Compatibility-Guided Neighbor Selection for ReSTIR

ORION JUNKINS, ETH Zürich, Switzerland
MARKUS KETTUNEN, NVIDIA, Finland
DAQI LIN, NVIDIA, USA
RAVI RAMAMOORTHI, University of California, San Diego, USA and NVIDIA, USA
CHRIS WYMAN, NVIDIA, USA

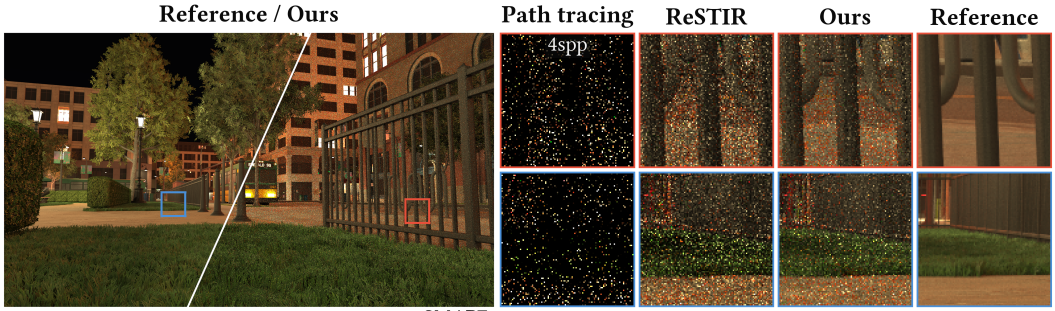


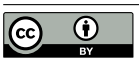
Fig. 1. Our method reduces ReSTIR's [Bitterli et al. 2020] variance and temporal covariance by picking the reused spatial neighbors from a larger pool of candidates proportional to our pixel compatibility heuristic.

Reservoir-based spatiotemporal importance resampling (ReSTIR) improves convergence of real-time Monte Carlo rendering by reusing samples between pixels and frames. Recent reservoir splatting and area ReSTIR methods improve ReSTIR's temporal robustness in challenging scenes, but much less work focuses on improving spatial reuse. We present a novel neighbor selection algorithm for ReSTIR's spatial reuse. Our simple method works with any pixel-space ReSTIR technique to improve image quality; we reduce SMAPE by 6–29% while also decreasing temporal covariance by 22–49%, all with a 2–5% incremental cost. We ground our work in an analysis of path compatibility between pixels and an empirical evaluation of the fundamental error-correlation tradeoff in ReSTIR; we improve the Pareto frontier of this tradeoff and provide tunable control over the balance between variance and correlation.

CCS Concepts: • **Computing methodologies** → **Computer graphics**.

ACM Reference Format:
Orion Junkins, Markus Kettunen, Daqi Lin, Ravi Ramamoorthi, and Chris Wyman. 2026. Compatibility-Guided Neighbor Selection for ReSTIR. *Proc. ACM Comput. Graph. Interact. Tech.* 9, 4, Article 52 (July 2026), 16 pages. https://doi.org/10.1145/3820024

Authors' Contact Information: Orion Junkins, ETH Zürich, Zurich, Switzerland, orionjunkins@gmail.com; Markus Kettunen, NVIDIA, Helsinki, Uusimaa, Finland, markus.kettunen@kapsi.fi; Daqi Lin, NVIDIA, Redmond, Washington, USA, daqil@nvidia.com; Ravi Ramamoorthi, University of California, San Diego, La Jolla, California, USA and NVIDIA, San Diego, California, USA, ravir@cs.ucsd.edu; Chris Wyman, NVIDIA, Redmond, Washington, USA, chris.wyman@acm.org.



ACM 2577-6193/2026/7-ART52
https://doi.org/10.1145/3820024

1 Introduction

With the advent of commodity ray tracing hardware [Kilgariff et al. 2018], ray and path tracing have gained traction in real-time game rendering. Even with this hardware, games are still limited to low per-pixel sample counts. With the slowing down of Moore's Law, improved rendering quality increasingly comes from improved rendering algorithms rather than more transistors.

Reservoir-based spatiotemporal importance resampling (ReSTIR) [Bitterli et al. 2020] allows games to render thousands of area lights with artifact-free shadows, often making shadow maps unnecessary. Developments in ReSTIR have enabled high-quality global illumination in scenes with both diffuse [Ouyang et al. 2021] and glossy [Lin et al. 2022] materials, all without baking or light probes. These improvements stem from ReSTIR's increase of effective sample count by reusing between pixels and frames.

ReSTIR's efficiency comes from amortization of costs: pixels sample paths independently, but then exchange both spatially and temporally, merging many paths into one with an improved distribution using generalized resampled importance sampling (GRIS) [Lin et al. 2022].

Scenes with complex light transport remain hard to sample well. ReSTIR's increased effective sample count helps, but it only acts as a sample count multiplier. When both sampling and path reuse are hard, ReSTIR's output becomes excessively correlated. Recent advances such as area ReSTIR [Zhang et al. 2024] and reservoir splatting [Liu et al. 2025] avoid temporal reuse failures near high-frequency geometry, reducing such correlation artifacts. But improving spatial reuse, despite its importance to ReSTIR's efficiency, has received relatively limited attention; Bitterli et al.'s [2020] original spatial neighbor selection still remains state of the art.

Bitterli et al. [2020] uniformly sample the reused neighbors within a given spatial radius, discarding those with highly dissimilar G-buffer normals and depths. As this reuses *all* sufficiently similar pixels, the number of neighbors tested is kept low to remain performant.

We present an improved, yet simple, spatial neighbor selection that works as a drop-in replacement. Due to performance limits on the number of samples we can feasibly reuse, we look to improve the *quality* of each neighbor. We perform a multi-tap spatial neighbor search, selecting only a subset to run through a full reuse pass. This subset is selected proportional to a continuous-valued generalization of Bitterli et al.'s [2020] binary rejection criteria that follows the form of denoiser edge-stopping functions [Dammertz et al. 2010; Schied et al. 2017].

Practical applications of ReSTIR [Wyman et al. 2023] often tweak ReSTIR for easier denoising. One approach, increasing the spatial reuse radius, reduces pre-denoised image quality even though the post-denoised result improves. This observation motivated our analysis of the relation between reuse radius, image error, and correlation. We find a fundamental tradeoff: with larger reuse radius, correlation decreases but error increases, and vice versa. This allows directly trading correlation for noise, but seeking to reduce *both* requires algorithmic improvements. Our neighbor selection heuristic moves this Pareto curve, allowing us to decrease both error and correlation (Figure 1).

To summarize, our contributions are as follows:

(1) A practical, geometry-based pixel compatibility heuristic for spatial resampling (Section 4).
(2) An improved spatial neighbor selection, compatible with ReSTIR DI [Bitterli et al. 2020], ReSTIR GI [Ouyang et al. 2021], ReSTIR PT [Lin et al. 2022], and more recent work (Section 5).
(3) Outlining ReSTIR's fundamental tradeoff between correlation and image error (Section 6), exploring how this affects selection of reuse parameters, and providing new recommended parameters given our improvements (Section 7).

2 Related Work

Bekaert et al. [2002] reduce path tracing noise by reusing paths from neighbors. Lehtinen et al. [2013] formalize *shift mappings* that map paths between pixels, with Jacobian determinants compensating for density change. Their shift maps reuse from adjacent pixels to evaluate discrete gradients for gradient-domain rendering. Kettunen et al. [2015] generalize gradient-domain rendering to path tracing, along with a new path tracing specific shift mapping. Bauszat et al. [2017] merge Bekaert et al.'s [2002] path reuse with shift mappings and gradient-domain rendering.

Manzi et al. [2014] optimize gradient-domain rendering by using G-buffer attributes like normal and depth to choose the pixel neighbors for discrete gradients, and Josse et al. [2025] do the same with neural networks. Dammertz et al. [2010] and Schied et al. [2017] use continuous-valued measures of geometric similarity as edge-stopping functions for denoising, combining position or depth terms with normal alignment to weight contributions from neighboring pixels. We use a similar function on G-buffer attributes to weight neighbor selection in ReSTIR's spatial reuse.

