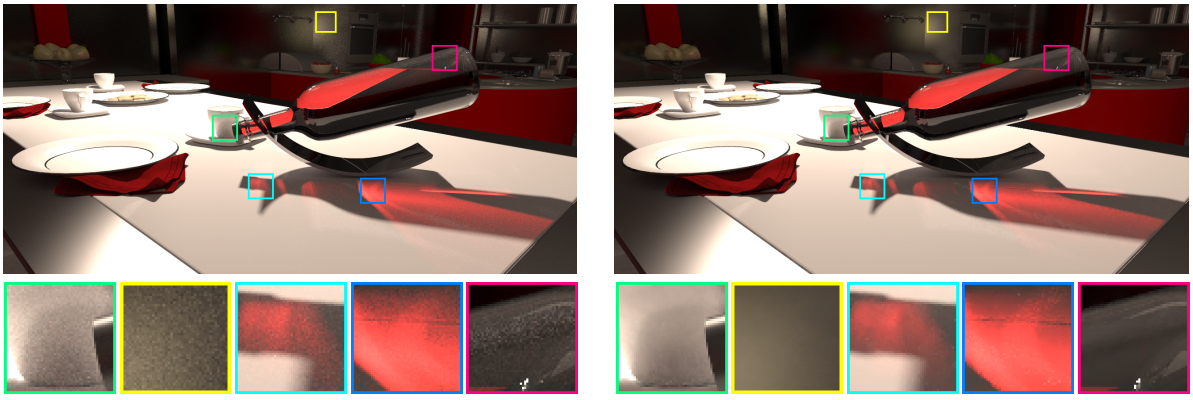
Eurographics Symposium on Rendering - Experimental Ideas & Implementations (2015)
J. Lehtinen and D. Nowrouzezahrai (Editors)

Gradient-Domain Bidirectional Path Tracing

Marco Manzi[1] Markus Kettunen[2] Miika Aittala[2] Jaakko Lehtinen[2,3] Frédo Durand[4] Matthias Zwicker[1]

[1]University of Bern [2]Aalto University [3]NVIDIA [4] MIT CSAIL



Bidirectional path tracing (BDPT) Gradient-domain bidirectional path tracing (G-BDPT)

Figure 1: *We compare results of bidirectional path tracing (BDPT, left) versus gradient-domain bidirectional path tracing (G-BDPT, right) after thirty minutes of render time. While BDPT still exhibits visible residual noise, G-BDPT is free of artifacts nearly everywhere with the exception of some difficult regions around caustics.*

Abstract

Gradient-domain path tracing has recently been introduced as an efficient realistic image synthesis algorithm. This paper introduces a bidirectional gradient-domain sampler that outperforms traditional bidirectional path tracing often by a factor of two to five in terms of squared error at equal render time. It also improves over unidirectional gradient-domain path tracing in challenging visibility conditions, similarly to how conventional bidirectional path tracing improves over its unidirectional counterpart. Our algorithm leverages a novel multiple importance sampling technique and an efficient implementation of a high-quality shift mapping suitable for bidirectional path tracing. We demonstrate the versatility of our approach in several challenging light transport scenarios.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Line and curve generation

1. Introduction

Gradient-domain methods have recently been introduced as efficient, general techniques for physically-based rendering [LKL*13, MRK*14, KMA*15]. Instead of directly estimating the radiance responses for each image pixel, they produce unbiased estimates of the *finite-difference gradients* between neighboring pixels by deterministically shifting paths between pixels. In a post-process step, the gradient estimates are integrated with a conventionally sampled, noisy "guide image" by solving the discrete screened Poisson equation. Together, these steps yield images with lower variance, and gradient-domain methods reduce the required render time to achieve the same quality compared to traditional samplers.

While gradient-domain rendering was originally proposed in the Markov Chain Monte Carlo (Metropolis) context, an upcoming paper [KMA*15] shows, both by theory and

DOI: 10.2312/sre.20151168

example, that similar benefits can also be claimed by a gradient-domain extension of standard path tracing with next event estimation. It is well known, however, that unidirectional path tracing is ineffective in scenes where light sources cannot be reached easily by tracing paths incrementally from the eye. Bidirectional path tracing deals with these situations much more robustly by constructing subpaths both starting at the eye and at light sources, and forming complete paths by making all possible connections.

In this paper, our objective is to combine the advantages of bidirectional path tracing and gradient-domain rendering. We describe a bidirectional gradient-domain light transport sampler (G-BDPT) that builds on bidirectional path tracing (BDPT). G-BDPT is useful in similar situations as conventional BDPT. This is the case in scenes with realistic light sources enclosed by light fixtures, or when the directly lit area is small, i.e., when sources contribute mainly indirect illumination. In both scenarios, connecting to light sources via shadow rays often fails, leading to excessive noise in unidirectional path tracers. In addition, we develop a novel multiple importance sampling technique, and describe an efficient implementation of a high-quality shift mapping to reduce sampling artifacts. Our results show that G-BDPT performs consistently better than its non-gradient counterpart, and that it yields significant improvement over standard (gradient) path tracing in scenarios that benefit from bidirectional sampling.

In summary, we make the following contributions:

- A bidirectional gradient-domain rendering algorithm (G-BDPT) based on a bidirectional light transport sampler;
- A multiple importance sampling (MIS) technique that combines MIS on gradients with conventional MIS for BDPT;
- An efficient implementation of a high-quality shift mapping using a modification of conventional BDPT path sampling.

