EUROGRAPHICS 2019
A. Giachetti and H. Rushmeier
(Guest Editors)

Volume 38 (2019), Number 2
STAR – State of The Art Report

A Survey on Gradient-Domain Rendering

†Binh-Son Hua[1] †Adrien Gruson[1,4] Victor Petitjean[2]
Matthias Zwicker[3] Derek Nowrouzezahrai[4] Elmar Eisemann[2] Toshiya Hachisuka[1]
[1]The University of Tokyo [2]Delft University of Technology [3]University of Maryland, College Park [4]McGill University

Abstract

Monte Carlo methods for physically-based light transport simulation are broadly adopted in the feature film production, animation and visual effects industries. These methods, however, often result in noisy images and have slow convergence. As such, improving the convergence of Monte Carlo rendering remains an important open problem. Gradient-domain light transport is a recent family of techniques that can accelerate Monte Carlo rendering by up to an order of magnitude, leveraging a gradient-based estimation and a reformulation of the rendering problem as an image reconstruction. This state of the art report comprehensively frames the fundamentals of gradient-domain rendering, as well as the pragmatic details behind practical gradient-domain uni- and bidirectional path tracing and photon density estimation algorithms. Moreover, we discuss the various image reconstruction schemes that are crucial to accurate and stable gradient-domain rendering. Finally, we benchmark various gradient-domain techniques against the state-of-the-art in denoising methods before discussing open problems.

CCS Concepts

• ***Computing methodologies*** → ***Ray tracing;***

1. Introduction

Faithfully simulating the physical behavior of light remains a long-standing problem in computer graphics. While Monte Carlo approaches provide an avenue to approximate solutions to this problem, existing methods still require minutes to hours to synthesize noise-free images. Improving the performance of physically-based rendering algorithms remains an important active research topic.

Among many others, recent advances of *gradient-domain rendering* provide significant improvement on visual convergence by exploiting image-space smoothness and coherence in sensor path subspaces [LKL*13, KMA*15, MKA*15]. The seminal work on gradient-domain Metropolis light transport [LKL*13] motivated gradient-domain variants of uni- and bi-directional path tracing [KMA*15, MKA*15], as well as gradient-domain density estimation on surfaces [HGNH17, SSC*17] and volumes [GHV*18]. The common idea is to estimate the *image gradients* in addition to the pixel intensities, and apply image reconstruction to synthesize the final image. Gradient-domain approaches tend to be more efficient than their conventional counterparts since solutions to the light transport problem often exhibit piecewise smooth variations in the image space: numerical estimates of the image gradients tend to be small for a majority of pixels.

This state of the art report outlines the theoretical and practical considerations needed when approaching gradient-domain rendering. After reviewing general concepts and components of gradient-domain rendering (Section 2), we discuss practical considerations that lead to successful gradient-domain methods and the pragmatics of implementing such methods. This report also serves as a comprehensive collection of the state-of-the-art in gradient-domain rendering. For readers who are familiar with conventional (i.e., primal-domain) light transport simulation, our report also includes a tutorial on implementing a basic gradient-domain path tracer.

We detail the fundamental components behind efficient image gradient estimation: effective shift mappings (Section 3) and their applications to uni- and bi-directional light transport in the gradient domain [KMA*15, MKA*15]. We follow with extensions to gradient-domain photon density estimation [HGNH17] and gradient-domain vertex connection and merging [SSC*17] (Section 4). After detailing the numerical estimation of image gradients, we describe various approaches used to reconstruct the final image from these gradients (Section 5). We also highlight more robust reconstruction algorithms [MVZ16, BYM18] related to Monte Carlo image denoising techniques [BRM*16, BB17] and advanced gradient sampling approaches (Section 6). The strengths of gradient-domain rendering for surface light transport have also recently been extended to higher dimensional problems, such as that of light transport in scenes comprising volumetric participating media [GHV*18], temporal image sequences [MKD*16] and spectral rendering [PBE18]. Section 7 details these works. We will draw relationships to applications of gradient-domain techniques for path reusing [BPE17], non-contiguous pixel differences [MRK*14], and adaptive sampling [LKL*13, BYM18], before concluding with a discussion of practical implementation details and important open problems for future work (Section 8).

† joint first authors

2. General Principles of Gradient-Domain Light Transport

We review the fundamentals of numerical image formation in the primal and gradient domains, below.

2.1. Monte Carlo Rendering

The foundation of physically-based light transport on surfaces is captured by the rendering equation [Kaj86]: given a light path $\mathbf{x} = \mathbf{x}_0 \dots \mathbf{x}_\ell$ of length ℓ in the path space $\mathcal{P}$ (the points $\mathbf{x}_0$ lies on an emitter and $\mathbf{x}_\ell$ on a sensor), the intensity I_p of pixel p is defined as

$$I_p = \int_{\mathcal{P}} h_p(\mathbf{x}) f(\mathbf{x}) d\mathbf{x}, \tag{1}$$

where $f(\mathbf{x})$ is the measurement contribution function and $h_p(\mathbf{x})$ is the reconstruction filter at pixel p. Commonly used Monte Carlo integration approximates this integral as

$$\langle I_p \rangle = \frac{1}{N} \sum_{k=1}^{N} \frac{h_p(\mathbf{x}_k) f(\mathbf{x}_k)}{p(\mathbf{x}_k)}, \tag{2}$$

where the estimate $\langle I_p \rangle$ for pixel p is computed using N path samples $\mathbf{x}_k$ from a probability density function $p(\mathbf{x})$ defined over the measure of the path space $\mathcal{P}$. Under reasonable constraints for $p(\mathbf{x})$, this estimator is unbiased and the expected squared-error is proportional to the estimator's variance which vanishes as $N \to \infty$.

In practice for finite N, this error manifests as noise in the image. A simple approach to reduce this noise is to increase N, however this suffers from the slow $\mathcal{O}(1/\sqrt{N})$ convergence of Monte Carlo integration: e.g., quadrupling N only reduces the error by a factor of two. As such, reducing the error for *fixed* N remains an active problem, despite its long history in physically-based rendering.

We focus on *gradient-domain rendering*, a family of methods that exploit image-space pixel coherence to accelerate Monte Carlo rendering. Here, the basic idea is to estimate image gradients *in addition to* pixel intensities, and then to reconstruct the final image using these gradients. The use of image gradients has been studied widely in computer vision and graphics, mainly for image editing [PGB03], image-based rendering [KLS*13], image filtering [Bha09], and surface reconstruction applications [ACR05], as well as in relevant fields such as tomography in medical imaging [MSKN17]. Readers can refer to the excellent course by Agrawal and Raskar for more technical details [AR07]. The common concept of such existing work and gradient-domain rendering is that they all share a reconstruction step based on the gradients. Unlike existing work on image gradients, however, gradient-domain rendering needs to estimate image gradients efficiently based on the rendering equation.

2.2. Gradient-Domain Rendering

Figure 1 illustrates the general concept of gradient-domain rendering. Conventional Monte Carlo rendering estimates pixel intensities by averaging the contribution of many sampled paths. Gradient-domain rendering is built on top of this process by adding two important tweaks: *shift mapping*, a path-space mapping function that generates a correlated path from a provided input path to build an efficient estimator of the difference of a pixel pair (i.e., image gradients), and *image-space reconstruction* that acts as a filtering process of the estimated values to obtain the final image.

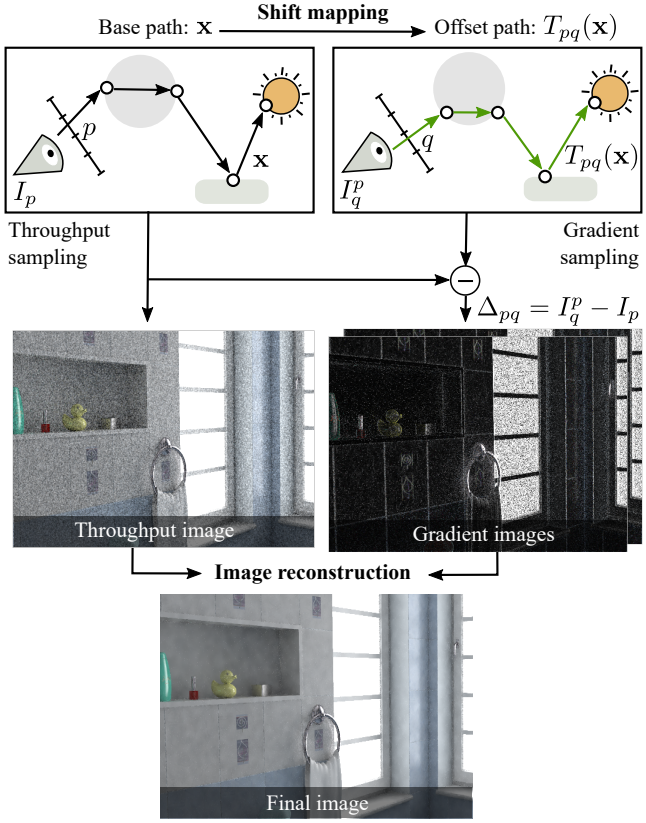


Figure 1: *The general concept of gradient-domain: First, a base path is generated with a conventional rendering algorithm like path tracing. Second, shift mapping (Section 3) is used to generate the offset path. The measurement contribution function is evaluated to create primal and gradient images. In the end, an image reconstruction process (Section 5) is used to generate the final image.*

A naive approach o estimate the image *gradient* between two pixels (i.e. the difference of the pixel intensities) is to estimate Equation 1 independently at each pixel and subsequently take the difference. The variance of this naive estimator is simply the sum of the variance of the two pixels, which means that this extra estimation does not bring any benefit in terms of estimation error, and it will therefore not lead to any improvement in the estimation of the final image. *Shift mapping* in gradient-domain rendering instead exploits path correlation between two pixels p and q, resulting in a lower variance estimator of image gradients. Using shift mapping, we can define the intensity of the neighboring pixel q of pixel p as

$$I_q^p = \int_{\mathcal{P}} h_q(T_{pq}(\mathbf{x})) f(T_{pq}(\mathbf{x})) \left| \frac{dT_{pq}(\mathbf{x})}{d\mathbf{x}} \right| d\mathbf{x}, \tag{3}$$

where T_{pq} is a path-space function (*shift mapping* function) that transforms a path through pixel p (*base* path) into a path through pixel q (*offset* path). The Jacobian determinant $\left| \frac{dT_{pq}(\mathbf{x})}{d\mathbf{x}} \right|$ accounts for the change of integration domain from p to q. As we estimate pixel q using paths from pixel p shifted exactly by one pixel, the reconstruction filter at pixel q can be defined as $h_q(T_{pq}(\mathbf{x})) = h_p(\mathbf{x})$.

The shift mapping function T_{pq} is designed such that the base path

$\mathbf{x}$ and offset path $T_{pq}(\mathbf{x})$ are as similar as possible. Using Equation 3, we can write the difference between two pixels as

$$\Delta_{pq} = I_q^p - I_p = \int_{\mathcal{P}} h_p(\mathbf{x}) \left(f(T_{pq}(\mathbf{x})) \left| \frac{dT_{pq}(\mathbf{x})}{d\mathbf{x}} \right| - f(\mathbf{x}) \right) d\mathbf{x}. \quad (4)$$

Since the shift mapping function T_{pq} implies $T_{pq}(x) \approx x$, the integrand in Equation 4 becomes zero almost everywhere. Applying Monte Carlo integration to this integral thus leads to a low variance estimator of image gradients, when compared to directly taking finite differences of the pixel intensities.

Given estimated gradients, computing the final image is an *image-space reconstruction* problem. A typical solution uses image-space Poisson reconstruction [BCCZ08], which finds an image that best fits the estimated pixel intensities and gradients [LKL*13]:

$$I = \arg\min_I \|D_x(I) - G_x\|_n^n + \|D_y(I) - G_y\|_n^n + \lambda \|I - I_0\|_n^n, \quad (5)$$

where D_x and D_y are horizontal and vertical image gradients computed using finite difference, G_x and G_y are gradients estimated with shift mapping, I_0 is the noisy Monte Carlo image, and λ controls the similarity between the reconstruction and the noisy image. Typical norms used for reconstruction are $n = 2$ and $n = 1$. Existing solvers [She94] can be applied to solve this optimization problem.

Relevance to Path-Space Gradients. The concept of gradients of light transport paths has been explored well in the literature. In classical ray tracing, ray differentials [Ige99] is developed to track the distance between two neighboring rays for texture filtering. Path differentials [SW01] is an extension of ray differentials to arbitrary sampled paths with diffuse and glossy interactions. However, as the path footprints are only approximated by analytical gradients, both techniques never require to trace two actual paths.

Illumination gradients are used in irradiance caching [WRC88] and its extensions [WH92, KGPB05, GBP07, JDZJ08] to position cache points for interpolating high-quality indirect illuminations. It is also used for bias and variance estimation in progressive photon mapping [HJJ10]. The theory of the first- and second-order derivatives of specular paths [CA00] was also studied, which leads to robust techniques for perturbing specular paths such as manifold exploration [JM12] and half-vector space exploration [KHD14, HKD15]. Path derivatives can also be computed with automatic differentiation together with the sampling of geometric edges [LADL18].

