

# Gradient-Domain Path Reusing

PABLO BAUSZAT, VICTOR PETITJEAN, and ELMAR EISEMANN,  
Delft University of Technology, Netherlands



Fig. 1. We compare our method to conventional and gradient-domain path tracing in an equal-time comparison. Gradient-domain path reusing produces visually pleasant images with much less noise than path tracing and significantly lower artifacts than gradient-domain path tracing given the same time.

Monte-Carlo rendering algorithms have traditionally a high computational cost, because they rely on tracing up to billions of light paths through a scene to physically simulate light transport. Traditional path reusing amortizes the cost of path sampling over multiple pixels, but introduces visually unpleasant correlation artifacts and cannot handle scenes with specular light transport. We present *gradient-domain path reusing*, a novel unbiased Monte-Carlo rendering technique, which merges the concept of path reusing with the recently introduced idea of gradient-domain rendering. Since correlation is a key element in gradient sampling, it is a natural fit to be performed together with path reusing and we show that the typical artifacts of path reusing are significantly reduced by exploiting the gradient domain. Further, by employing the tools for shifting paths that were designed in the context of gradient-domain rendering over the last years, we can generalize path reusing to support arbitrary scenes including specular light transport. Our method is unbiased and currently the fastest converging unidirectional rendering technique outperforming conventional and gradient-domain path tracing by up to almost an order of magnitude.

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

Additional Key Words and Phrases: Global illumination, light transport simulation, path reusing, gradient-domain rendering

## ACM Reference format:

Pablo Bauszat, Victor Petitjean, and Elmar Eisemann. 2017. Gradient-Domain Path Reusing. *ACM Trans. Graph.* 36, 6, Article 229 (November 2017), 9 pages. <https://doi.org/10.1145/3130800.3130886>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

© 2017 Association for Computing Machinery.

0730-0301/2017/11-ART229 \$15.00

<https://doi.org/10.1145/3130800.3130886>

## 1 INTRODUCTION

Monte-Carlo ray tracing techniques are today's standard for producing photo-realistic images. One of the most popular techniques is path tracing which accurately simulates the light transport in a scene and renders a large variety of natural phenomena. Path tracing was first proposed in 1986 as an algorithm to solve the rendering equation [Kajiya 1986] and is nowadays widely-used throughout the computer graphics community due to its simplicity and practicality. The rendering equation, a nested recursion of integrals over the space of incoming light, is solved by path tracing using Monte Carlo integration and light path samples are generated by random walks through the scene. Unfortunately, the computational cost of path tracing is very high since up to billions of paths are typically required for an image and several expensive ray-scene intersection tests have to be performed to generate a path. An insufficient number of path samples leads to high variance in the Monte Carlo estimate manifesting itself as image noise and computing smooth images often requires several hours or even days. An abundance of techniques have been developed over the last decades reducing the amount of required path samples, lowering the cost of path sampling, or trading off noise for bias (e.g., by filtering).

In 2002, *path reusing* was introduced as means to amortize the cost of tracing a path over multiple image pixels in an unbiased way [Bekaert et al. 2002]. By sharing path samples between neighboring pixels, the sampling rate is virtually increased at a low cost. Although path reusing has been around for a long time, it has not found wider application in rendering systems yet due to two major drawbacks. First, traditional path reusing is not efficient in scenes with specular or highly-glossy light transport (e.g., scenes containing mirrors or glass) which are prevalent today. Further, sharing paths between pixels in an efficient way involuntarily introduces

correlation, exchanging image noise for visually unpleasant and easily recognizable patterns.

Recently, gradient-domain rendering has been introduced by Lehtinen et al. [Lehtinen et al. 2013] as new concept for image synthesis and received a lot of attention in the research community since then. At its core, the main idea is to *directly* estimate image gradients (finite differences between pixels) in addition to pixel values. Image gradients are known to represent the image content in a sparser way [Ruderman 1994] and by directly sampling gradients using carefully correlated pairs of paths, the estimated gradients exhibit a significantly lower variance than their image domain counterpart. *Gradient-domain path tracing* has been shown to improve the convergence rate of traditional path tracing by up to an order of magnitude [Kettunen et al. 2015].

In this paper, we show that path reusing and gradient-domain rendering share conceptual similarities and by fusing them, we develop a novel powerful rendering algorithm which we term *gradient-domain path reusing*. Our method is based on the insight that the correlation in the path samples introduced by path reusing is actually a desired and essential property for gradient-domain rendering. We demonstrate that performing path reusing in the gradient domain significantly reduces the visually disturbing patterns that hindered traditional path reusing so far. First, we express path reusing more generally in the path-space framework of Veach and Guibas [Veach and Guibas 1997] and propose *generalized path reusing* which abstracts the process of reusing samples with a generic reusing function. Building on top of path shift methods developed in the context of gradient-domain rendering, we design a reusing function that improves performance for highly-glossy materials and supports specular light transport. Following, we apply the concept of path reusing to gradient sampling, leading to a novel gradient estimator which remains unbiased and benefits from sharing path samples between multiple pixels. By correlating paths over image neighborhoods for gradient estimation, we achieve a lower variance leading to improved convergence and perceptually more pleasant images. We evaluate our proposed method on several test scenes and show that it is able to harvest the benefits of both techniques efficiently, improving convergence compared to gradient-domain path tracing by up to almost an order of magnitude.

## 2 PREVIOUS WORK

Rendering photo-realistic images is a long-standing challenge in computer graphics. Monte-Carlo ray tracing methods interpret the incoming light at an image pixel as a random variable which expected value is given by the solution of the rendering equation and estimated using unbiased sampling. They have been established as a powerful approach since their early introduction in the form of distributed ray tracing [Cook et al. 1984] and path tracing [Kajiya 1986]. Since then, they received a lot of interest from researchers and an abundance of improvements and advances have been proposed. To alleviate the rendering cost, techniques either try to reduce the amount of required samples or to lower the individual sampling cost. Seminal advances such as bi-directional path tracing [Veach and Guibas 1995] and Metropolis Light Transport

(MLT) [Veach and Guibas 1997] fall in the first category and reuse sub-paths for multiple light and pixel measurements. Similarly, approaches based on virtual-point lights [Keller 1997] perform reusing for light paths. Techniques such as Russian roulette and splitting [Arvo and Kirk 1990; Vorba and Krivánek 2016] belong to the latter category and reduce the cost of individual sampling by adapting to the sample contribution. Interpolation-based techniques such as irradiance caching [Ward et al. 1988] or photon mapping [Jensen 1996; Shirley et al. 1995] amortize sample cost over multiple image pixels, but introduce bias and we will omit discussing biased rendering techniques (e.g., recent methods depending on image filtering) in the following. For an extensive overview of physically-based Monte-Carlo rendering techniques, we refer to the excellent book by Pharr et al. [2016].

