Gradient-Domain Vertex Connection and Merging

Weilun Sun

Gradient-Domain Vertex Connection and Merging

by Weilun Sun

Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, in partial satisfaction of the requirements for the degree of **Master of Science, Plan II**.

Approval for the Report and Comprehensive Examination:

Committee:

[illegible]

Professor Yi-Ren Ng
Research Advisor

(Date)

* * * * * * *

Ravi Ramamoorthi

Professor Ravi Ramamoorthi
Second Reader

(Date)

Abstract

Gradient Domain Vertex Connection and Merging

by

Weilun Sun

Master of Science in Computer Science

University of California, Berkeley

Professor Yi-Ren Ng, Chair

Recently, gradient-domain rendering techniques have shown great promise in reducing Monte Carlo noise and improving overall rendering efficiency. However, all existing gradient-domain methods are built exclusively on top of Monte Carlo integration or density estimation. While these methods can be effective, combining Monte Carlo integration and density estimation has been shown (in the primal domain) to more robustly handle a wider variety of light paths from arbitrarily complex scenes. We present gradient-domain vertex connection and merging (G-VCM), a new gradient-domain technique motivated by primal domain VCM. Our method enables robust gradient sampling in the presence of complex transport, such as specular-diffuse-specular paths, while retaining the denoising power and fast convergence of gradient-domain bidirectional path tracing. We show that G-VCM is able to handle a variety of scenes that exhibit slow convergence when rendered with previous gradient-domain methods.

Contents

Chapter 1

Introduction

The core of light transport simulation methods is to numerically solve the complex rendering equation[9]. Over the years, numerous solutions have been proposed; yet capturing all paths of light in both a fast and efficient manner remains challenging and illusive. One popular class of techniques is based on directly estimating the pixel intensities by sampling potential light paths between the camera and light sources. Bidirectional path tracing (BDPT) [11, 18] is one of the most general techniques along this line of work. The power of BDPT comes from combining multiple complementary sampling techniques through multiple importance sampling [19]. Despite the success of BDPT in many scenarios, specular-diffuse-specular (SDS) paths which contribute to important visual effects are still problematic for the method. Another separate class of techniques is based on density estimation. The representative work along this line is photon mapping (PM) [8] and its variants[3, 2, 15]. Complementary to BDPT, PM based methods are very efficient in handling SDS paths, but have difficulty sampling diffuse/glossy interactions. To take the best of both classes, vertex connection and merging (VCM)[1] or unified path sampling (UPS)[4] was proposed to unify them in the same framework. The key to the unification is to treat PM as a probabilistic connection based sampling technique which aligns well with the BDPT formulation. Through the combination, VCM is able to inherit both BDPT's fast convergence in multiple diffuse/glossy interactions and PM's ability to handle SDS paths. It is considered a leap forward towards robust light transport simulation.

Despite VCM's robustness it still suffers from noise, like other primal domain techniques. Recently, gradient-domain methods have demonstrated an ability to generate smooth images by extending standard pixel estimators with correlated gradient samples. Using a screened Poisson reconstruction of both the original pixel values and gradients, the resulting rendered image is often much smoother than those generated by primal domain counterparts.

The concept of gradient-domain rendering was first proposed in gradient-domain metropolis light transport[12] as a Markov chain Monte Carlo method. Later, gradient-domain path tracing[10] extended standard path tracing to the gradient domain through sampling correlated offset paths from the base paths. Gradient-domain bidirectional path tracing (G-BDPT)[13] was then proposed to generate offset paths from base paths sampled by BDPT in an efficient way, enabling robust gradient sampling in scenes with highly occluded light sources. Temporal gradient-domain path tracing[14] further extended G-PT in animation rendering which is orthogonal to our discussions.

We present gradient-domain vertex connection and merging (G-VCM), an extension of VCM to the gradient domain. Formulating gradient-domain density estimation in the path integral formalism is an open problem we address in order to combine G-BDPT and density estimation based gradient estimates. To do so, we propose a new gradient sampling strategy, *gradient-domain vertex merging*. Our new strategy inherits the path reuse power of density estimation to handle complex light transport, e.g. SDS paths, and is easy to combine with other complementary gradient-domain strategies. We use multiple importance sampling to combine G-BDPT and gradient-domain vertex merging strategies across (potentially) many vertices on sensor subpaths.

Our contributions include:

- a method to generate correlated samples and gradient estimates using a vertex merging strategy that is compatible with the path integral framework (Chapter 4),
- a robust gradient estimator using vertex connection and merging combined with multiple importance sampling (Chapter 5), and
- a new rendering algorithm, G-VCM, able to robustly sample light paths in both primal and gradient domains, greatly improving overall image quality in challenging scenes compared to previous methods (e.g., see Figures 1.1 and 7.4; Chapter 7).

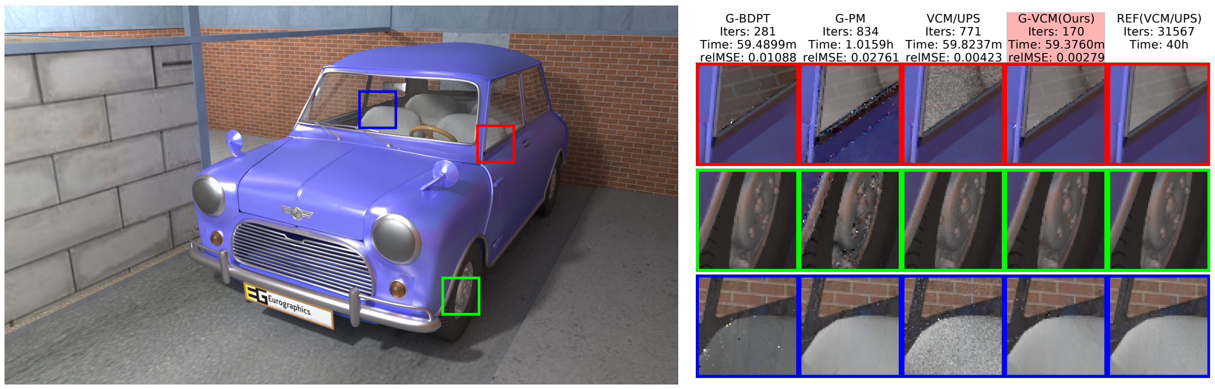

Figure 1.1: Equal time (1 hr) comparison in the Car scene. Here, G-BDPT iterations correspond to samples per pixel. Our method significantly reduces noise compared to VCM/UPS[1, 4]. G-BDPT[13] completely misses the SDS path contribution on the seats and G-PM[5] struggles with the glossy window frame and wheel hub. Our method robustly captures all of these transport contributions.

In concurrent work, gradient-domain photon density estimation (G-PM)[5] proposes gradient sampling in the density estimation framework. G-PM is shown to outperform G-BDPT and its primal domain counterpart, progressive photon mapping[3], in scenes dominated by SDS paths. However, this formulation relies (in a unidirectional sense) exclusively on density estimation. Without multiple integration techniques, G-PM remains susceptible to robustness issues with low photon density, i.e., as light paths that travel through multiple diffuse/glossy reflections (Figure 1.1). We elaborate on the key differences between our gradient-domain vertex merging strategy and G-PM in Section 7.2.

Chapter 2

Background

We quickly provide a technical overview of background in both primal- and gradient-domain path-space and density estimation.

2.1 Multiple Importance Sampling

Multiple importance sampling (MIS)[19] is a way to reliably combine multiple Monte Carlo (MC) estimators of the same integral. Consider the integral $I = \int_\Omega f(x) d\mu(x)$, where f is a real valued function and μ is a measure over the integration domain Ω. An MC estimate of I using MIS is

$$\hat{I} = \sum_{i=1}^{n} \frac{1}{n_i} \sum_{j=1}^{n_i} \omega(X_{i,j}) f(X_{i,j}) \Big/ p_i(X_{i,j}), \tag{2.1}$$

where n distributions $p_1, p_2, ..., p_n$ are sampled at points $X_{i,j}$ (the j^{th} of the n_i independent samples drawn from distribution p_i), and $\omega(X_{i,j})$ is the weight of the individual estimators at $X_{i,j}$. The power heuristic $\omega(X_{i,j}) = [n_i p_i(X_{i,j})]^\beta \Big/ \sum_{k=1}^{n} [n_k p_k(X_{i,j})]^\beta$ is a provably good strategy for the combination; using $\beta = 1$ corresponds to the balance heuristic which has been shown to work well in practice.

2.2 Bidirectional Path Tracing

We follow Veach's path space formulation[17], where the goal of light transport simulation is to estimate pixel integrals

$$I = \int_\Omega f(\mathbf{x}) d\mu(\mathbf{x}), \tag{2.2}$$

where $\mathbf{x}$ is a path consisting of $k+1$ vertices $x_0 ... x_k$, μ is the area product measure $d\mu(\mathbf{x}) = dA(x_0)...dA(x_k)$, Ω is the set of all paths contributing to the sensor location, and f is the

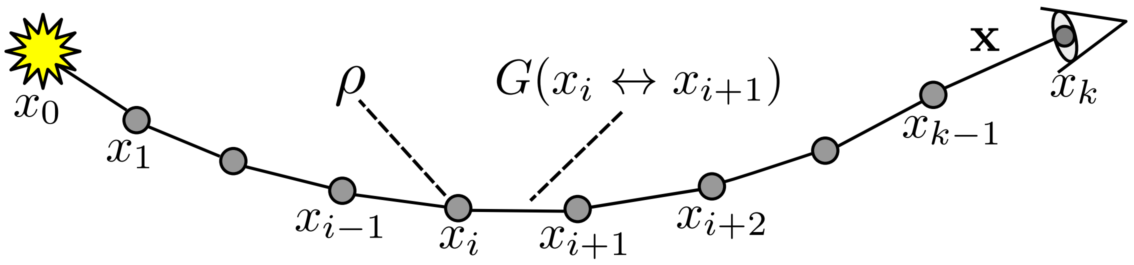

Figure 2.1: Path space integral.

path measurement contribution function of the following form

$$f(\mathbf{x}) = L_e(x_0)G(x_0 \leftrightarrow x_1)W_e(x_k)\prod_{i=1}^{k-1}\rho(x_{i-1},x_i,x_{i+1})G(x_i \leftrightarrow x_{i+1}), \tag{2.3}$$

where L_e is the emitter radiance, G the geometry term, ρ the BRDF and W_e the sensor importance (see the example in Figure 2.1).

