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

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

Stochastic Pairwise MIS for Unbiased Large-Kernel Reuse in Real-Time

Trevor Hedstrom[1], Markus Kettunen[2], Daqi Lin[3], Chris Wyman[3] and Tzu-Mao Li[1]

[1]University of California San Diego, USA
[2]NVIDIA, Finland
[3]NVIDIA, USA

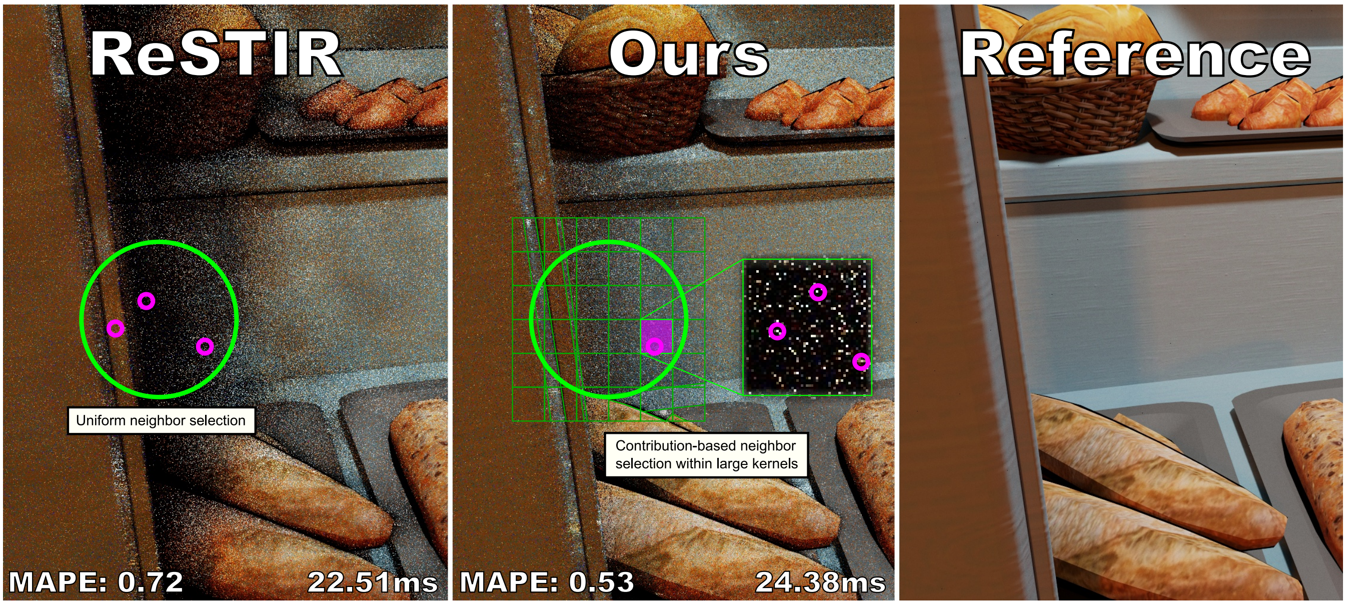

ReSTIR
Uniform neighbor selection
MAPE: 0.72
22.51ms
Ours
Contribution-based neighbor selection within large kernels
MAPE: 0.53
24.38ms
Reference

Figure 1: Bistro Exterior *rendered with ReSTIR and our method. Camera movement from left to right reveals a region behind the window bar with no pixel correspondences in the prior frame. Lacking temporal reuse, this disocclusion region contains heavy noise, which tone mapping turns into darkness. Standard ReSTIR recovers slowly due to picking input pixels for spatial reuse completely at random, often attempting reuse from reservoirs without a contributing path. Our novel stochastic pairwise MIS allows focusing reuse to reservoirs with contributing samples, significantly increasing the efficiency of spatial reuse.*

Abstract

Spatiotemporal resampling methods such as ReSTIR decrease noise in Monte Carlo rendering of dynamic content by reusing paths across frames and pixels. Standard ReSTIR reuses spatially from a small number of randomly selected neighbors. This reuse suffers when few neighbors contain contributing samples, reducing quality toward that of the underlying path sampler. This commonly occurs during camera or object motion, as regions not present in prior frames are revealed. Increasing the number of spatial neighbors helps but also increases cost. We propose a novel spatial neighbor selection technique, stochastic pairwise MIS, which enables unbiased reuse from many neighbors in real time and focuses reuse on pixels with contributing samples. This provides a significant increase in image quality overall, especially in regions with poor input samples.

CCS Concepts

• ***Computing methodologies*** → ***Ray tracing;***

Keywords: Global Illumination, Resampling, Multiple Importance Sampling

1. Introduction

Rendering algorithms often spatially reuse light paths across pixels to improve the sampling efficiency [BSH02]. ReSTIR (Reservoir-based Spatio-Temporal Importance Resampling) amortizes path sampling over space and time by reusing paths between pixels in the current (spatial reuse) and previous (temporal reuse) animation frames. ReSTIR methods first generate per-pixel samples using a standard sampling algorithm, e.g., path tracing. Then, resampling is applied to each pixel to randomly select from samples found by other pixels in the first step. Since reusing samples from other pixels is generally much cheaper than generating new ones, ReSTIR significantly improves sampling efficiency.

Spatial reuse is a fundamental component of the ReSTIR method [WKL*23]. Existing methods that employ spatial reuse are effective, especially when combined with temporal reuse. However, when the rendering lacks temporal coherency, e.g., due to disocclusion, methods like ReSTIR suffer from increased noise (Figure 1).

In this paper, we propose a spatial reuse solution for real-time ReSTIR that significantly improves the sampling efficiency, especially under camera or scene motion. We observe that spatial reuse is ineffective when the samples are sparsely populated in the image. Hence, we can potentially significantly improve the efficiency of spatial reuse by reusing from a large neighborhood. Naïvely generalizing spatial reuse to a large neighborhood, however, would not improve efficiency, as the evaluation cost increases significantly. Instead, we propose a stochastic spatial reuse scheme that unbiasedly selects from the neighbor samples with high contributions.

We propose a novel stochastic resampling multiple importance sampling (MIS) weight on top of the generalized resampled importance sampling (GRIS) method [LKB*22] to enable large-scale and unbiased spatial reuse. We show that applying our method to Lin et al.'s pairwise MIS weight leads to a computationally efficient estimator that frees us from needing to evaluate all the neighbors for each pixel. We further propose an efficient GPU implementation of our method to efficiently select neighbors.

In summary, our technical contributions are:

- A mathematical theory for efficient unbiased sample reuse from a large number of pixels with stochastic resampling MIS weights.
- A stochastic formulation of pairwise resampling MIS.
- A method of efficient spatial reuse of only initial candidates *contributing* to other pixels.
- A method to efficiently spatially reuse from many pixels.
- An efficient GPU implementation of our stochastic pairwise MIS estimator.

1.1. Related Work

Our work follows the line of light transport simulation research to *reuse* light paths across pixels in an unbiased manner. One of the earliest examples [BSH02] connects paths generated by one pixel to a vertex of paths generated by other pixels, and combines the contributions using multiple importance sampling [VG95]. Similar ideas show up in virtual point light rendering [Kel97, SIMP06, DKH*10] and bidirectional path tracing [PRDD15, NID20, SLW22].

Table 1: *Summary of notation.*

$[P]$	Iverson bracket; 1 if P is true, 0 otherwise
X_i	Input sample i
W_X	Unbiased Contribution Weight (UCW) for a sample X
T_i	Shift mapping from domain i to the target domain
Y_i	X_i shifted into target domain; $T_i(X_i)$
M	Total number of reuse candidates
$\tilde{N}$	Number of reuse candidates to evaluate
$\tilde{S}$	Set of $\tilde{N}$ random candidate indices
m_i	Standard deterministic resampling MIS weights
$\tilde{m}_i$	Our stochastic resampling MIS weights

ReSTIR [BWP*20, WKL*23, OLK*21] is a recent class of Monte Carlo light transport simulation estimators that combines the idea of path reusing with resampled importance sampling (RIS) [TCE05]. The idea is to store one path in each pixel, and progressively improve its distribution by borrowing samples from spatiotemporal neighbor pixels by RIS, which returns one of the candidates with probabilities chosen to yield an improved distribution. The original ReSTIR only applied to direct light sampling. For general path tracing, it was necessary to generalize resampled importance sampling to handle samples coming from different domains [LKB*22]. Significant progress has been made to further generalize ReSTIR to handle final gathering [KLR*23, KBG24], bidirectional sampling [HKL*25], distributed effects [ZLK*24], reducing correlation artifacts [SLK*22], subsurface scattering [WSD24], and differentiable rendering [CSN*23, WWWZ23].