An important milestone was the introduction of the path integral formulation of light transport by Veach and Guibas in 1995 [Veach and Guibas 1995, 1997] which allows to conveniently express the value  $I_j$  of a pixel  $j$  with a single integral over the space of light-transporting paths:

$$I_j = \int_{\mathcal{P}} h_j(\bar{x}) f^*(\bar{x}) d\mu(\bar{x}) = \int_{\mathcal{P}} f_j(\bar{x}) d\mu(\bar{x}) \quad (1)$$

The integration domain  $\mathcal{P}$  is called the *path space* and denotes the union of all paths of finite length along which light can potentially travel through a scene,  $h_j$  denotes the pixel filter function of  $j$  indicating how much the path measurement  $\bar{x} = (\mathbf{x}^0, \mathbf{x}^1, \mathbf{x}^2, \dots)$  (with  $\mathbf{x}^0$  being the image plane vertex) contributes to the pixel, and  $f^*$  is the (*spectral*) *image contribution function* which defines the amount of light that reaches the image through the path. The *path contribution function*  $f_j$  is the product of the filter and image contribution function and gives the contribution of a path specifically to the pixel  $j$ . The function  $f^*$  consists of the product of the bidirectional scattering distribution functions (BSDF), the geometric factors between the path vertices, and the emitted radiance. Finally,  $d\mu$  is the differential of the integral path-space measure which is the product of the surface area measures associated with the integration at each path vertex (see [Veach and Guibas 1997] for more details). Most of today's (unbiased) rendering algorithms can be expressed in this formulation of light transport.

### 2.1 Path Reusing

Path reusing [Bekaert et al. 2002] amortizes the cost of generating path samples by sharing them between nearby pixels. It is based on the insight that a path from one pixel often contributes to its neighboring pixels as well and independently sampled paths can be mutually reused after minor, low-cost changes. As an inexpensive way to create new path samples for a pixel, Bekaert et al. proposed to connect the primary hit point of the pixel's path to the secondary hit points of other path samples when they are mutually visible. Since a path sample can then be generated by more than one pixel, multiple importance sampling (MIS) [Veach and Guibas 1995] is required to re-weight the samples accordingly and account for the change in path density. Path reusing divides the image in distinct tiles, generates a path for each pixel using its associated pixel path sampler, and then defines the unbiased estimator of the pixel value

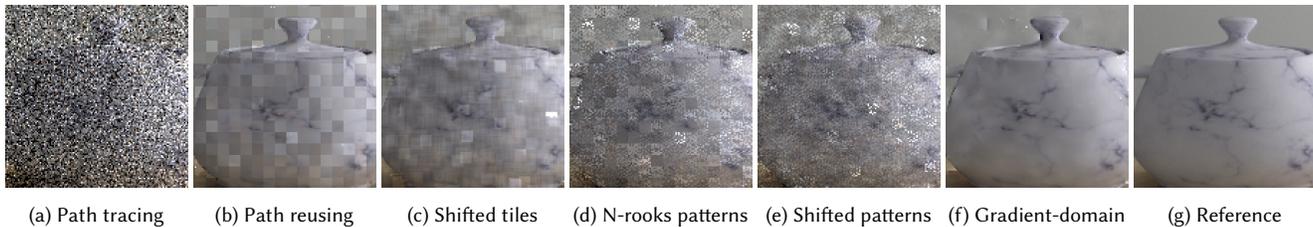


Fig. 2. **Path reusing artifacts.** The figure illustrates common artifacts from path reusing for the VEACH-DOOR scene with 64 samples and a tile size of 8. The inset (a) shows path tracing and (b) - (e) shows combinations of path reusing using tile shifts and four different N-rooks like patterns (see [Bekaert et al. 2002]). Even with extensive scrambling, the correlation remains clearly visible and is perceptually disturbing. Gradient-domain path reusing (f) significantly reduces these artifacts and, while not artifact-free for the low sampling rate chosen for illustration, provides a visually more pleasant image.

$I_j$  of a pixel  $j$  inside the tile as

$$I_j \approx \sum_{k \in \mathcal{T}} w_k(\bar{\mathbf{x}}_{j,k}) \frac{f_j(\bar{\mathbf{x}}_{j,k})}{p_k(\bar{\mathbf{x}}_k)} \quad (2)$$

$$\text{with } \bar{\mathbf{x}}_{j,k} = (\mathbf{x}_j^0, \mathbf{x}_j^1, \mathbf{x}_k^2, \mathbf{x}_k^3, \mathbf{x}_k^4, \dots) \mid V(\mathbf{x}_j^1 \rightarrow \mathbf{x}_k^2) = 1$$

where  $\mathcal{T}$  denotes the set of all the paths in the tile which contains  $j$ ,  $V$  defines the visibility between two surface points,  $p_k$  denotes the probability density function (PDF) for sampling a path with the path sampler of pixel  $k$ , and  $w_k(\bar{\mathbf{x}}_{j,k})$  is the MIS weight. The weight can be freely chosen as long as it fulfills the conditions for multiple importance sampling and traditional path reusing uses the *balance heuristic*:

$$w_k(\bar{\mathbf{x}}_{j,k}) = \frac{p_k(\bar{\mathbf{x}}_k)}{\sum_{m \in \mathcal{T}} p_m(\bar{\mathbf{x}}_{m,k})} \quad (3)$$

Since the path sampling strategies inside  $\mathcal{T}$  remain fixed due to the distinct tiling, the weight from Eq. 3 is constant over the tile for  $k$  and can be computed efficiently.

Unfortunately, the tiling introduces visually unpleasant correlation artifacts and disturbing discontinuities at tile edges. A measure proposed by Bekaert et al. for reducing these artifacts is to reuse paths not in rectangular tiles but in scrambled (N-rooks like) patterns and to shift tiles in-between sample iterations. Further, Xu and Sbert proposed the use of overlapping tiles [Xu and Sbert 2007] to overcome the problems at tile edges. Reusing paths from the camera has also been explored in the context of virtual-point light rendering by Segovia et al. [Segovia et al. 2006] and Davidovic et al. [Davidovič et al. 2010], where the latter further proposed an improved tiling scheme and the use of a linear ramp weight to make tile boundaries less perceivable. Despite all efforts, tiling often remain clearly visible especially in the presence of outliers (see Fig. 2). Another problem of conventional path reusing is that the way new path samples are created becomes less efficient for highly-glossy and specular surfaces because it lacks proper importance sampling of the local BSDFs and generates new path samples with low or even zero contributions. Nonetheless, path reusing is often beneficial in terms of variance reduction because it is typically less expensive to reuse a path sample than generating a complete new one. Path reusing has further been applied to other applications, e.g., for light source animation [Sbert and Halton 2004] and in the context of radiosity [Castro et al. 2008]. Recently, it has also been proposed for

light paths in bi-directional path tracing [Popov et al. 2015] where correlation does not become as obvious as for eye paths.

