RGB ReSTIR: Decorrelating Spatiotemporal Importance Resampling with Per-Channel Reservoirs

Markku Mäkitalo[a], Saku Haikio, Julius Ikkala[b], Alessandro Foi[c] and Pekka Jääskeläinen[d]
Tampere University, Finland
{*markku.makitalo,saku.haikio,julius.ikkala,alessandro.foi,pekka.jaaskelainen*}*@tuni.fi*

Keywords: Computer Graphics, Ray Tracing, Path Tracing, Photorealistic Rendering, ReSTIR.

Abstract: ReSTIR is a family of state-of-the-art spatiotemporal resampling algorithms utilized for improving the efficiency of photorealistic rendering. In particular, ReSTIR PT is commonly used for accelerating path tracing, a method that enables a high level of photorealism through a Monte Carlo based approximation of the global illumination. However, ReSTIR PT produces correlated samples due to its prominent reuse of spatially and temporally close pixels in the sample reservoirs, which typically manifests as visible color noise. ReSTIR-based algorithms typically only use luminance data for estimating their resampling target function, which means that in general, they cannot converge to a fully decorrelated image even with large reservoir sizes. In this paper, we present RGB ReSTIR, a multichannel variant of ReSTIR PT that maintains separate reservoirs and estimates separate target functions for each color channel. This approach allows the resampling to produce images with significantly less color noise than ReSTIR PT, especially for scenes with complex colored lighting. We demonstrate that RGB ReSTIR is able to converge towards a fully decorrelated image as the maximum confidence is increased (i.e., with longer temporal reservoir history), typically reaching an order of magnitude lower average sample autocovariance than ReSTIR PT.

1 INTRODUCTION

ReSTIR (Bitterli et al., 2020; Ouyang et al., 2021; Lin et al., 2022; Wyman et al., 2023) is a family of state-of-the-art spatiotemporal resampling algorithms utilized for improving the efficiency of photorealistic rendering. In particular, ReSTIR PT (Lin et al., 2022; Wyman et al., 2023) is the state-of-the-art path-traced resampler suitable for Monte Carlo rendering, which enables a high level of photorealism through approximating the global illumination for virtual scenes with potentially complex indirect lighting.

ReSTIR-based algorithms typically only use grayscale data for estimating the resampling target function, which means that while the pixel brightnesses are importance sampled, the chrominance data is ignored (see, e.g., (Lin et al., 2022), Section 10). This means that such approaches cannot in general converge to a fully decorrelated image even with large reservoir sizes. The remaining correlation manifests as visible artifacts, such as color noise (Kettunen et al., 2023).

While ReSTIR (Bitterli et al., 2020) can be made to converge by rendering independent frames with only spatial reuse in offline scenarios with enough sample budget (see (Lin et al., 2022), Section 6.5), temporal reservoirs can provide much more potentially reusable samples. However, the problem with temporal reuse in ReSTIR PT is that it causes temporal correlation in the samples, and thus slows down the convergence in a progressive renderer that does temporal accumulation (Lin et al., 2022).

In this paper, we introduce RGB ReSTIR, a variant of ReSTIR PT that maintains separate reservoirs and estimates separate target functions for each color channel. We show that this approach allows the resampling to produce images with significantly less color noise than ReSTIR PT, especially for scenes with complex colored lighting. In particular, we demonstrate that RGB ReSTIR continues to converge towards a fully decorrelated image as the maximum confidence is increased (i.e., with longer temporal reservoir history), as opposed to ReSTIR PT where the amount of correlation effectively stops decreasing after a certain point. Our main contributions are pro-

[a] https://orcid.org/0000-0001-8164-0031
[b] https://orcid.org/0000-0002-5373-3190
[c] https://orcid.org/0000-0001-8228-3187
[d] https://orcid.org/0000-0001-5707-8544

viding an algorithmic formulation of RGB ReSTIR with per-channel reservoirs[1]; and demonstrating with our proof-of-concept implementation that RGB ReSTIR converges temporally, typically yielding an order of magnitude lower correlation than ReSTIR PT.

Overall, this paper highlights the typically neglected correlation problems that arise from using a scalar target function (e.g., luminance) for multi-channel data (e.g., RGB) in ReSTIR-based algorithms, and presents a solution with promising results, although more research would be required for competitive real-time performance. While denoising is typically employed in conjunction with ReSTIR for visually more pleasing results, it introduces bias, manifesting as blur and loss of high-frequency details. This can be detrimental in many cases, for example when simulating medical imaging and other physics-based imaging systems (Lyu et al., 2022; Denisova et al., 2025). Hence, this paper focuses on addressing issues present in unbiased methods for accelerating physically-based path tracing.

