Histogram Stratification for Spatio-Temporal Reservoir Sampling

CORENTIN SALAÜN, Max Planck Institute for Informatics, Germany
MARTIN BALINT, Max Planck Institute for Informatics, Germany
LAURENT BELCOUR, Intel, France
ERIC HEITZ, Intel, France
GURPRIT SINGH, Max Planck Institute for Informatics, Germany
KAROL MYSZKOWSKI, Max Planck Institute for Informatics, Germany

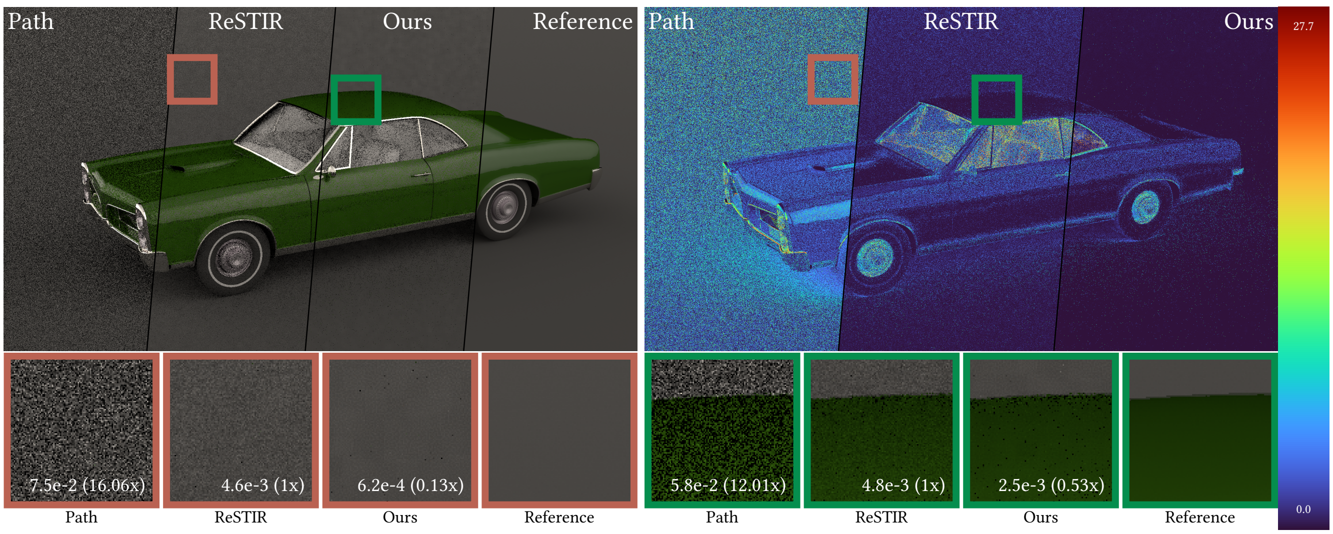


Fig. 1. Rendering comparison between path tracing, spatio-temporal reservoir resampling (ReSTIR) [Bitterli et al. 2020] and our stratified histogram sampling method at the 4th rendering frame. By locally ordering candidates for the resampling estimator, our method enables the selection of candidates with stratified properties, efficiently reducing rendering error. This error reduction is also visible in the RelMSE error map on the right side. The closeup view shows in detail the noise reduction and the relative difference in RelMSE compared to ReSTIR.

Monte Carlo (MC) rendering is a widely used approach for photorealistic image synthesis, yet real-time applications often limit sampling to one path per pixel, resulting in high noise levels. To mitigate this, resampled importance sampling (RIS) has shown promise by approximating ideal sample distributions through a discrete set of candidates, avoiding the complexity of neural models or data-intensive structures. However, current RIS techniques often rely on random sampling, which fails to maximize the potential of the candidate pool. We propose a two step approach that first organizes samples candidates into local histograms and then sample the histogram using Quasi Monte Carlo and antithetic patterns. This can be done with minimal overhead and allows to reduce error in rendering to increase visual quality. Additionally, we show how it can be combined with blue noise error distribution to perceptually reduce noise artifacts. Our approach yields a higher-quality resampling estimator with enhanced noise reduction, demonstrating significant improvements in real-time rendering tasks.

CCS Concepts: • **Computing methodologies** → **Ray tracing**.

Additional Key Words and Phrases: Ray Tracing, Quasi-Monte Carlo, Global Illumination, Real-Time Rendering

ACM Reference Format:
Corentin Salaün, Martin Balint, Laurent Belcour, Eric Heitz, Gurprit Singh, and Karol Myszkowski. 2025. Histogram Stratification for Spatio-Temporal Reservoir Sampling. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers (SIGGRAPH Conference Papers '25), August 10–14, 2025, Vancouver, BC, Canada.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3721238.3730723

Authors' Contact Information: Corentin Salaün, Max Planck Institute for Informatics, Saarbrücken, Germany, csalaun@mpi-inf.mpg.de; Martin Balint, Max Planck Institute for Informatics, Saarbrücken, Germany, mbalint@mpi-inf.mpg.de; Laurent Belcour, Intel, Grenoble, France, laurent.belcour@gmail.com; Eric Heitz, Intel, Grenoble, France, eheitz.research@gmail.com; Gurprit Singh, Max Planck Institute for Informatics, Saarbrücken, Germany, gurpritsbagga@gmail.com; Karol Myszkowski, Max Planck Institute for Informatics, Saarbrücken, Germany, karol@mpi-inf.mpg.de.

SIGGRAPH Conference Papers '25, Vancouver, BC, Canada

ACM ISBN 979-8-4007-1540-2/25/08
https://doi.org/10.1145/3721238.3730723

1 Introduction

Monte Carlo (MC) rendering has become a cornerstone of modern real-time rendering techniques, particularly with the recent advancements in hardware acceleration for ray tracing. This method is central to achieving photorealistic images by simulating light transport through random sampling of light paths. However, even with cutting-edge hardware, real-time applications often limit the number of samples to a single path per pixel, which results in noisy images with reduced perceptual quality due to the inherently stochastic nature of the sampling process.

To mitigate this noise, noise reduction techniques have emerged to improve the quality of MC and make it viable. Some *simple* methods such as better importance sampling [Litalien et al. 2024], low-discrepancy sequences [Christensen et al. 2018], antithetic sampling [Wang et al. 2023], control variates [Salaün et al. 2022b; Nicolet et al. 2023], and screen space error distribution [Heitz and Belcour 2019], are already compatible with the computational constraints of real-time path tracing. More *advanced* approaches such as path guiding, require complex data structures [Vévoda et al. 2018] or neural models [Müller et al. 2019], which introduce non-negligible computational overhead and complexity, limiting their use.

Recently, a novel approach based on resampled importance sampling (RIS) [Talbot 2005] has garnered attention. This method offers mathematically sound improvements in sampling distribution without the complexity of learning models or intricate data structures. By employing a large population of samples to approximate the ideal sampling distribution and then resampling this approximation, this approach has demonstrated significant improvements in rendering tasks like direct lighting [Bitterli et al. 2020], global illumination [Ouyang et al. 2021; Lin et al. 2022], and volumetric rendering [Lin et al. 2021]. It is particularly effective when combined with spatial and temporal reuse of samples, showing promise in real-time settings.