## 2.2 Gradient-Domain Rendering

Gradient-domain rendering has been recently proposed as an alternative way to reduce the noise of Monte-Carlo ray tracing. It was first introduced in the context of MLT by Lehtinen et al. [Lehtinen et al. 2013] who proposed to trace pairs of paths to estimate image gradient gradients directly and steer the Metropolis sampler more frequently towards image regions where large gradients are located. They derived a mathematical formulation to express the image gradient  $\Delta_{i,j}$  (finite difference) between the pixels  $i$  and  $j$  in a single integration:

$$\Delta_{i,j} = \int_{\mathcal{P}_i} \left( f_i(\bar{\mathbf{x}}) - f_j(T_{i \rightarrow j}(\bar{\mathbf{x}})) \left| T'_{i \rightarrow j} \right| \right) d\mu(\bar{\mathbf{x}}) \quad (4)$$

Here,  $T_{i \rightarrow j}(\cdot)$  is a shift mapping that deterministically maps a path ( $\bar{\mathbf{x}}$ ) originating at pixel  $i$  to a path that connects with the image at pixel  $j$ , and  $\left| T'_{i \rightarrow j} \right|$  is the Jacobian determinant of  $T$  that accounts for the change of integration domain for the pixel  $j$  (from  $\mathcal{P}_j$  to  $\mathcal{P}_i$ ). Since paths are only allowed to be shifted by a spacing of exactly one pixel, the filter functions  $h_i$  and  $h_j$  are identical. For a detailed derivation of the expression, we refer the reader to the original work ([Lehtinen et al. 2013], Sec. 4.1). Another benefit of directly sampling gradients via Eq. 4 is that it allows to correlate the base path samples of pixels  $i$  and the shifted paths for pixel  $k$ . By designing the shift function  $T$  in such a way that a shifted path becomes as similar as possible to its base path, variance from the random sampling process cancels and each gradient sample exhibits a significantly lower variance compared to conventional sampling. Several shift functions have been previously designed to support diffuse, glossy, and specular light transport, and we will rely on these tools for generalizing path reusing.

The formulation in Eq. 4 will only correctly estimate the gradient if the shift mapping is bijective in the path space and the mapped paths from  $\mathcal{P}_i$  cover  $\mathcal{P}_j$ , i. e.,  $\mathcal{P}_j = T(\mathcal{P}_i)$ . In practice, however, the path sampler of pixel  $i$  is not guaranteed to ensure the second property. To circumvent this issue, Manzi et al. [Manzi et al. 2014] proposed to sample the gradients symmetrically using the mappings

from  $i$  to  $j$  and conversely from  $j$  to  $i$ :

$$\Delta_{i,j} = \int_{\mathcal{P}_i} w_{i,j}(\bar{x}) g_{i,j}(\bar{x}) d\mu(\bar{x}) + \int_{\mathcal{P}_j} w_{j,i}(\bar{x}) g_{j,i}(\bar{x}) d\mu(\bar{x}) \quad (5)$$

with  $g_{a,b}(\bar{x}) = f_a(\bar{x}) - f_b(T_{a \rightarrow b}(\bar{x})) \left| T'_{a \rightarrow b} \right|$

The weights  $w_{i,j}(\bar{x})$  and  $w_{j,i}(\bar{x})$  either account for the duplicated appearance of estimates using multiple importance sampling, or handle non-invertible shifts. Manzi et al. [Manzi et al. 2014] further introduced *structure-aware gradients*, i. e., gradients between pixels that are not necessarily adjacent and which locations are selected based on scene features. Structure-aware gradients introduced path shifts over distances farther than one pixel which is also a key element in our method. However, an important distinction to our approach is that structure-aware gradients are still estimated solely from the samples of the two involved pixels. In contrast, our approach estimates the gradients using samples from multiple nearby pixels combined with proper multiple importance sampling.

Unfortunately, MLT typically suffers from an uneven convergence behavior which is further amplified when gradient sampling is used. To show that gradient sampling is also beneficial in the traditional Monte Carlo context, gradient-domain path tracing was introduced by Kettunen et al. [Kettunen et al. 2015] and improved the convergence of its traditional counterpart by up to an order of magnitude. Recently, direct gradient estimation was also applied in the context of bi-directional path tracing [Manzi et al. 2015], photon mapping [Hua et al. 2017], and vertex connection and merging [Sun et al. 2017]. Further, Manzi et al. proposed an extension of gradient sampling for the time domain [Manzi et al. 2016].

After the gradients have been estimated together with a coarse image, the final result is computed using a screened Poisson reconstruction step [Bhat et al. 2010; Pérez et al. 2003]. Lehtinen et al. [Lehtinen et al. 2013] also proposed the use of a L1-norm Poisson reconstruction, which produces smoother results but introduces bias. Recently, Manzi et al. [Manzi et al. 2016] demonstrated how to improve the quality of the Poisson reconstruction step and produce smoother results by involving features from the rendering process (e.g., per-pixel normals, depth values, texture colors, etc.). The Poisson reconstruction can also be replaced with an iterative reconstruction scheme that uses the estimated pixel values and gradients as image-space control variates [Rousselle et al. 2016]. Since these methods build on top of the estimated gradients but do not alter the gradient sampling process itself, we consider them orthogonal to our approach.

### 3 GRADIENT-DOMAIN PATH REUSING

While sample correlation leads to unwanted results in conventional path reusing, it is a welcoming property in gradient-domain rendering and our goal is to define a path reusing estimator for gradient sampling. Unfortunately, conventional path reusing lacks the ability to efficiently correlate paths with primary and secondary hit points on specular and highly-glossy surfaces which limits its applicability. To overcome this issue, we first propose *generalized path reusing* which abstracts the process of sharing path samples using a generic *reusing function*. The advantage of our formulation is that

it allows for more elaborated reusing strategies and we design a reusing function that builds on top of existing shift functions from gradient-domain rendering to fully support scenes with specular and highly-glossy light transport. Finally, we extend generalized path reusing from the image to the gradient domain.

#### 3.1 Generalized Path Reusing

Conventional path reusing is specifically designed for the case of sharing paths at primary hit points and mainly focuses on sample creation. Abstracting the process of sharing paths using a generic function  $R$  that performs the operation of reusing a path allows us to express path reusing more general in the path integral formulation. We define the value  $I_j$  of pixel  $j$  using the function  $R$  as the sum of integrals over the path spaces of all pixels in a tile  $\mathcal{T}$  which contains the pixel  $j$ :

$$I_j = \sum_{k \in \mathcal{T}} \int_{\mathcal{P}_k} w_k(R_{k \rightarrow j}(\bar{x})) f_j(R_{k \rightarrow j}(\bar{x})) \left| R'_{k \rightarrow j} \right| d\mu(\bar{x}) \quad (6)$$

Multiple importance sampling is required for unbiasedness as originally since the path spaces  $\mathcal{P}_j$  and  $\mathcal{P}_k$  for each  $k$  might overlap in certain regions and would be accounted for multiple times. Interestingly, in this expression the function  $R$  closely resembles the shift mapping  $T$  of gradient-domain rendering (although paths are shifted farther than just one pixel). However, the conceptual difference between  $R$  and  $T$  lies in the practical implementation and their way of creating new paths during the Monte-Carlo sampling process. A shift mapping  $T$  maps an input path from one pixel to another, but does not consider the original path sample of the shifted pixel. In contrast, we define the function  $R$  in the discrete case of the Monte-Carlo estimator to create a new path for the pixel  $j$  from the path sample  $\bar{x}_k$  of pixel  $k$  and the original path sample of  $j$ ,  $\bar{x}_j$ :

$$R_{k \rightarrow j}(\bar{x}) = R_j(\bar{x}_j, \bar{x}_k)$$

This definition leads to a more general form of the original path reusing estimator from Eq. 2:

$$I_j \approx \sum_{k \in \mathcal{T}} w_k(R_j(\bar{x}_j, \bar{x}_k)) \frac{f_j(R_j(\bar{x}_j, \bar{x}_k)) \left| R'_{k \rightarrow j} \right|}{p_k(\bar{x}_k)} \quad (7)$$

A "good" reusing function fulfills two criteria: first, it establishes a connection between the paths as early as possible to maximize the amount of work load that is reused, and second, it creates path samples with significant contributions. Interestingly, the first criteria also facilitates correlation which is a desired property for a shift mapping  $T$ . Further, the design of the function  $R$  must allow to efficiently compute the PDF of a newly generated path for each pixel sampling strategies of  $\mathcal{T}$  since it is required for the estimation of the MIS weights  $w_k$ . Conventional path reusing focuses solely on the first quality criteria, but neglect the latter which potentially reduces the efficiency of the estimator (e.g., in the case of specular surfaces where samples with zero contribution will be created). We design an improved version of  $R$  that creates new path samples with potentially higher contributions and improves the power of the estimator.