As illustrated in Figure 2.2, bidirectional path tracing samples the path integral by first generating an emitter subpath $\mathbf{y} = y_0...y_{n_L-1}$ and a sensor subpath $\mathbf{z} = z_0...z_{n_E-1}$ for each pixel. Then, $n_L \times n_E$ full paths can be constructed by connecting any pair of vertices, y_s and z_t, chosen from $\mathbf{y}$ and $\mathbf{z}$ respectively. Let $\mathbf{x}_{s,t} = y_0...y_sz_t...z_0 = x_0...x_sx_{s+1}...x_k$ be the full path constructed by connecting y_s and z_t and $p_i(\mathbf{x}_{s,t})$ be the probability of sampling path $\mathbf{x}_{s,t}$ by connecting at x_i and x_{i+1}. Then, $f(\mathbf{x}_{s,t})\big/p_s(\mathbf{x}_{s,t})$ can be used as an estimator of the partial path integral contributed by all paths of k vertices. Finally, to obtain an unbiased estimator of the path integral over all paths, the estimators from all $n_L \times n_E$ full paths are combined as

$$\hat{I} = \sum_{s\geq 0}\sum_{t\geq 0}\omega_s(\mathbf{x}_{s,t})f(\mathbf{x}_{s,t})\big/p_s(\mathbf{x}_{s,t}), \tag{2.4}$$

where $\omega_s(\mathbf{x}_{s,t}) = p_s(\mathbf{x}_{s,t})^\beta \big/ \sum_{i=0}^{k-1}[p_i(\mathbf{x}_{s,t})]^\beta$ is the multiple importance sampling weight to take all $k-1$ connection strategies of sampling path $\mathbf{x}_{s,t}$ into account.

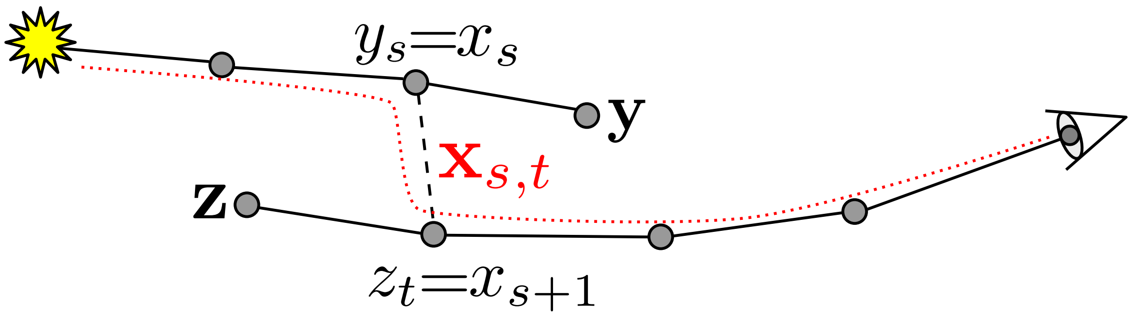

Figure 2.2: BDPT Paths.

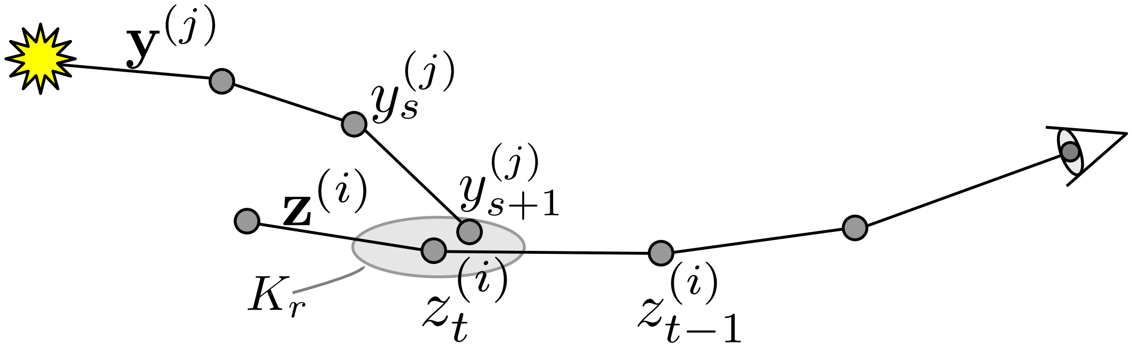

Figure 2.3: VCM paths.

2.3 Vertex Connection and Merging

Suppose we generate N pairs of emitter and sensor subpaths. Often, N is set to the number of pixels, to enable stratified sensor sampling. We denote $\mathbf{y}^{(i)}$ and $\mathbf{z}^{(i)}$ as emitter and sensor subpaths generated for pixel i. BDPT evaluates each pair separately through connections. That is, for pixel i, BDPT makes connections only between $\mathbf{y}^{(i)}$ and $\mathbf{z}^{(i)}$ to sample full paths. VCM extends BDPT by adding a vertex merging strategy where all N emitter subpaths can potentially be merged with the sensor subpath. Figure 2.3 shows an example of such a path, $\mathbf{y}^{(j)}$. Vertex merging happens when any of the vertices of the emitter subpath falls within the merging kernel K_r of a sensor subpath vertex. K_r is a 2D merging kernel with support radius r centered at $z_t^{(i)}$. For simplicity, we assume K_r is the uniform disk kernel: $K_r(z_t^{(i)}, y_{s+1}^{(j)}) = (\pi r^2)^{-1}$ if $||z_t^{(i)} - y_{s+1}^{(j)}|| < r$ and 0 otherwise. Consider $\mathbf{y}^{(j)}$ in Figure 2.3 as an example. Since $y_{s+1}^{(j)}$ falls in the kernel of $z^{(i)}$, vertex merging happens and an extended path $\mathbf{x}^*_{s,t} = y_0^{(j)}...y_s^{(j)}y_{s+1}^{(j)}z_t^{(i)}...z_0^{(i)}$ is generated. Note that $y_{s+1}^{(j)}$ introduces an extra integration dimension over the blurring kernel K_r. For simplicity, we omit superscripts in the following discussions. The vertex merging strategy approximates the path integral with density estimation on all extended paths collected in the merging kernel. The resulting vertex merging estimator is

$$I_{VM}(\mathbf{x}^*_{s,t}) = K_r(z_t, y_{s+1})[\prod_{n=0}^{s} \alpha_L(y_n)]\rho(z_{t-1}, z_t, y_s)[\prod_{n=0}^{t-1} \alpha_E(z_n)] \tag{2.5}$$

$$\alpha_L(y_n) = \begin{cases} \frac{L_e(y_0)G(y_0 \leftrightarrow y_1)}{p(y_0)p(y_1)}, & \text{for } n = 0 \\ \frac{\rho(y_{i-1}, y_i, y_{i+1})G(y_i \leftrightarrow y_{i+1})}{p(y_{i+1})}, & \text{for } n > 0 \end{cases} \tag{2.6}$$

$$\alpha_E(z_n) = \begin{cases} \frac{W_e(z_0)G(z_0 \leftrightarrow z_1)}{p(z_0)p(z_1)}, & \text{for } n = 0 \\ \frac{\rho(z_{i-1}, z_i, z_{i+1})G(z_i \leftrightarrow z_{i+1})}{p(z_{i+1})}, & \text{for } n > 0 \end{cases}, \tag{2.7}$$

where α_L and α_E are the Veach style weights [17] for each vertex in emitter and sensor subpath respectively. The core of VCM is to combine the BDPT (vertex connection) estimator denoted as $I_{VC}(\mathbf{x}_{s,t})$ with the vertex merging estimator $I_{VM}(\mathbf{x}^*_{s,t})$ using multiple importance sampling. To do this, we can think of the extended path $\mathbf{x}^*_{s,t}$ as an approximation of the physically valid path $y_0...y_s z_t...z_0$. The probability of sampling this path is

$$p(\mathbf{x}^*_{s,t}) = p(y_0...y_s z_t...z_0) p_{acc}(\mathbf{x}^*_{s,t}), \tag{2.8}$$

where $p(y_0...y_s z_t...z_0)$ is the probability density of sampling all the vertices of the physically valid path. $p_{acc}(\mathbf{x}^*_{s,t})$ is the approximate probability of the path being accepted. When y_s is not a specular interaction, $p_{acc}(\mathbf{x}^*_{s,t}) = p(y_s \to y_{s+1})\pi r^2$ which approximates the probability that y_{s+1} falls inside the merging kernel by assuming that the probability distribution is uniform inside the kernel, otherwise $p_{acc}(\mathbf{x}^*_{s,t}) = 1$. The final VCM estimator is the combination of the sum of the vertex connection and merging contribution:

$$I_{VCM} = C_{VC} + C_{VM} \tag{2.9}$$

$$C_{VC} = \sum_{s\geq 0}\sum_{t\geq 0} \omega_{VC,s}(\mathbf{x}_{s,t}) f(\mathbf{x}_{s,t}) \Big/ p_{VC,s}(\mathbf{x}_{s,t}) \tag{2.10}$$

$$C_{VM} = \sum_{j=1}^{N}\sum_{s\geq 1}\sum_{t\geq 1} \omega_{VM,s}(\mathbf{x}^{(j)}_{s,t}) I_{VM}(\mathbf{x}^{(j)}_{s,t}). \tag{2.11}$$

