Pacific Graphics (2020)
S. Lee and S. Zollmann (Editors) *Poster*

Reconstructing Monte Carlo Errors as a Blue-noise in Screen Space

Hongli Liu and Honglei Han †

School of Animation and Digital Arts, Communication University of China, China

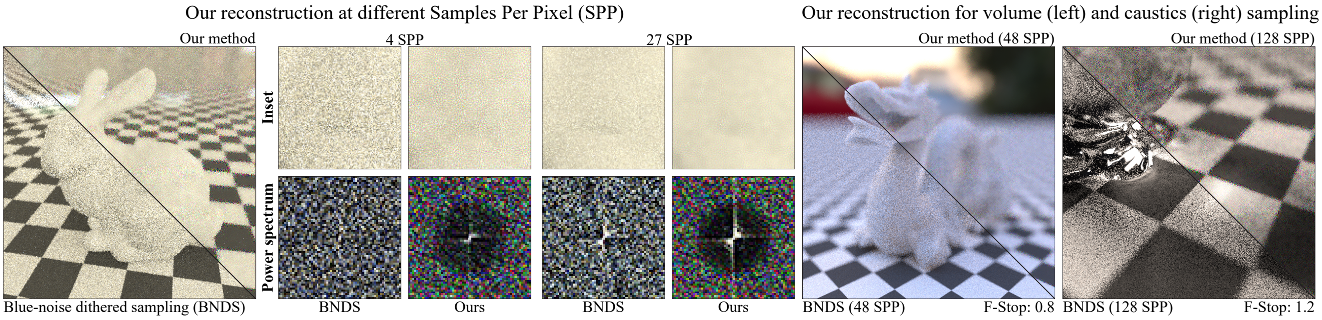

Figure 1: *Our method reconstructs the rendered images such that their errors due to Monte Carlo is a blue-noise in screen space. Compared to BNDS, it produces a more pleasing noise of the same variance, thus improving the visual fidelity of the renderings.*

Abstract

We present a novel method that reconstructs the Monte Carlo errors of renderings as a blue-noise in screen space. To this end, we conform the statistic result of per-pixel integration to a precomputed blue-noise mask. Thanks to the property of blue-noise, more visual fidelity is achieved through renderings after the reconstruction. The method has two key features. First, its realization is fast and straightforward. Second, it produces stable blue-noise-error renderings regardless of the correlation of the integrands. The preliminary results present robust blue-noise spectra with promising visual improvements in the renderings.

CCS Concepts

• ***Computing methodologies*** $\rightarrow$ ***Rendering;***

1. Introduction

Since Georgiev and Fajardo pioneered the research of blue-noise correlation of per-pixel errors in renderings [GF16], this goal has been a long-standing challenge for recent studies. Inspired by digital halftoning, they used a blue-noise mask to shift the sampling sequence (*Blue-noise Dithered Sampling*, a.k.a. BNDS), i.e. using a precomputed screen-space blue-noise as the first sample, such that the rendering errors is as well a screen-space blue-noise at 1 *sample per pixel* (SPP). Heitz et al. further introduced a sampler to address the restriction of dimensions and sample counts [HBO*19].

Nevertheless, as pointed out by Heitz and Belcour, correlation-preserving integrands are the key to the effectiveness of BNDS [HB19]. In other words, BNDS is prone to fail beyond simple scenes (e.g. smooth area lights, low-varying scenes, and for the details, please see the supplemental material). They proposed a posteriori method based on the temporal buffers which approximates the radiant distribution of the integrands and procedurally updates the scrambling seed to obtain stable blue-noise errors in more complicated scenes. Despite the computational cost, this method has ascendency over the former two methods. It works very well for real-time applications like video games with ray-tracing algorithms. Still, one major drawback is that the claim of a fixed sample count prevents it from offline rendering, where the progressive sample counts and the blue-noise errors are both expected by an artist rendering a static frame.

In this paper, we present a novel solution to respond to this challenge. We start the research directly from the statistic result of the integrations. Instead of manipulating the sampling sequences,

† Supported by the Fundamental Research Funds for the Central Universities, and the National Key R&D Program of China (2018YFB1403900)

DOI: 10.2312/pg.20201230

we conform the Monte Carlo (MC) errors to a precomputed blue-noise mask. To efficiently evaluate the expected values and variances of the pixels, we utilize the neighboring pixels based on a carefully designed weighting function. We adopt a dynamic kernel size, such that the reconstructed renderings converge identically to the ground-truth results.

2. Method

The crux of distributing the MC errors as a blue-noise lies in the estimation of the histogram of the per-pixel integrands – if one samples directly the inverse *Cumulative Distribution Functions* (iCDF) of the integrands using screen-space blue-noise, blue-noise errors are granted. However, in the context of static frame rendering or progressive rendering, the per-pixel integrands are unobservable until the frame has been rendered. As a consequence, our motivation all revolves around the errors of the rendered result.

