Perceptual error optimization for Monte Carlo animation rendering

MIŠA KORAĆ [*], Saarland University, DFKI, Germany
CORENTIN SALAÜN [*], Max Planck Institute for Informatics, Germany
ILIYAN GEORGIEV, Adobe, UK
PASCAL GRITTMANN, Saarland University, Germany
PHILIPP SLUSALLEK, Saarland University, DFKI, Germany
KAROL MYSZKOWSKI, Max Planck Institute for Informatics, Germany
GURPRIT SINGH, Max Planck Institute for Informatics, Germany

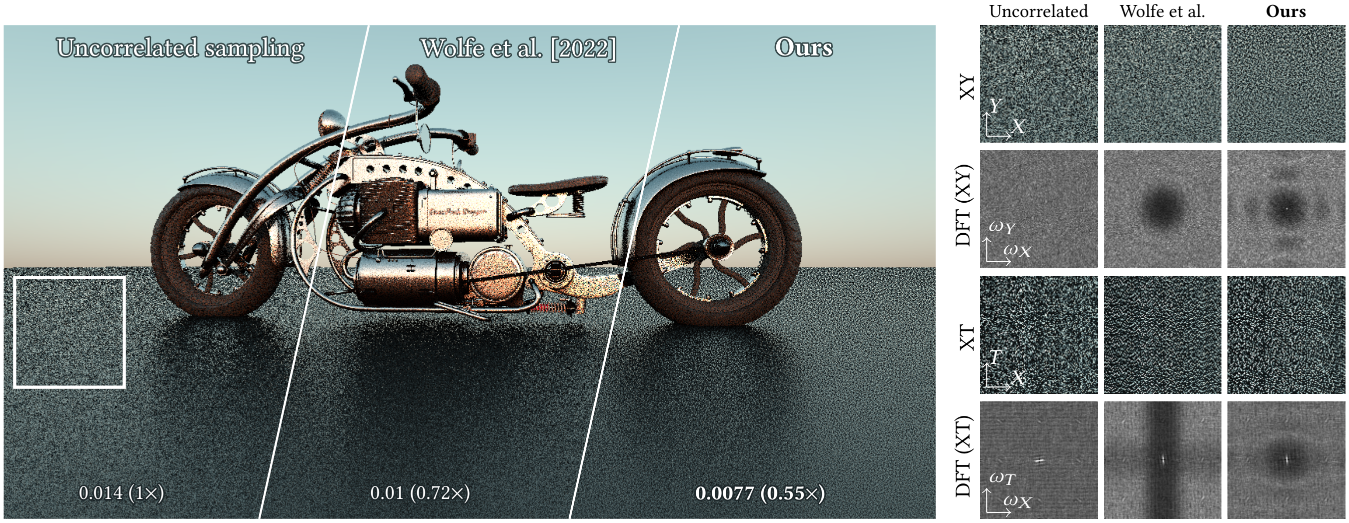


Fig. 1. We propose an optimization framework to obtain perceptually pleasing error distribution in Monte Carlo animation rendering. The output of our algorithm is a sample set spanning multiple image pixels and frames. Here we show an image of a 30-frame sequence rendered with 1 sample/pixel per frame. We display a version of the animation filtered temporally using the kernel of Mantiuk et al. [2021], to mimic its perception at one time instant. On the right we a show spatial (XY) crop and a spatio-temporal (XT) slice, along with the power spectra (DFT) of their corresponding error images. Our error distribution exhibits better blue-noise properties than that of previous work [Wolfe et al. 2022], also reflected in the perceptual error metric reported on the left (see Section 5). To fully appreciate these results, please refer to the supplemental video and HTML viewer.

Independently estimating pixel values in Monte Carlo rendering results in a perceptually sub-optimal white-noise distribution of error in image space. Recent works have shown that perceptual fidelity can be improved significantly by distributing pixel error as blue noise instead. Most such works have focused on static images, ignoring the temporal perceptual effects of animation display. We extend prior formulations to simultaneously consider the spatial and temporal domains, and perform an analysis to motivate a perceptually better spatio-temporal error distribution. We then propose a practical error optimization algorithm for spatio-temporal rendering and demonstrate its effectiveness in various configurations.