Bitterli et al. [2020] accelerate direct illumination by repeatedly reusing samples from close-by pixels and frames with resampled importance sampling (RIS) [Talbot et al. 2005], yielding the ReSTIR framework. Our work replaces their spatial neighbor selection.

Ouyang et al. [2021] apply ReSTIR to reuse rough global illumination between pixels by treating long paths as virtual point lights. Lin et al.'s [2022] GRIS gives mathematical foundations to ReSTIR and allows reusing paths with shift mappings. Their application of GRIS to path tracing yields ReSTIR PT. We run our experiments on ReSTIR PT, but our method also works with Bitterli et al.'s [2020] ReSTIR DI and Ouyang et al.'s [2021] ReSTIR GI.

Zhang et al. [2024] extend ReSTIR to integrate over pixel filters, effectively importance sampling subpixel locations. This stabilizes temporal reuse on high-frequency geometry by not rounding pixels' backward motion vectors. Liu et al. [2025] further stabilize temporal reuse by mapping prior-frame paths to the current frame following each path's *forward* motion vector. We augment ReSTIR PT with Zhang et al.'s [2024] area reservoirs and Liu et al.'s [2025] reservoir splatting in our experiments.

Further, Sawhney et al. [2024] reduce correlation artifacts with MCMC mutations, Hedstrom et al. [2025] apply ReSTIR to bidirectional path tracing, Kern et al. [2024] to photon mapping, and Werner et al. [2024] to sub-surface scattering. Zeng et al. [2025] use ReSTIR for path guiding. Our method is compatible with these improvements, but we do not experiment with them.

ReSTIR has further been extended to integrate depth of field [Zhang et al. 2024] and motion blur [Liu et al. 2025]. These extensions could make our G-buffer-based neighbor selection less effective.

Few papers focus on improving spatial reuse. Tokuyoshi [2023] temporally fits von Mises–Fischer (vMF) distributions to incident direct illumination and downweights spatial neighbors based on insufficient vMF overlap. Unlike our method, this solution is biased and does not generalize to global illumination. We use a simple G-buffer heuristic, but our neighbor selection is likely compatible with Tokuyoshi's [2023] continuous-valued heuristic. Salaün et al.'s [2025] histogram stratification decreases sampling noise by selecting spatial neighbors uniformly but antithetically, based on sample contribution. Their method is likely orthogonal to ours.

Hedstrom et al. [2026] focus reuse on contributing samples with *stochastic pairwise MIS*, improving quality in regions with poor inputs. They cluster pixels by G-buffer similarity and, using weighted reservoir sampling (WRS), select a cluster to reuse proportional to the confidence weights of clusters passing a binary G-buffer similarity test. We use WRS to select neighbors via a continuous-valued geometric compatibility score, but do not require clustering or stochastic pairwise MIS, though implementing these may be beneficial. Our focus is on practicality—we propose a simple improvement to spatial reuse that benefits any screen space ReSTIR algorithm.

3 Background

We briefly review the main concepts of GRIS [Lin et al. 2022] and ReSTIR, and point the reader to Wyman et al. [2023] for details.

3.1 Shift Mapping

Shift mappings map paths between domains for reuse. A shift map T from domain A to B may be undefined for some elements of A, but whenever T is defined, it must have an inverse. Technically, T is a bijection from a subset of A to its image, i.e., a partial bijection.

A common design goal for shift mappings is that they should aim to approximately retain the Jacobian-corrected path contribution, i.e.,

$$f(T(x))\,|T'(x)| \approx f(x). \tag{1}$$

The most common shift mappings used in ReSTIR are the reconnection shift [Lehtinen et al. 2013] and Lin et al.'s [2022] hybrid shift.

3.2 Unbiased Contribution Weight

Given a sample X, a weight W_X with expectation $\mathbb{E}\left[W_X \mid X = x\right] = 1/p_X(x)$ is called an unbiased contribution weight (UCW) [Lin et al. 2022]. It allows unbiased integration via $f(X)W_X$:

$$\mathbb{E}\left[f(X)W_X\right] = \mathbb{E}\left[\frac{f(X)}{p_X(X)}\right] = \int_{\operatorname{supp} X} f(x)\ \mathrm{d}x. \tag{2}$$

3.3 Generalized Resampled Importance Sampling

Given a target function $\hat{p}$ and samples $X_1, \ldots, X_M$ in a single domain, GRIS aggregates them into a representative $Y = X_s$ by resampling index s proportionally to resampling weights

$$w_i = m_i(X_i)\hat{p}(X_i)W_{X_i}, \tag{3}$$

where m_i is an MIS weight with $\sum_i m_i(x) = 1$. The output Y gets unbiased contribution weight

$$W_Y = \frac{1}{\hat{p}(Y)}\sum_{i=1}^{M} w_i, \tag{4}$$

enabling integration via $f(Y)W_Y \approx \int_\Omega f(x)\ \mathrm{d}x$. The returned sample Y is generally well distributed for the target function $\hat{p}$: by Equations 3 and 4, the single-sample integral estimate $\hat{p}(Y)W_Y$ $\left(\approx \int_\Omega \hat{p}(x)\ \mathrm{d}x\right)$ matches the multi-sample estimate using all the inputs:

$$\hat{p}(Y)W_Y = \sum_{i=1}^{M} w_i = \sum_{i=1}^{M} m_i(X_i)\hat{p}(X_i)W_{X_i}. \tag{5}$$

Choosing $\hat{p}(x) \approx f(x)$, e.g., the luminance of $f(x)$, yields well-distributed samples for f.

GRIS is often applied by reusing samples from varying domains via shift mappings: each $X_i \in \Omega_i$ is mapped into $Y_i = T_i(X_i) \in \Omega$ with shift mapping T_i. Resampling weights are then evaluated as

$$w_i = m_i(Y_i)\hat{p}(Y_i)\,|T'(X_i)|\,W_{X_i}, \tag{6}$$

where m_i is the MIS weight, and $|T'(X_i)|$ is the Jacobian determinant. GRIS then returns $Y = Y_s$ with UCW still given by Equation 4.

3.4 ReSTIR

ReSTIR is characterized by the following general sequence. A new frame is bootstrapped by each pixel producing *initial candidates* that are merged with GRIS to give that pixel's *initial sample*. This sample guarantees coverage of the target function's support, which is required for unbiased integration. In *temporal reuse*, prior-frame samples are shifted to the current frame and merged with the initial sample. In *spatial reuse*, each pixel selects a number of spatial neighbors and merges their temporal reuse outputs with its own.

At each stage, the sample returned by GRIS and its contribution weight are stored in a *reservoir*. Finally, the output of spatial reuse is evaluated via $f(Y)W_Y$. The sequence is repeated for every frame. The target function $\hat{p}$ is commonly set to the luminance of the path contribution f.

Our work focuses on improving spatial neighbor selection.

3.5 Spatial Neighbor Selection

In each frame, Bitterli et al.'s [2020] spatial reuse uniformly draws M pixels from a disk or box of radius R around the current pixel. Typically, fewer than 5 neighbors are chosen in the research literature, and only one in practical applications. The reuse radius is often 30 pixels.

As reusing from dissimilar radiance distributions can *increase* variance, Bitterli et al. [2020] discard neighbors suspected of having incompatible distributions. They use G-buffer depth and normal differences as proxies. This means *at most* M spatial neighbors are reused, and pixels whose similar neighbors are hard to find by uniform sampling may even get *no spatial reuse* at all.

