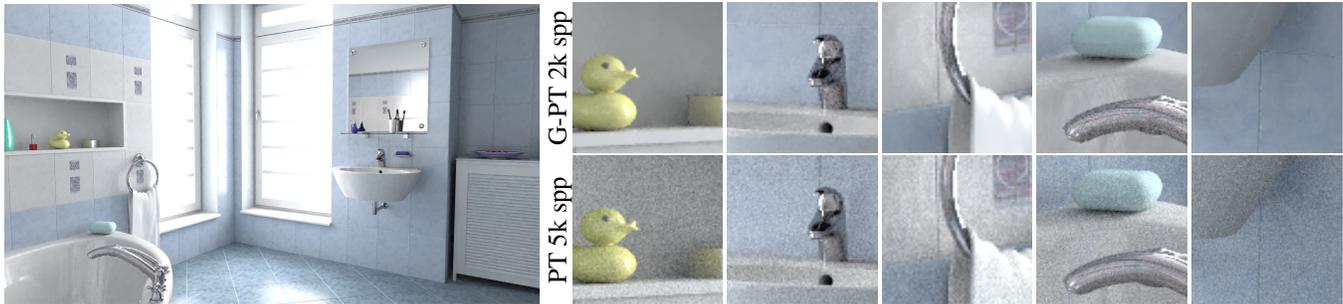


# Gradient-Domain Path Tracing

Markus Kettunen<sup>1</sup> Marco Manzi<sup>2</sup> Miika Aittala<sup>1</sup> Jaakko Lehtinen<sup>1,3</sup> Frédo Durand<sup>4</sup> Matthias Zwicker<sup>2</sup>  
<sup>1</sup>Aalto University <sup>2</sup>University of Bern <sup>3</sup>NVIDIA <sup>4</sup>MIT CSAIL



**Figure 1:** Comparing gradient-domain path tracing (G-PT,  $L_1$  reconstruction) to path tracing at equal rendering time (2 hours). In this time, G-PT draws about 2,000 samples per pixel and the path tracer about 5,000. G-PT consistently outperforms path tracing, with the rare exception of some highly specular objects. Our frequency analysis explains why G-PT outperforms conventional path tracing.

## Abstract

We introduce gradient-domain rendering for Monte Carlo image synthesis. While previous gradient-domain Metropolis Light Transport sought to distribute more samples in areas of high gradients, we show, in contrast, that estimating image gradients is also possible using standard (non-Metropolis) Monte Carlo algorithms, and furthermore, that even without changing the sample distribution, this often leads to significant error reduction. This broadens the applicability of gradient rendering considerably. To gain insight into the conditions under which gradient-domain sampling is beneficial, we present a frequency analysis that compares Monte Carlo sampling of gradients followed by Poisson reconstruction to traditional Monte Carlo sampling. Finally, we describe Gradient-Domain Path Tracing (G-PT), a relatively simple modification of the standard path tracing algorithm that can yield far superior results.

**CR Categories:** [Computer Graphics]: Rendering

**Keywords:** path tracing, gradient-domain, global illumination, light transport

## 1 Introduction

Global illumination algorithms seek to estimate the value of each image pixel defined as a complex integral over the space of light paths, relying on large numbers of Monte-Carlo samples to avoid noise. Recent advances [Lehtinen et al. 2013; Manzi et al. 2014]

take advantage of Metropolis sampling to distribute samples according not only to pixel values (or path throughput), but also finite-difference image gradients. This reduces costs by focusing computing resources onto areas of high variations. Metropolis light transport is notoriously hard to implement, however, and to the best of our knowledge, Mitsuba [Jakob 2012] is the only publicly-available implementation of Veach’s original algorithm [Veach and Guibas 1997]. Furthermore, the convergence behavior of Metropolis algorithms is often considered undesirable because they explore path space in a highly non-uniform fashion. As a consequence, error appears as entire regions being too dark or too bright, instead of as noise. This is even more challenging for gradient sampling, because the areas of high contribution are more sparse. In short, while Metropolis light transport remains unequalled for challenging light transport configurations, many practical scenarios can be more easily handled by standard approaches such as path tracing.

We show, that, maybe surprisingly, we can benefit from rendering image gradients rather than only pixel values also in standard path tracing, even without changing sample distributions (Figure 1). We extend path tracing and shoot, for each base path, additional finite-difference offset paths shifted by one pixel. This provides us not only with the image contribution for each path, but also an estimate of the finite-difference between the two pixels. In the end, we perform a screened Poisson reconstruction that combines information from both sampled pixel values (especially useful for low frequencies) and gradients (much more accurate for high frequencies).

Crucially, our gradient estimates provide much more information than finite differences of a conventionally sampled image. In both cases, a gradient is the difference of two pixels. But rather than using the difference between two sums of uncorrelated random paths, we estimate gradients by integrating the difference between pairs of paths that are carefully sampled in a highly correlated fashion. For this, we design shift mappings that generate path pairs that are “similar” so that the differences are small and result in lower variance, in a manner similar to variance reduction using correlated sampling or common random numbers in statistics.

We analytically study why gradient rendering can yield better quality than direct value rendering. A key insight is that the symmetric operations of finite differencing and Poisson image reconstruction do not cancel each other in the error of gradient rendering. This

This is an author-prepared preprint.  
 The definitive version appears in the ACM  
 Digital Library (dl.acm.org).

is because stochastic sampling, which occurs between finite differencing in path space and image reconstruction, scrambles frequency content through aliasing. We derive formulas that explain how the efficiency of gradient rendering depends on the power spectrum of the integrand, that is, the image contribution function. Gradient rendering works best for functions with a frequency falloff (less energy in high compared to low frequencies), which are typical in natural images and integrands in light transport. We show precisely how Poisson reconstruction combines gradient and pixel information to get the best of both worlds, by relying on regular rendering for low frequencies and gradient rendering for high frequencies.

In order to simplify implementation, we propose a new shift mapping that does not rely on manifold perturbations like the original shift by Lehtinen et al. [2013], and whose Jacobian, which is necessary to account for the change of path densities introduced by the shift [Lehtinen et al. 2013; Manzi et al. 2014], is also simple to evaluate. Finally, we observe that the difference between two pixels can be sampled according to either a forward or an inverse shift, and we show that applying multiple importance sampling between these two strategies significantly improves quality.

In summary, we make the following contributions:

- A mathematical formulation of the rendering equation suitable for estimating image gradients using standard Monte Carlo sampling.
- A gradient-domain path tracing algorithm that outperforms its conventional counterpart by up to an order of magnitude in relative mean squared error at equal render time.
- A frequency analysis of gradient sampling and reconstruction for Monte Carlo rendering that, under simplifying assumptions, provides a theoretical understanding of the practical benefits of gradient domain rendering.
- Sampling strategies for gradients in path tracing, including a multiple importance sampling technique for gradients, and suitable shift mappings to obtain correlated path pairs.

## 2 Related Work

**Derivative Analysis in Rendering** Gradients have found many uses in rendering, typically in criteria for adaptive sampling, reconstruction, or prefiltering mipmap levels. In contrast, we consider finite differences and seek to directly estimate the final image through gradients. Irradiance caching [Ward et al. 1988] is an early example that computes high quality irradiance samples at sparse locations, and estimates an upper bound on the irradiance gradients to control their density. Many improvements of this basic scheme have been proposed, including more accurate gradient estimates [Ward and Heckbert 1992], extensions to low-frequency glossy materials [Křivánek et al. 2005], participating media [Jarosz et al. 2008a; Jarosz et al. 2008b], or higher order derivatives [Jarosz et al. 2012; Schwarzhaupt et al. 2012]. While most variants were developed with an eye towards practical applicability, Ramamoorthi et al. [2007] conducted a more comprehensive first-order analysis of lighting, shading, and shadows in direct illumination. As a proof-of-concept, they also describe an adaptive sampling algorithm. Ray differentials [Igehy 1999] were developed to approximate partial derivatives of the ray geometry, in particular for texture filtering.

The (local) frequency analysis of light transport [Durand et al. 2005] is related to derivative analysis since it also attempts to obtain additional information in a neighborhood of light paths. Interestingly, there is a close connection between second order derivatives and Gaussian approximations of local frequency spectra [Belcour et al. 2014].

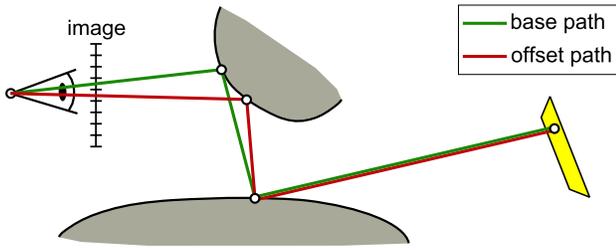
A key difference between the work discussed above and our approach is that we directly sample finite pixel differences, instead of using closed form approximations of the derivative. A crucial advantage of our approach is that our sampled image gradients are unbiased estimates, and we can exploit these samples to obtain unbiased final images. If closed form gradient approximations are used for image reconstruction, bias cannot be avoided.

**Gradient-domain Metropolis Light Transport** Metropolis Light Transport (MLT) [Veach and Guibas 1997] constructs a Markov chain of light paths that is distributed proportionally to image contributions. In gradient-domain MLT (GMLT) [Lehtinen et al. 2013], the random walk is performed on an extended space of *pairs of paths* in adjacent pixels, such that the sampler is driven towards pairs where the light throughput between the two paths differs significantly. The process directly estimates the horizontal and vertical finite differences (gradients) across the image, which are then integrated by solving a Poisson equation. It is easy to derive that the final reconstruction is an unbiased estimate, as is standard MLT, if the Poisson problem is solved under an  $L_2$  norm.

Lehtinen et al. [2013] showed that GMLT can reduce variance significantly at equal computation time compared to standard MLT. A key intuition for the effectiveness of GMLT is that gradients are sparser than the image itself, and can be sampled with less variance. In practice, however, robust sampling of gradients is challenging because of this very sparsity. Recently, Manzi et al. [2014] introduced improved gradient sampling techniques for GMLT that make unbiased reconstruction practical. An important insight of this work is that there is considerable freedom in how to sample gradients, which indicates that there is more room for future improvements. Drawing from GMLT, we develop a similar gradient sampling and reconstruction framework for Monte Carlo rendering, building on standard path tracing. Unlike GMLT, we do not change the sampling density of the underlying renderer.

**Gradients in Image Processing and Natural Images** Gradient-domain rendering is inspired by the role of gradients in image processing. Gradients of natural images tend to be sparse [Ruderman 1994; Simoncelli and Olshausen 2001], hence providing a succinct image representation. As the power spectrum of natural images is approximately inversely proportional to the square of frequency, gradients generally contain much less energy than the original image. This motivates us to sample gradients directly, expecting less variance because the signal has less energy. Interestingly, Tumblin et al. [2005] envision a camera that directly captures gradients. Manipulating image gradients instead of pixel values has proven to be an elegant strategy to develop numerous intuitive and effective image editing algorithms [Pérez et al. 2003; Georgiev 2005]. We similarly solve a Poisson equation to determine the final image from sampled gradients, also including a coarsely sampled primal image to aid the reconstruction of low frequencies [Bhat et al. 2010].