Despite the work of Ciklabakkal et al. [2022] aiming to combine RIS with QMC sequences, limited research has been done to combine variance reduction techniques with RIS. This work seeks to address this gap by demonstrating how to effectively combine simple noise reduction techniques with RIS at minimal computational cost. Key to our work is the idea that locally organizing RIS candidates prior to sample selection allows us to introduce different noise reduction techniques that increase the effectiveness of the algorithm. In summary, the main contributions of this work are:

- The construction of a local approximation of the distribution function for candidate selection.
- A method for selecting stratified candidate for a resampling estimator in both spatial and temporal domains.
- The integration of this error-reduction sampling technique with perceptual error distribution to further enhance rendering quality with minimal overhead.

2 Related work

Quasi-Monte Carlo sampling. Quasi-Monte Carlo (QMC) methods aim to improve the efficiency of Monte Carlo integration by using well-distributed, deterministic sample sets instead of random, independently distributed samples. The primary objective of QMC sampling is to distribute samples as uniformly as possible over the integration domain, enabling more effective and accurate function sampling. Over the years, numerous techniques have been proposed for constructing sample sets with varying degrees of quality and simplicity. Among the most notable are stratified sampling, which divides the sampling domain into non-overlapping regions with exactly one sample per region; Sobol sequences [Sobol' 1967; Doignies et al. 2024], known for their progressive and low-discrepancy construction; PMJ02 sequences [Christensen et al. 2018], which achieve multiple levels of stratification simultaneously; and Rank-1 lattices [Dammertz and Keller 2006], which follow a shifted grid pattern to maintain uniformity. Each of these methods improves sample distribution, leading to reduced variance and lower integration error compared to purely random sampling.

In Monte Carlo rendering, QMC sampling has become a cornerstone technique due to its ability to reduce noise and enhance image quality [Keller 1996; Singh et al. 2019]. Moreover, QMC methods often exhibit improved convergence rates compared to random sampling, achieving equal image quality with a lower sample count. This accelerates rendering performance for the same noise level while maintaining visual accuracy.

Antithetic Sampling. Antithetic sampling represents an alternative strategy for distributing multiple samples that, unlike quasi-Monte Carlo (QMC) methods focused on uniformity, aims to reduce error by balancing sample values [Kroese et al. 2013, Chapter 9.2]. The core idea is to select sample points that, when averaged, compensate each other to approximate the true function average more effectively. For example, a high-value sample is paired with a corresponding low-value sample to achieve error cancellation. While this approach can be challenging for Monte Carlo rendering—where function values are generally unknown in advance and poorly chosen pairs may increase error—it proves particularly useful for monotonic functions. In these cases, sampling can be guided by expected function behavior, enabling more controlled and effective error reduction. Antithetic sampling has shown promise in applications where sampling symmetry or compensation is feasible, complementing other variance reduction techniques [Subr et al. 2014]. The most common application is its use for inverse rendering application and gradient estimation [Zhang et al. 2021; Wang et al. 2023; Belhe et al. 2024]. Instead, our work emphasizes the application of antithetic sampling in traditional rendering.

Reservoir re-sampling. Building on the foundational concepts of importance sampling, Resampled Importance Sampling (RIS), introduced by Talbot [2005], tackles the challenge of dealing with desired sampling distributions that are too complex for analytical construction. RIS achieves this by approximating the target distribution through a set of candidate samples. These candidates are drawn from a simpler, sub-optimal distribution. The key idea lies in resampling a single sample from this candidate pool, where the probability of selection is proportional to the desired weight in the target distribution. This approach eliminates the need to compute the entire target distribution, requiring only evaluations at specific points. By repeating this resampling process, RIS builds a multiple-sample estimator that converges to the desired distribution.

Expanding upon RIS, Bitterli et al. [2020] presented Spatiotemporal Reservoir Resampling (ReSTIR). This technique employs a stream of candidate light samples, managed through Weighted Reservoir Sampling, to probabilistically resample candidates based on their anticipated contribution. To enhance efficiency, ReSTIR integrates candidates reservoir from both spatial and temporal neighboring pixels. This strategy allows for an unbiased rendering process without the need for complex data structures. Building on ReSTIR, recent works like ReSTIR GI [Ouyang et al. 2021] and ReSTIR PT [Lin et al. 2022] extend the algorithm to handle more general light transport scenarios. Advancements have been made to generalize ReSTIR to more complex rendering scenarios, such as subsurface scattering [Werner et al. 2024] and advanced camera modeling [Zhang et al. 2024]. ReSTIR has also shown potential for inverse rendering applications [Chang et al. 2023].

Ciklabakkal et al. [2022] introduced a stratified resampling method for ReSTIR that can produce blue noise error distribution in screen space for low-dimensional paths. Key to their method is a single-pass bidirectional CDF sampling approach that aims to stratify samples in primal space along a Hilbert curve. In contrast, our method scales to high-dimensional paths by stratifying samples in the 1D space of luminances, where CDF sampling simplifies to a sorting algorithm. Hence our method both stratifies samples in a more meaningful space and avoids a bidirectional setup by leveraging simultaneous access to all candidates. Additionally, Ciklabakkal et al. [2022] do not apply stratification during the initial candidate sampling and the resampling phase. Although it looks like both methods share similarities, they address different problem spaces.

3 Preliminaries

3.1 Spatio-temporal reservoir re-sampling

Resampled importance sampling. Resampled Importance Sampling (RIS) is a method designed to generate a sample proportionally to a target distribution q, from which direct sampling is not possible. Instead, we generate a set of k random candidates from an initial distribution p (Fig. 2 in red). For each candidate x_i, a weight $w_i = \frac{q(x_i)}{p(x_i)}$ is computed. A sample is then selected from the set by drawing according to the discrete probability distribution defined as $P(i) = \frac{w_i}{\sum_{j=1}^{k} w_j}$. This step is illustrated by the resampling arrow in Fig. 2. Using this two-step sampling procedure, the generated sample probability density function (PDF) approaches q as $k \rightarrow \infty$.

This approach can be used in an unbiased Monte Carlo estimator, as demonstrated by Talbot [2005], to estimate an integral using the following estimator:

$$\int_{\Omega} f(x)dx \approx \hat{I}_{RIS} = \frac{f(x_i)}{q(x_i)} \cdot \left(\frac{1}{k} \sum_{j=1}^{k} w_j \right) \tag{1}$$

where x_i is the selected sample from the k candidates, and the sum of the weights normalizes the estimator.

Spatio-temporal resampling. In the context of rendering, the RIS estimator can be effectively used to reduce noise by generating samples that better approximate the ideal light distribution in a scene. However, generating k candidates per pixel can be computationally expensive. To address this, spatio-temporal candidate reuse is employed, where sample candidates are shared across neighboring pixels and across frames in temporal sequences. This reuse of candidates amortizes the cost of candidate generation, making RIS particularly efficient in real-time applications.

This approach produces higher-quality output compared to simple Monte Carlo estimator, where only one light path per pixel is typically sampled without regard for neighboring pixel or frame information. By leveraging multiple candidate samples and reusing them efficiently, RIS with spatio-temporal reuse better approximates the ideal distribution for each pixel, resulting in lower noise and higher overall image quality, particularly in complex lighting scenarios such as indirect illumination.