Bitterli et al.'s [2020] specific neighbor acceptance conditions for neighbor pixel i are

$$\mathbf{n} \cdot \mathbf{n}_i \geq 0.5 \quad \text{and} \quad \frac{|d_i - d|}{d} < 0.1, \tag{7}$$

where $\mathbf{n}$ and d are the current pixel's normal and distance from the camera, and n_i and d_i those of pixel i. These conditions depend only on G-buffer values—not on individual samples or weights—and thus preserve GRIS's unbiasedness [Wyman et al. 2023].

We show how to improve these criteria and the overall neighbor selection process in Section 4.3.

4 Pixel Compatibility Heuristics

The criteria in Equation 7 are intuitive but informal and have been carried over from ReSTIR DI [Bitterli et al. 2020] to ReSTIR PT [Lin et al. 2022] without re-examination. We analyze the conditions under which path reuse succeeds for both direct and global illumination, and refine Equation 7 into a continuous-valued compatibility score.

To understand which pixels allow efficient path reuse by shift mapping, we reformulate the shift mapping design criterion (Equation 1) as

$$\frac{f(T(x))\,|T'(x)|}{f(x)} \approx 1 \tag{8}$$

and begin by studying direct illumination with the simple reconnection shift.

4.1 Direct Illumination

Given a direct illumination path $\bar{x} = [x_0, x_1, x_2]$, and its shifted counterpart $\bar{y} = T(\bar{x}) = [x_0, y_1, x_2]$ with x_0 the eye and x_2 an emitter (Figure 2, left), we substitute the path contribution[1]

$$f(\bar{x}) = \underbrace{f_s(x_0 \to x_1 \to x_2)}_{\text{BRDF}} \underbrace{G(x_1 \leftrightarrow x_2)}_{\text{geometry term}} \underbrace{V(x_1 \leftrightarrow x_2)}_{\text{visibility}} \underbrace{L_e(x_2 \to x_1)}_{\text{emission}} \tag{9}$$

[1]Technically, we parameterize primary hits by their image-space coordinates. This removes the sensor response W_e and yields $|T'(x)| = 1$ for reconnection. The choice of parameterization does not affect pixel compatibility.

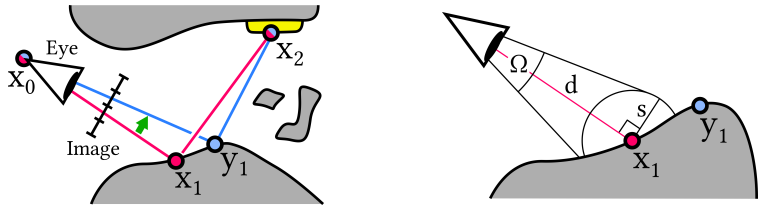


Fig. 2. **Left:** Paths $\bar{x}$ and $\bar{y} = T(\bar{x})$ (Section 4.1). **Right:** Our position heuristic's scale factor s (Section 4.3).

into Equation 8 for both $\bar{x}$ and $\bar{y}$ to give

$$\frac{f_s(x_0 \to y_1 \to x_2)}{f_s(x_0 \to x_1 \to x_2)} \frac{G(y_1 \leftrightarrow x_2)}{G(x_1 \leftrightarrow x_2)} \frac{V(y_1 \leftrightarrow x_2)}{V(x_1 \leftrightarrow x_2)} \frac{L_e(x_2 \to y_1)}{L_e(x_2 \to x_1)} |T'(\bar{x})| \approx 1. \tag{10}$$

As a general heuristic, the BSDF and geometry term ratios tend to diverge from 1 the farther y_1 is from x_1. Similarly, when y_1 is farther from x_1, the probability of losing visibility grows. With $|T'(\bar{x})| = 1$ for the reconnection shift, this suggests efficient reuse is more probable when the distance between x_1 and y_1 is not large. Bitterli et al.'s [2020] relative depth difference (Equation 7) tests whether x_1 and y_1 are roughly on the same camera-centric sphere, but this only partially controls the distance between x_1 and y_1.

Even if x_1 and y_1 are close, reuse may fail to be efficient. Assuming x_1 and y_1 are close enough, their directions from x_2 are very similar, often giving an emission ratio of approximately 1, and they are likely to share visibility to x_2, giving a visibility ratio of 1 (shadow edges are a notable exception [Tokuyoshi 2023]). With $x_1 \approx y_1$, the geometry term ratio becomes

$$\frac{G(y_1 \leftrightarrow x_2)}{G(x_1 \leftrightarrow x_2)} = \frac{\frac{(n_{y_1} \cdot \widehat{x_2 - y_1})(n_{x_2} \cdot \widehat{y_1 - x_2})}{\|y_1 - x_2\|^2}}{\frac{(n_{x_1} \cdot \widehat{x_2 - x_1})(n_{x_2} \cdot \widehat{x_1 - x_2})}{\|x_1 - x_2\|^2}} \approx \frac{n_{y_1} \cdot \widehat{x_2 - y_1}}{n_{x_1} \cdot \widehat{x_2 - x_1}} \approx \frac{n_{y_1} \cdot \widehat{x_2 - x_1}}{n_{x_1} \cdot \widehat{x_2 - x_1}}. \tag{11}$$

As secondary hit x_2 is sampled from x_1 and thus varies, we can easily say we should have $n_{x_1} \approx n_{y_1}$, but not much more. This condition aligns with Bitterli et al.'s [2020] normal constraint (Equation 7). With $x_1 \approx y_1$ and $n_{x_1} \approx n_{y_1}$, only the BSDF ratio in Equation 10 remains, suggesting limiting reuse where materials change from x_1 to y_1. Pixels with similar primary hit position, normal, and material are likely to yield efficient reuse of direct illumination path samples.

4.2 Global Illumination

For completeness, we generalize the argument to longer paths and Lin et al.'s [2022] hybrid shift, though the key insights do not change. Given $\bar{x} = [x_0, x_1, x_2, \dots, x_D]$ for $D \geq 3$ and its reconnection shift $\bar{y} = T(\bar{x}) = [x_0, y_1, x_2, \dots, x_D]$, substituting

$$f(\bar{x}) = \left(\prod_{k=1}^{D-1} f_s(x_{k-1} \to x_k \to x_{k+1}) G(x_k \leftrightarrow x_{k+1}) V(x_k \leftrightarrow x_{k+1}) \right) L_e(x_D \to x_{D-1}) \tag{12}$$

into Equation 8 for both $\bar{x}$ and $\bar{y}$ yields

$$\frac{f_s(x_0 \to y_1 \to x_2) G(y_1 \leftrightarrow x_2) V(y_1 \leftrightarrow x_2)}{f_s(x_0 \to x_1 \to x_2) G(x_1 \leftrightarrow x_2) V(x_1 \leftrightarrow x_2)} \frac{f_s(y_1 \to x_2 \to x_3) \cancel{G(x_2 \leftrightarrow x_3) \cdots}}{f_s(x_1 \to x_2 \to x_3) \cancel{G(x_2 \leftrightarrow x_3) \cdots}} |T'(\bar{x})| \approx 1, \tag{13}$$

where everything after the BSDF at x_2 cancels. The extra BSDF ratio at x_2 is also generally closer to 1 when x_1 and y_1 are close, and the same argument as above applies.

Lin et al.'s [2022] ReSTIR PT uses the hybrid shift, which postpones reconnection on specular surfaces by random replay. Hybrid shift's random-replayed vertices best match the originals if the primary hit positions, normals, and materials are similar. Regardless of shift mapping and path length, we want to select neighbors with similar primary hit position, normal, and material.

Our analysis also relates to recent temporal resampling advances such as reservoir splatting [Liu et al. 2025] and area ReSTIR's temporal reuse, which seek to *exactly* preserve primary hits across frames. We apply this concept to spatial resampling, explicitly seeking neighbors with similar primary hit position, normal, and material, to increase the probability of successful reuse.

4.3 Quantifying Compatibility

Bitterli et al.'s [2020] binary threshold helps cull damaging pixels. But if only borderline neighbors are proposed for reuse, a binary threshold may discard them all; we might prefer a slightly below-threshold neighbor over none at all. We instead opt to sample neighbors from a large pool, prioritizing reuse of neighbors with a higher heuristic compatibility score.

