big[& p(x,\pi)\,(f_\pi(x,\pi) + T_\pi(u,\pi)\,f_x(x,\pi)) \\ & - f(x,\pi)\,(p_\pi(x,\pi) + T_\pi(u,\pi)\,p_x(x,\pi)) \big]\,\mathrm{d}u. \end{aligned} \tag{11}$$

where $T(u, \pi)$ has been replaced by x for readability, and the subscripts x and π indicate partial derivatives with respect to the first and second function argument, respectively. What can we learn from this expression? When f and p are roughly proportional, then so are their derivatives f_x, f_π, p_x, and p_π. In this case, symmetries in the expression within square brackets cause it to be close to zero, which means that the differentiated sampling technique remains a good choice for the differential estimator. While that is excellent news, there are also multiple potential pitfalls involving this type of estimator.

Product integrals. In Monte Carlo rendering, the integrand is often a product of complex terms, $f = g \cdot h$, of which only one is targeted by the sampling strategy. Suppose that the sampling density p is perfectly proportional to the first term, i.e., $p = C \cdot g$ for $C \in \mathbb{R}$. Then the attached differential estimator reduces to

$$\partial_\pi I = C^{-1} \int_{\mathcal{U}} \left[h_\pi(x, \pi) + T_\pi(u, \pi)\, h_x(x, \pi) \right] du. \qquad (12)$$

This expression indicates that two properties carry over from the primal case: g is handled perfectly in the sense that no variance will arise from this term. Integration of the second term h proceeds through the parameterization $T(u, \pi)$.

In contrast to the primal case, an additional term captures the differential change of the reparameterized function $h(T(..))$. It has the potential to introduce significant variance when the parameterization T rapidly distorts h for small perturbations of π. This additional complication does not exist in the primal case. It would be tempting to mix and match, i.e., to attach the factor being sampled and detach the other term to avoid this additional source of variance. However, this generally introduces bias[2]. Figure 4 illustrates the difference between complete sampling of an integral and partial sampling based on a factor.

Attached sampling naturally handles integrands containing delta functions, which are not supported by most previous work on differentiable rendering. This case arises e.g. when computing gradients with respect to the surface normal of a perfectly specular mirror. In this situation, the product of (delta) BSDF and incident radiance simplifies to just the radiance term that will then be evaluated through the mapping T. This addresses a severe limitation of detached sampling techniques.

Discrete decisions. Sampling techniques often consume uniform variates to take discrete decisions like choosing the component of a multi-lobe BSDF. In contrast to the unified differentiation of integrand, density, and parameterization in Equation (10), the probabilities of such discrete decisions should never be handled in an *attached* manner, as doing so would introduce severe bias. For example, consider a path termination criterion such as *Russian Roulette*, which only continues the random walk with probability α, while applying a scaling correction to account for this change:

$$\mathrm{RR}(u, \alpha) = \begin{cases} 1/\alpha, & u < \alpha, \\ 0, & \text{otherwise.} \end{cases}$$

In practice, the probability α would be related to the albedo of prior scattering interaction, which introduces a dependence on the scene parameters (in the extreme case, $\alpha = \pi$). However, this expression then behaves like a parameter-dependent discontinuity that was explicitly forbidden at the beginning of this section. Handling such components using a detached estimator resolves the issue.

Creation of discontinuities. Techniques of the detached type can be used to compute unbiased estimates of discontinuous integrands if the positions of these discontinuities do not depend on scene parameters π. This common case arises e.g. when optimizing materials on static geometry. The ability to easily solve such problems despite the omnipresent visibility-induced discontinuities is a welcome simplification.

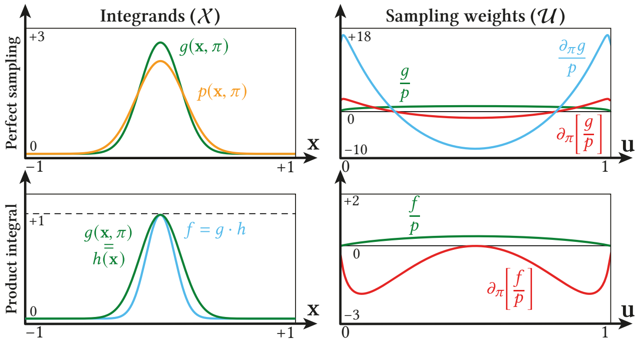

Fig. 4. 1D Examples of *attached* samplers. **Top row**: g follows a normal distribution parameterized by standard deviation $\pi = \sigma$ and is sampled with density p approximating g. Variance due to attached sampling weights (red) in $\mathcal{U}$ resembles the primal case (green), while a detached estimator (blue) performs substantially worse. **Bottom row**: Product integral of two scaled Gaussians $g = h$ sampled proportionally ($p \sim g$), where h is independent of π. Extra derivative terms involving the non-sampled factor in Equation (12) inject additional variance, causing the attached estimator (red) to perform poorly compared to an estimator of the primal product integral (green).

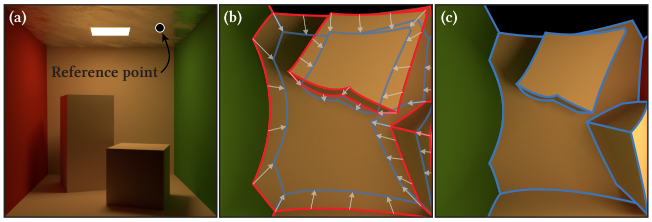

Fig. 5. The visible hemisphere from a given reference point in a Cornell box scene **(a)** is mapped through a microfacet importance sampling transform onto the unit square $\mathcal{U}$ for two different roughness values $\alpha = 0.4$ **(b)** and $\alpha = 0.6$ **(c)**. Motion vectors (white arrows) indicate how static discontinuities (red and blue lines) become dependent on α through this parameterization. The sampling routine used in this visualization targets the Beckmann microfacet distribution through a concentric disk mapping.

The previous discussion has shown that attached sampling strategies can be superior to detached ones, particularly when the former are built from high-quality primal methods. However, attempting to differentiate discontinuous integrands using such attached strategies reveals a fundamental problem: these methods warp the integrand in a parameter-dependent fashion, and this transformation will naturally also affect discontinuities. Consequently, as shown in Figure 5, discontinuities that were previously static on $\mathcal{X}$ will lose this property on the reparameterized domain $\mathcal{U}$, introducing bias. As-is, such attached techniques simply cannot be used with discontinuous integrands, which rules out most rendering-related applications. Fortunately, it is possible to address this limitation using *reparameterized attached* sampling that we present in Section 5 for the important special case of directly visible discontinuities on the unit sphere $\mathcal{X} = S^2$.

[2] A product integral with $g(x) = h(x) = x^\theta$ on $[0, 1]$ provides a simple example of this: all possible ways of attaching and detaching the terms and reciprocal probability lead to different derivative estimates for $\theta_0 = 1$.

3.3 Multiple importance sampling

Combinations of two or more sampling strategies via *multiple importance sampling* (MIS) [Veach and Guibas 1995] are widely used in Monte Carlo rendering. We consider MIS in gradient estimators given its essential role for variance reduction in the primal setting. Analogously to the choice of attaching or detaching the estimators themselves, the same decision must now be taken for their MIS weights, resulting in a 2×2 matrix of possible combinations (Sections 3.3.1 to 3.3.4).

The main benefit of attached strategies lies in their ability to consider the dependence on π for variance reduction. This is ultimately not very useful for MIS weights, where a strong dependence on π represents an unusual situation (this would mean that a perturbation of a scene parameter rapidly changes the sampling technique of choice). While MIS continues to play an important role in combining several strategies, the choice of whether to attach or detach its weights is thus largely irrelevant from the viewpoint of variance reduction. However, not all possible combinations of attached estimators and attached MIS are useful or even correct, and we now review the various possibilities from this viewpoint:

3.3.1 Detached estimators, detached MIS. Suppose that we are already working with detached estimators: in this case, it would be natural to similarly neglect the π-dependence of MIS weights during differentiation:

$$\partial_\pi I = \int_{\mathcal{X}} \sum_{i=1}^{n} w_i(\mathbf{x}_i, \pi_0) \cdot \frac{\partial_\pi f(\mathbf{x}_i, \pi)}{p_i(\mathbf{x}_i, \pi_0)}, \tag{13}$$

where n techniques with sampling transforms T_i and PDFs p_i are combined, denoting a sample drawn from strategy i as $\mathbf{x}_i = T_i(\mathbf{u}, \pi_0)$. This combination is a standard application of MIS to a particular function that happens to be a derivative, and its correctness thus follows from prior work [Veach and Guibas 1995].

3.3.2 Attached estimators, attached MIS. Alternatively, both estimators and MIS weights can be parameterized through corresponding inverse-transform mappings $T_i : \mathcal{U} \to \mathcal{X}$ to track all parameter dependencies during the differentiation process. This case is correct by definition as we are now simply looking at the derivative of the entire expression.

$$\partial_\pi I = \int_{\mathcal{U}} \sum_{i=1}^{n} \partial_\pi \left[w_i(T_i(\mathbf{u}_i, \pi), \pi) \cdot \frac{f(T_i(\mathbf{u}_i, \pi), \pi)}{p_i(T_i(\mathbf{u}_i, \pi), \pi)} \right]. \tag{14}$$

3.3.3 Attached estimators, detached MIS. In our experiments, we also considered a third logical option of combining attached estimators with detached MIS weights:

$$\partial_\pi I \overset{?}{=} \int_{\mathcal{U}} \sum_{i=1}^{n} w_i(T_i(\mathbf{u}_i, \pi_0), \pi_0) \cdot \partial_\pi \left[\frac{f(T_i(\mathbf{u}_i, \pi), \pi)}{p_i(T_i(\mathbf{u}_i, \pi), \pi)} \right]. \tag{15}$$

However, this combination can be severely biased. Differentiation of the fully attached case in Equation (14) via the product rule generates mixed terms of the form $(\partial_\pi w_i) \cdot f/p_i$ that are missing in Equation (15), and this introduces bias unless $\partial_\pi w_i = 0$ (which does not represent an interesting case).