2 BACKGROUND

2.1 Resampled Importance Sampling

If analytically solving an integral $\int_\Omega f(x)\,\mathrm{d}x$ for a complicated function $f(x)$ is intractable, we need to resort to approximating it numerically, for example via Monte Carlo sampling:

$$F_n = \frac{1}{n}\sum_{i=1}^{n} f(X_i) \underset{n\to\infty}{\to} \int_\Omega f(x)\,\mathrm{d}x\,, \tag{1}$$

where we evaluate $f(x)$ at n randomly selected values $X_i \in \Omega$. In practice, it is often more efficient not to use uniformly distributed random samples, but to pick them from a distribution that more closely matches the integrand $f(x)$. This technique is known as importance sampling: $F_n = \frac{1}{n}\sum_{i=1}^{n} \frac{f(X_i)}{p(X_i)}$, where $p(x)$ is a probability distribution function (PDF), and X_i a random sample with $X_i \sim p(x)$. Additionally we assume that $p(x) > 0$ when $|f(x)| > 0$.

In photorealistic rendering, we aim to solve the analytically intractable and recursive *Rendering Equation* (Kajiya, 1986)

$$L_o(a,\omega_o) = L_e(a,\omega_o) + \int_\Omega L_i(a,\omega_i)\rho(a,\omega_i,\omega_o)(\omega_i\cdot n)\,\mathrm{d}\omega_i\,, \tag{2}$$

where L_o is the outgoing radiance from surface point a towards direction ω_o, L_e the radiance emitted by the surface at a towards direction ω_o, L_i the incoming radiance to a from direction ω_i, ρ the bidirectional scattering distribution function (BSDF) for the material, and n the surface normal at a. In other words, the integrand in Eq. (2) is a product of multiple functions. In what follows, we denote this integrand as $f(x)$. Sampling proportionally to this product is in general infeasible, since it would require us to already know the value of the (unknown) integral. Hence, renderers typically pick samples proportionally to the individual factors of the product. However, we can still compute multiple samples following different sampling strategies (e.g., proportional to the different factors), and combine them through *Multiple Importance Sampling* (MIS) (Veach and Guibas, 1995):

$$F_n = \sum_{i=1}^{n} \frac{1}{n_i} \sum_{j=1}^{n_i} w_i(X_{i,j}) \frac{f(X_{i,j})}{p_i(X_{i,j})}\,, \tag{3}$$

where n is the number of sampling distributions, n_i is the number of samples from distribution p_i, and w_i is a weighting function to ensure the sampling remains unbiased. A typical weighting function is the *balance heuristic*:

$$w_i(x) = \frac{n_i p_i(x)}{\sum_j n_j p_j(x)}\,, \tag{4}$$

where n_i is the number of samples from strategy i, and p_i is the PDF of strategy i.

The more closely p_i matches f, the better approximation we can compute with a limited sample budget. However, it is difficult to predict which paths of light lead to high values of f without just evaluating f, making it also difficult to choose a good sampling distribution p_i. Moreover, f is in practice a function with N channels (typically red, green, and blue). Thus, in order to minimize the needed computation, and to be able to use a single distribution instead of a multi-channel function, we can approximate $f(x)$ via a *Resampled Importance Sampling* (RIS) (Talbot, 2005) target function $\hat{p}$. The RIS algorithm works by generating a set of random candidate samples $X_i \in \Omega$, and then randomly selecting one of them proportionally to the resampling weights w_i. This produces a sample X, which is drawn from a PDF that is approximately proportional to the target function $\hat{p}$ (Wyman et al., 2023). Substituting $\hat{p}$ into Eq. (3) yields the estimate

$$\hat{F}_n = \sum_{i=1}^{n} \frac{1}{n_i} \sum_{j=1}^{n_i} w_i(X_{i,j}) \frac{\hat{p}(X_{i,j})}{p_i(X_{i,j})}\,, \tag{5}$$

where p_i is the initial sample distribution for strategy i, X_i is an initial sample from p_i, and $w_i(X_i)$ is the MIS weight for X_i. For ReSTIR, a large amount of

[1] This paper expands upon the second Author's master's thesis (Haikio, 2025), where the concept of RGB ReSTIR was first introduced.

initial samples (also called *canonical samples* in the ReSTIR literature) is often impractically slow, so only one initial sample is picked per sample distribution: $n_i = 1$ for all i. Then, Eq. (5) can be reduced into

$$\hat{P}_n = \sum_{i=1}^{n} w_i(X_i) \frac{\hat{p}(X_i)}{p_i(X_i)}. \tag{6}$$

This allows us to importance sample the target function $\hat{p}(x)$ according to $\hat{P}_n$:

$$p_{\hat{p}}(x|\hat{P}_n) = \frac{\hat{p}(x)}{\hat{P}_n}. \tag{7}$$

Finally, we can importance sample the original function f with the devised better sampling strategy. Using Eqs. (6) and (7), we can write the final RIS estimate as

$$E_{\mathrm{RIS}}\langle f(X)\rangle = \frac{f(X)}{p_{\hat{p}}(X|\hat{P}_n)} = \frac{f(X)}{\hat{p}(X)} \sum_{i=1}^{n} w_i(X_i) \frac{\hat{p}(X_i)}{p_i(X_i)}, \tag{8}$$

where X is the sample chosen from the set of candidate samples X_i. The sum $\sum_{i=1}^{n} w_i(X_i) \frac{\hat{p}(X_i)}{p_i(X_i)}$ describes the MIS integral estimation of the target function $\hat{p}$, which is created by importance sampling the initial distribution p_i. Note that if $n = 1$, Eq. (8) reduces to a regular importance sampled value $f(X)/p(X)$.