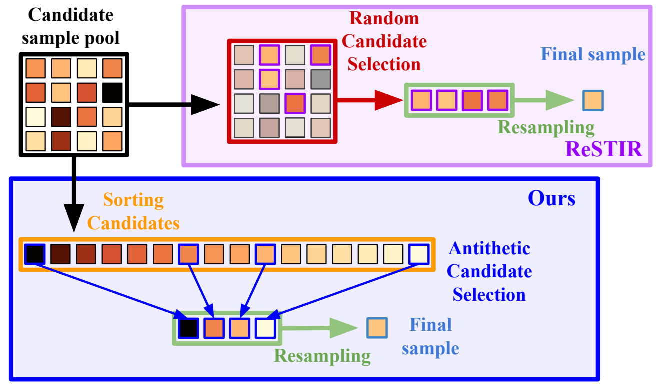


Fig. 2. Visualization of the difference between a classical RIS estimator and our proposed stratified histogram sampling. Starting from a same pool of candidates, RIS selects uniformly a random subset of candidates, used for a resampling step where a single candidate is finally selected. This second sampling is done with importance sampling. Instead, our method starts by sorting the candidates based on their output value before selecting the candidates for re-sampling. This selection is done using antithetic sampling, selecting samples with symmetric properties. Finally the resampling step is done similarly to RIS estimator.

3.2 Histogram sampling

Histogram sampling was introduced by Heitz and Belcour [2019] to provide an efficient method for sampling complex, high-dimensional functions by reformulating the integration problem into a simpler, one-dimensional space. Instead of directly sampling the original function $f(x)$, histogram sampling estimates the distribution of function values and then samples from this distribution using inverse cumulative distribution function (CDF) sampling. This approach abstracts away the complexities and variations of the original function, leveraging the properties of a 1D monotonic function with limited variation, which is highly compatible with Quasi-Monte Carlo (QMC) methods known for their variance reduction capabilities.

The histogram estimation process begins by generating random samples from the original function's sampling space to approximate the probability distribution of its output values. As the number of samples increases, the accuracy of the histogram improves. Sampling from this histogram is straightforward: the inverse CDF of the histogram is constructed, allowing direct sampling of the function values. The integral of the function over its domain can thus be transformed as:

$$\int_{\Omega} f(x)\, dx = \int_{\mathbb{R}} H(y)\, dy, \quad H(y) = p(y) \cdot y, \tag{2}$$

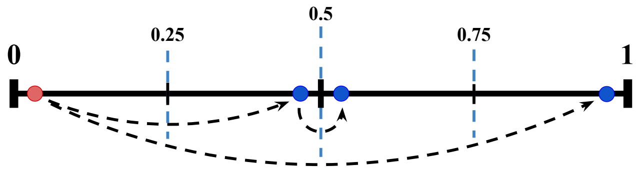


Fig. 3. Visualization of our 4 spp antithetic sequence. The initial sample (red dot) is sampled from the range $[0, 0.25]$, and the 3 other samples (blue dot) are constructed by mirroring around 0.25, 0.5, and both successively.

where $H(y)$ represents the histogram of function values $f(x)$. As demonstrated by Heitz and Belcour [2019], the inverse CDF sampling of a histogram is mathematically equivalent to sampling the index of a sorted list, enabling histogram sampling by simply sorting the samples and selecting from them uniformly.

Despite its benefits, histogram sampling has a notable drawback: constructing an accurate histogram requires a large number of samples, which can often be more costly than using all the samples directly in a standard Monte Carlo estimator. However, when the cost of histogram estimation can be reduced, the method provides substantial benefits. Heitz and Belcour [2019] demonstrated its use for blue-noise error distributions to improve perceptual quality. Meanwhile, we use it to enable stratified and antithetic sampling for resampling-based rendering, directly reducing rendering error.

4 Correlated histogram sampling for resampling candidates selection

In a ReSTIR-based rendering setup, each frame involves generating new random samples for every pixel. In the standard method, K candidates are randomly selected from neighboring pixels close to the target pixel for the resampling step (see Fig. 2 top). In contrast, we propose a more structured approach by constructing a histogram of all candidates to guide a more effective candidate selection process. By choosing candidates with improved distribution properties, the selected K candidates will better approximate the ideal target distribution, leading to a reduction in the final estimation error. Figure 2 (bottom) shows the main difference between a classical ReSTIR estimator and our method with the sorting of the candidates and the use of an antithetic sampling.

This approach leverages the core assumption from ReSTIR: neighboring pixels in an image evaluate similar integration functions, making candidate reuse across pixels not only viable but beneficial. By drawing from the candidate pool of adjacent pixels, we exploit spatial coherence, resulting in better sample diversity and smoother integration.

4.1 Local histogram construction

The key enhancement in our pipeline compared to the standard ReSTIR approach is the explicit construction of a local histogram of candidates. Constructing a histogram for every pixel at each frame would be computationally prohibitive. Instead, we build a shared histogram for a fixed-size group of pixels, referred to as a block, that are tilled over the image without overlap. Typical block sizes range from 8×8 to 32×32 pixels, balancing computational efficiency with better histogram estimation.

Rather than constructing the explicit histogram and its inverse cumulative distribution function (CDF) separately, we directly construct the inverse CDF, which can be sampled efficiently. As discussed in Section 3.2, this process is equivalent to sorting the candidates based on their function value, specifically the total luminance estimated from the path associated with each candidate. This ordering process is visualized in Fig. 2, where the initial pool of random candidates is sorted into a 1D list. The sorted candidates form a direct representation of the inverse CDF of the block. By leveraging this structure, our method significantly improves the quality of selected candidates for resampling, directly contributing to reduced variance in the final rendered image.

4.2 Stratified histogram sampling

We improve candidate selection in ReSTIR by applying stratified sampling across frames and within each frame. Since ReSTIR is a temporal method, coordinating sampling over multiple frames helps reduce noise and improve stability. Our approach uses a single Sobol sequence per pixel that will be used across frames to guide both candidate selection from a sorted list and the final resampling step. Such a sampling sequence benefits from stratified properties by construction.

At each frame, a single sample from the Sobol sequence provides values for the two sampling steps. The first dimension is used to select multiple candidates from the ordered list. This simultaneous sampling is obtained by an antithetic sampling strategy constructed from a single pseudo-random number. The second dimension determines the final candidate for resampling, chosen proportionally to its weight q (Section 3.1). Unlike the classical ReSTIR approach, which uses streaming reservoir updates, our method directly resamples using inverse CDF sampling with a single pseudo-random number, simplifying the process. Contrary to Ciklabakkal et al. [2022] which required a space filling curve (in sample space) to select stratified candidates, we obtain stratified samples directly from the candidate's output values, making this approach independent of the rendering method.

The antithetic sampling method that we used generates a symmetric and stratified pattern following the construction shown in Fig. 3. This construction is adapted for the sampling of 4 elements but could be extended to other power of two number of candidates. Such patterns are effective for reducing error in monotonic functions as symmetric samples balance high and low values, helping to compensate for variations.

By combining stratification in both sampling steps, our method improves the distribution of candidates contributing to the second term of Eq. (1), reducing estimation error and enhancing rendering quality.

4.3 Spatial masking

To maximize local similarity within a neighborhood of pixels and construct high-quality candidate histograms, we apply spatial filtering based on G-buffer information, ensuring that only similar pixels contribute to the same histogram. For each block of pixels, a clustering process groups pixels with similar G-buffer attributes. In our implementation, object ID was used as the primary guide for

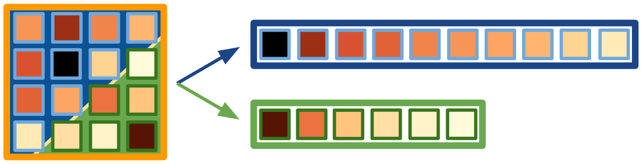


Fig. 4. Visualization of spatial masking on candidate sorted list. Two overlapping objects in the same block lead to the construction of separate sorted list of candidates to prevent path reuse from different objects.