3.3.4 Detached estimators, attached MIS. Finally, one can also attach the MIS weights of a set of detached estimators.

$$\partial_\pi I = \int_{\mathcal{U}} \sum_{i=1}^{n} \frac{\partial_\pi \left[w_i(T_i(\mathbf{u}_i, \pi_0), \pi) \cdot f(T_i(\mathbf{u}_i, \pi_0), \pi) \right]}{p_i(T_i(\mathbf{u}_i, \pi_0), \pi_0)}. \tag{16}$$

The validity of this approach follows from the correctness of the individual steps that can be used to derive it: introducing MIS, differentiating, followed by Monte Carlo importance sampling. See also the sequence of steps leading to the top left in the taxonomy in Figure 2. We mainly mention this case for completeness and have not found it to be a compelling strategy in our experiments.

3.3.5 Combining attached and detached strategies. The full set of options is even more fine-grained than the above list may suggest: mixing attached and detached estimators is also possible. The validity of this approach once more follows from the correctness of the individual steps, as highlighted in Figure 6. In some sense, this is not too surprising, as this type of combination will naturally arise if one of the strategies p_i is independent of the parameter π being differentiated.

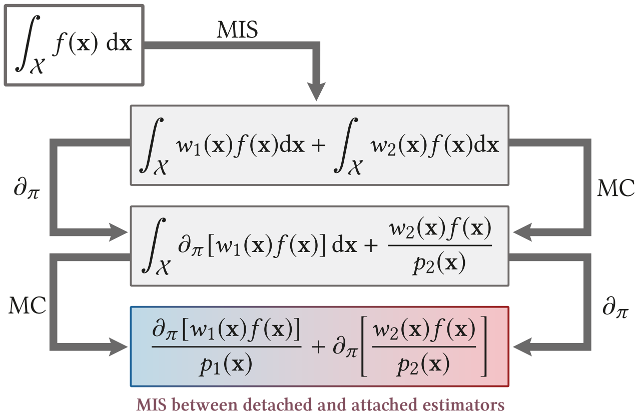
$\int_{\mathcal{X}} f(\mathbf{x})\,\mathrm{d}\mathbf{x}$
MIS
$\int_{\mathcal{X}} w_1(\mathbf{x}) f(\mathbf{x}) \mathrm{d}\mathbf{x} + \int_{\mathcal{X}} w_2(\mathbf{x}) f(\mathbf{x}) \mathrm{d}\mathbf{x}$
∂_π
MC
$\int_{\mathcal{X}} \partial_\pi [w_1(\mathbf{x}) f(\mathbf{x})]\,\mathrm{d}\mathbf{x} + \frac{w_2(\mathbf{x}) f(\mathbf{x})}{p_2(\mathbf{x})}$
MC
∂_π
$\frac{\partial_\pi [w_1(\mathbf{x}) f(\mathbf{x})]}{p_1(\mathbf{x})} + \partial_\pi \left[\frac{w_2(\mathbf{x}) f(\mathbf{x})}{p_2(\mathbf{x})} \right]$
MIS between detached and attached estimators

Fig. 6. The decision of whether to attach or detach a sampling technique and its MIS weight can be made separately for each technique, as illustrated by this derivation sketch.

3.3.6 Summary. In summary, MIS remains a helpful tool for combining sampling strategies. Not all possible combinations of attached MIS weights and estimators are useful or yield unbiased gradient estimates. Based on experimental evaluation, we recommend to either jointly attach or detach estimators and their MIS weights.

A curious thought that arises following this discussion is whether one can combine attached and detached versions of the *same* primal estimator via MIS to draw on each strategy where it performs best? This intuition from primal estimators sadly does not transfer to the differential world: MIS weights are guided by sampling probabilities, and those probabilities would be identical in such a combination (modulo minor differences in how differentiation is performed with respect to π). It will be interesting to explore extensions and generalizations of MIS that can perceive the deficiencies of a differential estimator and suitably adapt.

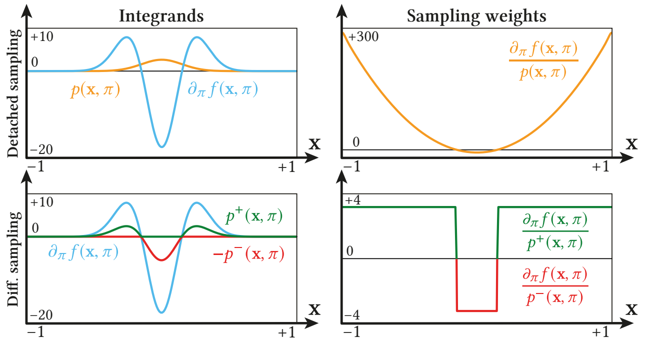

Fig. 7. 1D examples of custom *differential* samplers. **Top row**: The wrapped normal distribution p is ill-suited for sampling its derivative $\partial_\pi f = \partial_\pi p$ and produces large weights. **Bottom row**: the densities p^+ and p^- are proportional to the positive and negative regions of $\partial_\pi f$ and produce constant sampling weights that reduce the variance of the estimator.

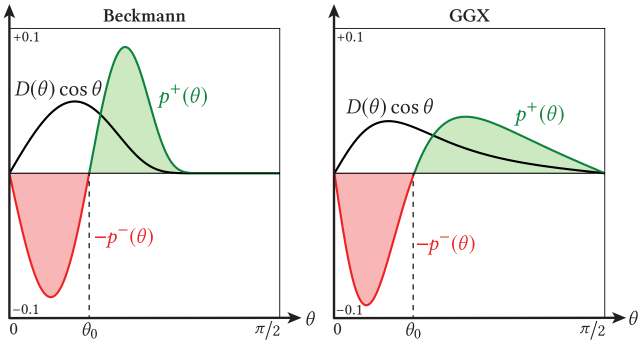

Fig. 8. Plots of two microfacet distributions (Beckmann and GGX) with roughness parameter $\alpha = 1/2$ (black). The derivative with respect to α produces a signed function with positive (green) and negative (red) lobes of equal area. Our differential microfacet sampling strategy specifically samples these two lobes to efficiently compute gradients that characterize how the transport simulation changes with respect to perturbations of α.

4 DIFFERENTIAL SAMPLING STRATEGIES

Section 1 hinted at a fascinating possibility that arises when a differentiable renderer relies on decoupled primal and differential phases: we can introduce additional strategies that are specifically designed to improve sampling of differential transport. From a high level, such a *differential sampling strategy* will involve an integral that looks identical to the detached case from Equation (9):

$$\partial_\pi I = \int_{\mathcal{U}} \frac{\partial_\pi f(T(\mathbf{u},\pi_0),\pi)}{p(T(\mathbf{u},\pi_0),\pi)}\mathrm{d}\mathbf{u}. \tag{17}$$

The key difference is that the sampling technique encoded in p and T is no longer constrained by the primal phase. Essentially anything could be used, and we can exploit this freedom to reduce variance in challenging situations. Figure 7 shows a simple 1D example of a differential sampling strategy tailored to the normal distribution.

Differential sampling strategies also address an issue that we had first observed in Equation (12), which appears when attached sampling techniques are invariably applied to product integrals that occur in rendering algorithms. Attached strategies will warp all factors, and this introduces additional derivative terms that can introduce significant variance. In contrast, the formulation in Equation (17) is static and does not suffer from this problem.

Not all scene parameters call for custom sampling strategies, however. Many material models include a directionally uniform albedo that is adequately handled by primal strategies. In contrast, scene parameters controlling surface roughness have a pronounced effect on the sampling process and constitute an example where differential strategies can make a large difference. This is apparent in Figure 1, where the differential strategy ("diff. detached") in the rightmost column generally performs best. We now discuss an example of a sampling technique targeting the family of microfacet BRDFs.

4.1 Differential microfacet sampling

Microfacet distributions [Torrance and Sparrow 1967; Cook and Torrance 1982] are integral building blocks of many widely used reflectance models. In numerical experiments, we found that derivatives with respect to their roughness parameter were characterized by severe variance. Figure 3 highlights the fundamental problem: changes in the shape of the integrand break detached estimators. On the other hand, attached estimators applied to a product integral with material and lighting terms tend to perform poorly when the parameter-dependent warp distorts the incident radiance function. We leverage the freedom of a decoupled differential transport simulation to introduce a specialized differential sampling strategy that will address these challenges.