2. Related Work

We base our work on the path space formulation of light transport due to Veach [Vea98]. That is, the intensity I_j for each pixel j in the image is obtained by integrating the radiance carried by all light paths with pixel filters:

$$I_j = \left(h(x) * \int_\Omega f(x,\bar{p})\,\mathrm{d}\mu(\bar{p}) \right)(x_j). \tag{1}$$

Here x is a pixel position, the $\bar{p}$ range over the set of all additional path parameters Ω, $f(x,\bar{p})$ is the image contribution function, and $h(\cdot)$ is the (shift-invariant) pixel filter. We obtain the value I_j of pixel j by evaluating the convolution at its position x_j.

Several Monte Carlo methods have been proposed for evaluating Equation 1. In particular, constructing light paths using successive independent sampling of scattering events results in path tracing [Kaj86]; combining the results of successive independent sampling from the camera and from the light yields bidirectional path tracing [LW93, VG94]. While not always superior to standard path tracing, bidirectional sampling is particularly effective in reducing noise in scenes with small, difficult to reach light sources. Our gradient-domain bidirectional sampler retains this significant advantage. In another vein, Markov Chain Monte Carlo methods perform random walks on light paths instead of drawing independent samples [VG97, KSKAC02].

2.1. Gradient-Domain Rendering

We cursorily describe the necessary theoretical background on gradient-domain rendering, and refer the reader elsewhere for complete details [KMA*15]. Gradient-domain rendering techniques [LKL*13, MRK*14, KMA*15] build on strictly the same basis as previous Monte Carlo methods — that is, they aim to evaluate Equation 1 using Monte Carlo sampling. In contrast to regular (Markov Chain) Monte Carlo methods, they do this indirectly by sampling image gradients (differences in brightness between neighboring pixels) in addition to the pixel intensities, using pairs of correlated path samples. The final intensities for all pixels are found using the sampled gradients and pixel values by solving a screened Poisson equation. Recent work has demonstrated that this, perhaps surprisingly, yields a significant reduction in total integration error [KMA*15], and when used in the Markov Chain context, diverts computational effort to regions of path space that contribute to significant changes in the image [LKL*13]. Our work follows the same line of thought.

2.2. Gradient-Domain Path Tracing

In gradient-domain rendering, differences between pixel intensities are computed by directly evaluating the difference in light throughput between two paths separated by one pixel and integrating this over all paths. More precisely, we denote the difference between the intensities of two pixels i and j by $\Delta_{i,j}$. As recently shown [KMA*15], this can be written as the integral of a path difference function $g_{ij}(x,\bar{p})$ instead of the usual image contribution function $f(x,\bar{p})$ as

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

where x is the image coordinate, $(x,\bar{p})$ is a light path with additional parameters $\bar{p}$ connecting a point on a light and a point on the sensor, f is the image contribution function, and T_{ij} is the *shift mapping* that deterministically maps a *base path* $(x,\bar{p})$ to a close-by *offset path* $T_{ij}(x,\bar{p})$. We only allow shifts that make sure that the offset path $T_{ij}(x,\bar{p})$ has the same pixel filter value as the base path, so we can express it as a single convolution. The factor $\left|T'\right| = \left|\partial T/\partial\bar{x}\right|$ denotes

the determinant of the Jacobian of $T(\bar{x})$ accounting for the change of integration variables [LKL*13].

Symmetric Gradients The previous formulation assumes the shift mapping is a bijection on path space. But this is not the case in practice because the shift may fail due to numerical reasons (see Section 3.1). In addition, it assumes that we can sample all paths $(x,\bar{p})$ that lead to a non-zero offset path $f(T_{ij}(x,\bar{p}))$. Monte Carlo path tracers like BDPT only guarantee to sample all base paths with non-zero contribution $f(x,\bar{p})$, however, and we may miss some non-zero offset paths, leading to biased gradients. As shown by Kettunen et al. [KMA*15], we can ensure that all relevant paths are sampled in both pixels by using the symmetric formulation

$$\Delta_{i,j} = \left(h(x) * \int_\Omega w_{ij}(x,\bar{p}) g_{ij}(x,\bar{p}) \mathrm{d}\mu(\bar{p})\right)(x_i) + \left(h(x) * \int_\Omega w_{ji}(x,\bar{p}) g_{ji}(x,\bar{p}) \mathrm{d}\mu(\bar{p})\right)(x_j). \quad (3)$$

The two integrals sample the same difference, once by shifting from pixel i to j and vice versa. The multiple importance sampling weights w_{ij} and w_{ji} serve two purposes: they are normalized to add up to one, such that the two integrals correctly add up to the desired gradients, and they reduce variance by tempering the effect of the local squeezing of path space caused by the shift. Finally, we need to take into account that the shift may not be invertible for some parts of path space, which means the symmetric formulation cannot be evaluated in these cases. We deal with this by simply sampling the contributions of paths to the two pixels i and j separately, without applying any shift mapping, and add them to (respectively subtract them from) $\Delta_{i,j}$.

Gradient MIS To set the weights in Equation 3, the forward and inverse mappings are interpreted as two sampling techniques to obtain the same base path $\bar{x} = (x,\bar{p})$. This makes it possible to derive multiple importance sampling weights

$$w_{ij}(\bar{x}) = \frac{p(\bar{x})}{p(\bar{x}) + p(T_{ij}(\bar{x}))|T'_{ij}(\bar{x})|}. \quad (4)$$

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.

G-PT Algorithm The gradient-domain path tracing algorithm (G-PT) simply draws a number of base paths from each pixel, shifts them to the four horizontal and vertical neighbor pixels, evaluates the differences between throughputs, weighted as shown above, and accumulates the results in a throughput image and four additional gradient images. The process yields the inputs required by the screened Poisson solver. When the shift mapping is designed so that throughput differences between base and offset are small (including the effect of the Jacobian), the resulting gradient estimates have low variance, which translates to higher quality in the final reconstructed image [KMA*15].