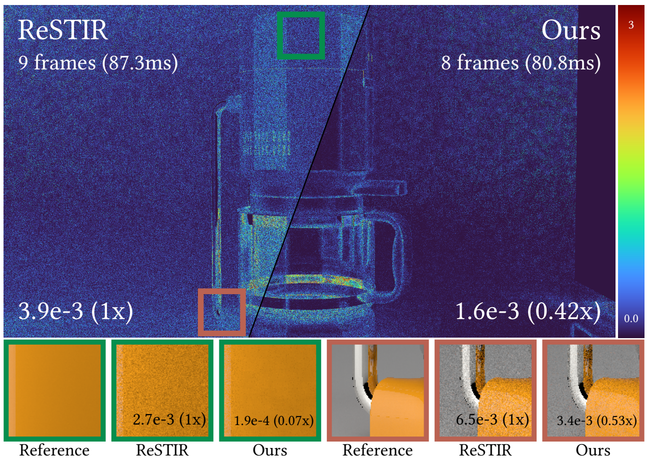


Fig. 5. **Comparison with temporal accumulation.** Equal time comparison of our method (8-frame temporal accumulation) against ReSTIR (9-frame temporal accumulation) using frame accumulation instead of temporal candidate sampling. RelMSE error maps and close-ups highlight significant error reduction across the image using our method in slightly less time. RelMSE values and relative differences to ReSTIR are provided for the entire image and the close-ups.

clustering; however, other attributes such as surface normals, depth, or material properties could yield similar improvements. Fig. 4 provides a visualization of a sorting step for a block containing two different objects. During resampling, each pixel samples candidates exclusively from the sorted list of candidates of its cluster (same object ID in our case). This spatial filtering enhances the coherence of candidate selection, reducing noise and improving resampling accuracy. Similar rejection-based approaches using surface normals and depth have been previously explored in the context of importance sampling, as demonstrated by Bitterli et al. [2020]. This localized filtering technique ensures that sampling distributions better match local scene characteristics, improving the overall rendering quality.

5 Experiments

5.1 Implementation setup

Rendering Setup. We implemented our ReSTIR-based method in the Falcor renderer [Kallweit et al. 2022], leveraging its provided path-tracing implementation and following ReSTIR PT algorithm [Lin et al. 2022]. Our resampling strategy uses the total path luminance, including visibility terms, as the target density. Each rendering frame consists of three primary passes: (1) candidate generation, where random paths are generated and rendered for each pixel; (2) inverse CDF construction, where local candidate

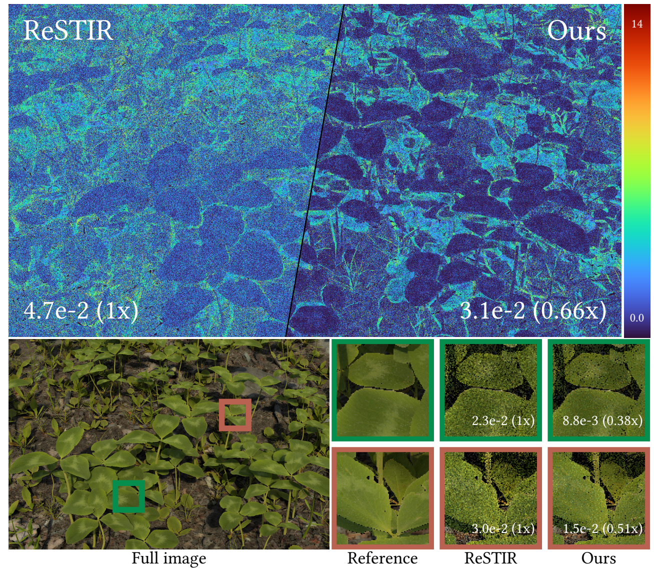


Fig. 6. **Comparison using temporal reservoir.** RelMSE error map and visual comparison (equal samples) of our method and ReSTIR on the second rendering frame and a static image. Error maps, close-up and relative difference highlight the reduced error achieved by our method.

sorting is performed to build the histogram structure; and (3) the resampling and final rendering step. All experiments were conducted on an NVIDIA RTX 3090 using DirectX acceleration. Rendering timing where also evaluated on Intel ARC A770.

Implementation Details. We utilized a bitonic sorting algorithm to leverage GPU acceleration for ordering candidates. In the conducted experiments, a block size of 16×16 pixels was used. Our spatial masking using object ID allows for as many cluster as objects on the block. For resampling, four candidates per pixel were selected using our antithetic strategy. As proposed by Bitterli et al. [2020], we limited the reservoir accumulation by capping the maximum number of sample accumulations. This maximum was set to 20 candidates, and an exponential moving average (EMA) was applied once this threshold was exceeded.

Sampling. For sampling, we employed a perceptually optimized XOR Sobol sequence, as described by Heitz et al. [2019], to drive our multi-frame Sobol sampler. Each pixel was assigned a unique Sobol sequence, ensuring a well-distributed sampling pattern across frames. This strategy improves both temporal stability and error distribution.

Evaluation. To evaluate the performance of our method, we compared it to the baseline ReSTIR algorithm using two primary metrics: Mean Squared Error (MSE), Relative Mean Squared Error (RelMSE). The MSE is defined as MSE $= \frac{1}{N}\sum_{i=1}^{N}(f_i - \mathbb{E}[f])^2$, where f_i represents the rendered pixel value and $\mathbb{E}[f]$ is the reference value. The RelMSE normalizes this by the reference value, computed as RelMSE $= \frac{1}{N}\sum_{i=1}^{N}\frac{(f_i-\mathbb{E}[f])^2}{\mathbb{E}[f]+0.0001}$, making it suitable for scenes with varying luminance.

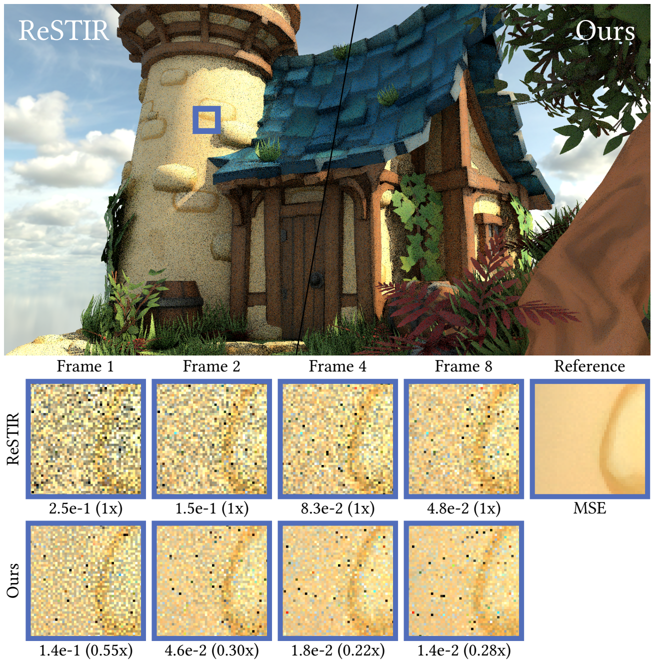


Fig. 7. **Evolution with increased temporal candidates.** We compare the rendering quality of our method (last row) compared to ReSTIR (first row) when accumulating 1, 2, 4, and 8 frames for a fixed viewpoint. For each number of accumulation, we consistently obtain a lower MSE than ReSTIR (computed for each crop). For the first frame of accumulation (frames 1 and 2), we also benefit from perceptual error distribution which reduce the perception of noise.