Originally, path reusing provides only a single strategy for reusing a path which always connects primary and secondary hit points. This process can be modeled by defining  $R$  as

$$R_j(\bar{x}_j, \bar{x}_k) = (\mathbf{x}_j^0, \mathbf{x}_j^1, \mathbf{x}_k^2, \mathbf{x}_k^3, \mathbf{x}_k^4, \dots)$$

when  $\mathbf{x}_j^1$  and  $\mathbf{x}_k^2$  are mutually visible. We propose a new definition for  $R$  that allows more than a single strategy and exploits the power of the shift mappings that have been developed in the context of gradient-domain rendering. The improved function  $R$  creates a path sample from the original vertices of  $\bar{x}_j$  and  $\bar{x}_k$ , and a new series of path vertices  $C = (c_2, c_3, \dots, c_a)$ :

$$R_j(\bar{x}_j, \bar{x}_k) = (\mathbf{x}_j^0, \mathbf{x}_j^1, c_2, c_3, \dots, c_{a-1}, \mathbf{x}_k^a, \mathbf{x}_k^{a+1}, \dots)$$

This definition allows to choose the strategy and the vertex of  $\bar{x}_k$  for the connection (which is indexed by  $a$ ) more freely. For example, imagine that the path  $\bar{x}_k$  has several specular interactions early on (e.g., a directly observed glass or a caustic). In these cases, the improved  $R$  can connect to a later, more diffuse vertex and the resulting path's contribution will be more significant if the vertices of  $C$  are chosen properly. Note that the conventional connection strategy is covered as well and is simply performed by setting  $a = 2$  which would result in  $C$  being empty.

To create the new vertices  $c_2, c_3, \dots, c_a$ , we employ a variation of the shift mapping  $T$ . Our motivation is that shift mappings are designed to create shifted paths with non-zero contributions (which we want to avoid) and they optimize correlation between paths. Shift mappings for gradient-domain rendering typically need to be bijective and this property ensures in our case that the new vertices will be deterministic and conditioned on  $\bar{x}_j$  and  $\bar{x}_k$ . If the strategy of choosing the vertices  $c_2, c_3, \dots, c_a$  would not be deterministic or bijective, computing the PDF of a new path would require a costly marginalization which is infeasible in practice. Since the mapping  $T$  is originally defined to shift an image plane position of a path, we have to slightly modify it to perform a shift between primary hit points instead. Using a variant of  $T$  which maps the sub-path of  $\bar{x}_k$  starting at the primary hit point  $\mathbf{x}_k^1$  to originate at  $\mathbf{x}_j^1$ , we define

$$(\mathbf{x}_j^1, c_2, c_3, \dots, c_{a-1}, \mathbf{x}_k^a) = T_{\mathbf{x}_k^1 \rightarrow \mathbf{x}_j^1}(\bar{x}_k)$$

for the new vertices to hold. Note that by definition the function  $R$  is then also bijective, i. e.,  $R_k(\bar{x}_k, R_j(\bar{x}_j, \bar{x}_k)) = \bar{x}_k$ .

The PDF of a new path  $\bar{y}_j = R_j(\bar{x}_j, \bar{x}_k)$  for a given sampling strategy of a pixel  $m$  is then uniquely defined and can be computed efficiently. First, recall that the PDF for the new path sample  $\bar{y}_j$  with respect to the pixel  $j$  itself is simply the product of all local sampling PDFs from the pixel's path sampler at each vertex. Further, the first two vertices of  $\bar{y}_j$  are always defined by the pixel  $j$  itself and do not depend on  $m$ . The PDF for sampling the following sub-path of  $\bar{y}_j$  from the pixel  $m$  is computed by shifting  $\bar{y}_j$  to  $m$ , giving a new path  $\bar{y}_m = R_m(\bar{x}_m, \bar{y}_j)$ . The bijectivity of  $R$  implies that the PDFs of the sub-paths  $(y_j^1, y_j^2, \dots)$  and  $(y_m^1, y_m^2, \dots)$  are the same for  $m$  and we can write:

$$p_m(\bar{y}_j) = p_j(\mathbf{x}_j^0 \rightarrow \mathbf{x}_j^1) \prod_{n=1} p_m(\mathbf{y}_m^n \rightarrow \mathbf{y}_m^{n+1})$$

Estimating the PDF of a path for the  $m$ -th pixel sampling strategy hence requires the path to be shifted to the pixel  $m$ , however, this process does not introduce any computational overhead and occurs naturally since each path is reused for each other pixel in the tile anyway. The Jacobian determinant of our proposed function  $R$  solely depends on the shift mapping since the Jacobian for the first path segment is always unity and it follows that  $|R'_{k \rightarrow j}| = |T'_{\mathbf{x}_k^1 \rightarrow \mathbf{x}_j^1}|$ . Note that the original formulation of path reusing from Bekaert et al. does not explicitly include a Jacobian term for converting between path spaces because it assumes that all paths and PDFs are expressed relative to the global path-space measure in which case the Jacobian determinant is simply one.

Several shift functions have been proposed in previous works and designing new shifts is an active area of research. To shift paths with highly-glossy and specular light transport, Lehtinen et al. proposed to employ the manifold exploration technique from Jakob and Marschner [Jakob and Marschner 2012]. Later, a less powerful but simpler shift was proposed for gradient-domain path tracing, whose Jacobian determinant only relies on local information and can be computed more conveniently without the need for costly numerical optimization. We will use the simpler shift from [Kettunen et al. 2015] to construct the vertices  $c_2, c_3, \dots, c_{a-1}$  for a fair comparison to gradient-domain path tracing. Note however that we are free to use any shift as long as it is bijective. The main idea of the shift from Kettunen et al. is to follow the half-vectors of the base path for the shifted path when a direct connection is not immediately possible. Once two consecutive vertices which are classified as connectable (i. e., they are not specular or highly-glossy) are found, the shifted path is reconnected to the base path. In the easiest case (every path vertex is connectable), the shift will closely resemble the standard operation of conventional path reusing. We refer the reader to [Kettunen et al. 2015] for more details on the shift mapping and the computation of the Jacobian determinant.

As a result of the generalized path reusing estimator (Eq. 7) and the newly designed reusing function, we are now able to significantly improve the efficiency of path reusing for scenes including specular and highly-glossy light transport. An example is shown in Fig. 3 where we compare conventional and generalized path reusing (still in the image domain) for a part of the popular Lamp scene from the bi-directional path tracing paper by Veach and Guibas [Veach and Guibas 1995]. The illustrated part features a glass egg made out of a perfectly specular material casting caustics on a diffuse table. It can be seen that our generalized version can handle paths with specular interactions more properly. An advantageous property of our method is that, once a difficult path is found (e.g., a caustic path), the neighboring pixels are able to explore that regions of path space as well. This property leads to a more even convergence in regions with paths that are hard to find with unidirectional techniques. We believe that our formulation of generalized path reusing further provides a useful framework for the future design of more sophisticated reusing functions.

