

# Gradient Domain Path Tracing

Dario Seyb

January 22, 2017

## Abstract

*In recent years the popularity of the path tracing rendering algorithm has risen dramatically. Especially the movie industry appreciates the accuracy and relative simplicity of physically based rendering. Unfortunately the drawbacks of path tracing are still holding back artistic expression. For complex scenes rendering times can be prohibitive. Hence extensions of path tracing which address these issues have to be found. In this report we will present "Gradient Domain Path Tracing", one of the more recent extensions. We will describe the algorithm, its advantages over classical path tracing and recent work in gradient domain rendering. Additionally we will provide an implementation of the algorithm in the rendering framework "zaphod" [Sey17].*

**Keywords** Path Tracing, Physically Based Rendering, Gradient Domain Light Transport

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>A Brief Introduction to Path Tracing</b>	<b>2</b>
<b>3</b>	<b>Introduction to Gradient Domain Light Transport</b>	<b>3</b>
3.1	Symmetric Gradient Formulation . . . . .	5
<b>4</b>	<b>Shift Mappings</b>	<b>6</b>
4.1	Construction . . . . .	6
4.1.1	Shift via Manifold Exploration . . . . .	7
4.1.2	Improved Shift via Manifold Exploration . . . . .	7
4.1.3	Half Vector Replication . . . . .	7
4.1.4	Discussion of Shift Mappings . . . . .	8
4.2	Computing the Jacobian . . . . .	8
4.3	Multiple Importance Sampling . . . . .	9
<b>5</b>	<b>Solving Screened Poisson Equations</b>	<b>10</b>
<b>6</b>	<b>A Theoretical Analysis of Gradient Domain Light Transport</b>	<b>11</b>
6.1	A Math Primer . . . . .	11
6.1.1	The Discrete Fourier Transform . . . . .	11
6.1.2	Power Spectra . . . . .	12
6.1.3	Dirac Delta Functions . . . . .	12
6.2	Analyzing the Components of Gradient Domain Methods . . . . .	13
6.2.1	The Gradient Operator . . . . .	13
6.2.2	Stochastic Sampling . . . . .	14
6.2.3	Integration . . . . .	15
6.2.4	Poisson Reconstruction . . . . .	15

<b>7 Recent Work</b>	<b>16</b>
7.1 Gradient Domain Bidirectional Path Tracing . . . . .	16
7.1.1 Shift Mapping . . . . .	16
7.1.2 Connection Strategy . . . . .	16
7.2 Temporal Gradient-Domain Path Tracing . . . . .	17
7.2.1 Shift Mapping . . . . .	17
7.2.2 Spatio-temporal Reconstruction . . . . .	17
<b>8 Discussion</b>	<b>18</b>
8.1 Equal Time Comparisons . . . . .	18
8.2 Analysing Shift Mappings . . . . .	18
<b>9 Acknowledgements</b>	<b>20</b>

## 1 Introduction

Gradient domain methods are a rather recent research area in light transport simulation. Especially in the last two years there have been multiple extensions to the original algorithm, applying gradient rendering to existing light transport strategies. This report intends to be a survey of the state of the art in gradient domain rendering. We will start by describing the basic algorithm and the components needed to implement it. Next we will present a theoretical analysis of gradient rendering which provides a mathematical explanation for the empirical results. Finally we will compare the various gradient domain adaptions. We are especially interested in the difference between Gradient Domain Metropolis Light Transport (MLT) and Gradient Domain Bidirectional Path Tracing (BDPT) [Vea98]. We focus on these algorithms because their non-gradient domain versions are popular in research and production. Additionally both MLT and BDPT are unbiased, sampling-based algorithms that handle a large variety of scenes. Even traditionally hard to render scenes are handled robustly. As such, the scenes chosen for analysis should be handled similarly well between algorithms, and the comparison can focus on the influence of gradient rendering on their results.

## 2 A Brief Introduction to Path Tracing

Path tracing is a technique used for light transport simulation which uses Monte Carlo sampling to evaluate the integral in the rendering equation by [Kaj86]

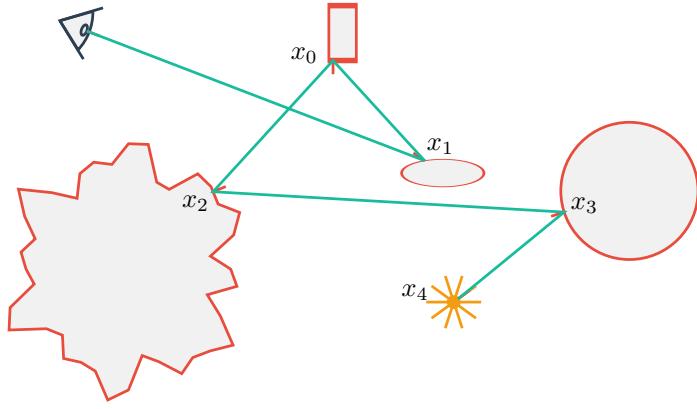
$$I(x, x') = \underbrace{g(x, x')}_{\text{Emission}}[\epsilon(x, x')] + \underbrace{\int_S \rho(x, x', x'') I(x', x'') dx''}_{\substack{\text{BRDF} \\ \text{Incoming light} \\ \text{Light reflected at } x' \text{ in direction } \vec{x}' \vec{x}}} \quad (1)$$

$I(x, x')$  is the radiance arriving from point  $x'$  at  $x$ . It depends on the light that is being reflected at  $x'$  and it is defined recursively. In modern literature the prevalent mathematical framework to describe light transport problems is the path integral formulation introduced by [Vea98].

$$I_j = \int_{\Omega} f_j(\bar{x}) d\mu(\bar{x}) \quad (2)$$

$I_j$  is the amount of radiance arriving at a pixel  $j$  on the sensor and  $f_j$  is the *measurement contribution function* which assigns an amount of transported energy to each path  $\bar{x}$ . The main difference between Equations (1) and (2) is that instead of describing a measurement  $I$  by a local recursive integral we describe it by globally integrating over *Path Space*  $\Omega$ , that is, the set of all paths  $\bar{x} = x_1 x_2 \dots x_n$  that are sampled by a path tracer.

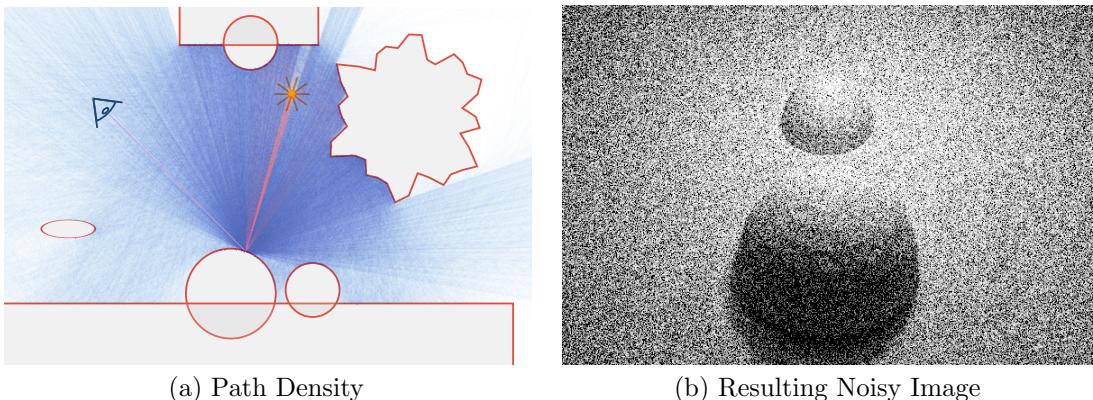
In this context Monte Carlo simulation corresponds to drawing random samples in path space. To compute a pixel value we randomly generate paths that connect a point on the sensor to a point on a light source. This is usually done by recursively tracing a ray through the scene and calculating a new outgoing direction at each intersection point. Once a path is sampled we can calculate its contribution to the image. Figure 1 illustrates this process.



**Figure 1:** Sampling a ray that contributes to a certain pixel on the sensor.

While the sampling process is identical whether we are using the path integral or the recursive scattering formulation, working in path space allows us to develop algorithms that work on complete paths and are decoupled from the path generation strategy.

The main issue with the Monte Carlo simulation approach is that naively sampling the high dimensional path space results in high variance between samples. This is visible in the final image as a characteristic high frequency noise. Because of the high number of samples required for a convergent image, path tracing was prohibitively expensive for many applications. The high variance is in part due to the fact that many paths never hit a light source. Figure 2 a) shows the distribution of light paths in a scene with diffuse materials. Blue paths never hit a light; yellow paths do. As you can see, since the light source is small and out of the way, very few paths hit it randomly. The result is a noisy image as shown in Figure 2 b).<sup>1</sup>



**Figure 2:** A scene illuminated by a small light source rendered with standard path tracing. A high sample count is necessary because very few of the paths contribute to the image.