Here, the meaning of $f(\mathbf{x}_{s,t})$ and $p_{VC,s}(\mathbf{x}_{s,t})$ are the same as those in Equation 2.4 because the BDPT estimators are inherited. The subscript VC is used to stress that BDPT is called vertex connection in VCM. $\omega_{VC,s}(\mathbf{x}_{s,t})$ is different because vertex merging can potentially sample the same paths that vertex connection samples. $\omega_{VC,s}$ in Equation 2.10 and $\omega_{VM,s}$ in Equation 2.11 can be derived by considering all potential strategies of sampling the same path. That is

$$\omega_{VC,s}(\mathbf{x}_{s,t}) = \frac{[p_{VC,s}(\mathbf{x}_{s,t})]^\beta}{\sum_{n=0}^{k-1}[p_{VC,n}(\mathbf{x}_{s,t})]^\beta + [N p_{VM,n}(\mathbf{x}_{s,t})]^\beta} \tag{2.12}$$

$$\omega_{VM,s}(\mathbf{x}^{(j)}_{s,t}) = \frac{[p_{VM,s}(\mathbf{x}^{(j)}_{s,t})]^\beta}{\sum_{n=0}^{k-1}[p_{VC,n}(\mathbf{x}^{(j)}_{s,t})]^\beta + [N p_{VM,n}(\mathbf{x}^{(j)}_{s,t})]^\beta}. \tag{2.13}$$

In Equation 2.12, the meaning of the numerator and $\sum_{n=0}^{k-1}[p_{VC,n}(\mathbf{x}_{s,t})]^\beta$ in the denominator is identical to those in the expression of $\omega_s(\mathbf{x}_{s,t})$ in Equation 2.4 which takes all connection strategies into account. The extra $[N p_{VC,n}(\mathbf{x}^{(j)}_{s,t})]^\beta$ term in the denominator is to consider all the merging strategies, where $p_{VM,n}(\mathbf{x}_{s,t}) = p_{VC,n}(\mathbf{x}_{s,t}) p_{acc,n}(\mathbf{x}_{s,t})$ is the probability density of sampling $\mathbf{x}_{s,t}$ through merging at x_{s+1} and $p_{acc,n}(\mathbf{x}_{s,t})$ is the acceptance probability. Note that $p_{acc,0}(\mathbf{x}_{s,t}) = p_{acc,k-1}(\mathbf{x}_{s,t}) = 0$ because we do not merge at the sensor or the light source. Factor N takes into account the fact that we use this strategy for all N emitter subpaths. Equation 2.13 can be interpreted in a similar way, where the denominator is the powered sum of the probability of all connection and merging strategies and the numerator is the powered probability of the strategy being used.

2.4 Gradient-Domain Bidirectional Path Tracing

G-BDPT extends BDPT by generating correlated samples for neighboring pixels using pairs of emitter $\mathbf{y}$ and sensor $\mathbf{z}$ subpaths[13]. These samples are used to obtain a correlated estimator of the path integrals of the pixel's neighbors, and to estimate gradients by finite differencing. To do so, a deterministic and reversible shift mapping T_{ij}, which we discuss in depth shortly, is first applied to the sensor subpath $\mathbf{z}$ to obtain an offset sensor subpath $\mathbf{z}^{\text{off}} = T_{ij}(\mathbf{z})$ (Figure 2.4), where we assume i and j are indices of neighboring pixels. For efficiency, G-BDPT classifies all vertices as connectable or unconnectable, with a roughness threshold; specifically, if a vertex's material roughness is too large, it is connectable, otherwise not. Unlike BDPT, G-BDPT only connects vertex pairs if both of them are connectable. With this classification, both the base and offset sensor subpaths are connected to the emitter subpath $\mathbf{y}$ to construct full paths for both pixels. Given a constructed base path $\mathbf{x}_{s,t}$ and its corresponding offset path as $\mathbf{x}^{\text{off}}_{s,t}$, pixel i's path integral estimator is $I_i = f(\mathbf{x}_{s,t}) \big/ p_s(\mathbf{x}_{s,t})$ and pixel j's correlated path integral estimator is $I_j = f(\mathbf{x}^{\text{off}}_{s,t}) \big/ p_s(\mathbf{x}_{s,t}) |T'_{ij}(\mathbf{x}_{s,t})|$, where $|T'_{ij}(\mathbf{x}_{s,t})| = \left| \partial z^{\text{off}}_0 ... \partial z^{\text{off}}_t \big/ (\partial z_0 ... \partial z_t) \right|$ is the determinant of the Jacobian of the shift mapping for the change of integration variables. $I_j - I_i$ can be used to estimate the gradient, however this estimator can be biased since the mapping does not guarantee that pixel j's full integration domain is covered. To solve this, the correlated estimator is combined with its neighbor's base path estimator using MIS. And so, the contribution to the full gradient estimator is

$$C_{ij} = \sum_{s \geq 0} \sum_{t \geq 0} \omega_{ij,s}(\mathbf{x}_{s,t}) \frac{f_j(\mathbf{x}^{\text{off}}_{s,t}) |T'_{ij}(\mathbf{x}_{s,t})| - f_i(\mathbf{x}_{s,t})}{p_s(\mathbf{x}_{s,t})}, \tag{2.14}$$

where

$$\omega_{ij,s}(\mathbf{x}_{s,t}) = \frac{p_s(\mathbf{x}_{s,t})^\beta}{\sum_{k=0}^{s+t} [p_k(\mathbf{x}_{s,t})]^\beta + [p_k(\mathbf{x}^{\text{off}}_{s,t}) |T'_{ji}(\mathbf{x}^{\text{off}}_{s,t})|]^\beta} \tag{2.15}$$

is the MIS weight. The first term in the denominator considers all connection strategies of the base path. The second term in the denominator takes the reverse mapping into account. That is, the same gradient can also be sampled through pixel j. If pixel j sampled $\mathbf{x}^{\text{off}}_{s,t}$ for

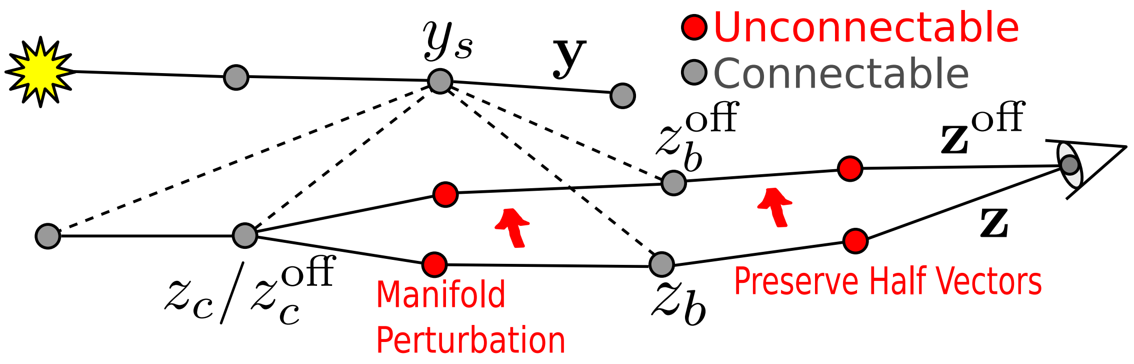

Figure 2.4: G-BDPT paths.

the base path, then the same gradient sample would be generated through inverse mapping $T_{ji}(\mathbf{x}^{\text{off}}_{s,t})$. Note that the full gradient estimator is from the contributions of 2 strategies C_{ij} and C_{ji}. That is $g_{ij} = I_j - I_i = C_{ij} - C_{ji}$.

Shift Mapping Sensor Sub Path

As illustrated in Figure 2.4, assume the first and second connectable vertices along the sensor subpath $\mathbf{z}$ are z_b and z_c respectively. The offset path can be constructed as follows. First, a sensor ray from the neighbor pixel with the same offset as that of the base pixel is initiated. Then, we recursively trace this ray to find vertices $z^{\text{off}}_0 z^{\text{off}}_1 ... z^{\text{off}}_b$ that correspond to $z_0 z_1 ... z_b$. If the ray hits an unconnectable vertex, we deterministically trace the recursive ray such that the vertex shares the same half vector as its base vertex. Then, we construct $z^{\text{off}}_b ... z^{\text{off}}_c$ such that $z^{\text{off}}_c = z_c$, which is done by either a simple connection if z_c is z_b's successor or a manifold perturbation[7] preserving half vectors otherwise. Here, the half vector of a vertex is the normalized average vector of the incoming and outgoing ray direction vectors. The BRDF value of many specular or glossy materials is only determined by the half vector. This way, by preserving half vectors on unconnectable vertices, high correlation between the offset and base paths can be obtained in G-BDPT. Manifold perturbation is a method to construct a new path from a given path such that the end vertex of the new path is at a target location and all half vectors of its unconnectable predecessors are preserved.