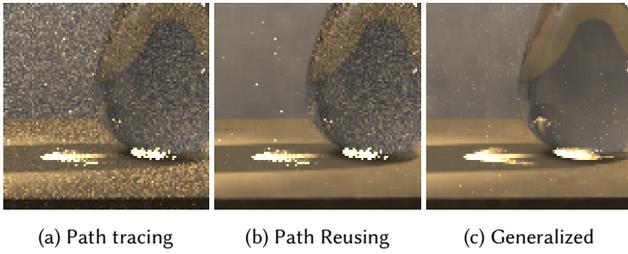


Fig. 3. **Conventional vs. Generalized Path Reusing.** The figure shows part of the popular Lamp scene from [Veach and Guibas 1995] rendered with 1024 samples. Conventional path reusing can reduce noise for diffuse paths, but fails to reuse paths for pixels that directly view the glass egg on the table or caustic paths. Generalized path reusing with an adequate shift handles these paths more properly.

### 3.2 Reusing Paths for Gradient Sampling

Our next goal is to combine path reusing and gradient sampling. In the spirit of conventional path reusing, we unify Eq. 4 and Eq. 6 and define the gradient  $\Delta_{i,j}$  between two pixels  $i$  and  $j$  as:

$$\Delta_{i,j} = \sum_{k \in \mathcal{T}} \int_{\mathcal{P}_k} w_k(\bar{\mathbf{x}}) \left( F_{k \rightarrow i}(\bar{\mathbf{x}}) - F_{k \rightarrow j}(\bar{\mathbf{x}}) \right) d\mu(\bar{\mathbf{x}}) \quad (8)$$

$$\text{with } F_{a \rightarrow b}(\bar{\mathbf{x}}) = f_b(R_{a \rightarrow b}(\bar{\mathbf{x}})) |R'_{a \rightarrow b}|$$

Note that by definition of the balance heuristic the weight  $w_k$  is constant over the tile for a given path  $\bar{\mathbf{x}}_k$  and is identical for  $\bar{\mathbf{x}}_k$  and its shifted paths  $R_{k \rightarrow i}(\bar{\mathbf{x}}_k)$  and  $R_{k \rightarrow j}(\bar{\mathbf{x}}_k)$ .

We make one more assumption about the reusing function  $R$  to define an efficient estimator for Eq. 8. Since the first path segments of the shifted paths  $R_i(\bar{\mathbf{x}}_i, \bar{\mathbf{x}}_k)$  and  $R_j(\bar{\mathbf{x}}_j, \bar{\mathbf{x}}_k)$  depend only on the original samples  $\bar{\mathbf{x}}_i$  and  $\bar{\mathbf{x}}_j$ , the values of the pixel filter  $h_i$  and  $h_j$  might differ and would be required to be evaluated for each term in the difference of Eq. 8. However, we can correlate the first path segment throughout  $\mathcal{T}$  for all sampled paths and guarantee that the relative image positions for each path are at the same distance to their respective pixel center making their pixel filters identical. This condition is easily achieved, e. g., by using the same random numbers for all pixels in  $\mathcal{T}$  when sampling their primary rays. We can then define the gradient-domain path reusing estimator as:

$$\Delta_{i,j} \approx \sum_{k \in \mathcal{T}} w_k(\bar{\mathbf{x}}_k) \frac{f_i(R_i(\bar{\mathbf{x}}_i, \bar{\mathbf{x}}_k)) |R'_{k \rightarrow i}| - f_j(R_j(\bar{\mathbf{x}}_j, \bar{\mathbf{x}}_k)) |R'_{k \rightarrow j}|}{p_k(\bar{\mathbf{x}}_k)}$$

The estimator resembles Eq. 7 with the important distinction that a path  $\bar{\mathbf{x}}_k$  is now reused twice. Since the gradient estimate is computed between two shifted versions of the same path, correlation is implicitly introduced. For the case that  $\mathcal{T}$  only contains the pixels  $i$  and  $k$ , the estimator resembles the symmetric gradient-domain path tracing estimator proposed in [Kettunen et al. 2015] (recall that  $R_j(\bar{\mathbf{x}}_j, \bar{\mathbf{x}}_j) = \bar{\mathbf{x}}_j$  and  $|R'_{j \rightarrow j}| = 1$ ). Note that the estimator automatically performs symmetric gradient sampling because paths are mutually shifted between pixels (given that the pixels  $i$  and  $j$  belongs to the same  $\mathcal{T}$ ).

A problem occurs at tile borders where the correctness of gradient sampling is not guaranteed. Consider two neighboring pixels  $i$  and  $j$  which are not in the same tile. In this case, the path sampler of pixel  $j$  would not be included in the sampling process of  $\Delta_{i,j}$  and it cannot be assured that all pairs involving paths from the path space  $\mathcal{T}_j$  are completely covered. To ensure this property, we have to make sure that both pixels of a gradient are inside the same tile. Therefore, we propose to extend each tile with a border of a single pixel to the right and the bottom (considering that we only need to compute the forward image gradients with one pixel spacing) as illustrated in Fig. 4. The final gradient image is stitched together using only the "inner" parts of each tile in the spirit of seamless image stitching [Levin et al. 2006]. Note that only a single row/column of pixels is added to each tile, which is different from the overlapping tiles approach from Xu and Sbert [Xu and Sbert 2007]. The introduced overhead is typically reasonable and further decreases with an increasing tile size.

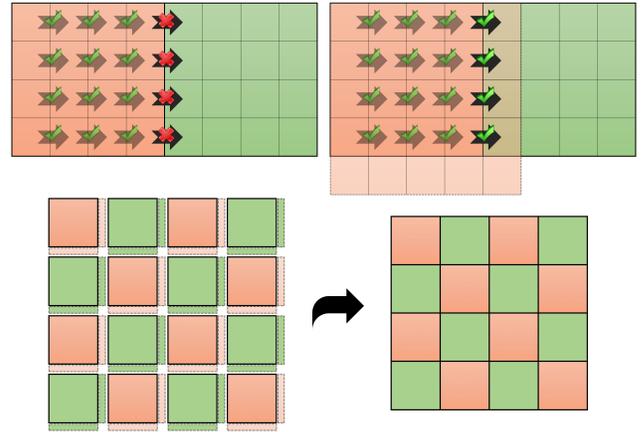


Fig. 4. **Tile Stitching.** The figure illustrates how the final image is stitched from the "inside" regions of the extended tiles.