### 3 Introduction to Gradient Domain Light Transport

The goal of almost any new light transport method is to reduce the variance in the evaluated samples since this leads to faster converging images which need less time to render. While there are many ways to reduce the variance of a given sampler, for example via importance sampling [Vea98], these are often

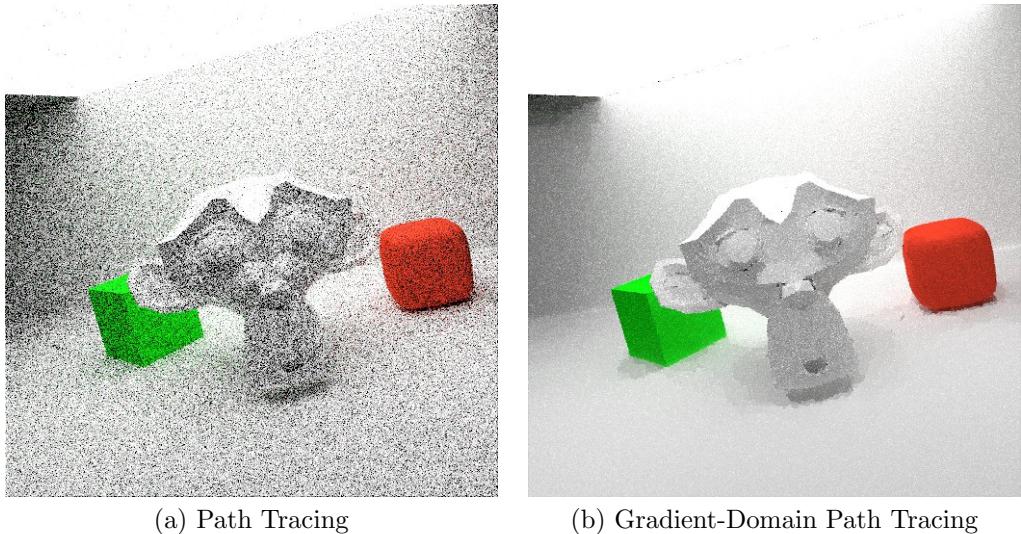
---

<sup>1</sup>The 2D figures in this report are created with a tool we created specifically for this purpose. You can find it at <https://darioseyb.com/pathgraph>. It is inspired by the paper "Theory, Analysis and Applications of 2D Global Illumination" [Jar+12].

quickly exhausted or provide only marginal improvements. Gradient Domain Path Tracing (G-PT) does not change the base sampling distribution at all, instead it relies on the spectral properties of gradients and screened Poisson reconstruction to reduce variance.

In natural images and many images we want to render with a path tracer, the energy is concentrated in the low frequencies [Rud94]. In a traditional path tracer we are sampling the low frequencies directly. Since the energy is the high, variance is high, too. This results in a noisy image. The idea of gradient domain methods is to instead sample the high frequencies, i.e. gradients, directly. These are a lot lower energy and thus can be sampled with less variance. The final image can be reconstructed from the gradients and a (noisy) base image using Screened Poisson Reconstruction as described in Section 5. A detailed analysis of these claims will be provided in Section 6.

The idea of direct gradient sampling was first introduced in the context of MLT [Leh+13]. Since MLT is notoriously difficult to implement correctly, [Ket+15] applied the same concepts to classical path tracing with great success. The resulting algorithm "Gradient Domain Path Tracing" is easy to implement and provides greatly improved results compared to regular path tracing.



**Figure 3:** Comparing the results of standard path tracing (left) and G-PT (right) at the same number of samples (10).

Instead of sampling the value  $I_j$  in Equation (2) directly we want to sample  $\Delta_{ij}$ , the finite difference between two pixels  $i$  and  $j$ .

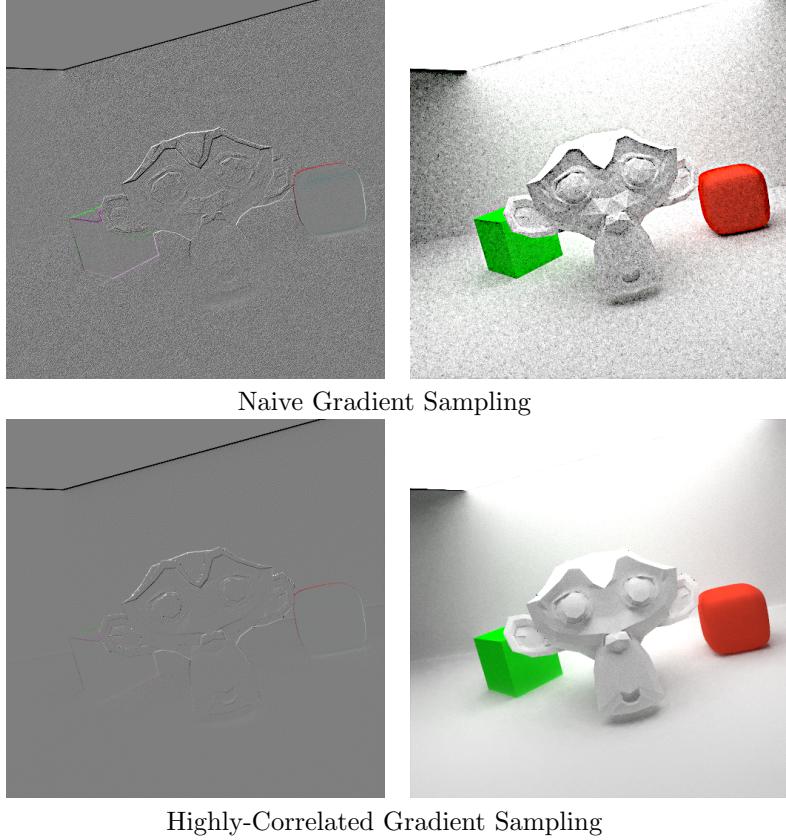
$$\Delta_{ij} = I_i - I_j \quad (3)$$

Naively computing finite differences is not enough to ensure low variance since  $\text{Var}(I_i - I_j) = \text{Var}(I_i) + \text{Var}(I_j)$  if  $I_i$  and  $I_j$  are uncorrelated. Hence we need to find a way to sample  $I_i$  and  $I_j$  in a strongly correlated fashion. Recall that a single measurement of  $I_j$  only depends on the value of the measurement contribution function  $f_j(\bar{x})$ . Thus, if we manage to generate a path  $\bar{x}$  contributing to pixel  $i$  and a path  $\bar{y}$  contributing a similar amount of radiance to pixel  $j$ , the variance of  $\Delta_{ij}$  will be low.

In all the gradient domain methods proposed so far this is achieved by deterministically shifting a *base path*  $\bar{x}$  through pixel  $i$  to an *offset path*  $\bar{y}$  through pixel  $j$ . We call this shift mapping  $T_{ij}$  and write  $T_{ij}(\bar{x}) = \bar{y}$ . With  $T_{ij}$  defined we can rewrite Equation (3) in terms of base and offset paths.

$$\begin{aligned} \Delta_{ij} &= I_i - I_j \\ &= \int_{\Omega} f_i(\bar{x}) d\mu(\bar{x}) - \int_{\Omega} f_j(\bar{x}) d\mu(\bar{x}) \\ &= \int_{\Omega} f_i(\bar{x}) - f_j(T_{ij}(\bar{x})) |T'_{ij}| d\mu(\bar{x}) \end{aligned} \quad (4)$$

The transformation is straightforward except for the introduction of the Jacobian determinant  $|T'_{ij}|$ . This is necessary because we are changing the integration variable of  $f_j$  from  $\bar{x}$  to  $T_{ij}(\bar{x})$ . The integration domain is not adjusted because we assume  $T$  to be bijective on  $\Omega$  and thus  $T(\Omega) = \Omega$ . Thanks to the



**Figure 4:** Comparing naive gradients with carefully sampled gradients. Both images are rendered with 100 SPP. The images on the right are the results of Poisson reconstruction based on the gradient images on the left.

formulation of the shift as a change in integration variables we have a lot of freedom in constructing a shift mapping. As long as we can compute the corresponding Jacobian the results will be unbiased.

### 3.1 Symmetric Gradient Formulation

Unfortunately it is impractical to require a shift mapping to be a bijection on the complete path space. [Man+14] introduce a *Symmetric Gradient* formulation which does not pose this requirement on  $T$ . Let us first investigate reasons why a shift may not be bijective. The first case is that there might be paths  $\bar{x}$  for which  $T(\bar{x})$  is not defined. This can happen when the shift procedure fails, e.g. for numerical reasons. We call these shifts *non-invertible*. The second case is that  $T(\bar{x})$  might map  $\bar{x}$  to a path  $\bar{y} \notin \Omega$ , which means the offset path  $\bar{y}$  would never be generated during regular path sampling. These shifts are called *non-symmetric*. Thirdly, the first two cases might apply to the inverse shift  $T^{-1}$ . Bijectivity requires  $T^{-1}(T(\bar{x})) = \bar{x}$ , but if the inverse shift is not well defined this requirement is violated. In the ideal case all shifts would be symmetric (a non-invertible shift can not be symmetric, hence shifts which are non-invertible as well as symmetric do not exist). If any of these cases apply, the integral in Equation (4) will not return the correct value and as a result the final image will be biased.

