DOI: 10.1111/cgf.70328

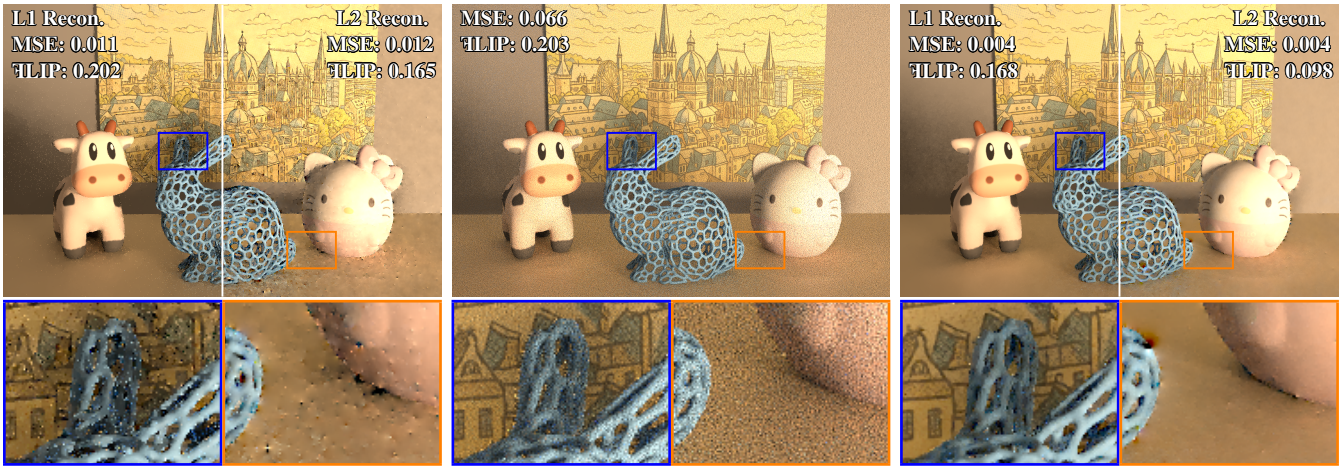
EUROGRAPHICS 2026 / B. Masia and J. Thies
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 45 (2026), Number 2

Gradient-Domain ReSTIR Path Tracing

Yu-Chen Wang[1], Markus Kettunen[2], Daqi Lin[2], Chris Wyman[2], Lifan Wu[2], Shuang Zhao[3]

[1]University of California, Irvine [2]NVIDIA [3]University of Illinois Urbana-Champaign



(a) **Gradient-PT** (b) **Area ReSTIR** (c) **Ours**

Figure 1: *Our ReSTIR G-PT reconstructs the image from Monte Carlo estimates of pixel colors and discrete image gradients, i.e., finite differences between adjacent pixels. Reusing both colors and gradients between pixels and frames, our method improves upon both gradient-domain path tracing [KMA*15] and Area ReSTIR [ZLK*24]. Here, we show an equal-time comparison (30 ms) in Aachen Toys with a moving camera. Gradient-domain path tracing produces noisy images due to the low sample budget and lack of spatiotemporal reuse. Area ReSTIR shows high-frequency noise due to not evaluating gradients, which efficiently capture high image frequencies. Our method produces cleaner images under L_1 and L_2 screened Poisson reconstruction. We leave advanced real-time image reconstruction from gradients as future work.*

Abstract

Gradient-domain rendering accelerates realistic image synthesis by also estimating pixel color differences, which helps reconstruct high frequencies in the image domain. Converged images still require many samples per pixel even with denoising, and to this date, no real-time gradient-domain rendering methods have been proposed. We enable gradient-domain methods in real-time rendering by spatiotemporal sample reuse with a novel path space extension in gradient image rendering.

We further explore this concept by implementing ReSTIR G-PT, ReSTIR gradient-domain path tracing, and find that relative sparsity of the gradient image allows highly selective spatial reuse and real-time frame rates. Our method outperforms the baseline methods visually and statistically.

Keywords: Path tracing, ReSTIR, gradient-domain, light transport simulation

1. Introduction

Provided a fully described 3D scene, physically-based rendering concerns with synthesizing physically accurate images of the scene. For scenes exhibiting complex light transport, such as environmental illumination, soft shadows, and inter-reflections, conventional rendering methods like path tracing [Kaj86] can suffer from slow convergence.

One approach to accelerate convergence in offline rendering is to also estimate *change* between pixels. The final images of this *gradient-domain rendering* [LKL*13; KMA*15], recovered by

solving a screened Poisson problem, show significantly reduced medium and high frequency noise compared to path tracing.

Reservoir-based spatiotemporal importance resampling (ReSTIR) [BWP*20; LKB*22; ZLK*24] has recently been introduced to accelerate interactive real-time situations, but has only been applied in the primal-domain. By chaining sample reuse from frame to frame and pixel to pixel, ReSTIR techniques can generate unbiased low-noise renderings of new frames with significantly less work than conventional methods that sample each pixel and frame from scratch.

Although both ReSTIR and gradient-domain rendering can outperform conventional rendering methods like path tracing, these two families of techniques have largely been orthogonal to each other: prior ReSTIR methods operate in the primal domain, and gradient-domain rendering techniques perform little, if any, sample reuse, and require many samples per pixel for high-quality images. A successful merge could allow realizing the cleaner images of gradient-domain rendering at real-time render budgets.

When later joined with real-time neural reconstruction in potential future work, this combination should excel at improved image clarity for light phenomena not present in the guide buffers. We leave this as an underlying motivation, and concentrate to merging ReSTIR with gradient-domain rendering. We show our results with the outdated L_1 and L_2 reconstructions that vastly underestimate the realistically realizable image quality [KHL19].

Previous ReSTIR methods produce pixel color estimates with single light paths. Estimating change, i.e., pixel differences or discrete image gradients, requires pairs of *correlated* paths; subtracting them produces low-variance difference estimates.

Combining ReSTIR with gradients turns out to be non-trivial. Should we evaluate gradients for ReSTIR paths, or run ReSTIR on path pairs corresponding to gradients? The answer turns out to be the latter. The gradient signal can be negative, but ReSTIR's target function can not; we solve this by positivization. Unbiased gradient estimation requires subtracting *two* difference integrals, one for each pixel, with shift mappings to the other; ReSTIR evaluates a *single* integral. We apply ReSTIR to a path space extension that joins two pixels into one domain. The gradient signal also turns out to be much more sensitive to sub-pixel precise temporal reuse than the primal signal; we adapt Area ReSTIR's fractional reservoirs to pixel pairs with pixel-pair motion vectors. The sparse nature of image gradients further allows selective spatial reuse of gradients, removing most of its cost. These observations together allow efficient real-time gradient-domain rendering with ReSTIR.

Concretely, we make the following technical contributions:

- We introduce a scheme to spatiotemporally reuse light path pairs while maintaining their correlation (Section 4.2);
- Leveraging this new reuse, we develop a new ReSTIR algorithm that operates in the gradient domain (Section 4.3); We develop our temporal reuse method based on Area ReSTIR [ZLK*24] with our fractional pixel pair motion vectors and introduce a new method to selectively reuse samples spatially.
- We further improve the effectiveness of our ReSTIR algorithm via positivization (Section 4.4).

We demonstrate the effectiveness of our technique by comparing with several state-of-the-art methods in Section 5.

2. Related Work

In the following, we position our work within prior gradient-domain and ReSTIR literature.

2.1. Gradient-domain rendering

LEHTINEN et al. [LKL*13] introduces gradient-domain rendering and shift mappings in the context of Metropolis Light Transport (MLT) [VG97]. They guide Markov chains with a mix of discrete gradient and primal signals to simultaneously produce gradient images I_{dx} and I_{dy} and a standard primal image I_{color}. They reconstruct the final image by solving a screened Poisson equation, with an L_2 reconstruction being unbiased, but L_1 often giving higher quality. They argue that gradient-domain rendering is beneficial due to the interplay of MLT's adaptivity and the sparsity of the gradient signal in the path space.

KETTUNEN et al. [KMA*15] introduces gradient-domain path tracing, showing gradient-domain rendering generalizes across rendering algorithms and is beneficial even without MLT's adaptivity. They trace the benefit to the interplay of the spectrum of natural images, the discrete gradient operator, and the screened Poisson solution. We present a ReSTIR version of gradient-domain path tracing.