CCS Concepts: • **Computing methodologies** → **Rendering**; **Perception**.

Additional Key Words and Phrases: Monte Carlo rendering, stochastic sampling, blue noise

ACM Reference Format:
Miša Korać , Corentin Salaün , Iliyan Georgiev, Pascal Grittmann, Philipp Slusallek, Karol Myszkowski, and Gurprit Singh. 2023. Perceptual error optimization for Monte Carlo animation rendering. In *SIGGRAPH Asia 2023 Conference Papers (SA Conference Papers '23), December 12–15, 2023, Sydney, NSW, Australia.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3610548.3618146

1 INTRODUCTION

Monte Carlo rendering numerically estimates light-transport integrals via random sampling which causes visible noise in the resulting image. Much work has focused on combating this noise by reducing the error in each pixel individually, e.g., via blue-noise or low-discrepancy sampling [Singh et al. 2019]. Applying such a pattern *independently* within each pixel improves the convergence rate towards a noise-free result. However, the resulting white-noise distribution of error over the image is visually sub-optimal.

It is well understood in digital half-toning literature that the human visual system (HVS) is less sensitive to image error that has high-frequency, i.e., blue-noise, distribution. Georgiev and Fajardo

[*] These authors have contributed equally to this work.

SA Conference Papers '23, December 12–15, 2023, Sydney, NSW, Australia

[2016] achieved such distribution in Monte Carlo rendering by carefully optimizing a global sample pattern *across image pixels*. This pattern yields higher perceptual fidelity by making the pixel estimates as different from each other as possible. This improvement occurs because the HVS applies a low-pass filter to the image [Chizhov et al. 2022], and the negative pixel correlation effectively stratifies the input to the low-pass convolution.

Following the work of Georgiev and Fajardo [2016], several practical methods have been devised to achieve high-quality blue-noise distribution for static-image rendering [Heitz and Belcour 2019; Belcour and Heitz 2021; Ahmed and Wonka 2020]. These are mostly heuristically derived. An exception is the method of Salaün et al. [2022] which leverages the perceptual framework of Chizhov et al. [2022] to compute a small sample set, tiled over the rendered image.

Reusing the same blue-noise sample set across the frames of an animation would maintain good blue-noise distribution, but the noise pattern would remain static over the image. This so-called shower-door effect [Kass and Pesare 2011] degrades visual quality and disrupts the perception of motion. To address this problem, Wolfe et al. [2022] made a first attempt at obtaining an error distribution for animation rendering that is blue-noise in both image space and time. Lacking firm perceptual grounding, they extend existing blue-noise-mask algorithms to optimize separately across screen-space and time, which leads to visually suboptimal results.

In this paper, we combine the image-space model of Chizhov et al. [2022] with a temporal perception model [Mantiuk et al. 2021] to quantify perceptual error in animation rendering and motivate the need for its high-frequency distribution in both space and time. We also incorporate explicit temporal filtering such as temporal anti-aliasing (TAA). Based on this spatio-temporal model, we adapt the optimization method of Salaün et al. [2022] to obtain scene-independent, precomputed sample sets. The resulting sample sets allow for low-sample animation rendering with higher perceptual fidelity than prior state of the art, thanks to the blue-noise distribution of error in both space and time. Figure 1 shows one frame of an animation rendered with our optimization algorithm.

2 RELATED WORK

Our goal is to optimize Monte Carlo rendering error *across pixels* as blue noise, in both image space and time. The survey of Singh et al. [2019] discusses methods for achieving blue noise on one integration domain (e.g., within a single pixel).