Our work is grounded on generalized resampled importance sampling, and we significantly improve the reuse efficiency over a large set of neighbors by unbiasedly estimating the multiple importance sampling weight. Our derivation shares some similarities to the recent continuous and marginal multiple importance sampling work [WGGH20, WGH22], and our idea of reusing over a large neighbor shares similarities to the recent image-space splatting work [TH24] which only allows splatting to the color buffer. We apply the idea of stochastic weight estimation in the context of resampling, hence it allows us to design more efficient unbiased estimators using weights that are specialized to resampling and run in real-time.

2. Background

We briefly review the generalized resampled importance sampling [LKB*22] in this section, and we refer the readers to standard textbooks [PJH16] and the ReSTIR tutorial [WKL*23] for a more comprehensive introduction to light transport simulation. Table 1 summarizes our notation.

2.1. Monte Carlo Integration

Monte Carlo integration estimates an integral $\int_{\mathcal{D}} f(x)\,\mathrm{d}x$ by evaluating $f(X)/p(X)$ for a random variable X with probability density

$p(X)$, where the support of X is assumed to cover f. The expected value yields the desired integral:

$$\mathbb{E}\left[\frac{f(X)}{p(X)}\right] = \int_{\text{supp } X} f(x)\,\mathrm{d}x. \tag{1}$$

A key component in Monte Carlo integration is importance sampling, i.e., drawing samples X according to some desired probability distribution p, where p should be close to the integrand f in its shape to decrease sampling noise.

2.2. Resampled Importance Sampling

Resampled Importance Sampling (RIS) [Tal05] is a resampling-based importance sampling method that, given candidate samples with known distributions, picks one proportionally to carefully chosen resampling weights. This outputs samples distributed *approximately* according to an unnormalized target distribution $\hat{p}$.

Given M candidates $X_i \sim p_i$, where p_i are easy to sample from but not necessarily good matches for the integrand f, a single sample $Y = X_z$ is selected, where $z \sim w_z^{\text{RIS}}$ is sampled according to resampling weights

$$w_i^{\text{RIS}} = \frac{\hat{p}(X_i)}{p_i(X_i)}. \tag{2}$$

As the candidate count M grows, the distribution of the output sample tends towards the normalized target function $\|\hat{p}\|$.

RIS can be efficiently implemented in a streaming manner by weighted reservoir sampling (WRS) [Cha82].

2.3. Generalized RIS

Generalized RIS (GRIS) [LKB*22] extends RIS to allow input samples from arbitrary domains (e.g., other pixels), using shift mappings to move samples between domains. Resampling MIS weights account for overlap between domains.

2.3.1. Unbiased Contribution Weights

Resampling with RIS does not produce closed-form PDFs, so Equation 1 cannot be used for integration. In GRIS, the reciprocal probability density of X is replaced with an Unbiased Contribution Weight (UCW) W_X. By definition, a UCW W_X implements

$$\mathbb{E}\Big[f(X)W_X\Big] = \int_{\text{supp } X} f(x)\,\mathrm{d}x \tag{3}$$

for any integrable function f. Equivalently, W_X is an unbiased estimate for X's reciprocal PDF,

$$\mathbb{E}\Big[W_X \mid X = x\Big] = \frac{1}{p_X(x)}. \tag{4}$$

When producing X with a PDF p, $W_X = 1/p(X)$ is a valid UCW.

2.3.2. Shift mappings

Shift mappings $T_i : \Omega_i \to \Omega$ bijectively map samples from their source domains i to the target domain. Bijectivity and Jacobian determinants $\left|\frac{\partial T_i}{\partial x}\right|$ are required as shift mappings are used as a change of variables between pixel integrals. Shift mappings can fail; T_i need not be defined for all $x \in \Omega_i$, so T_i are technically *partial bijections*.

2.3.3. Canonical samples

An input i is “canonical” if its shift map T_i to the target domain is identity and its sample X_i alone covers the target function $\hat{p}$. It is common to have exactly one canonical input per reuse, with index often denoted as c.

2.3.4. Resampling with GRIS

GRIS assumes M sample-UCW pairs (X_i, W_{X_i}) where samples are from arbitrary domains $X_i \in \Omega_i$. Each sample is first shifted into the target domain as

$$Y_i = T_i(X_i). \tag{5}$$

A single index z is then randomly selected according to resampling weights

$$w_i = [X_i \in \mathcal{D}(T_i)]\, m_i(Y_i) \cdot \hat{p}(Y_i) \cdot W_{X_i} \cdot \left|\frac{\partial T_i}{\partial X_i}\right|, \tag{6}$$

with the Iverson bracket evaluating to 1 if $T_i(X_i)$ is defined and 0 otherwise, and m_i is a *resampling MIS weight*. Index z is randomly selected with probability $P(z) = w_z / \sum_{j=1}^{M} w_j$. The resampling result is $Y = Y_z$, which is given a UCW

$$W_Y = \frac{1}{\hat{p}(Y)} \sum_{j=1}^{M} w_j. \tag{7}$$

The sample-UCW pair (Y, W_Y) can be stored for further reuse, or directly used to estimate the integral of f as $f(Y)W_Y$ following Equation 3.

2.4. Resampling MIS

After shifting, distributions of multiple samples can overlap in the target domain, resulting in overcounting or undercounting of the shifted samples. To account for this, samples are weighted by *resampling MIS (Multiple Importance Sampling) weights m_i*, which form a partition of unity

$$\sum_{i=1}^{M} m_i(y) = 1, \tag{8}$$

for all $y \in \text{supp } \hat{p}$, where $m_i(y) = 0$ if y could not have been produced from domain i via T_i.

2.4.1. Resampling MIS Families

While any choice of m_i satisfying Equation 8 works, in practice special care must be taken to control variance. A straightforward choice for m_i is the generalized balance heuristic [LKB*22], which extends the balance heuristic used in Talbot's RIS [Tal05] to multi-domain reuse using Jacobian-corrected target functions $\hat{p}$ as proxy PDFs, and confidence weights c_i as

$$m_i^{\text{bal}}(y) = \frac{c_i \hat{p}_{\leftarrow i}(y)}{\sum_{j=1}^{M} c_j \hat{p}_{\leftarrow j}(y)}, \tag{9}$$

where the symbol $\hat{p}_{\leftarrow i}$, referred to as “$\hat{p}$ from i”, is the Jacobian-corrected target distribution from domain i, defined as

$$\hat{p}_{\leftarrow i}(y) = [y \in T_i(\text{supp } X_i)]\, \hat{p}_i\left(T_i^{-1}(y)\right) \left|\frac{\partial T_i^{-1}}{\partial y}\right|. \tag{10}$$

2.5. Confidence weights

The $\hat{p}_{\leftarrow i}$ quantities are weighted by *confidence weights* c_i. Confidence weights only affect variance, not bias, and typically estimate how many path tracing samples would be needed for the same variance, i.e., the *effective sample count*.

Measuring effective sample count in real-time is hard, so confidence weights c_i are usually set to the total number of initial samples affecting X_i via GRIS, i.e., the sum of confidence weights over the inputs used for producing X_i. This grossly overestimates the effective sample count, both because it double-counts repeatedly reused samples, and because reuse between pixels typically causes some quality loss due to differences between the pixels' path spaces. Hence, c_i is normally capped to a user parameter called *confidence cap*. Without confidence capping, ReSTIR tends to diverge due to correlation artifacts. We refer to [LKB*22, Section 9] for further discussion on confidence capping.

2.6. Reservoir Definition

A *reservoir* refers to the tuple (X_i, W_{X_i}, c_i) defining how the sample X_i is processed in GRIS. A practical implementation might also cache expensive computation like $\hat{p}(X_i)$ or quantities speeding up shift mapping.

2.7. Pairwise MIS

The generalized balance heuristic (Equation 9) from M candidates requires $O(M^2)$ evaluations of $\hat{p}_{\leftarrow i}$, which quickly becomes intractable as M increases. A faster alternative is the generalized pairwise MIS [LKB*22] which only requires $O(M)$ evaluations while offering similar convergence behavior [Bit21].

The generalized pairwise MIS pairs each non-canonical domain $i \neq c$ with the canonical domain c and evaluates the balance heuristic between these pairs. The defensive variant (Equation 7.8 in [WKL*23]) is:

$$m_i^{\text{pw}}(y) = \left(\frac{c_\Sigma}{c_\Sigma + c_c}\right) \frac{c_i \hat{p}_{\leftarrow i}(y)}{c_\Sigma \hat{p}_{\leftarrow i}(y) + c_c \hat{p}(y)} \qquad (i \neq c) \tag{11a}$$

$$m_c^{\text{pw}}(y) = \frac{c_c}{c_\Sigma + c_c} + \sum_{j \neq c}^{M} \left(\frac{c_j}{c_\Sigma + c_c}\right) \frac{c_c \hat{p}(y)}{c_\Sigma \hat{p}_{\leftarrow j}(y) + c_c \hat{p}(y)}, \tag{11b}$$

where c_Σ is the sum of non-canonical confidence weights,

$$c_\Sigma = \sum_{k \neq c}^{M} c_k. \tag{12}$$

For further explanation of pairwise MIS, we refer to [WKL*23, Section 7.1.3].