MANZI et al. [MKD*16] evaluate finite differences between frames and reconstruct predefined animations in overlapping ten-frame blocks with a three-dimensional spatiotemporal screened Poisson solver; we target zero-latency interaction and thus use ReSTIR, rather than temporal finite differences, for taking advantage of other frames.

Others further generalize gradient-domain rendering, e.g., MANZI et al. [MKA*15] to bidirectional path tracing, HUA et al. [HGNH17] and GRUSON et al. [GHV*18] to photon density estimation, SUN et al. [SSC*17] to vertex connection and merging, and PETITJEAN et al. [PBE18] spectral rendering. FANG et al. [FH24] apply a basis expansion to primary hit BRDFs and solve each component separately. Our work is largely orthogonal to these developments.

KETTUNEN et al. [KHL19] and GUO et al. [GLL*19] introduced deep convolutional neural networks for gradient-domain reconstruction, with the same network size and architecture training well for both gradient and primal-domain [KHL19]. BACK et al. [BHHM20] trained a neural network to combine independent images and existing gradient-domain reconstructions for improved results. Recently, YAN et al. [YZYX24] introduced a non-neural filtering-based reconstruction that can be competitive with neural approaches. These methods reconstruct gradient-domain renderings but are too slow for real-time.

Our work focuses strictly on sampling, but we are motivated by the hope that future gradient-domain reconstructions could run in around 3 ms like modern primal-domain denoisers [NVI25].

Table 1: *List of symbols commonly used in the paper.*

Symbol	Definition
f	Path contribution
F_i	Pixel i's intensity
$\bar{x}$	Light path
Ω_i	Path space associated with pixel i
Δ_{ij}	Value difference between pixels i and j
T	Shift mapping of a light path
T^*	Shift mapping for *paired-pixel* sample
p_l	Candidate distribution for lth candidate path in ReSTIR
m_l	Resampling MIS weight for lth sample ReSTIR
$\hat{p}$	Target function for ReSTIR
w_l	Resampling weight for lth candidate path in ReSTIR
W_x	Unbiased contribution weight for sample x in ReSTIR
y	Chosen sample with ReSTIR
G_{ij}	*paired-pixel* sample contribution function
ω_{ij}	MIS weight used in gradient-domain rendering
ω^*_{ij}	MIS weight for *paired-pixel* samples

2.2. ReSTIR

TALBOT et al. [TCE05] introduces resampled importance sampling (RIS) to pick one or more samples from many, proportionally to a desired density. Building upon RIS, BITTERLI et al.'s [2020] ReSTIR DI reuses light samples across frames and pixels for real-time direct illumination. OUYANG et al. [OLK*21] extend this method to rough global illumination by treating path suffixes a s virtual point lights.

LIN et al. [LKB*22] introduces *generalized resampled importance sampling* (GRIS) to enable unbiased spatiotemporal path reuse with *shift mappings*. Their ReSTIR PT allows unbiased and efficient path reuse on glossy materials. ZHANG et al.'s [2024] Area ReSTIR improves temporal reuse on high-frequency pixels with fractional motion vectors. We build on ReSTIR PT with ZHANG et al.'s area reservoirs.

Spatiotemporal sample reuse introduces correlations between nearby pixels. SAWHNEY et al. [SLK*24] decreases the correlations by Markov chain mutations. KETTUNEN et al. [KLR*23] reuse parts of paths rather than full paths with their conditional ReSTIR. ZHANG et al. [ZLK*24] accelerates depth of field rendering with ReSTIR. CHANG et al. [CSN*23] and WANG et al. [WWWZ23] apply ReSTIR to differentiable and inverse rendering. These developments are orthogonal to our work.

3. Preliminaries

In this section, we outline the necessary preliminaries, but refer readers to WYMAN et al. [WKL*23] and HUA et al. [HGP*19] for details.

3.1. Gradient-Domain Path Tracing

Given a scene, gradient-domain path tracing [KMA*15] renders horizontal and vertical (discrete) gradient images, i.e., pixel differences between adjacent pixels, in addition to the primal image, to capture image-space high frequencies. These pixel value differences between the corresponding pixels i and j are

$$\Delta_{ij} = \int_{\Omega_i} f_i(\bar{x})\,\mathrm{d}\bar{x} - \int_{\Omega_j} f_j(\bar{x})\,\mathrm{d}\bar{x}. \tag{1}$$

The variance of a Monte Carlo difference estimate is determined by the variance of each estimator and their **covariance**. In gradient-domain rendering, differences Δ_{ij} are evaluated by *shift mappings*, deterministic bijections T_{ij} from Ω_i to Ω_j to correlate the two integrals' samples. The pixel difference becomes

$$\Delta_{ij} = \int_{\Omega_i} \left(f_i(\bar{x}) - f_j(T_{ij}(\bar{x})) \left| \frac{\partial T_{ij}}{\partial \bar{x}} \right| \right) \mathrm{d}\bar{x}. \tag{2}$$

Here, $\bar{x}$ is traditionally called the *base path* and $T_{ij}(\bar{x})$ the *offset path*, and $\left|\partial T_{ij}/\partial \bar{x}\right|$ is the Jacobian determinant of the shift mapping. In practice, T_{ij} is only a *partial* bijection, a bijection between a subset $\Omega_i{}' \subseteq \Omega_i$ and another subset $\Omega_j{}' \subseteq \Omega_j$. Not all paths in Ω_i can be shifted to Ω_j, and vice versa. To compute unbiased gradients, symmetric gradient estimation is used; both pixels sample base paths and shift them to the other pixel, and results are combined via multiple importance sampling (MIS),

$$\begin{aligned} \Delta_{ij} = &\int_{\Omega_i} \omega_{ij}(\bar{x}) \left(f_i(\bar{x}) - f_j(T_{ij}(\bar{x})) \left| \frac{\partial T_{ij}}{\partial \bar{x}} \right| \right) \mathrm{d}\bar{x} - \\ &\int_{\Omega_j} \omega_{ji}(\bar{x}) \left(f_j(\bar{x}) - f_i(T_{ji}(\bar{x})) \left| \frac{\partial T_{ji}}{\partial \bar{x}} \right| \right) \mathrm{d}\bar{x}, \end{aligned} \tag{3}$$

where $\omega_{ij}(\bar{x})$ are the MIS weights. A common choice is the balance heuristic [KMA*15],

$$\omega_{ij}(\bar{x}) = \frac{p_i(\bar{x})}{p_i(\bar{x}) + p_j(T_{ij}(\bar{x})) \left| \frac{\partial T_{ij}}{\partial \bar{x}} \right|}, \tag{4}$$

where p_i and p_j are the sampling PDFs for Ω_i and Ω_j, respectively. Our method essentially importance samples Equation 3 with ReSTIR.

3.2. Spatiotemporal Importance Resampling (ReSTIR)

The one-sample Monte Carlo estimator estimates $F = \int_\Omega f(x)\,\mathrm{d}x$ by drawing a sample x with probability density $p(x)$ and evaluating

$$\langle F \rangle_{\mathrm{MC}} = \frac{f(x)}{p(x)}. \tag{5}$$

This estimate has the lower variance, the better $p(x)$ approximates $f(x)$, modulo a constant multiplier. Unfortunately, building efficient path samplers with $p(x) \propto f(x)$ is impractical.

3.2.1. Resampled Importance Sampling

TALBOT et al.'s [2005] RIS draws samples *approximately* proportional to some target function $\hat{p}$, often chosen $\hat{p}(x) = f(x)$, giving a way to approximate the ideal distribution. RIS first draws M candidate samples $x_1, \ldots, x_M$ with PDFs $p_1, \ldots, p_M$ and computes a resampling weight for each sample,

$$w_l = m_l(x_l)\,\frac{\hat{p}(x_l)}{p_l(x_l)}. \tag{6}$$

Here, $m_l(x_l)$ is the resampling MIS weight, e.g., the balance heuristic [VG95]. RIS then picks sample y from the candidates proportionally to the w_l. This gives a one-sample estimator

$$\langle F \rangle_{\text{RIS}} = f(y) W_y, \tag{7}$$

with the *unbiased contribution weight* (UCW) [LKB*22]

$$W_y = \frac{1}{\hat{p}(y)} \sum_{l=1}^{M} w_l. \tag{8}$$

The RIS estimator converges faster *per-sample*, at the cost of sampling, storing, and picking from the M candidates. A common strategy was to use a cheaper target function [TCE05].

Weighted reservoir sampling [Cha82] allows implementing RIS in a streaming manner [BWP*20]. A *reservoir* stores the currently chosen sample y and the sum of resampling weights w_l streamed into it so far. Finally, W_y (Equation 8) is stored with y.