Consider the derivative of an isotropic microfacet BRDF [Walter et al. 2007] with respect to its roughness parameter α:

$$\partial_\alpha f_s(\boldsymbol{\omega},\boldsymbol{\omega}',\alpha) = \frac{F(\boldsymbol{\omega},\boldsymbol{\omega}_h)}{4\cos\theta\cos\theta'}\partial_\alpha\left[D(\boldsymbol{\omega}_h,\alpha)\cdot G(\boldsymbol{\omega},\boldsymbol{\omega}',\boldsymbol{\omega}_h,\alpha)\right]. \tag{18}$$

Here, F refers to the Fresnel term (which does not depend on α), D is the microfacet distribution, G is the shadowing-masking term, and ω_h denotes the half-direction vector between ω and ω'. We ignore the derivative in G as it only has a minor effect on the directional distribution and focus on the microfacet distribution D, limiting our discussion to two isotropic models by Beckmann and Spizzichino [1987] and Trowbridge and Reitz [1975]. In computer graphics, the latter is also known as the GGX distribution [Walter et al. 2007]. In spherical coordinates, these two distributions are defined as

$$D_{\mathrm{GGX}}(\theta,\alpha)\cos\theta = \frac{2\alpha^2\sin\theta}{\cos^3\theta(\alpha^2+\tan^2\theta)^2}, \tag{19}$$

$$D_{\mathrm{Beck.}}(\theta,\alpha)\cos\theta = \frac{2e^{\frac{-\tan^2\theta}{\alpha^2}}\sin\theta}{\alpha^2\cos^3\theta}, \tag{20}$$

where the cosine term on the left side is required for normalization. Following differentiation, the function splits into a positive and negative lobe of equal area (Figure 8) with a zero crossing at $\theta_0 = \tan^{-1}\alpha$. Our goal is to construct a method that samples proportionally to

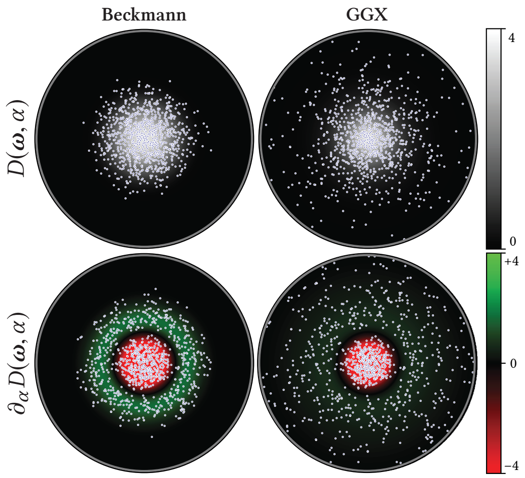

Fig. 9. Visualization of samples produced by the *Beckmann* and *GGX* (top) and *dBeckmann* and *dGGX* (bottom) sampling techniques for $\alpha = 1/2$. Positive and negative lobes are highlighted in red and green.

the absolute value of $\hat{p} = \partial_\alpha \left[D(\theta, \alpha) \cdot \cos\theta\right]$, where

$$\hat{p}_{\mathrm{dGGX}}(\theta) = \frac{4\alpha^2 \tan\theta(\tan^2\theta - \alpha^2)}{\cos^2\theta(\alpha^2 + \tan^2\theta)^3}, \tag{21}$$

$$\hat{p}_{\mathrm{dBeck.}}(\theta) = \frac{2e^{1-\frac{\tan^2\theta}{\alpha^2}} \tan\theta(\tan^2\theta - \alpha^2)}{\alpha^4 \cos^2\theta}. \tag{22}$$

At this point, it is important to observe that Monte Carlo importance sampling produces zero variance when the sampling density is perfectly proportional to a non-negative integrand. This property no longer holds for signed integrands—in the worst case, sign-related variance can fully negate the benefits of tailored importance sampling strategies even if they match the integrand in an absolute sense. Multiple techniques exist to handle such cases [Owen 2013]. We rely on *antithetic sampling* to generate paired and correlated samples from the two lobes. This involves two rendering passes using the same random generator state that are finally averaged. Details about the necessary inverse transform mapping can be found in Appendix A. Figure 9 contrasts samples drawn from primal and differential microfacet distributions.

One limitation of differential BSDF sampling strategies is that they require an incident illumination estimate for the newly sampled direction, which must be computed using recursive path tracing or an alternative primal algorithm. Regular BSDF sampling remains necessary to scatter the adjoint radiance to other parts of the scene, which propagates like normal light[3]. These branching random walks cause the method to have a quadratic time complexity as a function of path length, which is also a limitation of the original (unbiased) RB algorithm.

Other considerations. When the differentiation problem involves parameters beyond surface roughness, the differential transport simulation must also incorporate detached BSDF sampling to obtain low-variance estimates of the associated derivatives. These two strategies can then be combined via detached MIS. The previous derivation is not specific to the reflective case and also enables differential sampling of rough transmission. We did not investigate generalizations to more advanced models with anisotropy and vNDF sampling [Heitz and D'Eon 2014] and consider them beyond the scope of this article.

Combinations of differential and detached/attached strategies via MIS are also possible: our experiments for example combine differential sampling techniques with standard emitter sampling.

We mention for completeness that differential strategies could also be attached, which would entail tracking derivatives of T and p with respect to π in Equation (17). Attaching was of crucial importance when we were restricted to working with primal sampling techniques, but it is of limited use here as T and p can be arbitrarily chosen. Detached strategies can also directly handle integrands with static discontinuities, for instance when optimizing the materials of a scene with fixed geometry. No special handling of moving discontinuities (Section 5) is required in that case, which is beneficial as this comes at considerable additional runtime cost.

5 REPARAMETERIZING DISCONTINUOUS INTEGRANDS

We finally turn to the case of integrands containing discontinuities, whose position furthermore depends on π. To do so, we adopt the high-level framework of Loubet et al. [2019] and transform the integrals using a recently proposed parameterization by Bangaru et al. [2020]. This counteracts the motion of discontinuities, so that the tools from Section 3 are readily applicable.

However, both prior methods by Loubet et al. and Bangaru et al. are designed to work in a context where the primal and differential phases are rigidly coupled via reverse-mode differentiation. This means that these methods suffer from severe overheads to store primal program variables that are later needed for differentiation. In contrast, our work operates within the decoupled framework of *radiative backpropagation* (RB) [Nimier-David et al. 2020], which turns the differentiation into an independent simulation that transports derivative radiation from sensors to differentiable objects.

The goal of this section is to clarify how reparameterization-based techniques can be cast into a suitable form to enable their use within such a differential transport simulation. We will also revisit the case of attached samplers to finally address a severe limitation with moving discontinuities encountered in Section 3.2.

Background. We briefly review relevant background material from the works of Loubet et al. [2019] and Bangaru et al. [2020] and refer to these papers for an in-depth discussion of the theory and relevant mappings that we import without modifications.

Suppose that the following integral must be differentiated with respect to a scene parameter π that influences the position of discontinuities. Examples of such parameters include camera pose, object transformations, vertex positions, etc. To avoid biased estimates, we must then perform a change of variables

$$\partial_\pi I = \partial_\pi \int_{\mathcal{X}} f(\mathbf{x}, \pi)\,\mathrm{d}\mathbf{x} = \partial_\pi \int_{\mathcal{X}} f(R(\mathbf{x}, \pi), \pi) \cdot |J_R(\mathbf{x}, \pi)|\,\mathrm{d}\mathbf{x} \tag{23}$$

to freeze the discontinuities with respect to perturbations of π. Here, $R : \mathcal{X} \times \Pi \to \mathcal{X}$ is a *(re-)* parameterization of the domain $\mathcal{X}$ that

[3] For clarification, we refer to Section 5.1 that re-derives RB. In particular, indirect propagation occurs via term (T2), and differential strategies target term (T3).

must necessarily depend on the scene parameters to accomplish this task. Both prior works performed the parameterization on the unit sphere $\mathcal{X} = S^2$, in which case $|J_R|$ denotes the Jacobian determinant for a given π that counteracts the change in spherical area due to this mapping. Following these changes, it is legal to move the derivative into the integral. Note that some operations involving this parameterization (e.g. evaluating Jacobian determinants) implicitly assume that the underlying spherical domain is accessed using suitable 2-dimensional coordinates, as the ambient 3D space is too high-dimensional. We rely on spherical coordinates for this purpose.

We use the letter R to contrast with the other kind of parameterization T encountered in Section 3.1, which was used for importance sampling via the inverse transform method. We will for now ignore the additional complication of importance sampling and revisit this case in Section 5.2.

Similarly to Loubet et al., our proposed parameterization satisfies $R(\mathbf{x}, \pi_0) = \mathbf{x}$, which ensures that its introduction does not affect the primal simulation. Here, π_0 refers to the detached scene parameters. Furthermore, $\partial_\pi R(\mathbf{x}, \pi) = \partial_\pi P(\mathbf{x}, \pi)$ where $P : \mathcal{X} \times \Pi \to \mathcal{X}$ returns a position, whose *velocity* $\partial_\pi P$ must be carefully chosen so that the parameterization can accomplish its goal. The equations above are easily satisfied if R is defined as

$$R(\mathbf{x}, \pi) = \mathbf{x} + P(\mathbf{x}, \pi) - P(\mathbf{x}, \pi_0). \tag{24}$$

However, a suitable function P must still be chosen. In particular, P must be a smooth function that satisfies the following critical requirement: as $\mathbf{x}$ approaches another point $\mathbf{x}_b(\pi)$ that is located on a π-dependent discontinuity, the velocity $\partial_\pi P(\mathbf{x}, \pi)$ *must* tend to $\partial_\pi \mathbf{x}_b(\pi)$.