Our theory suggests a heuristic using primary hit position, normal, and material properties. In practice, we evaluate primary hits using G-buffer attributes. Guiding selection with G-buffer values—independent of the samples or weights used for integration—preserves GRIS's unbiasedness [Wyman et al. 2023]. Our heuristic follows the form of the edge-stopping functions of Dammertz et al. [2010] and Schied et al. [2017]: the product of normal and position compatibility terms.

Motivated by our analysis, we initially included a binary check for material consistency, but it provided no measurable benefit in our test scenes. We hypothesize that material changes strongly correlate with changes in position and normal, making an independent material term redundant. Explicitly handling material may still be beneficial in edge cases and is an interesting direction for future work.

Our heuristic for compatibility between primary hits x_1 and y_1 is

$$h(x_1, y_1) = h_p(x_1, y_1)\, h_n(x_1, y_1)\,, \tag{14}$$

with

$$h_p(x_1, y_1) = \exp\left(-\frac{\|x_1 - y_1\|}{s}\right), \qquad h_n(x_1, y_1) = \max\left(n_{x_1} \cdot n_{y_1}, 0\right)^{\beta}, \tag{15}$$

where x_1, y_1 are the primary hit positions, and n_{x_1}, n_{y_1} the primary hit normals from the G-buffer. Following Keller et al. [2016], $s = \sqrt{\Omega d^2/\pi}$ is the world-space radius corresponding to solid angle Ω at the current primary hit distance $d = \|x_1 - x_0\|$ (Figure 2, right). [2]

We get good results with $\Omega = 0.05$ and $\beta = 8$.

5 Compatibility-Guided Neighbor Selection

We now describe our spatial neighbor selection. We pick spatial neighbors from a pool of candidates, prioritizing proportional to the compatibility heuristic in Equation 14. In practice, we aim to always return M neighbors from an R-radius disk around the current pixel, given user-specified M and R.

With a typical radius of 30 pixels, naïvely evaluating per-pixel scores for the full disk is too costly. Instead, we evaluate Equation 14 at K random pixels in the radius R disk and resample M spatial neighbors from these K pixels, using weighted reservoir sampling (WRS) [Chao 1982]. We outline our method in Algorithm 1 and analyze the parameter choices in Section 6.

[2] We assume a perspective projection.

Algorithm 1: Compatibility-Guided Neighbor Selection

Input : G-buffer G, current pixel i, candidate count K, returned neighbor count M, search radius R
Output : M selected neighbors
Initialize WRS reservoir $\mathcal{R}$ of capacity M ; `// WRS initialization for M pixel indices`
for $k = 1$ ***to*** K **do**
  $j \leftarrow$ randomPixelFromDisk(i, R) ; `// Draw candidate neighbor`
  $w \leftarrow h(G[i], G[j])$; `// Heuristic compatibility score, Equation 14`
  $\mathcal{R}$.insert(j, w) ; `// Sample proportionally to compatibility`
end
return $\mathcal{R}$.chosenPixels ;

Weighted Reservoir Sampling. We use Chao's [1982] widely cited *A-Chao algorithm* for weighted reservoir sampling when selecting $M = 1$, as in prior ReSTIR papers [Bitterli et al. 2020]. When using $M > 1$, we use the *A-ES* algorithm [Efraimidis 2015], which guarantees the selection of M unique samples, given that the inputs are unique.

Early stopping. We track the number of high-quality candidates ($h(i, j) > 0.5$) and terminate early if the count exceeds M. This reduces runtime, particularly in simple scenes, with minimal effect on quality (see Section 7.5).

Disk sampling. We sample the K neighbor candidates from the disk by mapping a two-dimensional R_2 sequence with a randomized start index [Roberts 2018] through the concentric mapping [Shirley and Chiu 1997]. For performance reasons, our implementation precomputes 8192 random offsets.

6 Error–Correlation Tradeoff

With our spatial reuse in ReSTIR PT [Lin et al. 2022] using area reservoirs [Zhang et al. 2024] and reservoir splatting [Liu et al. 2025], we initially observed significantly improved image quality but also noticed an increase in correlation.

Practical implementations of ReSTIR sometimes increase their reuse radius to reduce output correlations. Despite increasing variance before denoising, this often improves final, denoised image quality when using denoisers not trained or designed to handle ReSTIR's correlations.

Motivated by this, we plot SMAPE [3] against covariance for different reuse radii (Figure 3; all scenes in supplement Figure S1). We observe a fundamental tradeoff between image error and covariance; changing the reuse radius allows trading covariance for error. At fixed radii, our neighbor selection sees increased covariance, due to more effective reuse. But increasing the spatial reuse radius lowers covariance while retaining significant quality improvement. We measure spatial covariance by averaging relative covariance within disks of radius 8 (see Appendix A).

We recommend resampling neighbors from a population of $8 \leq K \leq 32$ pixels; larger K increases cost with diminishing returns for quality. We find radii $30 \leq R \leq 100$ good compromises between error and correlation, i.e., at the elbow of our Pareto curves (Figure 3). This also validates Bitterli et al.'s [2020] empirical selection of $R = 30$, which regularly lies at the elbow of ReSTIR's Pareto curve before we improve neighbor selection.

7 Results

We implement our neighbor selection (Algorithm 1) into Liu et al.'s [2025] code that implements ReSTIR PT with reservoir splatting and builds on the Falcor framework [Kallweit et al. 2022].

[3] We use $\text{SMAPE}(I) = 100\,\text{mean}_i\left(\frac{|I_i - I_i^{\text{gt}}|}{|I_i| + |I_i^{\text{gt}}| + \varepsilon}\right)$, where $|\cdot|$ is luminance and I^{gt} is the ground truth. We use $\varepsilon = 10^{-4}$.

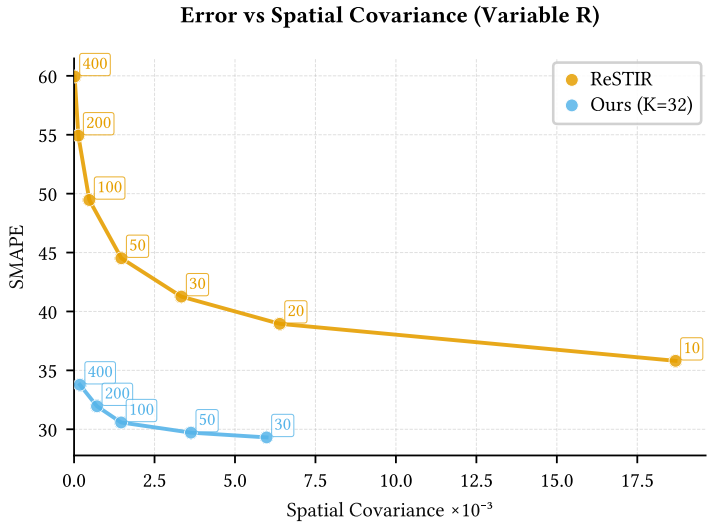


Fig. 3. Error vs. covariance at different reuse radii in Emerald Square. The balance between error and covariance can be controlled by reuse radius (numbers at points), but breaching the Pareto frontier (orange) requires an algorithmic change. Our compatibility-guided spatial neighbor selection (cyan) improves the Pareto frontier and allows improved error–covariance tradeoffs (see supplement Figure S1 for all scenes).

Our neighbor selection runs in a shader pass before spatial reuse, and we render at 1920×1080 on an NVIDIA RTX 5880 Ada with $M = 1$ spatial neighbor. We compare our results to Liu et al.'s [2025] method with Bitterli et al.'s [2020] spatial neighbor selection, but otherwise use identical configurations. We study the following configurations for our method (see Appendix B for $M \geq 2$):

- Fast ($R = 30, K = 32$),
- Balanced ($R = 50, K = 32$),
- Conservative ($R = 100, K = 32$).