Blue-noise error distribution. Blue-noise distributions of image error appear frequently in dithering or stippling applications [Ulichney 1988]. The reason for their use is the lower sensitivity of the HVS to high-frequency noise ("blue noise"), resulting in a less perceptible error. High-frequency noise distribution corresponds to negative correlation between pixel values in a neighbourhood. For Monte Carlo rendering, Georgiev and Fajardo [2016] proposed a first practical approach that optimizes a blue-noise sample mask via simulated annealing. Their approach is limited to low-dimensional integration with few samples. Heitz et al. [2019] addressed these limitations by optimizing the scrambling keys of a Sobol sequence [Sobol' 1967]. Belcour and Heitz [2021] extended this optimization to a rank-1 lattice sampler. The method of Ahmed and Wonka [2020] scrambles an image-space Sobol sequence according to a z-code ordering of pixels to achieve an approximate blue-noise distribution. Salaün et al. [2022] employed sliced optimal transport [Paulin et al. 2020] to obtain a sample set optimized according to the perceptual model of Chizhov et al. [2022]. Recently, Wolfe et al. [2022] proposed extensions to the void-and-cluster [Ulichney 1988] and Georgiev and Fajardo's [2016] algorithms to generate blue-noise sample masks for animation rendering. All the aforementioned methods are *a priori*, i.e., they compute scene-agnostic sample patterns. Such precomputation is beneficial for practical application, though superior quality can be achieved by tailoring the distribution to the specific image being rendered. This can be done through *a posteriori* adaptation of sample distributions, once the pixels have been sampled [Heitz and Belcour 2019; Chizhov et al. 2022]. We extend the image-space model of Chizhov et al. [2022] to the temporal domain and apply the a priori optimization approach of Salaün et al. [2022] to acquire a sample pattern for each animation frame.

Perceptual modeling and rendering. The contrast sensitivity function (CSF) is an important characteristic of the HVS that determines the threshold contrast that is perceivable in spatio-temporal signals. A vast majority of CSF measurements focus on spatial patterns [Daly 1993; Barten 1999; Wuerger et al. 2020], and the resulting CSFs are modeled by a family of band-pass filters whose parameters change with luminance, color, and retinal eccentricity. Spatio-temporal CSFs have also been derived [Kelly 1979; Daly 1998; Robson 1966; Mantiuk et al. 2022], where temporal sensitivity [de Lange 1958] can be explained by sustained and transient temporal channels dedicated to processing slowly and quickly changing signals [Burbeck and Kelly 1980; Hammett and Smith 1992; Mantiuk et al. 2021]. The so-called window of visibility [Watson 2013; Watson and Ahumada 2016; Watson et al. 1986] is an example of such spatio-temporal CSF modeling. The window of visibility approach accounts for spatio-temporal signal processing and sampling that are inherent to any imaging pipeline. In rendering applications, such spatio-temporal CSFs have been used to focus expensive computation on the most visible regions only [Myszkowski et al. 1999; Yee et al. 2001]. In this work, we reduce perceived rendering error by employing a spatio-temporal CSF to optimize space-time sampling patterns.

Temporal anti-aliasing. Temporal anti-aliasing (TAA) combines pixels across multiple frames to reduce noise [Shinya 1993; Schied et al. 2017, 2018]. Such temporal filtering is simple and cheap, though ghosting artifacts arise if the scene changes too rapidly. These can be reduced via the use of motion vectors or other means of temporal reprojection [Hanika et al. 2021]. Our method can optimize the perceived screen-space error distribution of a TAA-filtered animation.

Bounding integration error. Quasi-Monte Carlo (QMC) integration methods use deterministic sample sequences. These sequences are carefully designed to minimize discrepancy which is a quality metric used to bound integration error [Ermakov and Leora 2019]. Recent work [Paulin et al. 2020] has shown an analogous error bound based on the Wasserstein distance instead [Kantorovich and Rubinstein 1958; Villani 2008]. This bound has been extended by Salaün et al. [2022] to perceptual error in single-image rendering. We further extend their bound to our spatio-temporal setting.

Table 1. Commonly used notations throughout the document.