Overhead. Our method introduces a minor overhead. The memory footprint is minimal, requiring only storage for luminance (one float per reservoir) and G-buffer information (one integer per reservoir) for sorting purposes. The time overhead, primarily due to the ordering of candidates during histogram construction, remains relatively small, adding only 3 to 10% to the overall rendering time in our experiments, depending solely on the block size.

5.2 Results

Figures 5 and 10 presents a rendering comparison without temporal candidate reuse at equal time. In this setup, each frame independently employs spatial candidate sampling without relying on reservoirs from previous frames. Instead, frames are directly accumulated to produce a multi-sample rendering. In this context, our method demonstrates superior performance, achieving both significant error reduction and an improved perceptual error distribution, resulting in smoother, visually pleasing images. Unlike temporal candidate resampling, where the reuse of samples across frames can diminish perceptual error properties, explicit accumulation better preserves these properties, enhancing the overall visual quality. This approach allows us to fully leverage the benefits of our stratified histogram sampling, which not only reduces per-pixel error but also distributes error across the image, further improving perceptual quality.

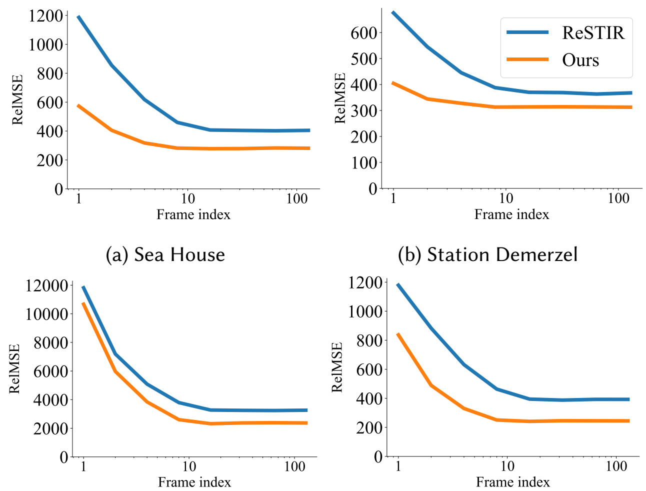
We compared our method to ReSTIR with temporal resampling across five scenes and various frame counts (2, 4, 8, 16, and 32), as shown in Figs. 1, 6 and 12. For each scene, we present visual comparisons, RelMSE error maps, and close-up views to highlight detailed improvements. All scenes are illuminated using environment maps, except the Veach-Ajar scene, which features only indirect illumination. In Fig. 6, the second frame demonstrates the benefits of our method before the candidate accumulation limit is reached. In this scenario, our approach significantly reduces noise and achieves a visibly lower RelMSE compared to ReSTIR. Moreover, thanks to the perceptually optimized Sobol sequence from Heitz et al. [2019], our method provides a more favorable perceptual error distribution. In the Station Demerzel scene (Fig. 12-top), most of the remaining error manifests as color noise rather than luminance noise, likely due to the use of a single shading ray during final rendering leading to a remaining high relative error while having smoother rendering in the close ups. In the Kitchen scene (Fig. 12-middle), improvements are observed in smooth regions but are limited by the use of a single shading ray at the final stage. Most of the error comes from shadowed ray, and very little noise remains in the other pixels. This error reduction is visible in the error map; however, the average value over the entire image does not show significant reduction due to the error being dominated by some high-error regions for both methods. Finally, in the Veach-Ajar scene (Fig. 12-bottom), rendered under indirect illumination, our method enhances rendering quality by leveraging antithetic sampling to improve candidate selection. This reduces the impact of outliers, further improving the overall visual fidelity.

Table 1. This table shows the rendering time per frame for two scenes as described in Fig. 9 with various block sizes. The values in parentheses indicate the percentage of rendering time allocated to the additional requirements of our method compared to ReSTIR. Based on these results, the best balance between computation time and quality is achieved with 16×16 blocks.

	Render time in ms (Overhead in % of the rendering)			
Scene	block 4×4	block 8×8	block 16×16	block 32×32
RTX 3090				
Living-room	15.1 (2.1%)	14.9 (1.1%)	15.8 (3.4%)	19.0 (7.9%)
Staircase	9.9 (3.4%)	9.7 (1.8%)	10.4 (6.3%)	13.6 (13.7%)
ARC A770				
Living-room	41.3 (2.5%)	42.0 (2.0%)	44.9 (2.5%)	55.8 (3.0%)
Staircase	36.5 (2.3%)	36.7 (2.0%)	40.5(2.5%)	51.2 (3.4%)

The full image error evolution with increasing frame count, shown in Fig. 8, provides further evidence of our method's effectiveness. This figure illustrates the RelMSE progression across four scenes: Sea House, Station Demerzel, and Sky Home and Veach Ajar. In all cases, our method consistently outperforms ReSTIR at every frame count. The plots exhibit a typical convergence pattern, with the error stabilizing after a few frames as the resampling process reaches its candidate accumulation limit. This behavior limits further reductions in error, but in all four scenes, our method achieves a consistently lower equilibrium error than ReSTIR. Our method remains limited by color noise—that a single shading ray cannot reduce—and the high variance region(s) that can dominate full image error. Still, these results demonstrate that our approach leads to lower error and improved overall rendering quality.

In Fig. 7, we present the evolution of rendering quality in an animated sequence, comparing our method with ReSTIR. The main comparison highlights the visual results at the 16th rendered frame.



(a) Sea House (b) Station Demerzel

(c) Veach ajar (d) Sky home

Fig. 8. **Evolution of RelMSE with the frame rendered.** This figure shows the evolution of the relative mean squared error (RelMSE) with respect to the number of rendered frames. Rendering without temporal accumulation, but with temporal candidate resampling, leads to improved results over time. Error stagnation occurs when the maximum number of samples is reached before applying exponential moving average (EMA). Our method consistently lowers error per frame, converging to a more accurate equilibrium state.

Close-up views illustrate the progression across different frames, showcasing consistent error reduction over time. For each close-up region, we include MSE comparisons for both methods and the relative error difference between them. Our approach consistently outperforms ReSTIR across all frames, achieving lower error and smoother results. Notably, in the early frames, some high-frequency error patterns are visible in our method due to the use of the perceptually optimized Sobol sequence from Heitz et al. [2019]. This sequence enhances perceptual error distribution, contributing to the overall improvement in rendering quality.

We also analyzed the impact of block size on candidate sorting, resampling, error reduction, and rendering performance. This evaluation, detailed in Fig. 9, compares renderings with block sizes ranging from 4×4 to 32×32 across two indoor scenes. For each block size, two close-up insets are shown, depicting the RelMSE and relative error relative to the 4×4 block. Additionally, Table 1 provides the render times per frame for each block size, highlighting the sorting overhead compared to standard ReSTIR. The results indicate that smaller block sizes yield faster render times but result in higher error and more noticeable low-frequency artifacts. In contrast, larger block sizes significantly increase computation time due to sorting overhead, with diminishing quality gains and some degradation at 32×32 compared to 16×16. The slight increase in error for the largest block size can be explained by the reduced similarity between the distant pixels of a large block. This reduced similarity affects path reuse, leading to a slight degradation in stratification quality as path throughput change when reuse in different pixel and increased error. Based on these findings, a block size of 16×16 offers the best trade-off, achieving lower error with a modest performance overhead of approximately 3% to 7%, depending on the scene.