2.8. ReSTIR

The above techniques form the core of ReSTIR [LKB*22]. In *initial sampling*, each pixel samples one or more *initial candidates*, often with standard path tracing. The initial candidates are *merged* into the *initial sample* with RIS. In *temporal reuse*, the initial sample of each pixel is merged with the one from the corresponding prior-frame pixel with GRIS. *Spatial reuse* merges each pixel's temporal resampling result with samples from other, randomly picked pixels, using GRIS. Finally, the result of spatial resampling is stored for the next frame and evaluated as $f(X)W_X$ to estimate the pixel integral.

3. Stochastic Resampling MIS Weights

Standard GRIS reuses from all M candidate domains, which is inefficient when many reservoirs have no contributing samples. We choose only candidates with non-zero contribution in their source pixel. This requires care, as naively choosing input pixels based on their samples leads to bias. Instead, we mathematically reuse from a large number M of pixels, but optimize the reuse to $\tilde{N} < M$ by culling non-contributing samples and reusing stochastically from the rest. This guarantees proper MIS weights and unbiased reuse.

We implement the stochastic reuse by replacing the resampling MIS weights m_i in Equation 6 with *stochastic resampling MIS weights* $\tilde{m}_i$, which unbiasedly estimate their deterministic counterparts. We prove unbiasedness in Appendix A. In this section, we first propose a general stochastic estimator that is unbiased, and then in Section 4 we specialize to pairwise MIS (Equation 11) for efficiency. Finally, in Section 5, we discuss our efficient implementation inside a GPU-based renderer.

Given the GRIS input samples $\bar{X} = (X_1, \ldots, X_M)$ and UCWs $\bar{W} = (W_{X_1}, \ldots, W_{X_M})$, our stochastic $\tilde{m}_i$ estimate the standard non-stochastic m_i using new random numbers Z as

$$\mathbb{E}\left[\tilde{m}_i(Y_i;Z) \mid \bar{X}, \bar{W}\right] = m_i(Y_i), \tag{13}$$

where $Y_i = T_i(X_i)$ are the shifted input samples, following Equation 5. We then replace the deterministic m_i in Equation 6 with our stochastic $\tilde{m}_i$, yielding stochastic resampling weights

$$\tilde{w}_i = [X_i \in \mathcal{D}(T_i)]\, \tilde{m}_i(T_i(X_i);Z) \cdot \hat{p}(T_i(X_i)) \cdot W_{X_i} \cdot \left|\frac{\partial T_i}{\partial X_i}\right|, \tag{14}$$

which simply replace the non-stochastic w_i in Equation 7.

3.1. General Stochastic Resampling MIS Estimator

Our general form of the stochastic MIS weight $\tilde{m}_i$ reuses $\tilde{N}$ out of the M candidates by replacing the full sum of MIS weights $\sum m_i$ with an $\tilde{N}$-sample importance sampling estimator with selection probabilities $P(i)$.

Conditioned on the samples $\bar{X} = X_1, \ldots, X_M$ and their contribution weights $\bar{W}$, we assume for each input pixel $i \in \{1, \ldots, M\}$ a selection probability $P(i)$ for reuse. We sample a multiset of $\tilde{N}$ pixels *with replacement*. The outcome is a set $\bar{S} = \left\{S_1, \ldots, S_{|S|}\right\}$ of unique pixels ($|S| \leq \tilde{N}$), with multiplicities $\mathbf{K}_{\bar{S}}(i)$ tracking how often each pixel i was chosen. Our stochastic resampling MIS weight for $Y_i = T_i(X_i)$ is

$$\tilde{m}_i(Y_i;Z) = \frac{\mathbf{K}_{\bar{S}}(i)}{\tilde{N}P(i)} m_i(Y_i), \tag{15}$$

where m_i is any non-stochastic resampling MIS heuristic (e.g., Equation 9 or 11). We allow P to depend on the input samples and UCWs, as the expectation in Equation 13 is conditioned on them.

If $P(i)=0$, also $\mathbf{K}_{\bar{S}}(i)=0$ and we define $\tilde{m}_i(Y_i;Z)=0$. For the result to remain unbiased, we require $P(i)$ to be positive wherever $m_i(Y_i)$ is positive. We prove Equation 15 satisfies Equation 13 in Appendix B.

The multiplicity in $\mathbf{K}_{\bar{S}}(i)$ in Equation 15 ensures that instead of M, at most $\tilde{N}$ candidates need to be processed. By carefully choosing selection probabilities P, we achieve a high-quality result at a fraction of the cost. We specify the MIS heuristic and the selection probability in the next section.

4. Stochastic Pairwise MIS

While Equation 15 can be used to estimate arbitrary resampling MIS weights, the choice of the MIS heuristic m_i greatly affects performance. Using a stochastic version of the balance heuristic (Equation 9) decreases the number of shifts required for each image pixel from $O(M^2)$ to $O(\tilde{N}\cdot M)$, which is still prohibitive for large M. Instead, we choose to focus on the pairwise MIS variants (Equation 11) as we can reduce their cost from $O(M)$ to $O(\tilde{N})$ shifts, with quality often comparable to the balance heuristic. With $\tilde{N}\ll M$, this allows cheap reuse from a large number of pixels.

Pairwise MIS weights have different formulas for canonical and non-canonical cases, and require different estimators. We estimate non-canonical MIS weights Equation 11a with our stochastic resampling MIS estimator Equation 15. We separately estimate the canonical weight Equation 11b by replacing the sum in m_c^{pw} with a standard $\tilde{N}_c$-sample estimator. The total number of paths to be shifted per-pixel is then $\tilde{N}+\tilde{N}_c$.

4.1. Stochastic Non-Canonical Pairwise MIS

We use Equation 15 directly with Equation 11a for non-canonical candidates $i\neq c$, i.e.,

$$\tilde{m}_i^{\text{pw}}=\frac{\mathbf{K}_{\bar{S}}(i)}{\tilde{N}P(i)}\left(\frac{c_\Sigma}{c_\Sigma+c_c}\right)\frac{c_i\hat{p}_{\leftarrow i}(y)}{c_\Sigma\hat{p}_{\leftarrow i}(y)+c_c\hat{p}(y)}. \tag{16}$$

The performance of this estimator largely depends on the choice of candidate selection probabilities P. Ideally, candidates should be selected according to Equation 6, which involves a shift mapping to the target domain. We aim to evaluate as few shift mappings as possible to maximize performance, and draw candidates according to the quantity

$$P(i)\propto c_i\hat{p}(X_i)W_{X_i}, \tag{17}$$

which is simply the luminance of pixel i scaled by its confidence weight c_i, since we set $\hat{p}(X_i)$ to the luminance of the integrand f. This quantity is readily available as input without extra computations. This works well when shift mappings are effective, i.e., the non-shifted $\hat{p}(X_i)$ is highly correlated with the shifted $\hat{p}(T_i(X_i))$. Figure 7 shows the effect of this approximation with varying amounts of evaluated candidates ($\tilde{N}$) to the standard deterministic reuse over all input candidates (Figure 7d).

A key benefit of choosing reuse pixels proportionally to $\hat{p}(X_i)W_{X_i}$ is that valid candidate samples (i.e., $\hat{p}(X_i)>0$) are always found when they exist within the large kernel of M pixels. This makes our reuse method much more likely to succeed with poor initial candidates than prior methods that must ignore the samples when selecting candidates.

4.2. Stochastic Canonical Pairwise MIS

To build a stochastic pairwise MIS estimator for the canonical case (Equation 11b), we sample a different set of $\tilde{N}_c$ pixels with probabilities $P_c(i)$, again *with replacement*. This produces another set of unique pixels $\bar{S}'=\left\{\bar{S}'_1,\ldots,\bar{S}'_{|\bar{S}'|}\right\}$ with multiplicities $\mathbf{K}_{\bar{S}'}(i)$. We then set

$$\tilde{m}_c^{\text{pw}}(Y_c)=\frac{c_c}{c_\Sigma+c_c}+\sum_{i\in\bar{S}'}\frac{\mathbf{K}_{\bar{S}'}(i)}{\tilde{N}_cP_c(i)}\beta_i(Y_c), \tag{18}$$

where

$$\beta_i(Y_c)=\left(\frac{c_i}{c_\Sigma+c_c}\right)\frac{c_c\hat{p}(Y_c)}{c_\Sigma\hat{p}_{\leftarrow i}(Y_c)+c_c\hat{p}(Y_c)}. \tag{19}$$

The result if unbiased if $P_c(i)>0$ whenever $c_i>0$. It would be natural to use $P_c(i)\propto c_i$, but we find $P_c(i)=\frac{1}{M}$ adequate. Figure 6 shows that lowering noise of $\tilde{m}_c^{\text{pw}}$ has little effect.