Path-space gradients differ fundamentally different from gradients in gradient-domain rendering. The latter is interested in estimating pixel intensity differences by manipulating paths. These differences are essentially intensity gradients in pixel-discretized image space, and thus they have little relevance to gradients in path space, despite the overloaded "gradient-domain" naming.

2.3. Proof-of-Concept Gradient-Domain Path Tracer

We now apply the basic ideas we introduced and derive a simple path tracer in the gradient domain. This path tracer generally follows the process of conventional path tracing: we trace paths through each pixel starting from the camera. To turn it into a gradient-domain path tracer, we need to implement: shift mapping to estimate gradients (Equation 3 and 4) as well as image reconstruction (Equation 5). We list these steps in Algorithm 1, and are explained below.

Shift Mapping. One simple shift mapping simply reuses the same set of random numbers to generate correlated paths. We call this shift mapping *random sequence replay*. This idea was first explored in Metropolis light transport in the primary sample space [KSKAC02] and in the pixel dependency tests for filtering [CSKKA02]. The same shift mapping is also used for temporal gradient-domain rendering [MKD*16]. The assumption here is that path samples with the same random numbers are likely to be similar in the path space. Let $\mathbf{x}$ be a path that is successfully sampled by path tracing in pixel p, and all the random numbers that are used to generate $\mathbf{x}$ are recorded as u. Random sequence replay generates the offset path $\mathbf{x}' = T_{pq}(\mathbf{x})$ by tracing through the offset pixel q using path tracing with $u' = u$. To achieve $u' = u$, every time a random number is requested by path tracing, we take the number from u instead of running a random number generator.

After generating $\mathbf{x}'$, we need to evaluate the Jacobian determinant in order to evaluate Equations 3 and 4. Following the derivation by Kelemen et al. [KSKAC02], the Jacobian determinant in this case is

$$\left| \frac{d\mathbf{x}'}{d\mathbf{x}} \right| = \left| \frac{d\mathbf{x}'}{du'} \right| \left| \frac{du'}{du} \right| \left| \frac{du}{d\mathbf{x}} \right| = \frac{p(\mathbf{x})}{p(T_{pq}(\mathbf{x}))}. \quad (6)$$

The contribution of the offset path is thus simplified to

$$\frac{f(T_{pq}(\mathbf{x}))}{p(\mathbf{x})} \left| \frac{dT_{pq}(\mathbf{x})}{d\mathbf{x}} \right| = \frac{f(T_{pq}(\mathbf{x}))}{p(\mathbf{x})} \frac{p(\mathbf{x})}{p(T_{pq}(\mathbf{x}))} = \frac{f(T_{pq}(\mathbf{x}))}{p(T_{pq}(\mathbf{x}))}, \quad (7)$$

meaning we just need to divide the offset measurement contribution function $f(T_{pq}(\mathbf{x}))$ by the offset path probability density $p(T_{pq}(\mathbf{x}))$, as if it was sampled by path tracing pixel q. The gradient estimator for Equation 4 is

$$\langle \Delta_{pq} \rangle = \frac{1}{N} \sum_{k=1}^{N} h_p(\mathbf{x}_k) \left(\frac{f(\mathbf{x}_k)}{p(\mathbf{x}_k)} - \frac{f(T_{pq}(\mathbf{x}_k))}{p(T_{pq}(\mathbf{x}_k))} \right). \quad (8)$$

Note that there are two strategies to estimate the difference between pixel p and q: sampling a base path through pixel p and shifting it to pixel q and vice versa. In this simple implementation, we perform both strategies and weigh each strategy with a constant factor of 0.5. The contribution of the base paths of each pixel are still accumulated to obtain an image due to conventional Monte Carlo rendering (called the *primal image*). Putting all the steps together leads to the function simpleGradientPathTracing() in Algorithm 1.

Reconstruction. After sampling the primal image and the horizontal and vertical gradients, we solve for the final image. In this section, we explain an iterative solver based on control variates [RJN16] that has similar effects to the $L2$-norm case in Equation 5. This solver first redefines the pixel intensity I_p as the sum of the neighboring pixel intensity I_q and the image gradient $(I_p - I_q)$:

$$I_p = I_p + (I_q - I_q) = I_q + (I_p - I_q). \quad (9)$$

Using this definition, we can decompose the pixel intensity estimator $\langle I_p \rangle$ into the gradient estimator $\langle I_p - I_q \rangle$ and another pixel intensity estimator $\langle I_q \rangle$:

$$\langle I_p \rangle = \langle I_q \rangle + \langle I_p - I_q \rangle. \quad (10)$$

Considering the image I as unknown values and the gradient estimates as fixed constants, this formulation defines I as a fixed point

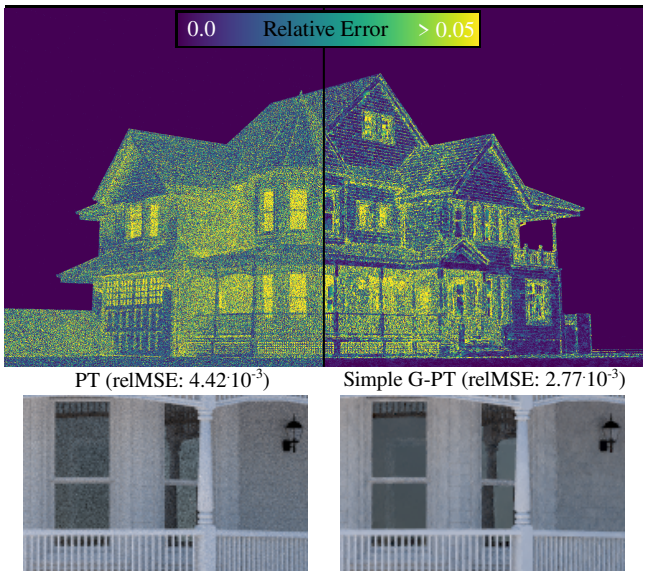


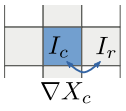
Figure 2: *Equal-time comparison between path tracing and the proof-of-concept gradient-domain path tracing in Section 2.3. With random sequence replay shift mapping and the iterative solver for image reconstruction that are straightforward to implement (Algorithm 1), the convergence is improved up to 2×. Using more sophisticated shift mapping and image reconstruction techniques lead to even faster convergence (see Sections 3, 4, 5).*

of a function F: $I = F(I)$. Typically, such an equation allows us to retrieve I in an iterative process, where F is consecutively applied to I:

$$I_p^{(n+1)} = I_q^{(n)} + \langle I_p - I_q \rangle \,, \tag{11}$$

where $I^{(0)}$ is initialized by the primal-domain image obtained by Monte Carlo rendering and $I^{(n)}$ is the resulting image at iteration n. Note that the estimated gradients $\langle I_p - I_q \rangle$ are kept fixed during the iterations. From Equation 11, we see that the reconstruction converges to an image that have the gradients equal to the estimated gradients when $n \to \infty$.

We now consider the center pixel $c = (x,y)$ and four neighbors including right $r = (x+1,y)$, bottom $b = (x,y+1)$, left $l = (x-1,y)$, and top $t = (x,y-1)$ to perform this reconstruction step. Since each pixel has four neighbors, shifting the base path from pixel c to its neighbors leads to the following four gradient estimators (as illustrated in the inline figure):



$$\begin{aligned} \nabla X_c &= \langle I_r - I_c \rangle & \nabla X_l &= \langle I_c - I_l \rangle \\ \nabla Y_c &= \langle I_b - I_c \rangle & \nabla Y_t &= \langle I_c - I_t \rangle \,. \end{aligned} \tag{12}$$

In addition to the trivial identity $\langle I_c \rangle = \langle I_c \rangle$, Equation 10 establishes four identities for $\langle I_c \rangle$ using each one of the four neighbors. We simply take the average of these five identities to obtain

$$\begin{aligned} \langle I_c \rangle = \frac{1}{5}(&\langle I_c \rangle + (\langle I_r \rangle - \nabla X_c) + (\langle I_l \rangle + \nabla X_l) \\ &+ (\langle I_b \rangle - \nabla Y_c) + (\langle I_t \rangle + \nabla Y_t)) \,. \end{aligned} \tag{13}$$

```
1  function simpleGradientPathTracing()
2      (P, G_x, G_y) = initialize images with zeros
3      for k := 1 to N do
4          foreach pixel sample (x, y) do
5              u = random_numbers()
6              base = radiance(u, x, y)
7              P[x, y] += base
8              G_x[x − 1, y] += 0.5 ∗ (base − radiance(u, x − 1, y))
9              G_y[x, y − 1] += 0.5 ∗ (base − radiance(u, x, y − 1))
10             G_x[x, y] += 0.5 ∗ (radiance(u, x + 1, y) − base)
11             G_y[x, y] += 0.5 ∗ (radiance(u, x, y + 1) − base)
12         end
13     end
14     P /= N, G_x /= N, G_y /= N
15     I = reconstruct(P, G_x, G_y)
16     return I
17 end

18 function reconstruct(P, G_x, G_y)
19     initialize final image I = P, temporary image J
20     for i := 1 to max iterations do
21         foreach pixel (x, y) do
22             v = I[x, y]
23             v += I[x − 1, y] + G_x[x − 1, y]
24             v += I[x, y − 1] + G_y[x, y − 1]
25             v += I[x + 1, y] − G_x[x, y]
26             v += I[x, y + 1] − G_y[x, y]
27             J[x, y] = v/5
28         end
29         I = J
30     end
31     return I
32 end
```

Algorithm 1: Implementation of a proof-of-concept gradient-domain path tracer (Section 2.3). Shift mapping can be implemented by simply reusing base path random numbers. This requires minimal changes to existing path tracers as only a function that can construct the path and evaluate its contribution from a set of random samples is needed. This simplicity is achieved at the cost of moderate correlation for the base and offset paths.

which leads to the iterative update of $I_c^{(n)}$ as

$$\begin{aligned} I_c^{(n+1)} = \frac{1}{5}(&I_c^{(n)} + (I_r^{(n)} - \nabla X_c) + (I_l^{(n)} + \nabla X_l) \\ &+ (I_b^{(n)} - \nabla Y_c) + (I_t^{(n)} + \nabla Y_t)) \,. \end{aligned} \tag{14}$$

Empirically, using 30 to 50 iterations works well to reconstruct a 1280×720 image. This reconstruction algorithm is implemented as the second function reconstruct(P, G_x, G_y) in Algorithm 1.

Discussion. The gradient-domain path tracer we detail here is designed to provide an easy-to-implement baseline prototype in order to better understand the basic concepts of gradient-domain rendering. This simplified gradient-domain path tracer already results in a speedup compared to primal-domain path tracing in some scenes (Figure 2), however it far from optimal in many important aspects. First, random sequence replay often fails to produce similar offset paths. The offset path from random sequence replay can diverge

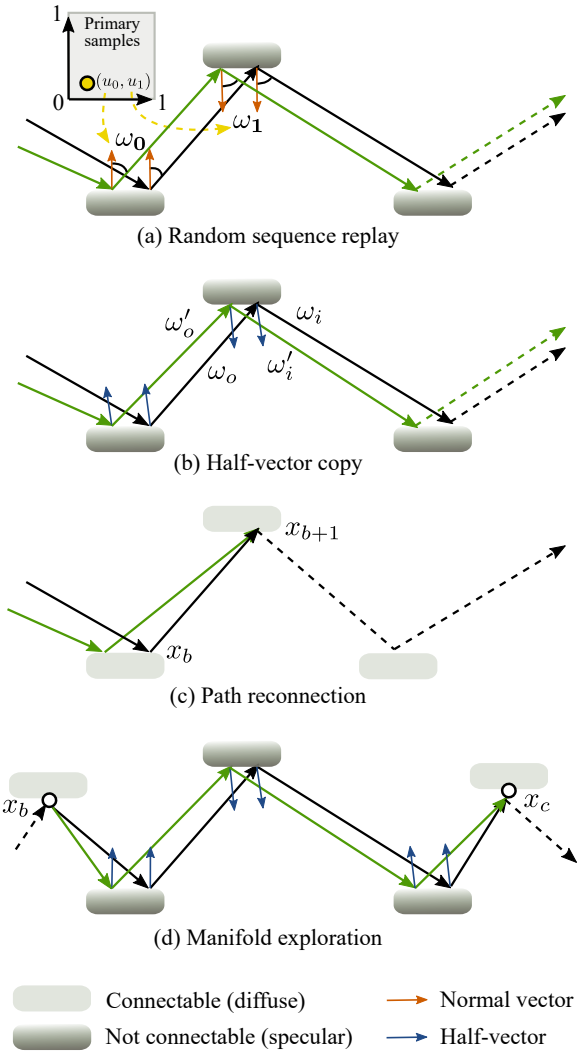


Figure 3: *Illustration of existing shift mapping techniques. Shift mapping generates a correlated offset path (green) from a base path (black) so that their contribution difference is a good estimate of the underlying pixel gradient.*

from a base path when the offset path hits a surface with a BSDF different from that on the base path. Section 3 introduces more sophisticated shift mapping functions in path space to achieve better correlation. Second, unidirectional path tracing is not the only technique that can be implemented in gradient-domain. Section 4 details how to incorporate bidirectional light transport techniques which are more robust to various scene configurations. Finally, the simple iterative solver we explained here might result in visual artifacts such as bright spots due to have noisy gradients. Image reconstruction from gradients is indeed a non-trivial task, and robust image reconstruction solutions are introduced in Section 5.