If the expected value $\mathrm{E}(P_{i,j})$ and the variance $\mathrm{Var}(P_{i,j})$ of the integrand of a pixel $P_{i,j}$ of an s-dimensional irradiance spectrum is known in advance, we can normalize a precomputed uniform screen-space blue-noise $X \sim U[0,1]^s$, such that the reconstructed pixel $\widetilde{P}_{i,j}$ has blue-noise-featured value

$$\widetilde{P}_{i,j} = \frac{X - \mathrm{E}(X)}{\sqrt{\mathrm{Var}(X)}} \cdot \sqrt{\mathrm{Var}(P_{i,j})} + \mathrm{E}(P_{i,j}), \tag{1}$$

where $\mathrm{E}(X) = \frac{1}{2}$ and $\mathrm{Var}(X) = \frac{1}{12}$ for uniform distributions are well-known. This is because the Fourier spectrum $\mathcal{F}(\mathrm{E}(P_{i,j}))$ of the noise-free $\mathrm{E}(P_{i,j})$ is relatively flat, and the remainder contributes to the blue-noise spectrum.

To efficiently capture the expected value and variance from the neighboring pixels, we adopt a weighting function

$$f(p) = \frac{1}{k} \cdot G_\sigma(r) \cdot w(p,q) \tag{2}$$

with Edge-Avoiding weight function w proposed by Dammertz et al. [DSHL10], pixel position q, neighboring pixel position p, and distance of the two pixel r. k denotes the normalization factor

$$k = \sum G_\sigma(r) \cdot w(p,q) \tag{3}$$

and G_σ is the Gaussian kernel with standard deviation σ

$$G_\sigma(r) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot \exp\left(-\frac{r^2}{2\sigma^2}\right). \tag{4}$$

The parameter σ plays a vital role. If it is chosen statically, the renderings won't converge to the ground-truth, because the reconstruction kernel over-blurs the rendered results. To make sure that it converges simultaneously with the MC errors, we make the parameter choice similar to photon mapping. First, assuming the expected value or variance at pixel q is $y(q)$, we expand $Y(p) = y(p) \cdot w(p,q)$ into Taylor series around q

$$Y(p) = Y(q) + \sum_{n=1}^{\infty} \frac{c_n}{n!} \cdot (p-q)^n. \tag{5}$$

The filtered result of $y(q)$ is thus

$$\begin{aligned}\widetilde{y}(q) &= \frac{1}{k} \cdot \sum G_\sigma(r) \cdot Y(p) = \frac{1}{k} \cdot \sum G_\sigma(r) \cdot \left(Y(q) + \sum_{n=1}^{\infty} \frac{c_n}{n!} \cdot (p-q)^n\right) \\ &= y(q) + \frac{1}{k} \cdot \sum_{n=1}^{\infty} \frac{c_{2n}\sigma^{2n}}{(n-1)!!} = y(q) + O(\sigma^2),\end{aligned} \tag{6}$$

where $n!!$ is the double factorial of n. Due to the fact that the convergence rate of quasi-MC w.r.t. sample count N is (close to) $O\left(N^{-1}\right)$ [Caf98], given an initial σ_0, we let

$$\sigma = \frac{1}{\sqrt{N}} \cdot \sigma_0, \tag{7}$$

such that $O(\sigma) = O\left(N^{-1}\right)$, and that the reconstructed images, along with the MC results, converge to the ground-truth at the same rate.

3. Results and Conclusion

Fig. 1 presents the rendering result (using $\sigma_0 = 64$) of our method and comparison against BNDS. With the application of our method, promising blue-noise spectrum is produced, which is absent in BNDS.

Compared to priori methods [GF16, HBO*19], our method robustly achieves high visual fidelity in complicated scenes containing highly decorrelated integrands like participating medium and caustics paths, as well as allowing random sample counts and dimensionalities. It neither intervenes the sampling and integration process nor postulates any specific sampling sequence, which doesn't affect the convergence of the renderings. Furthermore, it can serve as a post-process of the renderer with a lower engineering cost compared to the existing posteriori method [HB19], provided with the effectiveness in progressive renderers and static frame rendering.

References

[Caf98] Caflisch R. E.: Monte carlo and quasi-monte carlo methods. *Acta Numerica 7* (1998), 1–49. doi:10.1017/S0962492900002804. 2

[DSHL10] Dammertz H., Sewtz D., Hanika J., Lensch H. P. A.: Edge-avoiding À-trous wavelet transform for fast global illumination filtering. In *Proceedings of the Conference on High Performance Graphics* (Goslar, DEU, 2010), HPG '10, Eurographics Association, p. 67–75. 2

[GF16] Georgiev I., Fajardo M.: Blue-noise dithered sampling. In *ACM SIGGRAPH 2016 Talks* (New York, NY, USA, 2016), SIGGRAPH '16, Association for Computing Machinery. URL: https://doi.org/10.1145/2897839.2927430, doi:10.1145/2897839.2927430. 1, 2

[HB19] Heitz E., Belcour L.: Distributing monte carlo errors as a blue noise in screen space by permuting pixel seeds between frames. *Computer Graphics Forum 38* (07 2019). doi:10.1111/cgf.13778. 1, 2

[HBO*19] Heitz E., Belcour L., Ostromoukhov V., Coeurjolly D., Iehl J.-C.: A low-discrepancy sampler that distributes monte carlo errors as a blue noise in screen space. In *ACM SIGGRAPH 2019 Talks* (New York, NY, USA, 2019), SIGGRAPH '19, Association for Computing Machinery. URL: https://doi.org/10.1145/3306307.3328191, doi:10.1145/3306307.3328191. 1, 2