Chapter 3

Overview

We present our method G-VCM in this section. Algorithm 1 outlines a single G-VCM iteration, where the pipeline is similar to that of VCM, with additional gradient sampling and evaluation steps.

At each iteration, we first sample and store all emitter subpaths and build a kd-tree for their connectable vertices to perform fast queries (lines 1-8). For each pixel, we sample and shift map a sensor subpath to generate four offset sensor subpaths for its neighbors (lines 10-15). We evaluate primal and gradient contributions with connection strategies using the base sensor subpath, its four offset paths and emitter subpaths that correspond to the pixel, similarly to G-BDPT (see Section 2.4; lines 16-18). The only difference here is that, to combine with our new vertex merging strategies, we need to modify the MIS weights previously used for G-BDPT. We introduce these modifications in Chapter 5.

Next, we perform vertex merging by looking up emitter vertices inside the merging kernel of each connectable vertex along the sensor subpath (lines 19-24). We obtain the primal contribution of vertex merging with standard VCM (see Section 2.3). A major challenge here is how to generalize the original vertex merging to gradient sampling and how to combine it with vertex connection strategies. We introduce our gradient-domain vertex merging strategy in Chapter 4 and show how to compute MIS weights for vertex merging to combine with vertex connection in Chapter 5.

Algorithm 1 G-VCM Iteration

```
1: Initialize empty vertex array $Y$
   ▷ For collecting connectable vertices.
2: for pixel $i$ in all pixels do
3:     Sample and cache emitter subpath $\mathbf{y}_i$
4:     for connectable vertex $y$ in $\mathbf{y}_i$ do
5:         Append $y$ to $P$
6:     end for
7: end for
8: Build kd-tree index for $Y$
9: for pixel $i$ in all pixels do
10:    $\mathbf{y} \leftarrow \mathbf{y}_i$                                   ▷ Retrieve $\mathbf{y}_i$
11:    Sample sensor subpath $\mathbf{z}$
12:    Initialize empty offset sensor subpaths $\mathbf{z}^{\text{off}} = \{\mathbf{z}_1^{\text{off}}, ..., \mathbf{z}_4^{\text{off}}\}$
13:    for pixel $j$ in pixel $i$'s neighbors do
14:        $\mathbf{z}_j^{\text{off}} \leftarrow T_{ij}(\mathbf{z})$                  ▷ Sensor subpath shift mapping
15:    end for
16:    for strategy $(s, t)$ in all connection strategies do
17:        $\mathit{EvaluateConnection}(\mathbf{y}, \mathbf{z}, \mathbf{z}^{\text{off}}, s, t)$
   ▷ G-BDPT contribution
   ▷ with modified MIS weights in Section 5.
18:    end for
19:    for connectable vertex $z_t$ in $\mathbf{z}$ do
20:        for emitter vertex $y \in Y$ in $z_t$'s merging kernel do
21:            $\mathbf{y}, s \leftarrow$ GetPathAndPredecessorIndex($y$)
22:            $\mathit{EvaluateMerging}(\mathbf{y}, \mathbf{z}, \mathbf{z}^{\text{off}}, s, t)$
   ▷ Gradient-domain vertex merging in Section 4
   ▷ with MIS weights in Section 5.
23:        end for
24:    end for
25: end for
```

Chapter 4

Gradient Domain Vertex Merging

We will generalize the vertex merging strategy in VCM to gradient sampling. We use the same threshold as described in Section 2.4 to decide whether a vertex is connectable and only perform vertex merging at connectable vertices. In this way, two scenarios arise, depending on where merging happens as illustrated in Figure 4.1: either occurs at or after z_c (case 1), or at z_b (case 2).

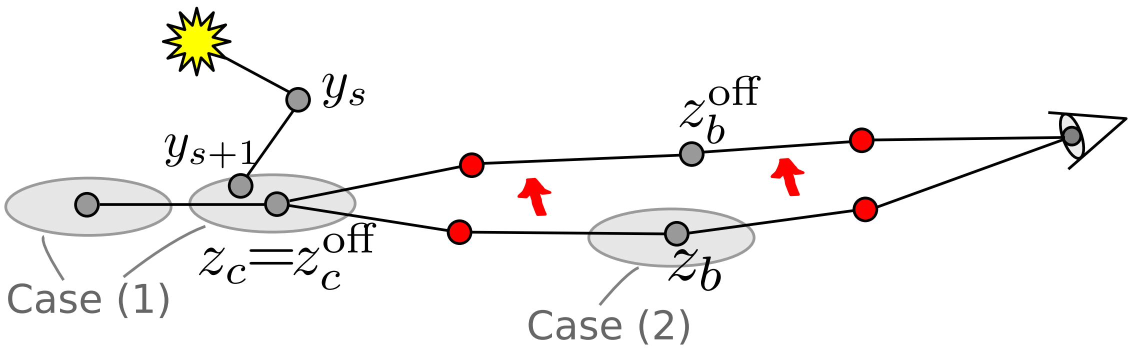

Figure 4.1: Gradient-domain vertex merging cases.

Recall that z_b and z_c are the first and second connectable vertices along the sensor subpath as in Section 2.4. In both cases, we use the same vertex merging estimator I_{VM} in Equation 2.5 for the base path. Note that we only illustrate an example emitter subpath for case (1) in Figure 4.1 and case (2) is discussed in separate figures.

Next, we describe how to generate the correlated estimator for the offset path in each case. Case (1) is simpler than case (2) because the base and offset sensor subpaths share the same emitter subpath and merging kernel. So let's deal with case (1) first. Assume merging happens at sensor subpath vertex $z_t (t \geq c)$. We consider how Equation 2.5 should change to obtain the correlated estimator from the offset path. All α_L should stay the same since all emitter vertices are sampled in the same way. The BRDF term $\rho(z_{t-1}, z_t, y_s)$ should be $\rho(z_{t-1}^{\text{off}}, z_t, y_s)$. Note that z_{t-1}^{off} and z_{t-1} are the same vertex when $t > c$. For α_E, the measurement values should be from the corresponding offset vertices instead and

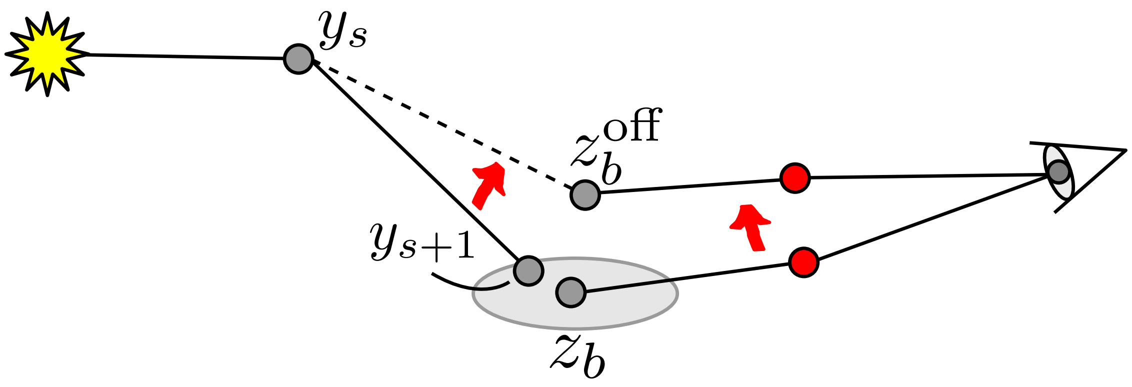

Figure 4.2: The case where y_s is connectable.

the probability should stay the same. To account for the change of variables in the shift mapping, we also need to multiply the determinant of the Jacobian. Putting them together, we have

$$I^{\text{off}}_{VM}(\mathbf{x}^*_{s,t}) = U(\mathbf{x}^*_{s,t})\rho(z^{\text{off}}_{t-1}, z_t, y_s)[\prod_{n=0}^{t-1} \alpha^{\text{off}}_E(z_n)]|T'_{ij}(\mathbf{z})|, \tag{4.1}$$

where $U(\mathbf{x}^*_{s,t}) = K_r(z_t, y_{s+1})[\prod_{n=0}^{s} \alpha_L(y_n)]$ is common for base and offset sensor subpaths, $|T'_{ij}(\mathbf{z})|$ is the determinant of the Jacobian of shift mapping and

$$\alpha^{\text{off}}_E(z_n) = \begin{cases} \frac{W_e(z_0^{\text{off}})G(z_0^{\text{off}} \leftrightarrow z_1^{\text{off}})}{p(z_0)p(z_1)}, & \text{for } n = 0 \\ \frac{\rho(z^{\text{off}}_{i-1}, z^{\text{off}}_i, z^{\text{off}}_{i+1})G(z^{\text{off}}_i \leftrightarrow z^{\text{off}}_{i+1})}{p(z_{i+1})}, & \text{for } n > 0 \end{cases} \tag{4.2}$$

are the weights for offset sensor subpaths. Note that for $t \geq c$, z_t and z^{off}_t are the same vertex.

In case (2), the major problem is that, since z_b and z^{off}_b do not share the same merging kernel, it is unclear how we can generate the correlated estimator under the density estimation formulation in Equation 2.5. To solve this problem, we propose to construct a physically valid full path as the offset path and treat the sampling process as a probabilistic connection. Assume the base emitter path and sensor path are $\mathbf{y}$ and $\mathbf{z}$ respectively and the offset sensor subpath is $\mathbf{z}^{\text{off}}$. Depending on the material property of y_s, our gradient sampling method is split into the two cases as follows.