**Image Reconstruction using Adaptive Filtering** Adaptive filtering techniques reconstruct the final image by computing weighted averages over image space, typically using spatially adaptive weights. The weights can be determined, for example, using spatially varying (anisotropic) Gaussians [Rousselle et al. 2011; Belcour et al. 2013], DCT [Bolin and Meyer 1995] or wavelet transforms [Overbeck et al. 2009], or (cross-)bilateral filtering [Sen and Darabi 2012; Li et al. 2012; Rousselle et al. 2013]. All these techniques introduce bias. Screened Poisson reconstruction also computes output pixels as weighted averages, however, including both unbiased pixel and gradient samples. In contrast to other techniques, this does not introduce bias.



**Figure 2:** Key principle of gradient-domain rendering: We sample base paths using a Monte Carlo estimator, such as path tracing. We shift each base path to an offset path using a deterministic shift mapping, and the difference between the base and offset path serves as a gradient sample (Figure from Lehtinen et al. [2013]).

### 3 Overview

The defining idea of gradient-domain rendering is to sample image gradients, that is differences between pairs of neighbor pixels, in addition to pixel values. In contrast to gradient-domain Metropolis light transport [Lehtinen et al. 2013; Manzi et al. 2014], we use a regular Monte Carlo sampling strategy to evaluate gradients. While a standard path tracer evaluates only the image contribution function for each sampled path, we evaluate, in addition, a path difference function defined by a *shift mapping* that maps each *base path* to a “similar” *offset path* through the neighbor pixel, as shown in Figure 2, and returns the difference between the contribution of the base and offset path. We will show that good shift mapping reduces variance by constructing offset paths such that the contributions of base and offset paths are as similar as possible. After sampling is complete, we reconstruct the final image by solving a screened Poisson problem, where the regular image provides lower frequencies, and the gradient image contributes most high frequencies, effectively suppressing high frequency noise in the result.

**Mathematical Formulation** We first need to precisely define the gradients (pixel differences) that we sample. While the basic mathematical formulation follows previous work [Lehtinen et al. 2013; Manzi et al. 2014], we will use fundamentally different sampling strategies. We denote the difference between two pixels  $i$  and  $j$  by  $\Delta_{i,j}$ . Using the path space formulation of light transport [Veach and Guibas 1997] this is

$$\Delta_{i,j} = \left( h(x) * \int_{\Omega} f(x, \bar{p}) d\mu(\bar{p}) \right) (x_i) - \left( h(x) * \int_{\Omega} f(x, \bar{p}) d\mu(\bar{p}) \right) (x_j), \quad (1)$$

where  $x$  is the image coordinate,  $h(x)$  is a pixel filter,  $\Omega$  is the space of all light paths of all lengths that can be sampled with a given standard Monte Carlo path tracer,  $(x, \bar{p})$  is a light path with additional parameters  $\bar{p}$  connecting a point on a light and a point on the sensor, and  $f$  is the image contribution function. We evaluate the convolution with the pixel filter at pixel centers  $x_i$  and  $x_j$ .

In contrast to Equation 1, in gradient-domain rendering we formulate gradients as a single integral and estimate the differences using correlated sample pairs. For each path sample that we draw for pixel  $i$ , we deterministically map it to a “similar” path through the nearby pixel  $j$ , and take the difference between them. Specifically, we rewrite Equation 1 as the integral of a path difference function  $g_{ij}(x, \bar{p})$  instead of the usual image contribution function  $f(x, \bar{p})$ ,

$$\begin{aligned} \Delta_{i,j} &= \left( h(x) * \int_{\Omega} f(x, \bar{p}) - f(T_{ij}(x, \bar{p})) |T'_{ij}| d\mu(\bar{p}) \right) (x_i) \\ &= \left( h(x) * \int_{\Omega} g_{ij}(x, \bar{p}) d\mu(\bar{p}) \right) (x_i), \end{aligned} \quad (2)$$

where  $T_{ij}$  is the shift mapping that deterministically maps a base path  $(x, \bar{p})$  to an offset path  $T_{ij}(x, \bar{p})$ , such that they are “close” to each other in path space (Figure 2). We only allow shifts by a unit pixel distance, which means that the offset path  $T_{ij}(x, \bar{p})$  has the same pixel filter value as the base path. Hence we can express pixel filtering of gradients as a single convolution. The factor  $|T'| = |\partial T / \partial \bar{x}|$  is the determinant of the Jacobian of  $T(\bar{x})$ , accounting for the change of integration variables [Lehtinen et al. 2013].

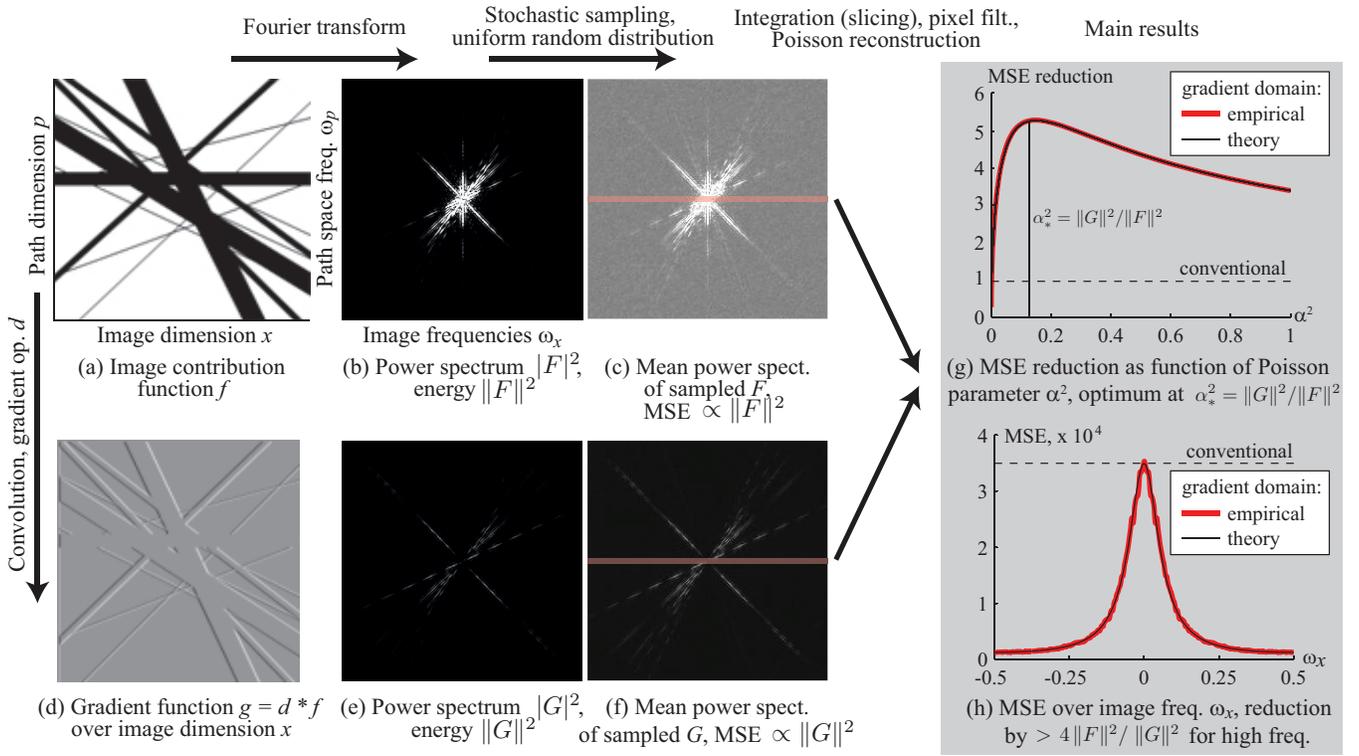
Next, we will next derive a frequency analysis to explain why sampling gradients is beneficial, and how to best combine regular and gradient information. We then describe a practical algorithm that makes it possible to extend standard Monte-Carlo path tracing to leverage gradient-domain rendering. In particular, we present a multiple importance sampling strategy that dramatically improves performance in cases where the shift map has singularities, and a new shift mapping that is much simpler to implement than previous work. This makes the modifications required to convert a standard path tracer to a gradient path tracer quite easily manageable.

### 4 Theoretical Analysis

This section describes the first end-to-end analysis of the sampling and reconstruction process involved in gradient-domain rendering. While previous work has investigated the frequency analysis of the screened Poisson equation [Bhat et al. 2008; Lehtinen et al. 2013], Lehtinen et al. [2013] only observed that Poisson reconstruction weights gradient information in high frequencies more, but they did not provide a full analysis that includes sampling and reconstruction. In contrast, we present a complete analysis that pinpoints why gradient sampling followed by Poisson reconstruction is beneficial. Under certain simplifications, the analysis predicts precisely by how much gradient-domain rendering reduces variance in each frequency compared to conventional sampling. We present an overview and the main results of our analysis here, and refer to the supplemental material for the detailed derivations. We first study the error in gradient estimation compared to usual pixel estimation that is caused by Monte Carlo integration (Section 4.1). Then we analyze screened Poisson reconstruction (Section 4.2) to understand the error distribution over frequencies of the final image.

#### 4.1 Error Analysis of Gradient Estimation

Gradient estimation involves computing and sampling path differences, Monte Carlo integration over path space, and pixel filtering. To make this problem amenable to Fourier analysis, we make the following simplifications: First, we work with 1D images to reduce clutter in the notation. Next, we assume paths are parameterized over a Cartesian hypercube, akin to the primary sample space by Kelemen et al. [2002], and the first path dimension is the image axis. We hide all terms related to the measure in path space in the image contribution function. We restrict the analysis to uniform random sampling in this parameterization. We also assume that sampling is a wide-sense stationary stochastic processes. This implies that it extends over an infinite domain, which simplifies its frequency analysis because there are no boundary effects, but also means that we cannot model the restriction of the sampling grid to the unit hypercube. Assuming sampling over an infinite domain instead of restricting the samples around the non-zero support region of the integrand overestimates variance compared to practical



**Figure 3:** Overview of our frequency analysis (a-f) and its main results (g,h). Given an image contribution function  $f$  that we need to integrate over path dimension  $p$  to get an image (a), we express gradients  $g$  as a convolution of  $f$  with a gradient operator  $d$  in path space (d). We analyze stochastic sampling of  $f$  (b,c) and  $g = d * f$  (e,f) in the frequency domain. The error due to sampling turns out to be constant over all frequencies. It appears as a flat, gray background in (c,f), and it is typically smaller for gradients (f). Next, integration over path dimension  $p$  corresponds to slicing along the red lines in (c) and (f). Poisson reconstruction combines the two slices to form the final image using a parameter  $\alpha$ . Our main results include the derivation of the optimal parameter  $\alpha_*$  that leads to the highest error reduction over conventional rendering (g), and an error analysis of the final output that shows how high frequency error (close to the Nyquist frequency  $1/2$  of the image sampling grid) is strongly reduced (h). Empirical results of the 2D example shown here (red lines in g,h) closely match our theory.

algorithms. Finally, we use a simple shift mapping that only shifts the image coordinate by one pixel and leaves the other parameters untouched. This mapping has a unit Jacobian and determinant.