In practice, in order to avoid having to store all candidate samples in memory, ReSTIR combines RIS with *weighted reservoir sampling* (Chao, 1982; Bitterli et al., 2020), which is a method for selecting one or more elements from a set or data stream $\{X_1, X_2, X_3, \ldots, X_n\}$, where the length n might be extremely large or unknown. The reservoir is updated on every iteration of the initial sample loop; updating replaces the current sample with a new sample with probability $p_r = \frac{s(X_i)}{\sum_{j=1}^{n} s(X_j)}$, where $s(X_i)$ is the weight of the current sample, and $\sum_{j=1}^{n} s(X_j)$ is the sum of all previous sample weights.

2.2 Spatiotemporal Reuse

When rendering images or video, there is typically significant correlation between the neighboring pixels, both spatially and temporally. Thus, the RIS estimator in Eq. (8) can be extended to multiple importance sample from different integration domains, specifically from the temporal frame distribution p_t and the spatial neighbor pixel distribution p_s:

$$E_{\mathrm{ReSTIR}}\langle f(X)\rangle = \frac{f(X)}{\hat{p}(X)} \cdot \left(w_{\hat{p}}(X_{\hat{p}}) \frac{\hat{p}(X_{\hat{p}})}{p_{\hat{p}}(X_{\hat{p}})} + w_t(X_t) \frac{\hat{p}(X_t)}{p_{\mathrm{t}}(X_t)} + w_s(X_s) \frac{\hat{p}(X_s)}{p_{\mathrm{s}}(X_s)} \right). \tag{9}$$

However, samples from distributions $p_{\hat{p}}$, p_t and p_s are chosen via RIS, which means that their real distributions are unknown. Thus, instead of using these functions for the MIS weighting, it is possible to use the target functions of these different domains. This is done by weighting $w_{\hat{p}}$, w_t and w_s via the target functions $\hat{p}$, $\hat{p}_t$ and $\hat{p}_s$. The commonly used weighting function m_i for calculating w_i is

$$m_i(x) = \frac{c_i \hat{p}_i(x)}{\sum_{j=1}^{n} c_j \hat{p}_j(x)}, \tag{10}$$

where $\hat{p}_i \in \{\hat{p}_{\hat{p}}, \hat{p}_t, \hat{p}_s\}$ (Wyman et al., 2023); then,

$$w_i(x) = m_i(x) \frac{\hat{p}_i(x)}{p_i(x)}. \tag{11}$$

The values c_j in Eq. (10) are *confidence weights* for weighting each function according to their contribution for estimating the target function $\hat{p}$. These confidence weights are the sum of the previous confidence weights when estimating the target function with RIS, and new initial samples are given weight 1. In practice, the summed confidence weights are further capped not to exceed a *maximum confidence* M_c:

$$c_j \leqslant M_c \;\forall j. \tag{12}$$

This cap helps in controlling the balance between noise and correlation (Wyman et al., 2023).

2.3 ReSTIR PT

In path tracing, we trace paths of light that potentially intersect with the scene at points $\{x_1, \ldots, x_N\}$ along the path from the virtual camera (located at x_0) to a light source. Different paths may share intersection points, prompting the idea of reusing their data between pixels. However, unless two pixels are identical, their integration domains will be different, preventing naïve reuse of the already computed neighboring target functions. Thus, in order to reuse paths from different domains, ReSTIR PT (Lin et al., 2022) employs *shift maps* to map the sampled paths from one domain to another. The commonly used shift maps are random replay shift (which replays the path in a different domain using its original random seed), reconnection shift (which reconnects two paths through the secondary intersection point x_2), and hybrid shift (combining the previous two, so that the reconnection can also happen through a later intersection point) (Lin et al., 2022). In practice, the shift maps T are used for modifying the weighting function m_i in Eq. (10) as follows:

$$m_i(x) = \frac{c_i \hat{p}_i(T_i(x)) |T_i'(x)|}{\sum_{j=1}^{n} c_j \hat{p}_j(T_j(x)) |T_j'(x)|}, \tag{13}$$

where $|T_i^{'}(x)|$ is the Jacobian determinant of shift map T_i. In summary, Algorithm 1 presents in pseudocode how RIS operates in ReSTIR PT. For a more comprehensive overview of RIS and ReSTIR PT, we refer the reader to (Wyman et al., 2023); our algorithmic notation follows their notational style for convenience.