Notation	Description
S_i, $\mathrm{S} = \{S_i\}$	Sample set for frame i, sample set for entire frame sequence
R_i, $\mathbf{R} = \{R_i\}$	Raw render result at frame i, sequence of all raw results
Q_i, $\mathbf{Q} = \{Q_i\}$	Displayed image at frame i, sequence of all displayed images
I_i, $\mathbf{I} = \{I_i\}$	Ground-truth image at frame i, sequence of all ground truths
ϵ_i, $\epsilon = \{\epsilon_i\}$	Perceptual-error image at frame i, perceptual-error sequence
g_s, g_t	Spatial perceptual kernel, temporal perceptual kernel
g_a	Explicit temporal accumulation (TAA) kernel
μ	Sample distribution (typically uniform)

3 SPATIO-TEMPORAL PERCEPTUAL MODEL

Our method builds on the perceptual model of Chizhov et al. [2022], which we extend to include the temporal model of Mantiuk et al. [2021] as well as explicit filtering via temporal anti-aliasing (TAA).

Notation. Given a sequence $\mathbf{Q} = \{Q_i\}$ of rendered images, we aim to minimize their perceived error compared to the sequence of corresponding references $\mathbf{I} = \{I_i\}$. Each image is a function $Q_i(S_i)$ of the sample pattern S_i that is used to render the i^{th} frame of an animation. We concisely express the sequence of rendered images $\mathbf{Q}(\mathrm{S})$ as a function of the sequence of sample patterns. Table 1 lists the most commonly used symbols throughout the paper.

Spatial perceptual error. We follow Chizhov et al. [2022] and model spatial perceptual response as a convolution. Hence the perceived error of the i^{th} frame viewed individually,

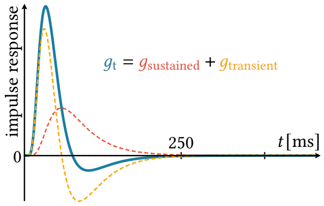
$$\epsilon_i(S_i) = g_\mathrm{s} * Q_i(S_i) - g_\mathrm{s} * I_i = g_\mathrm{s} * (Q_i(S_i) - I_i), \tag{1}$$

can be quantified by comparing the perceived image $g_\mathrm{s} * Q_i$ to the perceived reference $g_\mathrm{s} * I_i$. Here, g_s is an image-space Gaussian kernel that approximates the human visual system's (HVS) point spread function (PSF) [Chizhov et al. 2022]. The error image $\epsilon_i(S_i)$ then measures the error for each pixel in the i^{th} frame.

Spatio-temporal perceptual error. The human visual system (HVS) does not perceive each animation frame in isolation. Rather, it has been observed that temporal perception can be also modelled as a low-pass filter [Mantiuk et al. 2021; Burbeck and Kelly 1980; Hammett and Smith 1992]. We incorporate temporal filtering with a kernel g_t into the spatial model (1):

$$\epsilon(\mathrm{S}) = g_\mathrm{t} * g_\mathrm{s} * (\mathbf{Q}(\mathrm{S}) - \mathbf{I}). \tag{2}$$

Since both reference and rendered images are subject to temporal perception, the convolution with this kernel is applied to both. Here, $\epsilon(\mathrm{S})$ denotes the sequence of per-frame error images. In our experiments, we employ the kernel proposed by Mantiuk et al. [2021]. It is a sum of two components, a *sustained* kernel and a *transient* kernel, plotted in the inline figure. The sustained kernel encodes the response to slow temporal changes, and the transient kernel to fast changes [Hammett and Smith 1992; Burbeck and Kelly 1980]. Note that in Eq. (2) the filter g_t is applied as a sliding window over the frames.