3. Bidirectional Gradient Sampling

We now describe a gradient-domain version of bidirectional path tracing (G-BDPT). We follow the general outline of Kettunen et al. [KMA*15], and view the problem as formulating a bidirectional Monte Carlo sampler for Equation 3.

A direct translation of BDPT with multiple importance sampling to the gradient domain would, however, lead to a prohibitively expensive algorithm, because the number of individual paths sampled is large (all connections are made between the eye and light subpaths). The naive algorithm that applies the shift mapping to each one turns out to be too expensive. We alleviate the issue by selectively removing some bidirectional connection strategies. This reduction in work allows us to use a more sophisticated shift mapping compared to [KMA*15], which we demonstrate to yield a net performance win.

3.1. Shift Mapping

For G-BDPT we use the shift mapping proposed for gradient-domain Metropolis rendering by Lehtinen et al. [LKL*13], which builds on the manifold perturbation technique by Jakob and Marschner [JM12]. They express their shift in the path parameterization by surface position, which is the natural parameterization for BDPT. Hence, we can directly reuse their mathematical formulation (and implementation). To apply the shift to a given path we need to classify its vertices as diffuse or specular, and we follow their approach using a threshold on the material roughness.

The intuition behind the G-BDPT shift is to preserve half-vectors at vertices classified as specular, while trying to connect to the base path as soon as possible. The main advantage over the shift proposed for G-PT is that it always connects to the base path at the second diffuse vertex (starting from and excluding the eye), independent of any specular vertices before that. In contrast, the G-PT shift requires two *consecutive* diffuse vertices to reconnect to the base path. This means that the G-BDPT shift generally produces more similar base-offset pairs than the G-PT shift.

We briefly review the definition of the G-BDPT shift as introduced by Lehtinen et al. [LKL*13]. Let us describe a path $\bar{x}$ as a sequence of vertex positions $\mathbf{x}_i$, $\bar{x} = \langle \mathbf{x}_0, \ldots, \mathbf{x}_n \rangle$. We formulate the shift mapping as the concatenation of a path reparameterization, a *simple shift* that only modifies the image plane intersection of the reparameterized path without changing the other parameters, and a reparameterization back. We classify vertices into diffuse and specular vertices based on material roughness as mentioned before. The key idea now is to design the reparameterization such that the *simple shift* described above preserves half-vectors at specular vertices, while connecting to the base path as soon as possible.

Let a, b, and c be the indices of the first three vertices classified as diffuse along the path starting at the eye (including

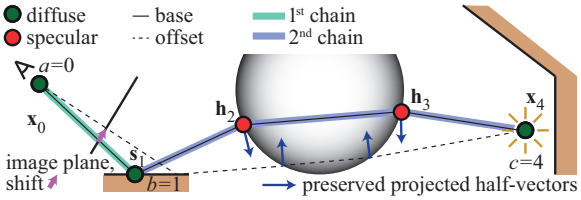


Figure 2: *Visualization of the reparameterization and the shift mapping in G-BDPT. Starting at the eye, the shift preserves two consecutive* half-vector chains *between the first three diffuse vertices a, b, and c along the path. It always connects to the base path at the third diffuse vertex (including the eye). In the figure, the first half-vector chain is* empty. *In contrast the G-PT shift [KMA*15] cannot connect to the vertex c on the light source.*

the eye vertex itself, which is classified as diffuse; hence we always have $a = 0$). In the reparameterization, we represent the specular vertices between the eye and b, and the specular vertices between b and c using projected half-vectors $\mathbf{h}_i$ (half-vectors projected onto local tangent planes) instead of vertex positions $\mathbf{x}_i$. We call vertices i with $a < i < b$ and $b < i < c$ the first and second *half-vector chain*, respectively. The first chain is *empty* if $b = 1$, and the second is empty if $c = b+1$. We write our reparameterization as $\hat{x} = \langle \mathbf{x}_0, \mathbf{s}_1, \mathbf{h}_1, \ldots, \mathbf{h}_b - 1, \mathbf{h}_{b+1}, \ldots \mathbf{h}_{c-1}, \mathbf{x}_c, \ldots, \mathbf{x}_n \rangle$, where $\mathbf{s}_1$ is the image plane intersection of the path, and the position of vertex b is determined implicitly by the parameters of previous vertices. In this parameterization, the simple shift only moves $\mathbf{s}_1$ to a neighboring pixel. We illustrate the reparameterization and the shift in Figure 2.

We implement the shift by moving $\mathbf{s}_1$ to a neighbor pixel, and re-tracing the first specular chain from the eye, which yields vertex b on the offset path. We reconnect offset vertex b to the base path via the second half-vector chain by applying the manifold perturbation by Jakob and Marschner [JM12]. To formulate the Jacobian of the shift, let us denote the shifted offset path in the original parameterization by surface position as $\bar{y}$, and in the reparameterization using half-vectors as $\hat{y}$. The Jacobian determinant of the shift is then

$$|T'(\bar{x})| = \left|\frac{\partial \bar{y}}{\partial \bar{x}}\right| = \left|\frac{\partial \mathbf{x}_0, \ldots, \mathbf{x}_n}{\partial \mathbf{y}_0, \ldots, \mathbf{y}_n}\right| = \left|\frac{\partial \bar{y}}{\partial \hat{y}}\right| \left|\frac{\partial \hat{x}}{\partial \bar{x}}\right|. \quad (5)$$

We compute the Jacobian determinants of the reparameterizations $|\partial \bar{y}/\partial \hat{y}|$ and $|\partial \hat{x}/\partial \bar{x}|$ as described by Lehtinen et al. [LKL*13].