The specific function P proposed by Loubet et al. is approximate and can introduce bias. We therefore follow the improved approach of Bangaru et al., who define P in terms of a spherical convolution of a base position $\bar{P}$ and a weighting kernel w requiring normalization through an additional integral in the denominator:

$$P(\mathbf{x}, \pi) = \frac{\int_{\mathcal{X}} w(\mathbf{x}, \mathbf{x}')\, \bar{P}(\mathbf{x}', \pi)\, \mathrm{d}\mathbf{x}'}{\int_{\mathcal{X}} w(\mathbf{x}, \mathbf{x}')\, \mathrm{d}\mathbf{x}'}. \tag{25}$$

The function $\bar{P}(\mathbf{x}, \pi)$ encapsulates the steps of tracing a ray into the associated direction $\mathbf{x}$ and returning the intersection position in spherical coordinates. This computed value is not particularly interesting, since the projection of the intersection onto the sphere is simply the original ray direction. However, the π-derivative of this value will now capture the object motion. These attached directions are then convolved via Equation (25), whose weights w become extremely large as $\mathbf{x}$ and $\mathbf{x}'$ approach a silhouette. This furthermore occurs in a "unidirectional" fashion to ensure that the final result $P(\mathbf{x}, \pi)$ follows the motion of the occluder as $\mathbf{x} \to \mathbf{x}_b$, while ignoring the occluded object. The weights use a *boundary test* function $\bar{B}$ to query approximate distances to the visible edges, e.g., based on shading normal dot products. Both integrals in Equation (25), are evaluated using Monte Carlo sampling and must use the same set of samples to reduce variance to an acceptable level. We refer to Bangaru et al. [2020] for further discussion and complete definitions of all quantities.

5.1 Reparameterized radiative backpropagation

The derivation of RB [Nimier-David et al. 2020] begins with the differential forms of the three equations that jointly define the problem solved by any rendering algorithm: scattering, transport, and measurement. We only cover the case of scattering, which is representative of the needed changes. This involves the rendering equation relating outgoing to incident illumination using the BSDF f_s. When reparameterized using R, its primal form reads

$$\begin{aligned} L_\mathrm{o}(\mathbf{p}, \boldsymbol{\omega}) &= L_\mathrm{e}(\mathbf{p}, \boldsymbol{\omega}) \\ &\quad + \int_{S^2} L_\mathrm{i}(\mathbf{p}, R(\boldsymbol{\omega}', \pi))\, f_s(\boldsymbol{\omega}, R(\boldsymbol{\omega}', \pi))\, |J_R(\mathbf{x}, \pi)|\, \mathrm{d}\boldsymbol{\omega}'^{\perp}. \end{aligned} \tag{26}$$

Since the discontinuities are now static, it is legal to differentiate under the integral sign. Application of the product rule then produces a total of four derivative terms:

$$\begin{aligned} \partial_\pi L_\mathrm{o}(\mathbf{p}, \boldsymbol{\omega}) &= \partial_\pi L_\mathrm{e}(\mathbf{p}, \boldsymbol{\omega}) && \text{(T1)} \\ &\quad + \int_{S^2} \partial_\pi L_\mathrm{i}(\mathbf{p}, R(\boldsymbol{\omega}', \pi))\, f_s(\boldsymbol{\omega}, \boldsymbol{\omega}')\, \mathrm{d}\boldsymbol{\omega}'^{\perp} && \text{(T2)} \\ &\quad + \int_{S^2} L_\mathrm{i}(\mathbf{p}, \boldsymbol{\omega}')\, \partial_\pi f_s(\boldsymbol{\omega}, R(\boldsymbol{\omega}', \pi))\, \mathrm{d}\boldsymbol{\omega}'^{\perp} && \text{(T3)} \\ &\quad + \int_{S^2} L_\mathrm{i}(\mathbf{p}, \boldsymbol{\omega}')\, f_s(\boldsymbol{\omega}, \boldsymbol{\omega}')\, \partial_\pi |J_R(\mathbf{x}, \pi)|\, \mathrm{d}\boldsymbol{\omega}'^{\perp}. && \text{(T4)} \end{aligned} \tag{27}$$

The above expression has been simplified by noting that R and its Jacobian reduce to the identity when they occur in a term that is not differentiated. Intuitively, this equation states that the process of differentiation can be modeled by simulating scattering, transport, and eventual measurement of a hypothetical "differential radiance" quantified by $\partial_\pi L_\mathrm{i}$ and $\partial_\pi L_\mathrm{o}$ (there is one such function per scene parameter π). Equation (27) takes the role of an energy balance equation that indicates the following properties that we discuss out of order:

T1. Differential radiance is emitted when the primal emission L_e depends on π.
T3. Objects, whose material model depends on the parameter π, convert some of the ordinary radiance incident on the surface (L_i) into differential radiance ($\partial_\pi L_\mathrm{o}$).
T4. Differential radiance is also added or subtracted when the parameterization R expands or contracts space depending on π.
T2. Finally, once created, differential radiance scatters like ordinary light (i.e. involving the BSDF of scene objects).

In contrast to RB without reparameterization, (T4) is new and all terms are now at least partially warped by the parameterization.

Bangaru et al. [2020] relate the parameterization's Jacobian determinant to a vector field divergence. We use this relationship as the divergence can be cheaply approximated together with Equation (25). After reordering terms,

$$\partial_\pi |J_R(\mathbf{x}, \pi)| = \nabla_{\boldsymbol{\omega}} \cdot \partial_\pi R = \partial_\pi \left(\nabla_{\boldsymbol{\omega}} \cdot R \right), \tag{28}$$

the divergence in parentheses can be handled analytically, producing derivatives of the weighting kernel w. We perform the outer derivative using reverse-mode AD, which will then backpropagate gradients to the scene geometry. At this point, Equation (27) could in principle be solved separately for scene parameter π to compute $\partial_\pi L_\mathrm{i}$ at the camera, resulting in a *derivative image*. This approach

```
def reparam_rb($\pi$, $\boldsymbol{\delta}_y$):
    # Initialize parameter gradient(s) to zero
    $\boldsymbol{\delta}_\pi = 0$
    for _ in range(num_samples):
        # Sample ray proportional to sensor response and pixel filter
        ($\mathbf{p}$, $\boldsymbol{\omega}$), s_val, s_pdf = sensor.sample()
        # Query adjoint emitted radiance associated with current ray
        A_e = $A_e(\boldsymbol{\delta}_y, p, \omega)$ / num_samples
        # Backpropagate through the reparameterized pixel filter
        $\boldsymbol{\delta}_\pi$ += adjoint([[ sensor.eval($\mathbf{p}$, $R(\boldsymbol{\omega}, \pi)$) ]],
                         A_e * $L_\mathrm{i}(\mathbf{p}, \boldsymbol{\omega})$ / s_pdf)
        # Backpropagate through the divergence
        $\boldsymbol{\delta}_\pi$ += adjoint([[ $\nabla_{\boldsymbol{\omega}} \cdot R(\boldsymbol{\omega}, \pi)$ ]],
                         A_e * s_val * $L_\mathrm{i}(\mathbf{p}, \boldsymbol{\omega})$ / s_pdf)
        # Propagate adjoint radiance into the scene
        $\boldsymbol{\delta}_\pi$ += reparam_rb_Li($\pi$, $\mathbf{p}$, $R(\boldsymbol{\omega}, \pi)$, A_e * s_val / s_pdf)
    # Finished, return gradients
    return $\boldsymbol{\delta}_\pi$
```

Listing 1. Reparameterized radiative backpropagation takes scene parameters π and an adjoint rendering $\boldsymbol{\delta}_y$ as input. This pseudocode fragment is responsible for the measurement integral. It samples a set of sensor rays, queries the associated emitted adjoint radiance A_e and propagates these gradients into the scene, while accounting for geometric discontinuities.

does not scale to scenes with high-dimensional parameter spaces, as millions of derivative images would potentially need to be rendered per gradient descent step.

RB exploits the reciprocal nature of this problem and transports derivatives in the opposite direction, i.e., from the camera towards scene objects. The radiation emanating from the camera in this phase is a signed quantity ("*adjoint radiance*") that specifies how the rendered image should change to optimally improve the optimization objective. Only a single transport problem needs to be solved in this case, which is substantially more efficient than the naïve approach mentioned above. Once the adjoint radiance reaches a specific surface location, it must still be converted into a scene parameter gradient. Here, it is useful to observe that a point is generally only characterized by a few local parameters, such as the positions of surrounding vertices, texels referenced by a texture lookup in a shader, etc. Whereas path tracing performs a random walk that *reads* such local surface properties, RB then performs an analogous random walk that *writes* local gradients at every interaction.

Listings 1 and 2 provide the pseudocode of the reparameterized RB method and can be contrasted to similar code fragments in the original paper [Nimier-David et al. 2020]. We reuse their notation `adjoint([[ <expr> ]], grad_out)` to refer to the reverse-mode derivative of the expression `<expr>` that backpropagates a gradient with respect to its output (`grad_out`) towards the scene parameters $\boldsymbol{\pi}$, returning another gradient resulting from this step.

5.2 Reparameterizing attached strategies

As discussed earlier in Section 3.2, attached sampling provides a convenient way to reuse a primal sampling procedure in a differential estimator. Accounting for its parameter dependence during differentiation can be an effective variance reduction strategy. Unfortunately, the parameter-dependent change of variables also causes previously static discontinuities to move with respect to perturbations in π (see Figure 5), which severely limits the utility of this approach.

```
def reparam_rb_Li($\pi$, $\mathbf{p}$, $\boldsymbol{\omega}$, $\delta L$):
    # Find an intersection with the scene geometry
    $\mathbf{p}'$ = $r(\mathbf{p}, \boldsymbol{\omega})$
    # T1: Backpropagate through reparameterized emitters, if any
    $\boldsymbol{\delta}_\pi$ = adjoint([[ $L_\mathrm{e}(\mathbf{p}', -\boldsymbol{\omega})$ ]], $\delta L$)
    # Sample a direction from the BSDF
    $\boldsymbol{\omega}'$, b_val, b_pdf = sample $f_s(\mathbf{p}', -\boldsymbol{\omega}, \cdot)$
    # T3: Backpropagate through the reparameterized BSDF
    $\boldsymbol{\delta}_\pi$ += adjoint([[ $f_s(\mathbf{p}', -\boldsymbol{\omega}, R(\boldsymbol{\omega}', \pi))$ ]],
                   $\delta L$ * $L_\mathrm{i}(\mathbf{p}, \boldsymbol{\omega}')$ / b_pdf)
    # T4: Backpropagate through the divergence
    $\boldsymbol{\delta}_\pi$ += adjoint([[ $\nabla_{\boldsymbol{\omega}'} \cdot R(\boldsymbol{\omega}', \pi)$ ]],
                   $\delta L$ * b_val * $L_\mathrm{i}(\mathbf{p}, \boldsymbol{\omega}')$ / b_pdf)
    # T2: Recurse to account for indirect differential radiance
    return $\boldsymbol{\delta}_\pi$ + reparam_rb_Li($\pi$, $\mathbf{p}'$, $R(\boldsymbol{\omega}', \pi)$, $\delta L$ * b_val / b_pdf)
```

Listing 2. This pseudocode fragment provides `reparam_rb_Li()` referenced in Listing 1. It implements the reparameterized RB version of the transport and scattering equation that transports derivatives through the scene and backpropagates adjoint radiance δL to objects with differentiable parameters.