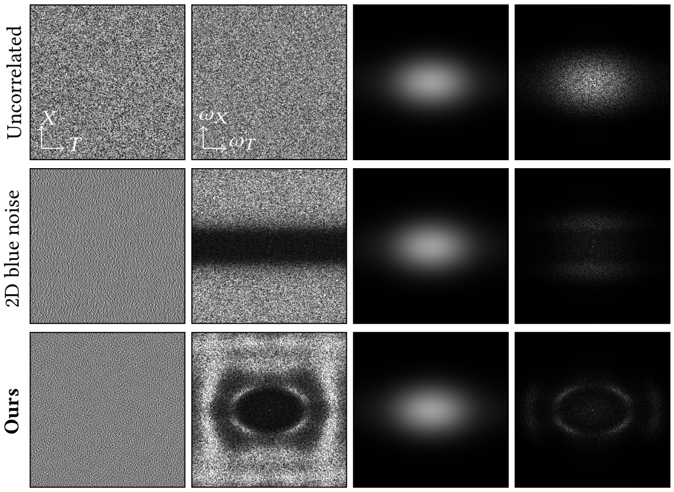


Fig. 2. Spatio-temporal (XT) slices of the error-image sequence (leftmost column) for white-noise (Uncorrelated), spatial-only blue-noise [Salaün et al. 2022] (2D blue noise), and spatio-temporal blue-noise (Ours) sample sets. The center two columns show the Fourier spectra of the error images (center left) and our perceptual kernel (center right). The rightmost column shows the product of these two (a.k.a. the perceptual error), i.e., the Fourier spectrum of the error image convolved with the kernel. Our optimization minimizes the error spectrum $\hat{\epsilon}$ (bottom row) and pushes the error outside of perceptible spatio-temporal frequency range (the window of visibility [Watson et al. 1986]; more details in Section 3).

Temporal anti-aliasing. TAA methods [Yang et al. 2020] compute pixel values as the weighted average of the current and previous frames. Such explicit filtering can be included in our model by expressing the image Q_i displayed at each frame as a convolution of the raw rendering results R_j at all (past) frames: $Q_i = [g_\mathrm{a} * \mathbf{R}]_i$. Substituting into Eq. (2), the error-image sequence becomes

$$\epsilon(\mathrm{S}) = g_\mathrm{t} * g_\mathrm{s} * (g_\mathrm{a} * \mathbf{R}(\mathrm{S}) - \mathbf{I}). \tag{3}$$

In our experiments, we use an exponential moving average (EMA) kernel g_a, with weights $g_\mathrm{a}(j) = \alpha(1-\alpha)^j$, for $j \geq 0$, where $\alpha \in [0, 1)$ is a smoothing parameter (we use $\alpha = 0.2$). Note here that the perceptual kernels g_s and g_t are applied to both the image estimates and the reference image, but the TAA kernel g_a is applied only to the raw image estimates.

Optimization objective. Our objective is then to find the sample sequence S that minimizes the norm of the error-image sequence (3):

$$\mathrm{S}' = \operatorname*{arg\,min}_{\mathrm{S}} \|\epsilon(\mathrm{S})\|. \tag{4}$$

In our optimization algorithm, presented in the following section, we use the L_1 norm, i.e., we find the sample sequence that minimizes the sum of absolute values of all error-image pixels over all frames.

Discussion. We illustrate the impact of spatio-temporal kernel filtering in Fig. 2, which provides a visual representation of the error image for three different methods, i.e., sample sequences S (rows). The first column shows temporal slices of the raw error, i.e.,

$q = \mathbf{Q}(\mathrm{S}) - \mathbf{I}$, and the second column shows the power spectra of the discrete Fourier transform (DFT) of those raw-error slices. The last column shows the DFT spectra of the convolution of the error images with our spatio-temporal perceptual kernel $g = g_\mathrm{t} * g_\mathrm{s}$ (plotted in the second last column). Assuming that the viewing conditions and frame rate correspond to the kernels g_s and g_t, our spatio-temporal kernel g approximates the window of visibility [Watson et al. 1986]. This window is defined in the frequency domain and its size is determined by the cut-off spatio-temporal frequencies. Signal outside the window is considered *invisible* (imperceptible). By optimizing the sample sequence S to solve Eq. (4), we not only reduce the magnitude of the spectrum, but also push the energy outside the window of visibility as much as possible. This reduces the residual perceived error for all visible spatio-temporal frequencies.