Scenario: y_s is connectable

In this case, we simply connect y_s and z^{off}_b to form the offset path $\mathbf{x}^{\text{off}} = y_0...y_s z^{\text{off}}_b...z^{\text{off}}_0$ as in Figure 4.2. The measurement contribution $f(\mathbf{x}^{\text{off}})$ can then be evaluated for the offset path. Assume that the extended base path is $\mathbf{x}^* = y_0...y_s y_{s+1} z_b...z_0$. Then the probability density of sampling this path is

$$p^{\text{off}}(\mathbf{x}^*) = p(y_0...y_s z_b...z_0)p(y_{s+1})\pi r^2 J_z, \tag{4.3}$$

which is the probability of sampling and accepting the base path, multiplied by the shift mapping's determinant of the Jacobian from z_0 to z_b, $J_z = \left|\partial z_0 ... \partial z_b \middle/ (\partial z_0^{\text{off}} ... \partial z_b^{\text{off}})\right|$. This way, the offset estimator is simply $I_{VM}^{\text{off}} = f(\mathbf{x}^{\text{off}}) \big/ p^{\text{off}}(\mathbf{x}^*)$.

Scenario: y_s is unconnectable

In this case, we can still use simple connection to obtain consistent estimators. However, since y_s is specular or highly glossy, the connected offset path is likely to have zero contribution, making the gradient estimator noisy. To solve this, as illustrated in Figure 4.3, we utilize manifold perturbation [7] to obtain the offset path in this case. Assume that the closest connectable predecessor of y_s is y_d. We perturb y_{s+1} to $y_{s+1}^{\text{off}} = z_b^{\text{off}}$ such that half vectors of vertices in between y_d and y_{s+1} are preserved. For the mapping to be reversible, we also check if y_{s+1}^{off} can be perturbed back to y_{s+1}. Note that this reversibility is biased, but converges consistently as the merging kernel $r \to 0$ progressively. If the reversibility test fails, we set the offset path's contribution to 0, otherwise, a full offset path $\mathbf{x}^{\text{off}} = y_0 ... y_d y_{d+1}^{\text{off}} ... y_s^{\text{off}} z_b^{\text{off}} ... z_0^{\text{off}}$ can be constructed and its measurement value can be evaluated as $f(\mathbf{x}^{\text{off}})$. Next, we derive $p^{\text{off}}(\mathbf{x}^*)$, the probability of the offset path being sampled. To do this, we observe that any base chain $y_d ... y_{s+1}$ with the same half vectors for all unconnectable vertices will be perturbed to the same offset path as illustrated in Figure 4.4. Let's denote the half vector offset space of the base and the offset chain as follows:

$$\{y_{d+1} ... y_s\} \equiv \{o_{d+1} ... o_s\}$$

$$\{y_{d+1}^{\text{off}} ... y_s^{\text{off}}\} \equiv \{o_{d+1}^{\text{off}} ... o_s^{\text{off}}\},$$

where $o_{d+1} ... o_s$ correspond to the half vectors of $y_{d+1} ... y_s$ and $o_{d+1}^{\text{off}} ... o_s^{\text{off}}$ correspond to the half vectors of $y_{d+1}^{\text{off}} ... y_s^{\text{off}}$. Then, we arive at the following equation for p^{off}:

$$p^{\text{off}}(\mathbf{x}^*) = p(z_b ... z_0) J_z p(y_0 ... y_{s+1}) J_y \pi r^2, \tag{4.4}$$

where J_z is the same determinant in Equation 4.3 that accounts for the change of variables in the sensor subpath, πr^2 normalizes the merging kernel K_r at z_b and $J_y = \left|\partial y_{d+1} ... \partial y_s \middle/ (\partial y_{d+1}^{\text{off}} ... \partial y_s^{\text{off}})\right|$ accounts for the change of variables from $y_{d+1} ... y_s$ to $y_{d+1}^{\text{off}} ... y_s^{\text{off}}$.

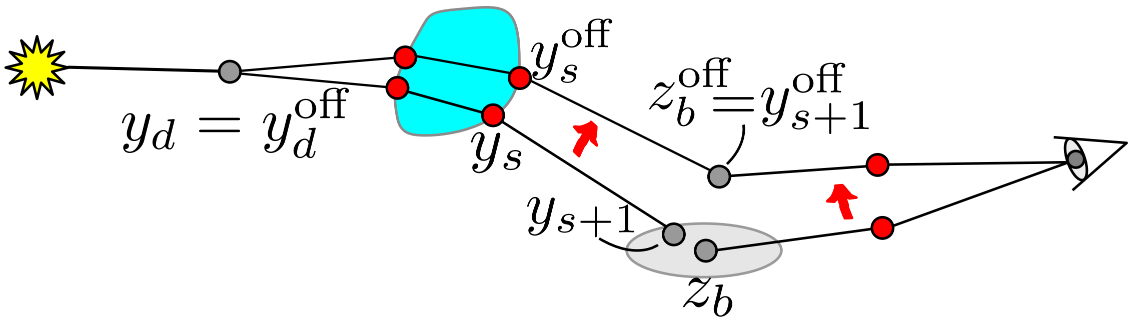

Figure 4.3: The case where y_s is unconnectable.

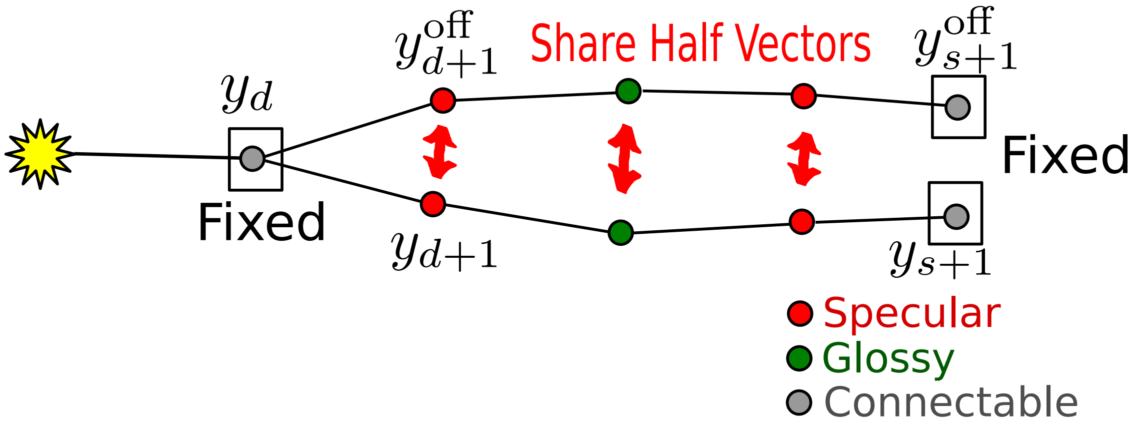

Figure 4.4: An example of a specular/glossy chain.

To derive J_y we use the half vector offset space as an intermediate representation, similarly to G-MLT [12]. That is

$$J_y = \left|\frac{\partial y_{d+1} \ldots \partial y_s}{\partial o_{d+1} \ldots \partial o_s}\right| \left|\frac{\partial o_{d+1} \ldots \partial o_s}{\partial o^{\text{off}}_{d+1} \ldots \partial o^{\text{off}}_s}\right| \left|\frac{\partial y^{\text{off}}_{d+1} \ldots \partial y^{\text{off}}_s}{\partial o^{\text{off}}_{d+1} \ldots \partial o^{\text{off}}_s}\right|^{-1}, \tag{4.5}$$

where the middle term evaluates to 1 because the Jacobian $\partial o_{d+1} \ldots \partial o_s \big/ (\partial o^{\text{off}}_{d+1} \ldots \partial o^{\text{off}}_s)$ is an identity matrix by nature of the manifold perturbation. The left and right terms can be obtained from the derivative matrices of the half vector manifold exploration constraint functions [7]. We reuse the open source Mitsuba implementation of this component [6], which takes into account the case where the chain comprises glossy and specular vertices (by omitting certain rows and columns in the matrices).

Given $p^{\text{off}}(\mathbf{x}^*)$, the offset pixel's path integral is $I^{\text{off}}_{VM} = \frac{f(\mathbf{x}^{\text{off}})}{p^{\text{off}}(\mathbf{x}^*)}$.

Chapter 5

MIS for G-VCM

In Chapter 4, we derived how to estimate a neighboring pixel's integral using vertex merging. To robustly sample gradients, we need to combine all estimators from vertex connection and vertex merging using MIS. Assume a pair of gradient base path $\mathbf{x} = x_0 ... x_n$ and its offset $\mathbf{x}^{\text{off}} = x_0^{\text{off}} ... x_n^{\text{off}}$ is generated using strategy VC or VM. Its corresponding gradient estimator is $g_{VC/VM} = I_{VC/VM} - I_{VC/VM}^{\text{off}}$. To obtain MIS weight for this estimator, we consider all potential strategies to sample the same path pair. Depending on which pixel initiates the paths, there are 2 basic categories. One possibility is that pixel i sampled $\mathbf{x}$ and then suggested $\mathbf{x}^{\text{off}}$ for pixel j. In this case, path $\mathbf{x}$ can be sampled by connecting or merging between every 2 consecutive vertices. We call this category active strategies. The powered sum of all active strategies' pdf is

$$p_{act}(\mathbf{x}) = \sum_{i=0}^{n-1} [p_{VM,i}(\mathbf{x})]^{\beta} + [N p_{VC,i}(\mathbf{x})]^{\beta}. \tag{5.1}$$

The other possibility is that pixel j sampled $\mathbf{x}^{\text{off}}$ and suggested $\mathbf{x}$ for pixel i. In this case, path $\mathbf{x}^{\text{off}}$ can also be sampled by connecting or merging between every 2 consecutive vertices. We call this category passive strategies. For passive strategies to compare with active strategies, we need to multiply each of passive strategies' pdf with their corresponding reverse mapping Jacobian to convert them to base path's area probability density. The powered sum of all passive strategies' converted pdf is

$$p_{pas}(\mathbf{x}^{\text{off}}) = \sum_{i=0}^{n-1} ([p_{VM,i}(\mathbf{x}^{\text{off}})]^{\beta} + [N p_{VC,i}(\mathbf{x}^{\text{off}})]^{\beta}) |T'_{ji}(\mathbf{x}^{\text{off}})|^{\beta}. \tag{5.2}$$

In summary, the MIS weight is the powered proportion of the strategy being used to all potential active and passive strategies:

$$\omega_{VC/VM,s}(\mathbf{x}, \mathbf{x}^{\text{off}}) = \frac{[p_{VC/VM,s}(\mathbf{x})]^{\beta}}{p_{act}(\mathbf{x}) + p_{pas}(\mathbf{x}^{\text{off}})}. \tag{5.3}$$

Chapter 6

Implementation Details

We implemented our method on top of the publicly available G-BDPT implementation in the Mitsuba renderer[6] and followed the pipeline in Algorithm 1.