To address these challenges, we define a secondary parameterization analogous to Equation (24), which will similarly counteract the movement of the samples to prevent issues with bias due to moving discontinuities:

$$R(\mathbf{x}, \pi) = \mathbf{x} - B(\mathbf{x})\, T(\mathbf{u}, \pi) + B(\mathbf{x})\, T(\mathbf{u}, \pi_0) \tag{29}$$

In the equation above, $B(\mathbf{x})$ can be interpreted as a smooth scaling factor used to slow down the samples movement as they approach discontinuities. Interestingly, setting a constant value $B(\mathbf{x}) = 1\, \forall\, \mathbf{x}$ would freeze *all* sample movement resulting in the detached estimator. Based on this observation, we will use $B(\mathbf{x}_b)$ to transition back to a detached estimator near the discontinuities where the attached strategies are troublesome. We found the boundary test $\bar{B}$ from Bangaru et al. [2020] to work well for this purpose when turned into a smoothed interpolant using Equation (25).

As this new type of reparameterization is based on similar principles as prior work [Loubet et al. 2019; Bangaru et al. 2020], it also inherits its limitations. In particular it is only defined for integration over the unit sphere $\mathcal{X} = S^2$ and is not sufficient for attached strategies that involve differentiation through multiple scattering events at once. To ensure the correctness of the computed gradients in our implementation, we thus split the integrand in Equation (26) into two parts. The indirect illumination component of L_i is handled analogously to the previous section and uses a detached estimator. The direct component however involves a nested reparameterization where R now counteracts both types of discontinuities discussed in this section. Reverse-mode differentiation will then automatically propagate gradients through both parameterizations. Finally, we can apply the attached estimator from Equation (10) to the integral. We call this approach the *reparameterized attached* strategy, and an example comparison to "naïve" attaching can be seen in Figure 10.

6 RESULTS

We evaluated our methods experimentally in a differentiable rendering system based on Mitsuba 2 [Nimier-David et al. 2019]. All experiments were performed on an NVIDIA TITAN RTX graphics card (23 GiB of RAM) using OptiX 7.2 [Parker et al. 2010] for hardware-accelerated ray tracing.

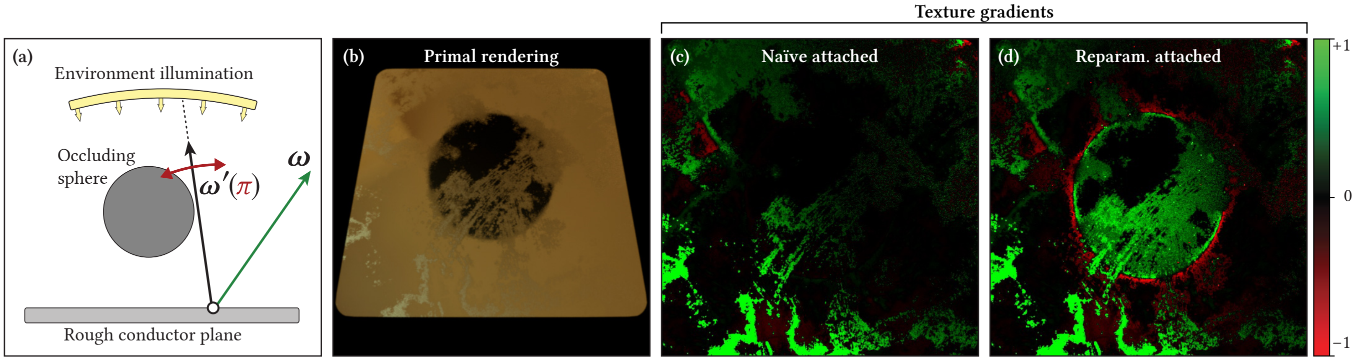

Fig. 10. We differentiate pixel intensity w.r.t. the textured roughness π of a metal surface. A reflected sphere occludes the light source **(a, b)**, causing problems in attached sampling strategies: sampled directions $\boldsymbol{\omega}'$ depend on π, which creates a parameter-dependent discontinuity in the integration domain. Consequently, naïve attached sampling misses important gradients at the reflected silhouettes **(c)** which our reparameterized attached strategy can capture correctly **(d)**.

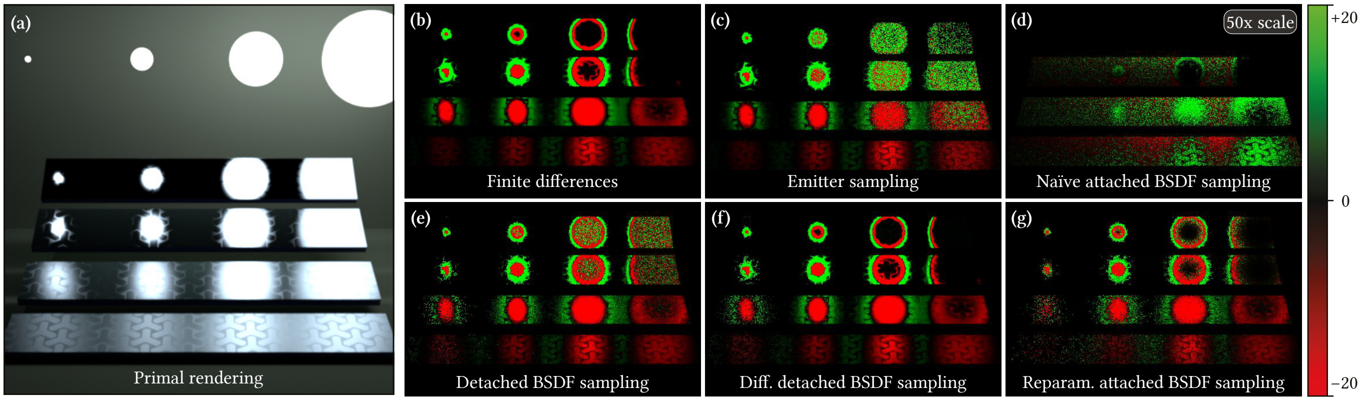

Fig. 11. This equal-time comparison based on a classic scene by Veach and Guibas illustrates the performance of several estimators described in this article. We differentiate the roughness textures of the metal plates; the average Beckmann roughness increases from top to bottom ($\alpha_{\text{avg}} \in \{0.01, 0.02, 0.06, 0.13\}$). **(a)** Primal rendering of the scene. **(b)** Ground-truth gradients computed using finite differences at a high sample count. **(c)** Gradients computed using emitter sampling. **(d)** As in Figure 10, naïve attached BSDF sampling exhibits bias due to parameter-dependent discontinuities. We had to scale the gradients of this technique by a factor of 50× so that they are visible. **(e-g)**: Three unbiased estimators using detached, differential detached, and reparameterized attached BSDF strategies. The supplemental material contains additional results using multiple importance sampling, along with variance visualizations.

6.1 Variance analysis

We now turn to concrete example scenes to analyze the statistical behavior of several differential estimators presented in this paper. Figure 1 showcases a complex scene with many glossy interreflections involving varying degrees of roughness and complex illumination from a combination of area lights and an environment map. Figure 11 represents a more controlled test setup based on a scene by Veach and Guibas [1995] with single-bounce glossy reflections for varying roughness values and light source sizes. In both cases we compute gradient images with respect to a single value π that is added to all roughness parameters in the scene. In other words, they illustrate what happens to the renderings when all glossy objects are roughened slightly. Like in primal rendering, the efficiency of estimators depends also on the concrete values of the (differentiated) scene parameters so this allows us to assess the variance at various levels of roughness at once.

Like in primal rendering, (detached) emitter sampling is good at handling concentrated illumination reflected by relatively rough surfaces. In contrast, the analysis of material-based differential estimators is more nuanced. While weaknesses of individual strategies mostly carry over into their differentiated versions, the same is not the case for their strengths. For instance, the effectiveness of detached BSDF sampling on highly specular materials is greatly reduced compared to its primal counterpart, which occurs due to mismatches between integrand and sampling density (Section 3.1). The differential sampling strategy for microfacet BSDFs from Section 4 is generally the most robust in these tests. None of the discussed strategies is specifically designed to handle chains with multiple glossy interactions, and variance is consequently high in such image regions. Attached sampling (Section 3.2) is most complex in terms of differentiation, since it must also consider the parameter dependence of samples. If done naïvely, this dependence can introduce discontinuities that can add bias (Figure 11). Our reparameterized attached strategy (Section 5.2) avoids this bias and at times achieves significant improvements over detached BSDF sampling. Its effectiveness in Figure 1 is held back by limitations regarding product sampling between BSDF and incident illumination discussed in Section 3.2.