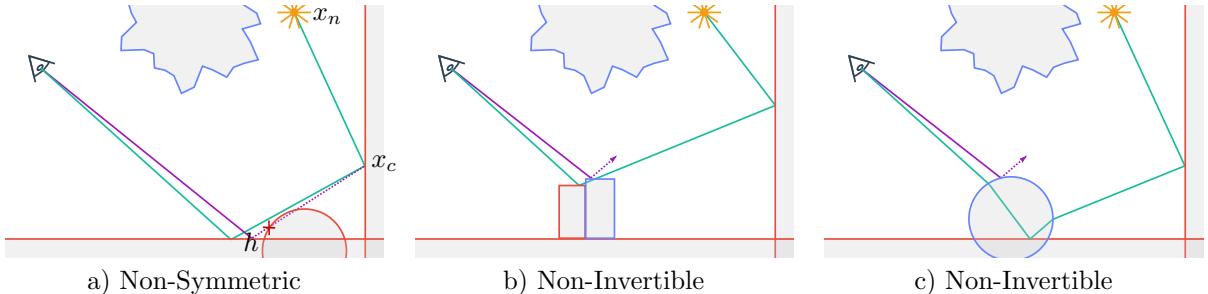
[Man+14] formulate a different way to express  $\Delta_{ij}$  which can handle cases of bijectivity violation robustly. This formulation is extended by [Ket+15] to allow for multiple importance sampling. Instead of just sampling the forward finite difference between pixels  $i$  and  $j$  we also sample the *backwards* finite difference and combine them using weights.

$$\Delta_{ij} = \overbrace{\int_{\Omega} w_{ij}(\bar{x})(f_i(\bar{x}) - f_j(T_{ij}(\bar{x}))|T'_{ij}|)d\mu(\bar{x})}^{\text{Forward Difference}} + \overbrace{\int_{\Omega} w_{ji}(\bar{x})(f_j(\bar{x}) - f_i(T_{ji}(\bar{x}))|T'_{ji}|)d\mu(\bar{x})}^{\text{Backward Difference}} \quad (5)$$

Non-symmetric and non-invertible shifts are handled via the weights  $w_{ij}$  and  $w_{ji}$ . In both cases the difference in the other direction will not be sampled and we set the weight of the current sampling

strategy to 1. If the shift is symmetric the difference will be sampled twice, once via forward and once via backward differences, so we weight it by 1/2.

Since the definition of bijectivity violations is fairly abstract we want to give practical examples for each type of shift. Figure 5 shows the three ways shift computations can proceed.



**Figure 5:** Shown are three examples for how a shift can fail. The shift in a) is non-symmetric because the connection to vertex  $x_c$  is blocked and a blocked path will never be sampled by the path tracer. In Figure b) the offset path hits a material with a different classification, and hence the shift is non-invertible. Figure c) shows another case of non-invertibility. The primary ray gets refracted, but the offset path encounters total reflection.

## 4 Shift Mappings

The shift mapping, that is the way we compute offset paths, is at the heart of G-PT. A good shift mapping produces offset paths which have a similar throughput to the corresponding base path. This is not always easy. Even small changes in the position of a path vertex can completely change the throughput of the path if specular materials are involved. The first two vertices of an offset path do not depend on the mapping. The first vertex is fixed on the point on the sensor that corresponds to the pixel shift from  $i$  to  $j$ . The second vertex is the primary scene intersection of the ray defined by the camera position and the first vertex. In this section well will take a look how to continue the offset path after the primary intersection. In all so far proposed shift mappings the aim is to connect the offset path back to the base path as soon as possible. This technique has proved successful, since if two paths share segments their throughputs are likely to be similar. The points at which we connect the paths have to be chosen carefully.

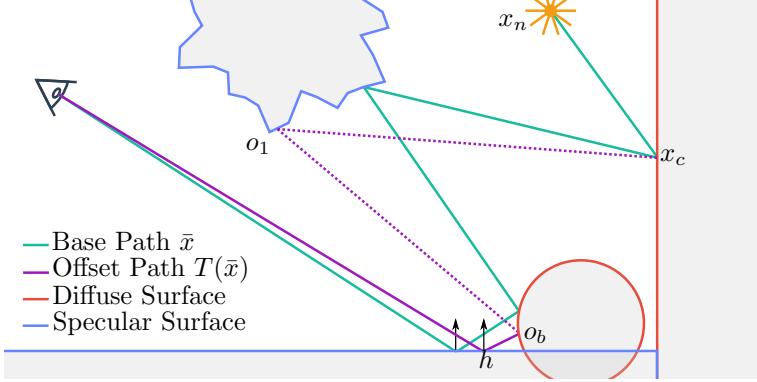
### 4.1 Construction

To construct an efficient shift mapping we need to classify vertices into *diffuse* and *specular*. Diffuse vertices are easier to handle and allow more flexibility in how to connect paths. This is because their BRDF is smooth and the directions to the next and previous path vertices don't change the throughput much. Specular vertices need to be treated with more care. Assume a vertex lies on the surface of a perfect mirror. Moving either of the adjacent vertices on the path will completely nullify any throughput through the whole path since the BRDF is only non zero for a very specific direction. To describe the various shift mappings we introduce a general description of an offset path.

$$\begin{aligned} T_{ij}(\bar{x}) &= \bar{y} = (s, h, o_0, \dots, o_m, x_c, \dots, x_n) \\ s &= \text{shift}_{ij}(x_0) \\ h &= R(s, \omega_s) \end{aligned} \tag{6}$$

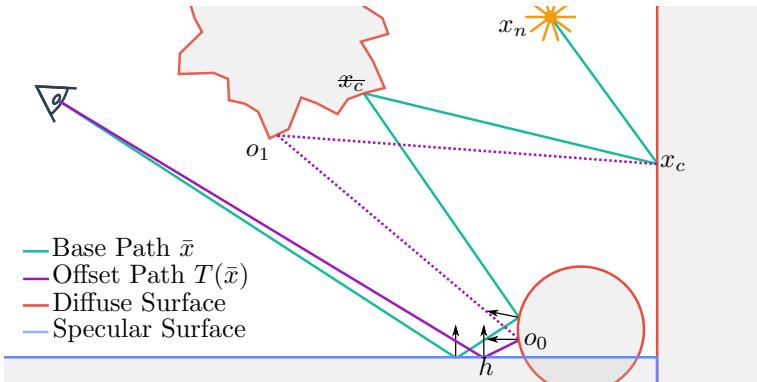
where  $\text{shift}_{ij}(x_0)$  denotes shifting the vertex  $x_0$  in primary sample space from pixel  $i$  to pixel  $j$ ,  $R$  is the ray casting function and  $R(s, \omega_s)$  the primary hit of the ray starting at position  $s$  in direction  $\omega_s$  which is determined by the orientation of the sensor.  $x_c$  is the connection vertex. After connection, the vertices of  $\bar{x}$  and  $\bar{y}$  coincide. In this framework the shift mapping decides the length  $m$  of the offset vertex chain  $\bar{o}$  by picking a connection vertex  $x_c$  and chooses the vertices in  $\bar{o}$  such that the change in throughput is minimal.

**4.1.1 Shift via Manifold Exploration** The first shift mapping, introduced in [Leh+13], uses techniques from *Manifold Exploration* [JM12] to construct parts of  $\bar{o}$  (Figure 6). This is a natural choice since Manifold Exploration was developed in the context of MLT and solves the problem of connecting two diffuse vertices via a chain of specular vertices. Here  $\bar{o}$  is split into three parts.  $x_c$  is chosen as the second diffuse vertex after  $h$ . A specular chain is traced from  $h$  (like in standard path tracing) until a diffuse vertex  $o_b$  is hit. A second specular chain connecting  $o_b$  and  $x_c$  is then constructed using the optimization technique from [JM12]. Note that this second chain can be empty if  $x_c$  follows directly after  $o_b$ . In this case they are immediately connected and no Manifold Exploration is performed.



**Figure 6:** Shifting a path using manifold exploration. Since the primary hit  $h$  on the offset path lies on a diffuse surface we continue the path like in standard path tracing. The next hit  $o_0$  lies on a diffuse surface. We connect the offset path to the base path at the next diffuse base path vertex  $x_c$  via manifold exploration.

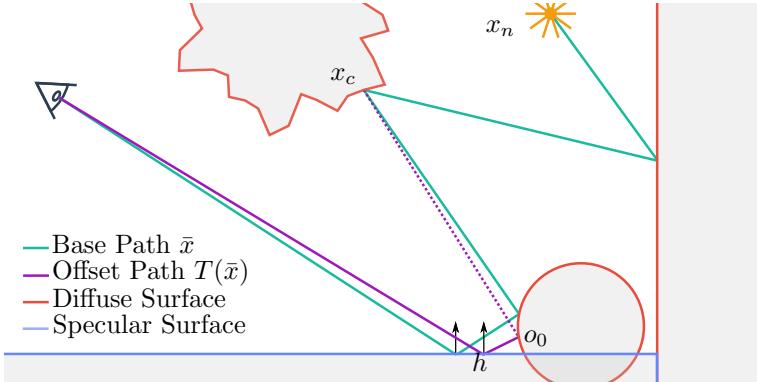
**4.1.2 Improved Shift via Manifold Exploration** [Man+14] contribute a new shift mapping which reduces variance in the case where the second specular chain of the shift by [Leh+13] is empty (Figure 7). This can be an issue because the direct connection can result in a low geometry term. If the second chain is empty the improved shift treats  $x_c$  as specular and finds the next diffuse vertex in  $\bar{x}$ . It then uses the found vertex as the new connection vertex and performs Manifold exploration for connection as before. This completely eliminates the case of empty second specular chains.



**Figure 7:** Shifting a path using the improved manifold exploration shift by [Man+14]. The first hit  $h$  is on a specular surface and we copy the half vector of the corresponding base path vertex. The second vertex  $o_0$  is on a diffuse surface and we could connect to the next diffuse vertex on the base path. Since this would leave the second specular chain empty we use manifold exploration to construct a path connecting to  $x_c$ .

**4.1.3 Half Vector Replication** Finally, [Ket+15] introduce a shift not based on Manifold Exploration.  $x_c$  is chosen as the first vertex in  $\bar{x}$  where both  $x_c$  as well as  $x_{c+1}$  are classified as diffuse (Figure

8). This eliminates the need for connecting diffuse vertices through specular chains. For the vertices in between  $h$  and  $x_c$  a path is traced by replicating the projected half vectors of the base path vertices at each step. Replicating half vectors of specular vertices proves to result in similar values for the BRDF at each vertex, since specular BRDFs are usually well defined by the half vector between incoming and outgoing directions.