All three variants use $K = 32$, with R tuned to select a desired operating point on the error-correlation tradeoff. Runtime differences arise due to reduced cache coherence at larger radii (see Section 7.3). We report costs averaging ten trials over 2048-frame sequences. Our quality metrics include SMAPE to evaluate variance, relative autocovariance (Appendix A) with a three-frame lag to measure temporal correlations, and average relative covariance (Appendix A) over 8-pixel disks to quantify spatial correlations. We cap ReSTIR confidence at 20 and warm up for 40 frames before capturing data or images. We demonstrate our results on four scenes: Emerald Square, Bistro Exterior, Sun Temple, and Veach Ajar. All results in the paper use a static camera; see the supplemental video for animations.

Easy and hard pixels. Renders often include regions where quality spatial neighbors are hard to find. These areas suffer the most without effective spatial reuse, yet full-frame metrics often fail to clearly measure changes in such pixels. We rank pixels by the ratio of neighbors within Bitterli et al.'s [2020] 30-pixel reuse radius surviving the compatibility test (Equation 7), and classify the most challenging 10% as *hard*. We run separate evaluations on just these hard pixels.

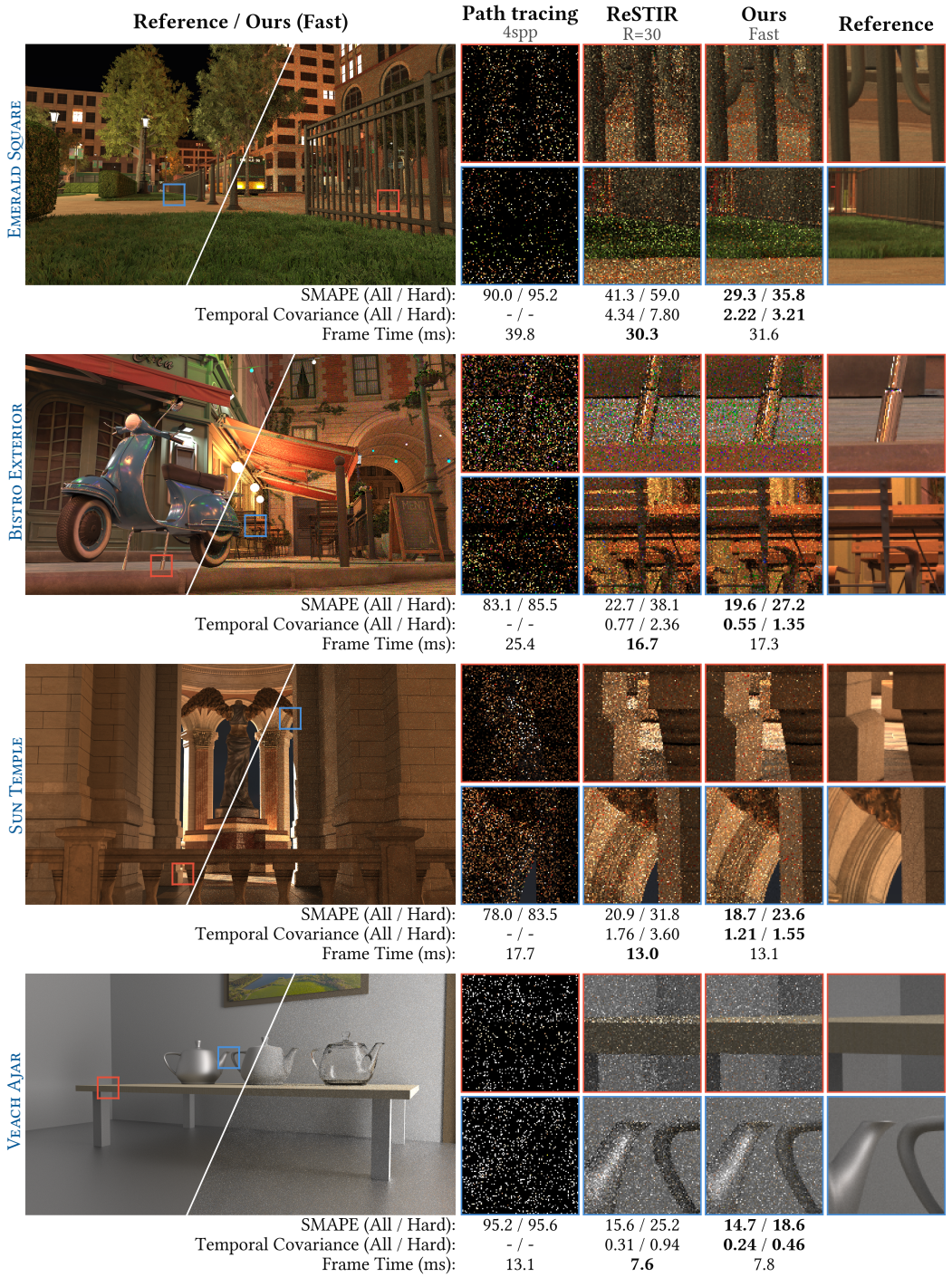


Fig. 4. Our compatibility-guided neighbor selection reduces both error and temporal covariance with little runtime overhead. In pixels surrounded by complex or rapidly changing geometry (foliage, grazing angles, fine detail, etc.), the assumptions of prior work break down – pixel space proximity is a poor proxy for domain compatibility. Our selection algorithm improves the quality of reuse in these pixels.

7.1 Reduction in Noise and Temporal Correlation

We first study our spatial reuse with the Fast configuration against Bitterli et al.'s [2020] neighbor selection in Figure 4. We consistently improve both noise and temporal covariance across scenes, with only a small increase in cost. The improvement is more pronounced for hard pixels.

In Emerald Square, where reuse heuristics are especially important for effective reuse due to high geometric complexity, improvement is large (29% reduction in SMAPE, 49% in temporal covariance). In simpler scenes (Bistro Exterior, Sun Temple, Veach Ajar) where identifying good neighbors is fairly straightforward, our improvements have a smaller effect, giving a reduction of 5.8–13.7% in SMAPE and 22–31% in temporal covariance. Looking at only hard pixels reveals more nuance: even simpler scenes feature numerous pixels that benefit from improved reuse, giving a 26–29% reduction in SMAPE and a 43–57% reduction in temporal covariance over hard pixels.

Our approach increases the stability of spatial reuse in both easy and hard scenes, especially in regions prone to correlation artifacts due to challenging reuse.

Largest improvement near geometric complexity. In Figure 5, we study how geometric complexity impacts reuse using a spatiotemporal (x, t) plot. Bitterli et al.'s [2020] neighbor selection already works well in regions with similar neighborhoods (top, second from right), but fails for pixels surrounded by geometric variation (bottom, second from right). Our neighbor selection improves spatial reuse especially when compatible spatial neighbors are hard to find (bottom, right).

Spatial reuse decreases temporal covariance. Figure 5 also explains why improved spatial reuse decreases temporal covariance: successful spatial reuse decreases the relative weight of each individual initial sample showing up as a vertical stripe in the (x, t) plot. Our neighbor selection makes spatial reuse more effective, speeding up the mixing of the initial samples.