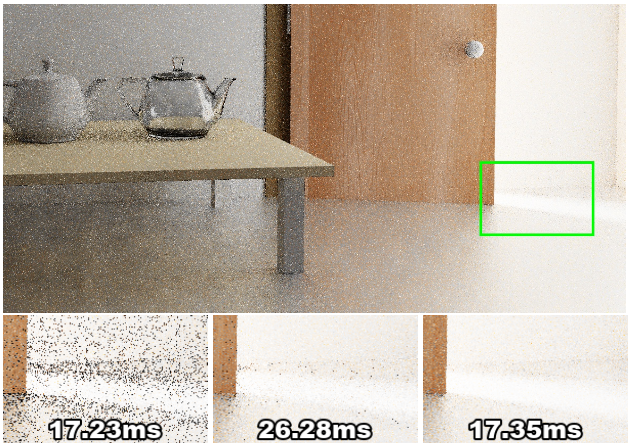
4.3. Non-Canonical Confidence Adjustment

The fractions in Equation 11 use the sum of non-canonical confidence weights c_Σ to weight the non-canonical $\hat{p}$ values in the denominator. Thus, when M is large, so is c_Σ, and the weight of the canonical sample becomes low, even if the $\tilde{N}$ selected samples turn out to be poor. This causes variance from stochastic MIS to manifest as dark pepper noise, where the canonical sample is given too low a weight (Figure 2a).

To help explain this, we analyze a hypothetical situation where $\tilde{N}=1$ and a neighbor containing a zero-contribution sample (after being shifted to the target domain) is unluckily chosen by our stochastic MIS estimator. If we further assume that a good canonical sample exists and is successfully shifted to all neighbors with a Jacobian approximately equal to 1, then pairwise MIS gives a weight of approximately $c_c/(c_c+c_\Sigma)$ to the canonical sample, and a weight of approximately $c_\Sigma/(c_c+c_\Sigma)$ to the zero-contribution neighbor sample. Thus, when c_Σ is large, this scenario results in a near-zero contribution to the canonical sample, even if it is the only "good" sample present.

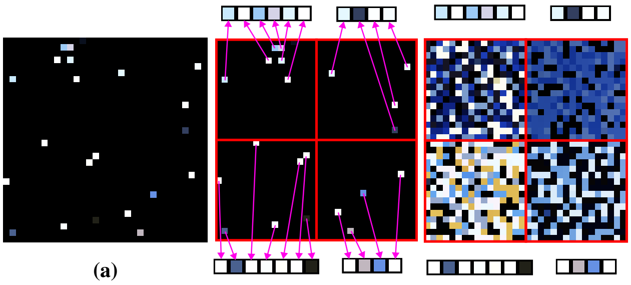
To alleviate this, we apply a defensive strategy of scaling all non-canonical confidence weights, and therefore c_Σ, by the factor $\tilde{N}/M$. This effectively lowers the weight of non-canonical samples, limiting the variance introduced by stochastic MIS. Our strategy ensures that $c_\Sigma\propto\tilde{N}$, while still naturally degrading to deterministic pairwise MIS as $\tilde{N}$ approaches M.

Figure 2 empirically studies this problem in penumbra, a phenomenon caused by a visibility discontinuity. Shift mapping through this visibility discontinuity fails, causing the pepper noise. Using larger $\tilde{N}$ alleviates the problem, but at a significant cost.



(a) $\tilde{N} = 1$ **(b)** $\tilde{N} = 8$ **(c)** $\tilde{N} = 1$, *scaled* c_i

Figure 2: *When $\tilde{N}$ is very low, stochastic sampling can drastically increase variance (2a). Increasing $\tilde{N}$ helps, but increases cost (2b). We instead scale non-canonical confidences c_i by the ratio $\tilde{N}/M$, which lowers them to account for the fact that only $\tilde{N}$ neighbors are actually evaluated (2c).*



(a) **(b)** **(c)**

Figure 3: *Our reservoirs store one sample per pixel (3a). We divide the image into small screen-space* reuse cells *and find samples within each cell (3b). Finally, pixels choose their reused samples from the cells based on their contribution to the source pixel (3c).*

5. Implementation using Screen-Space Cells

Our method allows increasing the number of reuse candidates M by orders of magnitude, but the input pixels must still be carefully chosen. We could simply reuse from all pixels in the image, but reusing from far away pixels generally increases variance as shift mappings become less effective and ideal path distributions diverge. Instead, we classify pixels into screen-space *reuse cells* and perform stochastic pairwise reuse within each cell (Figure 3). We assign per-pixel cell IDs by first dividing the screen into small tiles, then hashing each pixel's tile ID and auxiliary information.

We refer to *tiles* as individual 8×8 regions of pixels, and *cells* as regions within each tile containing similar auxiliary information, such that a tile can contain multiple cells. To construct cells containing similar pixels, we hash each pixel's tile ID, object ID, and quantized surface normal using the PCG hash function [O'N14].

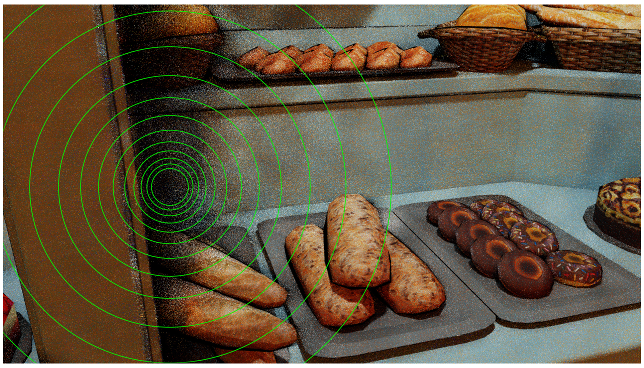


Figure 4: *We use WRS to randomly select a high-confidence cell within a radius around each pixel. To help find cells outside of large disocclusions, we grow the radius after every WRS iteration by 25%.*

5.1. Searching Relevant Neighbor Cells

Only reusing from the closest screen-space cell introduces obvious correlation artifacts and often does not help with large disocclusions, as they may reside completely within the disocclusion. Closest cells outside of the disocclusion could be far. Therefore, we search around the pixel with increasing radii to find a sufficiently similar cell containing pixels with high confidence weights.

We select a neighbor cell according to the sum of its pixels' confidence weights using a single weighted reservoir sampling (WRS) [Cha82] pass: For each iteration, we randomly sample a pixel within the current radius and fetch its associated cell. The cell is rejected if it has dissimilar G-buffer normal or object ID, and is otherwise treated as a candidate sample, weighted by its pixel confidence sum. To help find cells outside of disocclusions, we increase the search radius after every iteration, as shown in Figure 4. Our implementation uses 12 iterations to find a neighbor cell, with an initial 30px radius which is increased by 25% after each iteration.

5.2. GPU Implementation

In this subsection we provide a detailed explanation of our efficient GPU implementation. We begin with cell construction, which enables the subsequent spatial resampling.

5.2.1. Cell Construction

Algorithm 1 sorts pixels into cells on the GPU using two parallel multimaps. Each pixel determines its hashed cell ID, then inserts its pixel index into the "cellPixelIndices" multimap. If the pixel contains a non-zero sample, it also inserts its index into the "cellReservoirs" multimap.

The multimaps are efficiently sorted on the GPU so that each cell's data is adjacent in memory, following the hash grid implementation described in [Boi21]. This forms per-cell lists which can be easily sampled. In Algorithm 2, the RandomPixelInCell function chooses a pixel index from the cell's list uniformly, while the ResampleReservoirInCell function chooses a pixel from the cell according to Equation 17 using WRS.