3. Shift Mapping

Shift mapping is a deterministic perturbation in the path space which transforms a base path into an offset path. Ideally, one would like to make the integrand of Equation 4 small for any $\mathbf{x}$, such that the variance of its Monte Carlo estimator also becomes small. Shift mapping achieves this by making offset paths $T_{pq}(\mathbf{x})$ highly correlated with base paths $\mathbf{x}$. This section categorizes and discusses existing shift-mapping techniques which be adopted for building various integrators in the later sections. Figure 3 illustrates the shift mapping techniques that we discuss below.

To simplify our exposition, we assume that shift mapping is always successful and that offset paths can always be generated. In practice, there is no such guarantee since offset paths might not hit an object, or the BSDF evaluation approaches zero. We will discuss how to robustly deal with these scenarios later in Section 3.5.

3.1. Random Sequence Replay

For completeness, we briefly review random sequence replay from Section 2, below. The idea of random sequence replay [MKD*16] is to generate an offset path by repeating the same set of random numbers that were used to generate its associated base path (Figure 3(a)). This approach enables each vertex in the offset path to be generated using existing path sampling techniques. Equation 6 shows how to calculate the Jacobian determinant of this shift mapping.

3.2. Half-Vector Copy

While random sequence replay is simple and general, reusing random numbers does not necessarily lead to highly correlated paths. A more effective solution is to directly manipulate path space samples, exposing more direct control over inter-path correlation.

One such idea is to contrain the same half-vector at each path vertex [KMA*15] (Figure 3(b)). Given a base path of the form $(x_0 \dots x_{k-1} x_k x_{k+1} \dots)$ and a partial offset path $(x'_0 \dots x'_{k-1} x'_k)$, to determine the offset vertex x'_{k+1} by preserving the half-vectors, we should have

$$h = \frac{\omega_i + \omega_o}{\|\omega_i + \omega_o\|} = \frac{\omega'_i + \omega'_o}{\|\omega'_i + \omega'_o\|} = h', \tag{15}$$

where $\omega_o, \omega'_o, \omega_i, \omega'_i$ are the outgoing (from vertex k to vertex $k-1$) and incoming directions (from vertex k to vertex $k+1$) at the base and offset vertex k. As ω_o, ω_i, and ω'_o are known, one can solve for direction ω'_i. We can determine vertex x'_{k+1} by tracing a ray from x'_k toward the direction ω'_i. Please refer to Kettunen et al.'s supplement [KMA*15] for the Jacobian determinant derivation.

3.3. Path Reconnection

While half-vector copy provides more direct control over the shape of the offset path, the offset paths are still generated from scratch and no vertices are shared between the base and the offset paths. This is not only computationally expensive, but also results in sub-optimal path correlation. Path reconnection addresses this issue by connecting the offset path back to the base path. When successful, it enables reuse of all the vertices on the base path after the point

of reconnection. The idea of path reconnection can be dated back to the early work by Bekaert et al. [BSH02] in the name of path reusing for Monte Carlo path tracing.

Consider a base path $\mathbf{x} = (x_0 \dots x_b x_{b+1} \dots)$ where x_0 is on the camera, and the partially constructed offset path $\mathbf{x}' = (x'_0 \dots x'_b)$ using random replay or half-vector copy. Path reconnection improves correlation by connecting the offset path vertex x'_b to x_{b+1} (Figure 3(c)). After vertex x_{b+1}, all the vertices of the base and offset paths are the same. The Jacobian determinant for reconnection is simply one in the area measure [KMA*15].

Path reconnection is applicable only when the BSDFs at x_b, x_{b+1} and x'_{b+1} do not significantly differ. Specular or highly-glossy BSDFs at those vertices often lead to paths with zero throughput, thus we classify vertices whose BSDFs are "diffuse" enough as *connectable*. For path reconnection to be efficient, all the vertices x_b, x_{b+1}, and x'_{b+1} need to be connectable In gradient-domain path tracing [KMA*15], a threshold on the BSDF roughness is used to classify a vertex as connectable or not.

3.4. Manifold Exploration

A common scenario in complex light transport is the existence of consecutive specular vertices between two connectable vertices. A more sophisticated shift mapping is required to handle such *specular chains*. Suppose that the base path has the form $\mathbf{x} = (x_0 \dots x_b \dots x_c \dots)$ where x_b and x_c are connectable and vertices between x_b and x_c are specular. Jakob et al. [JM12] proposed *manifold exploration* that can be applied to connect a partial offset path $(x'_0 \dots x'_b)$ to x_c (Figure 3(d)). Similar to half-vector copy, manifold exploration preserves all half-vectors on the specular chain during offset path generation. The main difference is that the half-vectors serve as constraints that define an implicit path space for which the vertices of the specular chain can be searched for using a Newton solver.

While manifold exploration leads to more robust shift mapping for base paths that contain specular or glossy vertices, it is usually computationally expensive. Our experience is that it is good to balance the use of manifold exploration with the shift failure rate. Sometimes it could be cheaper to let the shift fail and sample the gradients with finite differences on the primal domain to trade for more rendering passes.

3.5. Practical Issues

We discuss a few important practical considerations, next.

Forward and Backward Shift. As we briefly mentioned in Section 2.3, one can generally estimate gradients in two different ways. For a base pixel p and an offset pixel q, the gradient Δ_{pq} can be estimated by shifting a base path from p to q (*forward* shift), or vice versa (*backward* shift). When a base path can be shifted to an offset path and vice versa, this shift mapping is called *reversible*. Therefore, Equation 4 can be extended to a weighted sum of two

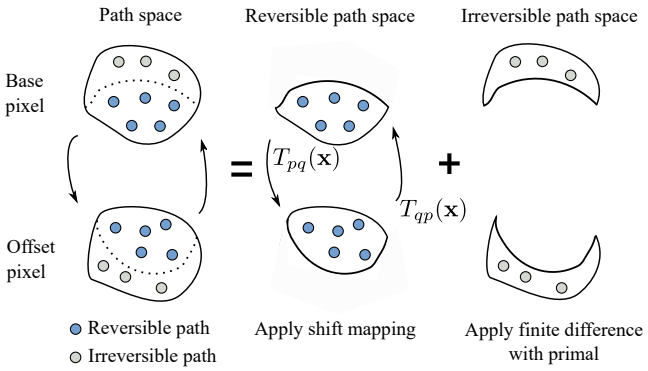


Figure 4: *Reversible and irreversible paths are handled separately in gradient estimation in Equation 16. Only a subspace in the integral domain has bijective shift mapping, i.e., a base path can be shifted into an offset path using a forward shift, and applying the backward shift on the offset path results in the same base path. When both base and offset paths belong to this subspace, Equation 16 results in low-variance estimation of the gradients. When one of the paths does not fall into this subspace, the gradients are simply estimated by taking the finite difference on pixel values of the primal domain.*

gradient estimates:

$$\begin{aligned}\Delta_{pq} = &\int_{\mathcal{P}} h_p(\mathbf{x}) w_{pq}(\mathbf{x}) \left(f(T_{pq}(\mathbf{x})) \left| \frac{dT_{pq}(\mathbf{x})}{d\mathbf{x}} \right| - f(\mathbf{x}) \right) d\mathbf{x} \\ &+ \int_{\mathcal{P}} h_q(\mathbf{x}) w_{qp}(\mathbf{x}) \left(f(\mathbf{x}) - f(T_{qp}(\mathbf{x})) \left| \frac{dT_{qp}(\mathbf{x})}{d\mathbf{x}} \right| \right) d\mathbf{x},\end{aligned} \tag{16}$$

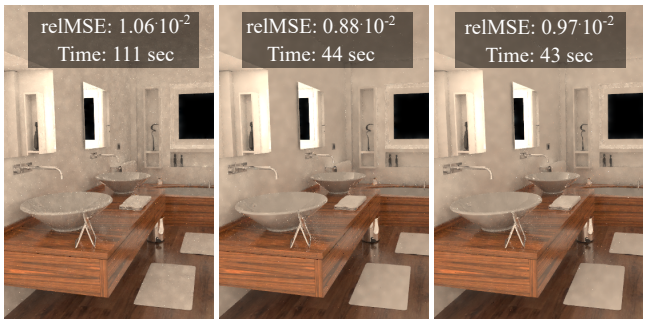
where w_{pq} and w_{qp} are the weights of the forward and backward shift mapping. The first integral corresponds to shifting base path from p to q. The second integral corresponds to the reversed case from q to p. A simple strategy is to weigh both shifts equally (i.e., $w_{pq}(x) = w_{qp}(x) = 0.5$).

A more efficient weighting scheme is to use multiple importance sampling (MIS) [VG95b]:

$$w_{pq}(\mathbf{x}) = \frac{p(\mathbf{x})}{p(\mathbf{x}) + p(T_{pq}(\mathbf{x})) \left| \frac{dT_{pq}(\mathbf{x})}{d\mathbf{x}} \right|}. \tag{17}$$

In the case of reversible shifts, we can apply this MIS weight to combine the gradient estimation of both the forward and backward strategies. The forward shift uses w_{pq} and the backward shift uses w_{qp} based on the same formulation. In Section 2.3, we set the weights for the forward and backward shift equally to 0.5. In fact, this value can also be interpreted as a special case in Equation 17 when random sequence replay is used. In this case, the Jacobian determinant in Equation 6 cancels out the offset path probability density, leaving only the base path probability terms in Equation 17, which results in value 0.5.

Unsuccessful Shift Mapping. In practice, shift mapping is not always reversible and there is no guarantee that the mapping is bijective. For example, a base path can be mapped to an offset path that cannot be sampled (i.e., the probability density to generate the offset path using the traditional Monte Carlo path sampling is



(a) Random replay (b) Random replay + path reconnection (c) Half-vector copy + path reconnection

Figure 5: *Equal-sample comparison of different shift mappings for gradient-domain path tracing provides the following observations. (1) With path reconnection, the base and offset paths become more correlated, which reduces variance in gradient estimates. Path reconnection also helps lower the overhead of shift mapping once the offset path merges into the base path. (2) Random sequence replay has very competitive performance to half-vector copy when used with path reconnection.*

zero). In such cases, shift mapping is marked as unsuccessful and *irreversible*, and Equation 17 is no longer applicable. To handle this case, one can set the weights of both forward and backward shift mapping to 1, and return zero as the contribution of the offset path. Equation 16 then degenerates into computing the finite difference by the contributions of the base paths at pixel p in the first integral, and the base paths at pixel q in the second integral. Figure 4 further explains reversibility by illustrating the path space at the base and offset pixels.

Combined Shift Mapping. Generally speaking, no single shift mapping method works well for all the cases. For example, random sequence replay and half-vector copy can maintain specular chains effectively but do not take path correlation into account. Path reconnection maximizes correlation but might result in zero-contribution offset paths since it does not consider BSDFs during the shift. In practice, shift mapping used in gradient-domain light transport algorithms combine multiple approaches.

For example, the original gradient-domain path tracing [KMA*15] employed half-vector copy with the combination of path reconnection. As an alternative, combining random sequence replay with path reconnection is a simple but effective approach to improve robustness of shift mapping, when the scene contains a mix of specular and diffuse objects. In gradient-domain photon density estimation [HGNH17] and its volumetric rendering version [GHV*18], path reconnection is used with half-vector copy to shift density estimation paths.

Figure 5 compares the performance of three possible shift mappings for gradient-domain path tracing: random sequence replay, random sequence replay with path reconnection, and half-vector with path reconnection. We observe that applying path reconnection with either a random sequence replay or a half-vector copy improves path correlation, and so serves to reduce the variance of gradient estimates. Furthermore, it also reduces the computational cost due to vertex reuse. Lastly, we do not observe any noticeable difference in the effectiveness of random sequence replay and half-vector copy.

4. Path Sampling Techniques

While we initially focused on path tracing in Section 2, gradient domain rendering is also amenable to more sophisticated light transport algorithms. We begin with unidirectional path tracing, following the original gradient-domain path tracing [KMA*15] work, before detailing bidirectional path tracing and photon density estimation in the gradient-domain [MKA*15, HGNH17, SSC*17]. Figure 6 summarizes typical base and offset path configurations for each of these gradient-domain light transport methods.