**Figure 8:** Shifting a path using Half Vector Replication. At the first hit  $h$  we need to replicate the half vector of the corresponding base path vertex because the surface is specular. As a result we hit  $o_0$  which is on a diffuse surface. The next base path vertex  $x_c$  is also on a diffuse surface and we can connect the offset path back to the base path.

**4.1.4 Discussion of Shift Mappings** The shift mappings based on Manifold Exploration produce excellent results, but are hard to implement and require a costly optimization process. This is why [Ket+15] prefer their novel shift mapping. It results in a slightly higher amount of artifacts caused by singularities that [Man+14] avoids, but those are canceled by using the multiple importance sampling techniques also introduced in [Ket+15]. Overall [Ket+15] is the most promising. It requires about 30 lines of C++ code to implement correctly (including Jacobian calculation) and produces great results at little cost.

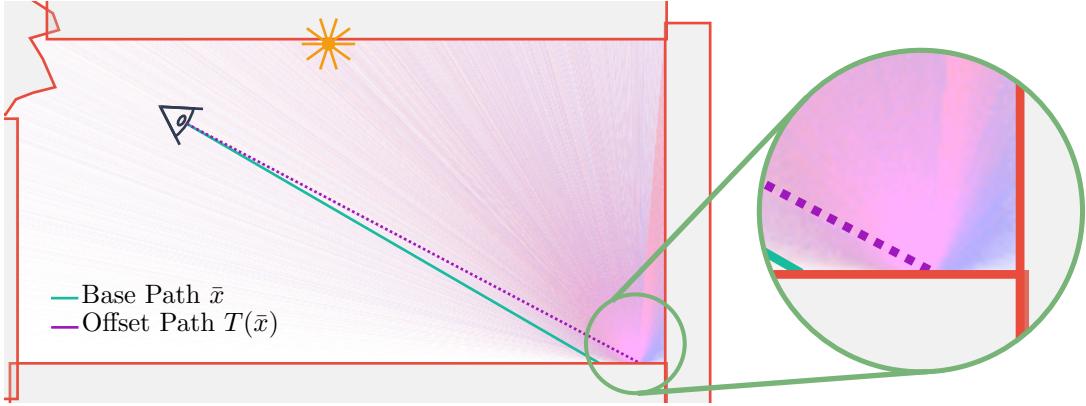
## 4.2 Computing the Jacobian

To account for the change in integration variables in Equation (4) we need to compute the determinant of the Jacobian  $|T'_{ij}(\bar{x})| = |dT_{ij}(\bar{x})/d\bar{x}|$  for the used shift mapping. This is important to make sure that the results of the integration are mathematically correct, otherwise we would introduce bias. In this section we aim to give an intuition for what this value denotes, but we will not derive the Jacobian for any of the presented shift mappings. The value of  $|T'_{ij}(\bar{x})|$  gives us a measure for how much  $T$  stretches or squishes the path space around  $\bar{x}$ . Figure 9 shows how path density can be a lot lower after shifting, resulting in high Jacobian values.

Computing this value can be challenging. In [Ket+15] the derivations are made easier by two insights. First, the authors show that the calculation can be split into three steps. Namely, reparametrizing the path in a way so the shift itself has a trivial to compute Jacobian, computing the Jacobian of the shifted path in the reparameterized space and transforming the path back into the original space. Mathematically this is expressed as multiplying the Jacobian determinants of the three steps.

$$|T'_{ij}(\bar{x})| = \left| \frac{d\bar{y}}{d\bar{x}} \right| = \underbrace{\left| \frac{d\hat{y}}{d\hat{x}} \right|}_{\text{Trivial Shift}} \underbrace{\left| \frac{d\hat{y}}{d\hat{x}} \right|}_{\text{Reparametrization}} \underbrace{\left| \frac{d\hat{x}}{d\bar{x}} \right|}_{\text{Back Transform}} \quad (7)$$

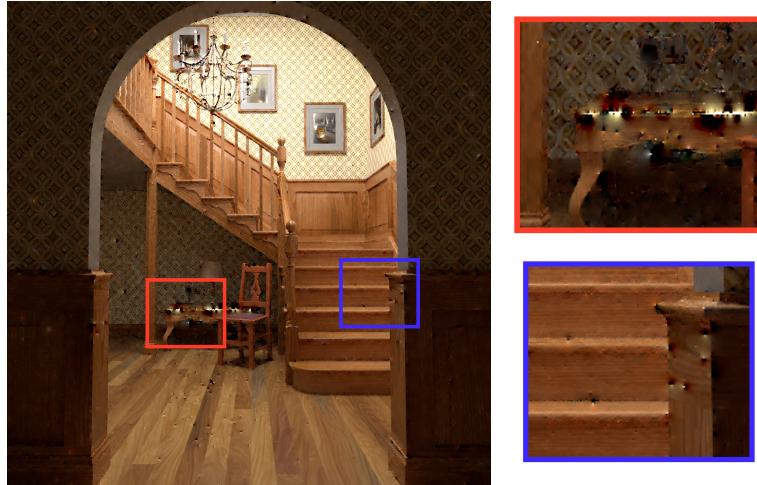
The second insight is that computing the reparametrization at each path vertex separately and multiplying them together results in the final Jacobian for the whole shift. This allows us to combine ideas from different shift mappings and reuse the derived Jacobians.



**Figure 9:** Visualizing the path density changes caused by  $T_{ij}$ . Blue paths are sampled via regular path tracing, red paths via the strategy by [Ket+15]. In the bottom right corner the density of red paths is much lower than the density of blue paths, leading to high Jacobian values.

### 4.3 Multiple Importance Sampling

As shown in the previous section the Jacobian values can get very large. These high values can be an issue, because they cause outliers in the gradients which manifest themselves after reconstruction as very obvious artifacts. As shown in Figure 10 they mostly appear in corners and are rather unpleasant as far as rendering artifacts go. The typical way to deal with outliers like these is to employ multiple importance sampling ([Vea98]). Due to the *Symmetric Gradient Formulation* we already weight the forward and backward gradient sampling strategies. If we extend these weights to be continuous we can weight the strategies by their importance in the symmetric shift case, where they would otherwise be 1/2 for both strategies.



**Figure 10:** Even with multiple importance sampling, artifacts appear in corners and on edges at low sample counts.

To fit the difference sampling into the framework needed for importance sampling, we interpret the base path  $\bar{x}$  and the inverse shift  $T_{ji}(\bar{y}) = \bar{x}$  as two strategies to sample the same distribution. The weights are then computed as follows when using the balance heuristic.

$$w_{ij}(\bar{x}) = \frac{p(\bar{x})}{p(\bar{x}) + p(T_{ij}(\bar{x}))|T'_{ij}(\bar{x})|} \quad (8)$$

The intuition here is that if the Jacobian  $|T'_{ij}(\bar{x})|$  is large the contribution of the sample will be weighted by approximately  $1/|T'_{ij}(\bar{x})|$ . Hence outliers will be suppressed.

## 5 Solving Screened Poisson Equations

The outputs of a gradient domain path tracer are a primal image  $I^g$  and a set of gradient images  $I^{d*}$ . With G-PT the gradient images are the spatial differences  $I^{dx}$  and  $I^{dy}$ . We want to reconstruct the final image  $I$  from these values. That is, we want to estimate the image function  $I$  from its gradients. It is easy to see that  $I$  is optimal if it minimizes the constraint in Equation (9), because that means that the gradient  $\Delta I$  is as close as possible to our computed gradient images. [Bha+08]

$$\int \int \|\Delta I - \begin{pmatrix} I^{dx} \\ I^{dy} \end{pmatrix}\|^2 dx dy \quad (9)$$

Reconstructing an image based on this constraint introduces issues if there is error in the gradients. Especially low frequencies are not reconstructed well, because small gradient error accumulates over longer distances. To improve the reconstruction we use the primal image  $I^g$ . The primal image is noisy in the high frequencies, but represents low frequencies fairly well, even at low sample counts. [Bha+08] add an additional term to the integrand in Equation (9).

$$\int \int \alpha(I - I^g)^2 + \|\Delta I - \begin{pmatrix} I^{dx} \\ I^{dy} \end{pmatrix}\|^2 dx dy \quad (10)$$

Here  $\alpha$  is a factor that controls the influence of the primal image on the solution. Note that setting  $\alpha$  to 0 reduces Equation (10) to Equation (9). Expressing the final image as in Equation (11) allows us to use standard numerical solvers.

$$I = \operatorname{argmin}_{\bar{I}} (\|\alpha(\bar{I} - I^g)\|^p + \|\Delta \bar{I} - \begin{pmatrix} I^{dx} \\ I^{dy} \end{pmatrix}\|^p) \quad (11)$$

The variable  $p$  allows us to change the norm used for minimization. Only the L2 norm results in bias-free images, but [Ket+15] illustrate that using the L1 norm, while introducing bias, results in images with less artifacts and converges to the same results when the sample count is high. Artifacts in L2 reconstructions stem from the outliers in the gradient sampling result discussed in the previous section. The reconstruction spreads out the error to minimize it. L1 reconstruction does not spread out error as much and as such outliers generally only influence one pixel and manifest as fireflies which are a common artifact in path tracing and hence do not stand out as much. While multiple importance sampling reduces the gradient outliers and makes L2 reconstruction more practical, L1 reconstruction still produces cleaner images.



**Figure 11:** Reconstruction using different norms. The base gradient and primal images are exactly the same.

## 6 A Theoretical Analysis of Gradient Domain Light Transport

Early papers on gradient domain methods provided little in the way of rigorous mathematical analysis. [Leh+13] discuss the frequency characteristics of the gradient operator and Poisson reconstruction, but do not go into any further depth. Particularly the factor  $\alpha$  in Equation (11) is chosen empirically. The goal of analysing the frequency characteristics of light transport is to gain a deeper insight into where error is coming from and why gradient methods are successful. In the following we will introduce commonly used techniques and motivate gradient based light transport by analyzing the frequency content of natural images and their gradients.