Given an image contribution function  $f$  defined over an image axis  $x$  and an arbitrarily long vector of path parameters  $\bar{p}$  (Figure 3a), the problem is to integrate over  $\bar{p}$  to obtain a sampled image and its gradients. We model gradient-domain rendering by first defining the shift mapping  $T$  and the corresponding path difference function  $g$  in path space. For this analysis we use a simple shift mapping  $T(x, \bar{p}) = (x - 1, \bar{p})$ . We omit the indices  $ij$  because we employ this same shift everywhere over the image. Therefore, the path difference function is simply  $g(x, \bar{p}) = f(x, \bar{p}) - f(x - 1, \bar{p})$ , which we may write as a convolution  $g(x, \bar{p}) = (d * f)(x, \bar{p})$  with a difference operator  $d(x, \bar{p}) = \delta(x) - \delta(x - 1)$ . This is shown in Figure 3d, where we apply finite differencing horizontally along the image dimension  $x$ , but not vertically over path parameters  $\bar{p}$ .

In the Fourier domain (Figure 3b and 3e)  $g = d * f$  is a multiplication  $G = DF$ . The power spectrum of the path difference function  $|G|^2$  is related to the power spectrum of the image contribution function  $|F|^2$  by

$$\begin{aligned} |G(\omega_x, \omega_{\bar{p}})|^2 &= (2 - 2 \cos(2\pi\omega_x)) |F(\omega_x, \omega_{\bar{p}})|^2 \\ &= |D(\omega_x, \omega_{\bar{p}})|^2 |F(\omega_x, \omega_{\bar{p}})|^2, \end{aligned} \quad (3)$$

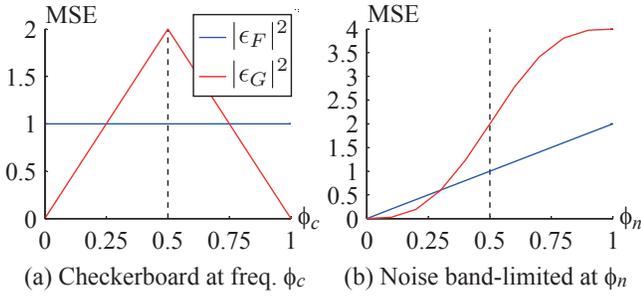
where  $\omega_x$  and  $\omega_{\bar{p}}$  are frequencies over the image and path space, respectively, and  $|D|^2 = (2 - 2 \cos(2\pi\omega_x))$  is the power spectrum of

the finite-difference operator (as opposed to the continuous derivative, which would attenuate frequencies by  $1/\omega_x^2$ ). We get the well known result that finite-differencing cancels out the DC, attenuates low frequencies, and boosts square magnitudes of high frequencies by a factor up to four for the Nyquist limit  $\omega_x = 1/2$  of the image (unit pixel spacing).

We then analyze stochastic sampling of both the image contribution  $F$  and the path difference function  $G$  in the frequency domain (Figure 3c and 3f), and we derive the mean square error (MSE) introduced by sampling, which is equivalent to the variance. Sampling is a multiplication by a union of Diracs in the primal and a convolution in the Fourier domain, and we can interpret the error due to sampling as aliasing [Dippé and Wold 1985; Durand 2011]. We model uniform random sampling as a Poisson process, whose power spectrum is flat except for a Dirac at the DC [Leneman 1966]. As a key consequence, the stochastic convolution results in constant expected errors  $|\epsilon_F(\omega_x, \omega_{\bar{p}})|^2$  and  $|\epsilon_G(\omega_x, \omega_{\bar{p}})|^2$  for all frequencies for both pixels and gradients,

$$\begin{aligned} |\epsilon_F(\omega_x, \omega_{\bar{p}})|^2 &= \frac{1}{n} \|F\|^2, \\ |\epsilon_G(\omega_x, \omega_{\bar{p}})|^2 &= \frac{1}{n} \|G\|^2, \end{aligned} \quad (4)$$

which are inversely proportional to the sampling density  $n$ . In addition, the constants  $\|F\|^2$  and  $\|G\|^2$  are given by the total energy of the signals, that is, the integral over image and path dimensions



**Figure 4:** MSEs of sampled pixels  $|\epsilon_F|^2$  and gradients  $|\epsilon_G|^2$  for two image contribution functions, a (multidimensional) checkerboard, and band-limited noise, for different pattern frequencies  $\phi_c$  and  $\phi_n$ . For the checkerboard, the MSE of gradients oscillates between values zero and two, as the pattern switches between positive and negative correlation with the finite differencing stencil. The pixel MSE is independent of the checkerboard frequency, since we sample uniformly without stratification. The dotted line indicates the Nyquist frequency of the pixel grid. Even for pattern frequencies close to the Nyquist limit, gradients exhibit less MSE than pixels.

of the power spectra in Equation 3,

$$\begin{aligned} \|F\|^2 &= \int |F(\omega_x, \omega_{\bar{p}})|^2 d\omega_x d\omega_{\bar{p}}, \\ \|G\|^2 &= \int (2 - 2 \cos(2\pi\omega_x)) |F(\omega_x, \omega_{\bar{p}})|^2 d\omega_x d\omega_{\bar{p}}. \end{aligned} \quad (5)$$

The MSE appears as a flat gray background in Figure 3c and 3f. The difference in brightness of the flat background indicates the difference in MSEs of the sampled signals.

Next, we model integration over path space to obtain the sampled image and its sampled gradients by slicing in the Fourier domain. After slicing, we convolve with an ideal pixel filter, which eliminates errors in frequencies above the Nyquist limit  $\omega_x = 1/2$ . We denote the resulting MSE of the integrated pixels  $|\epsilon_F(\omega_x)|^2 = |\epsilon_F(\omega_x, 0)|^2$  and the MSE of the gradients  $|\epsilon_G(\omega_x)|^2 = |\epsilon_G(\omega_x, 0)|^2$ , and conclude

$$\begin{aligned} |\epsilon_F(\omega_x)|^2 &= \frac{1}{n} \|F\|^2, \quad \text{if } |\omega_x| < 1/2, \text{ otherwise } 0, \\ |\epsilon_G(\omega_x)|^2 &= \frac{1}{n} \|G\|^2, \quad \text{if } |\omega_x| < 1/2, \text{ otherwise } 0. \end{aligned} \quad (6)$$

Our MSEs come out as energies instead of variances here because of our sampling assumptions, which are crucial to be able to perform the derivation in simple terms. The means of our functions over the infinite sampling domain are zero, hence their energies represent their variances.

**Discussion** The integrals for  $\|F\|^2$  and  $\|G\|^2$  are the same except for the weight  $|D(\omega_x)|^2 = (2 - 2 \cos(2\pi\omega_x))$  introduced by finite differencing. *This reveals that the difference between the pixel error  $|\epsilon_F|^2$  and gradient error  $|\epsilon_G|^2$  depends on the relative amount of low and high frequencies in the image contribution function.* In the best case for gradient estimation, all the energy is in the low frequencies, and as  $|D(\omega_x)|^2$  weights them down, the gradient energy can be arbitrarily smaller. At worst, all the energy is in the high frequencies and gradient estimates are four times as bad as pixel estimates. Interestingly, the spectra of typical image contribution functions appear to be favorable to gradient estimation: like natural images, they are dominated by sharp edges, and hence follow an inverse square power law.

In Figure 4 we illustrate the benefits and limitations of gradient compared to pixel sampling for two prototypical signals. The first defines the image contribution function as a (multidimensional) checkerboard pattern, and the second uses band-limited noise, both with amplitude one. Both cover an infinite domain of arbitrary dimensionality (arbitrary many path parameters  $\bar{p}$ ). Dimensionality does not matter for our analysis since the difference operator  $|D|^2$  does not change the frequency spectra over the path dimensions.

For the checkerboard pattern (Figure 4a), we compare the MSE of the pixels  $|\epsilon_F|^2$  and the MSE of their gradients  $|\epsilon_G|^2$  for different frequencies  $\phi_c$  of the checkerboard tiles. Frequency  $\phi_c = 0.5$  means the side-length of tiles is one pixel unit (two pixels for one cycle of the pattern). For the noise pattern (Figure 4b), we show the same comparison for different band-limits  $\phi_n$  of the noise. Similarly,  $\phi_n = 0.5$  means noise is cut off at the Nyquist frequency of the pixel grid.

In both cases, as the pattern frequency increases, their energies become less and less dominated by low frequencies. Hence the gradient MSE increases, and climbs above the pixel MSE for high frequency patterns close to the Nyquist limit. This is to be expected since  $|D|^2$  amplifies the square magnitudes of these frequencies by a factor of up to four. But for frequencies not much below, gradient MSEs drop below pixel MSEs. For the square wave, the frequency where gradients have less MSE is at exactly  $\phi_c = 0.25$ , which corresponds to checkerboard tiles of only two pixels (four pixels per cycle). For the noise pattern, gradients have less MSE than pixels at even slightly higher noise cutoff frequencies  $\phi_n \approx 0.3$ . This indicates that gradient sampling is effective even for high-frequency patterns close to the Nyquist limit of the pixels.