The supplemental material contains a more complete comparison, including the use of multiple importance sampling, gradient and full-size images from Figure 1, and variance visualizations for Figure 11.

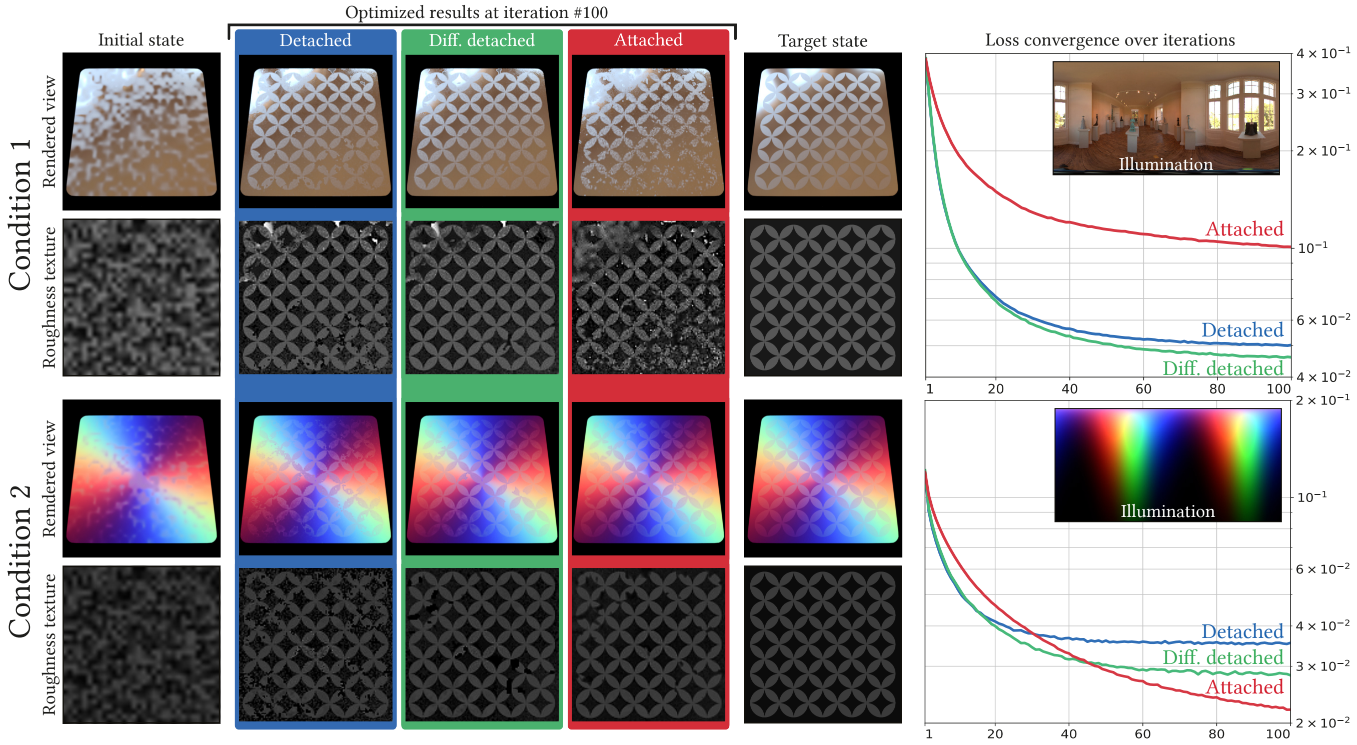

Fig. 12. Convergence comparison of various BSDF sampling techniques in the context of roughness texture optimization. The statistical efficiencies when computing gradients using the various estimators (detached, differential detached, and attached) also manifest themselves in terms of varying convergence rates when using them during optimization based on stochastic gradient descent. Differential detached sampling (**green**) performs robustly in different settings and always outperforms the detached method (**blue**) that relies on less efficient, primal microfacet sampling in this case. Attached sampling (**red**) can outperform both detached variants in cases where incident illumination is smooth (**bottom**) but can be inefficient in case of more complex illumination (**top**) due to its limitations involving product integrals. An animated version including all intermediate states is shown in the supplemental video.

6.2 Optimizing spatially varying roughness

Figure 12 compares the convergence of different estimators in a simple optimization task. We optimize the spatially varying roughness of a flat surface using standard stochastic gradient descent and a single view. We compare three estimators (detached, differential, and attached BSDF sampling) and two conditions: a rough microfacet (GGX) material under natural environment illumination, and a more specular microfacet (Beckmann) material under smooth synthetic directional illumination (three colored light sources with emission profiles modeled by spherical von Mises-Fisher distributions). Reparameterization is not needed here due to the lack of discontinuities.

All three methods compute the correct gradients in expectation, and they generate samples in a comparable amount of time. Therefore, the main distinguishing factor is their variance and the resulting impact on convergence speed. In all cases, we begin with a randomly initialized texture and run 100 iterations of SGD with a fixed learning rate. The observed convergence behavior is unsurprising and matches our previous observations on the variance of specific estimators: detached sampling handles rough reflections relatively well, but is clearly outperformed by the other methods in the more specular setting. Attached sampling based on the BSDF is expected to perform poorly when the reparameterization warps another factor with significant variation like the interior environment map. Nonetheless, this approach actually performs best in the second setting with smooth illumination. Differential sampling is robust in both cases and always outperforms detached sampling.

6.3 Efficient differentiation of geometric discontinuities

We compare our approach to geometric discontinuities to the method of Loubet et al. [2019]. The differential evaluation of this prior work was based on conventional AD and therefore rigidly coupled to the primal computation. Both steps proceeded via *wavefronts*, where one or more computational kernels were launched per scattering event, exchanging intermediate state via global memory. However, both reverse-mode AD and wavefront-style execution come at the cost of severe storage requirements that are proportional to both scene complexity (path length) and rendering quality (resolution, samples per pixel). Once the available GPU memory is exhausted, the computation must be split into multiple passes to curtail memory usage, which tends to further increase the cost of gradient evaluation.

The technique described in Section 5 improves upon this in two ways: first, it incorporates Loubet et al.'s [2019] change of variables into a differential light transport simulation that propagates the derivative of received radiance in reverse mode. This breaks the rigid coupling between primal and differential phases and thus also the need to memorize primal program variables associated with

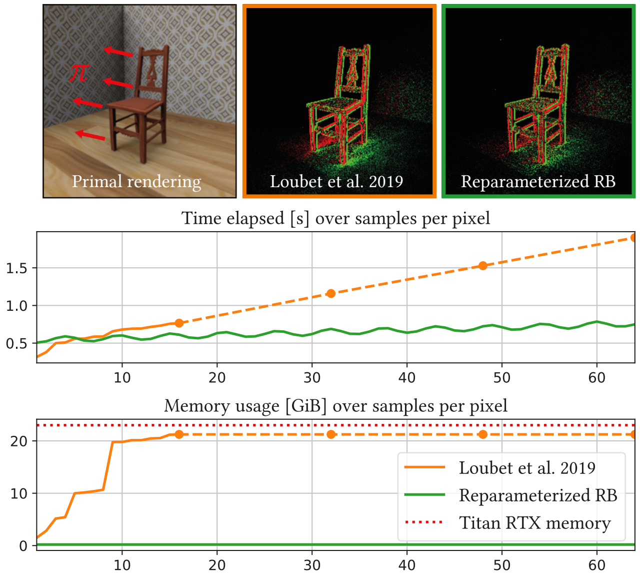

Fig. 13. Performance comparison between reparameterized RB (**green**) and the method of Loubet et al. [2019] (**orange**). When computing image gradients with respect to translation of the chair, both methods produce gradients with roughly equivalent variance, but involving substantially different time and memory requirements. Loubet et al.'s method is based on wavefront-style evaluation that saturates the 23 GiB of VRAM of a Titan RTX card at a resolution of 256^2 with 16 samples per pixel. Beyond this point, computation needs to be split into multiple passes or crops, causing a steeper increase in computation time (dashed lines). The adjoint approach of reparameterized RB is compatible with a more efficient megakernel-style evaluation with minimal memory requirements and improved runtime cost.

each scattering interaction. The second improvement comes as an immediate corollary and matches a corresponding step in prior work [Nimier-David et al. 2020]: casting the differentiation task as a transport simulation enables the implementation of the method using *megakernel*-style evaluation that finally removes all need to maintain large memory regions for intermediate program state.

The benchmark in Figure 13 compares our method's computation time and memory usage to the method of Loubet et al. using their reference implementation. The test scene exhibits low complexity, and the path length was limited to only two bounces. Still, memory usage of the method of Loubet et al. easily exceeds the total capacity of the used NVIDIA TITAN RTX graphics card (23 GiB) at low image resolution and sample count. Increasing either quality knob then requires rendering in passes, which has an adverse effect on the computation time. The memory usage of our method is independent of resolution and sample count as no primal simulation variables must be stored, and this leads to improved scalability in such cases.

We now demonstrate two prototypical optimization tasks performed using this method. The convergence behavior over time is visualized as part of the supplemental video.