Table 2. Ablation of histogram sampling strategies with RelMSE and PrelMSE averaged over 4 Scenes (Station-Demerzel, Sea-House, Sun Temple and Veach Ajar).

Method	RelMSE	pRelMSE
Uncorrelated candidates (ReSTIR)	0.049	0.132
Heitz et al. 2019a (4 samples per frame)	0.034	0.100
Random offset stratification	0.043	0.103
Random offset + temporal blue noise	0.034	0.095
Antithetic + temporal blue noise (Ours)	**0.034**	**0.095**

Figure 11 presents a comparison between our method and ReSTIR using multiple temporal reservoirs. This is accomplished by sampling temporal reservoirs similarly to spatial candidates—first by constructing a histogram of the temporal reservoir and then sampling it using antithetic strategies. Our method benefits from using multiple candidates, as this approach enables stratification. However, this comes with additional overhead, comparable to that of frame candidates histogram.

Table 2 presents an ablation of histogram sampling methods: Uncorrelated random, Low discrepancy (multiple samples [Heitz et al. 2019]), Random offset stratification (per frame, then with temporal coherence using single sample [Heitz et al. 2019]), and our full method (antithetic per frame + temporal coherent [Heitz et al. 2019], 1 sample/frame). Evaluated with RelMSE and perceptual pRelMSE (RelMSE on Gaussian-filtered output to account for blue noise benefits [Salaün et al. 2022a]). Our full method yields the best results for both metrics.

5.3 Discussion and future work

The use of block-based histograms for candidate selection can introduce visual artifacts that manifest as block-like patterns in the raw image, particularly when significant statistical differences exist between neighboring histograms. Such artifacts diminish as either the block size or the number of accumulated frames increases, smoothing the distribution of samples across the image. Additionally, shifting the block positions between frames can help mitigate these artifacts by averaging out local variations over time.

A promising direction for future work is evaluating the impact of the positive correlation between neighboring pixels introduced by ReSTIR, as they often share similar candidates. While this improves temporal stability, it may negatively affect neural denoisers used in real-time applications. Understanding the magnitude of this effect and exploring whether it can be leveraged to further enhance rendering quality presents an important opportunity for future research.

Our stratification algorithm, to our understanding, does not introduce bias into the ReSTIR framework. However, the deterministic nature of our sampling could affect error distribution with a fixed sample count, though not as a systematic bias. The synergistic combination of advanced ReSTIR techniques such as ReSTIR GI

[Ouyang et al. 2021] or AreaReSTIR [Zhang et al. 2024] and theoretical understanding with state of the art sampling methods presents a significant and interesting direction for future research.

6 Conclusion

We presented a method to improve candidate selection in ReSTIR by integrating stratified sampling techniques across frames and within each frame. Our approach leverages Sobol sequences to guide the sampling process, using a single sample per pixel at each frame to drive both candidate generation and resampling. Unlike classical ReSTIR, which employs streaming reservoir updates, we use direct inverse CDF sampling for resampling, simplifying the implementation and reducing estimator error. The introduction of antithetic sampling further refines candidate selection by generating symmetric patterns that effectively reduce error when sampling candidates.

Acknowledgments

We would like to thank the anonymous reviewers for their detailed and constructive comments. We gratefully acknowledge the support of the Saarland/Intel Joint Program on the Future of Graphics and Media. We thank the following for scenes used in our experiments: 3dhdscan daniel sklář (Clover), Sander Vander Meiren (Sky home), wa_rouk (Station Demerzel), Alyona Shek (Sea House), Artjoms Horosilovs (Lonely Watcher) and Eric Veach (Veach-Ajar).

References

Yash Belhe, Bing Xu, Sai Praveen Bangaru, Ravi Ramamoorthi, and Tzu-Mao Li. 2024. Importance Sampling BRDF Derivatives. *ACM Trans. Graph.* 43, 3, Article 25 (April 2024), 21 pages. https://doi.org/10.1145/3648611

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 Transactions on Graphics (TOG)* 39, 4 (2020), 148–1.

Wesley Chang, Venkataram Sivaram, Derek Nowrouzezahrai, Toshiya Hachisuka, Ravi Ramamoorthi, and Tzu-Mao Li. 2023. Parameter-space ReSTIR for Differentiable and Inverse Rendering. In *ACM SIGGRAPH 2023 Conference Proceedings* (Los Angeles, CA, USA) *(SIGGRAPH '23)*. Association for Computing Machinery, New York, NY, USA, Article 18, 10 pages. https://doi.org/10.1145/3588432.3591512

Per Christensen, Andrew Kensler, and Charlie Kilpatrick. 2018. Progressive multi-jittered sample sequences. In *Computer Graphics Forum*, Vol. 37. Wiley Online Library, 21–33.

Ege Ciklabakkal, Adrien Gruson, Iliyan Georgiev, Derek Nowrouzezahrai, and Toshiya Hachisuka. 2022. Single-pass stratified importance resampling. In *Computer Graphics Forum*, Vol. 41. Wiley Online Library, 41–49.

Sabrina Dammertz and Alexander Keller. 2006. Image synthesis by rank-1 lattices. In *Monte Carlo and Quasi-Monte Carlo Methods 2006*. Springer, 217–236.

Bastien Doignies, David Coeurjolly, Nicolas Bonneel, Julie Digne, Jean-Claude Iehl, and Victor Ostromoukhov. 2024. Differentiable Owen Scrambling. *ACM Trans. Graph.* 43, 6, Article 255 (Nov. 2024), 12 pages. https://doi.org/10.1145/3687764

Eric Heitz and Laurent Belcour. 2019. Distributing Monte Carlo Errors as a Blue Noise in Screen Space by Permuting Pixel Seeds Between Frames. *Computer Graphics Forum* (2019). https://doi.org/10.1111/cgf.13778

Eric Heitz, Laurent Belcour, V. Ostromoukhov, David Coeurjolly, and Jean-Claude Iehl. 2019. A low-discrepancy sampler that distributes monte carlo errors as a blue noise in screen space. In *ACM SIGGRAPH 2019 Talks* (Los Angeles, California) *(SIGGRAPH '19)*. Association for Computing Machinery, New York, NY, USA, Article 68, 2 pages. https://doi.org/10.1145/3306307.3328191

Simon Kallweit, Petrik Clarberg, Craig Kolb, Tom'aš 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 https://github.com/NVIDIAGameWorks/Falcor.

Alexander Keller. 1996. Quasi-Monte Carlo methods in computer graphics: The global illumination problem. *LECTURES IN APPLIED MATHEMATICS-AMERICAN MATHEMATICAL SOCIETY* 32 (1996), 455–470.

Dirk P Kroese, Thomas Taimre, and Zdravko I Botev. 2013. *Handbook of monte carlo methods.* John Wiley & Sons.

Daqi Lin, Markus Kettunen, Benedikt Bitterli, Jacopo Pantaleoni, Cem Yuksel, and Chris Wyman. 2022. Generalized resampled importance sampling: foundations of ReSTIR. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 1–23.