3.2. Efficient Gradient Sampling

The computational cost of the G-BDPT shift is not negligible, and we need to employ it carefully to avoid large overheads. A naive implementation of the shift mapping would apply it separately to each path sampled by BDPT. For each eye and light subpath, however, BDPT samples all paths that can be obtained by connecting these subpaths. Shifting all connected paths and computing the Jacobians separately from scratch is prohibitively expensive. We reduce this cost by slightly modifying the usual BDPT sampling strategy.

Our key modification of usual BDPT is to omit sampling techniques that include a specular vertex (according to our classification) as a connecting vertex between eye and light subpaths. Omitting these sampling techniques has little impact on the effectiveness of BDPT, since connections involving non-diffuse vertices typically contribute very little. On the other hand, it allows us to reduce the cost to compute the shift mappings and their Jacobians.

For the Jacobian of the shift mapping described above only vertices $\leq c$ are relevant, since the shift is independent of the others. With our restriction on BDPT sampling techniques, vertices $\leq c$ may have been sampled in only three different ways illustrated in Figure 3: (i) all vertices (both half-vector chains) are sampled on the eye path, (ii) vertices up to and including b (only the first half-vector chain) are sampled on the eye path, and vertices $> b$ on the light path, (iii) only vertex a is sampled on the eye path (the second half-vector chain is sampled on the light path). We call such paths *light tracing paths*. Case (ii) implies $c = b+1$ (the second half-vector chain is empty), since we do not make connections with non-diffuse vertices. Also, in this case the Jacobian is given by vertices $\leq b$, since the shift is independent of vertex $c = b+1$. Similarly, case (iii) implies $b = a+1$ (the first half-vector chain is empty)†.

We take advantage of these observations as follows: Given an eye subpath $\bar{x}^E$, let us again denote the indices of its first three diffuse vertices a, b, c. We then apply the shift to vertices $\mathbf{x}^E_a, \ldots, \mathbf{x}^E_c$, yielding an offset path $\bar{y}^E$ for the eye subpath. Under the previous considerations, this is sufficient to construct *all* connected offset paths for cases (i), where the connection is with a vertex $\geq c$ on the eye subpath, and (ii), where the connection is with vertex b . Hence we need only two different Jacobians for all these paths, in case (i) for the shift of both half-vector chains $a, \ldots, b, \ldots, c$,

$$|T^{(i)}(\bar{x}^E)| = \left|\frac{\partial \mathbf{x}^E_a, \ldots, \mathbf{x}^E_c}{\partial \mathbf{y}^E_a, \ldots, \mathbf{y}^E_c}\right|, \quad (6)$$

and in case (ii) for only the first chain $a, \ldots, b$,

$$|T^{(ii)}(\bar{x}^E)| = \left|\frac{\partial \mathbf{x}^E_a, \ldots, \mathbf{x}^E_b}{\partial \mathbf{y}^E_a, \ldots, \mathbf{y}^E_b}\right|. \quad (7)$$

Dealing with light tracing paths in case (iii) is more expensive. Given a ligth subpath $\bar{x}^L$, each of its diffuse vertices needs to be connected to the eye to form a complete light tracing path. For each connected light tracing path we need

† Since we assume a pinhole camera, we omit a fourth case where both chains and the eye vertex are sampled from the light.

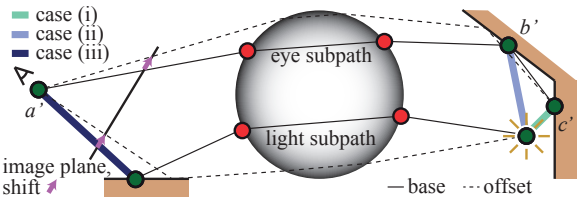


Figure 3: *Visualization of modified sampling strategy in G-BDPT. We do not sample subpath connections that include a specular vertex. We color code connections according to three cases: (i) both half-vector chains are sampled from the eye (the second chain is empty here, but this is not always the case); (ii) the first half-vector chain is sampled from the eye (the second one is always empty, since we omit connections to non-diffuse vertices); (iii) the second half-vector chain is sampled from the light (implying the first half-vector chain is always empty). To reduce clutter, the figure does not show connections of type (i) and (ii) to the last vertex on the light subpath, and it does not show the corresponding subpath connections on the offset paths.*

to recompute a shift and its Jacobian. Again, we compute all Jacobians as described by Lehtinen et al. [LKL*13].

3.3. Multiple Importance Sampling

Let us use the common notation (s,t) to represent the different sampling strategies in BDPT, where s is the number of vertices on the light subpath, and t the number of vertices on the eye subpath. In BDPT any given path with n vertices can be sampled using all techniques (s,t) where $s+t=n$. Multiple importance sampling (MIS) introduces a weight for each sampling technique to reduce variance in a provably good manner [VG95]. Here we extend usual MIS for G-BDPT by combining it with the gradient-MIS technique outlined in Section 2.

For each gradient sample we not only consider all the potential sampling techniques that could be used to sample the base path, but we also take into account that the gradient could be sampled using either the forward or the inverse mapping, as described in Section 2. The combined MIS weight for a gradient sample using the balance heuristics is then

$$w_{ij;st}(\bar{x}) = \frac{p_{s,t}(\bar{x})}{\sum_{k=0}^{k\leq s+t} p_{k,s+t-k}(\bar{x}) + p_{k,s+t-k}(T_{ij}(\bar{x}))|T'_{ij}|}, \quad (8)$$

where $p_{s,t}(\bar{x})$ is the probability density function (PDF) for the (s,t) sampling technique evaluated for the base path $\bar{x}$. Note that this implies that the sum of the weights over the two gradient directions is normalized, that is, $w_{ij;st}(\bar{x}) + w_{ji;st}(T_{ij}(\bar{x})) = 1$.