### 3.3 Implementation

The implementation of gradient-domain path reusing first divides the image into tiles as previously described. For each tile, we initially trace and store a primary ray for each pixel inside it and use the same random numbers to achieve correlation. Next, we construct a complete base path for the first pixel and store it. Having access to the full path at once allows for performance optimizations in form of early outs and Russian roulette. We then shift the base path to all other pixels in the tile one bounce at a time, eventually tracing new rays for following the half-vectors of the base path or connecting to it. During this process, the contribution from all shifted paths and the multiple importance sampling weight (which is constant over the whole tile) are updated sequentially and accumulated to the gradient estimates. Once a base path has been used all over the tile, we continue to the next pixel until all original paths in the tile have been reused. Afterwards, we move on to the next sample iteration. We jitter the tile centers in the same way for the whole image after each iteration to give every pixel roughly the same chance of

being close to the center or nearby tile borders on average which eliminates the need for overlapping tiles when a sufficient number of sample iterations is performed. Finally, after all path samples are exhausted, the final image is reconstructed from the estimated gradients and pixel values using the screened Poisson reconstruction.

Our method potentially performs a larger number of shifts in contrast to gradient-domain path tracing which only creates up to four shifts per base path. Since the shift functions are computationally involved, eliminating fruitless shifts becomes more important in our approach. We decided to store the current base path to exploit full knowledge of the complete path and provide an early out for connection attempts which are guaranteed to add no contribution, i. e., base paths with zero contribution after the connection vertex. To avoid reusing paths with very small contributions, we further perform proportional Russian roulette for the base paths. To this end, we choose a survival probability  $q$  for each base path  $\bar{x}_j$  depending on the luminance of its contribution:

$$q_j = \min \left( 1, \frac{\text{luminance}(\bar{x}_j)}{\delta} \right)$$

We select  $\delta = 0.1$  for all our scenes, which means that only paths with a contribution smaller than 0.1 undergo Russian roulette. If the base path "survives" it will be reused for all other pixels in the tile. A problem with considering only the contribution of the base path for Russian roulette in gradient sampling is that a shifted path might have a non-zero contribution while the base path does not carry any light. Giving a zero survival probability to these path pairs would lead to an incorrect Monte-Carlo estimation. Therefore, we prohibit the survival probability to fall below  $q_{\min} = 0.1$ , which means that every base path with a contribution below 0.1 has at least a 10% chance of being reused. While this is not optimal in terms of estimator variance, it does not introduce bias and performed reasonable well for all our test scenes. Developing more sophisticated Russian roulette strategies (e.g., based on efficiency-optimized Russian roulette [Veach 1998]) for path reusing is a potentially fruitful direction for future work.

## 4 RESULTS

We integrated our method into the Mitsuba renderer [Jakob 2010a]. The rendering integrator was implemented on the CPU with multi-threading support using C++, while the Poisson solver was taken from the implementation provided in [Kettunen et al. 2015] which is implemented on the GPU using NVIDIA CUDA. Reconstructing the final image using the screened Poisson step takes only a few seconds and is negligible compared to the total rendering cost. All results were captured on a Windows 7 PC with an Intel(R) Core i7-4770 CPU with 3.40 GHz, 24GB of system memory, and an NVIDIA Geforce GTX 760.

We evaluated our method in four popular test scenes for light transport; VEACH-DOOR, BATHROOM, BOOKSHELF, and KITCHEN. All scenes feature complex glossy and specular light transport and, with the exception of the BOOKSHELF scene, are dominated by indirect illumination. All scenes were rendered in HD (720p) resolution and the references were generated with bi-directional path tracing and

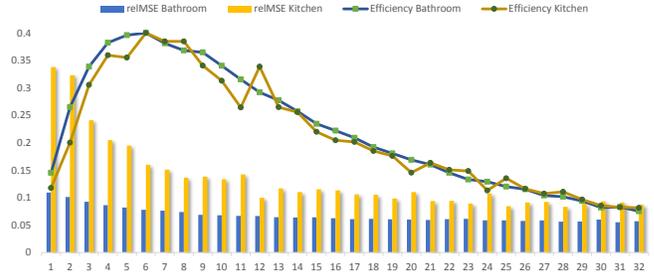


Fig. 5. **Tile Size Evaluation.** The figure shows the influence of the tile size parameter for the BATHROOM and KITCHEN scene with 64 samples per pixel. The bars show the relative MSE for each tile size, while the line plots indicate the efficiency of the estimator which is defined as the inverse of the product of time and error. It can be seen that tile sizes around 6 - 8 give a good trade-off between error and render time.

100000 samples per pixel, taking up to days on a render cluster with 64 CPU cores. We used a maximum path length of 6-8 which is rather low for complex scenes, but were used as default parameters in [Kettunen et al. 2015]. We expect the benefit of our method to be even greater with an increasing number of bounces. We also used the default parameters provided in [Kettunen et al. 2015] for the shift threshold that classifies the connectivity of glossy materials and for the parameter of the Poisson reconstruction. All errors were measured using the relative mean-squared error (relMSE) [Rousselle et al. 2012] which is defined as the mean over all pixel errors given by  $\sum_c (I_k^c - ref_k^c)^2 / (\overline{ref_k^c}^2 + \epsilon)$  where  $c$  sums over the color channels,  $\overline{ref_k^c}$  is the mean of all color channels of the reference pixel, and  $\epsilon = 0.001$ .

The main parameter of our method is the tile size which has a significant influence on the performance. Choosing the tile size too small leads to only a minor noise reduction, while a too large tile size will perform many connection attempts that are likely to fail and wastes computation time. Bekaert et al. estimated the gain of path reusing to be dependent on the average path length, the cost along a path (e.g., from next even estimation), and the tile size. Since generalized path tracing also supports highly-glossy and specular light transport, the material complexity of a scene further effects the efficiency of the estimator. We evaluate the quality for varying tile sizes in the BATHROOM and KITCHEN scene with 64 samples in Fig. 5. We report errors as well as the efficiency of the estimator which is defined as  $\frac{1}{\text{error} \cdot \text{time}}$  (c.f. [Veach 1998]). It can be seen that the highest efficiencies are typically found around a size of 6 - 8 which corresponds to 36 - 64 paths per tile. Our findings roughly coincide with the ones from Bekaert et al. and Xu et al. which typically used a slightly lower number of 16 - 36 paths.

We further report statistics of our estimator in our four test scenes using 16 samples per pixel in Table 1. On average, around 83.7% of base paths are skipped by Russian roulette and we found that the number remains stable over varying tile sizes and sampling rates. While the value seems high at first, it is mainly due to the fact that

	RR %	$\mathcal{T} = 4$	$\mathcal{T} = 6$	$\mathcal{T} = 8$
VEACH-DOOR	85.3%	3.66 (15.2%)	7.15 (14.8%)	11.85 (14.8%)
BATHROOM	81.6%	4.59 (19.1%)	8.98 (18.7%)	14.86 (18.5%)
BOOKSHELF	84.8%	3.79 (15.7%)	7.40 (15.4%)	12.24 (15.3%)
KITCHEN	83.3%	4.16 (17.3%)	8.12 (16.9%)	13.45 (16.8%)

Table 1. **Reusing Statistics.** The table shows the average percentage of base paths skipped per tile by Russian roulette (RR %), and the average number of base paths used for each gradient and pixel estimate. The number in brackets gives the percentage of reused base paths relative to the tile size.

a unidirectional path sampler has difficulties finding high contributions paths in our test scenes and only a small fraction of generated path samples actually carry light. Nonetheless, around 16.5% of the path samples in a tile are reused on average for the estimation of the gradients and pixel values compared to gradient-domain path tracing which always uses two paths per gradient estimate. For the following convergence analysis, we chose a fixed tile size of 6.