6.3.1 Geometry and shading optimization. Figure 14 showcases the joint reconstruction of shape and material from a set of target images, using our reparameterized RB algorithm with detached BSDF sampling (all materials are diffuse, hence attached and detached strategies coincide). The reference is synthetic and rendered from 5 surrounding viewpoints, and we furthermore validate the resulting reconstruction using a hold-out viewpoint placed on top of the target. We parameterize the object using a displaced ellipsoid base mesh (64^2 displacement texture) and a diffuse albedo texture (1024^2 pixels), which requires simultaneous differentiation with respect to more than one million parameters. For this to be feasible within a realistic amount of time, reverse-mode differentiation is key. We have found the relative L1 loss function to be well-suited for this challenge as it focuses evenly on all regions of the images despite different brightness levels. To avoid convergence to local minima during the optimization, we use a multi-resolution scheme where the optimized textures start out at low resolution and are gradually upsampled to their target sizes throughout the process.

6.3.2 Camera and light source pose estimation. Another interesting application of inverse rendering is the problem of 6D pose estimation, see Figure 15. In this example we determine the position and orientation of the camera in a target image, as well as the location of a spherical area light. The scene used in this experiment exhibits complex effects like glossy interreflections, soft shadows, and global illumination which are all naturally handled using physically based methods but would be challenging for differentiable rasterizers. Our method reparameterizes all ray directions to avoid discontinuities from the moving light source. Recall from Section 5 that this step takes derivatives of surface intersections that were previously projected onto the unit sphere. Unsurprisingly, the moving camera causes a similar type of discontinuity requiring another reparameterization. It uses a slightly modified projection step, in which the camera origin and direction are affected by the differentiated scene parameters instead of the intersected surface position.

7 CONCLUSION

Differentiable Monte Carlo rendering provides a powerful new instrument in the pursuit of complex visual inverse problems in computer graphics and beyond.

The initial problem definition is easily stated: one must simply evaluate the derivative of an estimator. Yet, pursuing the path of this harmless differential leads to an astonishing proliferation of estimators, parameterizations, and parameterizations of parameterizations, revealing that we must now revisit many previously well-understood aspects of rendering in a different light.

Our work represents a first survey of the large space of differential Monte Carlo transport estimators. Many specimens encountered by this exploration were still fundamentally based on an underlying primal algorithm, although we show that specialized differential estimators hold significant promise in improving the efficiency of differentiable rendering in the future. Many other directions are conceivable: we envision next event estimators for emitted differential radiance and differential path guiding. At the same time, our analysis shows that intuition from the primal world may not always transfer.

Discontinuous integrands remain a bothersome element of differentiable rendering. Our work shows how suitable reparameterizations can be integrated into an efficient adjoint method, enabling geometric optimization of scenes with vast numbers of parameters and essentially no memory overheads. On the flipside, these

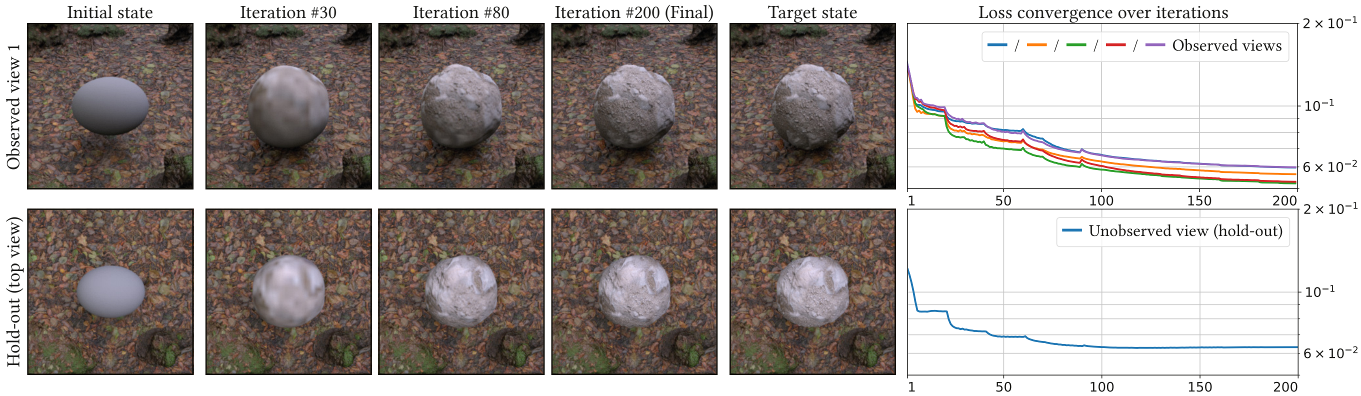

Fig. 14. Joint optimization of a displacement map and a diffuse texture. More than a million parameters are optimized simultaneously (1024^2 diffuse texture, 64^2 displacement texture) for five different view points scattered around the object. **Top**: Subsequent states of the target object during the optimization. **Bottom**: The same states from a hold-out point of view, looking at the object from the top. **Right**: Convergence rates for both optimized and unoptimized view points. An animated version including all optimized views can be seen in the supplemental video.

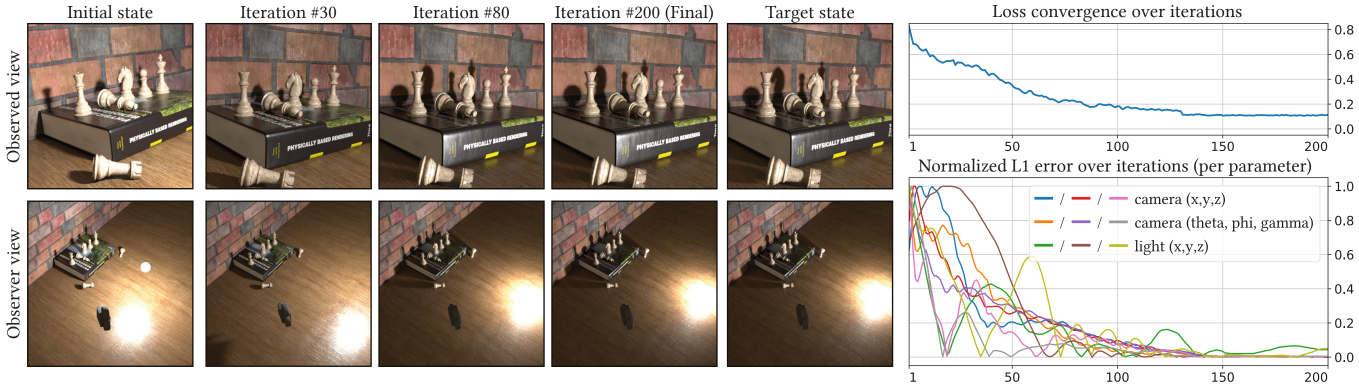

Fig. 15. 6D camera pose and light position estimation in a scene with complex light transport involving soft shadows and glossy interreflections. **Top**: Camera view at different steps during optimization. **Bottom**: Visualization of the scene from the top at the same steps (camera shown as an actual object). **Right**: Convergence of the loss and error plots of the 9 parameters that are jointly optimized. The supplemental video shows an animated version of this result.

mappings increase the cost of differentiable rendering considerably, and their stochastic nature can inject extra variance into otherwise benign integrals.

Multiple importance sampling for differentiable estimators remains highly useful, but other aspects of it are still poorly understood: differentiable estimators are potentially much worse than their primal analogues, and this is not "perceived" by MIS weights that are based on primal probabilities. Developing a truly differential form of MIS that transfers the optimality will be a promising avenue of future research. Another current issue entails chains of specular interactions that may each introduce a separate set of discontinuities. Current parameterization-based techniques only focus on directly visible discontinuities and therefore cannot handle such cases.

While gradients are important for high-dimensional optimization, they alone may not be enough when the objective is highly nonconvex. Certain scene representations (e.g. vertex positions of a triangle mesh) are particularly susceptible and produce undesirable local minima. Further research is necessary to understand how scene parameterizations affect the energy landscape of optimization tasks.

ACKNOWLEDGMENTS

We would like to thank Delio Vicini for many useful discussions about differentiable rendering.

Our test scenes use textures from CC0 Textures, Share Textures, and Texture Haven and are lit by environment maps courtesy of HDRI Haven and Paul Debevec. The scene in Figure 13 is adapted from Benedikt Bitterli's repository [2016] and the chess pieces in Figure 15 are from by Free3D user printable_models.

This research was supported by a grant from Autodesk.

REFERENCES

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

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

Petr Beckmann and Andre Spizzichino. 1987. The scattering of electromagnetic waves from rough surfaces. *Norwood* (1987).

Laurent Belcour, Cyril Soler, Kartic Subr, Nicolas Holzschuch, and Fredo Durand. 2013. 5D Covariance Tracing for Efficient Defocus and Motion Blur. *ACM Transactions on Graphics* 32, 3 (July 2013). https://doi.org/10.1145/2487228.2487239

Benedikt Bitterli. 2016. Rendering resources. https://benedikt-bitterli.me/resources/.

Chengqian Che, Fujun Luan, Shuang Zhao, Kavita Bala, and Ioannis Gkioulekas. 2018. Inverse Transport Networks. *arXiv preprint arXiv:1809.10820* (2018).

Min Chen and James Arvo. 2000. Theory and Application of Specular Path Perturbation. *ACM Transactions on Graphics* 19, 4 (Oct. 2000), 246–278.

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

Luc Devroye. 1986. *Non-Uniform Random Variate Generation.* Springer-Verlag.

Ioannis Gkioulekas, Anat Levin, and Todd Zickler. 2016. An evaluation of computational imaging techniques for heterogeneous inverse scattering. In *European Conference on Computer Vision*. Springer, 685–701.

Ioannis Gkioulekas, Shuang Zhao, Kavita Bala, Todd Zickler, and Anat Levin. 2013. Inverse Volume Rendering with Material Dictionaries. *ACM Transactions on Graphics* 32, 6, Article 162 (Nov. 2013).

Andreas Griewank and Andrea Walther. 2008. *Evaluating derivatives: principles and techniques of algorithmic differentiation.* Vol. 105. SIAM.