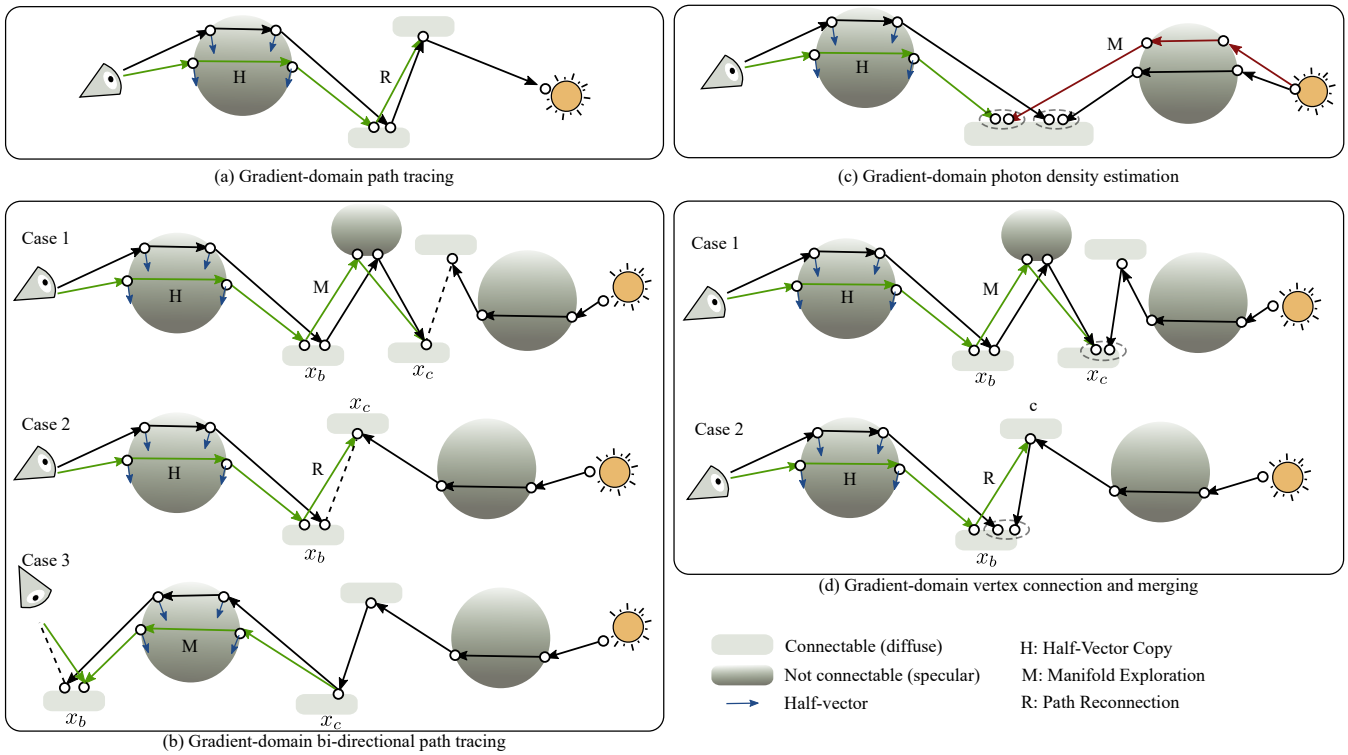
4.1. Unidirectional Path Tracing

Unlike our simplified version in Section 2, the original gradient-domain path tracing algorithm [KMA*15] uses half-vector copy to shift a specular chain from the base to the offset path. Path reconnection is applied when we encounter two consecutive connectable vertices on the base path. Anderson et al. [ALLD17] provide a more compact implementation of gradient-domain path tracing, where the path probability density is evaluated using automatic differentiation, at the cost of runtime efficiency. Figure 6(a) illustrates the original algorithm of gradient-domain path tracing. Compared to our simplified version, the original shift mapping is much more efficient due to the use of the combination of half-vector copy and path reconnection.

4.2. Bidirectional Path Tracing

We briefly review the foundations of bidirectional path tracing before discussing its gradient-domain variant [MKA*15]. In bidirectional path tracing [LW93, VG95a], subpaths are traced from both the camera and light, with vertices connected between the two subpaths form complete paths. Since a path can be generated by multiple sampling strategies depending on how two subpaths are connected, one needs to weigh the contribution of the paths properly, which is usually given by multiple importance sampling [VG95b].

To perform bidirectional path tracing in the gradient domain, one might consider simply applying manifold exploration shift mapping to each complete light path in a family of connected subpaths. While this approach is straightforward, Manzi et al. [MKA*15] showed that this is computationally too costly. They thus proposed to ignore connections that involve specular light transport, for the following reason. Consider a complete base path that we can shift using manifold exploration [JM12]: using only diffuse-diffuse vertex connections to generate base paths, we can *guarantee* that reconnection always occurs at the third diffuse vertex on the complete light path, counting from the camera towards the light, denoting these first three diffuse vertices as x_a, x_b, and x_c (see Figure 6; with the vertex, x_a always on the camera). By carefully weighting path sampling strategies using multiple importance sampling, we can still obtain unbiased estimators even if we ignore some connections. Manzi et al. classified the strategies according to three cases:



(a) Gradient-domain path tracing

(c) Gradient-domain photon density estimation

(b) Gradient-domain bi-directional path tracing

(d) Gradient-domain vertex connection and merging

Figure 6: *Base and offset paths in typical algorithms of gradient domain light transport simulation. For clarity, we demonstrate the shift in slightly different scene setups, when needed, with shift mapping labeled when it occurs. A combination of shift mappings is used: half-vector copy (H), reconnection (R), and manifold exploration (M). (a) Half-vector copy shifts the specular chain before reconnection occurs. (b) Bidirectional path tracing always has reconnection at the third connectable vertex, i.e., vertex c. The dashed line represents where vertex connection occurs. (c) The camera subpath is shifted with half-vector copy, and the light subpath is shifted with manifold exploration. (d) Vertex connection and merging technique further considers density estimation when reconnection occurs.*

1. vertices x_b and x_c are sampled on the camera subpath,
2. a vertex x_b is the last vertex of the camera subpath, or
3. light tracing, i.e., connections from the light subpath to the eye.

Figure 6 illustrates that we can cache the result of shift mapping and its Jacobian determinant for the camera subpath ending at x_b and the camera subpath ending at x_c in the first and second cases. Only paths in the third case have to be shifted separately, which is more expensive.

We emphasize that this approach by Manzi et al. is not the only manner in which one can implement gradient domain bidirectional path tracing. A simpler alternative proposed in their work is to only shift the camera subpath, and then attempt to connect it to the base light subpath, i.e., only performing case two from Figure 6b. They, however, showed that this simple approach is less efficient in practice. Another alternative is to apply Russian roulette, probabilistic connection [PRDD15], or the matrix formulation [CBH*18] to pick a path among all possible connections and then shift only this path. Further investigations and evaluations could lead to more insights and lightweight implementations of bidirectional path tracing integrators in the gradient domain.

4.3. Photon Density Estimation

An important class of integrators in light transport simulation is based on photon density estimation [Jen01, HJ09]. This class of integrators is well known to be able to sample complex light transport such as specular-diffuse-specular (SDS) interactions.

Hua et al. [HGNH17] proposed gradient-domain photon density estimation using a hybrid shift mapping to separately treat the camera subpath, the density kernel, and the light subpath (Figure 6(c)). Their shift mapping comprises three steps: first, the offset camera subpath is computed, followed by the offset photon, and finally the offset light subpath. Offset photons are generated such that the relative position between a gather point and photon remains unchanged during shift mapping. Gradient-domain photon density estimation is compatible with progressive radius reduction schemes [HJ09] and leads to a consistent estimator. Gruson et al. [GHV*18] later extend this algorithm to volume rendering, on which we elaborate in Section 7.1.

4.4. More Advanced Algorithms

Sun et al. [SSC*17] extended gradient-domain bidirectional path tracing with photon density estimation under the unified light trans-

port framework [GKDS12,HPJ12]. They built upon the bidirectional integrator by Manzi et al. [MKA*15]. The difference is that, in addition to vertex-vertex connection, density estimation can also be applied at connectable vertices to generate complete paths. As suggested by Manzi et al. [MKA*15], the shift mapping used in Sun et al. [SSC*17] can be narrowed down to a few cases, each of which can be implemented efficiently. Figure 6(d) illustrates such cases and relates them to the corresponding cases in gradient-domain bidirectional path tracing [MKA*15]. The photon density estimation component of Sun et al. [SSC*17] is quite similar to that in gradient-domain photon density estimation [HGNH17], except that they did not conserve the relative photon positions in the kernel during shift. This difference becomes negligible when the kernel size shrinks during progressive photon density estimation.

It would be interesting to see more sophisticated integrators being ported to the gradient domain to achieve even more robust gradient sampling. For example, as mentioned by Sun et al. [SSC*17], a potential extension is to bring Markov chain Monte Carlo with bidirectional estimators [ŠOHK16] to the gradient domain. Another interesting extension is to establish connections from local path space perturbation techniques for Markov chain Monte Carlo [JM12, KHD14, HKD15] to shift mapping since such techniques could explore the path space that corresponds to neighbor pixels.

4.5. Comparative Study

To paint a clearer picture of how different gradient-domain light transport algorithms work, we gathered all the publicly available implementations of gradient-domain light transport algorithms into the same framework, and performed a comprehensive equal-time comparison. The resulting framework is publicly available at `https://github.com/gradientpm/gradient-mts`. Shift mapping of each one of the light transport algorithms is as described in their original publications. Figure 7 shows the results of our numerical experiments. We selected the SPONZA, BATHROOM, BOOKSHELF and TORUS scenes, and rendered each scene with primal- and gradient-domain light transport algorithms. We use the same reconstruction technique for all the algorithms based on the L1-norm Poisson reconstruction shown in Equation 5. The error metric is relative MSE. In the BOOKSHELF scene, we discarded 0.01% of the highest error pixels in the reconstructed image due to the corruption from strong light paths that reach the light through specular-only interactions. Note that this only affects the metric, not the reconstruction itself. Comparisons using the SSIM metric [WBSS04] could be found in our supplemental document. It also includes a relative speed comparison by plotting the time of a single rendering iteration for each technique.

In general, gradient-domain rendering converges faster than the primal-domain counterpart as expected. This observation aligns well with the experiments published in previous work [KMA*15, MKA*15, HGNH17, SSC*17]. The improvement ranges from a few times to an order of magnitude depending on the scene configuration. We generally found that gradient-domain rendering works well when paths can be efficiently sampled and shifted at the same time. In some cases, however, gradient-domain rendering did not outperform traditional Monte Carlo rendering due to high overhead. For example, in the TORUS scene, only a small subset of sampling strategies in VCM is efficient. Performing shift mapping with all sampling strategies makes gradient-domain rendering very expensive, resulting in G-VCM being less efficient than VCM. Sun et al. [SSC*17] also made similar observations in their G-VCM algorithm.

In terms of implementation effort, we found that gradient-domain path tracing [KMA*15] is the easiest to implement. All the other techniques are far more challenging to implement because they are built on top of the manifold exploration shift mapping that requires differential geometry [JM12]. We also speculate integrating gradient-domain algorithms to a production rendering system is not an easy task because of complex material models and custom shaders that have to be made compatible with shift mapping. Investigating how to integrate gradient-domain algorithm into a production rendering system is interesting future work.

5. Image Reconstruction

Image reconstruction is an important step in gradient-domain rendering, after sampling pairs of correlated paths. We summarize the tradeoffs across exiting reconstruction approaches, below. At a high-level, two major reconstructions strategies have been applied to this problem: the first is Poisson image reconstruction with regularization [LKL*13, MVZ16], and the second is iterative optimization based on control variates [RJN16]. We briefly explain these two approaches, as well as relating the control variates approach to a broader class of denoising techniques applied to Monte Carlo rendering [BRM*16, CKS*17, BVM*17, VRM*18]. This relationship has yet to be discussed in the literature and we believe that our new observation here will lead to future, crosspolinating research.

5.1. Poisson Image Reconstruction

As we have learned in Equation 5, solving the final image can be formulated as a Poisson reconstruction problem. The formula was first proposed by Lehtinen et al. [LKL*13] that reconstruct the image from its primal and gradient estimation. To facilitate discussion, we repeat Equation 5 here:

$$I = \arg\min_I \|D_x(I) - G_x\|_n^n + \|D_y(I) - G_y\|_n^n + \lambda\|I - I_0\|_n^n, \quad (18)$$

where n takes the value of 1 or 2 individually to define the norm. With L2-norm, this problem becomes the classical Poisson reconstruction problem [BCCZ08]. To solve this optimization, we can set the derivative of the cost to zero, which is essentially solving the following linear system of equations:

$$(D_x^\top D_x + D_y^\top D_y + \lambda)I = D_x^\top G_x + D_y^\top G_y + \lambda I_0, \quad (19)$$

where the images and gradients here are vectorized to 1D and the functions D_x and D_y are discretized into matrix form. Equation 18 with L1-norm can be solved by casting the optimization as an iterative re-weighted least squares problem [LKL*13].