Algorithm 1: RIS in ReSTIR PT.

```
1  class Reservoir
2      X ← 0
3      w_sum ← 0
4      function update(X_i, w_i)
5          w_sum ← w_sum + w_i
6          if random_number
             u ∈ [0,1] < w_i/w_sum then
7            X ← X_i
8  function resampled_importance_sampling()
9      Reservoir r
10     for i ← 1 to n do
11         generate X_i ~ p_i
12         Y_i ← T_i(X_i)
13         w_i ← m_i(Y_i) p̂(Y_i) |T_i'(X_i)| / p_i(Y_i)
14         r.update(Y_i, w_i)
15     return f(r.X)/p̂(r.X) · r.w_sum
```

3 RGB ReSTIR

The key idea in this paper is to perform RIS estimation for every channel separately, whether it is for three RGB channels, or more generally for N channels, e.g., for spectral rendering. Hence, it is no longer necessary to approximate an N-channel function f by a single-channel target function $\hat{p}$ (as was done in Eq. (5)), so we can replace $\hat{p}$ in the RIS estimator with the original f, simplifying Eq. (8) into

$$E_{\mathrm{RIS}}\langle f(X)\rangle = \sum_{i=1}^{n} w_i(X_i)\frac{f(X_i)}{p_i(X_i)}. \qquad (14)$$

Nevertheless, we must perform this sampling process for each of the N channels. For simplicity, from now on we consider the case $N = 3$ with RGB channels (i.e., RGB ReSTIR), but it is straightforward to generalize our approach for any N. However, it is evident that as N grows, so does the amount of computation.

For RGB ReSTIR, we substitute f in turn with the red, blue, and green color channel functions f_r, f_g, and f_b, respectively. The candidate probability function p_i is the same for all three estimators. Algorithm 2 summarizes this process, showing how we no longer resample candidates, since the target function is the final function; and how we repeat the calculations for each channel. The balance heuristic (Eq. (4)) is used for computing the weights w_{r_i}, w_{g_i}, and w_{b_i}.

Algorithm 2: Per-channel sampling in RGB ReSTIR, without spatiotemporal reuse.

Input : color function f
number of initial samples n
Output: integral estimation for function $f(x)$

```
1  w_sum_r ← 0   // MIS est. for red f_r
2  w_sum_g ← 0   // MIS est. for green f_g
3  w_sum_b ← 0   // MIS est. for blue f_b
4  for i ← 1 to n do
5      generate X_i ~ p_i
6      // calculate new weight for
          every color channel
7      w_{r_i} ← w_{r_i}(X_i) f_r(X_i) / p_i(X_i)
8      w_{g_i} ← w_{g_i}(X_i) f_g(X_i) / p_i(X_i)
9      w_{b_i} ← w_{b_i}(X_i) f_b(X_i) / p_i(X_i)
10     // add weight to weight sum
11     w_sum_r ← w_sum_r + w_{r_i}
12     w_sum_g ← w_sum_g + w_{g_i}
13     w_sum_b ← w_sum_b + w_{b_i}
14 return w_sumr, w_sumg, w_sumb
```

However, Algorithm 2 does not take temporal and spatial reuse into account. Since temporal and spatial candidates can be from different domains, we still need resampling and shift mapping, as in ReSTIR PT. However, unlike ReSTIR PT, which uses a single reservoir, RGB ReSTIR needs three reservoirs; each color channel maintains its own reservoir to hold the candidate and MIS weight values. Algorithm 3 shows how we modify Algorithm 2 to incorporate the reservoir sampling stage, and to include the shift mapping from Algorithm 1. The weights m_i are computed as in Eq. (13).

4 IMPLEMENTATION

We implement a proof-of-concept version of RGB ReSTIR (as presented in Algorithm 3) on top of Tauray (Ikkala et al., 2022), an open-source Vulkan-based rendering framework that supports both efficient path tracing and ReSTIR PT, with hardware acceleration.