3.2.2. Generalized RIS

LIN et al. [LKB*22] extend resampling, enabling integral estimation by reusing samples across domains. Suppose we want to estimate integral F over some domain Ω with samples from domains $\Omega_1, \ldots, \Omega_M$. GRIS first maps the samples into our target domain Ω with *shift mappings* $T_l : \Omega_l \to \Omega$. Then it computes the resampling weight w_i for each shifted sample as

$$w_l = m_l(T_l(x_l))\, \hat{p}(T_l(x_l)) W_{x_l} \left| \frac{\partial T_l}{\partial x_l} \right|. \tag{9}$$

Shifted samples $T_l(x_l)$ are streamed to a new reservoir r with these resampling weights. Because generalized resampling accepts samples from other domains and Equation 9 uses W_{x_l} in place of the often unknown[†] $1/p_l(x_l)$, it can be applied iteratively, borrowing samples from neighbor pixels and prior frames. This iterative application of (generalized) RIS defines reservoir-based spatiotemporal importance resampling (i.e., ReSTIR). After iterative reuse, ReSTIR estimates F using Equation 7 with the stored y and W_y.

4. Our Method

We now introduce a technique that enables ReSTIR for gradient-domain path tracing, i.e., estimating pixel differences via spatiotemporal reuse. Because gradient-domain rendering and ReSTIR leverage correlations in different ways, we use two types of shift mappings: one to maintain path correlations for difference evaluation (Equation 3), another for spatiotemporal sample reuse (Equation 9).

In Section 4.1, we present a naïve approach, *Gradient after ReSTIR*, which shifts paths from primal-domain ReSTIR [LKB*22; ZLK*24] to other pixels for gradient evaluation. We discuss why this leads to high variance.

In Section 4.2 we introduce *Gradient with ReSTIR* and discuss why it exhibits less variance. In Section 4.3 we show how we apply *Gradient with ReSTIR* to compute image gradients, and in Section 4.4 we discuss positivization to further reduce variance.

[†] The PDF after resampling is intractable.

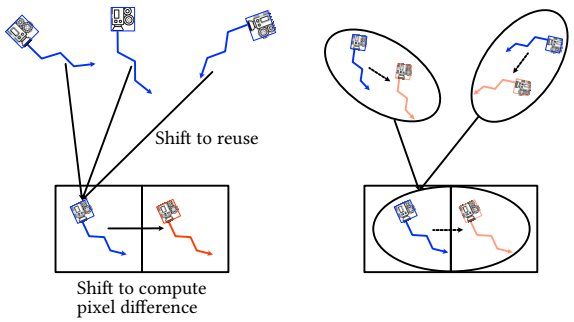


Figure 2: *(a) Gradient after ReSTIR (Section 4.1) reuses single paths spatiotemporally with ReSTIR and shifts each pixel's final path to adjacent pixels to form paired-pixel samples for gradient computation. (b) Gradient with ReSTIR (Section 4.2) reuses paired-pixel samples spatiotemporally with ReSTIR and uses the final paired-pixel sample for a gradient computation.*

Our method helps improve the quality of gradient images in real-time rendering. However, gradient-domain rendering also requires a primal rendering and a reconstruction pass. In the rendering pipeline with our method, we use a separate Area ReSTIR [ZLK*24] pass to generate the primal image and the traditional L1 and L2 Poisson reconstruction method to reconstruct the rendering output. We will discuss this in more detail in Section 5.

4.1. *Gradient after ReSTIR*

A naïve approach we call *Gradient after ReSTIR* reuses single paths spatiotemporally with primal-domain ReSTIR (e.g., ZHANG et al. [ZLK*24]), and evaluates gradients by shift mapping the ReSTIR paths to adjacent pixels. We illustrate this in Figure 2 (left).

We first do exactly as ReSTIR: initial sampling is followed by temporal reuse, spatial reuse, and primal color evaluation via

$$\langle F_i \rangle = f_i(\bar{y}_i)\, W_{\bar{y}_i}, \tag{10}$$

where $\bar{y}_i$ is ReSTIR's chosen path, and $W_{\bar{y}_i}$ is the unbiased contribution weight.

Next, we compute the differences Δ_{ij} for the adjacent pixels j by shifting $\bar{y}_i$ to pixel j with the shift mapping T_{ij} and estimate *one-sided* pixel value differences

$$\langle \Delta'_{ij} \rangle = \omega_{ij}(\bar{y}_i) \left(f_i(\bar{y}_i) - f_j(T_{ij}(\bar{y}_i)) \left| \frac{\partial T_{ij}}{\partial \bar{y}_i} \right| \right) W_{\bar{y}_i}. \tag{11}$$

Subtracting the corresponding one-sided estimators gives the full difference (see Equation 3), since

$$\Delta_{ij} = \mathbb{E}\left[\langle \Delta'_{ij} \rangle - \langle \Delta'_{ji} \rangle \right]. \tag{12}$$

This subtraction is required, since as noted in Section 3.1, T_{ij} may be a *partial* bijection, and paths from only one pixel do not cover the full difference.

Since resampling loses access to exact PDFs, we replace the balance heuristic (Equation 4) with the generalized balance heuristic

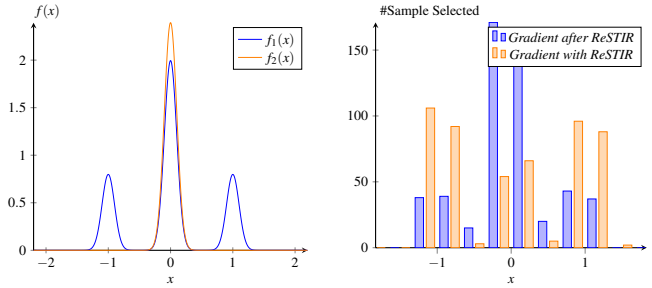


Figure 3: *Importance of Gradient with ReSTIR. **Left**: Functions $f_1(x)$ and $f_2(x)$, defined over $[-2,2]$, agree near the origin but differ significantly in other regions. We estimate $\int_{-2}^{2} f_1(x) - f_2(x)\ \mathrm{d}x$ with both Gradient after ReSTIR and Gradient with ReSTIR. **Right**: We repeat each algorithm 500 times and show the histogram of x-values selected by each method. Gradient after ReSTIR selects samples based on either $f_1(x)$ or $f_2(x)$, which leads to over-sampling near the peak, while under-sampling regions where $|f_1(x) - f_2(x)|$ is high. In contrast, Gradient with ReSTIR prioritizes sample selection based on the difference between $f_1(x)$ and $f_2(x)$, enabling more efficient estimation of the difference integral.*

([LKB*22]),

$$\hat{\omega}_{ij}(\bar{x}) = \frac{\hat{p}_i(\bar{x})}{\hat{p}_i(\bar{x}) + \hat{p}_j(T_{ij}(\bar{x}))\left|\frac{\partial T_{ij}}{\partial \bar{x}}\right|}, \tag{13}$$

where, as usual, the corresponding term is dropped if the shift mapping is undefined.

Gradient after ReSTIR is straightforward to implement, but can suffer from high variance since primal-domain ReSTIR only looks at the path contributions in a single pixel, neglecting the actual change between pixels. Thus, it often poorly samples pixel differences, and every shift mapping failure potentially causes an outlier.

In Figure 3, we employ *Gradient after ReSTIR* in a toy example to estimate the integral of $f_1(x) - f_2(x)$. This approach overlooks the difference between $f_1(x)$ and $f_2(x)$, selecting samples that are significant f or e ither $f_1(x)$ o r $f_2(x)$ r ather t han t hose f or which $f_1(x) - f_2(x)$ has high contribution. We should prioritize sampling in regions critical for estimating the $f_1 - f_2$. Our Gradient with ReSTIR achieves this.

4.2. *Gradient with ReSTIR*

To specifically reduce variance in pixel difference estimation, we introduce a new *Gradient with ReSTIR* method to choose samples based on the integrand differences between two pixels. To capture the difference between two pixels, we store and reuse samples based on a pixel pair (i, j) instead of a single pixel in primal rendering. In Section 4.2.1, we introduce the sample we reused in our method. We call it *paired-pixel* sample. In what follows, we introduce our formulation for estimating pixel differences shown in Section 4.2.2. We then design a shift mapping required by ReSTIR to reuse *paired-pixel* samples spatialtemporally in Section 4.2.3.