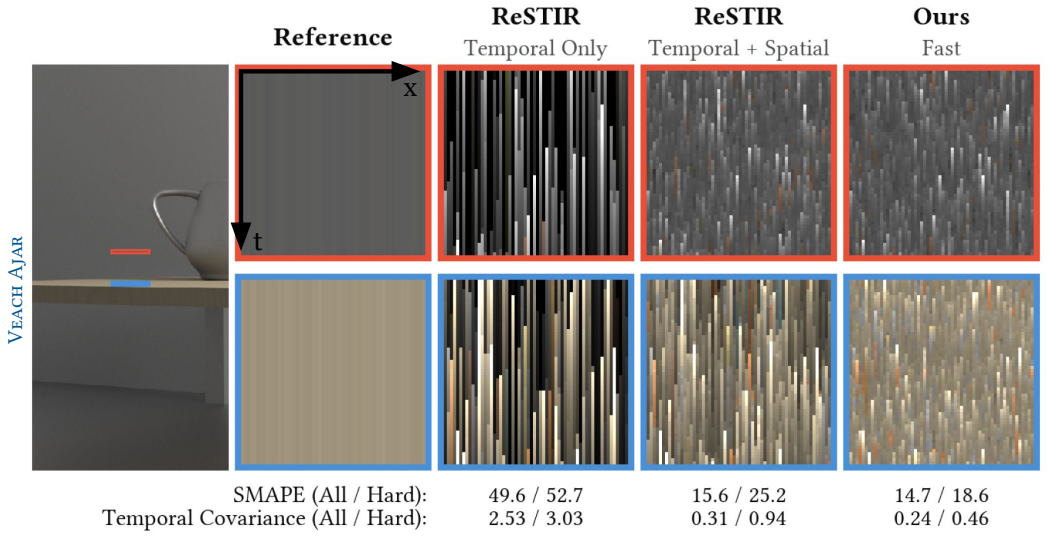


Fig. 5. Two pixel strips visualized over time as an (x, t) plot. Each row shows a different video frame. Vertical streaks indicate temporal correlations; horizontal blobs indicate spatial correlations; noise indicates neither. Temporal reuse introduces temporal correlations. Spatial resampling disrupts them, but prior work degrades near rapidly changing geometry or discontinuities (bottom). We reuse better in such cases (bottom right).

Table 1. Per-pass GPU runtime (ms) for two scenes. Bold passes contain incremental costs from our approach.

	Emerald Square				*Veach Ajar*			
	ReSTIR†	Ours^{F}	Ours^{B}	Ours^{C}	ReSTIR†	Ours^{F}	Ours^{B}	Ours^{C}
Initial Sampling	16.0	16.2	16.4	16.3	2.5	2.5	2.5	2.6
Temporal Resampling	7.5	7.3	7.4	7.3	2.4	2.4	2.4	2.4
Neighbor Selection	—	0.3	0.5	0.7	—	0.1	0.2	0.3
Spatial Reuse	5.4	6.5	6.7	6.9	2.1	2.1	2.1	2.1
Full Frame	30.3	31.6	32.2	32.5	7.6	7.8	7.8	8.0

† ReSTIR with $R=30$ ^{F} Fast ($K=32$, $R=30$) ^{B} Balanced ($K=32$, $R=50$) ^{C} Conservative ($K=32$, $R=100$)

7.2 Runtime Cost

Our method incurs a total 1.3–4.4% overhead for Fast, 2.0–6.4% for Balanced, and 4.3–7.4% for Conservative. Often, a minority of this overhead comes from the neighbor selection pass with the G-buffer taps; we measure an overhead of 0.1ms – 0.3ms or 1.1% – 1.7% for Fast.

In scenes like Emerald Square, with large regions of high complexity (e.g., foliage), Bitterli et al.'s [2020] neighbor selection often discards all spatial neighbors as incompatible, effectively disabling spatial reuse and finishing the frame faster. Our neighbor selection fixes these reuse failures and pays the (previously skipped) cost of spatial reuse; in our experiments, this overhead is 0ms – 1.0ms or 0% – 3.4% for Fast.

We show the full performance breakdown for Emerald Square and Veach Ajar in Table 1.

7.3 Effect of Parameters

We study the effect of reuse radius R and neighbor candidate count K to SMAPE, temporal covariance, spatial covariance and frame time in Table 2.

Larger R increases the population size. This decreases spatial correlations, but also reduces spatial reuse efficiency, slightly increasing SMAPE and temporal covariance. Larger R also reduces cache coherence when sampling the G-buffer, increasing frame time.

Increasing K shapes the sampling distribution towards the heuristic score. This improves spatial reuse efficiency, reducing SMAPE and temporal covariance. But a more converged distribution focuses reuse on a smaller set of neighbors with higher scores, increasing spatial covariance. This increase can be undone by *also* increasing R. The additional G-buffer taps increase frame time.

We choose $K = 32$ as a practical balance between SMAPE, correlations and cost. Fast provides lower frame time, Conservative minimizes spatial and temporal correlations at a slight cost to SMAPE and frame time, and Balanced offers a midpoint.

Table 2. All-pixel SMAPE / Temporal Cov. / Spatial Cov. $\times 10^{-3}$ / Frame Time (ms) in Emerald Square.

	$K = 8$	$K = 32$	$K = 128$
$R = 30$	29.9 / 2.3 / 4.6 / 31.5	29.3 / 2.2 / 6.0 / 31.6^{F}	29.3 / 2.2 / 6.6 / 32.1
$R = 50$	30.7 / 2.4 / 2.1 / 32.0	29.7 / 2.3 / 3.6 / 32.2^{B}	29.5 / 2.2 / 4.2 / 32.4
$R = 100$	32.0 / 2.5 / 0.8 / 32.4	30.6 / 2.4 / 1.5 / 32.5^{C}	30.0 / 2.3 / 2.4 / 33.7

^{F} Fast ^{B} Balanced ^{C} Conservative

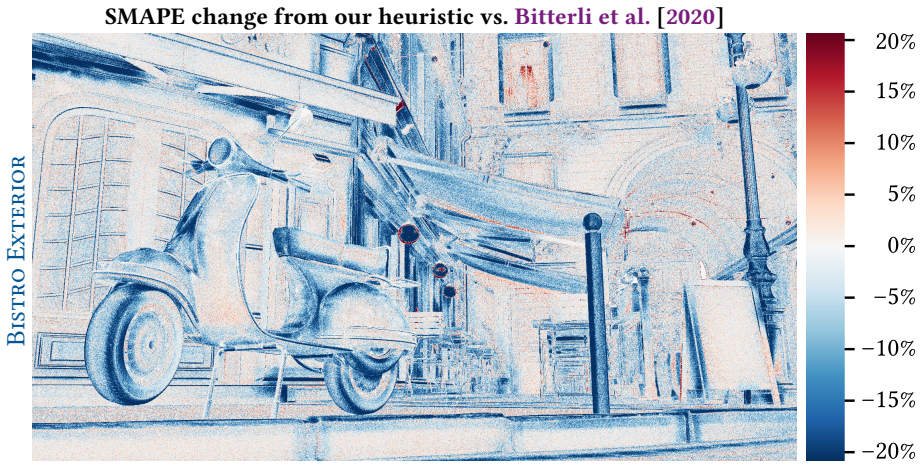


Fig. 6. Importance of our pixel compatibility heuristic (Equation 14). We compare our method against a variant that picks uniformly from pixels surviving Bitterli et al.'s [2020] threshold (Equation 7). Using our continuous-valued heuristic often significantly decreases SMAPE outside of large flat areas.

7.4 Effect of Continuous-Valued Heuristic

In Figure 6, we study the importance of our similarity heuristic (Equation 14). We compare our multi-tap spatial search (Section 5) using our similarity heuristic against our multi-tap search using Bitterli et al.'s [2020] binary-valued heuristic (Equation 7). Our heuristic often significantly reduces SMAPE in areas with fast geometric variation, with smaller improvements near large flat surfaces.

The figure also shows a minority of small areas where our heuristic performs worse. For example, in the top window, our heuristic assigns a non-zero weight to reuse between the room and the building facade, which the binary-valued heuristic rejects. We leave adaptive tuning of our heuristic's strictness for future work.

7.5 Effect of Early Stopping

Early stopping reduces runtime with minimal effect on quality ($< 1\%$ relative change in SMAPE across all test scenes). Table 3 compares the runtime of neighbor selection with early stopping disabled and enabled across $K \in \{8, 32, 128\}$ with $R = 30$.