We compare our method to traditional path tracing (PT), conventional path reusing (PR), and gradient-domain path tracing (G-PT) in terms of convergence and visual quality in Fig. 6. Gradient-domain path reusing gives much smoother results than traditional path tracing and reusing, and drastically reduces the artifacts from gradient-domain path tracing. This is especially visible at lower sampling rates and around discontinuities where our method produces sharp and clean edges which reflects the virtually higher sampling rate and robustness of our estimator. In the VEACH-DOOR and the BOOKSHELF scenes, the convergence is improved by up to almost an order of magnitude. The improvement is not as drastic in the KITCHEN scene which is mainly due to the fact that the scene contains many highly-glossy surfaces for which the shift function cannot efficiently perform an early-out as explained in Sec. 3. However, our method still performs better by comparison and gives visually more pleasant results. Overall, gradient-domain path reusing performs best for all test scenes in terms of convergence and visual quality.

One limitation that our method shares with conventional path reusing is its reduced efficiency when surfaces are viewed at grazing angles. In this case, fewer neighboring samples can be reused, however, even in the worst case our estimator recedes to classic gradient sampling. Further, outliers in the gradient estimates have a similar impact as for gradient-domain path tracing. To address these issues, the L1 Poisson reconstruction as proposed in [Lehtinen et al. 2013] could be applied if image smoothness is preferred over unbiasedness. Currently, our method does not incorporate a temporal component and we expect it to perform similarly as gradient-domain path tracing for scenes with motion blur and animations. However, we believe that extending our method to the temporal domain in the spirit of [Manzi et al. 2016] is possible. Finally, adaptive sampling of the image is feasible but must be properly integrated in the tiled sampling process and requires a more involved computation of the multiple importance sampling weights.

## 5 CONCLUSION AND FUTURE WORK

We proposed gradient-domain path reusing, a novel rendering technique which unifies the strength of path reusing and gradient-domain rendering. Further, we generalized path reusing and improved its efficiency for scenes with highly-glossy and specular light transport. We believe that the proposed mathematical framework of *generalized path reusing* facilitates future designs of reusing functions and extends beyond the scope of this paper. Our proposed method outperforms conventional and gradient-domain path tracing and significantly reduces the perceptually disturbing correlation patterns of traditional path reusing.

We hope that our method inspires future work and believe that fruitful directions are importance sampling strategies for path connections and the design of novel reusing functions. Further, extending our concept to bidirectional approaches has the potential to lead to very powerful unbiased rendering algorithms.

## ACKNOWLEDGMENTS

We thank Wenzel Jakob for the making the Mitsuba renderer publicly available [Jakob 2010b]. Further, we thank Kettunen et al. [Kettunen et al. 2015] for providing the test scenes used in our evaluation and Eric Veach, Miika Aittala, Samuli Laine, Jaakko Lehtinen, Tiziano Portenier, and Anton Kaplanyan for modeling or porting them. This work is part of the research program "LED it be 50%" (project number P13-20), which is funded by the Netherlands Organisation for Scientific Research (NWO) and supported by LTO Glaskracht, Philips, Nunhems, WUR Greenhouse Horticulture.

## REFERENCES

- James Arvo and David Kirk. 1990. Particle Transport and Image Synthesis. In *Proceedings of the 17th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '90)*. ACM, 63–66.
- Philippe Bekaert, Mateu Sbert, and John Halton. 2002. Accelerating Path Tracing by Re-using Paths. In *Proceedings of the 13th Eurographics Workshop on Rendering (EGRW '02)*. Eurographics Association, 125–134.
- Pravin Bhat, C. Lawrence Zitnick, Michael Cohen, and Brian Curless. 2010. GradientShop: A Gradient-domain Optimization Framework for Image and Video Filtering. *ACM Trans. Graph.* 29, 2, Article 10 (April 2010), 14 pages.
- Francisco Castro, Mateu Sbert, and John H. Halton. 2008. Efficient Reuse of Paths for Random Walk Radiosity. *Comput. Graph.* 32, 1 (Feb. 2008), 65–81.
- Robert L. Cook, Thomas Porter, and Loren Carpenter. 1984. Distributed Ray Tracing. In *Proceedings of the 11th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '84)*. ACM, 137–145.
- Tomáš Davidovič, Jaroslav Krivánek, Miloš Hašan, Philipp Slusallek, and Kavita Bala. 2010. Combining Global and Local Virtual Lights for Detailed Glossy Illumination. In *ACM SIGGRAPH Asia 2010 Papers*. ACM, Article 143, 8 pages.
- Binh-Son Hua, Adrien Gruson, Derek Nowrouzezahrai, and Toshiya Hachisuka. 2017. Gradient-Domain Photon Density Estimation. *Computer Graphics Forum* (2017).
- Wenzel Jakob. 2010a. Mitsuba renderer. (2010). <http://www.mitsuba-renderer.org>.
- Wenzel Jakob. 2010b. Mitsuba renderer. (2010). <http://www.mitsuba-renderer.org>.
- Wenzel Jakob and Steve Marschner. 2012. Manifold Exploration: A Markov Chain Monte Carlo Technique for Rendering Scenes with Difficult Specular Transport. *ACM Trans. Graph.* 31, 4, Article 58 (July 2012), 13 pages.
- Henrik Wann Jensen. 1996. Global Illumination Using Photon Maps. In *Proceedings of the Eurographics Workshop on Rendering Techniques '96*. 21–30.
- James T. Kajiya. 1986. The Rendering Equation. *SIGGRAPH Comput. Graph.* 20, 4 (Aug. 1986), 143–150.
- Alexander Keller. 1997. Instant Radiosity. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '97)*. 49–56.
- Markus Kettunen, Marco Manzi, Miika Aittala, Jaakko Lehtinen, Frédo Durand, and Matthias Zwicker. 2015. Gradient-domain Path Tracing. *ACM Trans. Graph.* 34, 4, Article 123 (July 2015), 13 pages.
- Jaakko Lehtinen, Tero Karras, Samuli Laine, Miika Aittala, Frédo Durand, and Timo Aila. 2013. Gradient-domain Metropolis Light Transport. *ACM Trans. Graph.* 32, 4,