Kettunen et al.'s frequency analysis [KMA*15] provides insights on how reconstruction with gradients can improve convergence: their work shows that gradient sampling has lower variance than intensity sampling due to the variance being proportional to the signal's energy. More accurate gradient estimates translate to better high-frequency signals when the gradients are integrated to reconstruct

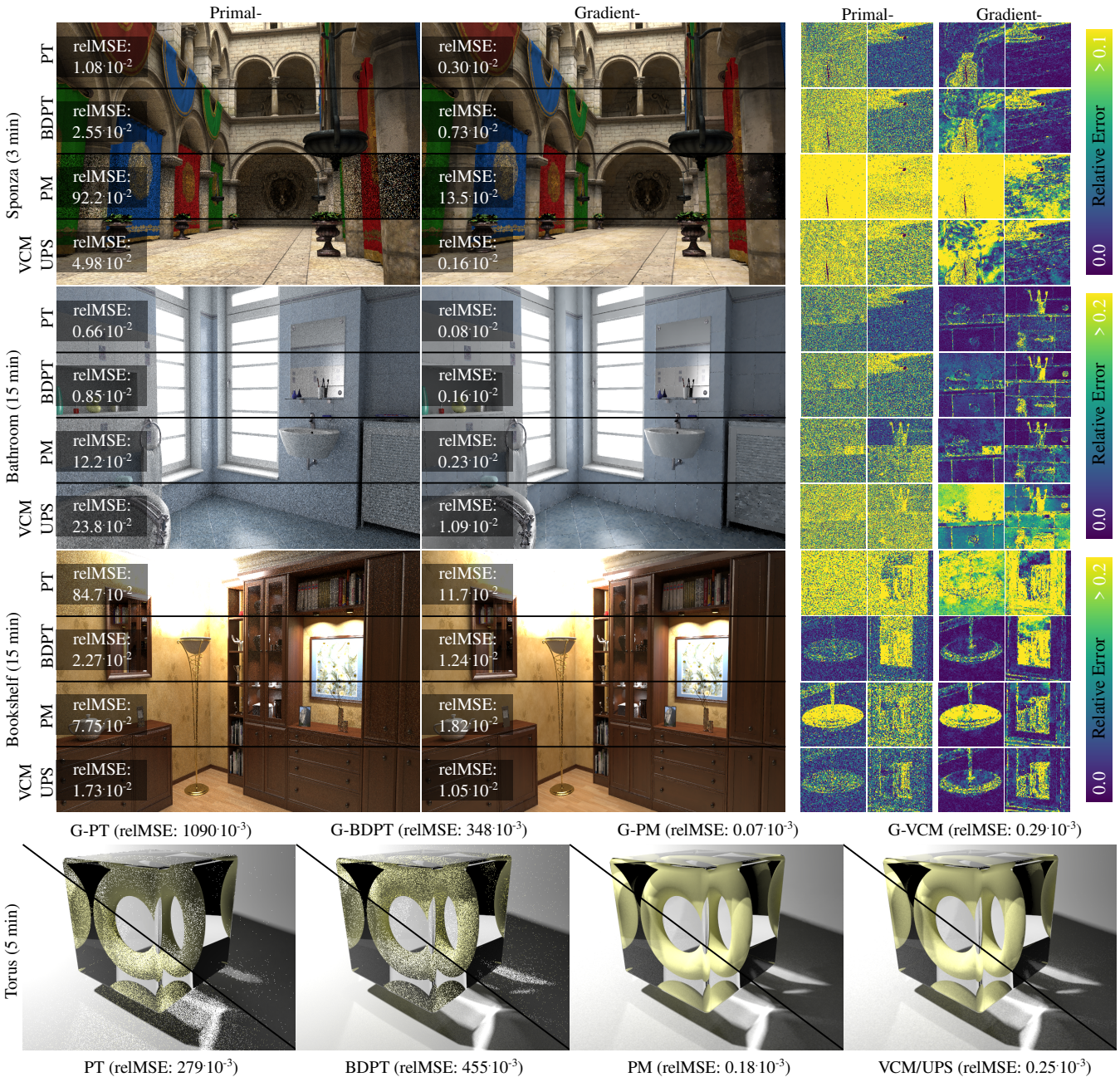


Figure 7: *Equal-time rendering of primal- and gradient-domain methods for the* SPONZA, BATHROOM, BOOKSHELF *and* TORUS *scenes. For scenes dominated by diffuse light transport, such as* SPONZA, *gradient-domain methods significantly outperform their primal counterparts. In more complex scenarios such as the* BATHROOM *and* BOOKSHELF *scenes, the performance of gradient-domain methods depends on: (1) how efficient is the primal counterpart to render the scene and (2) the cost of performing shift mapping. For scenes containing a lot of SDS paths, such as the* TORUS *scene, the methods using photon density estimation outperformed the others. For all the rendering results, please refer to our supplemental document.*

the image. However, as gradients do not contain much information about the low frequency component of the signal, it is generally better to recover this component from a primal-domain simulation, which results in the Poisson reconstruction in Equation 18.

In general, L2-norm reconstruction is more convenient to implement, but it is quite sensitive to outliers. A spike in the gradients can yield a local bright spot in the image because the reconstruction step tries to explain such an erroneous gradient via final pixel

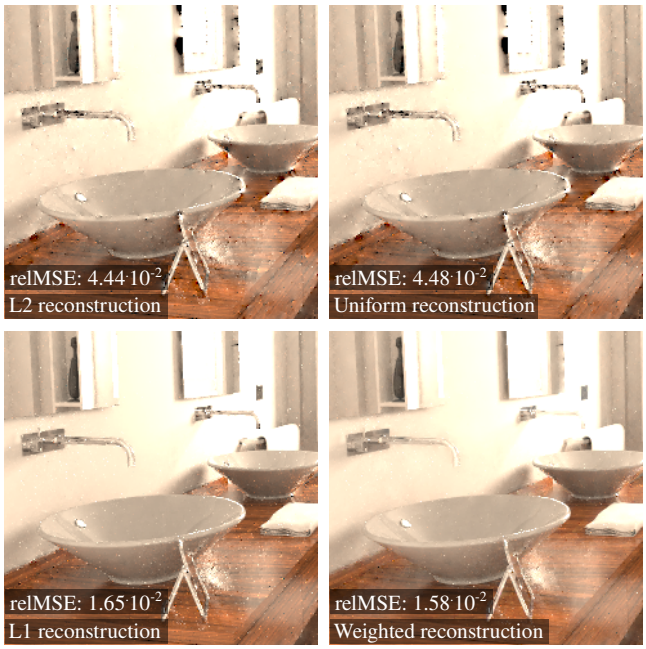


Figure 8: *Comparison of image reconstruction approaches. L2-norm Poisson reconstruction [LKL*13] and uniform control variates [RJN16] results in visual artifacts. L1-norm Poisson reconstruction is more robust in discarding the visual artifacts but exhibit strong energy loss, i.e., the image might appear darker. Weighted control variates [RJN16] performs the best here despite of some minor artifacts near the edges.*

intensities. L1-norm reconstruction is more robust in this situation because it this large error influences the norm by its absolute value instead of the squared value. Manzi et al. [MVZ16] proposed to add local patch constraints using auxiliary images such as albedo and normal maps to regularize Poisson reconstruction results. Such a constraint is similar to a local linear model used in Monte Carlo image denosing [BRM*16] and is generally more robust.

5.2. Iterative Optimization

Rousselle et al. [RJN16] proposed a different image reconstruction approach based on iterative optimization. As mentioned in Section 2, the basic idea is to rewrite the pixel intensity I_p as the sum of the difference and the intensity of the neighbor pixel I_q. Generalizing Equation 10, we have

$$\langle I_p \rangle = \sum_{q \in \mathcal{N}_p} w_{p,q} (\langle I_q \rangle + \langle I_p - I_q \rangle), \tag{20}$$

where $\mathcal{N}_p$ is the set of pixels neighboring to p, $w_{p,q}$ is the weight that defines how good pixel p can be approximated by its neighbor q, and $\langle\rangle$ denotes a Monte Carlo estimator. This reformulation leads to a simple algorithm in which a pixel iteratively receives a contribution from their neighbor pixels (see Algorithm 1). Using uniform weights as in Equation 14, this approach in fact yields visually similar results as Poisson reconstruction with L2-norm (see Figure 8), although the filters to distribute the energy are different in the frequency domain.

It is possible to further reduce visual artifacts in the reconstruction by using the image and gradient variance to weigh the control variates. Rousselle et al. proposed to use the following weight:

$$w_{p,q}^{(n)} = \frac{\sigma_{p,q}^{-2}}{\sum_{r \in \mathcal{N}_p} \sigma_{p,r}^{-2}}, \tag{21}$$

where $\sigma_{p,q}^2$ is the variance that corresponds to the shift from neighbor q, which can be estimated by

$$\sigma_{p,q}^2 = \mathrm{var}(I_p^{(n)}) + \mathrm{var}(\langle I_q - I_p \rangle). \tag{22}$$

Note that they use $\mathrm{var}(I_p^{(n)})$ in place of $\mathrm{var}(I_q^{(n)})$ to reduce energy loss in the reconstruction image. As the pixels are iteratively updated, the variance of $I_p^{(n)}$ is approximated by the variance of the primal-domain value I_p divided by a constant value (see [RJN16] for details). Figure 8 shows an example that demonstrates the robustness of each reconstruction technique.

One challenge for weighted reconstruction is to have variance of the primal image and the gradients. For path tracing, unbiased variance estimation could be easily performed by using an online variance estimator [Knu97]. For progressive photon mapping, as each gather point needs to keep track of kernel radius reduction, variance can by approximated by taking the difference of two independent buffers, each of which accumulates separate photon passes. Such two buffers are also useful for collaborative filtering in Monte Carlo denoising [BRM*16]. A more systematic approach for estimating variance in photon mapping is to estimate and remove bias from the intensity before applying variance estimation [HJJ10].

Another approach to reducing visual artifacts is to perform gradient resampling [BYM18]. The idea is to perform some independent runs of primal and gradient estimation, and then for each pixel, randomly sample the results from the runs and take the average of the sampled values for image reconstruction. This approach decorrelates spikes that both appears in the primal and gradient domain, and thus reducing artifacts.

5.3. Relationship to Denoising

Filtering noise of Monte Carlo rendering by post processing is becoming increasingly popular in film and animation production. Such techniques often exploit the correlation between the image and feature vectors from the auxiliary buffer such as albedo, normal, and depth map for filtering. We discuss the relevance of gradient-domain reconstruction with existing models for general image-space filtering.

The general idea of denoising Monte Carlo rendering is to determine a function that maps a *feature vector* to a color value of each pixel. The feature vector could be defined using noisy pixel values and auxiliary information such as albedo, normal, and depth within a neighboring region around each pixel. Let F be such a function such that $F_p = F(y_p)$ is the denoised pixel intensity and y is the feature vector at pixel p. Let I be the noisy image so that $I_p = F_p + \varepsilon$ where ε is the noise. The zero-order model assumes that each pixel can be written as a weighted sum of neighboring values, i.e., $F_p = \sum_{q \in \mathcal{N}_p} w_{p,q} F_q$, which is equivalent to the following

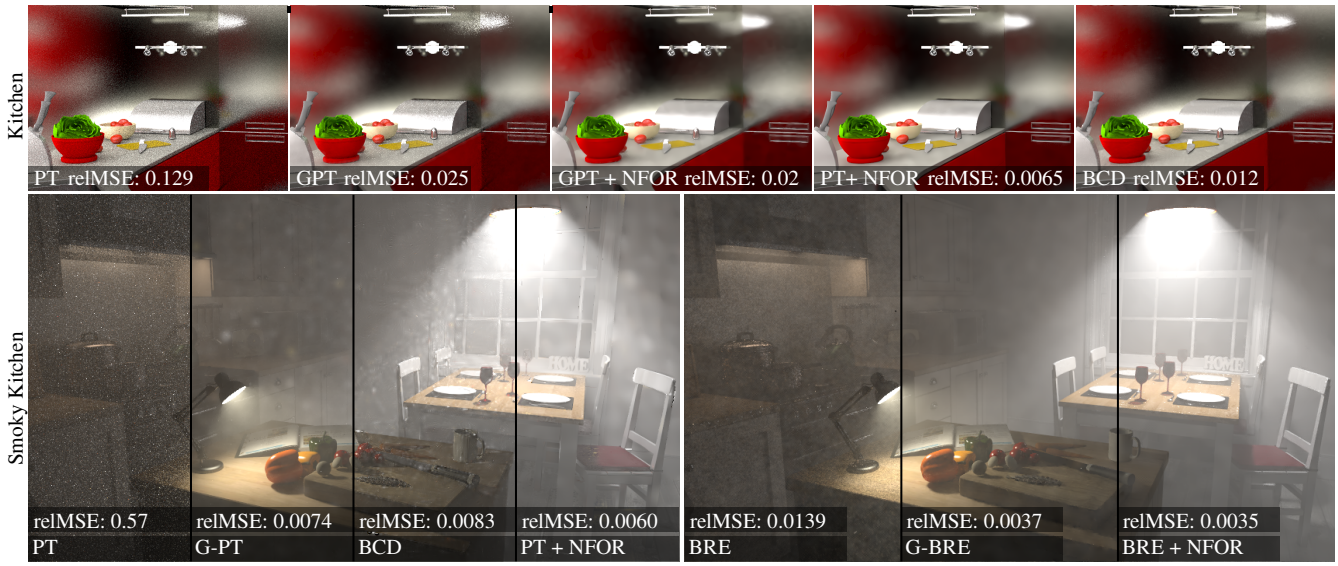


Figure 9: *Comparison between gradient-domain reconstruction and Monte Carlo denoising. For surface rendering, gradient-domain rendering is less efficient than Monte Carlo denoisers that use auxiliary buffers (NFOR [BRM*16]) or histograms of path samples (BCD [BB17]). NFOR could also be applied to address noisy regions remaining in gradient-domain path tracing by using the reconstructed image as guiding features, leading to improved image quality (see G-PT + NFOR in the* KITCHEN *scene). For volume rendering, gradient-domain rendering is comparable to Monte Carlo denoisers, particularly with photon density estimation.*

optimization for denoising:

$$\min_{F_p} \sum_{q \in \mathcal{N}_p} w_{p,q} (F_p - I_q)^2 , \tag{23}$$

where $w_{p,q}$ could be defined using strategies such as bilateral filtering, non-local means, and joint filtering with auxiliary buffers [RMZ13, BRM*16]. The optimization is done at each local image patch centered at each pixel to recover the pixel value F_p. The neighborhood $\mathcal{N}_p$ is a window of 19×19 pixels [BRM*16]. The first-order model further considers the derivative $\nabla F_p = \nabla F(y_p)$ with respect to features [MCY14, BRM*16]:

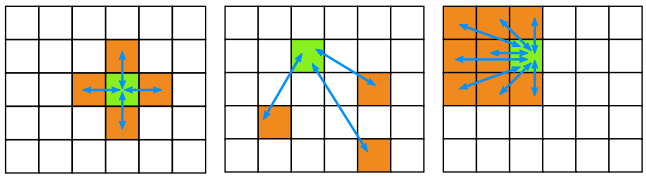
$$\min_{F_p, \nabla F_p} \sum_{q \in \mathcal{N}_p} w_{p,q} \left(F_p + \nabla F_p^\top (y_q - y_p) - I_q \right)^2 . \tag{24}$$

The term $\nabla F_p^\top (y_q - y_p)$ belongs to the first-order Taylor expansion that approximates F_q using F_p and its derivative ∇F_p. By solving F_p and ∇F_p, the entire local image patch can be reconstructed. The patch can then be accumulated to yield the final image.

From the optimization above, one may expect $F_p \approx I_q - \nabla F_p^\top (y_q - y_p)$. This would correspond to the filtered pixel p being approximately equal to a neighbor noisy pixel q compensated by a difference term. This interpretation is in fact related to the control variate in Equation 20. A subtle difference is that general filtering works in a window of roughly 19×19 pixels [BRM*16] while the window in control variates reconstruction [RJN16] is limited to direct neighbors, i.e, left, right, top, and bottom due to the extra cost of performing shift mapping to pixels that are further away. The control variates approach compensates for this limitation by performing the reconstruction iteratively to propagate the filtering effects to a larger neighborhood.

The current thinking is that gradient-domain reconstruction is not yet as effective as image denoising [RJN16]. Based on the above-mentioned relationship between control variates and denoising, we conjecture that this is due to denoising techniques often utilizing more features (e.g., from auxiliary buffers) than gradient-domain reconstruction. Such feature maps are usually highly correlated with the final image, particularly for surface rendering, making them effective candidates to define filtering. Manzi et al. [MVZ16] in fact proposed an extended Poisson reconstruction technique to include local patch constraints derived from feature buffers. Another possibility is to use the gradient-domain reconstruction as a yet another feature map with Monte Carlo denoising techniques to filter the primal-domain image. Back et al. [BYM18] apply a similar idea in the context of adaptive sampling.

Figure 9 compares gradient-domain reconstruction and Monte Carlo image denoising. Gradient-domain reconstruction is an effective way to reduce noise but not comparable to Monte Carlo denoisers in surface rendering. To fill this gap, one can post-process gradient-domain rendering with Monte Carlo denoisers, e.g., using NFOR [BRM*16] by treating the reconstruction image as an additional feature map [BYM18]. This approach does not require any auxiliary buffers such as albedo map. In volume rendering, gradient-domain rendering, particularly with photon density estimation, gradient-domain rendering is as competitive as Monte Carlo denoisers. We generally found that denoisers that use feature maps [BRM*16] work better than those that use sample histogram [BB17]. We believe that filling the gap between image reconstruction in the gradient domain and Monte Carlo image denoising is an interesting research avenue.



(a) Direct neighbors (b) Non-contiguous pairs (c) Bigger tile

Figure 10: *Shift mapping in gradient-domain estimation can be extended beyond traditional neighbors. (a) Gradients can be estimated with shift mapping from a base pixel to its four direct neighbors [LKL*13, KMA*15, MKA*15, HGNH17, SSC*17, GHV*18]. (b) Finite difference between spread-out pixels with similar intensities can be estimated by shifting between two pixels in the pairs [MRK*14]. (c) Gradients can be estimated by considering all possible shift mapping using all the pixels inside a tile [BPE17].*

6. Advanced Gradient Sampling

This section goes beyond the basic algorithms for gradient-domain rendering. In general, the original gradients refers to the finite differences of two neighboring pixels. This can be generalized to the finite difference of two arbitrary (not necessarily neighboring) pixels in the image space. We first explain the ideas of gradient estimation with a group of non-contiguous pixels and a pixel tile (illustrated by Figure 10). We then explain advanced sampling techniques that can further reduce variance in gradient-domain rendering including Markov Chain Monte Carlo methods and adaptive sampling.

6.1. Sampling Pixel Difference

Non-contiguous Pixel Pairs. Manzi et al. [MRK*14] investigated the idea of computing gradient estimates between two non-adjacent pixels. This pair of pixels is chosen to improve the efficiency of shift mapping by selecting pixels with similar measurement contribution functions. The idea is to detect similar pixel values using feature maps and compute finite difference for such pixels. To find the pixel pairs, a hand-crafted affinity metric is derived from a set of feature maps. This set of feature map contains normals, depth and albedo information. This is similar to what is used in image-space denoising techniques as they assume that feature buffers are generally in correlation with the final image. For each feature map, a weight is computed similarly to a denoising method [BRM*16]. The weights are concatenated and the N best pixels are extracted to determine where to shift the paths.

Larger Neighborhood. Bauszat et al. [BPE17] proposed to perform shift mapping between all the pixels in a tile to improve the efficiency of gradient sampling. They formulated this idea under the framework of path reusing [BSH02]: the base paths in a pixel are allowed to shift to all the other pixels within a rectangular tile, resulting in more accurate primal- and gradient-domain estimates. While in traditional path reusing, shifting paths inside a rectangular tile results in visible artifacts in the final image, this does not happen in gradient-domain rendering. The trick is to properly compute the gradients at the tile boundary by making the tiles overlapped by one pixel. This makes tile artifacts less visible in both the gradients and the reconstruction.

With a larger neighborhood, the overhead of shift mapping has to be addressed to make gradient-domain path reusing efficient because now the number of shifts grows proportionally to the total number of pixels in the tile. The evaluation of the optimal tile size [BPE17] shows that using tiles of 6×6 or 8×8 pixels improves the convergence while still having a reasonable overhead. To further improve performance, Russian roulette is performed before actual shift mapping is done in order to avoid shifting paths that do not contribute much.

6.2. Advanced Sampling

Markov Chain Monte Carlo Methods. Gradient-domain rendering can also be applied with Markov Chain Monte Carlo (MCMC) methods. In fact, the seminal work of gradient-domain rendering is about gradient-domain Metropolis light transport by Lehtinen et al. [LKL*13]. Instead of concentrating samples on pixels with higher intensity as in traditional Metropolis light transport, it is also possible to concentrate samples on gradients by including gradient magnitudes into the MCMC target function. Manzi et al. [MRK*14] proposed an extension that uses shift mapping on selective pairs of pixels in order to sample image features more accurately. This allows more efficient gradient estimates, which is demonstrated by the improved performance over the original gradient-domain Metropolis light transport [LKL*13].

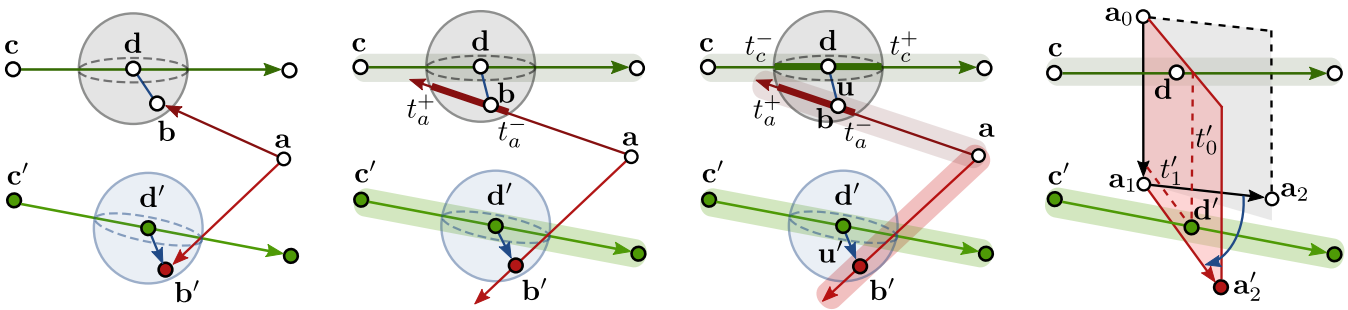
Adaptive Sampling. Rendered images from Monte Carlo estimation often exhibit noise, which can be mitigated by adaptive sampling and reconstruction [ZJL*15]. A common idea of adaptive sampling is to control the sampling densities in the image space, which is applicable to gradient-domain rendering as well. For example, a simple but effective technique is to focus samples to image regions that have large pixel and gradient variance. Instead of modifying the standard gradient-domain light transport algorithms, Back et al. [BYM18] linked gradient-domain path tracing to an off-the-shelf adaptive sampling technique [MCY14] that can handle error estimation and sample distribution. They proposed to treat the reconstructed image as an additional feature map to auxiliary features (e.g., color, normal, and albedo map), and show that such adaptive sampling and reconstruction could benefit gradient-domain path tracing and result in better image quality.

7. Higher Dimensional Integrals

Beyond rendering surfaces, gradient-domain rendering is applicable to higher dimensional integrals, particularly volumetric rendering, animation, and spectral rendering.

7.1. Volumetric Light Transport

Motivated by the efficiency of photon density estimation techniques for surface rendering, Gruson et al. [GHV*18] extended gradient-domain light transport to handle homogeneous participating media. They showed that, for volumetric rendering, gradient-domain path-based techniques are not as efficient as density estimation techniques. They then focused on adopting higher dimensional photon representations including photon points, beams [JZJ08, JNSJ11] and



(a) Point-Point 3D (VPM) (b) Point-Beam 3D (BRE) (c) Beam-Beam 3D (Beam) (d) Plane-Beam 0D (Plane)

Figure 11: *Density estimation techniques for volume rendering [GHV*18]. This diagram illustrates the shift mapping for the gathering kernel. We group the techniques based on the types of photon representation and camera ray query: (a) photon point to camera point, (b) photon point to camera beam, (c) photon beam to camera beam, and (d) photon plane to camera beam. Here all techniques use 3D kernels except photon plane. The shift mapping of such techniques differs mainly in how the camera and light subpath vertices in the kernel are sampled and shifted. By following the ray direction, we denote the last two vertices on the light subpath as* **a** *and* **b**, *and those on the camera subpath as* **c** *and* **d**. *The prime symbol denotes offset vertex, and t is the distance from the ray origin.*

planes [BJ17] for gradient-domain rendering. The resulting algorithms, namely: gradient-domain volumetric photon mapping, beam radiance estimate, photon beams, and photon planes, can handle various scenarios in volumetric rendering and outperform their primal counterparts. Shift mappings used in their algorithms are shown in Figure 11.

Gruson et al. also draw some insights for applying their techniques as follows. First, as stated by Křivánek et al. [KGH*14] for the primal-domain counterpart, rendering with photon beams is more efficient when the medium is thin, i.e., it does not block light from passing through the medium too much. This is generally applicable to volumes such as haze, smoke, or fruit juice. For thick media, such as dense clouds or full-cream milk, rendering with photons is better because the photons mostly scatter around the surface-medium interaction area; very few of them can travel into the medium as most of their energy is absorbed. Second, it is observed that the initial kernel radius for progressive rendering in the gradient domain is not as sensitive as in the primal domain, which enables more robust parameter setups when working in the gradient domain.

While Gruson et al. [GHV*18] showed how each technique performs differently under different configurations, they let the choice of the technique to users. Instead of manually picking the right technique for the right scene, it would be interesting to combine all volumetric photon density estimation techniques for the gradient domain into the same framework. Theoretically, multiple importance sampling that works directly in each individual path space [KGH*14] could be employed.