Algorithm 1: CreateReuseCells()

```
Input: Samples $X_i$ for $i \in [1, N_{pixels}]$
Output: cellPixelIndices
Output: cellReservoirs
Output: cellConfidenceSums
// Initialize outputs
cellPixelIndices ← GPUMultiMap()
cellReservoirs ← GPUMultiMap()
cellConfidenceSums ← zeros($N_{pixels}$)
// Construct cells in parallel
parallel for $i \leftarrow 1$ to $N_{pixels}$
    // Get cell ID of pixel $i$ by hashing G-buffer information
    cell ← GetCellID($i$)
    // Atomically count cell confidence sums
    cellConfidenceSums[cell] += $c_i$
    // Insert pixel index into per-cell lists
    cellPixelIndices.Insert(cell, $i$)
    if $\hat{p}(X_i) > 0$ then
        // Store just the reservoir index
        cellReservoirs.Insert(cell, $i$)
```

Algorithm 2: StochasticPairwiseReuse()

```
Input: Samples $X_i$ for $i \in \{1, \ldots, N_{pixels}\}$
Input: cellPixelIndices: Maps cell ID to list of pixels
Input: cellReservoirs: Maps cell ID to list of pixels with non-zero samples
Input: cellConfidenceSums: Maps cell ID to $c_\Sigma$ (Equation 12)
Output: Per-pixel sample-UCW pairs $(Y, W_Y)$
1  parallel for $c \leftarrow 1$ to $N_{pixels}$
2      cell ← SampleNeighborCell(c)
3      $c_\Sigma$ ← cellConfidenceSums[cell]
4      $Y \leftarrow \emptyset$ // Initialize output sample
5      $w_{\mathrm{sum}} \leftarrow 0$
       // Estimate canonical MIS weight (Equation 18)
6      if $\hat{p}(X_c) > 0$ then
           // Estimate canonical MIS weight
7          $\tilde{m}_c \leftarrow \frac{c_c}{c_\Sigma + c_c}$ // Equation 18
8          for $j \leftarrow 1$ to $\tilde{N}_c$ do
               // Sample a pixel $z' \sim P'_s(z')$ from cellPixelIndices
9              $z' \leftarrow$ RandomPixelInCell(cell)
10             $\tilde{m}_c \leftarrow \tilde{m}_c + \frac{\beta(z')}{P'_s(z')} \cdot \frac{1}{\tilde{N}_c}$ // Equation 18
11         $\tilde{w}_c \leftarrow \tilde{m}_c \cdot \hat{p}(X_c) \cdot W_{X_c}$ // Equation 14
           // Merge $Y$ with the canonical sample $X_c$
12         $Y, w_{\mathrm{sum}} \leftarrow$ ReservoirMerge($Y, w_{\mathrm{sum}}, X_c, \tilde{w}_c$)
       // Merge $Y$ with samples from cell
13     for $j \leftarrow 1$ to $\tilde{N}$ do
           // Sample a pixel $z \sim P_s(z)$ from cellReservoirs
14         $z \leftarrow$ ResampleReservoirInCell(cell)
15         $y \leftarrow T_z(X_z)$ // Shift sample $X_z$ to output pixel $c$
16         $\tilde{m}_z \leftarrow \frac{1}{\tilde{N}} \frac{1}{P_s(z)} m_z^{\mathrm{pw}}(y)$ // Equation 16
17         $\tilde{w}_z \leftarrow \tilde{m}_z \cdot \hat{p}(y) \cdot W_{X_z} \cdot \left|\frac{\partial T_z}{\partial X_z}\right|$ // Equation 14
           // Merge $Y$ with the candidate $y$
18         $Y, w_{\mathrm{sum}} \leftarrow$ ReservoirMerge($Y, w_{\mathrm{sum}}, y, \tilde{w}_z$)
       // Not shown: set $W_Y = 0$ if $\hat{p}(Y) = 0$
19     $W_Y \leftarrow w_{\mathrm{sum}} / \hat{p}(Y)$ // Equation 7.
20     outputReservoirs[c] = $(Y, W_Y)$
```

Algorithm 3: ReservoirMerge($Y, w_{\mathrm{sum}}, y_i, w_i$)

```
Input: Reservoir $Y, w_{\mathrm{sum}}$
Input: Sample $y_i$ with weight $w_i$
Output: Updated $Y, w_{\mathrm{sum}}$
$w_{\mathrm{sum}} \leftarrow w_{\mathrm{sum}} + w_i$
if rand() $< w_i / w_{\mathrm{sum}}$ then
    $Y \leftarrow y_i$
return $Y, w_{\mathrm{sum}}$
```

5.2.2. Spatial Resampling

Our spatial reuse algorithm is shown in Algorithm 2. This starts by sampling a neighbor cell (Section 5.1), which provides a dense set of M neighboring samples. We first estimate the MIS weight for the canonical sample by shifting it to $\tilde{N}_c$ pixels uniformly selected from the reuse cell (lines 7-12). Then, we choose $\tilde{N}$ random samples from the cell according to selection probabilities $P \propto c_i \hat{p}(X_i) W_{X_i}$ (Equation 17), and finally select one according to $\tilde{w}_i$ (Equation 14) using our stochastic resampling MIS weights (lines 13-18). Finally, we compute the UCW for the selected sample by dividing the sum of resampling weights w_{sum} by the $\hat{p}(Y)$ from the selected sample (line 19).

We implicitly scale all non-canonical confidence weights c_i in Algorithm 2, including c_Σ, by the factor $\tilde{N}/M$, following Section 4.3. The multiplicities in Equation 18 and Equation 15 are implicitly considered by allowing the loops on lines 8 and 13 to sample the same neighbor (z' and z) multiple times.

As done by prior ReSTIR methods, we use reservoir merging (Algorithm 3) to efficiently select a single output sample proportionally to resampling weights in a single pass.

6. Results

We implemented our method on the GPU using the Slang shading language within the Falcor rendering framework [KCK*22]. All results use an RTX 4090 GPU and a Ryzen 7 3700x CPU. We render all images at 1920×1080. We compare numerical errors using mean absolute percentage error (MAPE). We use MAPE $=$ mean$(|I - I_{\mathrm{ref}}| / (I_{\mathrm{ref}} + 0.01 \cdot \mathrm{mean}(I_{\mathrm{ref}})))$.

We compare our method against path tracing and standard ReSTIR. We sample one initial candidate per pixel by path tracing with next event estimation, with a maximum of 8 bounces. Rather