Time savings from early stopping are larger in simpler scenes like Veach Ajar, as the ease of finding compatible neighbors increases the likelihood of triggering early stopping. Time savings are also greater with larger K, as the cost without early stopping grows, yet the chance of early stopping increases. While early stopping is normally quite effective, the worst-case frame time can be an important consideration in interactive applications, which may suggest keeping K moderate.

Table 3. Neighbor selection GPU runtime (ms) with early stopping disabled and enabled for two scenes.

	Emerald Square			*Veach Ajar*		
	$K = 8$	$K = 32$	$K = 128$	$K = 8$	$K = 32$	$K = 128$
Early Stopping Disabled	0.27	0.88	3.35	0.28	0.90	3.31
Enabled	0.20	0.36	0.76	0.15	0.15	0.19

8 Limitations and Future Work

Our neighbor selection heuristic does not address disocclusions. Additional factors could be added to our heuristic, e.g., based on confidence weights, to favor neighbors with stable temporal histories. Alternatively, our approach could be combined with Hedstrom et al.'s [2026] stochastic pairwise MIS to unbiasedly select neighbors based on both geometric compatibility and pixel contribution.

Our neighbor selection heuristic also does not consider visibility, and may select a neighbor on the other side of a shadow edge, which is a known source of noise in ReSTIR [Tokuyoshi 2023]. Tokuyoshi's [2023] vMF similarity provides one solution for direct light, but extensions for multi-bounce paths are an interesting direction for future work.

More broadly, our heuristic is based on a G-buffer, and as such inherits its fundamental issues. When integrating over lens [Zhang et al. 2024] or time [Liu et al. 2025], a single G-buffer value may not adequately represent the full path space behind a pixel. Storing multiple G-buffer values at different time or lens coordinates could help.

Lin et al. [2021] extend ReSTIR to volumes, but it is not clear what should be stored in the G-buffer in this case. One idea could be to store primary hits at a fixed transmittance value.

9 Conclusion

We propose a simple drop-in replacement for Bitterli et al.'s [2020] spatial neighbor selection, based on searching *better quality* neighbors with multiple G-buffer taps and picking proportionally to a geometry-based similarity heuristic. By replacing Bitterli et al.'s [2020] binary compatibility threshold with our continuous-valued score, we enable spatial reuse from the best available neighbors, rather than giving up when above-threshold neighbors are not found. As suggested by our analysis, our heuristic is based on primary hit closeness rather than depth difference in prior work.

Our method improves both error and correlation in ReSTIR. Improvement is largest where Bitterli et al.'s [2020] neighbor selection fails to find reusable neighbors, effectively disabling reuse. Our method has a small overhead, and most of it often comes from enabling reuse in these pixels.

Acknowledgments

We would like to thank Aaron Lefohn for supporting this research.

References

Pablo Bauszat, Victor Petitjean, and Elmar Eisemann. 2017. Gradient-domain path reusing. *ACM Trans. Graph.* 36, 6, Article 229 (Nov. 2017), 9 pages. doi:10.1145/3130800.3130886

Philippe Bekaert, Mateu Sbert, and John H. Halton. 2002. Accelerating path tracing by re-using paths. In *Proceedings of the 13th Eurographics Workshop on Rendering*. Eurographics Association, 125–134. doi:10.2312/EGWR/EGWR02/125-134

Benedikt Bitterli, Chris Wyman, Matt Pharr, Peter Shirley, Aaron Lefohn, and Wojciech Jarosz. 2020. Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting. *ACM Trans. Graph.* 39, 4, Article 148 (Aug. 2020), 17 pages. doi:10.1145/3386569.3392481

Min-Te Chao. 1982. A general purpose unequal probability sampling plan. *Biometrika* 69, 3 (1982), 653–656. doi:10.1093/biomet/69.3.653

Holger Dammertz, Daniel Sewtz, Johannes Hanika, and Hendrik P. A. Lensch. 2010. Edge-avoiding à-trous wavelet transform for fast global illumination filtering. In *High Performance Graphics*, Michael Doggett, Samuli Laine, and Warren Hunt (Eds.). Eurographics Association, 67–75. doi:10.2312/EGGH/HPG10/067-075

Pavlos S. Efraimidis. 2015. Weighted random sampling over data streams. In *Algorithms, Probability, Networks, and Games*, Christos Zaroliagis, Grammati Pantziou, and Spyros Kontogiannis (Eds.). Lecture Notes in Computer Science, Vol. 9295. Springer, Cham, 183–195. doi:10.1007/978-3-319-24024-4_12

Trevor Hedstrom, Markus Kettunen, Daqi Lin, Chris Wyman, and Tzu-Mao Li. 2025. ReSTIR BDPT: Bidirectional ReSTIR path tracing with caustics. *ACM Trans. Graph.* 44, 5, Article 169 (Sept. 2025), 16 pages. doi:10.1145/3744898

Trevor Hedstrom, Markus Kettunen, Daqi Lin, Chris Wyman, and Tzu-Mao Li. 2026. Stochastic pairwise MIS for unbiased large-kernel reuse in real-time. *Comput. Graph. Forum* 45, 2 (May 2026). doi:10.1111/cgf.70391

Matthieu Josse, Joey Litalien, and Adrien Gruson. 2025. Adaptive neural kernels for gradient-domain rendering. In *SA Conference Papers '25*. ACM, Article 67, 11 pages. doi:10.1145/3757377.3763920

Simon Kallweit, Petrik Clarberg, Craig Kolb, Tomáš Davidovič, Kai-Hwa Yao, Theresa Foley, Yong He, Lifan Wu, Lucy Chen, Tomas Akenine-Möller, Chris Wyman, Cyril Crassin, and Nir Benty. 2022. The Falcor rendering framework. https://github.com/NVIDIAGameWorks/Falcor

Alexander Keller, Ken Dahm, and Nikolaus Binder. 2016. Path space filtering. In *Monte Carlo and Quasi-Monte Carlo Methods*, Ronald Cools and Dirk Nuyens (Eds.). Springer Proceedings in Mathematics & Statistics, Vol. 163. Springer, Cham, 423–436. doi:10.1007/978-3-319-33507-0_21

René Kern, Felix Brüll, and Thorsten Grosch. 2024. ReSTIR FG: Real-time reservoir resampled photon final gathering. In *Eurographics Symposium on Rendering*, Eric Haines and Elena Garces (Eds.). Eurographics Association. doi:10.2312/sr.20241155

Markus Kettunen, Marco Manzi, Miika Aittala, Jaakko Lehtinen, Frédo Durand, and Matthias Zwicker. 2015. Gradient-domain path tracing. *ACM Trans. Graph.* 34, 4, Article 123 (July 2015), 13 pages. doi:10.1145/2766997

Emmett Kilgariff, Henry Moreton, Nick Stam, and Brandon Bell. 2018. NVIDIA Turing architecture in-depth. https://developer.nvidia.com/blog/nvidia-turing-architecture-in-depth/ Accessed: 31 March 2026.