Figure 4 illustrates the effectiveness of our combined MIS

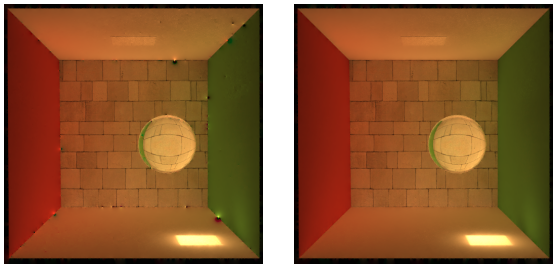


Figure 4: *Comparison of MIS techniques in G-BDPT using L_2 reconstruction. Left: G-BDPT with modified MIS weights given by Equation 9, that is, the balance heuristics for BDPT sampling of base paths with a discrete case distinction to avoid double counting of gradients. Right: G-BDPT with our combined MIS from Equation 8. Combined MIS effectively reduces gradient sampling artifacts in concave regions.*

approach. We compare our combined MIS weights from Equation 8 with a modified approach that only performs conventional BDPT MIS, but does not consider the two techniques to sample gradients. More precisely, the modified approach uses weights based on the balance heuristics

$$w_{ij;st}(\bar{x}) = \frac{r_{i,j}(\bar{x})p_{s,t}(\bar{x})}{\sum_{k=0}^{k\leq s+t} p_{k,s+t-k}(\bar{x})}, \quad (9)$$

where the factor $r_{i,j}(\bar{x})$ is necessary to make sure we do not double count gradient contributions in the symmetric formulation (Equation 3). We set $r_{i,j}(\bar{x}) = 1/2$ if there is a technique k that samples the gradient from the opposite direction, that is, there is a k with $p_{k,s+t-k}(T_{ij}(\bar{x})) > 0$. Otherwise, the gradient can be sampled only from one direction, and we set $r_{i,j}(\bar{x}) = 1$. The figure shows that combined MIS effectively reduces sampling artifacts in concave regions, which otherwise can only be avoided with sophisticated shift mappings [MRK*14].

4. Implementation

We implemented G-BDPT on top of the standard BDPT implementation in the freely available Mitsuba renderer [Jak12]. The basic structure of G-BDPT is very similar to BDPT, as shown in the pseudocode in Algorithm 1. For every sample, we draw an eye subpath $\bar{x}^E$ and a light subpath $\bar{x}^L$ (line 1). Then (lines 3-5), for each of the four horizontal and vertical neighbor pixels, we apply the shift mapping to the eye subpath to construct four shifted eye subpaths $\bar{y}^{E,j} = T_{ij}(\bar{x}^E)$. For each we obtain the corresponding Jacobians $|T_{ij}^{(i)}|$ and $|T_{ij}^{(ii)}|$ for cases (i) and (ii) as described in Equation 6 and Equation 7.

We then construct all complete base paths $\bar{x}$ by connecting $\bar{x}^E$ and $\bar{x}^L$ with all valid connection strategies (line 6), skipping connections between vertices classified as specular

Input: Scene and camera specification, total number of bidirectional samples N.
Output: Image I, gradient images $\Delta_{\cdot,\cdot}$.
for *all pixels and samples* **do**
[1] $\bar{x}^E, \bar{x}^L =$ sample eye and light subpath
[2] $i =$ screen-space position of $\bar{x}_E$
$a,b,c =$ first three diffuse vertices on $\bar{x}^E$
for *all neighbours j of i* **do**
[3] $\bar{y}^{E,j} = T_{ij}(\bar{x}^E)$ **// shift eye subpath**
[4] $|T_{ij}^{(i)}| = \left|\frac{\partial[\mathbf{y}_a^{E,j},\ldots,\mathbf{y}_c^{E,j}]}{\partial[\mathbf{x}_a^E,\ldots,\mathbf{x}_c^E]}\right|$ **// case (i) Jacobian**
[5] $|T_{ij}^{(ii)}| = \left|\frac{\partial[\mathbf{y}_a^{E,j},\ldots,\mathbf{y}_b^{E,j}]}{\partial[\mathbf{x}_a^E,\ldots,\mathbf{x}_b^E]}\right|$ **// case (ii) Jacobian**
end
for *all connection strategies (s,t)* **do**
[6] $\bar{x} = connect(s,t,\bar{x}^E,\bar{x}^L)$ **// base path**
[7] $i =$ screen-space position of $\bar{x}$
$a,b,c =$ first three diffuse vertices on $\bar{x}$
[8] **if** ***case (iii)*** **// light tracing path,** $t = 1$ **then**
for *all neighbours j of i* **do**
[9] $\bar{y} = T_{ij}(\bar{x})$ **// recompute shift**
[10] $|T_{ij}| = \left|\frac{\partial[\mathbf{y}_a,\ldots,\mathbf{y}_c]}{\partial[\mathbf{x}_a,\ldots,\mathbf{x}_c]}\right|$ **// Jacobian**
[11] $\Delta_{ij} = \Delta_{ij} + w_{ij;st}(\bar{x})\left[f(\bar{y})|T_{ij}| - f(\bar{x})\right]/p_{st}(\bar{x})$
end
else
for *all neighbours j of i* **do**
[12] $\bar{y} = connect(s,t,\bar{y}^{E,j},\bar{x}^L)$
if ***case (ii)*** **then**
[13] $\Delta_{i,j} = \Delta_{i,j} + w_{ij;st}(\bar{x})\left[f(\bar{y})|T_{ij}^{(ii)}| - f(\bar{x})\right]/p_{st}(\bar{x})$
else
[14] $\Delta_{i,j} = \Delta_{i,j} + w_{ij;st}(\bar{x})\left[f(\bar{y})|T_{ij}^{(i)}| - f(\bar{x})\right]/p_{st}(\bar{x})$
end
end
end
[15] $I_i = I_i + w_{i;st}(\bar{x}) f(\bar{x})/p_{st}(\bar{x})$
end
end
[16] $I = I/N; \Delta_{\cdot,\cdot} = \Delta_{\cdot,\cdot}/N$
[17] Reconstruct(I, $\Delta_{\cdot,\cdot}$,α)