## 4.2 Analysis of Screened Poisson Reconstruction

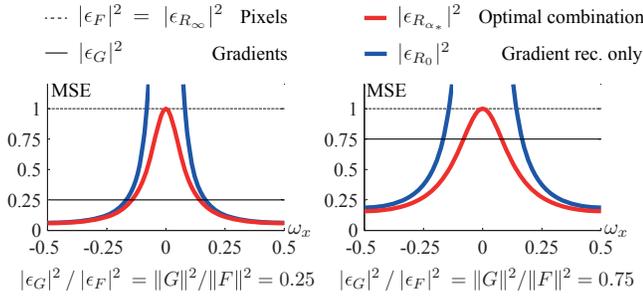
We conclude the end-to-end analysis by deriving the spectral MSE of the final image obtained through screened Poisson reconstruction. Screened Poisson reconstruction combines the image and its gradients using a parameter  $\alpha$  that specifies the relative weights of the sampled image and the gradients. We explicitly include the MSEs of the input image pixels  $|\epsilon_F|^2 = \|F\|^2/n$  and their gradients  $|\epsilon_G|^2 = \|G\|^2/n$  (Equation 4) in our analysis and derive the per-image frequency reconstruction error

$$|\epsilon_{R_\alpha}(\omega_x)|^2 = \frac{1}{n} \frac{\alpha^4 \|F\|^2 + |D(\omega_x)|^2 \|G\|^2}{(\alpha^2 + |D(\omega_x)|^2)^2}. \quad (7)$$

**Discussion** To clearly understand the benefits of Poisson reconstruction, we plot the frequency-dependent reconstruction error  $|\epsilon_{R_\alpha}(\omega_x)|^2$  for different values  $\alpha$  (Figure 5). If we set  $\alpha = \infty$ , we consider only the pixel information and the reconstruction error becomes equivalent to the pixel error  $|\epsilon_{R_\infty}|^2 = \|F\|^2/n$ . More interesting is setting  $\alpha = 0$ , which means we consider only gradients. Then the reconstruction error amounts to

$$\begin{aligned} |\epsilon_{R_0}(\omega_x)|^2 &= 1/n \cdot \|G\|^2 / |D(\omega_x)|^2 \\ &= 1/n \cdot \|G\|^2 / (2 - 2 \cos(2\pi\omega_x)). \end{aligned} \quad (8)$$

This confirms that gradient rendering is most beneficial for high frequencies and has a singularity for the DC. The image frequency where it may become beneficial depends on the relative energy of the path difference function and the image contribution function,  $\|G\|^2/\|F\|^2$ . In the worst case, we saw that  $\|G\|^2$  is four times bigger than  $\|F\|^2$ , and the factor four gets canceled by the denominator in Equation 8 at the Nyquist limit. Compared to direct pixel rendering, the gradient reconstruction error therefore is the same



**Figure 5: Poisson reconstruction combines noisy pixels with MSE  $|\epsilon_F|^2$  and gradients with MSE  $|\epsilon_G|^2$  using a parameter  $\alpha$ , weighting the influence of these two inputs. Reconstruction  $|\epsilon_{R_0}|^2$  with  $\alpha = 0$  (blue lines) uses only gradients. It reduces high frequency error, but error explodes at low frequencies. At the optimal  $\alpha_*$  (red lines), the reconstruction error is at most the pixel variance  $|\epsilon_F|^2$  at low frequencies, but is smaller than  $|\epsilon_F|^2$  by a factor  $> 4\|F\|^2/\|G\|^2$  at high frequencies. The signal on the left corresponds to a checkerboard pattern with a tile size of eight pixels ( $\|G\|^2/\|F\|^2 = 0.25$  corresponds to  $\phi_c = 0.0625$ , and  $1/(2 \times 0.0625) = 8$ ). The signal on the right corresponds to a tile size of about 2.5 pixels.**

for the highest frequency and worse for lower frequencies. As discussed above, typical image contribution functions tend to follow an inverse square power law. Hence finite differencing reduces their energies significantly and gradient reconstruction is beneficial for much of the spectrum except very low frequencies.

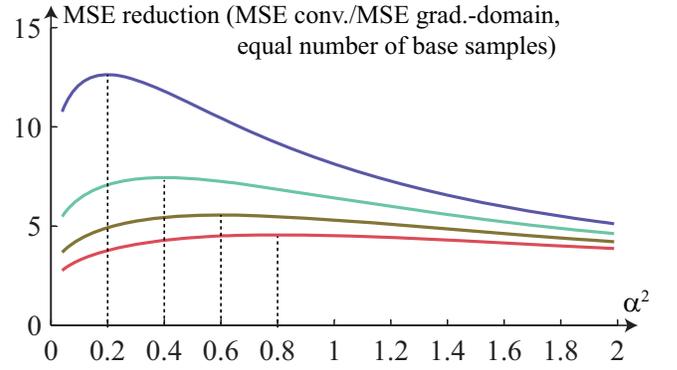
A key observation about Equation 8 is that the factor  $2 - 2\cos(2\pi\omega_x)$  in the denominator is the inverse of the finite differencing operator we applied in the beginning to sample path differences. Hence, both finite differencing and Poisson reconstruction correspond to a per-frequency weighting of the power spectrum by a cosine function, which appears both in the numerator and the denominator in Equation 8. However, the factors do not cancel each other out, because the error of sampling finite differences is hidden in the integral of the gradient energy  $\|G\|^2$  (Equation 5) due to the frequency scrambling or aliasing caused by stochastic sampling. It results in a lower energy and lower sampling error for gradients of functions with a frequency falloff. In contrast, the reconstruction affects the denominator and applies the inverse weight to the final image, which explains why gradient solutions reduce high frequency error.

**Optimal Reconstruction** It is easy to derive the value  $\alpha_*(\omega_x)$  at each frequency that optimally combines pixels and gradients as  $\alpha_*^2(\omega_x) = \|G\|^2/\|F\|^2$ . It is interesting to plot the per frequency error at the optimal value  $\alpha_*$ , which turns out to be

$$|\epsilon_{R_{\alpha_*}}(\omega_x)|^2 = \frac{1}{n} \frac{\|G\|^2\|F\|^2}{\|F\|^2|D(\omega_x)|^2 + \|G\|^2},$$

shown as red lines in Figure 5. At low frequencies, the reconstruction MSE approaches the pixel MSE  $\|F\|^2/n$ , but it falls off quickly. For high frequencies approaching  $\omega_x = 0.5$  we take full advantage of the gradients and the MSE goes below  $\|G\|^2/(4n)$ .

It is important to understand that so far we compared conventional sampling with  $n$  samples with gradient-domain rendering with  $n$  conventional and  $n$  gradient samples. In this setup gradient-domain rendering at optimal  $\alpha_*$  cannot be worse than conventional rendering. However, at equal number of individual path samples (counting



**Figure 6: Error reduction of gradient sampling and reconstruction for 2D images: We plot the factor by which gradient sampling and reconstruction reduces the total MSE compared to conventional sampling and integration at equal number of base samples. We show different energy ratios  $\|G\|^2/\|F\|^2 \in \{0.2, 0.4, 0.6, 0.8\}$ , corresponding to the curves from top to bottom (purple to red). Note that each curve achieves its maximum at  $\alpha^2 = \|G\|^2/\|F\|^2$ .**

each gradient as two path samples), gradient-domain rendering becomes ineffective if  $\|F\|^2/\|G\|^2 < 1$ . In addition, our practical algorithm obtains correlated gradient and conventional samples, and as a consequence, the optimal  $\alpha_*$  is not applicable as we discuss in Section 6.

**Main Insights** Our complete analysis leads to two main insights: first (Figure 3g), under our simplifications, the optimal reconstruction parameter is given by the ratio of the total energies of the gradient function and the image contribution function,  $\alpha_*^2 = \|G\|^2/\|F\|^2$ ; second (Figure 3h), gradient-domain rendering reduces high frequency variance of the reconstructed output compared to conventional sampling by more than  $4\|F\|^2/\|G\|^2$ .

Finally, we investigate the reduction in total (as opposed to per-frequency) MSE of gradient-domain over conventional rendering at equal number of base path samples, for 2D images. That is, we compare the MSE of conventional rendering with  $n$  samples to gradient-domain rendering with  $n$  base paths, and in addition, one horizontal and vertical offset path. For this we numerically integrate a 2D version of the per-frequency reconstruction error in Equation 7 over all image frequencies. We plot the resulting ratio of the total error of conventional compared to gradient-domain rendering in Figure 6 for various ratios energy ratios  $\|G\|^2/\|F\|^2 \in \{0.2, 0.4, 0.6, 0.8\}$  over values  $\alpha \in (0, 1)$ . For each ratio, the MSE reduction is best when that ratio is used as  $\alpha$ , as described above. We will demonstrate similar practical benefits in Section 6.

## 5 Gradient-Domain Path Tracing

We now introduce gradient-domain path tracing (G-PT), a simple extension of its standard counterpart [Kajiya 1986; Veach and Guibas 1995] that leverages additional gradient estimates (Algorithm 1). The gradient-domain path tracer draws a number of random paths using standard path sampling, and it writes their contributions into a “primal” image like in conventional renderers. The key difference is that we trace additional paths through neighboring pixels to compute finite differences (innermost loop in Algorithm 1). For each base path  $\bar{x} = (x, \bar{p})$  that contributes to a pixel  $i$ , we obtain the offset paths needed for finite-difference computation with a set of neighboring pixels  $j \in \Phi_i$  by applying a shift mapping  $T_{ij}$ . We use each offset path  $\bar{y} = T_{ij}(\bar{x})$  to compute one gradient

**Input:** Scene and camera specification, number of samples  $N$ .

**Output:** Path-traced image  $I$ , gradient images  $\Delta_{\cdot,j}$ .

```

for all sampled base paths  $\bar{x} = (x, \bar{p})$  do
  for all pixels  $i$  where  $h(x - x_i) > 0$  do
    // Write path contribution to primal image
     $I_i := I_i + h(x - x_i)f(\bar{x})/p(\bar{x})$ 
    for all neighbor pixels  $j \in \Phi_i$  of  $i$  do
       $\bar{y} := T_{ij}(\bar{x});$  // offset path using shift  $T_{ij}$ 
      // gradient MIS weight  $w_{ij}(\bar{x})$  see Section 5.1
       $\Delta_{i,j} := \Delta_{i,j} + w_{ij}(\bar{x})h(x - x_i)(f(\bar{x}) - f(\bar{y})|T'_{ij}|)$ 
    end
  end
end
 $I := I/N; \Delta_{\cdot,j} := \Delta_{\cdot,j}/N,$  for all  $j$ 
Reconstruct( $I, \Delta_{\cdot,\cdot}, \alpha$ )
    
```

**Algorithm 1:** Pseudocode for gradient-domain path tracing, where we sample gradients in the innermost loop. Here  $N$  is the total number of samples.

sample by taking the difference to the base path. We store the gradient samples in gradient images  $\Delta_{\cdot,j}$ , that is, we have one gradient image for each neighbor index  $j$ . Finally we perform screened Poisson reconstruction to combine primal and gradient values as in previous work [Lehtinen et al. 2013; Manzi et al. 2014].