Jaakko Lehtinen, Tero Karras, Samuli Laine, Miika Aittala, Frédo Durand, and Timo Aila. 2013. Gradient-domain Metropolis light transport. *ACM Trans. Graph.* 32, 4, Article 95 (July 2013), 12 pages. doi:10.1145/2461912.2461943

Daqi Lin, Markus Kettunen, Benedikt Bitterli, Jacopo Pantaleoni, Cem Yuksel, and Chris Wyman. 2022. Generalized resampled importance sampling: Foundations of ReSTIR. *ACM Trans. Graph.* 41, 4, Article 75 (July 2022), 23 pages. doi:10.1145/3528223.3530158

Daqi Lin, Chris Wyman, and Cem Yuksel. 2021. Fast volume rendering with spatiotemporal reservoir resampling. *ACM Trans. Graph.* 40, 6, Article 279 (Dec. 2021), 18 pages. doi:10.1145/3478513.3480499

Jeffrey Liu, Daqi Lin, Markus Kettunen, Chris Wyman, and Ravi Ramamoorthi. 2025. Reservoir splatting for temporal path resampling and motion blur. In *SIGGRAPH Conference Papers '25*. ACM, Article 35, 11 pages. doi:10.1145/3721238.3730646

Marco Manzi, Fabrice Rousselle, Markus Kettunen, Jaakko Lehtinen, and Matthias Zwicker. 2014. Improved sampling for gradient-domain Metropolis light transport. *ACM Trans. Graph.* 33, 6, Article 178 (Nov. 2014), 12 pages. doi:10.1145/2661229.2661291

Yaobin Ouyang, Shiqiu Liu, Markus Kettunen, Matt Pharr, and Jacopo Pantaleoni. 2021. ReSTIR GI: Path resampling for real-time path tracing. *Comput. Graph. Forum* 40, 8 (Nov. 2021), 17–29. doi:10.1111/cgf.14378

Martin Roberts. 2018. The unreasonable effectiveness of quasirandom sequences. https://extremelearning.com.au/unreasonable-effectiveness-of-quasirandom-sequences/ Accessed: 5 April 2026.

Corentin Salaün, Martin Bálint, Laurent Belcour, Eric Heitz, Gurprit Singh, and Karol Myszkowski. 2025. Histogram stratification for spatio-temporal reservoir sampling. In *SIGGRAPH Conference Papers '25*. ACM, Article 83, 10 pages. doi:10.1145/3721238.3730723

Rohan Sawhney, Daqi Lin, Markus Kettunen, Benedikt Bitterli, Ravi Ramamoorthi, Chris Wyman, and Matt Pharr. 2024. Decorrelating ReSTIR samplers via MCMC mutations. *ACM Trans. Graph.* 43, 1, Article 10 (Jan. 2024), 15 pages. doi:10.1145/3629166

Christoph Schied, Anton Kaplanyan, Chris Wyman, Anjul Patney, Chakravarty R. Alla Chaitanya, John Burgess, Shiqiu Liu, Carsten Dachsbacher, Aaron Lefohn, and Marco Salvi. 2017. Spatiotemporal variance-guided filtering: real-time reconstruction for path-traced global illumination. In *Proceedings of High Performance Graphics*. ACM, Article 2, 12 pages. doi:10.1145/3105762.3105770

Peter Shirley and Kenneth Chiu. 1997. A low distortion map between disk and square. *Journal of Graphics Tools* 2, 3 (Jan. 1997), 45–52. doi:10.1080/10867651.1997.10487479

Justin Talbot, David Cline, and Parris Egbert. 2005. Importance resampling for global illumination. In *Eurographics Symposium on Rendering (2005)*. Eurographics Association, 139–146. doi:10.2312/EGWR/EGSR05/139-146

Yusuke Tokuyoshi. 2023. Efficient spatial resampling using the PDF similarity. *Proc. ACM Comput. Graph. Interact. Tech.* 6, 1, Article 4 (May 2023), 19 pages. doi:10.1145/3585501

Mirco Werner, Vincent Schüßler, and Carsten Dachsbacher. 2024. ReSTIR subsurface scattering for real-time path tracing. *Proc. ACM Comput. Graph. Interact. Tech.* 7, 3, Article 36 (Aug. 2024), 19 pages. doi:10.1145/3675372

Chris Wyman, Markus Kettunen, Daqi Lin, Benedikt Bitterli, Cem Yuksel, Wojciech Jarosz, Pawel Kozlowski, and Giovanni De Francesco. 2023. A gentle introduction to ReSTIR: Path reuse in real-time. In *ACM SIGGRAPH 2023 Courses*. ACM, Article 1, 38 pages. doi:10.1145/3587423.3595511

Zheng Zeng, Markus Kettunen, Chris Wyman, Lifan Wu, Ravi Ramamoorthi, Ling-Qi Yan, and Daqi Lin. 2025. ReSTIR PG: Path guiding with spatiotemporally resampled paths. In *SA Conference Papers '25*. ACM, Article 12, 11 pages. doi:10.1145/3757377.3763813

Song Zhang, Daqi Lin, Markus Kettunen, Cem Yuksel, and Chris Wyman. 2024. Area ReSTIR: Resampling for real-time defocus and antialiasing. *ACM Trans. Graph.* 43, 4, Article 98 (July 2024), 13 pages. doi:10.1145/3658210

A Quantifying Correlations

To robustly quantify spatial correlations, we evaluate mean radial covariance [Sawhney et al. 2024], but apply it on scalar luminance images normalized by the ground truth. As in RelMSE and MAPE, this normalization stops brighter pixels from dominating the mean over the image. More concretely, we estimate relative covariance between pixels i and j by averaging N independent trials:

$$r_{ij} = \frac{1}{N-1} \sum_{n=1}^{N} \left(\frac{I_{ni} - \bar{I}_i}{\bar{I}_i + \varepsilon} \right) \left(\frac{I_{nj} - \bar{I}_j}{\bar{I}_j + \varepsilon} \right). \tag{16}$$

Here, I is the grayscale luminance, $\bar{I}$ is the grayscale sample mean, and we use $\varepsilon = 10^{-4}$. We average r_{ij} over a disk of radius 8 around i to obtain the radial relative covariance. We average the radial relative covariance over the image and call it *average relative covariance* for simplicity.

We measure temporal co-variation by relative autocovariance. We collect N independent video sequences in a static scene and measure autocovariance through a lag of $T = 3$ frames:

$$a_i^{(t)} = \frac{1}{N-1} \sum_{n=1}^{N} \left(\frac{I_{ni}^{(t)} - \bar{I}_i^{(t)}}{\bar{I}_i^{(t)} + \varepsilon} \right) \left(\frac{I_{ni}^{(t+T)} - \bar{I}_i^{(t+T)}}{\bar{I}_i^{(t+T)} + \varepsilon} \right). \tag{17}$$

Here, $I^{(t)}$ is the grayscale image of frame t. Averaging these $a_i^{(t)}$ over all pixels and frames gives the relative autocovariance. We observe similar results with $1 \leq T \leq 5$.

B More Spatial Neighbors

Our spatial neighbor selection can also be applied to spatial resampling from $M \geq 2$ spatial neighbors. Figure 7 compares our error-covariance tradeoff to Bitterli et al.'s [2020] neighbor selection for $M \in \{1, 2, 3\}$. We find our improvements generalize to larger M, and the same configurations Fast, Balanced and Conservative may be used.

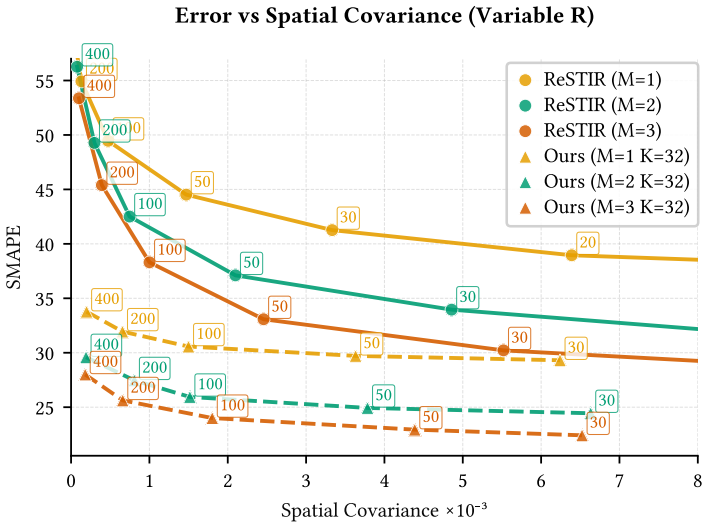


Fig. 7. Error vs. covariance at different reuse radii in Emerald Square across $M \in \{1, 2, 3\}$. Our approach also improves the Pareto frontier of multi-neighbor spatial reuse. Returns diminish for $M > 3$, suggesting spatial neighbor count should be kept relatively low.