Carole K. Hayakawa, Jerome Spanier, Frédéric Bevilacqua, Andrew K. Dunn, Joon S. You, Bruce J. Tromberg, and Vasan Venugopalan. 2001. Perturbation Monte Carlo methods to solve inverse photon migration problems in heterogeneous tissues. *Opt. Lett.* 26, 17 (Sep 2001), 1335–1337.

Paul S Heckbert. 1989. *Fundamentals of texture mapping and image warping.* Master's thesis.

Eric Heitz and Eugene D'Eon. 2014. Importance Sampling Microfacet-Based BSDFs using the Distribution of Visible Normals. *Computer Graphics Forum* 33, 4 (July 2014), 103–112. https://doi.org/10.1111/cgf.12417

Binh-Son Hua, Adrien Gruson, Victor Petitjean, Matthias Zwicker, Derek Nowrouzezahrai, Elmar Eisemann, and Toshiya Hachisuka. 2019. A Survey on Gradient-Domain Rendering. *Computer Graphics Forum* 38, 2 (2019), 455–472.

Homan Igehy. 1999. Tracing Ray Differentials. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 99)*. 179–186. https://doi.org/10.1145/311535.311555

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* 31, 4 (July 2012).

Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. 2018. Neural 3D Mesh Renderer. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jaroslav Křivánek, Pascal Gautron, Sumanta Pattanaik, and Kadi Bouatouch. 2005. Radiance caching for efficient global illumination computation. *IEEE Transactions on Visualization and Computer Graphics* 11, 5 (2005), 550–561.

Samuli Laine, Janne Hellsten, Tero Karras, Yeongho Seol, Jaakko Lehtinen, and Timo Aila. 2020. Modular Primitives for High-Performance Differentiable Rendering. *ACM Transactions on Graphics* 39, 6 (2020).

Tzu-Mao Li, Miika Aittala, Frédo Durand, and Jaakko Lehtinen. 2018. Differentiable Monte Carlo Ray Tracing through Edge Sampling. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)* 37, 6 (2018), 222:1–222:11.

Shichen Liu, Weikai Chen, Tianye Li, and Hao Li. 2019. Soft Rasterizer: Differentiable Rendering for Unsupervised Single-View Mesh Reconstruction. *CoRR* abs/1901.05567 (2019). arXiv:1901.05567 http://arxiv.org/abs/1901.05567

Matthew M Loper and Michael J Black. 2014. OpenDR: An approximate differentiable renderer. In *European Conference on Computer Vision*. Springer.

Guillaume Loubet, Nicolas Holzschuch, and Wenzel Jakob. 2019. Reparameterizing discontinuous integrands for differentiable rendering. *Transactions on Graphics (Proceedings of SIGGRAPH Asia)* 38, 6 (Dec. 2019).

Iván Lux and Lázló Koblinger. 1990. *Monte Carlo Particle Transport Methods: Neutron and Photon Calculations.* CRC Press, Boston.

Don Mitchell and Pat Hanrahan. 1992. Illumination from curved reflectors. In *Proceedings of the 19th annual conference on Computer graphics and interactive techniques*. 283–291.

Merlin Nimier-David, Sébastien Speierer, Benoît Ruiz, and Wenzel Jakob. 2020. Radiative Backpropagation: An Adjoint Method for Lightning-Fast Differentiable Rendering. *Transactions on Graphics (Proceedings of SIGGRAPH)* 39, 4 (July 2020).

Merlin Nimier-David, Delio Vicini, Tizian Zeltner, and Wenzel Jakob. 2019. Mitsuba 2: A Retargetable Forward and Inverse Renderer. *Transactions on Graphics (Proceedings of SIGGRAPH Asia)* 38, 6 (Dec. 2019).

Art B. Owen. 2013. Monte Carlo theory, methods and examples. https://statweb.stanford.edu/~owen/mc/

Steven G. Parker, James Bigler, Andreas Dietrich, Heiko Friedrich, Jared Hoberock, David Luebke, David McAllister, Morgan McGuire, Keith Morley, Austin Robison, and Martin Stich. 2010. OptiX: A General Purpose Ray Tracing Engine. *ACM Transactions on Graphics* 29, 4, Article 66 (July 2010), 13 pages. https://doi.org/10.1145/1778765.1778803

Felix Petersen, Amit H. Bermano, Oliver Deussen, and Daniel Cohen-Or. 2019. Pix2Vex: Image-to-Geometry Reconstruction using a Smooth Differentiable Renderer. *CoRR* abs/1903.11149 (2019). arXiv:1903.11149 http://arxiv.org/abs/1903.11149

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.

Ravi Ramamoorthi, Dhruv Mahajan, and Peter Belhumeur. 2007. A first-order analysis of lighting, shading, and shadows. *ACM Transactions on Graphics (TOG)* 26, 1 (2007).

Helge Rhodin, Nadia Robertini, Christian Richardt, Hans-Peter Seidel, and Christian Theobalt. 2015. A Versatile Scene Model with Differentiable Visibility Applied to Generative Pose Estimation. In *Proceedings of ICCV 2015*.

Kenneth E Torrance and Ephraim M Sparrow. 1967. Theory for off-specular reflection from roughened surfaces. *Josa* 57, 9 (1967), 1105–1114.

T. S. Trowbridge and K. P. Reitz. 1975. *J. Opt. Soc. Am.* 65, 5 (May 1975), 531–536.

Eric Veach and Leonidas J. Guibas. 1995. Optimally Combining Sampling Techniques for Monte Carlo Rendering. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '95)*. Association for Computing Machinery, New York, NY, USA, 419–428. https://doi.org/10.1145/218380.218498

Darko Veberic. 2010. Having Fun with Lambert W(x) Function. *CoRR* abs/1003.1628 (2010). arXiv:1003.1628 http://arxiv.org/abs/1003.1628

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). https://doi.org/10.1145/3450626.3459804

Bruce Walter, Stephen R. Marschner, Hongsong Li, and Kenneth E. Torrance. 2007. Microfacet Models for Refraction through Rough Surfaces. In *Proceedings of the 18th Eurographics Conference on Rendering Techniques* (Grenoble, France) *(EGSR'07)*. Eurographics Association, Goslar, DEU, 195–206.

G J Ward and P S Heckbert. 1992. *Irradiance gradients.* Technical Report. Lawrence Berkeley Lab., CA (United States); Ecole Polytechnique Federale, Lausanne (Switzerland); Technische Hogeschool Delft (Netherlands). Dept. of Technical Mathematics and Informatics.

Tizian Zeltner, Iliyan Georgiev, and Wenzel Jakob. 2020. Specular Manifold Sampling for Rendering High-Frequency Caustics and Glints. *Transactions on Graphics (Proceedings of SIGGRAPH)* 39, 4 (July 2020). https://doi.org/10.1145/3386569.3392408

Cheng Zhang, Bailey Miller, Kai Yan, Ioannis Gkioulekas, and Shuang Zhao. 2020. Path-Space Differentiable Rendering. *ACM Transactions on Graphics* 39, 4 (2020), 143:1–143:19.

Cheng Zhang, Lifan Wu, Changxi Zheng, Ioannis Gkioulekas, Ravi Ramamoorthi, and Shuang Zhao. 2019. A Differential Theory of Radiative Transfer. *ACM Transactions on Graphics* 38, 6 (2019), 227:1–227:16.

Shaung Zhao, Lifan Wu, Frédo Durand, and Ravi Ramamoorthi. 2016. Downsampling Scattering Parameters for Rendering Anisotropic Media. *ACM Transactions on Graphics* 35, 6 (2016).

A DIFFERENTIAL MICROFACET SAMPLING

Section 4.1 presented the derivative of standard microfacet models with respect to their roughness parameter α, producing functions that consisted of a positive and negative lobe. Both lobes have equal area and can be individually normalized using normalization constants $N_{\text{GGX}} = 2/\alpha$ and $N_{\text{Beck.}} = 2/(\alpha\, e)$ respectively.

Here, we show how they can be sampled using inverse transform sampling [Devroye 1986]. We first integrate the derivative with respect to α over elevation angles θ to obtain associated CDFs P^{+}_{NDF} and P^{-}_{NDF} that we invert to arrive at the following sampling techniques that suitably transform uniform random variates $\xi \in [0, 1)$:

$$\theta^{+}_{\text{GGX}} = \tan^{-1}\left(\alpha\sqrt{-\left(1 + \frac{2}{\sqrt{\xi} - 1}\right)}\right),$$

$$\theta^{-}_{\text{GGX}} = \frac{1}{2}\tan^{-1}\left(\frac{2\alpha\sqrt{\xi(2 - 2\sqrt{1-\xi} + 2\alpha^4(1 + \sqrt{1-\xi}) - \xi(\alpha^2-1)^2)}}{\xi + 4\alpha^2\sqrt{1-\xi} - \xi\alpha^4}\right),$$

$$\theta^{+}_{\text{Beck.}} = \tan^{-1}\left(\alpha\sqrt{-\mathrm{W}_{-1}\left(-\frac{\xi - 1}{e}\right)}\right), \quad \theta^{-}_{\text{Beck.}} = \tan^{-1}\left(\alpha\sqrt{-\mathrm{W}_{0}\left(-\frac{\xi}{e}\right)}\right).$$

Note that the Beckmann variant uses the branches $k \in \{-1, 0\}$ of the *Lambert W function* $\mathrm{W}_k(x)$ which unfortunately is not available in analytic form. We found it easiest to use a numeric evaluation based on a few iterations of a Newton solver, though alternatively, approximated version that can be evaluated directly are available as well [Veberic 2010].