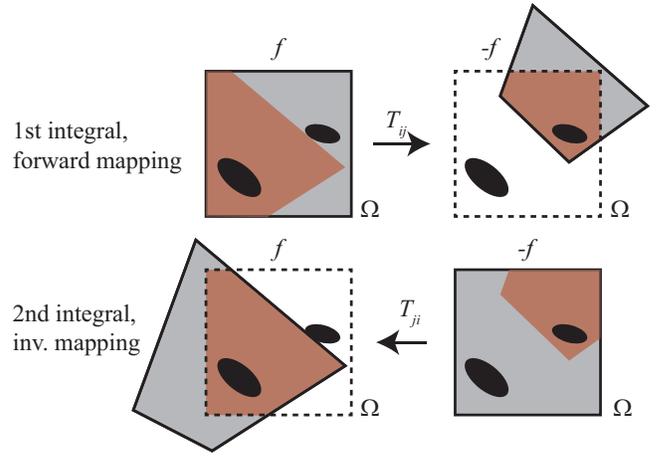
While the principles of gradient rendering are simple, we must make sure that sampling our single integral of the path difference function (Equation 2), in fact, corresponds to the difference of the two corresponding pixel integrals (Equation 1). We must pay attention to the invertibility of the shift to make sure all path differences are considered equally. We achieve this by sampling a given pixel difference both as a forward or backward finite difference, as in Algorithm 1, and weighting them appropriately. In addition, this provides us with an opportunity to reduce variance through multiple importance sampling (MIS), which is highly effective to suppress gradient sampling artifacts. Finally, shift maps proposed in previous work can be hard to implement and costly to compute because they rely on manifold perturbation [Jakob and Marschner 2012]. We propose a simpler shift map that is suitable for path tracing.

## 5.1 Symmetric Gradients and MIS

The gradient integral in Equation 2 only computes pixel differences correctly if the shift mapping is a bijection of all of path space sampled by the Monte Carlo path tracer onto itself. In practice, it is hard to design such shift mappings. We build on the symmetric gradient formulation by Manzi et al. [2014] to obtain a formulation that does not impose such requirements. In addition, we adapt the formulation to include continuous weights, which we will exploit to reduce gradient sampling artifacts using MIS.

**Symmetric Gradients** A practical shift mapping may not generate all paths  $\Omega$  that can be sampled by the path tracer, and it may also produce blocked paths that will never be sampled by the path tracer (i.e., it may sample paths outside  $\Omega$ ; their contribution will always be zero). First, assuming the shift is invertible, we compute gradients symmetrically using the sum of two integrals, one using the original forward shift and the other its inverse (Figure 7),

$$\Delta_{i,j} = \left( h(x) * \int_{\Omega} w_{ij}(x, \bar{p}) g_{ij}(x, \bar{p}) d\mu(\bar{p}) \right) (x_i) + \left( h(x) * \int_{\Omega} w_{ji}(x, \bar{p}) g_{ji}(x, \bar{p}) d\mu(\bar{p}) \right) (x_j). \quad (9)$$

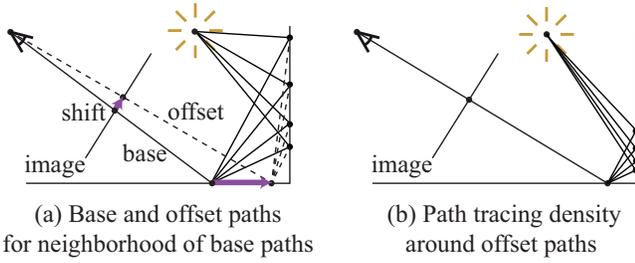


**Figure 7:** To compute a correct gradient we need to integrate  $f$  over all of path space  $\Omega$  twice, once with positive and once with negative sign (Equation 1). Shift mappings may not be invertible in some regions  $\Omega_0$  (black), and we fall back to computing naive gradients there. In addition, the shift may not cover all of  $\Omega$ , and may map some paths to regions outside  $\Omega$ . The forward mapping formulation correctly integrates  $f$  over  $\Omega \setminus \Omega_0$ , but covers only a part of  $\Omega \setminus \Omega_0$  with  $-f$ , and similar for the inverse mapping. Red areas are covered twice, and by weighting them with weights that sum to one we make sure to get their correct contribution.

We observe that if both a base path and its offset path can be sampled by the path tracer then the corresponding path difference occurs in both integrals. We call such base-offset pairs *symmetric*. We handle symmetric paths by introducing weights  $w_{ij}$  and  $w_{ji}$ , and making sure they add up to one. If an offset path cannot be sampled by the path tracer, that is  $T_{ij}(x, \bar{p}) \notin \Omega$ , we simply set  $w_{ij}(x, \bar{p}) = 1$ , and similarly for the second integral.

Second, we also allow shift mappings  $T_{ij}$  that may not have an inverse, denoted  $T_{ji}$ , in a subspace  $\Omega_0$  of all sampled base paths  $\Omega$ . We deal with non-invertibility by simply evaluating the gradient naively over  $\Omega_0$  using Equation 1. We implement this by ignoring offset paths  $T_{ij}(\bar{x})$  generated by non-invertible shifts, and setting the weights to  $w_{ij}(\bar{x}) = 1$  and  $w_{ji}(\bar{x}) = 0$ .

**Multiple Importance Sampling for Gradients** We introduce a multiple importance sampling (MIS) technique for gradients that addresses an important issue that has hampered previous gradient-domain rendering techniques [Lehtinen et al. 2013; Manzi et al. 2014]. Intuitively, shift mappings cause a change of path densities (Figure 8). If one were to shift a set of close-by base paths to a set of offset paths, the density of the obtained offset paths will be different from the density in that region that the underlying path sampler (a path tracer in our case) would produce. Mathematically, the Jacobian determinant of the shift accounts for this stretching or squeezing of path space. Unfortunately, this Jacobian determinant can become arbitrarily large if the shift moves a path into a region of path space that is sampled with much higher density by the underlying path sampler. This forces  $g_{ij}(x, \bar{p})$  to take large values, and yields samples of high variance that produce strong artifacts, in particular when using  $L_2$  Poisson reconstruction. It is important to observe that the inverse case, shifting from a high to a low path density region, is much less problematic. In this case, an arbitrarily small Jacobian determinant (Equation 2) simply means that we will record the base path as the gradient, which is not worse than what happens in naive gradient sampling (Equation 1).



**Figure 8:** A simple shift (like case (iii) in Section 5.2) maps a neighborhood of base paths to corresponding offset paths by moving a vertex on the base paths. The density of the shifted offset paths in (a) is much lower than the density produced by the path tracer in (b) in that region. The Jacobian for the shift compensates for the change in path density, and it becomes arbitrarily large as the shifted vertex approaches the corner.

We exploit the two ways we have available for sampling symmetric gradients to address this issue, and weigh them using a novel MIS strategy for gradients. We interpret the forward and inverse mappings as two sampling techniques to obtain the same base path  $\bar{x}$ : (1) we sample  $\bar{x}$  directly, or (2), we sample an offset path  $\bar{y} = T_{ij}(\bar{x})$ , and then map  $\bar{y}$  to  $\bar{x}$  via the inverse shift,  $\bar{x} = T_{ji}(\bar{y})$ . To compute MIS weights we need to derive the probability densities  $p_1(\bar{x})$  and  $p_2(\bar{x})$  corresponding to these two techniques.

Let us denote the probability density for paths generated by our path sampler as  $p$ , and observe that both paths  $\bar{x}$  and  $\bar{y}$  in our description are distributed according to  $p$ . This means  $p_1(\bar{x}) = p(\bar{x})$ . To obtain  $p_2(\bar{x})$ , we evaluate  $p(\bar{y})$  by interpreting  $\bar{y}$  as a function of  $\bar{x}$ . In other words, we substitute  $\bar{y} = T_{ij}(\bar{x})$ , and following the rules for variable substitutions for probability densities this yields  $p_2(\bar{x}) = p(T_{ij}(\bar{x}))|T'_{ij}(\bar{x})|$ . This leads to the balance heuristic weights

$$w_{ij}(\bar{x}) = \frac{p(\bar{x})}{p(\bar{x}) + p(T_{ij}(\bar{x}))|T'_{ij}(\bar{x})|}. \quad (10)$$

We see that gradients with a large Jacobian determinant  $|T'_{ij}(\bar{x})|$  obtain a MIS weight of approximately  $1/|T'_{ij}(\bar{x})|$ , which cancels their large contribution. In Algorithm 1, we need to check if a path  $\bar{x}$  is symmetric. If this is the case, we use the MIS weight for  $w_{ij}(\bar{x})$ , otherwise the weight is one. We show in Figure 9 how MIS effectively reduces gradient sampling artifacts.

## 5.2 A Novel Shift Mapping for G-PT

We demonstrated in Section 4 that the energy  $\|G\|^2$  of the path difference function  $g(x, \bar{p})$  directly affects the final error in the Poisson reconstructed image (Equation 7); hence shift mappings should minimize this energy. This happens if they produce offset paths whose image contributions (times their Jacobians) are as similar as possible to those of their base paths, such that the integrand  $g(x, \bar{p}) = f(x, \bar{p}) - f(T_{ij}(x, \bar{p}))|T'_{ij}|$  contributes as little as possible to the total energy  $\|G\|^2$ . Lehtinen et al. [2013] and Manzi et al. [2014] developed such mappings in the context of gradient-domain Metropolis sampling. We propose a simpler alternative that is more efficient to compute because its Jacobian determinant only relies on local information, and we can compute it step by step without requiring costly numerical optimization inherent to the manifold perturbation [Jakob and Marschner 2012] in previous work.

Intuitively, we achieve similar path contributions when the geometry of the base and offset paths are similar, and when the BSDF values at each vertex do not change much. A natural idea is hence



**Figure 9:** Effect of multiple importance sampling on  $L_2$  reconstruction quality at one sample per pixel. Without MIS (right) large Jacobians close to concave edges lead to gradient outliers, and they cause artifacts in the reconstruction.

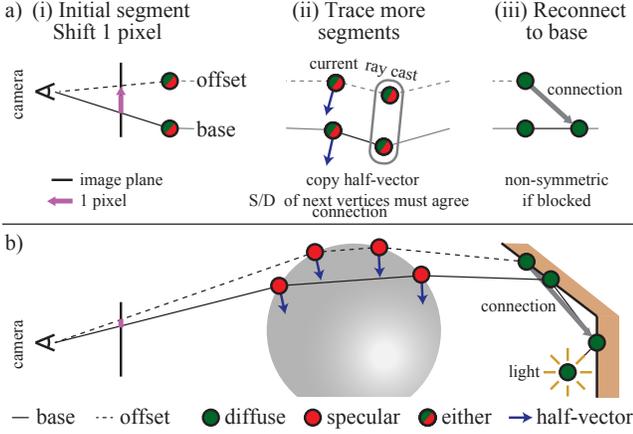
to re-use as many segments of the base path in the offset path as possible. The offset path, which starts with a shifted segment at the eye, should be reconnected to the base path as soon as possible. We need to be careful, however, when choosing the connecting segment to the base path. If (near-)specular interactions are involved, this may lead to large changes in BSDF values and path contributions, which we want to avoid. Therefore, our strategy is to replicate the half-vectors (projected to the local tangent plane) of the base path in the offset path as long as we encounter (near-)specular interactions. We connect to the base path only when the connecting segment does not include any (near-)specular vertices. This reasoning follows Jakob and Marschner [2012] and Kaplanyan et al. [2014], who show that for glossy materials path throughput is a mostly smooth function in the half-vector parameterization. Similar to Jakob and Marschner [2012] we also classify each path vertex as either diffuse or specular using a threshold on material roughness to detect (near-)specular interactions.