### 6.1 A Math Primer

Many of the techniques used in signal processing and image analysis are not commonly taught in undergraduate courses. For the benefit of the reader we will give a short overview over the mathematical concepts and notation used in this section.

**6.1.1 The Discrete Fourier Transform** Our aim is to analyze the frequency content of a discretely sampled signal  $f$  (e.g an image). For now we can assume  $f$  to be a continuous function defined over the real numbers. The frequency content of a continuous function, also called its *Fourier spectrum* can be found by *projecting* it onto a basis of complex exponential functions  $E_\omega(x) = e^{i2\pi\omega x}$ . This makes sense if you recall that a complex exponential function can be decomposed into *sine* and *cosine* as follows  $e^{i2\pi\omega x} = \cos(2\pi\omega x) + i\sin(2\pi\omega x)$ . Hence, once we can express a function  $f$  in terms of a set of periodic parts  $E_\omega$  and thus know the factor for each frequency  $\omega$  we know how much that frequency influences the function  $f$ . The forward and backwards Fourier transforms are given by [SSJ16] as follows.<sup>2</sup>

$$F(\omega) = \int_{-\infty}^{\infty} f(x)e^{-i2\pi\omega x} dx \quad (12)$$

$$f(x) = \int_{-\infty}^{\infty} F(\omega)e^{i2\pi\omega x} d\omega \quad (13)$$

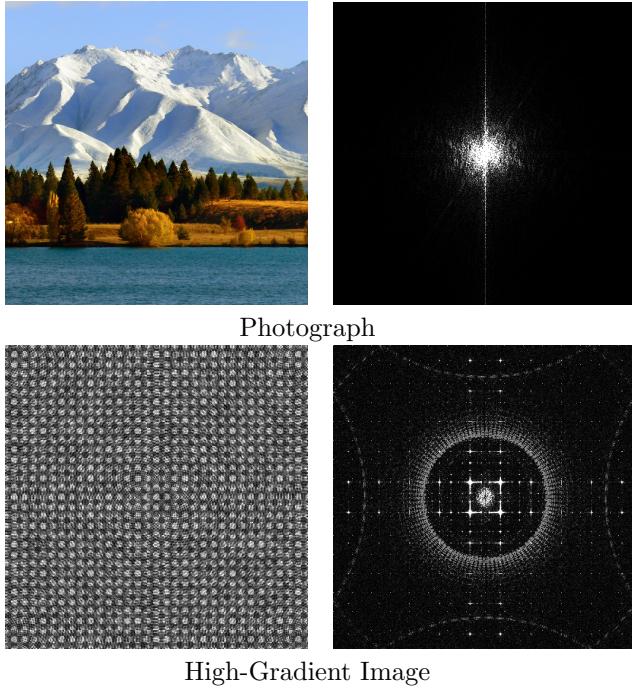
$F(\omega)$  is the factor for the frequency  $\omega$  we are looking for. Equation (13) shows how to reconstruct a function from its periodic parts. This definition only works with continuous functions, but not with a set of discrete samples which we are typically trying to analyze. To deal with this we need two further facts about the Fourier transform. First, if  $f$  is periodic, its Fourier spectrum is non-zero only at discrete points. Secondly, if  $f$  is discrete, its Fourier spectrum will be periodic [SSJ16]. Generally we sample  $f$  over a finite range and can assume that the sampled values repeat outside that range. Hence we can assume  $f$  to be periodic and thus  $f$  has a discrete Fourier spectrum. Also, as already said,  $f$  is sampled at discrete points and hence its Fourier spectrum is periodic. Combining these two properties we can see that we can represent the Fourier spectrum of a discrete, periodic function with a finite amount of values (since it itself is discrete and periodic).

In practice we need exactly as many samples of the Fourier spectrum as we have samples of the original function. From those spectrum samples the original samples can be reconstructed perfectly.

Another important concept is that a multiplication  $f \times g$  in the spatial domain corresponds to a convolution  $F * G$  in the frequency domain. Correspondingly, a convolution in the spatial domain corresponds to a multiplication in the frequency domain. This is easy to prove using the definitions of the Fourier transform and its inverse.

Furthermore, evaluating  $F(\omega)$  at 0 equals the integral of  $f$  over its complete domain as shown in Equation (14).

$$\begin{aligned} F(0) &= \int_{-\infty}^{\infty} f(x)e^{-i2\pi w \times 0} dx \\ &= \int_{-\infty}^{\infty} f(x)dx \end{aligned} \quad (14)$$



**Figure 12:** Left: Images with different frequency content, Right: Their power spectra. Note the vertical spike in the power spectrum of the photograph, indicating the existence of higher frequencies in the y direction of the image. This corresponds to the clear separation between sky, mountains, trees and the lake in the image.

**6.1.2 Power Spectra** A *power spectrum* is the square of a Fourier transform  $|F(\omega)|^2$ . It is often used to represent the results of analysis since it presents the frequency content  $f$  in a very visual way.

Figure 12 shows a natural image and an image constructed to have high gradients on the left and their power spectra on the right. In a power spectrum the values in the centre of the image correspond to low frequencies. As you can see, the power of the natural image is focused in the low frequencies while the high-gradient image has a spread out power spectrum.

**6.1.3 Dirac Delta Functions** A common problem in light transport is representing and analyzing functions which are 0 everywhere except for one specific point. This comes up for example in the BRDF of a perfectly specular material. It reflects light into only one very specific direction. Unfortunately it is hard to work with such a function since they are not integrable via usual methods. To solve this we introduce the *Dirac Delta Function*  $\delta(x)$ . "Function" is in this case a bit of a misnomer, because it is technically not a function, but a limit of a series of functions defined as follows. [Bra00]

$$g(x, t) = \frac{1}{\sqrt{2\pi t}} e^{-x^2/2t}$$

$$\delta(x) = \lim_{t \rightarrow 0} g(x, t) \quad (15)$$

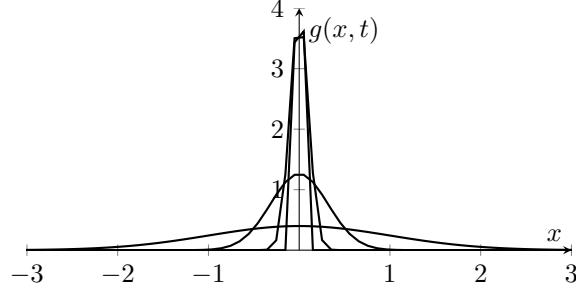
$g(x, t)$  is a Gaussian centered around the origin, which gets sharper as  $t$  decreases. Figure 13 shows how  $g$  approaches to have one of the properties we want, namely, vanishing everywhere except for a shrinking footprint around 0. We call  $\delta$  a function, because we can work with it like it is one in the context of integration. Specifically, multiplying it with a function  $f$  inside of an integral, like in Equation (1), results in the following:

$$\langle \delta(x), f(x) \rangle = \lim_{t \rightarrow 0} \int_{-\infty}^{\infty} g(x, t) f(x) dx = f(0) \quad (16)$$

---

<sup>2</sup>From here on out a Fourier transform will always be denoted by a capital letter.

An important property for the next section is its Fourier transform, given by  $\Delta(\omega) = 1$ . This follows trivially from the definition of the Fourier transform in Equation (12) and the identity in Equation (16).

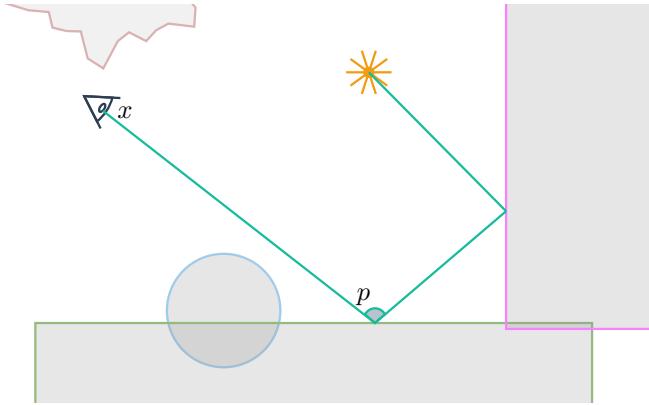


**Figure 13:** Plotting  $g$  for multiple values of  $t$ . Note how it approaches a spike at 0.

## 6.2 Analyzing the Components of Gradient Domain Methods

This section is a summary of the analysis performed in [Ket+15]. At the heart of the analysis is the result that the efficiency of gradient methods depends on the power spectrum of the image function. Images with sparse gradients (which means power is concentrated in the low frequencies) benefit the most from gradient rendering. Luckily this is a property that most natural images have [Rud94]. Since we mostly want to depict natural scenes, this is also the case for the subjects of light transport simulation. Note that natural in this context doesn't refer to "scenes in nature", but to scenes that depict reality in some (even stylized) way.