Algorithm 1: *Pseudocode for gradient-domain bidirectional path tracing (G-BDPT).*

according to our roughness criterion. We also determine the pixel index i of the base path (line 7, this may be different from the pixel index corresponding to the eye subpath). Next (line 8) we check if the current base path is a light tracing path, that is, whether it connects a vertex on the light subpath directly to the eye (case (iii), meaning $t = 1$). Because in these cases the eye subpath has only one vertex (the eye), we cannot make use of the shifted eye subpaths $\bar{y}^{E,j}$ for shifting these paths. Instead, we must apply the shift mapping (and compute its Jacobian) to each light-traced path separately (line 9 and 10). We then compute the gradient sample contribution and weight it by the MIS weight defined in Equation 8 (line 11). For connection strategies that do not directly connect with the eye vertex, we form the offset path by connecting the shifted eye subpath $\bar{y}^{E,j}$ with the light subpath $\bar{x}^L$ (line 12). We account for the two cases (i) and (ii) by choosing the correct Jacobian determinants (lines 13 and 14).

We also store the value of the base sample in the primal image (line 15). The weight $\omega_{i;st}$ is the usual power or balance heuristic, not the one from Equation 8.

Finally, we normalize both the primal image and the gradient images by the total number of bidirectional samples (line 16), which also accounts for the light paths that are distributed non-uniformly over the image. Then we perform screened Poisson reconstruction on the output of the renderer (line 17), as in previous work [KMA*15]. Our L_1 solver is based on iteratively reweighed least squares (IRLS) implemented through the conjugate gradient method in CUDA. Its performance is less than a second for a 720p image.

As an important detail, we treat some situations in a slightly different manner than implied by the pseudocode. This is when in lines 9 and 12 the offset paths $\bar{y}$ cannot be constructed because the shift failed for numerical reasons, or when these offset paths are blocked, and when the base path $\bar{x}$ (line 6) is blocked. In these cases we fall back to naive gradient sampling, that is, we simply set the contribution of the offset path to zero, and we use conventional MIS weights for the base path, instead of combined MIS (Equation 8). In the case of failing shifts, this is our only option. In the case of blocked paths, it allows us to take a number of early exits in our implementation that lead to some performance gains.

5. Results and Discussion

We evaluate G-BDPT by comparing to standard bidirectional path tracing (BDPT), standard path tracing (PT), and gradient-domain path tracing (G-PT) [KMA*15]. All methods were implemented in the Mitsuba renderer [Jak12]. We generated reference images using BDPT with 32000 samples per pixel. Except where expressly stated otherwise, all evaluations use L_1 reconstruction. All results are computed with 24 rendering threads on a workstation with dual Intel Xeon E5645 processors with a total of twelve cores at 2.5GHz.

For comparisons, we use relative mean squared error (relMSE), which we compute as relMSE $=$ average$[(X - R)^2/(R^2 + 0.001)]$, where R is a reference pixel and X our estimate. With all four compared methods, two of our test scenes (GLASS EGG and BOTTLE) suffer from massive

spike noise due to difficult caustics. Because this corrupts the metric, we ignored the 0.01% of the highest pixel errors in the relMSE computation in these scenes[‡].

5.1. Evaluation of G-BDPT

In Figure 6, we visually compare the four methods at equal render time. Full resolution images with error scores are also provided as supplemental material. In Figure 5, we plot the numerical convergence of all methods. We now briefly discuss each scene.

GLASS EGG is a standard benchmark scene for BDPT. The light from the lamp on the left illuminates most of the scene indirectly, making it hard for unidirectional PT to connect paths with the light source. Additionally, the glass egg on the table is lit directly by a second light source which creates a strong caustic that is notoriously hard to sample with PT. Unsurprisingly, BDPT performs much better in this scene than PT. The behaviour of G-PT and G-BDPT is more interesting: G-PT reduces the error compared to PT by a very large margin, but for low sampling rates, it improves sublinearly with time. The reason for this is that direct caustics lead to strong spike noise with unidirectional sampling, and the L_1 reconstruction suppresses the effect of such noise as outliers. This means the L_1 reconstruction reduces the error significantly at the price of removing most of the caustic. With higher sampling rates, the convergence rate becomes more linear again since less of the caustic is removed in reconstruction. Since a bidirectional sampler can resolve caustics much better, G-BDPT does not suffer from this. Compared to its non-gradient counterpart, G-BDPT leads to an improvement of a factor of five, in terms of render time to same quality.

DOOR is a benchmark scene for testing rendering algorithms under challenging illumination conditions. All visible light has to pass from another room through a thin crack of the door into the visible part of scene. For an unidirectional sampler it is very unlikely to find a path that connects the eye to the light, since it has to randomly pass through the thin crack. For bidirectional path sampler this is slightly easier since it is enough if either the eye subpath or the light subpath randomly passes through the crack. However, the somewhat higher chance of finding valid paths is nullified by the higher overhead of BDPT. Therefore, the performance of both non-gradient algorithms is approximatively equal. Still, G-BDPT outperforms G-PT by a factor of approximately two. This is most likely due to the superior shift-mapping (see Section 5.3).