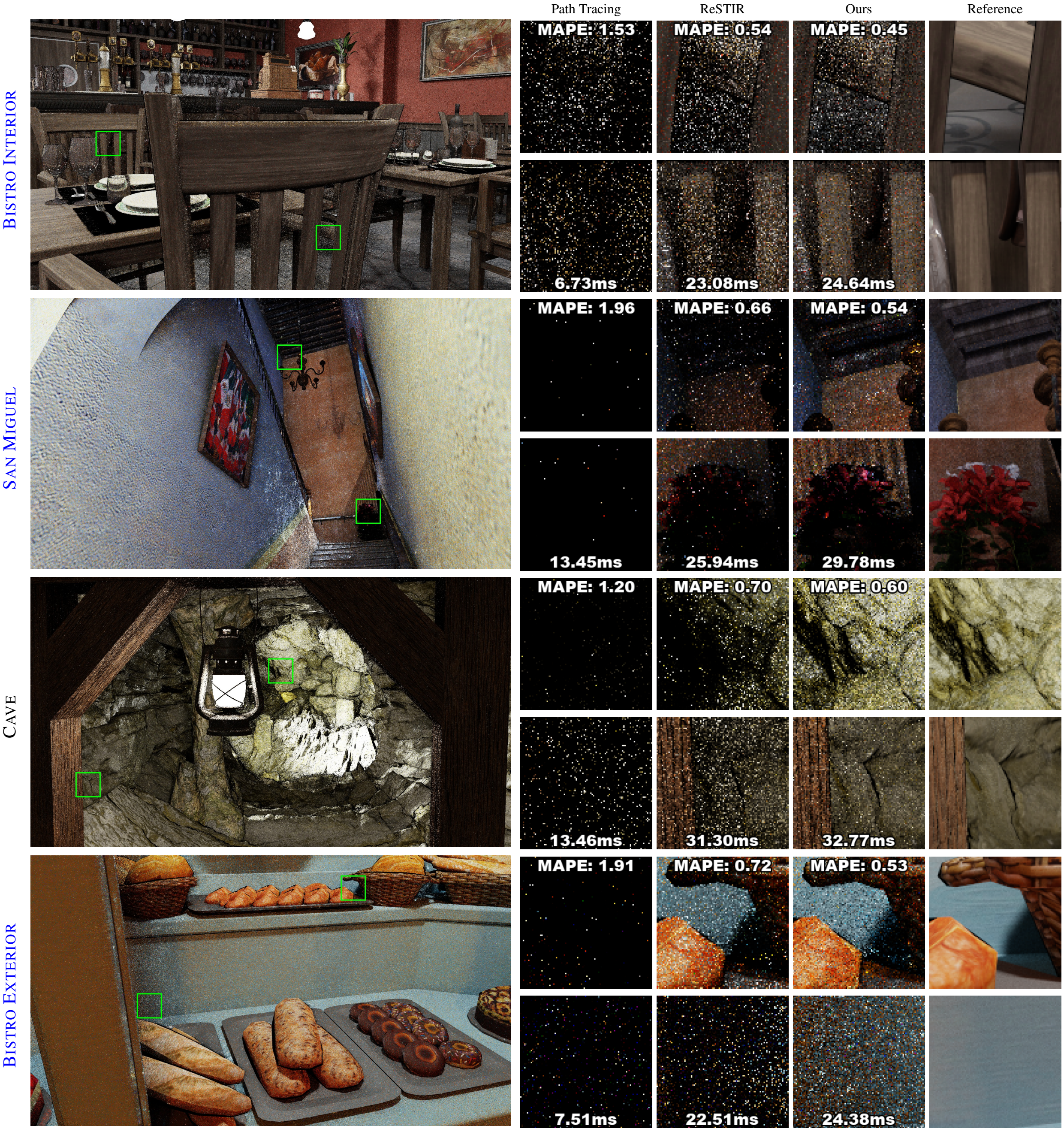


Figure 5: *Results on various scenes. All scenes feature disocclusions caused by camera motion, scene animation, or both. Our method significantly improves quality, especially in disoccluded regions where there is little useful temporal information. "ReSTIR" is standard ReSTIR PT [LKB*22] with $M = 3$ spatial neighbors using defensive pairwise MIS. Our method uses $\tilde{N} = 3$ candidates from a single neighbor cell which contains at most $M = 64$ neighbors. We pick the cell proportionally to confidence weights using 12-sample WRS as described in Section 5.1. Results were measured on the last frame of a 50-frame animation sequence, with confidence weights capped to 20.*

	Initial Sampling	8.68ms ± 2.12
	Temporal Reuse	4.06ms ± 1.36
Ours	Create Reuse Cells	1.57ms ± 0.13
	Compute $\tilde{m}_c^{\mathrm{pw}}$	4.40ms ± 0.36
	Stochastic Neighbor Reuse	9.19ms ± 0.98

Table 2: *Timings of individual components of our method, averaged over 10 runs on the same four scenes in Figure 5. "Create reuse cells" refers to Algorithm 1, while "Compute $\tilde{m}_c^{\mathrm{pw}}$" and "Stochastic neighbor reuse" are performed by Algorithm 2, lines 6-12, and lines 13-20, respectively. Initial sampling and temporal reuse always run before our method.*

than separating direct and indirect illumination into different reservoirs, we use a single reservoir for both.

To minimize implementation differences, we use the same code and parameters for our method and standard ReSTIR, with only the spatial reuse method differing. For standard ReSTIR, we select $M = 3$ candidates from a 30px radius, following [LKB*22]. For our method, we classify pixels into cells according to their G-buffer normal, object ID, and tile ID using 8×8 tiles. We then select a random cell (Section 5.1) without looking at the path samples, giving $M \leq 64$. We then select $\tilde{N} = 3$ candidates based on their contributions and confidence weights from the chosen cell as part of stochastic pairwise MIS evaluation (Section 5.2.2). We estimate the canonical MIS weight with $\tilde{N}_c = 1$.

6.1. General Comparison

Figure 5 analyzes our results on a multitude of scenes. We find our method to significantly improve the image quality both visually and numerically, both overall and especially in disocclusions.

Table 2 shows the timings for the stages in our method, as well as those for standard spatial and temporal reuse passes.

6.2. Improved Disocclusions

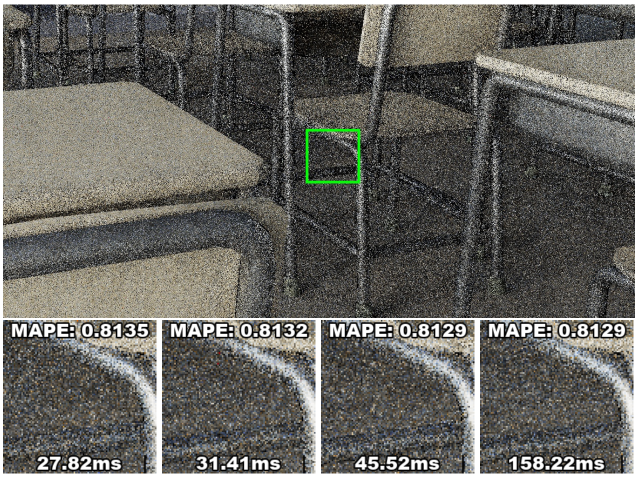
Figure 1 studies the effect of our novel spatial reuse method on disocclusions. Our stochastic pairwise MIS effectively searches for contributing pixels in the chosen 8×8 cell. This greatly improves the efficiency of spatial reuse and reduces noise in the presence of disocclusions.

6.3. Adjusted Non-Canonical Confidences

In Figure 2 we study the effect of adjusting the non-canonical confidence weights by multiplier $\tilde{N}/M$ at MIS weight evaluation. This re-balances the contributions of the spatial neighbors and the current pixel when using our stochastic pairwise MIS, removing the distracting pepper noise otherwise present in many scenes.

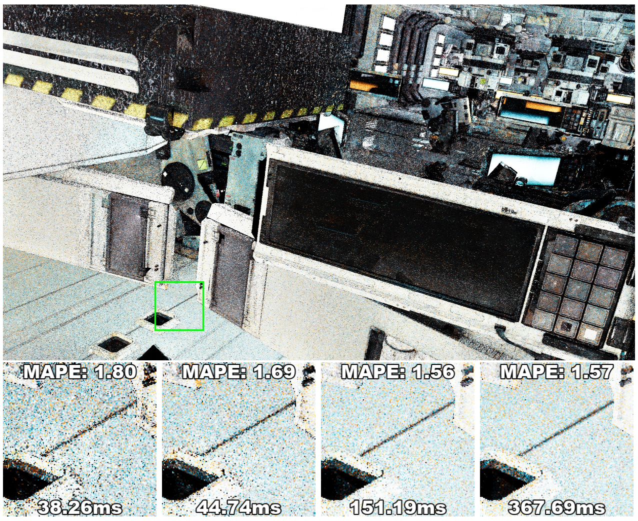
6.4. Canonical MIS Weights: Use $\tilde{N}_c = 1$

Standard pairwise MIS evaluates the canonical sample's MIS weight Equation 11b by shifting it to all input domains, which is prohibitively costly with large M. Our stochastic pairwise MIS uses $\tilde{N}_c < M$ samples, which theoretically increases noise. In practice, we observe that variance is dominated by the non-canonical sample contributions, while $\tilde{m}_c^{\mathrm{pw}}$ has little effect on the overall noise (Figure 6). Hence, we always use $\tilde{N}_c = 1$ which maintains unbiasedness while minimizing the performance cost.



(a) $\tilde{N}_c = 1$ **(b)** $\tilde{N}_c = 2$ **(c)** $\tilde{N}_c = 8$ **(d)** $\tilde{N}_c = M$

Figure 6: *We estimate the sum in the canonical MIS weight m_c^{pw} (Equation 11b) with $\tilde{N}_c$ samples. Increasing $\tilde{N}_c$ only slightly effects the error, quickly yielding diminishing returns. For the $\tilde{N}_c = M$ case (Figure 6d), we simply evaluate the full m_c^{pw} without estimation.*



(a) $\tilde{N} = 1$ **(b)** $\tilde{N} = 2$ **(c)** $\tilde{N} = 16$ **(d)** $\tilde{N} = M$

Figure 7: *Our method optimizes full M-candidate reuse by stochastically reusing only $\tilde{N}$ of the M candidates, offering a significant cost reduction at some loss of quality. The case where $\tilde{N} = M$ is equivalent to standard deterministic reuse over the whole tile.*

6.5. Effect of $\tilde{N}$ on Stochastic MIS

Standard pairwise MIS shifts all M candidate samples to the target domain for evaluation, which is expensive. We optimize by using our stochastic variant (Equation 15) and only have to shift $\tilde{N}$ well-chosen samples. Figure 7 compares our method with different values of $\tilde{N}$ and full evaluation of pairwise MIS, showing that a relatively low number of $\tilde{N}$ may be a good choice. Our results use $\tilde{N} = 3$, but lower numbers can be used for faster rendering.

6.6. Supplemental Results

Our supplemental material includes the full images of our results, as well as a video comparing our method to standard ReSTIR in ZERO DAY. Our method generally shows improved image quality over standard ReSTIR, especially at disocclusions, which are common in animated scenes.

7. Discussion and Future work

Our spatial reuse method optimizes the reuse of M candidate samples by stochastically evaluating only $\tilde{N}$. The contributions of the other $M - \tilde{N}$ reuses are conveyed by the compensation factor $1/(\tilde{N}P(i))$ in the stochastic MIS weights Equation 15. We get best results when this stochastic estimate has the lowest error—when the M pixels in the cell are chosen to be as similar as possible. This is a key observation behind our method, and justifies our cell-based implementation.

While our insights improve image quality over standard ReSTIR, our research also opens multiple promising directions.