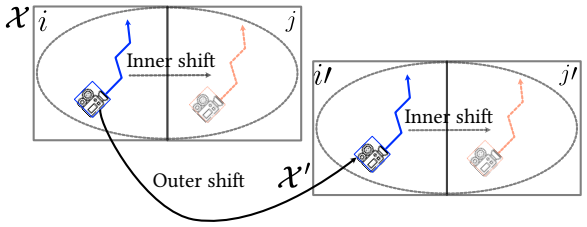


Figure 4: *When we shift a gradient-domain sample $(\bar{x}, k)$ to another domain, firstly we shift $\bar{x}$ to the corresponding pixel in the target pixel pair. Then we use the Inner Shift in the new gradient-domain $\mathcal{X}'$ to obtain the shifted sample in target domain.*

4.2.1. Paired-Pixel Samples

Given a pixel pair (i, j) associated with path spaces (Ω_i, Ω_j), let T_{ij} and T_{ji} be shift maps from Ω_i to Ω_j and Ω_j to Ω_i, respectively. Similar to Equation 51 in previous ReSTIR works [LKB*22], we rewrite Equation 3 by merging the two integrals into a single one over an extended *paired-pixel* domain: $\mathcal{X} := (\Omega_i \times \{0\}) \cup (\Omega_j \times \{1\})$:

$$\Delta_{ij} = \int_{\mathcal{X}} \omega^*_{ij}(\bar{x}, k)\, G_{ij}(\bar{x}, k)\ \mathrm{d}(\bar{x}, k), \tag{14}$$

where the variable of integration $(\bar{x}, k)$ involves a base path $\bar{x}$ and an indicator $k \in \{0, 1\}$ encoding the domain in which $\bar{x}$ resides (i.e., $k = 0$ and $k = 1$ indicate, respectively, $\bar{x} \in \Omega_i$ and $\bar{x} \in \Omega_j$). Additionally,

$$G_{ij}(\bar{x}, k) := \begin{cases} f_i(\bar{x}) - f_j(T_{ij}(\bar{x}))\,|\partial T_{ij}/\partial \bar{x}| & \text{if } k = 0, \\ f_i(T_{ji}(\bar{x}))\,|\partial T_{ji}/\partial \bar{x}| - f_j(\bar{x}) & \text{if } k = 1, \end{cases} \tag{15}$$

where T_{ij} shifts path between i and j and we call it *Inner Shift*. It can be any shift mapping designed in previous works [LKB*22; KMA*15] to shift a path between two pixels. The MIS weight $\omega^*_{ij}(\bar{x}, k)$ is given by

$$\omega^*_{ij}(\bar{x}, k) := \begin{cases} \omega_{ij}(\bar{x}) & \text{if } k = 0, \\ \omega_{ji}(\bar{x}) & \text{if } k = 1, \end{cases} \tag{16}$$

with w_{ij} from Equation 4.

4.2.2. Estimating Pixel Difference with *paired-pixel* sample

Now we discuss our formulation to estimate pixel difference with *paired-pixel* sample from different pixel pairs.

Suppose we have M candidate *paired-pixel* samples $\{(\bar{x}_l, k_l)\}_{l=1}^{M}$ and they are sampled from $\mathcal{X}_1, \ldots, \mathcal{X}_M$. The target pixel pair is denoted as (i, j) and the associated *paired-pixel* domain is $\mathcal{X}$. We firstly shift all *paired-pixel* samples to $\mathcal{X}$ with shift mapping $T^*_l := \mathcal{X}_l \to \mathcal{X}$. We call T^* *Outer Shift* and will discuss it later. We then select a *paired-pixel* sample $(\bar{y}^*, k^*)$ from all candidate samples. Following GRIS theory[LKB*22], our pixel difference estimator takes the form of:

$$\langle \Delta_{ij} \rangle = W_{(\bar{y},k)}\left(\omega^*_{ij}(\bar{y}^*, k^*)\, G_{ij}(\bar{y}^*, k^*)\right). \tag{17}$$

$W_{(y^*,k^*)}$ is the unbiased contribution weight. It takes the form of:

$$W_{(y^*,k^*)} = \frac{1}{\hat{p}(y^*,k^*)} \sum_{l=1}^{M} w_l(\bar{x}_l,k_l). \tag{18}$$

where w_l is the resampling weight in GRIS described in Section 3.2. Since the target function $\hat{p}$ has to be non-negative, we initially present the theory with

$$\hat{p}(x,k) = \omega^*_{ij}(T^*(\bar{x},k)) \left| G_{ij}(T^*(\bar{x},k)) \right| \tag{19}$$

and resampling weights

$$w_l(\bar{x}_l,k_l) = m_l(\bar{x}_l,k_l)\ \hat{p}(\bar{x}_l,k_l) W_{(\bar{x}_i,k_i)} \left| \frac{\partial T^*}{\partial(\bar{x}_l,k_l)} \right|. \tag{20}$$

In Section 4.4, we robustify our method with positivization, separating the positive and negative parts into different reservoirs.

By reusing *paired-pixel* samples and weighting them with the sampled difference shown in Equation 19, we can sample correlated paths with high pixel difference values. Shown in Figure 3, *Gradient with ReSTIR* selects samples in the two small peaks, where $f_1(x)$ and $f_2(x)$ differ the most, leading to cleaner pixel difference estimation.

4.2.3. Gradient-Domain Shift Mapping

As discussed in previous sections, to reuse *paired-pixel* samples spatialtemporally, it requires an *Outer Shift* T^* to shift *paired-pixel* samples from other pixel pairs to the current pixel pair. We now introduce our *Outer Shift*. It takes form of

$$T^*(\bar{x},k) = (\tau(\bar{x}),\gamma(k)), \tag{21}$$

where τ shifts the base path and γ shifts the base path indicator k.

Let $\mathcal{X}$ and $\mathcal{X}'$ be associated with the pixel pairs (i,j) and (i,j'), respectively. As illustrated in Figure 4, to shift a sample $(\bar{x},k) \in \mathcal{X}$ to the other *paired-pixel* domain $\mathcal{X}'$, we first shift the base path indicator k with an identity shift $\gamma(k) = k$. Then we shift the light path $\bar{x}$ to the corresponding pixel among (i',j'), resulting in a new path $\bar{x}'$ and obtain the shifted sample $(\bar{x}',k)$ in the target domain $\mathcal{X}'$.

Evaluating Equation 20 requires computing the Jacobian of the *Outer Shift* T^*. We recall that, given a sample $(\bar{x},k)$, we shift our indicator k with an identity shift. Therefore, it is easy to verify that the Jacobian of our *Outer Shift* simply equals that of the shift mapping τ we use to shift the base path $\bar{x}$ of the current *paired-pixel* sample:

$$\left| \frac{\partial T^*}{\partial(\bar{x},k)} \right| = \left| \frac{\partial \tau}{\partial \bar{x}} \right|, \tag{22}$$

where the Jacobian $|\partial\tau/\partial\bar{x}|$ is that of shift mapping between two pixels derived in prior works [KMA*15; LKB*22].

4.3. Gradient-Domain Rendering with *Gradient with ReSTIR*

In Section 4.2, we introduce *Gradient with ReSTIR* to compute pixel value differences with ReSTIR. In what follows, we adopt *Gradient with ReSTIR* to compute pixel difference between neighbour pixel pairs in [KMA*15] and introduce the first gradient-domain method in real-time rendering. Specifically, in Section 4.3.1, we will discuss reusing *paired-pixel* samples temporally with Area ReSTIR [ZLK*24] and a new method to compute a pixel pair's motion vector. In Section 4.3.2, we will introduce a new strategy to reuse *paired-pixel* samples adaptively to make our method efficient in real-time rendering.

4.3.1. Temporal Reuse

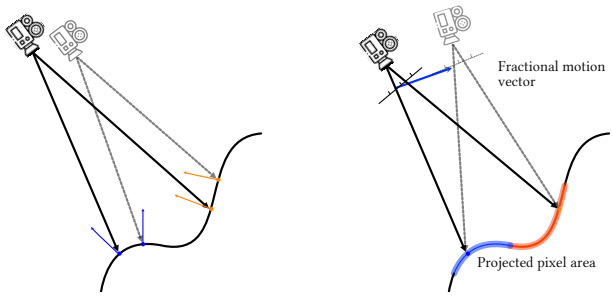


Figure 5: *(a) shows point-based ReSTIR. It fixes the primary intersection x_1 and reuses the rest of the paths. In image gradient computation, both primary intersections in a paired-pixel sample may be different from these in another paired-pixel sample, which greatly downgrade the reuse quality. (b) shows area-based ReSTIR, it reuses fractional motion vector to find closer temporal neighbor and improve the sample reuse quality.*

We first discuss temporal reuse in our gradient image rendering. In primal rendering, there are mainly two ways to reuse paths from past frames. LIN et al. [LKB*22] fix the primary intersection x_1 of a path $\bar{x}$ and reuse the rest of the path. Area ReSTIR [ZLK*24] expands the path reuse to pixel area and it works better in high-frequency areas (e.g, Normal map, complex textures).