BOTTLE is a complex scene with many glossy and specular surfaces, and a prominent direct caustic due to a small area light source. Similar to GLASS EGG, PT and G-PT fail to capture the caustic in a satisfactory way, while BDPT and G-BDPT succeed. The non-linear convergence for low sampling rates is stronger here for G-PT than in GLASS EGG because the caustic covers a bigger fraction of the image. Again, G-BDPT does not suffer from this, and provides a benefit over BDPT by a factor of two.

Scene	G-PT	G-BDPT w/o LTP	G-BDPT w/ LTP	G-BDPT HV
Glass Egg	x2.87	x3.83	x5.71	x5.56
Door	x3.18	x3.34	x3.49	x3.48
Bottle	x2.16	x3.42	x3.48	x3.72
Bathroom	x2.39	x3.42	x3.83	x4.12
Sponza	x2.47	x3.88	x4.03	x4.14

Table 1: *The overhead of the different gradient-domain methods compared to their non-gradient counterparts at equal number of base samples. An overhead of* 5 *means that gradient and conventional samples are equally expensive. We compare G-BDPT in three set-ups: without light tracing paths and with the Manifold perturbation shift mapping (3rd column), with light tracing paths and the Manifold perturbation shift mapping (4th column), and with light tracing paths and the half-vector preserving mapping (5th column).*

Since the performance of the gradient-domain methods is highly dependent on the performance of the underlying sampler, we also analyzed scenes where bidirectional sampling strategies are not beneficial.

BATHROOM is a complex scene that is illuminated by a large light source from the outside through a glass window. As the light source is large, even the unidirectional sampler has a good chance to randomly hit the source, even in presence of the glass in between, making bidirectional sampling of not much use. Because of this, the unidirectional methods (PT and G-PT) both have, at higher sample counts, an error about 30% lower than their bidirectional counterparts. Nonetheless, G-BDPT still improves over BDPT by a factor of six.

Finally, SPONZA is a simple scene consisting of diffuse surfaces only that are illuminated by a large area light. Since there are no caustics and no challenging illumination conditions, the overhead of bidirectional sampling is not amortized in equal time comparisons. Thus in comparison both bidirectional samplers are beaten by the unidirectional ones roughly by a factor of two. Again, G-BDPT improves on its non-gradient counterpart, here by almost an order of magnitude.

5.2. Computational Overhead

Intuitively, there are two reasons why gradient-domain rendering improves over conventional approaches at equal render time: first, sampled gradient samples have less

[‡] All images, including the references, are available in the supplemental material for inspection.

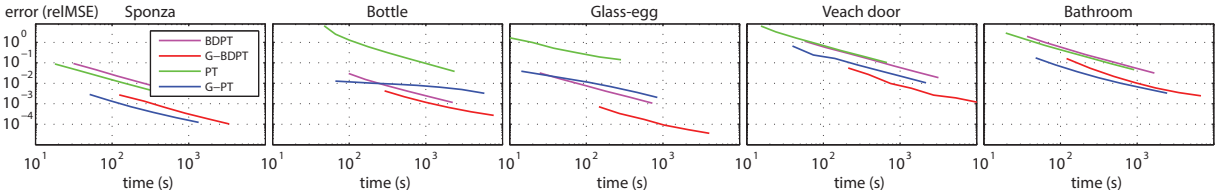


Figure 5: *Error plots of the scenes used in this paper, comparing bidirectional path tracing (BDPT), gradient-domain bidirectional path tracing (G-BDPT), path tracing (PT), and gradient-domain path tracing (G-PT) at equal render time. The error is measured as relative mean squared error (relMSE).*

variance than sampled pixels in general, as shown recently [KMA*15]; and second, the overhead for computing a gradient sample is typically cheaper than obtaining a conventional sample, because it does not require tracing a full path.

To show this empirically, we measured the overhead of gradient-sampling by comparing the rendering time of gradient and non-gradient methods with the same number of base samples per pixel. That is, we compare gradient rendering with n base samples and $4n$ offset paths to conventional rendering with n samples. Hence at equal costs per sample, gradient-domain rendering would have an overhead factor of 5 in this comparison. We summarize our empirical results in Table 1. For G-PT (first column) we measured a scene-dependent overhead factor of 2.2 to 3.2, which agrees with Kettunen et al. [KMA*15]. For G-BDPT we report on two configurations (second and third column), first without the expensive sampling strategies for light tracing paths, and then with it. To make the comparison meaningful we configured BDPT in the same way. The results show that without light tracing paths the overhead is roughly around 3.5 for all scenes. Including light tracing paths increases the overhead in general, but the increase is highly scene dependent. The overhead can even become larger than 5, meaning that gradient samples become more expensive than conventional samples. This is the case in GLASS EGG where many, potentially long light tracing paths need to be shifted for each base path. The different overheads of G-PT and G-BDPT without light tracing paths can probably be attributed to different levels of code optimization.

5.3. Evaluation of the shift mapping

To justify our decision to use the manifold perturbation shift mapping from Lehtinen et al. [LKL*13] we compared it to the simpler "half-vector shift mapping" from Kettunen et al. [KMA*15]. In a nutshell, this shift preserves the half-vectors of the base path along the offset path starting at the eye, and reconnects the offset path to the base path as soon as it encounters two consecutive diffuse vertices. For the comparison we implemented both shift mappings in our G-BDPT framework. In all tested scenes, G-BDPT with the manifold perturbation shift yielded more pleasing results than with the half-vector shift at equal time. Figure 7 shows an example. Surprisingly, despite the gradient descent optimization required in manifold perturbation, the overhead compared to the half-vector shift is very small in practice. We report empirical measurements in Table 1, rightmost column, which shows that the overheads are consistently very similar. There are two reasons. First, the gradient descent is only applied for a small fraction of all shifts; second, the manifold perturbation shift can often connect earlier to the base path, and thus must shift fewer vertices.