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. https://doi.org/10.1145/3478513.3480499

Joey Litalien, Miloš Hašan, Fujun Luan, Krishna Mullia, and Iliyan Georgiev. 2024. Neural Product Importance Sampling via Warp Composition. *ACM SIGGRAPH Asia 2024 Conference Proceedings.* https://doi.org/10.1145/3680528.3687566

Thomas Müller, Brian Mcwilliams, Fabrice Rousselle, Markus Gross, and Jan Novák. 2019. Neural Importance Sampling. *ACM Trans. Graph.* 38, 5, Article 145 (Oct. 2019), 19 pages. https://doi.org/10.1145/3341156

Baptiste Nicolet, Fabrice Rousselle, Jan Novak, Alexander Keller, Wenzel Jakob, and Thomas Müller. 2023. Recursive Control Variates for Inverse Rendering. *ACM Trans. Graph.* 42, 4, Article 62 (July 2023), 13 pages. https://doi.org/10.1145/3592139

Yaobin Ouyang, Shiqiu Liu, Markus Kettunen, Matt Pharr, and Jacopo Pantaleoni. 2021. ReSTIR GI: Path resampling for real-time path tracing. In *Computer Graphics Forum*, Vol. 40. Wiley Online Library, 17–29.

Corentin Salaün, Iliyan Georgiev, Hans-Peter Seidel, and Gurprit Singh. 2022a. Scalable Multi-Class Sampling via Filtered Sliced Optimal Transport. 41, 6, Article 261 (nov 2022), 14 pages. https://doi.org/10.1145/3550454.3555484

Corentin Salaün, Adrien Gruson, Binh-Son Hua, Toshiya Hachisuka, and Gurprit Singh. 2022b. Regression-based Monte Carlo integration. *ACM Trans. Graph.* 41, 4, Article 79 (July 2022), 14 pages. https://doi.org/10.1145/3528223.3530095

Gurprit Singh, Cengiz Öztireli, Abdalla G.M. Ahmed, David Coeurjolly, Kartic Subr, Oliver Deussen, Victor Ostromoukhov, Ravi Ramamoorthi, and Wojciech Jarosz. 2019. Analysis of Sample Correlations for Monte Carlo Rendering. *Computer Graphics Forum* 38, 2 (2019), 473–491. https://doi.org/10.1111/cgf.13653 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.13653

I.M Sobol'. 1967. On the distribution of points in a cube and the approximate evaluation of integrals. *U. S. S. R. Comput. Math. and Math. Phys.* 7, 4 (1967), 86–112. https://doi.org/10.1016/0041-5553(67)90144-9

Kartic Subr, Derek Nowrouzezahrai, Wojciech Jarosz, Jan Kautz, and Kenny Mitchell. 2014. Error analysis of estimators that use combinations of stochastic sampling strategies for direct illumination. In *Computer Graphics Forum*, Vol. 33. Wiley Online Library, 93–102.

Justin F Talbot. 2005. *Importance resampling for global illumination.* Brigham Young University.

Petr Vévoda, Ivo Kondapaneni, and Jaroslav Křivánek. 2018. Bayesian online regression for adaptive direct illumination sampling. *ACM Trans. Graph.* 37, 4, Article 125 (July 2018), 12 pages. https://doi.org/10.1145/3197517.3201340

Yu-Chen Wang, Chris Wyman, Lifan Wu, and Shuang Zhao. 2023. Amortizing Samples in Physics-Based Inverse Rendering Using ReSTIR. *ACM Trans. Graph.* 42, 6, Article 214 (Dec. 2023), 17 pages. https://doi.org/10.1145/3618331

Mirco Werner, Vincent Schüßler, and Carsten Dachsbacher. 2024. ReSTIR Subsurface Scattering for Real-Time Path Tracing. *Proceedings of the ACM on Computer Graphics and Interactive Techniques (Proceedings of High Performance Graphics)* 7, 3 (2024). https://doi.org/10.1145/3675372

Cheng Zhang, Zhao Dong, Michael Doggett, and Shuang Zhao. 2021. Antithetic sampling for Monte Carlo differentiable rendering. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–12.

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. https://doi.org/10.1145/3658210

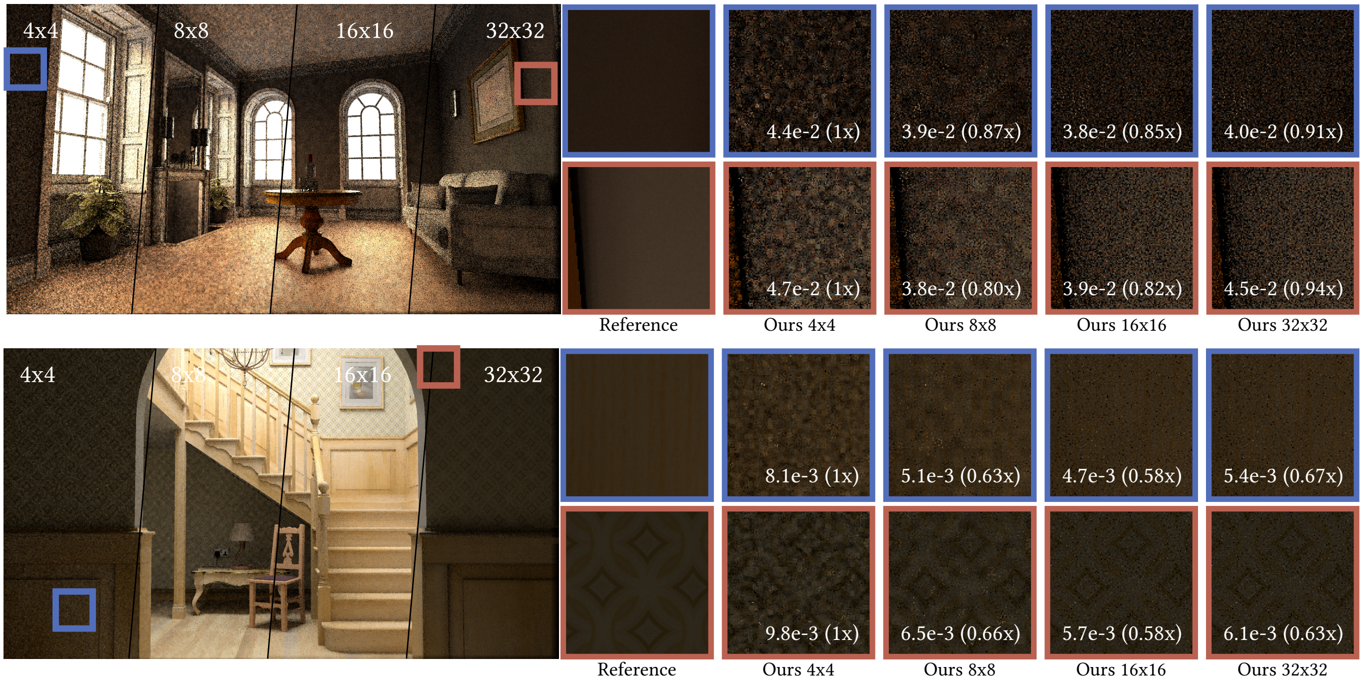


Fig. 9. **Comparison of the block size evolution.** Comparison of rendering results using our method across multiple block sizes, ranging from 4×4 to 32×32. Smaller block sizes produce visible low-frequency artifacts and result in lower RelMSE, as indicated by the reported values. Conversely, excessively large block sizes tend to reduce pixel similarity and diminish path reuse quality. In general, a block size of 16×16 offers the best trade-off.