We observe that the integrand of the image gradient in Equation 3 changes rapidly between different pixel pairs. Shown in Figure 5, both pixels have different geometry properties from other pixels, which leads to large differences in image gradient. Therefore, we rely on Area ReSTIR to find temporal neighbors. Following Area ReSTIR, given a pixel pair (i,j), we first project it back to the last frame with a fractional motion vector δ. We then search a sample for the pixel pair $(i+\delta, j+\delta)$ in the past frame and shift it back to the current frame.

Unlike primal rendering, the G-buffer does not provide a motion vector for a pixel pair (i,j). We therefore design a new way to compute the fractional motion vector for a pixel pair. We pick the pixel with the smaller depth as the pivot and use its fractional motion vector for the entire pair. Anchoring on the nearer pixel helps preserve the correct depth ordering—since, under camera motion, the farther pixel may become occluded by the closer one. Once the pair's motion vector is established, we apply temporal reuse using Area ReSTIR [ZLK*24] as discussed above.

4.3.2. Adaptive Spatial Reuse

Now we introduce a novel method to find and reuse *paired-pixel* samples spatially. In previous primal ReSTIR works [LKB*22; ZLK*24], given a pixel i, ReSTIR randomly chooses a pixel in a

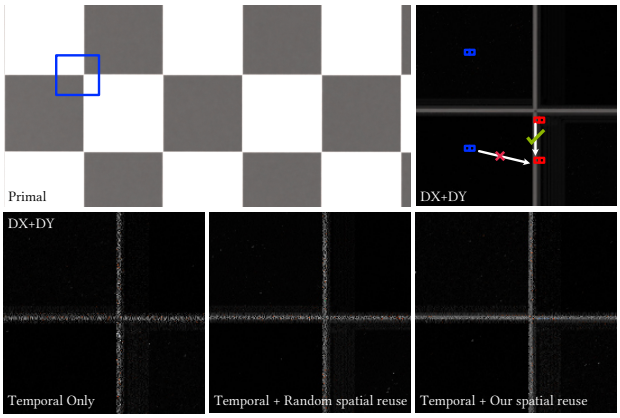


Figure 6: *We employ Gradient with ReSTIR with different spatial reuse variants to render gradient images of a dark-white grid. The image gradient is very different between interior pixel pair and edge pixel pair, so it is not good to reuse samples between them. The previous random spatial neighbor search fails to find suitable spatial neighbors, and it wastes a lot of time on interior pixel pair where the magnitude of image gradients is small. Our adaptive and edge-aligned spatial neighbor search enables efficient paired-pixel samples reuse in gradient-domain rendering's image gradient estimation.*

screen-space neighborhood and reuses its path. Because of geometry and material similarities between neighboring pixels, the paths from spatial neighbors can further reduce the noise of rendering results.

Unlike primal rendering, our *paired-pixel* sample is based on two pixels, and the image gradient can be sparse in a neighborhood. Shown in Figure 6, most of the pixel pairs in a black-white grid have zero gradient; we call these pixel pairs *interior pixel pair*. Only pixel pairs on the edge of grids have large image gradients, we call these pixel pairs *edge pixel pair*. Because of the sparse distribution of image gradient, the spatial neighbor search used in primal rendering is not practical in gradient-domain spatial reuse.

We introduce our novel spatial reuse for *paired-pixel* samples. It includes two key ideas. (1). Adaptively decide whether to do spatial reuse based on the G-Buffer. (2). Edge aligned spatial neighbor search and reuse. Firstly, we introduce adaptive gradient-domain spatial reuse. Shown in Figure 6, we calculate gradient images of a white-black grid. We find t hat t he magnitude of pixel difference between *interior pixel pair* (shown as blue dots) is small and it is relatively easy to sample. Due to performance consideration, we omit the spatial reuse between *interior pixel pair* by looking at G-Buffer before spatial reuse and only conduct spatial reuse for *edge pixel pair* (shown as orange dot in Figure 6). In our experiments, for a pixel pair (i,j), if $|\text{albedo(i)} - \text{albedo(j)}| > 0.1$, $\text{Dot}(\text{normal(i)}, \text{normal(j)}) > 0.9$, or $|\text{depth(i)} - \text{depth(j)}| / \text{depth(i)} > 0.1$, we will consider (i,j) as an edge pixel pair.

Now we discuss how we find a spatial neighbor for an *edge pixel pair*. Shown in Figure 6, the pixel difference is very different in *interior pixel pair* and *edge pixel pair*, so it is bad to reuse *paired-pixel* samples from *interior pixel pair* to *edge pixel pair*. For a pixel pair (i,j), we randomly generate at most N neighbor pixel pairs in the current pixel pair's neighborhood and check if they are similar to the current *edge pixel pair* by looking at albedo, depth and normal in the G-Buffer. We find that it is very efficient even when N is large (We choose $N = 256$ for experiments, which takes 0.1ms on spatial neighbor search in our experiment setup). We provide an ablation study of spatial reuse in Figure 10.

After spatial neighbor look-up, we shift the *paired-pixel* sample associated with it to the current *paired-pixel* domain with *paired-pixel* shift mapping we introduce in Section 4.2.3 and resample it with GRIS, same as previous primal ReSTIR works [LKB*22].

4.4. Positivization

In gradient-domain rendering, the image gradient is a real-valued function. To use *Gradient with ReSTIR*, we use the *paired-pixel* sample contribution function $G_{ij}(\bar{x},k)$ as the target function. To reduce the sign noise introduced by sign changes during image gradient estimation, we apply positivization to estimate the gradient. Positivization splits a real-valued function $f(x)$ into a positive part and a negative part and estimates them separately:

$$f(x) = \max(f(x),0) - \max(-f(x),0). \tag{23}$$

Following this, our integral for pixel value difference for a pixel pair (i,j) introduced in Equation 14 becomes:

$$\Delta_{ij} = \int_{\mathcal{X}} \hat{p}_+(\bar{x},k)\,\mathrm{d}(\bar{x},k) - \int_{\mathcal{X}} \hat{p}_-(\bar{x},k)\,\mathrm{d}(\bar{x},k), \tag{24}$$

where

$$\hat{p}_+(\bar{x},k) := \max\left(\omega^*_{ij}(\bar{x},k)\,G_{ij}(\bar{x},k),0\right) \tag{25}$$

$$\hat{p}_-(\bar{x},k) := \max\left(-\omega^*_{ij}(\bar{x},k)\,G_{ij}(\bar{x},k),0\right). \tag{26}$$

Our implementation has two reservoirs r_+ and r_- for each pixel pair with target functions $\hat{p}_+(\bar{x},k)$ and $\hat{p}_-(\bar{x},k)$, respectively, to estimate the two integrals in Equation 24. When generating initial candidate samples, r_+ and r_- will share the same set of candidates. We add each initial candidate *paired-pixel* $(\bar{x}_l,k_l)$, to both reservoirs.

We are estimating the image gradient with positivization, and a gradient sample that has a positive contribution may have a negative contribution after a shift mapping. Previously, CHANG et al. [CSN*23] has proposed to reuse samples **across signs**. That is, to generate new r_+ or r_- for a pixel pair, they try to reuse the samples from both r_+ **and** r_- from other pixel pairs. In practice, we find that reuse across signs does not perform better than reusing samples within the same sign, but it takes longer. To make our method more efficient, we choose not to use samples across signs. When we generate new r_+, we only reuse samples from r_+ from other pixel pairs, and the same for r_-. We provide an ablation study Figure 9 in the result section.