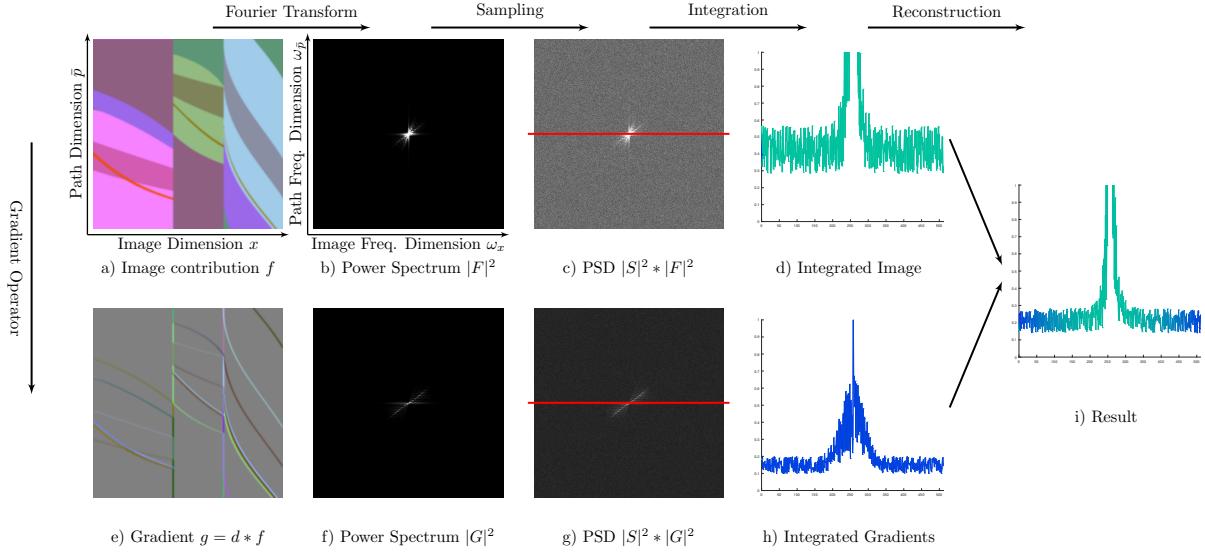
The analysis is done in parts and makes assumptions and simplifications to reduce its verbosity. The main simplification is that we are only investigating two dimensional scenes and hence one dimensional sensors. We also limit ourselves to one bounce. That way path space can be represented in two dimensions. The primary sample space dimension  $x$  and the secondary sample space at the first bounds  $p$ . Figure 14 shows the 2D scene we are using for the analysis.



**Figure 14:** A two dimensional scene with a camera and a path starting at the camera, bouncing once and then being connected to the light.

Figure 15 is meant as a visual guide through this analysis. Sections advance from left to right. The top row shows the analysis performed on the primal image, the bottom row on the gradient image. We advice to come back to the figure after each section.

**6.2.1 The Gradient Operator** First we investigate the power spectrum of the gradient image  $G$ . We use a simplified shift mapping  $T(x, \bar{p}) = (x + 1, \bar{p})$  which only changes the primary sample space location of the path  $(x, \bar{p})$ . The gradient function is then defined as  $g(x, \bar{p}) = f(x, \bar{p}) - f(x + 1, \bar{p})$ . For convenience we can express  $g$  as a convolution  $g(x, \bar{p}) = (d * f)(x, \bar{p})$  between the finite difference operator  $d(x) = \delta(x) - \delta(x + 1)$  and  $f$  where  $\delta$  is the Dirac delta function. As noted in section 6.1.1 this



**Figure 15:** This figure shows the steps of the analysis. The top and the bottom row show the same transformations applied to the primal image  $f$  and the gradient image  $g$  respectively. On the left (a) and e)) we show the reference images with no sampling error. The next column shows the respective Fourier transforms. Note that that b) contains a lot more power than f). This is the result of Equation (17) and the fact that all of the power of  $F$  is concentrated in the low frequencies. Next we show the power spectral densities of  $f$  and  $g$  after sampling. The result from Equation (18) is visually represented by the lighter constant background of c). This is because sampling adds noise to all frequencies equally and the power of the added noise is proportional to the power of the sampled function. Next we integrated of the path dimension, which corresponds to slicing along the red line in c) and g). The results of the integration are shown in d) and h). Not the considerably higher base noise in d). Poisson reconstruction corresponds to combining the results in d) and h) while low-pass filtering d) and high-pass filtering h). Figure i) shows the result of the reconstruction in the frequency domain. It retains the low level of noise of h) while representing low frequencies accurately.

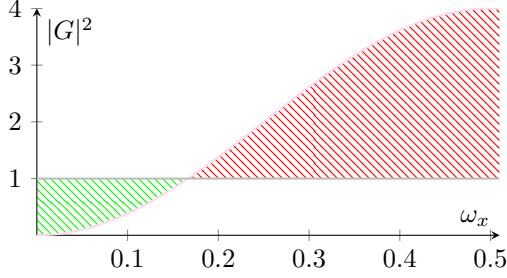
convolution corresponds to a multiplication in the Fourier domain and hence the Fourier transform of  $g$  is  $G(\omega_x, \omega_{\bar{p}}) = D(\omega_x, \omega_{\bar{p}})F(\omega_x, \omega_{\bar{p}})$ . We can not give  $F$  directly since it depends on the scene, but we can investigate  $d$ . Its Fourier transform is given by  $D(\omega_x, \omega_{\bar{p}}) = \delta(\omega_x, \omega_{\bar{p}}) - \delta(\omega_{x+1}, \omega_{\bar{p}}) = 1 - e^{i2\pi\omega_x}$ . This follows trivially from the identities given in section 6.1.3. An important property is the power spectrum of  $D$ ,  $|D|^2 = |1 - e^{i2\pi\omega_x}|^2 = 2 - 2\cos(2\pi\omega_x)$ <sup>3</sup>. In Equation (17) we summarize the spectral properties of the gradient function.

$$\begin{aligned} G(\omega_x, \omega_{\bar{p}}) &= (1 - e^{i2\pi\omega_x})F(\omega_x, \omega_{\bar{p}}) \\ |G(\omega_x, \omega_{\bar{p}})|^2 &= (2 - 2\cos(2\pi\omega_x))|F(\omega_x, \omega_{\bar{p}})|^2 \end{aligned} \quad (17)$$

Especially the second result is interesting. Figure 16 shows how the finite difference operator attenuates low frequencies  $\omega_x < \frac{1}{6}$ .

**6.2.2 Stochastic Sampling** The complete analysis of stochastic sampling is fairly lengthy and would be out of scope for this report. We limit ourselves to presenting the assumptions and results. For a more detailed description we refer to the supplemental materials of [Ket+15]. The sampling process is modeled as a multiplication  $s \times f$  where  $s$  is the sampling function. Its *power spectral density* ([MC12]) is given by Equation (18). It can be interpreted as the expected value of a power spectrum.

<sup>3</sup>This could also be simplified to  $4\sin^2(\pi\omega_x)$ . This is not done in the original paper and we keep it to avoid confusion.



**Figure 16:** The power spectrum of  $G$  assuming  $F$  is constant. Areas shaded green correspond to a attenuation of power in the corresponding frequencies. While the power of frequencies approaching the Nyquist limit is amplified significantly, most images do not contain these frequencies as discussed in section 6.1.2.

$$(|S|^2 * |F|^2)(\omega_x, \omega_{\bar{p}}) = n^2 |F|^2(\omega_x, \omega_{\bar{p}}) + \underbrace{n ||F||^2}_{\text{Frequency independent term}} \quad (18)$$

where  $n$  is the number samples drawn. The important result is that sampling adds power to all frequencies equally and the added power is proportional to the energy of the signal.

**6.2.3 Integration** Finally we need to compute the power spectrum of the Monte Carlo integration. We integrate  $f$  over all paths  $\bar{p} \in \Omega$  to obtain a pixel value  $I$  as in Equation (2). The Fourier transform of that integral corresponds to slicing the Fourier transform of  $f$  at  $w_{\bar{p}} = 0$  (see Equation (14)). Since we are taking  $n$  discrete samples and simply adding them up we need to multiply the output of the sampling process by  $1/n$  (and the power spectrum by  $1/n^2$ ).<sup>4</sup>

$$\frac{1}{n^2} (|F|^2 * |S|^2)(\omega_x, 0) = |F(\omega_x, 0)|^2 + \frac{1}{n} ||F||^2 \quad (19)$$

Note that this is just the power spectrum of the converged image with an additional error term  $|\epsilon_F(\omega_x)|^2 = ||F||^2/n$  which gets smaller when the number of samples increases. It is *proportional to the energy of the image and constant over all frequencies*.

Having derived the error term  $|\epsilon_F(\omega_x)|^2$  for the image, we can similarly give the error term for gradient sampling as  $|\epsilon_G(\omega_x)|^2 = ||G||^2/n$ . We see in Equation (20) that the relative error between gradient and direct sampling only depends on the relative power of the gradient image versus the primal image.

$$\frac{|\epsilon_G(\omega_x)|^2}{|\epsilon_F(\omega_x)|^2} = \frac{||G||^2}{||F||^2} \quad (20)$$

Recalling the result from section 6.2.1 this means that, since the energy of the gradient image will be lower than the energy of the primal image for most natural images, the error introduced by sampling will be lower as well. This is the first important result of the analysis.

**6.2.4 Poisson Reconstruction** As a last step the analysis turns to the screened Poisson reconstruction. Again, we will only present results since the derivations are fairly lengthy. First we find the error term (analogous to Equation (20)) for the image error after reconstruction. This is done by transforming the Poisson reconstruction problem into the Fourier domain [Bha+08] and plugging in the definitions of  $F$  and  $G$ .

$$|\epsilon_{R_\alpha}|^2 = \frac{\alpha^4 |\epsilon_F|^2 + |D|^2 |\epsilon_G|^2}{(\alpha^2 + |D|^2)^2} \quad (21)$$

Here  $\alpha$  is the same values as in Equation (11) and sets the weight of the primal image during the reconstruction. Next we derive an optimal value  $\alpha_*$  for the factor  $\alpha$  in Equation (21).