We illustrate a shift mapping based on these ideas in Figure 10.

**Step (i): Initial Offset Segment** We start from the camera and shift the direction of the primary ray by one pixel. This results in a new primary hit on the offset path (Figure 10a, step (i)).

**Step (ii): Tracing Additional Segments** If the current or next base vertex is classified as specular, we continue tracing offset segments to avoid large changes in (near-)specular BSDFs. We determine the incident direction for the next offset segment from the projected half-vector of the corresponding base vertex (Figure 10a, step (ii)). We need to be careful, however, to check whether the resulting shift is invertible. First, the shift is not invertible if the next base and the new next offset vertex have a different specular/diffuse classification. Second, consider a refraction that leads to total internal reflection after the shift. The reverse shift of a reflected path will also be a reflection, hence the shift is not invertible. We deal with these issues by rejecting these offset paths (Section 5.1).

**Step (iii): Reconnect** If the current offset vertex and both the current and next base vertex are classified as diffuse, we reconnect to the base path by setting the incident direction at the current offset vertex to point at the position of the next base vertex (Figure 10a, step (iii)). The connection, however, may be occluded. This means this offset path would never be sampled by the path tracer as a base path (with an important exception, explained in the next paragraph). Hence the base-offset pair is not symmetric according to our definition in Section 5.1, and we set the gradient weight to 1. Otherwise we compute MIS weights as described in Section 5.1. After a successful connection, the rest of the offset path coincides with the remaining base path. Finally, if the next base vertex is a point light,



**Figure 10:** *a) Constructing the offset path (see text). b) An example base/offset pair constructed by the shift mapping. As soon as we encounter two consecutive diffuse vertices on the base path, we connect the offset path to the second of the two.*

we always connect to it. If we can never connect to a light source, the image contribution of the offset path is zero.

We need to pay special attention when reconnecting to a base vertex on an area light source. The path tracer has two techniques to sample emitter vertices, either using BSDF or area sampling. Hence there are four instead of only two sampling strategies to compute a path difference: the forward and inverse shift for the base-offset pair, multiplied by the two strategies for sampling the emitter vertex. We include all four strategies in our MIS weighting. As a final subtlety, if the path tracer samples a base vertex on the emitter using area sampling, it may produce occluded paths. Consequently, in this special case (in contrast to the previous paragraph) an occluded connecting segment does not imply that the base-offset pair must be non-symmetric. We must, however, treat all sampling strategies consistently, hence we still apply the same rule as above, which is to treat path-pairs that include any occlusion as non-symmetric. This means we ignore the offset path, and weigh the base path with conventional MIS for emitter sampling only.

We show an example base-offset pair generated by this shift in Figure 10b. The procedure implies that as soon as we encounter two consecutive diffuse vertices on the base path, we connect the offset path to the second of the two. Since the shift is a concatenation of independent steps, each step adds a separate factor to the Jacobian determinant. We give the results here and refer to the supplemental material and Stam [2001] for details.

**Jacobians** The Jacobian for step (i) is unity. In step (ii), we need to distinguish between reflection and refraction. Let us denote the incident directions in the base path  $\bar{x}$  and shifted offset path  $\bar{y}$  by  $\omega_i^x$  and  $\omega_i^y$ , and the projected half-vectors  $\mathbf{h}^x$  and  $\mathbf{h}^y$ . The outgoing directions (towards the eye) in the base and offset paths are  $\omega_o^x$  and  $\omega_o^y$ . The Jacobian determinant for reflection is

$$\left| \frac{\partial \omega_i^y}{\partial \omega_i^x} \right| = \left| \frac{\partial \omega_i^y}{\partial \mathbf{h}^y} \right| \left| \frac{\partial \mathbf{h}^x}{\partial \omega_i^x} \right| = \frac{\omega_o^y \cdot \mathbf{h}^y}{\omega_o^x \cdot \mathbf{h}^x}, \quad (11)$$

and the Jacobian determinant for refraction is

$$\left| \frac{\partial \omega_i^y}{\partial \omega_i^x} \right| = \left| \frac{\partial \omega_i^y}{\partial \mathbf{h}^y} \right| \left| \frac{\partial \mathbf{h}^x}{\partial \omega_i^x} \right| = \frac{\left| \omega_i^y + \frac{n_2^y}{n_1^y} \omega_o^y \right|^2 \omega_i^x \cdot \mathbf{h}^x}{\left| \omega_i^x + \frac{n_2^x}{n_1^x} \omega_o^x \right|^2 \omega_i^y \cdot \mathbf{h}^y}, \quad (12)$$

where  $n_1^x$  and  $n_2^x$  are the refractive indices on the base path  $\bar{x}$ , and similarly for the offset path  $\bar{y}$ . For step (iii), which reconnects two consecutive diffuse vertices, we denote the positions of the two vertices on the base path as  $\mathbf{x}_1^x$  and  $\mathbf{x}_2^x$ , and the cosine of the angle between surface normal and path segment between  $\mathbf{x}_1^x$  and  $\mathbf{x}_2^x$  as  $\theta_2^x$ . We use analogous notation for the offset path  $\bar{y}$ . The Jacobian simply consists of the ratio of two geometry terms,

$$\left| \frac{\partial \omega_i^y}{\partial \omega_i^x} \right| = \left| \frac{\partial \omega_i^y}{\partial \mathbf{x}_2^y} \right| \left| \frac{\partial \mathbf{x}_2^x}{\partial \omega_i^x} \right| = \frac{\cos \theta_2^y}{\cos \theta_2^x} \frac{|\mathbf{x}_1^x - \mathbf{x}_2^x|^2}{|\mathbf{x}_1^y - \mathbf{x}_2^y|^2}. \quad (13)$$

Finally, observe that if the connecting vertex  $\mathbf{x}_2$  on the base path was sampled on an area light source, then the corresponding geometry term cancels out with the geometry term  $\cos \theta_2^x / |\mathbf{x}_1^x - \mathbf{x}_2^x|^2$  of the base path here. The Jacobian determinant for connecting to environment maps is one.

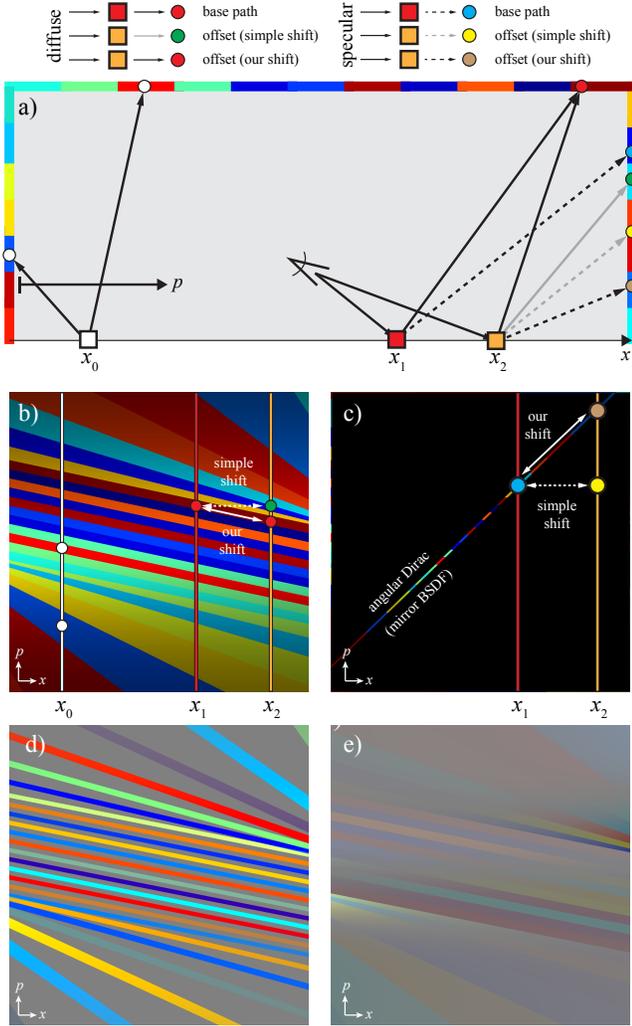
**Analysis** We study a simple 2D scene (Figure 11) to illustrate why our shift leads to smaller-energy gradients compared to the trivial alternative of always copying the incident direction (corresponding to the simple shift in the analysis in Section 4). The scene consists of walls and a ceiling made of colored emissive tiles, with the camera looking at the floor (Figure 11a).

First we assume a diffuse ground plane. Figure 11b shows the path throughput (incident radiance times cosine) parameterized over the ground plane ( $x$  axis) and incident direction ( $p$  axis), which we define using the tangent of the angle of incidence. We form a base path by connecting the camera to  $x_1$  (red square) on the ground and the red dot on the ceiling. We visualize the base path as the red circle in (b) that lies on the line  $x_1$ . We obtain the first segment of the offset path by shifting the direction of the primary ray, resulting in  $x_2$  (orange square). The simple shift copies the incident direction, yielding the green secondary hit (respectively in (b), the green circle on line  $x_2$ ). In contrast, our shift connects the primary hit of the offset  $x_2$  to the secondary hit of the base path (in (b), red circle on line  $x_2$ ). In (b) we see how our shift attempts to follow the isophote in path throughput by reconnecting to the original secondary hit. The path difference function is the subtraction of the path throughputs between the base and offset path, as determined by the shift mapping, multiplied by the Jacobian determinant. We show this in (d) for the simple shift, and in (e) for our shift (we visualize 0.5 + signed difference). Our shift results in lower variance (0.002 vs. 0.046), which translates directly to better image quality according to Equation 7.

Let us now assume the ground plane is a mirror. The corresponding path throughput, a Dirac distribution over angle, is visualized in (c). We show a base path connecting the camera,  $x_1$ , and the blue circle in the mirror direction. Copying the incident direction after the shift (in (a) and (c), yellow circle connected to  $x_2$ ) breaks the mirror geometry, and no light is carried by the offset path. Our half-vector copying shift (in (a) and (c), brown dot connected to  $x_2$ ), on the other hand, stays on the mirror manifold and results again in a lower-variance path difference function.