7.2. Temporal Coherent Gradient Sampling

Manzi et al. [MKD*16] explored animation with gradient-domain rendering. They observed that performing image reconstruction for each animation frame separately leads to flickering. To address this problem, they suggest to further consider shift mapping across two consecutive frames. For shift mapping such as half-vector copy and manifold exploration, one needs to consider the scene geometry in each frame to generate the offset paths, which will cause an additional complexity in the implementation and more memory requirements. Instead, they proposed to employ random number replay so that random samples for base paths in the first frame could be captured for generating the offset paths in the second frame. While this approach might not produce the most coherent path pairs, it still leads to significant improvements in temporal gradient sampling. For this technique, a modified Poisson reconstruction which considers the temporal dimension is required.

7.3. Spectral Gradient Sampling

Spectral rendering is concerned with sampling every wavelength of the spectrum instead of just red, green, and blue components. The goal is to more precisely simulate the light spectrum distribution in a scene and to render specific light phenomena such as dispersion or thin-film materials. After completing the simulation, the spectral distribution can be resampled to red, green, and blue, to display on monitors. The distribution of light intensity across wavelengths forms an additional dimension of the rendering integral. Depending on its wavelength, light reacts differently to some materials such as dielectrics, in particular during refraction. When a light ray intersects with such a *dispersive* material, the ray's wavelength must be sampled since the ray's direction depends on it. Since the ray can only carry this one wavelength afterwards, it only contributes to a small section of the pixel's spectrum, introducing so-called color noise. Consequently, many more samples are necessary for a high-quality simulation.

To improve spectral rendering, gradient-domain light transport can be extended to the spectral domain, as explored by Petitjean et al. [PBE18]. The idea is to estimate the gradients in the 1D wavelength domain by exploiting the correlation of light paths in this domain. Particularly, Petitjean et al. used shift mapping to construct pairs of paths for neighboring wavelengths. Each pair is used to estimate both the light spectrum of a pixel and its gradients. Both are integrated to reconstruct the final image using a fast one-dimensional

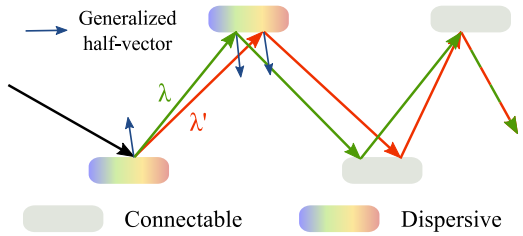


Figure 12: *Extension of half-vector copy for spectral rendering to compute gradients in the wavelength dimension. The offset path is the same as the base path at start. If a dispersive interaction is encountered, the paths diverge. At each successive intersection, the wavelength-dependent generalized half-vector is copied. Reconnection is done as soon as two consecutive connectable vertices are encountered on the base path.*

Poisson reconstruction that can be parallelized over the entire image. To create a shifted path with wavelength λ' from a base path with wavelength λ, Petitjean et al. generalized the half-vector shift to take the dispersion into account. When considering refractions, the half-vector defined by Equation 15 becomes

$$h = \frac{\eta_i \omega_i + \eta_o \omega_o}{\|\eta_i \omega_i + \eta_o \omega_o\|}, \tag{25}$$

where η_i and η_o are the wavelength-dependent indices of refraction of the inside and outside media. To maintain as much correlation as possible between the paths, the paths are split only when the encounter a dispersive surface and are recombined as soon as they reach two consecutive diffuse surfaces. The process of splitting and recombining paths is explained in Figure 12. Two already recombined paths can split again if they reach another dispersive surface - in which case the overhead cost increases.

One limitation of this current approach is that it does not consider image-space gradients simultaneously. Similar to temporal rendering [MKD*16], an interesting extension would be to further consider image-space gradients together with wavelength gradients and apply a 3D Poisson solver to gain further noise removal in the image reconstruction.

8. Conclusions and Future Work

Gradient-domain rendering has developed into a promising family of techniques to accelerate Monte Carlo light transport simulations. It exploits path correlations to effectively sample image-space gradients, leading to a low-noise image after reconstruction. We discuss several potential future research in this area.

8.1. Open Problems

More Effective Shift Mapping. We summarized the various shift mapping functions: random sequence replay, reconnection, half-vector copy, and manifold exploration. No single shift mapping method is currently robust enough for all situations, nor sufficiently cheap to apply in all situations. Random sequence replay is generic but the resulting offset path lacks correlation when compared to other shift mappings. Path reconnection is easy to implement but only works with diffuse connections. Half-vector copy cannot search for paths constrained at both endpoints. Manifold exploration is robust but often too costly to implement and execute broadly. As such, designing a cheaper, faster, and more generic shift mapping remains an open problem. Moreover, given a family of shift mapping techniques, choosing the optimal shift mapping technique for a given base path could lead to more effective gradient sampling.

Better Reconstruction. Poisson reconstruction is usually sensitive to noise in the primal or gradient estimates. In contrast, denoising techniques can robustly handle noise based on supplemental information such as feature maps and sample histograms. Recent denoising methods leverage machine learning [CKS*17, BVM*17, VRM*18], with promising performance and quality. Combining recent machine learning techniques and Poisson reconstruction may lead to better and more robust reconstruction solutions.

Higher-Order Derivative. The current gradient-domain techniques focus only on gradients, which is the first order image derivative. A recent work in image reconstruction [LSR18] shows that a higher order derivative (e.g. Laplacian) could be useful to reconstruct the image. Designing a shift mapping for a low-variance estimation of higher order derivatives could be an interesting research direction. In addition, adaptively selecting gradient or higher order derivatives may lead to more efficient estimators.

Unifying Gradient Sampling Techniques in Volumetric Rendering. Given the derivation of gradient-domain volumetric rendering using photon points and beams [GHV*18], gradient-domain variant of the *unified photon point and beams* method for simulating volumetric light transport [KGH*14] may lead to some non-obvious theoretical challenges, while also necessitating solutions to complex renderer software engineering/architecture challenges. Here, it remains unknown when combining all such estimators using multiple importance sampling leads to better results (e.g., in an equal-time comparison) since the individual estimators in the gradient domain could be too costly to exhaustively evaluate.

Heterogeneous Participating Media Rendering. With some careful engineering surrounding the treatment of ray-medium interface interactions, gradient-domain path tracing [KMA*15] may be adopted to rendering smoke with spatially-varying density. In contrast to homogeneous media, heterogeneous media precludes the evaluation of transmittance along a light path using a closed-form formula. Instead, one must resort to biased approximations (i.e., the ray-marching realization of Simpson's integration rule) or sampling-based solutions, in order to perform this evaluation.

A preliminary result of the SMOKE scene is shown in Figure 13. As can be seen, the direct extension of gradient-domain volumetric path tracing produces smoother image than path tracing. This promising result hints at interesting future work to address the remaining but fundamental issues: bringing well known delta tracking techniques [NGHJ18] to the gradient domain to achieve unbiased transmittance estimates, which will require in-depth investigations. The challenge here is that the probability of the offset paths can no longer be evaluated in closed form in this case, and to use multiple importance sampling, one has to resort to some path probability density approximations.

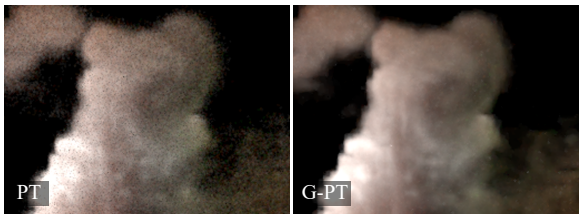


Figure 13: *A direct extension of gradient-domain path tracing to rendering heterogeneous participating media results in a promising result, with noise in the primal domain effectively removed. The transmittance is approximated with ray marching. Supporting delta tracking for unbiased transmittance evaluation in the gradient domain could be an interesting future work.*

Path Guiding. Another family of techniques that improves Monte Carlo rendering is path guiding [Jen95, VKv*14, MGN17, RHJD18]. While path guiding is generally orthogonal to gradient domain rendering, having it could lead to further convergence improvement. One idea is that shift mapping might benefit from path guiding so that offset paths can be contributive in challenging light transports. More research into this direction is desirable.

Gradient-Domain Light Transport in Production. With recent promising results, bringing gradient-domain light transport algorithms to production would be an exciting next step. Several gaps remain to be filled, e.g., to improve the performance of gradient sampling in production scenes with many fine geometric or texture details, to investigate the rendering with terrains, trees, or particle systems. The complexity of making an existing production path tracer compatible to gradient-domain light transport is also worth a thorough discussion.

8.2. Implementation Issues

Beside the theoretical improvement, in practice, there remains work to be done to improve the robustness of gradient-domain rendering algorithms. We discuss them below.

BSDF Classification. Existing gradient-domain integrators depend on the classification of BSDFs at path interactions to decide which shift mapping strategy could be used. This greatly affects the efficiency of gradient sampling. For example, for a pair of vertices, if the classification predicts both of them as connectable (diffuse interactions), vertex reconnection could be applied and the offset path follows the base path from subsequent vertices. Nevertheless, if both vertices are non-connectable (specular interactions), half-vector copy and manifold exploration could be used. The classification used in existing open-source implementations [KMA*15, MKA*15, HGNH17] is simply to check the BSDF roughness value with a threshold. This obviously becomes problematic when the BSDF has spatially varying roughness, or it is unclear whether the BSDF is specular or diffuse. A possible solution is to replace BSDF roughness with BSDF sampling density in the classification. Hereby, we can handle specularity by simply checking the sampled vertex probability with zero.

Multi-Component BSDF. While classifying the roughness of a single-component BSDF is straightforward, several real-world surface materials are often complex: e.g., some could have layered coatings, each of which commonly modelled as a potential multi-lobe BSDF. There are several strategies to sample and evaluate multi-component BSDFs, such as considering all components, or sampling a component in each vertex interaction. Such sampling and evaluation might, however, interfere with classification and affect path sampling, e.g., whether a path should continue to bounce or terminate. In the case that all components in the BSDF are considered, classification can be done based on the lowest roughness value across all components. The advantage of this strategy is that it is simpler to evaluate offset paths in shift mapping. The trade-off is that there might be more variance when estimating the contribution of each component because the components are forced to be sampled the same way. Gradient-domain uni- and bi-direction path tracing [KMA*15, MKA*15] use this strategy.

When a BSDF contains a diffuse and a glossy component, there might be no good way to decide whether to continue recursing (bouncing) or to terminate a path: here, the diffuse component might be sampled efficiently when the path stops, favoring connections to light sources, whereas the glossy component is more efficiently sampled when the path continues bouncing. In this case, handling each component separately by stochastically choosing a single BRDF component at each interaction is preferable. The advantage of this strategy is that the classification can be more fine-grained and applied to each component separately. For best correlation in shift mapping, we can enforce the sampled component of the current vertex on the base and offset path to be the same. The trade-off is that now the BSDFs implementation and path evaluation has to be modified to account for component sampling events. Gradient-domain photon density [HGNH17] and gradient-domain vertex connection and merging [SSC*17] use this strategy.

It could be interesting to perform some empirical tests to see which strategy is better for multi-component BSDFs. Beyond parametric BSDF models, it is unclear whether shift mapping can be performed with vertex interactions that are defined by programmable shaders. More investigations into this direction is needed to draw deeper insights into designing a robust material system for gradient-domain rendering.

Acknowledgment. We are grateful to all the authors of gradient-domain rendering for making their source code available, and the anonymous reviewers for their detailed and constructive comments. We thank Nicolas Vibert for helping with early experiments of image denoising techniques, Jamorn Sriwasansak for proofreading the paper. Special thanks should go to Wenzel Jakob for the Mitsuba renderer, and the following researchers and artists for making the scenes used in this paper: Tiziano Portenier (Bookshelf, Bottle, and Blue Bathroom), MrChimp2313 (House), Mareck (Bathroom), Marko Dabrovic (Sponza), JayArtist (Kitchen). The Red Kitchen, Blue Bathroom, Bookshelf scene are also a part of the Evermotion Archinteriors vol. 1 package. This project was partly funded by JSPS KAKENHI grant numbers 15H05308 and 17K19958, and Swiss National Science Foundation grant number 163045.