The Trowbridge–Reitz/GGX material model is used for shading the specular and transmission components of the lighting; the diffuse component is as-

Algorithm 3: RIS in RGB ReSTIR. Reservoir is defined as in Algorithm 1.

function resampled_importance_sampling()
  Reservoir r_{red}
  Reservoir r_{green}
  Reservoir r_{blue}
  for $i \leftarrow 1$ **to** n **do**
    generate $X_i \sim p_i$
    $Y_i \leftarrow T_i(X_i)$
    $\text{w}_{r_i} \leftarrow m_i(Y_i) f_r(Y_i) |T_i^{'}(X_i)| / p_i(Y_i)$
    $\text{w}_{g_i} \leftarrow m_i(Y_i) f_g(Y_i) |T_i^{'}(X_i)| / p_i(Y_i)$
    $\text{w}_{b_i} \leftarrow m_i(Y_i) f_b(Y_i) |T_i^{'}(X_i)| / p_i(Y_i)$
    $r_{\text{red}}.update(X_i, \text{w}_{r_i})$
    $r_{\text{green}}.update(X_i, \text{w}_{g_i})$
    $r_{\text{blue}}.update(X_i, \text{w}_{b_i})$
  return $r_{red}.w_{sum}, r_{green}.w_{sum}, r_{blue}.w_{sum}$

sumed to be Lambertian. The combined BSDF is importance sampled on every bounce, and a next event estimation (NEE) ray to a random light source is also traced on every bounce.

The core calculations of RGB ReSTIR are done in four shader stages: canonical, temporal, spatial trace, and spatial gather. The canonical stage creates the initial (i.e., canonical) RIS samples, and path traces the BSDF and NEE samples for the RGB RIS. Then, the temporal stage updates the temporal reservoirs in order to reuse the corresponding pixels on the previous frame, tracking surfaces using motion vectors. The spatial trace stage searches for good spatial neighbors; and finally, the spatial gather stage performs spatial reuse according to the found neighbors and the spatial reservoir of the current pixel, and writes the final pixel color to the G-buffer.

5 EXPERIMENTS

All experiments are done with the Sponza, Bistro, and Victorian House scenes, modified to provide a varying set of lighting conditions: Sponza features strong indirect colored lighting via spotlights pointed at the curtains, Bistro presents strong colored direct lighting, and Victorian House is mostly lit through indirect low-intensity white light.

We evaluate the correlation (describing the amount of color noise) in images produced by RGB ReSTIR and ReSTIR PT, as a function of the temporal reservoir size (i.e., maximum confidence). In more detail, the convergence towards a fully decorrelated image is measured through the sample autocovariance between two consecutive frames $M_c - 1$ and M_c (analogously to how (Sawhney et al., 2024) quantify spatial correlation), after running for as many frames as indicated by the set maximum confidence value M_c (as defined in Eq. (12)); this allows the temporal reservoir history to be fully populated for maximal reuse. The measurements are done for $M_c \in \{2, 4, 8, 16, 32, \ldots, 16384, 32768\}$; the upper limit $M_c = 32768$ is imposed by the ReSTIR PT implementation in Tauray. For each M_c, we perform 1000 independent runs with a different random seed. Sample autocovariance across these 1000 runs is computed for each pixel, and then averaged over all pixels and over all three color channels.

In order to more clearly isolate the effect of varying maximum confidence on the correlation, only temporal reservoirs are used; in other words, spatial resampling is disabled. Each image is rendered at a 640×360 resolution, due to the vast amount of frames needed to be rendered to produce 1000 independent data points for all selected M_c values. Random replay shift was chosen as the shift map for these measurements, as it gave the most consistent results across a wide range of maximum confidence values.

We also evaluate the image quality for the temporally converged frames ($M_c = 32768$) via PSNR and SSIM, on 1280×720 resolution images. As Tauray renders the frames with a high dynamic range, the metrics are computed from Reinhard-tonemapped RGB values in linear space ($C_{\text{out}} = C_{\text{in}}/(C_{\text{in}} + 1)$ for each color channel); RGB values are used instead of luminance values, as we focus on examining the effect of correlated color noise. The results are reported as an average of 100 consecutive frames.

6 RESULTS

We illustrate the temporal convergence results in terms of reduced correlation (with $M_c \in \{2, 4, 8, 16, 32, \ldots, 16384, 32768\}$), and we also present visual and objective quality results for the converged frames (with $M_c = 32768$). The exposure in all shown example frames has been increased for a better visualization.

Figures 1–3 show the sample autocovariance as a function of maximum confidence, for Sponza, Bistro, and Victorian House, respectively. Note that the axes are logarithmic (`log2` for the x-axis, with M_c increasing in powers of two; and `log10` for the y-axis). RGB ReSTIR provides a lower autocovariance in all cases, corresponding to the reduction of correlated noise. For Sponza and Bistro, the difference to ReSTIR PT

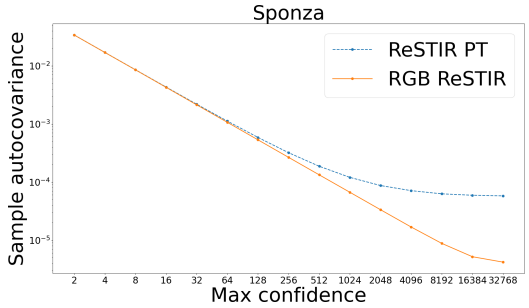

Figure 1: Sample autocovariance (average of 1000 runs) as a function of maximum confidence, for Sponza.