Our stochastic resampling MIS weights increase the freedom in input selection by allowing us to look at the samples without introducing bias. Variants of this idea could be applied to other problems where MIS is used to combine a large set of techniques.

Our implementation selects one cell for reuse, and picks $\tilde{N} = 3$ samples from it. Future work could explore stratified reuse, for example by selecting three cells and drawing one sample from each.

Our importance weights for selecting the $\tilde{N}$ inputs from the M candidates (Equation 17) are not perfect: samples should ideally be selected proportionally to $w_i = m_i(Y_i)\hat{p}(Y_i)W_{X_i}\left|\frac{\partial T_i}{\partial X_i}\right|$, not $c_i\hat{p}(X_i)W_{X_i}$ in the source pixel. While our clustering by G-buffer helps, there is likely more to be done.

In cases where the initial sampling is very poor, our method exhibits similar "boiling" noise as many other ReSTIR methods due to correlations, where a single sample is spread to many pixels. This is a general limitation of many ReSTIR methods. A potential solution is to explicitly de-correlate the input samples, as done by Sawhney et. al. [SLK*22]. Our method could easily be combined with theirs by e.g., applying our method after their mutation sampling.

8. Conclusion

We present a novel, improved spatial reuse method for ReSTIR. By picking the actually reused samples based on their contributions and confidence weights, our method drastically helps in recovering from disocclusions, where temporal history is lost and has to be rebuilt.

While picking input pixels based on their samples would normally result in bias, we formulate our sample-based input selection as a stochastic resampling MIS weight, applied to a large number of input pixels chosen carefully but independently of their samples. Our reuse then has the same unbiased expectation as using non-stochastic MIS weights.

Our new spatial reuse can be easily integrated into existing ReSTIR renderers while maintaining high framerates. Our method has the highest improvement where temporal history is unavailable and must be rebuilt from a sparse set of samples, and improves the quality on all scenes we tested.

Acknowledgements

This work was partially funded Google and Adobe's gifts, and the NSF IIS grants 2127544 and 2238839. We thank Aaron Lefohn for the discussions and support, and the anonymous reviewers for their constructive feedback.

References

[Bit21] BITTERLI B.: *Correlations and reuse for fast and accurate physically based light transport.* PhD thesis, Dartmouth College, 2021. 4

[Boi21] BOISSÉ G.: World-space spatiotemporal reservoir reuse for ray-traced global illumination. In *SIGGRAPH Asia 2021 Technical Communications* (New York, NY, USA, 2021), SA '21, Association for Computing Machinery. URL: https://doi.org/10.1145/3478512.3488613, doi:10.1145/3478512.3488613. 6

[BSH02] BEKAERT P., SBERT M., HALTON J.: Accelerating path tracing by re-using paths. In *Eurographics Workshop on Rendering* (2002). 2

[BWP*20] BITTERLI B., WYMAN C., PHARR M., SHIRLEY P., LEFOHN A., JAROSZ W.: Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting. *ACM Trans. Graph. (Proc. SIGGRAPH) 39*, 4 (2020). 2

[Cha82] CHAO M. T.: A general purpose unequal probability sampling plan. *Biometrika 69* (1982), 653–656. 3, 6

[CSN*23] CHANG W., SIVARAM V., NOWROUZEZAHRAI D., HACHISUKA T., RAMAMOORTHI R., LI T.-M.: Parameter-space ReSTIR for differentiable and inverse rendering. In *SIGGRAPH Conference Proceedings* (2023). 2

[DKH*10] DAVIDOVIČ T., KŘIVÁNEK J., HAŠAN M., SLUSALLEK P., BALA K.: Combining global and local virtual lights for detailed glossy illumination. *ACM Trans. Graph. (Proc. SIGGRAPH Asia) 29*, 6 (2010). 2

[HKL*25] HEDSTROM T., KETTUNEN M., LIN D., WYMAN C., LI T.-M.: Restir bdpt: Bidirectional restir path tracing with caustics. *ACM Trans. Graph. 44*, 5 (Sept. 2025). URL: https://doi.org/10.1145/3744898, doi:10.1145/3744898. 2

[KBG24] KERN R., BRÜLL F., GROSCH T.: ReSTIR FG: Real-time reservoir resampled photon final gathering. In *Eurographics Symposium on Rendering* (2024), Haines E., Garces E., (Eds.). 2

[KCK*22] KALLWEIT S., CLARBERG P., KOLB C., DAVIDOVIČ T., YAO K.-H., FOLEY T., HE Y., WU L., CHEN L., AKENINE-MÖLLER T., WYMAN C., CRASSIN C., BENTY N.: The Falcor rendering framework, 8 2022. https://github.com/NVIDIAGameWorks/Falcor. URL: https://github.com/NVIDIAGameWorks/Falcor. 7

[Kel97] KELLER A.: Instant radiosity. In *SIGGRAPH* (1997), pp. 49–56. 2

[KLR*23] Kettunen M., Lin D., Ramamoorthi R., Bashford-Rogers T., Wyman C.: Conditional resampled importance sampling and restir. In *SIGGRAPH Asia (Conference Track)* (December 2023). doi:10.1145/3610548.3618245. 2

[LKB*22] Lin D., Kettunen M., Bitterli B., Pantaleoni J., Yuksel C., Wyman C.: Generalized resampled importance sampling: Foundations of restir. *ACM Transactions on Graphics 41*, 4 (2022), 75:1–75:23. doi:10.1145/3528223.3530158. 2, 3, 4, 8, 9

[NID20] Nabata K., Iwasaki K., Dobashi Y.: Two-stage resampling for bidirectional path tracing with multiple light sub-paths. *Comput. Graph. Forum (Proc. Pacific Graphics) 39*, 7 (2020), 219–230. 2

[OLK*21] Ouyang Y., Liu S., Kettunen M., Pharr M., Pantaleoni J.: ReSTIR GI: Path resampling for real-time path tracing. *Comput. Graph. Forum (Proc. High Performance Graphics) 40*, 8 (2021), 17–29. 2

[O'N14] O'Neill M. E.: *PCG: A Family of Simple Fast Space-Efficient Statistically Good Algorithms for Random Number Generation*. Tech. Rep. HMC-CS-2014-0905, Harvey Mudd College, Claremont, CA, Sept. 2014. 6

[PJH16] Pharr M., Jakob W., Humphreys G.: *Physically Based Rendering: From Theory to Implementation (3rd ed.)*, 3rd ed. Morgan Kaufmann Publishers Inc., oct 2016. 2

[PRDD15] Popov S., Ramamoorthi R., Durand F., Drettakis G.: Probabilistic connections for bidirectional path tracing. *Comput. Graph. Forum (Proc. EGSR) 34*, 4 (2015), 75–86. 2

[SIMP06] Segovia B., Iehl J. C., Mitanchey R., Péroche B.: Bidirectional instant radiosity. *Rendering Techniques (Proc. EGSR)* (2006), 389–397. 2

[SLK*22] Sawhney R., Lin D., Kettunen M., Bitterli B., Ramamoorthi R., Wyman C., Pharr M.: Decorrelating ReSTIR samplers via MCMC mutations. *arXiv preprint arXiv:2211.00166* (2022). 2, 10

[SLW22] Su F., Li S., Wang G.: SPCBPT: subspace-based probabilistic connections for bidirectional path tracing. *ACM Trans. Graph. (Proc. SIGGRAPH) 41*, 4 (2022). 2

[Tal05] Talbot J. F.: Importance resampling for global illumination. Brigham Young University. 3

[TCE05] Talbot J. F., Cline D., Egbert P.: Importance resampling for global illumination. *Rendering Techniques (Proc. EGSR)* (2005), 139–146. 2

[TH24] Tong X., Hachisuka T.: Efficient image-space shape splatting for monte carlo rendering. *ACM Trans. Graph. 43*, 6 (Nov. 2024). URL: https://doi.org/10.1145/3687943, doi:10.1145/3687943. 2

[VG95] Veach E., Guibas L. J.: Optimally combining sampling techniques for Monte Carlo rendering. In *SIGGRAPH* (1995), pp. 419–428. 2

[WGGH20] West R., Georgiev I., Gruson A., Hachisuka T.: Continuous multiple importance sampling. *ACM Trans. Graph. (Proc. SIGGRAPH) 39*, 4 (2020). 2

[WGH22] West R., Georgiev I., Hachisuka T.: Marginal multiple importance sampling. 2

[WKL*23] Wyman C., Kettunen M., Lin D., Bitterli B., Yuksel C., Jarosz W., Kozlowski P.: A gentle introduction to restir path reuse in real-time. In *ACM SIGGRAPH 2023 Courses* (New York, NY, USA, 2023), SIGGRAPH '23, Association for Computing Machinery. URL: https://doi.org/10.1145/3587423.3595511, doi:10.1145/3587423.3595511. 2, 4

[WSD24] Werner M., Schüssler V., Dachsbacher C.: ReSTIR subsurface scattering for real-time path tracing. *ACM Comput. Graph. Interact. Tech. (Proc. I3D) 7*, 3 (2024). 2

[WWWZ23] Wang Y.-C., Wyman C., Wu L., Zhao S.: Amortizing samples in physics-based inverse rendering using ReSTIR. *ACM Trans. Graph. (Proc. SIGGRAPH Asia) 42*, 6 (2023). 2

[ZLK*24] Zhang S., Lin D., Kettunen M., Yuksel C., Wyman C.: Area ReSTIR: Resampling for real-time defocus and antialiasing. *ACM Trans. Graph. (Proc. SIGGRAPH) 43*, 4 (2024), 98:1–98:13. 2

Appendix A: GRIS with Stochastic MIS

Here, we prove that GRIS with our stochastic resampling weights gives an unbiased contribution weight for the chosen output Y.