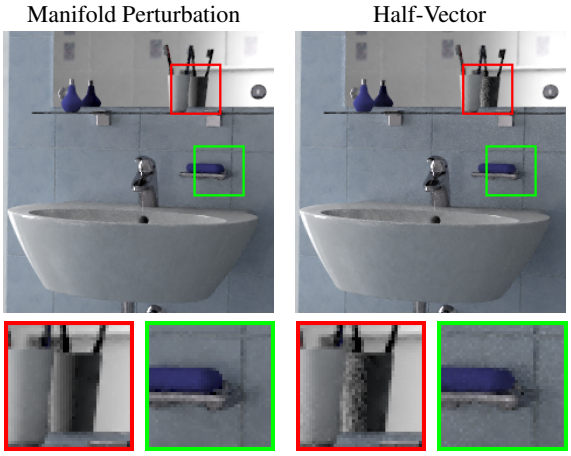


Figure 7: *Comparison of the Manifold perturbation shift mapping [LKL*13] and the half-vector preserving shift mapping [KMA*15] on the* BATHROOM *scene at 1024spp.*

6. Conclusions

We presented gradient-domain bidirectional path tracing, a gradient-domain rendering algorithm that significantly and consistently improves performance in comparison to standard bidirectional path tracing. Compared to previous unidirectional gradient-domain path tracing, this is most useful in scenarios where the additional cost of bidirectional sampling is justified, in particular for scenes with caustics or light sources that are not easily reachable for unidirectional path tracers. Our method retains the attractive prop-

erties of gradient-domain path tracing in that it is an unbiased estimator when using L_2 reconstruction, and can be used in conjunction with the more outlier-friendly L_1 reconstruction. In addition, we have shown that a shift mapping based on Manifold perturbation is advantageous compared to the half-vector preserving shift proposed previously for gradient-domain path tracing, providing improved image quality at almost no additional cost.

While this paper shows the viability and benefits of gradient-domain bidirectional path tracing, there are many attractive avenues for future research to further reduce variance and sampling artifacts. We will investigate more powerful reconstruction techniques, combining different shift mappings, and more advanced gradient sampling techniques.

Acknowledgments

The "Door" and "Glass-Egg" scenes were modelled after scenes by Eric Veach by Miika Aittala, Samuli Laine, and Jaakko Lehtinen. The "Sponza" scene is courtesy of Marko Dabrovic. The "Bottle" 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

[Jak12] JAKOB W.: Mitsuba v0.4. http://mitsuba-renderer.org, 2012. 5, 7

[JM12] JAKOB W., MARSCHNER S.: Manifold exploration: A Markov Chain Monte Carlo technique for rendering scenes with difficult specular transport. *ACM Trans. Graph. 31*, 4 (2012), 58:1–58:13. 3, 4

[Kaj86] KAJIYA J. T.: The rendering equation. In *Proc. ACM SIGGRAPH 86* (1986), pp. 143–150. 2

[KMA*15] KETTUNEN M., MANZI M., AITTALA M., LEHTINEN J., DURAND F., ZWICKER M.: Gradient-domain path tracing. *ACM Trans. Graph. (to appear) 35*, 4 (2015). 1, 2, 3, 4, 6, 7, 8

[KSKAC02] KELEMEN C., SZIRMAY-KALOS L., ANTAL G., CSONKA F.: A simple and robust mutation strategy for the Metropolis light transport algorithm. *Comput. Graph. Forum 21*, 3 (2002), 531–540. 2

[LW93] LAFORTUNE E. P., AND WILLEMS Y. D.: Bi-Directional Path Tracing. *Proceedings of Computergraphics* (1993), p. 145–153. 2

[LKL*13] LEHTINEN J., KARRAS T., LAINE S., AITTALA M., DURAND F., AILA T.: Gradient-Domain Metropolis Light Transport. *ACM Trans. Graph. 32*, 4 (2013). 1, 2, 3, 4, 5, 8

[MRK*14] MANZI M., ROUSSELLE F., KETTUNEN M., LEHTINEN J., ZWICKER M.: Improved sampling for gradient-domain metropolis light transport. *ACM Trans. Graph. 33*, 6 (Nov. 2014), 178:1–178:12. 1, 2, 5

[Vea98] VEACH E.: *Robust Monte Carlo methods for light transport simulation*. PhD thesis, Stanford University, 1998. 2

[VG94] VEACH E., AND GUIBAS L. J.: Bidirectional Estimators for Light Transport. *Eurographics Rendering Workshop Proceedings* (1994), pp. 147–162. 2

[VG95] VEACH E., GUIBAS L. J.: Optimally combining sampling techniques for Monte Carlo rendering. In *Proc. ACM SIGGRAPH 95* (1995), pp. 419–428. 5

[VG97] VEACH E., GUIBAS L. J.: Metropolis light transport. In *Proc. ACM SIGGRAPH 97* (1997), pp. 65–76. 2

Figure 6: *Visual equal-time comparison of path tracing (PT), gradient-domain path tracing (G-PT), bidirectional path tracing (BDPT) and gradient-domain bidirectional path tracing (G-BDPT). We provide the full resolution images with numerical error measurements in the supplemental material.*