For each sensor subpath, we decide the merging kernel radius of the first connectable vertex using ray differentials such that the size of the merging kernel projected onto the sensor is about the same as the pixel filter size. We use this kernel size to prevent the smoothing effects of the merging kernel and gradient reconstruction from conflicting with each other.

Adaptive Merging Kernel We find that using the same kernel size along a sensor subpath is detrimental to both VCM and G-VCM since, unlike connection estimators, merging estimators are highly correlated (especially at "deeper" vertices along the sensor subpath). This correlation corrupts the MIS weights, which assume independent estimators. Given this, we shrink the initial merging radius based on vertex material roughness along the sensor subpath: for sensor subpath vertex z_t (on $\mathbf{z}$), we use a kernel radius of $r_t = 0.5^{10\tau(z_{t-1})} r_{t-1}$, where $\tau(z_{t-1})$ is the material roughness at vertex z_{t-1}. Note that we estimate an initial merging radius r_1 using ray differentials. Figure 6.1 shows the impact of adapting the kernel merging radius in VCM: bright speckles on the wall are caused by merges that occur after the first connectable vertex of the sensor subpath. We smooth these speckles out using an adaptive kernel.

Reconstruction As with previous work, any reconstruction technique can be used to combine primal and gradient estimation obtained from our method. We tested L1 and L2 reconstructions, and the recent control variate-based reconstruction technique [16] can also be used given additional variance statistics.

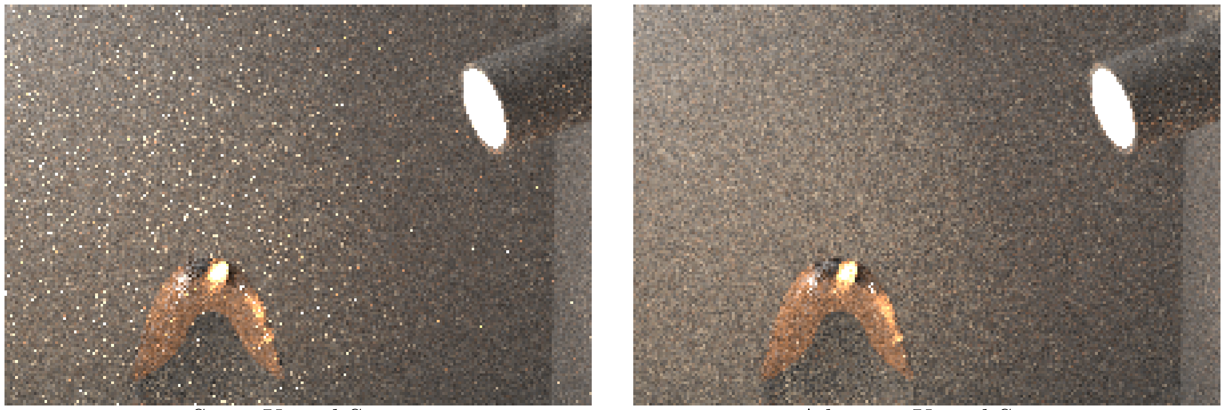

Figure 6.1: Comparison between with and without adaptive kernel in VCM with 16 iterations.

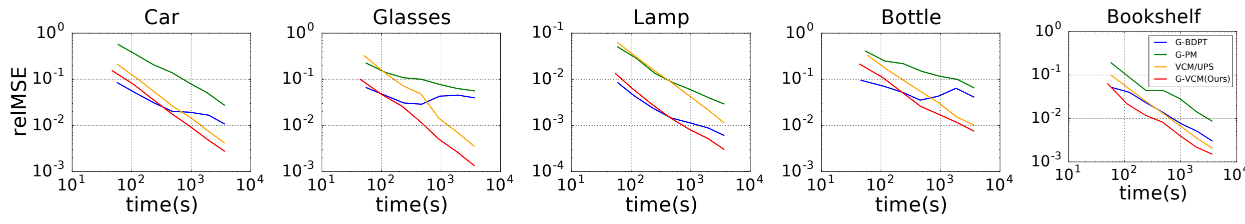

Figure 6.2: Error (relMSE) plots of all 5 test scenes, comparing G-BDPT, G-PM, G-VCM using L1 reconstruction and VCM.

Chapter 7

Results and Discussions

We compare our method to VCM, G-BDPT and the concurrent work G-PM, all implemented in the Mitsuba renderer [6]. Since the code for G-PM was not readily available at the time, we implemented G-PM ourselves. For our G-PM implementation, we only use vertex merging at the first connectable sensor subpath vertex and completely take out connection strategies in G-VCM. To obtain consistent convergence, the MIS weights introduced in Chapter 5 should be modified such that the pdf of all strategies not being used is set to 0. Our G-PM implementation is not identical to the original G-PM, but we show strong similarities between the two in Section 7.2. The subtle difference should not change the effectiveness of the algorithm.

All of our experiments are done on a desktop with 12-core Intel i7-5930K 3.5GHz processor and 20GB memory. Our VCM implementation takes about twice as much time per iteration compared to the original BDPT implementation due to the photon gathering overhead. For the same reason, our method has about 60% overhead per iteration compared to G-BDPT. We only show L1 reconstruction for comparison in this section and pick the reconstruction parameter $\alpha = 0.3$ for all gradient-domain methods. We use a more conservative α than used in previous works because our scenes involve many glossy objects. We set the BRDF roughness threshold to 0.01 and merging kernel radius reduction ratio to 0.9 for all methods. Full resolution images of all methods using both L1 and L2 reconstruction are provided in our supplemental materials.

We use five test scenes: Glasses, Lamp, Bottle, and Bookshelf (Figure 7.4) and Car (Figure 1.1). In Bottle and Lamp, we modified the original scene to include SDS paths and glossy reflections. As with previous gradient-domain methods, we use relMSE = average$[(X - R)^2/(R^2 + 0.001)]$, where R is the reference pixel color and X is the estimated image color. We discard the 50 pixels with highest error due to the corruption from strong light paths that reach the sensor through specular-only interactions.

Figure 6.2 shows the relative mean squared error (relMSE) convergence curve for all scenes. We can see that our method has lower relMSE than VCM in all test scenes. In Glasses and Bottle, G-BDPT has almost flat convergence due to SDS paths that it fails to efficiently capture. In Car, Bookshelf and Lamp where the contribution of SDS paths is relatively small, G-BDPT has less error than our method initially due to its smaller overhead, but it slows down significantly and gets surpassed by our method after a few minutes when SDS paths become its bottleneck. On the other hand, although G-PM is very good at handling SDS paths, the overall image quality is hardly competitive due to the large contribution from multiple diffuse/glossy interactions in our test scenes. In comparison, our method has robust convergence and outperforms VCM, G-BDPT and G-PM in the long run.

In Figure 1.1 and Figure 7.4, we show equal time comparison of our test scenes. Thanks to the denoising power of gradients, our method achieves significant noise reduction compared to VCM. Compared with gradient-domain methods, our method robustly captures all potential light paths across every scene. G-BDPT has difficulty in sampling SDS paths, such as the interior of the Car and the caustics from the bottle (seen through the goblet) in Bottle. Here, our method is able to sample these paths well using vertex merging strategies. Although G-PM is also able to capture SDS paths, it generally fails on glossy surfaces (as its primal domain counterpart). Examples of this case include the glossy reflection of the lamp in Bookshelf and of the table in Glasses. Here, our method is able to rely more on the more efficient vertex connection strategy. Overall, our method demonstrates all-round performance improvements by leveraging the best of both vertex connection *and* merging schemes, where techniques that rely exclusively on connection *or* merging can have difficulty sampling certain transport paths.

7.1 Limitations

Although our method is generally more robust than G-PM and G-BDPT, there are still certain types of light paths that are hard to capture even with G-VCM. For example, in the first row of insets for Lamp, the yellowish glossy caustics near the bottom of the glass egg are poorly captured by all methods. This is because neither vertex connection nor vertex merging is efficient at handling specular-glossy interactions. In addition, our method inherits the low frequency image assumption as with all current gradient-domain rendering techniques. Therefore, G-VCM may perform worse than VCM in scenes with very rich high frequency details.