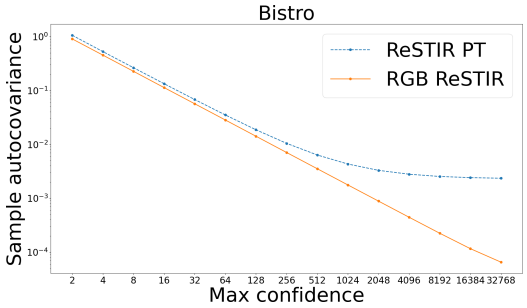

Figure 2: Sample autocovariance (average of 1000 runs) as a function of maximum confidence, for Bistro.

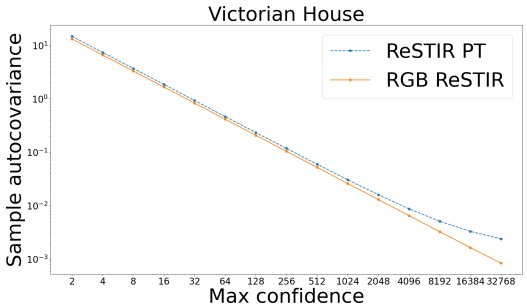

Figure 3: Sample autocovariance (average of 1000 runs) as a function of maximum confidence, for Victorian House.

begins to grow significantly at around $M_c = 128$. As M_c increases further, the amount of correlation for ReSTIR PT eventually stops decreasing, whereas RGB ReSTIR continues to reduce the correlation, reaching 1–2 orders of magnitude lower autocovariance. For example at $M_c = 32768$, for Sponza, the autocovariance of RGB ReSTIR is $4.2 \cdot 10^{-6}$, and that of ReSTIR PT is $5.8 \cdot 10^{-5}$. For Bistro, RGB ReSTIR reaches $6.5 \cdot 10^{-5}$, whereas ReSTIR PT is only able to reach $2.3 \cdot 10^{-3}$. For Victorian House, the asymptotic difference is less visible, with ReSTIR PT not yet fully plateauing at $M_c = 32768$. Nevertheless, RGB ReSTIR is still improving at a notably more rapid pace, yielding $8.4 \cdot 10^{-4}$ at $M_c = 32768$, compared to $2.4 \cdot 10^{-3}$ for ReSTIR PT.

Temporally converged example frames ($M_c = 32768$) are shown in Figures 4–5, confirming that RGB ReSTIR has successfully removed almost all color noise, whereas ReSTIR still suffers from significant color artifacts, especially in the red channel.

PSNR and SSIM results for $M_c = 32768$ are presented in Table 1. For Bistro and Victorian House, RGB ReSTIR provides significantly better results, whereas for Sponza, the improvement is more modest. However, we note that the choice of the tonemapping operator, and whether to compute PSNR and SSIM for RGB or luminance, and in linear or gamma-corrected space, has a major influence on the results, making it difficult to provide clear conclusions through PSNR and SSIM (Yeganeh and Wang, 2012; Mantiuk, 2016). Thus, we consider the sample autocovariance to be a more reliable and direct metric for assessing the amount of color noise.

7 RELATED WORK

In (Sawhney et al., 2024), the authors use a Markov Chain Monte Carlo (MCMC) approach for reducing the correlations produced by ReSTIR PT. In particular, they interleave reservoir resampling with MCMC mutations. By randomly mutating samples after temporal resampling (but not after spatial resampling), they are able to diversify the set of samples in the reservoir, thus decorrelating them to some extent. They reduce the spatial covariance over ReSTIR PT by about 30 % on average. However, their main focus is on improving the real-time performance, and they observe that increasing M_c increases *spatial correlation* with their approach, as it does with ReSTIR PT. In contrast, our convergence experiments consider the *temporal correlation* as a function of M_c, with spatial reuse disabled. Their method is not expected to converge to a decorrelated image, but it could still aid RGB ReSTIR in performing better in real-time spatiotemporal reuse scenarios with lower maximum confidence values and limited temporal accumulation.

The idea of subpath reuse is explored in (Kettunen et al., 2023) through so-called suffix subpaths, where the usual path reuse of ReSTIR PT is postponed by at least one bounce. This also reduces correlation compared to ReSTIR PT, albeit at the cost of increased noise, which they address with a photon mapping style final gather pass. Their practical approach is a proof of concept, and they do not provide objective measurements regarding the amount of correlation, or how it scales as a function of M_c. They consider both spatial and temporal reuse, but they restrict themselves to a relatively low maximum confidence of $M_c = 50$ due to the temporal reuse limitation of

Table 1: PSNR and SSIM values (average of 100 consecutive frames) for temporally converged frames ($M_c = 32768$).