We first write the left side of Equation 3 as an expectation over all possible outputs (Y_i, W_{Y_i}), following Equation 7. We replace w_i with the stochastic $\tilde{w}_i$, which changes the selection probability of index i to $\tilde{w}_i / \sum_{j=1}^{M} \tilde{w}_j$, giving

$$\mathbb{E}\Big[f(Y)W_Y\Big] = \sum_{i=1}^{M} \mathbb{E}\Big[[\tilde{w}_i > 0]\, f(Y_i) W_{Y_i} \frac{\tilde{w}_i}{\sum_{j=1}^{M} \tilde{w}_j}\Big].$$

Substituting in W_{Y_i} in Equation 7 with stochastic $\tilde{w}_i$, we get

$$= \sum_{i=1}^{M} \left[[\tilde{w}_i > 0]\, f(Y_i) \left(\frac{1}{\hat{p}(Y_i)} \cancel{\sum_{j=1}^{M} \tilde{w}_j} \right) \frac{\tilde{w}_i}{\cancel{\sum_{j=1}^{M} \tilde{w}_j}} \right].$$

Next, we substitute in $\tilde{w}_i$ using Equation 14, and use the fact that $\tilde{w}_i = 0$ if $Y_i \notin \operatorname{supp} \hat{p}$. Denoting $[D_i] := [X_i \in \mathcal{D}(T_i)][Y_i \in \operatorname{supp} \hat{p}]$, we have

$$= \sum_{i=1}^{M} \mathbb{E} \left[[D_i]\, f(Y_i) \, \frac{\tilde{m}_i(Y_i; Z) \cancel{\hat{p}(Y_i)} W_{X_i} \left| \frac{\partial T_i}{\partial X_i} \right|}{\cancel{\hat{p}(Y_i)}} \right].$$

Next, we denote $\bar{X} = (X_1, \ldots, X_M)$ and $\bar{W} = (W_{X_1}, \ldots, W_{X_M})$, remembering that $Y_i = T_i(X_i)$. With this, we write

$$= \sum_{i=1}^{M} \mathbb{E} \left[\mathbb{E} \left[[D_i]\, f(Y_i) \tilde{m}_i(Y_i; Z) W_{X_i} \left| \frac{\partial T_i}{\partial X_i} \right| | \bar{X}, \bar{W} \right] \right].$$

This allows separating functions of X_i and W_{X_i} (like $[D_i]$, because $Y_i = T_i(X_i)$) from the stochastic MIS weight, giving

$$= \sum_{i=1}^{M} \mathbb{E} \left[[D_i]\, f(Y_i)\, \mathbb{E}\left[\tilde{m}_i(Y_i; Z) \mid \bar{X}, \bar{W}\right] W_{X_i} \left| \frac{\partial T_i}{\partial X_i} \right| \right].$$

By assuming Equation 13, this simplifies to

$$= \sum_{i=1}^{M} \mathbb{E} \left[[X_i \in \mathcal{D}(T_i)][Y_i \in \operatorname{supp} \hat{p}]\, f(Y_i)\; m_i(Y_i) W_{X_i} \left| \frac{\partial T_i}{\partial X_i} \right| \right].$$

Using Equation 3 and $Y_i = T_i(X_i)$, we convert the expectations into integrals as

$$= \sum_{i=1}^{M} \int\limits_{\operatorname{supp} X_i} [x \in \mathcal{D}(T_i)][T_i(x) \in \operatorname{supp} \hat{p}]\, f(T_i(x)) m_i(T_i(x)) \left| \frac{\partial T_i}{\partial x} \right| \mathrm{d}x$$

$$= \sum_{i=1}^{M} \int\limits_{\mathcal{D}(T_i)} [x \in \operatorname{supp} X_i][T_i(x) \in \operatorname{supp} \hat{p}]\, f(T_i(x)) m_i(T_i(x)) \left| \frac{\partial T_i}{\partial x} \right| \mathrm{d}x.$$

Change of variables $y = T_i(x)$ with inverse shifts $x = T_i^{-1}(y)$ gives

$$= \sum_{i=1}^{M} \int_{\mathcal{I}(T_i)} \left[T_i^{-1}(y) \in \operatorname{supp} X_i \right] [y \in \operatorname{supp} \hat{p}]\, f(y) m_i(y)\, \mathrm{d}y$$

$$= \sum_{i=1}^{M} \int_{\operatorname{supp} \hat{p}} \left[T_i^{-1}(y) \in \operatorname{supp} X_i \right] [y \in \mathcal{I}(T_i)]\, f(y) m_i(y)\, \mathrm{d}y$$

$$= \int_{\mathrm{supp}\ \hat{p}} \left(\sum_{i=1}^{M} \left[T_i^{-1}(y) \in \mathrm{supp}\ X_i \right] [y \in \mathcal{I}(T_i)]\, m_i(y) \right) f(y)\,\mathrm{d}y$$

$$= \int_{\mathrm{supp}\ \hat{p}} \left(\sum_{\substack{i=1 \\ y \in T_i(\mathrm{supp}\ X_i)}}^{M} m_i(y) \right) f(y)\,\mathrm{d}y$$

$$= \int_{\mathrm{supp}\ \hat{p} \cap \bigcup_i T_i(\mathrm{supp}\ X_i)} \left(\sum_{\substack{i=1 \\ y \in T_i(\mathrm{supp}\ X_i)}}^{M} m_i(y) \right) f(y)\,\mathrm{d}y$$

$$= \int_{\mathrm{supp}\ Y} \left(\sum_{\substack{i=1 \\ y \in T_i(\mathrm{supp}\ X_i)}}^{M} m_i(y) \right) f(y)\,\mathrm{d}y = \int_{\mathrm{supp}\ Y} f(y)\,\mathrm{d}y.$$

■

Appendix B: Stochastic MIS estimator

In this appendix, we prove that Equation 15 satisfies Equation 13.

We start by expanding the expectation of Equation 15, where Z is a vector of random numbers used for the random selections, given $\bar{X}$ and $\bar{W}$:

$$\mathbb{E}\left[\tilde{m}_i(y;Z) \mid \bar{X},\bar{W}\right] = \mathbb{E}\left[[P(i;\bar{X},\bar{W}) > 0] \frac{\mathbf{K}_{\bar{S}}(i)}{\tilde{N}P(i;\bar{X},\bar{W})} m_i(y) \mid \bar{X},\bar{W} \right].$$

We immediately see that if $m_i(y) = 0$, then $\mathbb{E}\left[\tilde{m}_i(y;Z) \mid \bar{X},\bar{W}\right] = 0 = m_i(y)$. Let us assume $m_i(y) > 0$, which we assume implies $P(i;\bar{X},\bar{W}) > 0$. Then,

$$\mathbb{E}\left[\tilde{m}_i(y;Z) \mid \bar{X},\bar{W}\right] = \mathbb{E}\left[\frac{\mathbf{K}_{\bar{S}}(i)}{\tilde{N}P(i;\bar{X},\bar{W})} m_i(y) \mid \bar{X},\bar{W} \right] \quad (20)$$

$$= \mathbb{E}\left[\mathbf{K}_{\bar{S}}(i) \mid \bar{X},\bar{W} \right] \frac{1}{\tilde{N}P(i;\bar{X},\bar{W})} m_i(y). \quad (21)$$

Multiplicity $\mathbf{K}_{\bar{S}}(i)$ of sampling with replacement follows a binomial distribution with parameters $\tilde{N}$ and $P(i;\bar{X},\bar{W})$. Hence, we have

$$= \frac{\cancel{\tilde{N}P(i;\bar{X},\bar{W})}}{\cancel{\tilde{N}P(i;\bar{X},\bar{W})}} m_i(y). \quad (22)$$

■