References

[ACR05] AGRAWAL A., CHELLAPPA R., RASKAR R.: An algebraic approach to surface reconstruction from gradient fields. In *International Conference on Computer Vision* (2005). 2

[ALLD17] ANDERSON L., LI T.-M., LEHTINEN J., DURAND F.: Aether: An embedded domain specific sampling language for monte carlo rendering. *ACM Transactions on Graphics* (2017). 7

[AR07] AMIT A., RAMESH R.: Gradient domain manipulation techniques in vision and graphics. In *International Conference on Computer Vision* (2007). 2

[BB17] BOUGHIDA M., BOUBEKEUR T.: Bayesian collaborative denoising for monte carlo rendering. In *Computer Graphics Forum* (2017). 1, 12

[BCCZ08] BHAT P., CURLESS B., COHEN M., ZITNICK C. L.: Fourier analysis of the 2d screened poisson equation for gradient domain problems. In *The European Conference on Computer Vision* (2008). 3, 9

[Bha09] BHAT P.: *Gradient-Domain Techniques for Image and Video Processing*. PhD thesis, University of Washington, 2009. 2

[BJ17] BITTERLI B., JAROSZ W.: Beyond points and beams: Higher-dimensional photon samples for volumetric light transport. *ACM Transactions on Graphics* (2017). 14

[BPE17] BAUSZAT P., PETITJEAN V., EISEMANN E.: Gradient-domain path reusing. *ACM Transactions on Graphics* (2017). 1, 13

[BRM*16] BITTERLI B., ROUSSELLE F., MOON B., IGLESIAS-GUITIÃAN J. A., ADLER D., MITCHELL K., JAROSZ W., NOVÃAK J.: Nonlinearly weighted first-order regression for denoising monte carlo renderings. *Computer Graphics Forum* (2016). 1, 9, 11, 12, 13

[BSH02] BEKAERT P., SBERT M., HALTON J.: Accelerating path tracing by re-using paths. In *Proceedings of the 13th Eurographics Workshop on Rendering* (2002). 6, 13

[BVM*17] BAKO S., VOGELS T., MCWILLIAMS B., MEYER M., NOVÁK J., HARVILL A., SEN P., DEROSE T., ROUSSELLE F.: Kernel-predicting convolutional networks for denoising monte carlo renderings. *ACM Transactions on Graphics* (2017). 9, 15

[BYM18] BACK J., YOON S.-E., MOON B.: Feature generation for adaptive gradient-domain path tracing. In *Computer Graphics Forum* (2018). 1, 11, 12, 13

[CA00] CHEN M., ARVO J.: Theory and application of specular path perturbation. *ACM Transactions on Graphics* (2000). 3

[CBH*18] CHAITANYA C. R. A., BELCOUR L., HACHISUKA T., PREMOZE S., PANTALEONI J., NOWROUZEZAHRAI D.: Matrix bidirectional path tracing. In *Proceedings of the Eurographics Symposium on Rendering: Experimental Ideas & Implementations* (2018). 8

[CKS*17] CHAITANYA C. R. A., KAPLANYAN A. S., SCHIED C., SALVI M., LEFOHN A., NOWROUZEZAHRAI D., AILA T.: Interactive reconstruction of monte carlo image sequences using a recurrent denoising autoencoder. *ACM Transactions on Graphics* (2017). 9, 15

[CSKKA02] CSONKA F., SZIRMAY-KALOS L., KELEMEN C., ANTAL G.: Dependent Tests Driven Filtering in Monte-Carlo Global Illumination. In *Eurographics Short Presentations* (2002). 3

[GBP07] GAUTRON P., BOUATOUCH K., PATTANAIK S.: Temporal radiance caching. *IEEE Transactions on Visualization and Computer Graphics* (2007). 3

[GHV*18] GRUSON A., HUA B.-S., VIBERT N., NOWROUZEZAHRAI D., HACHISUKA T.: Gradient-domain volumetric photon density estimation. *ACM Transactions on Graphics* (2018). 1, 7, 8, 13, 14, 15

[GKDS12] GEORGIEV I., KŘIVÁNEK J., DAVIDOVIČ T., SLUSALLEK P.: Light transport simulation with vertex connection and merging. *ACM Transactions on Graphics* (2012). 9

[HGNH17] HUA B.-S., GRUSON A., NOWROUZEZAHRAI D., HACHISUKA T.: Gradient-domain photon density estimation. *Computer Graphics Forum* (2017). 1, 7, 8, 9, 13, 16

[HJ09] HACHISUKA T., JENSEN H. W.: Stochastic progressive photon mapping. *ACM Transactions on Graphics* (2009). 8

[HJJ10] HACHISUKA T., JAROSZ W., JENSEN H. W.: A progressive error estimation framework for photon density estimation. In *ACM Transactions on Graphics* (2010). 3, 11

[HKD15] HANIKA J., KAPLANYAN A. S., DACHSBACHER C.: Improved half vector space light transport. *Computer Graphics Forum* (2015). 3, 9

[HPJ12] HACHISUKA T., PANTALEONI J., JENSEN H. W.: A path space extension for robust light transport simulation. *ACM Transactions on Graphics* (2012). 9

[Ige99] IGEHY H.: Tracing ray differentials. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques* (1999), pp. 179–186. 3

[JDZJ08] JAROSZ W., DONNER C., ZWICKER M., JENSEN H. W.: Radiance caching for participating media. *ACM Transactions on Graphics* (2008). 3

[Jen95] JENSEN H. W.: Importance driven path tracing using the photon map. In *Rendering Techniques* (1995). 16

[Jen01] JENSEN H. W.: *Realistic Image Synthesis Using Photon Mapping*. A. K. Peters, Ltd., 2001. 8

[JM12] JAKOB W., MARSCHNER S.: Manifold exploration: a markov chain monte carlo technique for rendering scenes with difficult specular transport. *ACM Transactions on Graphics* (2012). 3, 6, 7, 9

[JNSJ11] JAROSZ W., NOWROUZEZAHRAI D., SADEGHI I., JENSEN H. W.: A comprehensive theory of volumetric radiance estimation using photon points and beams. *ACM Transactions on Graphics* (2011). 13

[JZJ08] JAROSZ W., ZWICKER M., JENSEN H. W.: The beam radiance estimate for volumetric photon mapping. *Computer Graphics Forum* (2008). 13

[Kaj86] KAJIYA J. T.: The rendering equation. *ACM Transactions on Graphics* (1986). 2

[KGH*14] KŘIVÁNEK J., GEORGIEV I., HACHISUKA T., VÉVODA P., ŠIK M., NOWROUZEZAHRAI D., JAROSZ W.: Unifying points, beams, and paths in volumetric light transport simulation. *ACM Transactions on Graphics* (2014). 14, 15

[KGPB05] KRIVANEK J., GAUTRON P., PATTANAIK S., BOUATOUCH K.: Radiance caching for efficient global illumination computation. *IEEE Transactions on Visualization and Computer Graphics* (2005). 3

[KHD14] KAPLANYAN A. S., HANIKA J., DACHSBACHER C.: The natural-constraint representation of the path space for efficient light transport simulation. *ACM Transactions on Graphics* (2014). 3, 9

[KLS*13] KOPF J., LANGGUTH F., SCHARSTEIN D., SZELISKI R., GOESELE M.: Image-based rendering in the gradient domain. *ACM Transactions on Graphics* (2013). 2

[KMA*15] KETTUNEN M., MANZI M., AITTALA M., LEHTINEN J., DURAND F., ZWICKER M.: Gradient-domain path tracing. *ACM Transactions on Graphics* (2015). 1, 5, 6, 7, 9, 13, 15, 16

[Knu97] KNUTH D. E.: *The Art of Computer Programming, Volume 2 (3rd Ed.): Seminumerical Algorithms*. Addison-Wesley Longman Publishing Co., Inc., 1997. 11

[KSKAC02] KELEMEN C., SZIRMAY-KALOS L., ANTAL G., CSONKA F.: A simple and robust mutation strategy for the metropolis light transport algorithm. In *Computer Graphics Forum* (2002). 3

[LADL18] LI T.-M., AITTALA M., DURAND F., LEHTINEN J.: Differentiable monte carlo ray tracing through edge sampling. *ACM Transactions on Graphics* (2018). 3

[LKL*13] LEHTINEN J., KARRAS T., LAINE S., AITTALA M., DURAND F., AILA T.: Gradient-domain metropolis light transport. *ACM Transactions on Graphics* (2013). 1, 3, 9, 11, 13

[LSR18] LEIMKÜHLER T., SEIDEL H.-P., RITSCHEL T.: Laplacian kernel splatting for efficient depth-of-field and motion blur synthesis or reconstruction. *ACM Transactions on Graphics* (2018). 15

[LW93] LAFORTUNE E. P., WILLEMS Y.: Bi-directional path tracing. In *Proceedings of Compugraphics* (1993). 7

[MCY14] MOON B., CARR N., YOON S.-E.: Adaptive rendering based on weighted local regression. *ACM Transactions on Graphics* (2014). 12, 13

[MGN17] MÜLLER T., GROSS M., NOVÁK J.: Practical path guiding for efficient light-transport simulation. *Computer Graphics Forum* (2017). 16

[MKA*15] MANZI M., KETTUNEN M., AITTALA M., LEHTINEN J., DURAND F., ZWICKER M.: Gradient-domain bidirectional path tracing. *Proceedings of the Eurographics Symposium on Rendering: Experimental Ideas & Implementations* (2015). 1, 7, 9, 13, 16

[MKD*16] MANZI M., KETTUNEN M., DURAND F., ZWICKER M., LEHTINEN J.: Temporal gradient-domain path tracing. *ACM Transactions on Graphics* (2016). 1, 3, 5, 14, 15

[MRK*14] MANZI M., ROUSSELLE F., KETTUNEN M., LEHTINEN J., ZWICKER M.: Improved sampling for gradient-domain metropolis light transport. *ACM Transactions on Graphics* (2014). 1, 13

[MSKN17] MAGDICS M., SZIRMAY-KALOS L., NEUMANN L.: Gradient-domain pet reconstruction. In *IEEE Nuclear Science Symposium and Medical Imaging Conference* (2017). 2

[MVZ16] MANZI M., VICINI D., ZWICKER M.: Regularizing image reconstruction for gradient-domain rendering with feature patches. *Computer Graphics Forum* (2016). 1, 9, 11, 12

[NGHJ18] NOVÁK J., GEORGIEV I., HANIKA J., JAROSZ W.: Monte carlo methods for volumetric light transport simulation. *Computer Graphics Forum* (2018). 15

[PBE18] PETITJEAN V., BAUSZAT P., EISEMANN E.: Spectral gradient sampling for path tracing. In *Computer Graphics Forum* (2018). 1, 14

[PGB03] PÉREZ P., GANGNET M., BLAKE A.: Poisson image editing. *ACM Transactions on Graphics* (2003). 2

[PRDD15] POPOV S., RAMAMOORTHI R., DURAND F., DRETTAKIS G.: Probabilistic connections for bidirectional path tracing. In *Computer Graphics Forum* (2015). 8

[RHJD18] REIBOLD F., HANIKA J., JUNG A., DACHSBACHER C.: Selective guided sampling with complete light transport paths. *ACM Transactions on Graphics* (2018). 16

[RJN16] ROUSSELLE F., JAROSZ W., NOVÁK J.: Image-space control variates for rendering. *ACM Transactions on Graphics* (2016). 3, 9, 11, 12

[RMZ13] ROUSSELLE F., MANZI M., ZWICKER M.: Robust denoising using feature and color information. *Computer Graphics Forum* (2013). 12

[She94] SHEWCHUK J. R.: *An Introduction to the Conjugate Gradient Method Without the Agonizing Pain*. Tech. rep., Carnegie Mellon University, 1994. 3

[ŠOHK16] ŠIK M., OTSU H., HACHISUKA T., KŘIVÁNEK J.: Robust light transport simulation via metropolised bidirectional estimators. *ACM Transactions on Graphics* (2016). 9

[SSC*17] SUN W., SUN X., CARR N. A., NOWROUZEZAHRAI D., RAMAMOORTHI R.: Gradient-domain vertex connection and merging. In *Proceedings of the Eurographics Symposium on Rendering: Experimental Ideas & Implementations* (2017). 1, 7, 8, 9, 13, 16

[SW01] SUYKENS F., WILLEMS Y. D.: Path differentials and applications. In *Rendering Techniques* (2001). 3

[VG95a] VEACH E., GUIBAS L.: Bidirectional estimators for light transport. In *Photorealistic Rendering Techniques*. Springer, 1995. 7

[VG95b] VEACH E., GUIBAS L. J.: Optimally combining sampling techniques for monte carlo rendering. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques* (1995). 6, 7

[VKv*14] VORBA J., KARLÍK O., ŠIK M., RITSCHEL T., KŘIVÁNEK J.: On-line learning of parametric mixture models for light transport simulation. *ACM Transactions on Graphics* (2014). 16

[VRM*18] VOGELS T., ROUSSELLE F., MCWILLIAMS B., RÖTHLIN G., HARVILL A., ADLER D., MEYER M., NOVÁK J.: Denoising with kernel prediction and asymmetric loss functions. *ACM Transactions on Graphics* (2018). 9, 15

[WBSS04] WANG Z., BOVIK A. C., SHEIKH H. R., SIMONCELLI E. P.: Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing* (2004). 9

[WH92] WARD G. J., HECKBERT P. S.: Irradiance gradients. In *Eurographics Symposium on Rendering* (1992). 3

[WRC88] WARD G. J., RUBINSTEIN F. M., CLEAR R. D.: A ray tracing solution for diffuse interreflection. *ACM Transactions on Graphics* (1988). 3

[ZJL*15] ZWICKER M., JAROSZ W., LEHTINEN J., MOON B., RAMAMOORTHI R., ROUSSELLE F., SEN P., SOLER C., YOON S.: Recent advances in adaptive sampling and reconstruction for monte carlo rendering. *Computer Graphics Forum* (2015). 13