Scene	PSNR (dB)		SSIM	
	ReSTIR PT	RGB ReSTIR	ReSTIR PT	RGB ReSTIR
Sponza	34.33	**34.58**	0.946	**0.953**
Bistro	26.32	**28.00**	0.742	**0.898**
Victorian House	29.16	**31.29**	0.862	**0.928**

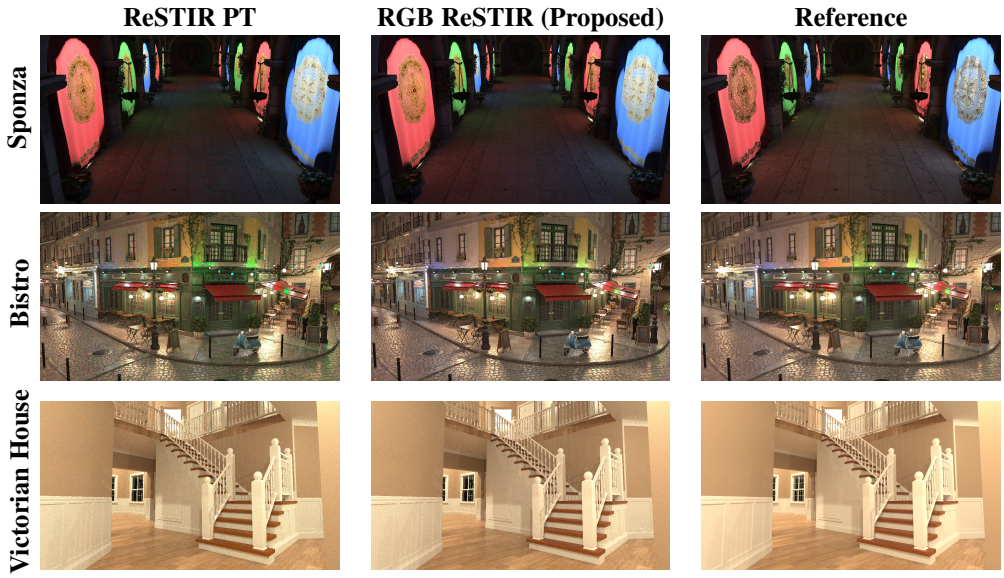

Figure 4: A comparison of the temporally converged ($M_c = 32768$) results. The reference is path traced at 262144 spp.

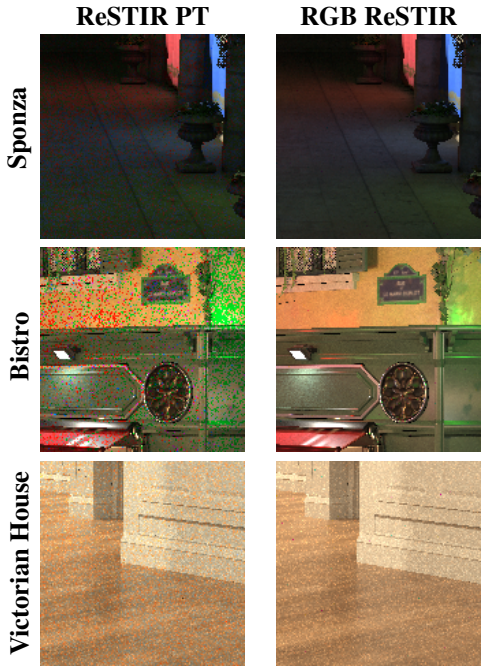

Figure 5: Zoomed-in sections of Figure 4, comparing the temporally converged ($M_c = 32768$) results.

(Lin et al., 2022) discussed in Section 1. They focus on improving sample reuse for difficult light paths and disoccluded areas, in a way resembling bidirectional path tracing. This could clearly benefit RGB ReSTIR as well, especially in dynamic cases where disocclusions complicate temporal accumulation.

In (Lin et al., 2022), the authors briefly mention the problem with the scalar (luminance) target function, speculating that "Modifications could perhaps resample between wavelengths or use hero wavelengths to help with color noise", but not expanding the discussion beyond that. Hero wavelengths (Wilkie et al., 2014) could also potentially be utilized with per-channel ReSTIR for spectral rendering.

8 FUTURE WORK

Overall, the previous works do not address the problems arising from using a scalar target function for multi-channel data, which our work centers on addressing; instead, they focus on other types of improvements that are orthogonal to our work. This is also why our comparisons are done against the original ReSTIR PT. Combining RGB ReSTIR with for

example the MCMC path mutation (Sawhney et al., 2024) and subpath reuse (Kettunen et al., 2023) would be a fruitful topic for future work, potentially reducing the correlation more rapidly in the real-time spatiotemporal sample reuse cases with a low maximum confidence. Moreover, it would be useful to analyze the effect of different shift maps more thoroughly.

While this paper focuses on unbiased rendering, it would also be interesting to evaluate the benefits of reduced correlation when combining RGB ReSTIR with real-time denoising; the latter point is also touched upon in (Kettunen et al., 2023).