5. Results

We adopt ReSTIR for image gradient estimation, enabling gradient-domain rendering in real-time rendering. To demonstrate the efficiency of our method, we compare it with three baselines:

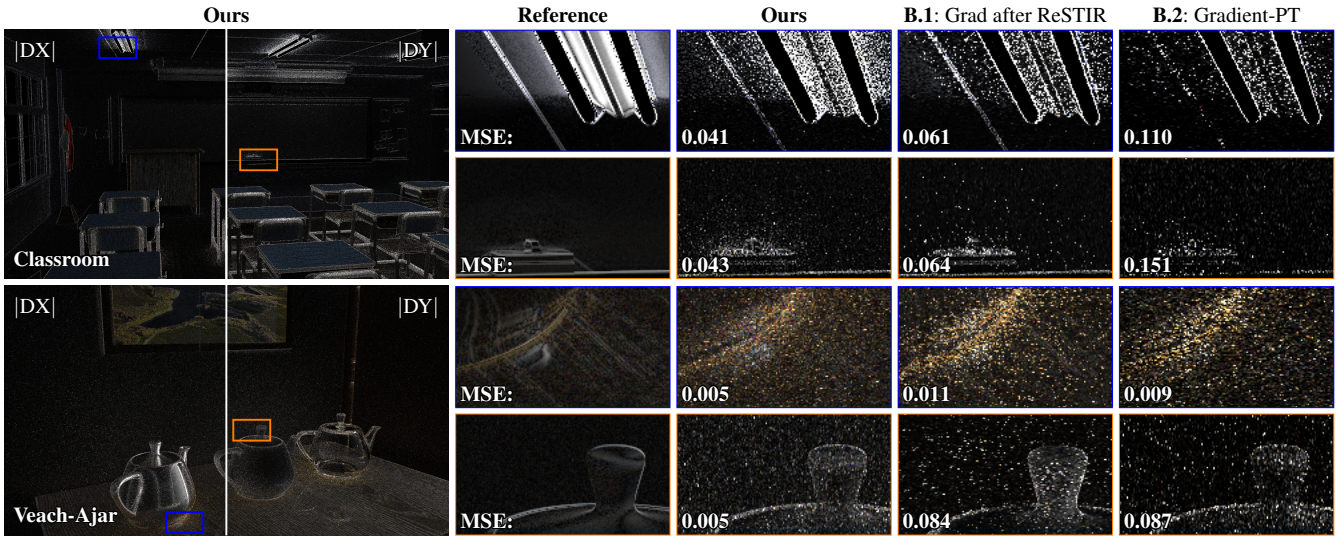


Figure 7: *Gradient image comparison. To make the results equal-time (30 ms), we run our method at 1 spp, gradient-domain path tracing [KMA*15] at 3 spp, and average the gradients of two instances of Gradient after ReSTIR. Our method tends to produce highest-quality gradients.*

B.1 ***Gradient after ReSTIR***: The method we describe in Section 4.1. This method is a gradient-domain method, so it requires a primal pass and a gradient pass to produce rendering outputs. We use the Area ReSTIR [ZLK*24] in the primal pass. We then use the primal paths generated in the primal pass to compute the image gradients by shifting them to the neighboring pixels, as we describe in Section 4.1. To achieve equal-time comparison, we run multiple independent copies of this method, averaging the results.

B.2 **Gradient-PT**: KETTUNEN et al.'s [2015] gradient-domain path tracer. For fair comparison, we use LIN et al. [LKB*22]'s hybrid shift mapping to shift paths between pixels. We use the standard PT in the primal pass and gradient-domain path tracing in the gradient pass, the same as in the original paper [KMA*15]. To achieve equal-time comparison, we run multiple spps and report the results.

B.3 **Area ReSTIR**: ZHANG et al.'s [2024] recent Area ReSTIR; Since this is not a gradient-domain method, we achieve equal-time comparisons by running multiple independent copies of Area ReSTIR, averaging the results.

Like **B.1**, our method also has a primal and a gradient pass. We use Area ReSTIR [ZLK*24] in the primal pass. We then use the *Gradient with ReSTIR* in Section 4.2 to compute the image gradients. We set the number of initial candidates to 1 and the number of spatial neighbors to 1 for *Gradient with ReSTIR*.

In the primal Area ReSTIR we are using in **B.1**, **B.3**, and our method, we set the number of initial candidates to 1 and the number of spatial neighbors to 2.

Reconstruction Gradient-domain rendering requires a reconstruction step to produce the final rendering result. We use L1 and L2 Poisson reconstructions for rendering output generation. They take 300ms and 11ms, respectively. Please note that modern neural denoisers (e.g. DLSS [NVI25]) can be more efficient and may take as low as 3ms. We expect future neural gradient-domain reconstruction methods to have similar efficiency. In our experiments, we include the L2 reconstruction time in all gradient domain rendering methods when we conduct an equal-time comparison with **B.3**.

We implemented our method and all baselines using the Falcor framework [KCK*22]. All methods share the same interfaces for lighting, geometry and materials. We run experiments on a PC with an Nvidia GeForce RTX 5090 GPU and AMD Ryzen 5900X CPU. All results are rendered in 1920x1080 resolution under a moving camera.

5.1. Evaluations and Ablations

Gradient Image Comparisons We compare gradient images between our method, *Gradient after ReSTIR* (**B.1**) and Gradient-PT (**B.2**). We show the gradient image comparisons in Figure 7. We present with *Classroom* and *Veach-Ajar*. Gradient-PT (**B.2**) produces noisy gradient images because of low sampling budget and no spatial-temporal reuse. *Gradient after ReSTIR* (**B.1**) improves the quality of the gradient image by reusing paths in primal rendering. However, it fails to obtain high-contribution paths in the gradient domain since its target function $\hat{p}$ is associated with the primal image. Our method improves the quality of the gradient image by reusing *paired-pixel* samples and directly estimating the image gradient. It generally produces cleaner gradient images than other methods.

Reconstructed Image Comparisons In this paper, we introduce methods to reuse samples in the gradient domain. We find that with our method, we can have cleaner gradient images, and this leads to better reconstructed rendering output. To compare the final render-