---

<sup>4</sup>We omit the pixel filter here since it is non essential to the result of the analysis.

$$\alpha_*^2 = \frac{\|G\|^2}{\|F\|^2} \quad (22)$$

In practice the optimal value is unknown since we need the converged images to calculate it, but a value of 0.2 seems to be good for many different scenes and is used for all the reconstructions in this report. Substituting  $\alpha_*$  for  $\alpha$  in Equation (21) gives us an expression returning the minimal error after reconstruction for a given image.

$$|\epsilon_{R_{\alpha_*}}|^2 = \frac{|\epsilon_G|^2 |\epsilon_F|^2}{|\epsilon_F|^2 |D|^2 + |\epsilon_G|^2} \quad (23)$$

For comparison purposes we'd like to know the factor by which gradient sampling reduces the error versus direct sampling. For this we rearrange Equation (23) and use the result in Equation (20) to simplify, giving us:

$$r = \frac{|\epsilon_F(\omega_x)|^2}{|\epsilon_{R_{\alpha_*}}(\omega_x)|^2} = \frac{\|G\|^2/\|F\|^2 + 2 - 2\cos(2\pi\omega_x)}{\|G\|^2/\|F\|^2} \quad (24)$$

This is the central result of the analysis. If  $r$  has a high value the error is reduced by a large margin. Reconstruction effectively combines the high frequencies of the gradient images with the low frequencies of the primal images. In the process it filters the low frequency error from gradients and the high frequency error in the primal image and thus reduces the overall error.

## 7 Recent Work

Because of the promising results provided by G-PT there has been additional research conducted in this area of light transport simulation. A focus has been extending G-PT to more robust algorithms than path tracing and exploiting temporal coherence in animated image sequences.

### 7.1 Gradient Domain Bidirectional Path Tracing

While G-PT does considerably better than traditional path tracing in most scenes it is still subject to the same limitations path tracing suffers from. Rendering scenes with difficult light transport characteristics like the one in Figure 17 still results in high variance, even at high sample counts. To address these issues [Man+15] introduces Gradient Domain Bidirectional Path Tracing (G-BDPT). G-BDPT is the natural extension of the Bidirectional Path Tracing Algorithm by [Vea98] into the gradient domain.

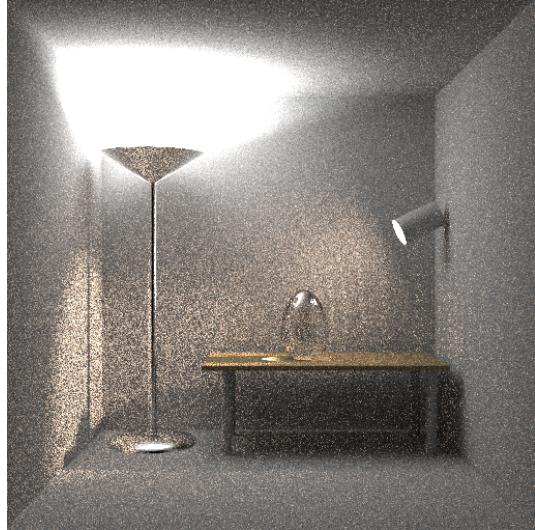
The main contributions are an adaption of the shift mapping by [Man+14] to the bidirectional path construction algorithm, an efficient BDPT connection strategy and a multiple importance sampling technique which reduces gradient sampling artifacts drastically.

**7.1.1 Shift Mapping** The shift mapping used in [Man+15] is identical to the one proposed by [Man+14]. It is similar to the G-PT shift described earlier in this paper. The main difference is that instead of requiring two consecutive diffuse vertices to connect to the base path this shift always connects at the third diffuse vertex. The number of specular vertices between the second and the third diffuse vertex is not considered. Hence the variance of the throughput of the resulting shift paths is potentially higher, but the shift mapping retains properties that can be exploited to make gradient sampling drastically more efficient in the context of BDPT. We are basically trading higher variance for more samples which in this case proves to result in lower error at equal render time.

**7.1.2 Connection Strategy** The cost of Bidirectional Path Tracing over traditional Path Tracing is not negligible. Hence a practical shift mapping must be efficient. Naively shifting every path generated by the many connection strategies would be prohibitively expensive.

[Man+15] propose to drop connection strategies where the connecting path vertices are classified as specular. They reason that these strategies don't contribute much energy anyway since the BRDF at the connecting vertices is likely to have small values when evaluated into essentially random directions.

Ignoring these connection strategies allows us to shift the light and eye paths once and reuse large parts of the original shifts (including their Jacobian values).



**Figure 17:** A scene which is difficult to render with standard path tracing, because the light sources are small and covered. Even after rendering 20 000 SPP it did not converge to a clear image.

## 7.2 Temporal Gradient-Domain Path Tracing

Temporal methods have become popular in computer graphics in recent years. Most of them focus on exploiting temporal coherence through filtering rendered images in a post processing step. Especially "Temporal Anti Aliasing" is widely used in computer games and there's been a lot of research on how to deal with the artifacts introduced by temporal filtering. There have been attempts to apply temporal filtering to path tracing, but the common post processing based methods have trouble dealing with very noisy images and view dependent lighting effects [Sey16].

[Man+16] introduce *Temporal Gradient Domain Path Tracing* (T-GPT), a technique which extends gradient domain path tracing into the spatial domain. Instead of just computing gradients between pixels in space, they also compute differences over time. Additionally, instead of applying a post processing filter they take advantage of temporal coherence at a sample level. Similarly to how G-PT computes offset paths  $T_{ij}(\bar{x})$  between pixels in space, T-GPT adds offset paths between pixels in time as well. Additionally, second order differentials (differences in time and space) are computed. The resulting images  $I^{dx}$ ,  $I^{dy}$ ,  $I^{dt}$ ,  $I^{dxdt}$  and  $I^{dydt}$  are combined in a 3D Poisson reconstruction step.

**7.2.1 Shift Mapping** The shift mapping for spatial derivatives is the same as the one used for G-PT. For time derivatives two mappings are proposed. The first one is straight forward. The primary sample space (pixel) position of a path is kept constant between frames and paths are reconnected as usual. This can result in large path space differences if objects or the camera move between frames since the chance of an object moving away from under a ray is quite large. The second proposed shift mapping takes motion vectors, that is object and camera movement projected into primary sample space, into account. Using the motion vectors pixels positions are reprojected between frames. Instead of naively using the same primary sample space position for the temporal shift the reprojected one is used.

In practice the shifts with a temporal mapping are not computed explicitly. Instead we render two consecutive frames independently from each other. We ensure coherence between frames by starting off the random number generator with the same seed both times. In multi-threaded or multi-machine environments additional care has to be taken to ensure that rendering process is deterministic.

**7.2.2 Spatio-temporal Reconstruction** To ensure temporal coherence in the output image sequence (resulting in more pleasing, flicker-less animations) we need to solve for image *sequences* instead of single images. This is achieved by solving the following minimization problem. Note that all the symbols in the equation denote *image sequences*.

$$I = \underset{\bar{I}}{\operatorname{argmin}} (\|\alpha(\bar{I} - I^g)\|^p + \left\| \begin{pmatrix} H^{dx}\bar{I} \\ H^{dy}\bar{I} \\ H^{dt}\bar{I} \\ H^{dxdt}\bar{I} \\ H^{dydt}\bar{I} \end{pmatrix} - \begin{pmatrix} I^{dx} \\ I^{dy} \\ I^{dt} \\ I^{dxdt} \\ I^{dydt} \end{pmatrix} \right\|^p) \quad (25)$$

Where  $H^*$  denotes the finite difference operator and  $I^g$  is the sequence of primal images. This is basically identical to the reconstruction in Equation (11) except for the fact that we reconstruct a sequence of images. To achieve arbitrary length reconstruction [Man+16] use a sliding window of 20 frames.

## 8 Discussion

In this report we presented the state of the art in gradient domain rendering. We have provided an analysis that shows, under certain simplifications, how the process of sampling gradients followed by Poisson reconstruction suppresses high frequency noise and reduces the overall error. The papers which describe the individual techniques already have detailed analysis comparing their work to the non-gradient domain counterparts. Also, a direct comparison between gradient domain methods is not very insightful. For example, it is obvious that gradient domain path tracing is going to perform worse than gradient domain bidirectional path tracing in scenes with difficult light transport. Gradient domain rendering does not change the characteristics of the underlying technique, but provides a constant error reduction factor (see Equation (20)). Even with a very bad shift mapping, which always produces non-symmetric paths, we will not do worse than the base algorithm at equal sample counts as long as we handle non-symmetric shifts correctly. Something that is worth investigating is the run time overhead of gradient sampling, which is dominated by the time taken to construct offset paths. This is why equal time comparisons tend to be more meaningful than equal sample count comparisons. Additionally, it would be interesting to get statistics about how scene geometry influences the effectiveness of shift mappings. Getting insights into where the problem areas lie is helpful in the construction of new shift mappings.

### 8.1 Equal Time Comparisons

We want to compare the effectiveness of gradient sampling in various scenarios. For this we measure the relative mean squared error (relMSE) of the generated images after letting them render for fixed amount of time. The relMSE of a sampled image  $I$  is defined as  $\epsilon_{rel} = \|I_{ref} - I\|^2 / \|I_{ref}\|^2$  where  $I_{ref}$  is a converged reference image, usually obtained via path tracing a large number of samples. For each technique we render two images, one using the classical version of the techniques and one using gradient sampling. We measure the relative improvement by dividing their relMSEs. We also compute the predicted improvement by computing the factor given by Equation (20) for the reference image.