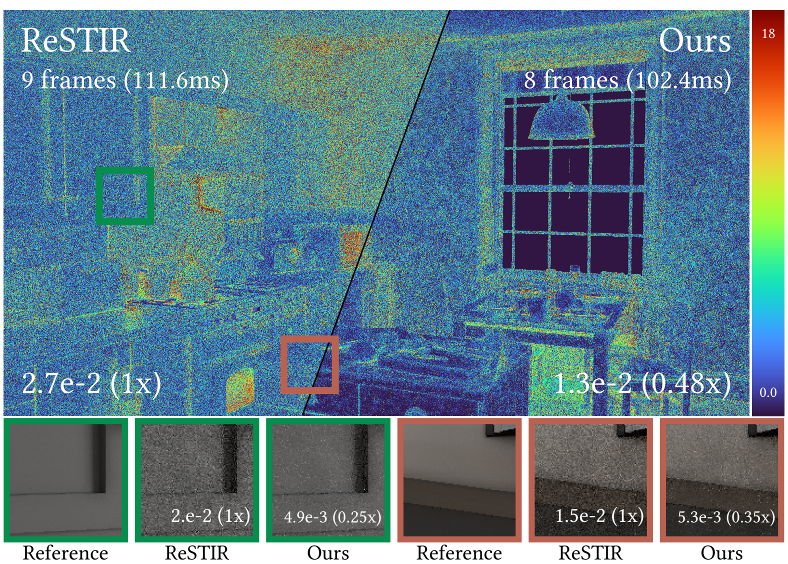


Fig. 10. **Comparison with temporal accumulation.** Equal time comparison of rendering using frame accumulation instead of temporal candidate sampling over 8 and 9 frames. The RelMSE error map and three close-up views are shown for both methods, along with the RelMSE values and their relative differences to ReSTIR.

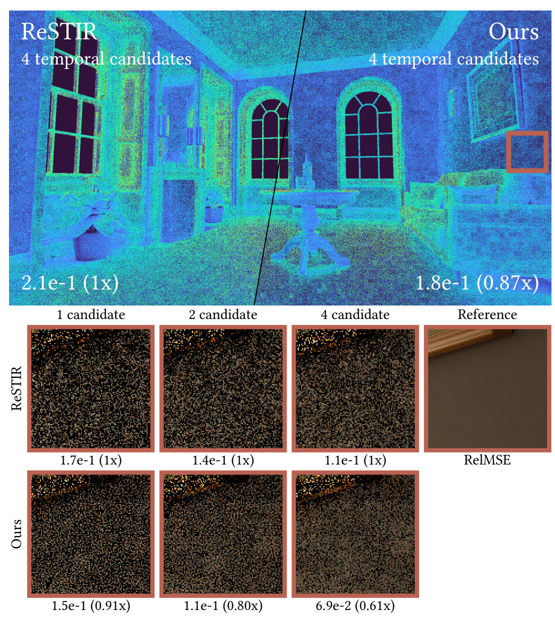


Fig. 11. **Comparison with multiple temporal reservoir** Rendering comparison across varying temporal reservoir sample counts. The RelMSE map shows results with 4 temporal reservoirs. Close-ups compare outputs with 1, 2, and 4 candidates. By employing our histogram-stratified sampling method for temporal candidates, we achieve error reduction over ReSTIR, independent of the temporal reservoir count.

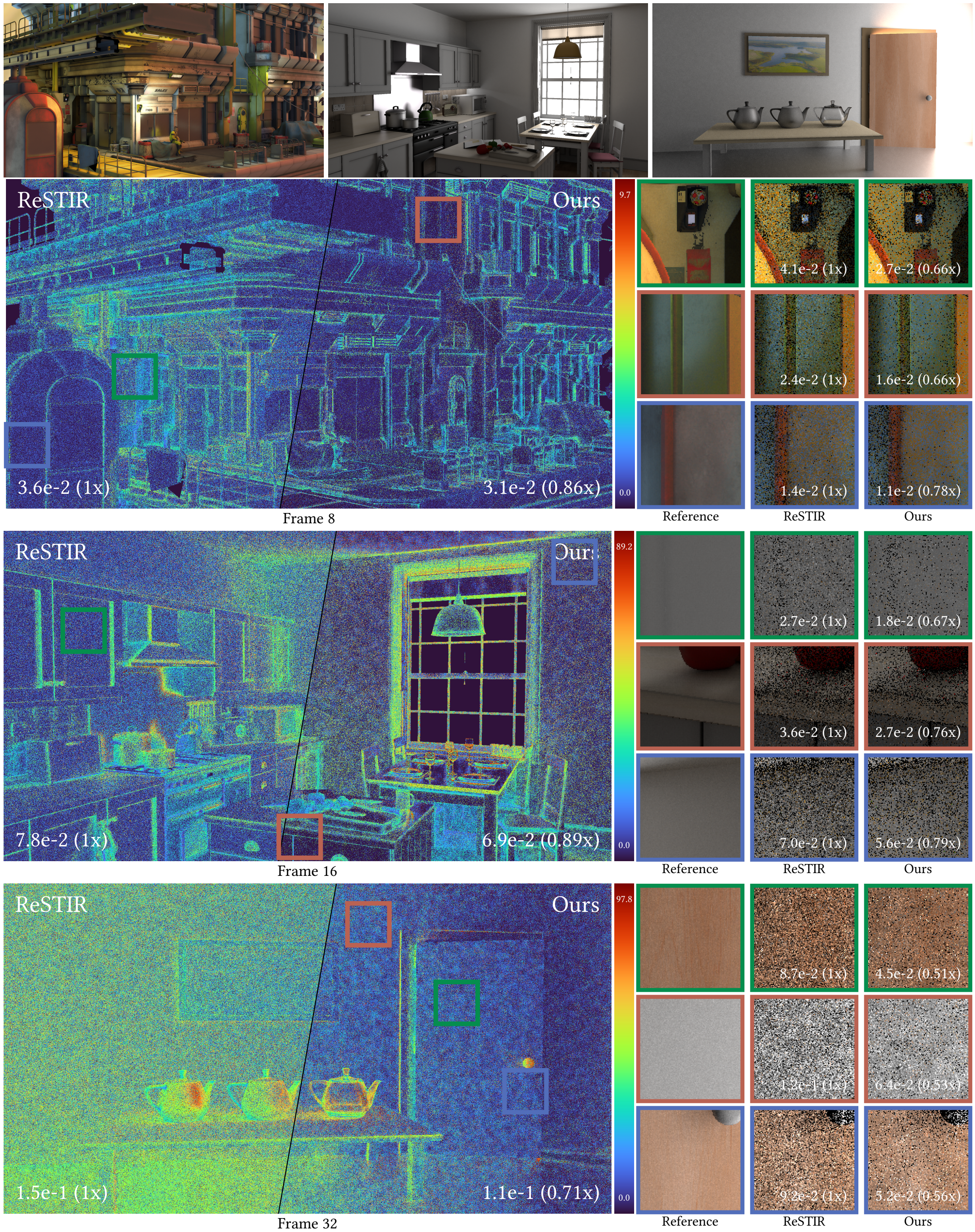


Fig. 12. **Comparison using temporal reservoir.** Visual and RelMSE comparisons are shown with equal sample counts for three different scenes, rendered as the 8th, 16th, and 32nd respectively frames of static scene. The close-up report the RelMSE for the region and its relative difference to ReSTIR. Our method shows reduced noise levels for both direct illumination using environment maps (top and middle) and for indirectly illuminated scenes (bottom).