## 6 Results and Discussion

We implemented G-PT on top of the standard path tracer in the open source Mitsuba renderer. Our gradient path tracer shifts each base path to its four vertical and horizontal neighbors, resulting in standard finite difference gradient estimates. This means we compute four offset paths for each base path. Since our shift mapping typically connects offset paths back to base paths after tracing only a few new offset path segments, the computational cost of offset paths is significantly lower than for base paths. In practice, we observe



**Figure 11:** Analysis of shift mappings. (a) A 2D scene consisting of a ground plane, colored emissive tiles, and a camera pointing to the floor. Camera rays ending at  $x_1$  are base paths, and the ones ending at  $x_2$  are offset paths. (b) Assuming a diffuse ground plane, path throughput  $f(x, \bar{p})$  visualized over the  $x$  coordinate on the ground plane and incident direction  $p$  (tangent parameterization). The locations of the vertical lines ( $x_0, x_1, x_2$ ) correspond to the points in (a). The colored circles correspond to rays in (a). The relationship of the simple shift mapping (that copies incident direction) and our shift mapping (that reconnects to the second base path vertex) is visualized by the arrows. The energy is  $\|F\|^2 = 0.06$ . (c) Assuming a mirrored ground plane, the path throughput is a Dirac in angle. Again, the base and offset paths are visualized for both the simple shift and our shift. Our shift remains on the specular submanifold. (d) The finite difference  $g(x, \bar{p}) = f(x, \bar{p}) - f(x - 1, \bar{p})$  for the simple shift, computed from (b). The energy is  $\|G\|^2 = 0.046$ . (e) The finite difference for our shift, computed from (b). The energy is  $\|G\|^2 = 0.002$ .

an overhead of about a factor 2.5 between our method with  $n$  base and  $4n$  offset samples per pixel compared to standard path tracing with  $n$  samples per pixel. Slightly deviating from the pseudocode in Algorithm 1, we also apply our MIS technique (Section 5.1) to the samples of the image contribution function. For each base and offset path, we write their MIS weighted image contribution to the

base and offset pixel, respectively. We found that this provides a small additional error reduction.

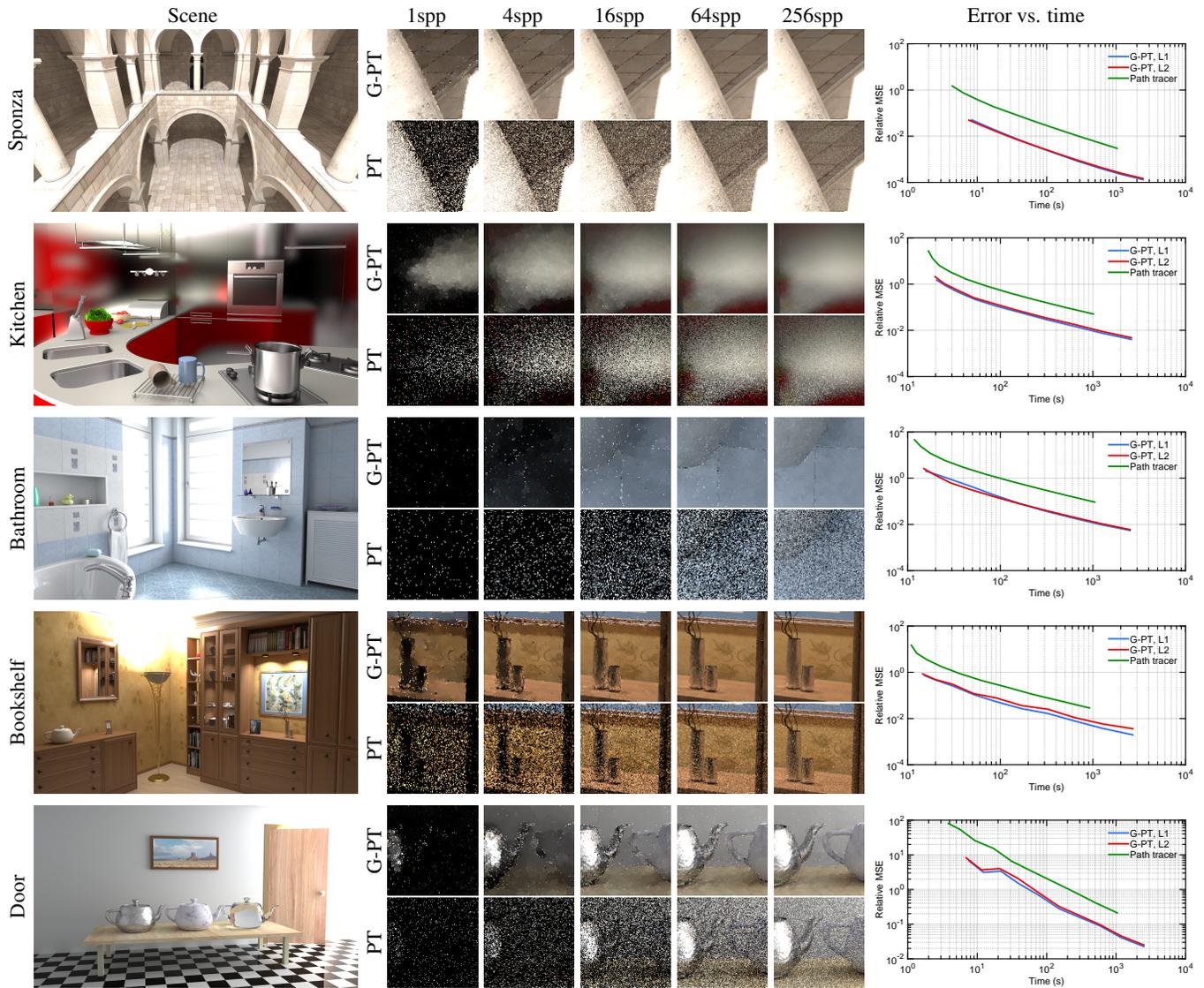
We provide an overview of results and comparisons to standard path tracing in Figure 12. The convergence plots on the right show the relative mean square error (relMSE) of PT and G-PT as a function of render time. We compute per-pixel relMSE by summing  $(\text{img} - \text{ref})^2 / (\text{grayscale}(\text{ref})^2 + \epsilon)$  over the color channels. We use  $\epsilon = 0.001$  and report the average error over all pixels. The convergence plots show that we achieve a relMSE reduction by a factor of five to twelve, which is a render time advantage by the same factor to obtain equal numerical error. Diffuse scenes like Sponza favor G-PT the most, because the shift mapping always connects to the second base vertex (excluding the eye) and BSDF values stay constant under the shift. Scenes with many glossy surfaces like Kitchen are more challenging because they lead to higher path differences and noisier gradients, but G-PT still achieves a significant error reduction. The close-ups show results at equal number of base samples. Because of the 2.5 times overhead of G-PT, our images should be compared to the next PT image diagonally down to the right for an approximately equal time comparison (slightly skewed in favor of PT, since PT has  $4/2.5 = 1.6$  times longer to compute these images). The close-ups reveal how G-PT effectively reduces high frequency noise without blurring texture detail or geometric edges, which is most apparent in Sponza and Bookshelf. Even in scenes that are notoriously hard to render for PT, like Door, G-PT provides a significant advantage. It cannot, however, avoid artifacts and outliers due to the underlying unidirectional path sampling strategy.

We examine the influence of the  $L_2$  reconstruction parameter  $\alpha$  on the relMSE for our five scenes in Figure 13. We plot the error reduction compared to conventional rendering at an equal number of base samples, that is, conventional rendering with  $n$  samples is compared to gradient-domain rendering with  $n$  base samples and four horizontal and vertical offset samples (at equal render time our advantage is about 2.5 times lower). These plots are the same for different  $n$ . It is interesting to observe the correspondence to the theoretical prediction in Figure 6. While the shapes of the curves are remarkably similar, at the same values for the optimal  $\alpha$  we obtain about twice the error reduction in practice than predicted by the theory. We explain this with the fact that we use four highly correlated gradient samples in practice (they all reuse the same base path), whereas the theory assumes two gradient samples that are uncorrelated to each other. For the same reason, the optimal  $\alpha$  parameter predicted by the theory usually overestimates  $\alpha$ , which means it gives too much weight to the primal image. In practice, we prefer  $L_1$  reconstruction because it suppresses occasional outliers. In addition, its quality is rather independent of  $\alpha$  and a standard setting of  $\alpha = 0.2$  leads to results that are indistinguishable from the optimal value for all our scenes.

Finally, we compare gradient path tracing to gradient-domain Metropolis sampling in Figure 14. The figure shows typical non-uniform convergence behavior of MLT, while G-PT performs similarly well. This figure demonstrates that G-PT can take advantage of gradients similarly effectively as G-MLT in scenes where the underlying path sampler performs reasonably well.

## 7 Conclusions

We have presented gradient-domain path tracing, a simple variation of standard path tracing that leverages correlated path samples to estimate finite pixel differences. We provided a theoretical analysis of this approach that sheds light on the benefits and limitations of such an approach. We find that for typical image contribution functions, which are dominated by low frequency energy like natural images, taking finite differences between close-by paths in path



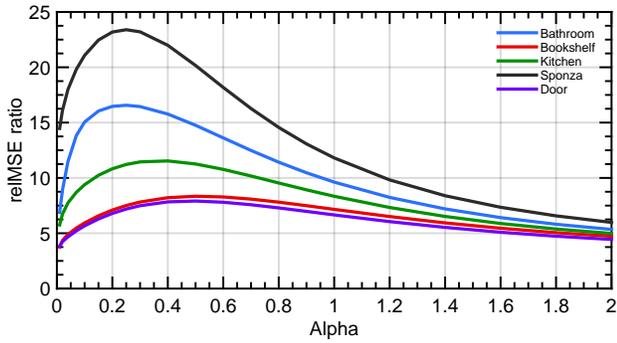
**Figure 12:** We compare conventional path tracing (PT) to gradient-domain path tracing (G-PT) on five scenes. Convergence plots on the right show the relative mean squared error (relMSE) as a function of rendering time. G-PT reduces relMSE by a factor of five to ten at equal time. Counting only base path samples, G-PT has a performance overhead of a factor 2.5 to compute four offset paths. Hence the close-ups of G-PT (with  $L_1$  reconstruction) should be compared to PT diagonally down to the right for an approximately equal time comparison, although PT takes  $4/2.5 = 1.6$  times longer to render the corresponding images. G-PT effectively reduces high frequency noise without blurring details.

space removes much of the energy from the signal. This leads to lower variances in the sampled estimates, which we exploit in a Poisson reconstruction step. Relying on pixel differences only to reconstruct an image amounts to inverting the finite differencing operator. This works well for high frequencies, but it cannot recover low frequency information. By combining gradient information with conventionally sampled pixels, however, we obtain a result that preserves the low frequencies of the conventional image and takes advantage of the lower variance in the sampled gradients to suppress high frequency noise.