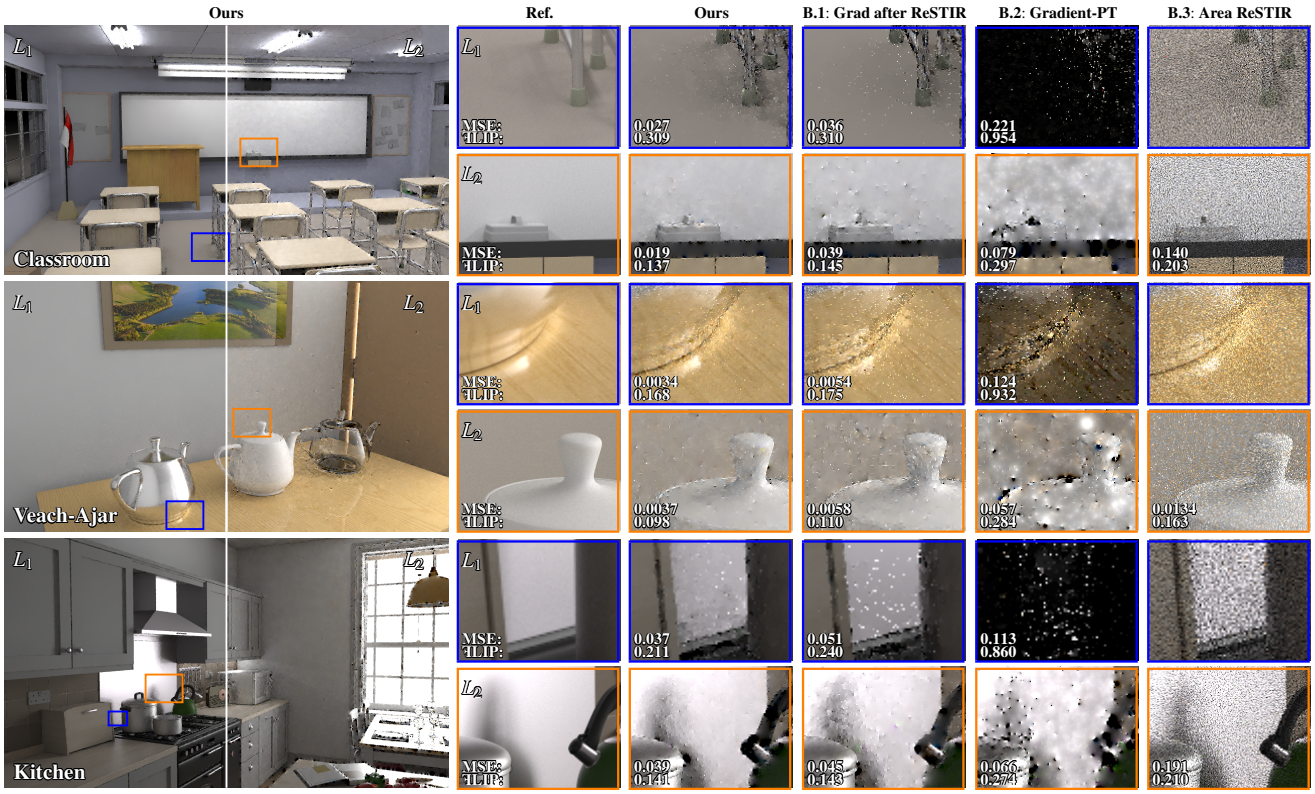


Figure 8: *Comparison to the most relevant baselines. To make results equal-time (30 ms), we average one instance of our method, two instances of Gradient after ReSTIR, four instances of Area ReSTIR [ZLK*24], and run Gradient-PT [KMA*15] at 3 spp. For gradient-domain methods, we average the primal and gradient images before reconstruction and show L_1 and L_2 reconstruction results. We show the error of L_1 and L_2 reconstruction for ours, **B.2** and **B.1** in the first and second row, respectively. **B.3** is a primal rendering method; we show the error of **B.3** in the second row only. Gradient-PT performs badly due to a lack of spatiotemporal reuse. Area ReSTIR [ZLK*24] shows high-frequency noise due to not evaluating gradients. Gradient after ReSTIR evaluates gradients based on primal-domain ReSTIR, which results in unoptimal sampling distributions and turns ineffective shift mappings into gradient outliers. Our method typically produces cleaner results, though L_1 and L_2 should be replaced with fast neural reconstructions.*

ing output with baselines, we report results from L1 and L2 Poisson reconstructions for all gradient domain methods. We show *Classroom*, *Veach-Ajar* and *Kitchen* in the paper and the result is in Figure 7. For more results, please refer to the supplemental materials.

Positivization helps reduce sign variance in image gradients estimation and produces cleaner rendering outputs. In Figure 9, we compare our method with and without positivization. For our method without positivization, we use Equation 19 as the target function and keep one reservoir only for each pair of pixels. We keep two reservoirs for one pair of pixels when positivization is turnt on. Although this costs more memory and computation time, it still outperforms our method without positivization.

We compare our method with different spatial reuse methods in Figure 10. When we turn off spatial reuse, the results have artifacts caused by gradient image noise due to the lack of spatial reuse. Dense spatial reuse randomly selects spatial neighbors for all pixel pairs. It is slow and does not find helpful spatial neighbors. Our spatial reuse reduces gradient noise in edge pixel pairs and leads to

In Figure 11, we compare the denoised Area ReSTIR [ZLK*24] and the rendering output of our method with a neural reconstruction method. We apply NGPT [KHL19] to both methods. With the help of gradient images, our method preserves more geometric details than the primal Area ReSTIR.

In gradient-domain rendering, it requires a primal and a gradient pass to produce the outputs. In Figure 12, we compare different combinations of primal and gradient image rendering settings on *Veach-Ajar*. Noise in both the primal images and the gradient images leads to artifacts in the final rendering output. Thanks to sample reuse, our method provides clean rendering output.

6. Discussion and Conclusion

Limitations and future work In this paper, we introduce the first gradient-domain method in real-time rendering. Our method is implemented in a real-time rendering framework, and it is non-trivial to apply it in offline rendering. We view this as an interesting future work.

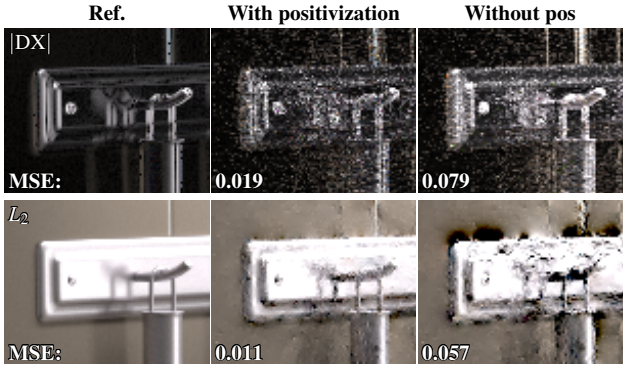


Figure 9: *Ablation on gradient positivization. The metal rack is lit by two lights, creating both positive and negative gradient contributions. Positivization reduces sign variance, leading to significantly improved image quality at increased frame time (39.8 ms vs. 29.9 ms).*

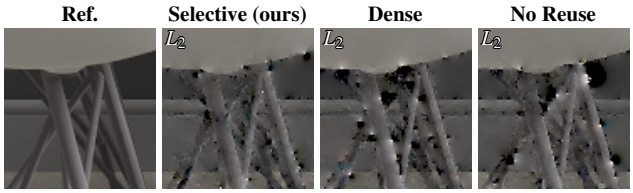


Figure 10: *Ablation on spatial reuse of gradient samples at 1 spp. Our adaptive variant skips spatial gradient reuse on flat areas, as these gradients are often already handled well by the inner shift and temporal gradient reuse, decreasing runtime cost by roughly 30%. Based on a G-buffer, selective reuse also selects reuse inputs from similar pixel-pairs, often improving gradient quality despite the lower cost.*

In our work, we assume the transformation of all objects is affine. Improving our method to better handle dynamic scenes with complex deformations is an interesting topic for future research. The video our method produces has some temporal turbulence in the high-frequency area due to the lack of temporal gradients. Extending our method to compute the temporal gradient will be an interesting research topic. Additionally, extending our technique to reuse antithetic samples used in differentiable rendering [ZDDZ21; YZN*22] is worth exploring.

Conclusion In this paper, we introduce a first gradient-domain real-time rendering method by reusing *paired-pixel* samples with ReSTIR. To make our method more efficient, we introduce an adaptive and edge-aligned spatial reuse. We further reduce the sign noise by adopting positivization. Finally, we evaluate our methods by comparing with several primal- and gradient-domain baselines in equal-time settings.

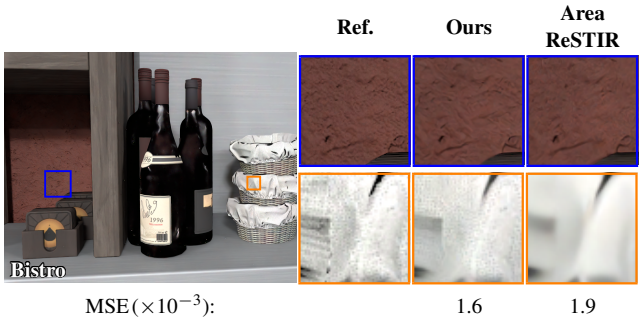


Figure 11: *Equal-time comparison between denoised Area ReSTIR [ZLK*24] and our method with neural reconstruction. We use the primal version of NGPT [KHL19] to denoise Area ReSTIR. The gradient images in our method help capture geometric details in the scene, thereby improving the rendering quality.*

References

[BHHM20] BACK, JONGHEE, HUA, BINH-SON, HACHISUKA, TOSHIYA, and MOON, BOCHANG. "Deep combiner for independent and correlated pixel estimates". *ACM Transactions on Graphics* 39.6 (2020), 1–12 2.

[BWP*20] BITTERLI, BENEDIKT, WYMAN, CHRIS, PHARR, MATT, et al. "Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting". 39.4 (July 2020) 2–4.

[Cha82] CHAO, M. T. "A General Purpose Unequal Probability Sampling Plan". *Biometrika* 69.3 (1982), 653–656. (Visited on 01/18/2025) 4.

[CSN*23] CHANG, WESLEY, SIVARAM, VENKATARAM, NOWROUZEZAHRAI, DEREK, et al. "Parameter-space ReSTIR for Differentiable and Inverse Rendering". *ACM SIGGRAPH 2023 Conference Proceedings*. SIGGRAPH '23. Los Angeles, CA, USA: Association for Computing Machinery, 2023. ISBN: 9798400701597 3, 7.

[FH24] FANG, QIQIN and HACHISUKA, TOSHIYA. "Lossless Basis Expansion for Gradient-Domain Rendering". *Computer Graphics Forum*. Vol. 43. 4. Wiley Online Library. 2024, e15153 2.

[GHV*18] GRUSON, ADRIEN, HUA, BINH-SON, VIBERT, NICOLAS, et al. "Gradient-domain volumetric photon density estimation". *ACM Transactions on Graphics (TOG)* 37.4 (2018), 1–13 2.