7.2 Comparison with G-PM

In this section, we illustrate the similarities and differences between case (2) of the vertex merging part of our method introduced in Chapter 4 and the concurrent work G-PM [5]. In general, G-PM shift maps the base photon from the base sensor vertex's kernel to the offset sensor vertex's kernel, while our method directly shift maps the base photon to the offset sensor vertex. Both methods apply different shift mapping depending on the connectability of a photon's predecessor.

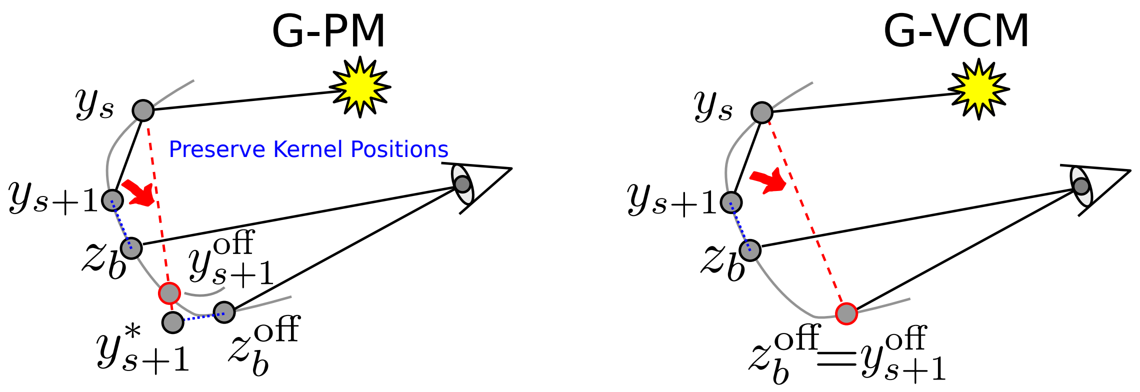

Figure 7.1: G-PM and vertex merging of G-VCM in connectable predecessor case.

In the connectable predecessor case as shown in Figure 7.1, G-PM first decides the offset photon's position y^*_{s+1} within the offset sensor vertex z_b^{off}'s kernel such that the base and offset photon positions in the local frames of their kernels are preserved. Then, a ray from y_s to y^*_{s+1} is fired to find the physical intersection point y^{off}_{s+1} in the scene, which is then used as the offset photon for the offset kernel's density estimation. In this case, our method directly connects y_s and z_b^{off} and computes the offset path's contribution as introduced in Chapter 4.

In the unconnectable predecessor case as shown in Figure 7.2, G-PM first finds the intersection point y^{off}_{s+1} in the scene following the same procedure as in the connectable predecessor case. Then, a manifold perturbation is performed to move the base photon vertex y_{s+1} to the offset photon vertex y^{off}_{s+1}, preserving half vectors of preceding unconnectable vertices. This way, an offset emitter sub-path can be constructed and used in density estimation. Finally, to check reversibility, G-PM tests the visibility between z_b and y_s^{off}. If the visibility test fails, it suggests that the shift mapping is not reversible and the offset path's contribution is set to zero. In our case, we directly apply a manifold perturbation to move the photon vertex y_{s+1} to the offset sensor vertex z_b^{off} and estimate the offset path's contribution as introduced in Chapter 4.

The behaviors of these two methods become increasingly similar as the merging kernel size progressively shrinks to 0. As mentioned in Chapter 6, we use ray differentials to initialize the merging kernel radius of the first connectable vertex. The same strategy is also used in G-PM. This suggests that the initial merging radius is very small so that the behaviors of both methods in this case are very similar from the beginning.

Despite similar asymptotic behavior, the power of G-VCM comes from our probabilistic connection formulation, which allows us to easily combine gradient-domain vertex merging with other complementary strategies (as in Algorithm 1). This combination is the key to the increased robustness of our algorithm compared to previous work. In contrast, G-PM only performs density estimation once at the first connectable vertex, resulting in robustness issues in many scenarios (see Chapter 7).

In figure 7.3, we show a full image comparison between G-PM and G-VCM in the Bottle scene. The scene settings and run time are the same as in figure 7.4. Here, we use L2

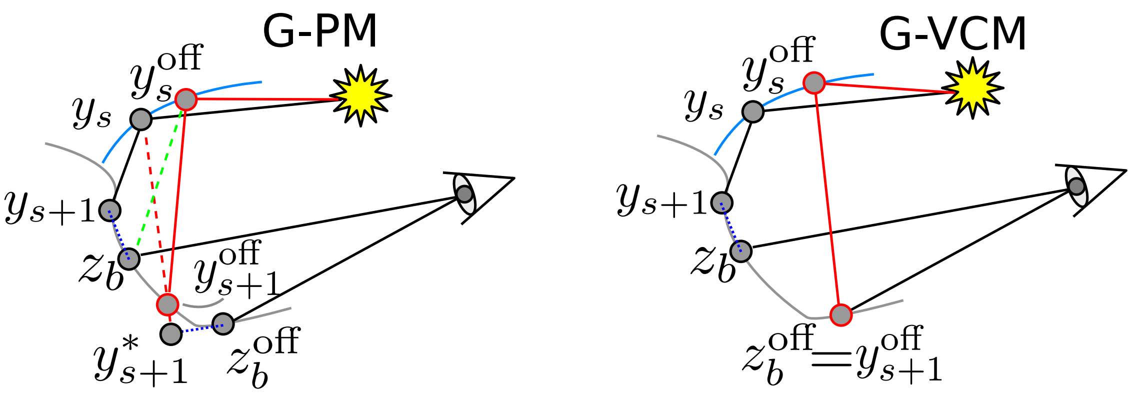

Figure 7.2: G-PM and vertex merging of G-VCM in unconnectable predecessor case.

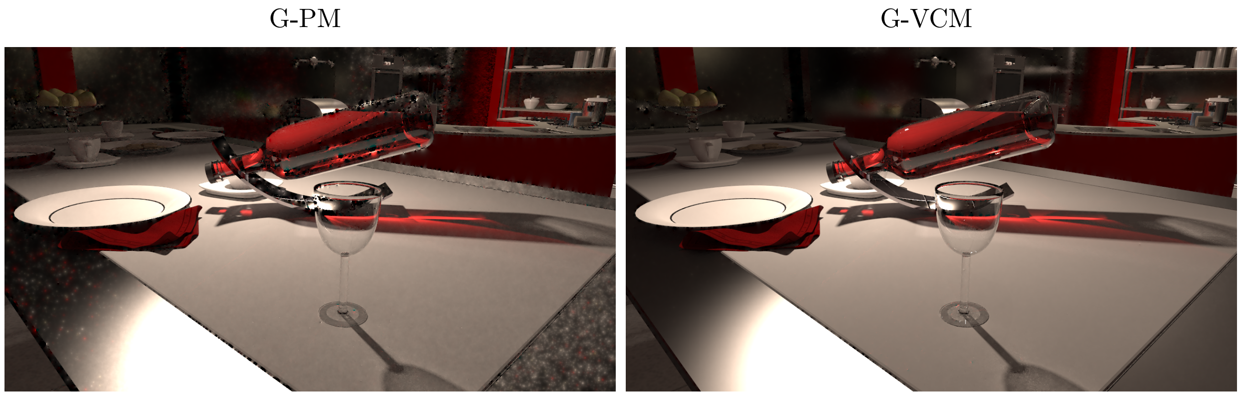

Figure 7.3: Full image comparison between G-PM and G-VCM using L2 reconstruction with the same run time. Scene settings are the same as in Figure 7.4 where we show insets of L1 reconstruction results.

reconstruction in both methods in order to highlight the quality in gradient estimation. As can be seen from the images, G-PM struggles to obtain high quality gradients on specular surfaces due to low photon density. In comparison, our method achieves much better gradient estimation overall by leaning more on vertex connection and alternative vertex merging strategies when necessary.