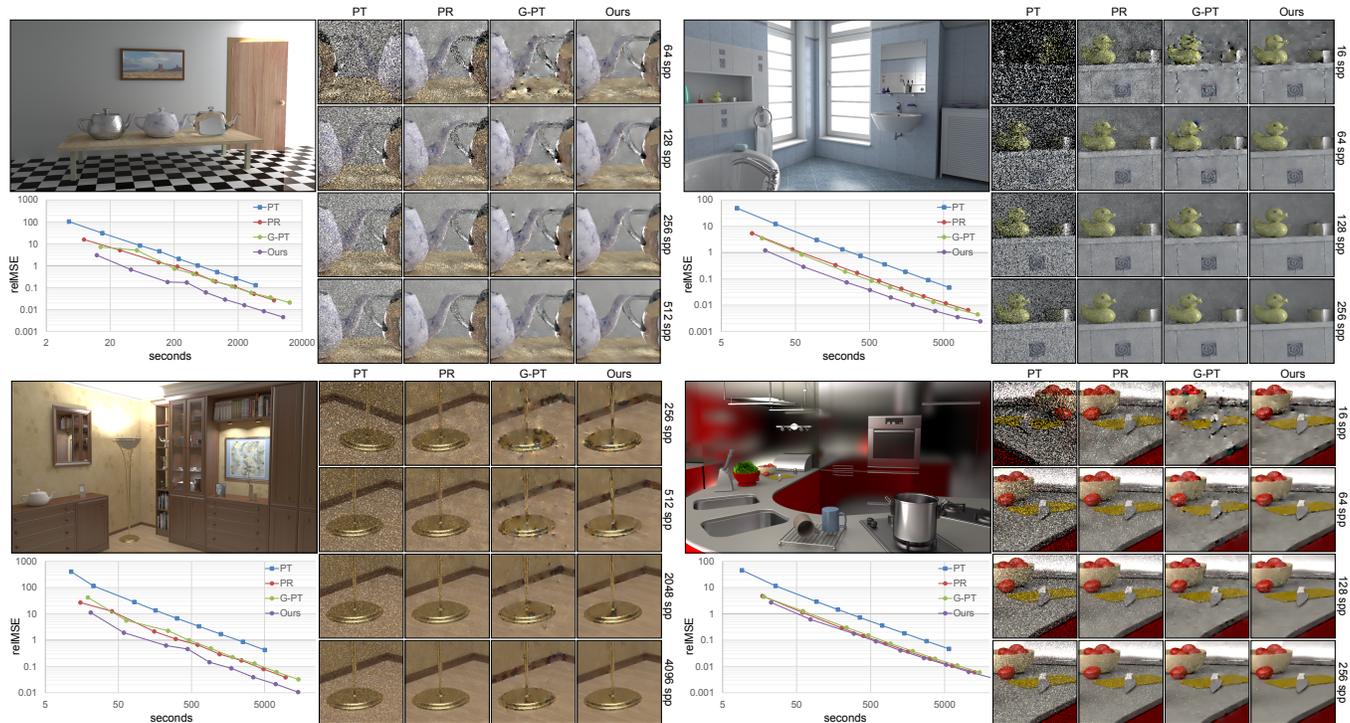


Fig. 6. **Convergence Evaluation.** A comparison of our method with conventional and gradient-domain path tracing, and traditional path reusing (using 4 patterns) for a tile size of 6 and varying sampling rates. The convergence behavior is illustrated in the graph plotting the time in seconds vs. the error in relMSE. Note that both axis are given in the log<sub>10</sub>-scale. Our method performs best in all cases, with significant improvements especially for the VEACH-DOOR and BOOKSHELF scenes. Visually, our approach produces much smoother results than conventional path tracing and greatly reduces the typical artifacts of gradient-domain path tracing especially around image edges. The images are best viewed in the digital version using the zoom function.

- Article 95 (July 2013), 12 pages.
- Anat Levin, Assaf Zomet, Shmuel Peleg, and Yair Weiss. 2006. Seamless image stitching in the gradient domain. In *Proceedings of the European Conference on Computer Vision*.
- Marco Manzi, Markus Kettunen, Miika Aittala, Jaakko Lehtinen, Frédo Durand, and Matthias Zwicker. 2015. Gradient-Domain Bidirectional Path Tracing. In *Eurographics Symposium on Rendering - Experimental Ideas & Implementations*.
- Marco Manzi, Markus Kettunen, Frédo Durand, Matthias Zwicker, and Jaakko Lehtinen. 2016. Temporal Gradient-domain Path Tracing. *ACM Trans. Graph.* 35, 6, Article 246 (2016), 9 pages.
- Marco Manzi, Fabrice Rousselle, Markus Kettunen, Jaakko Lehtinen, and Matthias Zwicker. 2014. Improved Sampling for Gradient-domain Metropolis Light Transport. *ACM Trans. Graph.* 33, 6, Article 178 (Nov. 2014), 12 pages.
- M. Manzi, D. Vicini, and M. Zwicker. 2016. Regularizing Image Reconstruction for Gradient-Domain Rendering with Feature Patches. *Computer Graphics Forum* 35, 2 (2016), 263–273.
- Patrick Pérez, Michel Gangnet, and Andrew Blake. 2003. Poisson Image Editing. *ACM Trans. Graph.* 22, 3 (July 2003), 313–318.
- Matt Pharr, Wenzel Jakob, and Greg Humphreys. 2016. *Physically Based Rendering, Third Edition: From Theory To Implementation* (3rd ed.). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- Stefan Popov, Ravi Ramamoorthi, Frédo Durand, and George Drettakis. 2015. Probabilistic Connections for Bidirectional Path Tracing. In *Proceedings of the 26th Eurographics Symposium on Rendering (EGSR '15)*. Eurographics Association, 75–86.
- Fabrice Rousselle, Wojciech Jarosz, and Jan Novák. 2016. Image-space Control Variates for Rendering. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)* 35, 6 (December 2016), 169:1–169:12.
- Fabrice Rousselle, Claude Knaus, and Matthias Zwicker. 2012. Adaptive rendering with non-local means filtering. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH Asia 2012)* 31, 6 (Nov. 2012).
- Daniel L Ruderman. 1994. The statistics of natural images. *Network: Computation in Neural Systems* 5, 4 (1994), 517–548.
- Mateu Sbert and J Halton. 2004. Reuse of paths in light source animation. In *Proceedings of Computer Graphics International*. IEEE, 532–535.
- B. Segovia, J. C. Iehl, R. Mitanchey, and B. Péroche. 2006. Bidirectional Instant Radiosity. In *Proceedings of the 17th Eurographics Conference on Rendering Techniques (EGSR '06)*. Eurographics Association, 389–397.
- Peter Shirley, Bretton Wade, Philip M. Hubbard, David Zareski, Bruce Walter, and Donald P. Greenberg. 1995. *Global Illumination via Density-Estimation*. 219–230.
- Weilun Sun, Xin Sun, Nathan A. Carr, Derek Nowrouzezahrai, and Ravi Ramamoorthi. 2017. Gradient-Domain Vertex Connection and Merging. In *Eurographics Symposium on Rendering - Experimental Ideas & Implementations*.
- Eric Veach. 1998. *Robust Monte Carlo Methods for Light Transport Simulation*. Ph.D. Dissertation. Stanford, CA, USA. Advisor(s) Guibas, Leonidas J. AAI9837162.
- Eric Veach and Leonidas J. Guibas. 1995. Optimally Combining Sampling Techniques for Monte Carlo Rendering. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '95)*. ACM, 419–428.
- Eric Veach and Leonidas J. Guibas. 1997. Metropolis Light Transport. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '97)*. 65–76.
- Jiří Vorba and Jaroslav Křivánek. 2016. Adjoint-Driven Russian Roulette and Splitting in Light Transport Simulation. *ACM Trans. Graph.* 35, 4 (July 2016), 1–11. (accepted for publication).
- Gregory J. Ward, Francis M. Rubinstein, and Robert D. Clear. 1988. A Ray Tracing Solution for Diffuse Interreflection. *SIGGRAPH Comput. Graph.* 22, 4 (June 1988), 85–92.
- Qing Xu and Mateu Sbert. 2007. A New Way to Re-using Paths. In *Proceedings of the 2007 International Conference on Computational Science and Its Applications - Volume Part II (ICCSA'07)*. 741–750.