[GLL*19] GUO, JIE, LI, MENGTIAN, LI, QUEWEI, et al. "GradNet: unsupervised deep screened poisson reconstruction for gradient-domain rendering". *ACM Transactions on Graphics (TOG)* 38.6 (2019), 1–13 2.

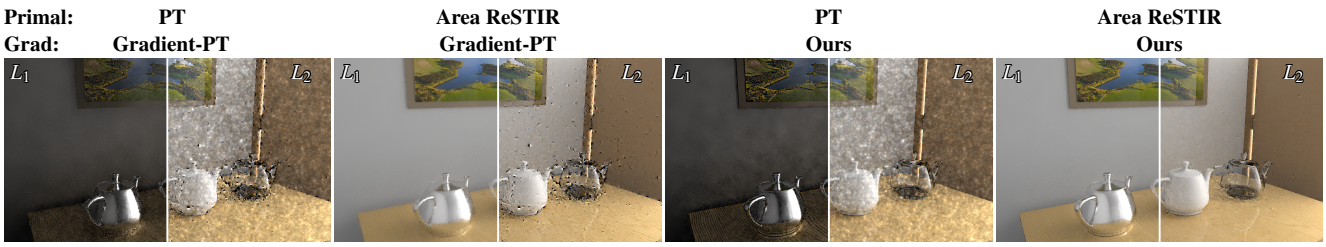
[HGNH17] HUA, BINH-SON, GRUSON, ADRIEN, NOWROUZEZAHRAI, DEREK, and HACHISUKA, TOSHIYA. "Gradient-domain photon density estimation". *Computer Graphics Forum*. Vol. 36. 2. Wiley Online Library. 2017, 31–38 2.

[HGP*19] HUA, BINH-SON, GRUSON, ADRIEN, PETITJEAN, VICTOR, et al. "A Survey on Gradient-Domain Rendering". *Computer Graphics Forum*. Vol. 38. 2. Wiley Online Library. 2019, 455–472 3.

[Kaj86] KAJIYA, JAMES T. "The rendering equation". *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*. 1986, 143–150 1.

[KCK*22] KALLWEIT, SIMON, CLARBERG, PETRIK, KOLB, CRAIG, et al. *The Falcor Rendering Framework*. `https://github.com/NVIDIAGameWorks/Falcor`. Aug. 2022. URL: `https://github.com/NVIDIAGameWorks/Falcor` 8.

[KHL19] KETTUNEN, MARKUS, HÄRKÖNEN, ERIK, and LEHTINEN, JAAKKO. "Deep convolutional reconstruction for gradient-domain rendering". *ACM Transactions on Graphics (TOG)* 38.4 (2019), 1–12 2, 9, 10.



MSE:	0.113	0.042	0.081	0.008	0.029	0.035	0.003	0.003
ꟻLIP:	0.932	0.284	0.923	0.249	0.175	0.167	0.168	0.098

Figure 12: *Ablation on different combinations of primal and gradient image rendering under equal time. As shown in the first and third columns, the lack of sample reuse in primal rendering results in dark L1 reconstructions and blobs in L2 reconstructions. The second row illustrates that the lack of reuse in gradient image rendering leads to artifacts in rendering outputs. Thanks to the sample reuse in both primal and gradient image renderings, our method produces cleaner rendering outputs than other gradient-domain rendering configurations.*

[KLR*23] KETTUNEN, MARKUS, LIN, DAQI, RAMAMOORTHI, RAVI, et al. "Conditional Resampled Importance Sampling and ReSTIR". *SIGGRAPH Asia (Conference Track)*. Dec. 2023 3.

[KMA*15] KETTUNEN, MARKUS, MANZI, MARCO, AITTALA, MIIKA, et al. "Gradient-domain path tracing". *ACM Transactions on Graphics (TOG)* 34.4 (2015), 1–13 1–3, 5, 6, 8, 9.

[LKB*22] LIN, DAQI, KETTUNEN, MARKUS, BITTERLI, BENEDIKT, et al. "Generalized resampled importance sampling: foundations of ReSTIR". *ACM Trans. Graph.* 41.4 (July 2022). ISSN: 0730-0301 2–8.

[LKL*13] LEHTINEN, JAAKKO, KARRAS, TERO, LAINE, SAMULI, et al. "Gradient-domain metropolis light transport". *ACM Transactions on Graphics (TOG)* 32.4 (2013), 1–12 1, 2.

[MKA*15] MANZI, MARCO, KETTUNEN, MARKUS, AITTALA, MIIKA, et al. "Gradient-Domain Bidirectional Path Tracing." *EGSR (EI&I)*. 2015, 65–74 2.

[MKD*16] MANZI, MARCO, KETTUNEN, MARKUS, DURAND, FRÉDO, et al. "Temporal gradient-domain path tracing". *ACM Transactions on Graphics (TOG)* 35.6 (2016), 1–9 2.

[NVI25] NVIDIA. *NVIDIA DLSS 4: Transforming Real-Time Graphics with AI*. `https://research.nvidia.com/labs/adlr/DLSS4/`. [Online; accessed 22-May-2025]. 2025 2, 8.

[OLK*21] OUYANG, Y., LIU, S., KETTUNEN, M., et al. "ReSTIR GI: Path Resampling for Real-Time Path Tracing". *Computer Graphics Forum* 40.8 (2021), 17–29 3.

[PBE18] PETITJEAN, VICTOR, BAUSZAT, PABLO, and EISEMANN, ELMAR. "Spectral gradient sampling for path tracing". *Computer Graphics Forum*. Vol. 37. 4. Wiley Online Library. 2018, 45–53 2.

[SLK*24] SAWHNEY, ROHAN, LIN, DAQI, KETTUNEN, MARKUS, et al. "Decorrelating ReSTIR Samplers via MCMC Mutations". *ACM Trans. Graph.* 43.1 (Jan. 2024). ISSN: 0730-0301 3.

[SSC*17] SUN, WEILUN, SUN, XIN, CARR, NATHAN A, et al. "Gradient-Domain Vertex Connection and Merging." *EGSR (EI&I)* 83 (2017), 92 2.

[TCE05] TALBOT, JUSTIN, CLINE, DAVID, and EGBERT, PARRIS. "Importance Resampling for Global Illumination". *Eurographics Symposium on Rendering*. 2005, 139–146 3, 4.

[VG95] VEACH, ERIC and GUIBAS, LEONIDAS J. "Optimally Combining Sampling Techniques for Monte Carlo Rendering". *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*. 1995, 419–428 4.

[VG97] VEACH, ERIC and GUIBAS, LEONIDAS J. "Metropolis Light Transport". *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*. 1997, 65–76 2.

[WKL*23] WYMAN, CHRIS, KETTUNEN, MARKUS, LIN, DAQI, et al. "A Gentle Introduction to ReSTIR: Path Reuse in Real-time". *ACM SIGGRAPH 2023 Courses*. Los Angeles, California, Aug. 2023 3.

[WWWZ23] WANG, YU-CHEN, WYMAN, CHRIS, WU, LIFAN, and ZHAO, SHUANG. "Amortizing Samples in Physics-Based Inverse Rendering using ReSTIR". *ACM Trans. Graph.* 42.6 (2023), 214:1–214:17 3.

[YZN*22] YU, ZIHAN, ZHANG, CHENG, NOWROUZEZAHRAI, DEREK, et al. "Efficient Differentiation of Pixel Reconstruction Filters for Path-Space Differentiable Rendering". *ACM Trans. Graph.* 41.6 (2022), 191:1–191:16 10.

[YZYX24] YAN, DIFEI, ZHENG, SHAOKUN, YAN, LING-QI, and XU, KUN. "Filtering-Based Reconstruction for Gradient-Domain Rendering". *SIGGRAPH Asia 2024 Conference Papers*. SA '24. New York, NY, USA: Association for Computing Machinery, 2024. ISBN: 9798400711312 2.

[ZDDZ21] ZHANG, CHENG, DONG, ZHAO, DOGGETT, MICHAEL, and ZHAO, SHUANG. "Antithetic sampling for Monte Carlo differentiable rendering". *ACM Trans. Graph.* 40.4 (2021), 77:1–77:12 10.

[ZLK*24] ZHANG, SONG, LIN, DAQI, KETTUNEN, MARKUS, et al. "Area ReSTIR: Resampling for Real-Time Defocus and Antialiasing". *ACM Trans. Graph.* 43.4 (July 2024). ISSN: 0730-0301 1–4, 6, 8–10.