Figure 18 shows the result of this investigation. We use three scenes with different light transport characteristics. The scene a) has little power in the gradients but highly specular materials. We predict an improvement due to gradient sampling by 28x over classical sampling, but neither G-PT nor G-BDPT achieve factors close to this. Scene b) is the classic "Sponza" scene. It contains only diffuse materials and is illuminated by a large light, ideal conditions for both path tracing as well as the shift mapping employed for G-PT. In fact, we see that G-PT almost achieves the ideal improvement over PT. On the other hand, G-BDPT only reduces relMSE by 8 times, half of the theoretically predicted value. At last, the scene c) presents a difficult light transport scenario for PT. This is the only scene where G-BDPT outperforms PT in terms of relative error reduction.

### 8.2 Analysing Shift Mappings

To evaluate the performance of shift mappings we compute the percentage of failed paths per pixel. For this we modify the implementation by [Ket+15] and add an additional output. In Figure 19 we display the final image on the left and a heat map of failed samples on the right. Generally, we can see that shift failures concentrate on edges, though the specularity of the interacting materials also seems to have an influence.

To further investigate this we study a scene with materials of varying reflectivity and otherwise fairly simple light transport in Figure 20. Of interest here is that the shift does not fail frequently when hitting

a)					PT	BDPT	MLT
	$ F ^2$	0.3526		Classical	0.0090	0.0123	-
	$ G ^2$	0.0122		Gradient	0.0012	0.0021	-
	Predicted	28.86		Improvement	7.6683	5.8167	-
b)					PT	BDPT	MLT
	$ F ^2$	0.2814		Classical	0.0027	0.0068	-
	$ G ^2$	0.0184		Gradient	0.0002	0.0008	-
	Predicted	15.27		Improvement	14.063	8.6862	-
c)					PT	BDPT	MLT
	$ F ^2$	0.3003		Classical	0.0092	0.0086	-
	$ G ^2$	0.0175		Gradient	0.0043	0.0016	-
	Predicted	17.17		Improvement	2.1458	5.4328	-

Figure 18

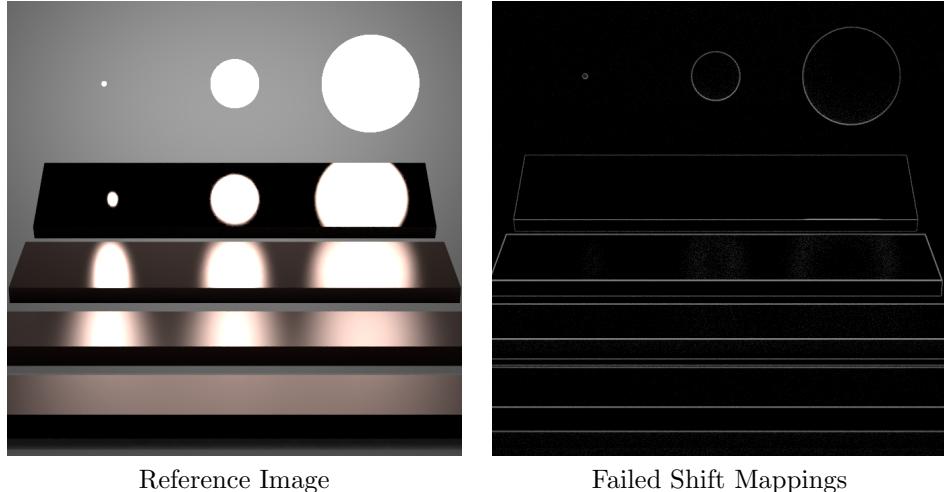


Figure 19

perfectly specular materials. It does have an issue with materials that are glossy but classified as diffuse (the second bar from the top).

Figure 21 serves as an example of a scene with difficult light transport. Again, we can see that most shifts fail on edges of objects. The complex light transport does not seem to reduce the effectiveness of the shift mapping much.

Concluding we want to note that, since many of the failing shifts lie on edges, edges will have higher variance in the gradients, because we discard failing shifts and revert back to the naive gradient sampling illustrated in Figure 4. This is unfortunate since these edges also tend to be the parts of the image where the gradients are high. Hence the effectiveness of gradient sampling is reduced and we conjecture that a shift mapping which can deal with shifts across object edges would improve results significantly. Even just sampling a gradient multiple times in case a shift fails might reduce overall error effectively. This would be a variant of adaptive sampling.



**Figure 20**



**Figure 21**

## 9 Acknowledgements

The scene files in this report have been mostly provided by Benedikt Bitterli ([Bit16]). "staircase" is made by the Blend Swap user "Wig42". "three spheres" and "glass egg" are modeled after scenes from Eric Veach. The photograph in Figure 12 is in the public domain. The renders used in Figure 18 are supplemental images provided as part of [Man+15]. The Matlab code used for analysis is ours and available on request.

## References

- [Bha+08] Pravin Bhat et al. "Fourier Analysis of the 2D Screened Poisson Equation for Gradient Domain Problems". In: *Proceedings of the 10th European Conference on Computer Vision: Part II*. ECCV '08. Marseille, France: Springer-Verlag, 2008, pp. 114–128. ISBN: 978-3-540-88685-3. DOI: 10.1007/978-3-540-88688-4\_9. URL: [http://dx.doi.org/10.1007/978-3-540-88688-4\\_9](http://dx.doi.org/10.1007/978-3-540-88688-4_9).
- [Bit16] Benedikt Bitterli. *Rendering resources*. <https://benedikt-bitterli.me/resources/>. 2016.
- [Bra00] R.N. Bracewell. *The Fourier Transform and Its Applications*. Electrical engineering series. McGraw Hill, 2000. ISBN: 9780073039381. URL: <https://books.google.de/books?id=ZNQQAQAAIAAJ>.

- [JM12] Wenzel Jakob and Steve Marschner. "Manifold Exploration: A Markov Chain Monte Carlo Technique for Rendering Scenes with Difficult Specular Transport". In: *ACM Trans. Graph.* 31.4 (July 2012), 58:1–58:13. ISSN: 0730-0301. DOI: 10.1145/2185520.2185554. URL: <http://doi.acm.org/10.1145/2185520.2185554>.
- [Jar+12] Wojciech Jarosz et al. "Theory, Analysis and Applications of 2D Global Illumination". In: *ACM Transactions on Graphics (Presented at SIGGRAPH)* 31.5 (Sept. 2012), 125:1–125:21. DOI: 10.1145/2231816.2231823.
- [Kaj86] James T. Kajiya. "The Rendering Equation". In: *SIGGRAPH Comput. Graph.* 20.4 (Aug. 1986), pp. 143–150. ISSN: 0097-8930. DOI: 10.1145/15886.15902. URL: <http://doi.acm.org/10.1145/15886.15902>.
- [Ket+15] Markus Kettunen et al. "Gradient-Domain Path Tracing". In: *ACM Trans. Graph.* 34.4 (2015).
- [Leh+13] Jaakko Lehtinen et al. "Gradient-domain Metropolis Light Transport". In: *ACM Trans. Graph.* 32.4 (July 2013), 95:1–95:12. ISSN: 0730-0301. DOI: 10.1145/2461912.2461943. URL: <http://doi.acm.org/10.1145/2461912.2461943>.
- [Man+15] Marco Manzi et al. "Gradient-Domain Bidirectional Path Tracing". In: *Eurographics Symposium on Rendering - Experimental Ideas and Implementations*. Ed. by Jaakko Lehtinen and Derek Nowrouzezahrai. The Eurographics Association, 2015. DOI: 10.2312/sre.20151168.
- [Man+14] Marco Manzi et al. "Improved Sampling for Gradient-domain Metropolis Light Transport". In: *ACM Trans. Graph.* 33.6 (Nov. 2014), 178:1–178:12. ISSN: 0730-0301. DOI: 10.1145/2661229.2661291. URL: <http://doi.acm.org/10.1145/2661229.2661291>.
- [Man+16] Marco Manzi et al. "Temporal Gradient-domain Path Tracing". In: *ACM Trans. Graph.* 35.6 (Nov. 2016), 246:1–246:9. ISSN: 0730-0301. DOI: 10.1145/2980179.2980256. URL: <http://doi.acm.org/10.1145/2980179.2980256>.
- [MC12] S.L. Miller and D.G. Childers. *Probability and Random Processes: With Applications to Signal Processing and Communications*. Academic Press, 2012. ISBN: 9780123869814. URL: <https://books.google.de/books?id=1rTcR3nIkJcC>.
- [Rud94] Daniel L Ruderman. "The statistics of natural images". In: *Network: computation in neural systems* 5.4 (1994), pp. 517–548.
- [Sey16] Dario Seyb. "Efficient Ray Tracing Techniques". Pro Seminar Report "Selected Topics in Computer Graphics" at RWTH Aachen. 2016.
- [Sey17] Dario Seyb. *Zaphod Raytracer Code*. 2017. URL: <http://github.com/daseyb/zaphod> (visited on 01/20/2017).
- [SSJ16] Kartic Subr, Gurprit Singh, and Wojciech Jarosz. "Fourier Analysis of Numerical Integration in Monte Carlo Rendering: Theory and Practice: Understanding Estimation Error in Monte Carlo Image Synthesis". In: *ACM SIGGRAPH 2016 Courses*. SIGGRAPH '16. Anaheim, California: ACM, 2016, 10:1–10:40. ISBN: 978-1-4503-4289-6. DOI: 10.1145/2897826.2927356. URL: <http://doi.acm.org/10.1145/2897826.2927356>.
- [Vea98] Eric Veach. "Robust Monte Carlo Methods for Light Transport Simulation". AAI9837162. PhD thesis. Stanford, CA, USA, 1998. ISBN: 0-591-90780-1.