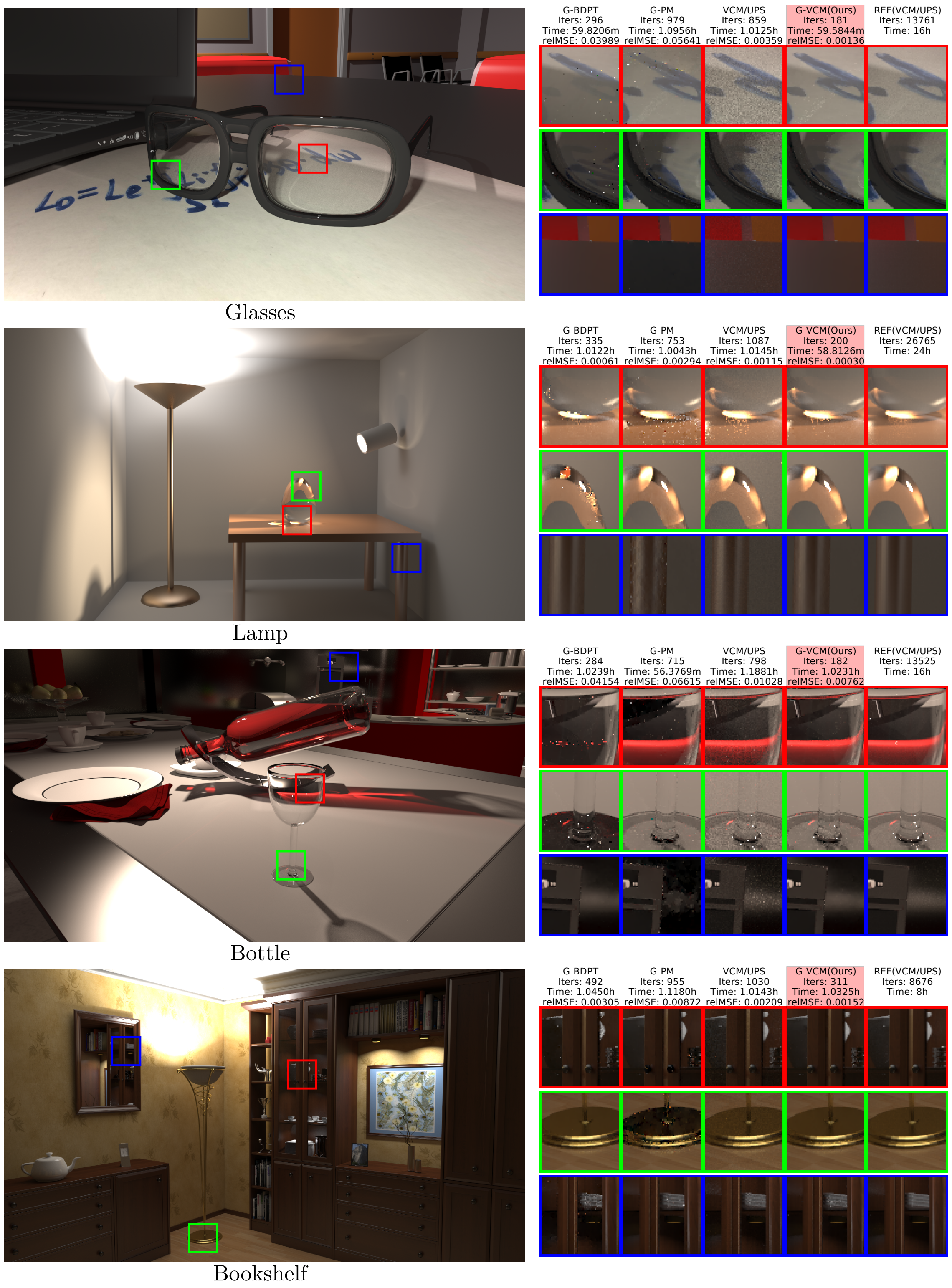
G-BDPT Iters: 296 Time: 59.8206m relMSE: 0.03989
G-PM Iters: 979 Time: 1.0956h relMSE: 0.05641
VCM/UPS Iters: 859 Time: 1.0125h relMSE: 0.00359
G-VCM(Ours) Iters: 181 Time: 59.5844m relMSE: 0.00136
REF(VCM/UPS) Iters: 13761 Time: 16h
Glasses
G-BDPT Iters: 335 Time: 1.0122h relMSE: 0.00061
G-PM Iters: 753 Time: 1.0043h relMSE: 0.00294
VCM/UPS Iters: 1087 Time: 1.0145h relMSE: 0.00115
G-VCM(Ours) Iters: 200 Time: 58.8126m relMSE: 0.00030
REF(VCM/UPS) Iters: 26765 Time: 24h
Lamp
G-BDPT Iters: 284 Time: 1.0239h relMSE: 0.04154
G-PM Iters: 715 Time: 56.3769m relMSE: 0.06615
VCM/UPS Iters: 798 Time: 1.1881h relMSE: 0.01028
G-VCM(Ours) Iters: 182 Time: 1.0231h relMSE: 0.00762
REF(VCM/UPS) Iters: 13525 Time: 16h
Bottle
G-BDPT Iters: 492 Time: 1.0450h relMSE: 0.00305
G-PM Iters: 955 Time: 1.1180h relMSE: 0.00872
VCM/UPS Iters: 1030 Time: 1.0143h relMSE: 0.00209
G-VCM(Ours) Iters: 311 Time: 1.0325h relMSE: 0.00152
REF(VCM/UPS) Iters: 8676 Time: 8h
Bookshelf

Figure 7.4: Equal time comparison between our method, G-BDPT, G-PM and VCM with L1 reconstruction for all gradient domain methods.

Chapter 8

Conclusions and Future Work

Robustness and noise reduction are two major long standing challenges in light transport simulation. VCM was a leap forward in improving robustness through unifying both BDPT and PM. Despite the success in alleviating robustness issues, VCM still generates images with noticeable noise at low sampling rates. On the other hand, gradient domain methods (G-MLT, G-PT, G-BDPT and G-PM) are proposed to extend their primal domain counterparts by adding correlated samples. These methods are able to reduce image noise significantly compared to primal domain methods through a screened Poisson reconstruction of both the primal and gradient samples. However, they inherit robustness issues from their primal domain predecessors because the primal samples are driven by the same set of techniques.

In this thesis, we propose G-VCM, a new method to utilize the advantages of VCM in the gradient domain context in order to achieve the best of both robustness and noise reduction. The core of our method is a set of novel correlated vertex merging sampling techniques which can be easily combined with correlated vertex connection (G-BDPT) sampling techniques via MIS. By adding vertex merging techniques on top of G-BDPT, our method is able to robustly sample gradients in scenes with both multiple diffuse/glossy interactions and specular-diffuse-specular paths, which are challenging for both G-BDPT and G-PM. Through this robust gradient sampling unification, we are also able to achieve significant noise reduction and lower RMSE error compared to VCM at the same time. As we have demonstrated in Chapter 7, our method has robust convergence in scenes with multiple difficult light path types, where both G-PM and G-BDPT have slow convergence.

For future work, we believe that an asymptotic error analysis of the progressive kernel shrinkage strategy for G-VCM similar to the one in progressive photon mapping [3] will provide some useful insights. The challenge here is how to generalize the primal domain analysis with the added complexity from gradient sampling. Another promising avenue is to add Markov chain Monte Carlo strategies to VCM in the gradient domain context following the primal domain work [20].

Bibliography

[1] Iliyan Georgiev et al. “Light Transport Simulation with Vertex Connection and Merging”. In: *ACM Trans. Graph.* 31.6 (Nov. 2012), 192:1–192:10. ISSN: 0730-0301.

[2] Toshiya Hachisuka and Henrik Wann Jensen. “Stochastic Progressive Photon Mapping”. In: *ACM Trans. Graph.* 28.5 (Dec. 2009), 141:1–141:8. ISSN: 0730-0301.

[3] Toshiya Hachisuka, Shinji Ogaki, and Henrik Wann Jensen. “Progressive Photon Mapping”. In: *ACM Trans. Graph.* 27.5 (Dec. 2008), 130:1–130:8. ISSN: 0730-0301.

[4] Toshiya Hachisuka, Jacopo Pantaleoni, and Henrik Wann Jensen. “A Path Space Extension for Robust Light Transport Simulation”. In: *ACM Trans. Graph.* 31.6 (Nov. 2012), 191:1–191:10. ISSN: 0730-0301.

[5] B.-S. Hua et al. “Gradient-Domain Photon Density Estimation”. In: *Euorgraphics 2017* (2017). URL: `https://beltegeuse.github.io/research/publication/2017_GradientPM/`.

[6] Wenzel Jakob. *Mitsuba renderer*. http://www.mitsuba-renderer.org. 2010.

[7] 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.

[8] Henrik Wann Jensen. “Global Illumination Using Photon Maps”. In: *Proceedings of the Eurographics Workshop on Rendering Techniques 96*. Porto, Portugal, 1996, pp. 21–30. ISBN: 3-211-82883-4.

[9] James T. Kajiya. “The Rendering Equation”. In: *SIGGRAPH 86* (1986), pp. 143–150. ISSN: 0097-8930.

[10] Markus Kettunen et al. “Gradient-Domain Path Tracing”. In: *ACM Trans. Graph.* 34.4 (2015).

[11] Eric P. Lafortune and Yves D. Willems. “Bi-directional path tracing”. In: *Proceedings of Third International Conference on Computational Graphics and Visualization Techniques (Compugraphics 93)*. 1993, pp. 145–153.

[12] Jaakko Lehtinen et al. “Gradient-domain Metropolis Light Transport”. In: *ACM Trans. Graph.* 32.4 (July 2013), 95:1–95:12. ISSN: 0730-0301.

[13] Marco Manzi et al. "Gradient-Domain Bidirectional Path Tracing". In: *Eurographics Symposium on Rendering - Experimental Ideas & Implementations.* 2015.

[14] Marco Manzi et al. "Temporal Gradient-domain Path Tracing". In: *ACM Trans. Graph.* 35.6 (Nov. 2016), 246:1–246:9. ISSN: 0730-0301.

[15] Hao Qin et al. "Unbiased Photon Gathering for Light Transport Simulation". In: *ACM Trans. Graph.* 34.6 (Oct. 2015), 208:1–208:14. ISSN: 0730-0301.

[16] Fabrice Rousselle, Wojciech Jarosz, and Jan Novák. "Image-space Control Variates for Rendering". In: *ACM Transactions on Graphics* 35.6 (2016), 169:1–169:12.

[17] Eric Veach. "Robust Monte Carlo Methods for Light Transport Simulation". AAI9837162. PhD thesis. Stanford, CA, USA, 1998. ISBN: 0-591-90780-1.

[18] Eric Veach and Leonidas Guibas. "Bidirectional estimators for light transport". In: *EGWR.* 1994, pp. 147–162.

[19] Eric Veach and Leonidas J. Guibas. "Optimally Combining Sampling Techniques for Monte Carlo Rendering". In: SIGGRAPH 95. 1995, pp. 419–428. ISBN: 0-89791-701-4.

[20] Martin Šik et al. "Robust Light Transport Simulation via Metropolised Bidirectional Estimators". In: *ACM Trans. Graph.* 35.6 (Nov. 2016), 245:1–245:12. ISSN: 0730-0301.