In terms of runtime, RGB ReSTIR takes 1.5–2.3 times as long as ReSTIR PT (as measured on a system with an NVIDIA RTX 3080 GPU), which is reasonably good considering the naïve expectation of 3 times the runtime (1 reservoir vs. 3 reservoirs), still reaching up to 30 frames per second. However, as the real-time quality benefits of RGB ReSTIR appear relatively minor (as seen in Figures 1–3 for low M_c values), significant speedups would be required to make it competitive with ReSTIR PT in real-time scenarios. The canonical stage is approximately as fast for both algorithms; the temporal stage and spatial trace stage contribute the most to the increased runtime of RGB ReSTIR, as they manage three reservoirs instead of one. Speedups could likely be achieved, e.g., through inter-channel reuse (i.e., reusing not only spatially and temporally, but also by wavelength). However, it is not trivial to devise an efficient reuse strategy, while keeping the ReSTIR algorithm unbiased.

9 CONCLUSIONS

We propose RGB ReSTIR, a variant of ReSTIR PT with per-channel reservoirs, which is able to decorrelate the samples more efficiently than ReSTIR PT, thus yielding images with visibly less color noise. The improvement is especially significant when ReSTIR is allowed to converge through a long temporal reservoir history: RGB ReSTIR is able to continue converging towards a fully decorrelated image, whereas ReSTIR PT is unable to reduce the correlation after a certain point. In our temporal convergence experiments, RGB ReSTIR typically reaches an order of magnitude lower correlation than ReSTIR PT when the maximum confidence is set to 32768.

ACKNOWLEDGEMENTS

This work was supported by the Research Council of Finland under Grant 351623.

REFERENCES

Bitterli, B., Wyman, C., Pharr, M., Shirley, P., Lefohn, A., and Jarosz, W. (2020). Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting. *ACM Transactions on Graphics*, 39(4).

Chao, M.-T. (1982). A general purpose unequal probability sampling plan. *Biometrika*, 69(3).

Denisova, E., Francia, P., Nardi, C., and Bocchi, L. (2025). Advancements in biomedical rendering: A survey on AI-based denoising techniques. *Computers in Biology and Medicine*, 197.

Haikio, S. (2025). RGB ReSTIR: Per-Channel Sample Reuse – Method for asymptotically noise-free ReSTIR. Master's thesis, Tampere University, Finland.

Ikkala, J., Mäkitalo, M., Lauttia, T., Leria, E., and Jääskeläinen, P. (2022). Tauray: A Scalable Real-Time Open-Source Path Tracer for Stereo and Light Field Displays. In *SIGGRAPH Asia Tech. Comm.*

Kajiya, J. T. (1986). The rendering equation. In *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques*.

Kettunen, M., Lin, D., Ramamoorthi, R., Bashford-Rogers, T., and Wyman, C. (2023). Conditional resampled importance sampling and ReSTIR. In *SIGGRAPH Asia 2023 Conference Papers*.

Lin, D., Kettunen, M., Bitterli, B., Pantaleoni, J., Yuksel, C., and Wyman, C. (2022). Generalized Resampled Importance Sampling: Foundations of ReSTIR. *ACM Transactions on Graphics*, 41(4).

Lyu, Z., Goossens, T., Wandell, B. A., and Farrell, J. (2022). Validation of physics-based image systems simulation with 3-D scenes. *IEEE Sensors Journal*, 22(20).

Mantiuk, R. K. (2016). Practicalities of predicting quality of high dynamic range images and video. In *IEEE International Conference on Image Processing (ICIP)*.

Ouyang, Y., Liu, S., Kettunen, M., Pharr, M., and Pantaleoni, J. (2021). ReSTIR GI: Path resampling for real-time path tracing. *Computer Graphics Forum*, 40(8).

Sawhney, R., Lin, D., Kettunen, M., Bitterli, B., Ramamoorthi, R., Wyman, C., and Pharr, M. (2024). Decorrelating ReSTIR samplers via MCMC mutations. *ACM Transactions on Graphics*, 43(1).

Talbot, J. F. (2005). *Importance resampling for global illumination*. Brigham Young University.

Veach, E. and Guibas, L. J. (1995). Optimally combining sampling techniques for Monte Carlo rendering. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*.

Wilkie, A., Nawaz, S., Droske, M., Weidlich, A., and Hanika, J. (2014). Hero wavelength spectral sampling. *Computer Graphics Forum*, 33(4).

Wyman, C., Kettunen, M., Lin, D., Bitterli, B., Yuksel, C., Jarosz, W., and Kozlowski, P. (2023). A Gentle Introduction to ReSTIR: Path Reuse in Real-Time. In *ACM SIGGRAPH 2023 Courses*.

Yeganeh, H. and Wang, Z. (2012). Objective quality assessment of tone-mapped images. *IEEE Transactions on Image Processing*, 22(2).