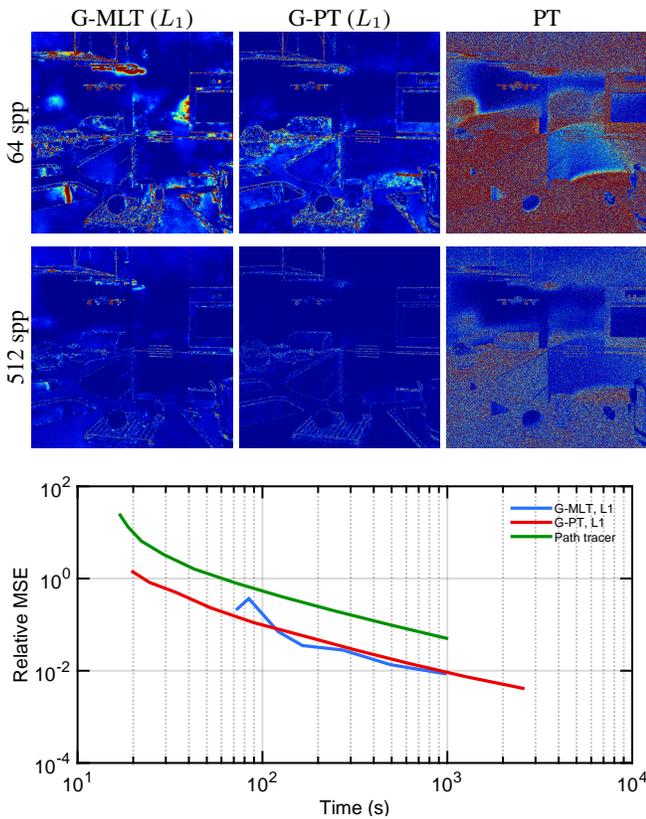
We put these insights into practice by developing a gradient-domain path tracer. We developed a novel shift mapping that is simple to compute and effective at producing highly correlated path pairs. In

addition, we introduced a multiple importance sampling technique that suppresses gradient outliers that can be introduced when the shift mapping introduces large distortions in path space. Our approach provides significant improvements over standard path tracing at equal render time, reducing squared error by up to an order of magnitude.

Our approach builds on the simplicity of standard unidirectional path tracing, but it is also limited by its inability to robustly sample light paths in difficult scenarios. Most importantly, we believe our study reveals the opportunities that gradient-domain rendering will provide for other Monte Carlo sampling strategies. In the future, we will explore gradient-domain bidirectional path tracing and related approaches. In addition, our results open up research challenges re-



**Figure 13:** Relative MSE (*relMSE*) of path tracing over *relMSE* of gradient-domain path tracing at the same number of base samples, and for different values  $\alpha$ . Lower optimal  $\alpha$  values indicate lower variance in the sampled gradients, and they correspond to better error reduction. Note the correspondence to Figure 5.



**Figure 14:** Comparing the characteristics of the error between gradient-domain MLT, gradient-domain path tracing, and regular path tracing. At the top we visualize the error of G-MLT, G-PT, and PT to a reference image of the Kitchen scene, and we show equal-time convergence plots at the bottom.

lated to various aspects of gradient-domain Monte Carlo rendering, including the development of more advanced shift mappings, custom tailored shift mappings for specific effects or materials, adaptive sampling, and so on.

## Acknowledgments

The “Door” scene was modeled after a scene by Eric Veach by Miika Aittala, Samuli Laine, and Jaakko Lehtinen. The “Sponza” scene is courtesy of Marko Dabrovic. The “Bookshelf” and “Bathroom” scenes have been ported to Mitsuba by Tiziano Portenier. This work was partially supported by the Academy of Finland grant no. 277833, NSF grant no. IIS-1420122, Swiss National Science Foundation grant no. 143886, and by the Helsinki Doctoral Education Network in Information and Communications Technology (HICT).

## References

BELCOUR, L., SOLER, C., SUBR, K., HOLZSCHUCH, N., AND DURAND, F. 2013. 5D covariance tracing for efficient defocus and motion blur. *ACM Trans. Graph.* 32, 3 (July), 31:1–31:18.

BELCOUR, L., BALA, K., AND SOLER, C. 2014. A local frequency analysis of light scattering and absorption. *ACM Trans. Graph.* 33, 5 (Aug.).

BHAT, P., CURLESS, B., COHEN, M., AND ZITNICK, C. 2008. Fourier analysis of the 2D screened poisson equation for gradient domain problems. In *Computer Vision, ECCV 2008*, vol. 5303 of *Lecture Notes in Computer Science*. Springer, 114–128.

BHAT, P., ZITNICK, L., COHEN, M., AND CURLESS, B. 2010. GradientShop: A gradient-domain optimization framework for image and video filtering. *ACM Trans. Graph.* 29, 2, 10:1–10:14.

BOLIN, M. R., AND MEYER, G. W. 1995. A frequency based ray tracer. In *Proc. ACM SIGGRAPH 95*, 409–418.

DIPPÉ, M. A. Z., AND WOLD, E. H. 1985. Antialiasing through stochastic sampling. *SIGGRAPH Comput. Graph.* 19, 3 (July), 69–78.

DURAND, F., HOLZSCHUCH, N., SOLER, C., CHAN, E., AND SILLION, F. X. 2005. A frequency analysis of light transport. *ACM Trans. Graph.* 24, 3, 1115–1126.

DURAND, F. 2011. A frequency analysis of Monte Carlo and other numerical integration schemes. Tech. Rep. MIT-CSAIL-TR-2011-052, MIT CSAIL.

GEORGIEV, T. 2005. Image reconstruction invariant to relighting. In *Proc. Eurographics 2005*, 61–64.

IGEHY, H. 1999. Tracing ray differentials. In *Proc. ACM SIGGRAPH '99*, 179–186.

JAKOB, W., AND MARSCHNER, S. 2012. Manifold exploration: A Markov Chain Monte Carlo technique for rendering scenes with difficult specular transport. *ACM Trans. Graph.* 31, 4, 58:1–58:13.

JAKOB, W., 2012. Mitsuba v0.4. <http://mitsuba-renderer.org>.

JAROSZ, W., DONNER, C., ZWICKER, M., AND JENSEN, H. W. 2008. Radiance caching for participating media. *ACM Trans. Graph.* 27, 1 (Mar.).

JAROSZ, W., ZWICKER, M., AND JENSEN, H. W. 2008. Irradiance gradients in the presence of participating media and occlusions. *Computer Graphics Forum* 27, 4 (June), 1087–1096.

JAROSZ, W., SCHÖNEFELD, V., KOBELT, L., AND JENSEN, H. W. 2012. Theory, analysis and applications of 2D global illumination. *ACM Trans. Graph.* 31, 5 (Sept.), 125:1–125:21.

- KAJIYA, J. T. 1986. The rendering equation. In *Proc. ACM SIGGRAPH 86*, 143–150.
- KAPLANYAN, A. S., HANIKA, J., AND DACHSBACHER, C. 2014. The natural-constraint representation of the path space for efficient light transport simulation. *ACM Trans. Graph.* 33, 4.
- KELEMEN, C., SZIRMAY-KALOS, L., ANTAL, G., AND CSONKA, F. 2002. A simple and robust mutation strategy for the Metropolis light transport algorithm. *Computer Graphics Forum* 21, 3, 531–540.
- KŘIVÁNEK, J., GAUTRON, P., PATTANAIK, S., AND BOUATOUCH, K. 2005. Radiance caching for efficient global illumination computation. *IEEE Trans. Vis. and Compu. Graph.*, 11, 5, 550–561.
- LEHTINEN, J., KARRAS, T., LAINE, S., AITTALA, M., DURAND, F., AND AILA, T. 2013. Gradient-domain metropolis light transport. *ACM Trans. Graph.* 32, 4, 95:1–95:12.
- LENEMAN, O. A. 1966. Random sampling of random processes: Impulse processes. *Information and Control* 9, 4, 347 – 363.
- LI, T.-M., WU, Y.-T., AND CHUANG, Y.-Y. 2012. Sure-based optimization for adaptive sampling and reconstruction. *ACM Trans. Graph.* 31, 6, 194:1–194:9.
- MANZI, M., ROUSSELLE, F., KETTUNEN, M., LEHTINEN, J., AND ZWICKER, M. 2014. Improved sampling for gradient-domain metropolis light transport. *ACM Trans. Graph.* 33, 6 (Nov.), 178:1–178:12.
- OVERBECK, R., DONNER, C., AND RAMAMOORTHI, R. 2009. Adaptive wavelet rendering. *ACM Trans. Graph.* 28, 5, 140:1–140:12.
- PÉREZ, P., GANGNET, M., AND BLAKE, A. 2003. Poisson image editing. *ACM Trans. Graph.* 22, 3, 313–318.
- RAMAMOORTHI, R., MAHAJAN, D., AND BELHUMEUR, P. 2007. A first-order analysis of lighting, shading, and shadows. *ACM Trans. Graph.* 26, 1 (Jan.).
- ROUSSELLE, F., KNAUS, C., AND ZWICKER, M. 2011. Adaptive sampling and reconstruction using greedy error minimization. *ACM Trans. Graph.* 30, 6, 159:1–159:12.
- ROUSSELLE, F., MANZI, M., AND ZWICKER, M. 2013. Robust denoising using feature and color information. *Computer Graphics Forum* 32, 7, 121–130.
- RUDERMAN, D. 1994. The statistics of natural images. *Network: computation in neural systems* 5, 4, 517–548.
- SCHWARZHAUPT, J., JENSEN, H. W., AND JAROSZ, W. 2012. Practical Hessian-based error control for irradiance caching. *ACM Trans. Graph.* 31, 6 (Nov.), 193:1–193:10.
- SEN, P., AND DARABI, S. 2012. On filtering the noise from the random parameters in Monte Carlo rendering. *ACM Trans. Graph.* 31, 3 (June), 18:1–18:15.
- SIMONCELLI, E. P., AND OLSHAUSEN, B. A. 2001. Natural image statistics and neural representation. *Annual Review of Neuroscience* 24, 1193–1216.
- STAM, J. 2001. An illumination model for a skin layer bounded by rough surfaces. In *Rendering Techniques 2001*, S. Gortler and K. Myszkowski, Eds., Eurographics. Springer, 39–52.
- TUMBLIN, J., AGRAWAL, A., AND RASKAR, R. 2005. Why I want a gradient camera. In *Proc. CVPR’05*, 103–110.
- VEACH, E., AND GUIBAS, L. J. 1995. Optimally combining sampling techniques for Monte Carlo rendering. In *Proc. ACM SIGGRAPH 95*, 419–428.
- VEACH, E., AND GUIBAS, L. J. 1997. Metropolis light transport. In *Proc. ACM SIGGRAPH 97*, 65–76.
- WARD, G. J., AND HECKBERT, P. 1992. Irradiance gradients. In *Proc. Eurographics Workshop on Rendering ’92*, 85–98.
- WARD, G. J., RUBINSTEIN, F. M., AND CLEAR, R. D. 1988. A ray tracing solution for diffuse interreflection. In *Proc. ACM SIGGRAPH ’88*, 85–92.