In summary, in this section we have presented a perception-driven model (3) to assess the spatio-temporal quality of a sample sequence. We model human perception by a series of convolutions, and (optionally) include explicit temporal filtering (TAA). In our main results we use this objective for a priori sample optimization, as discussed in the next section. Figure 5 shows that a posteriori optimization can benefit from this formulation, too.

Similarly to prior work on spatial-only error optimization [Heitz et al. 2019; Heitz and Belcour 2019; Chizhov et al. 2022], we assume integrated (radiance) function to be locally smooth (Lipschitz continuous) in space and time. This smoothness assumption is essential for achieving a desirable outcome in the optimization process. The spatio-temporal CSF [Kelly 1979; Daly 1998; Mantiuk et al. 2022] further supports our assumption since the HVS is mostly sensitive towards low- to mid-frequency signals. In practice, this implies that the sampling quality is less relevant in regions where the smoothness assumption is not met.

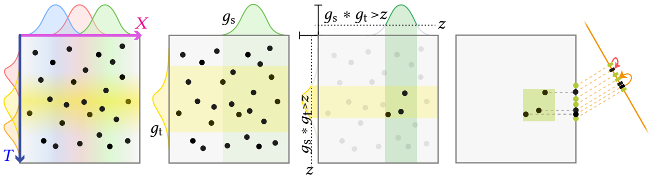
4 A PRIORI OPTIMIZATION

Our optimization problem (4) is similar in structure to that of Chizhov et al. [2022] who consider single-image optimization. This problem can be tackled in a priori or a posteriori manner (see Section 2). We focus on a priori optimization due to its higher practical value of computing a sample set once that can be used on any scene. To that end, we extend the method of Salaün et al. [2022] to our spatio-temporal setting.

A priori methods assume that the ground-truth image is constant [Georgiev and Fajardo 2016; Heitz and Belcour 2019; Belcour and Heitz 2021]. We extend this assumption to the temporal domain. Convolving $\mathbf{I}$ with the TAA kernel g_a thus becomes a no-op that allows us to simplify our objective function (3): we combine all kernels into a single spatio-temporal kernel g:

$$\epsilon(\mathrm{S}) = g_\mathrm{s} * g_\mathrm{t} * g_\mathrm{a} * (\mathbf{R}(\mathrm{S}) - \mathbf{I}) = g * (\mathbf{R}(\mathrm{S}) - \mathbf{I}). \tag{5}$$

In the a priori setting, both the raw sequence $\mathbf{R}(\mathrm{S})$ and reference sequence $\mathbf{I}$ are unknown, preventing the exact minimization of the error (5). Instead, we aim to minimize an upper bound of that error.


X
T
g_s
g_t
$g_\mathrm{s} * g_\mathrm{t} > z$
z
$g_\mathrm{s} * g_\mathrm{t} > z$
z

(a) XT domain (b) Kernel selection (c) Filtering operation (d) 1D optimal transport

Fig. 3. Visualization of how the gradients are estimated for our optimization. Given the spatio-temporal (XT) space and the kernels (a,b), we randomly threshold their convolution to select a subset of samples (c). The filtered sample set is then projected to a random 1D slice to compute the 1D-Wasserstein gradient (d). The process is repeated multiple times to obtain a sufficiently low-noise gradient estimate.

Perceptual-error bound. Under our perceptual model, the value of the j^{th} pixel in the i^{th} frame of the perceived raw sequence $g * \mathbf{R}(\mathrm{S})$ is an average of the responses of all samples, weighted by the kernel g centered at (i, j). Salaün et al. [2022, Appendix D] derived a bound for the absolute error of weighted integral estimates, based on filtered optimal transport. In our case their bound reads

$$|\epsilon_{i,j}(\mathrm{S})| \;\leq\; L \int_{\mathbb{R}} W\left(\mathrm{S}_{g_{i,j}>z}, \mu_{g_{i,j}>z}\right) \mathrm{d}z. \tag{6}$$

The bound assumes a smooth rendering function, i.e., the incident radiance on the continuous image plane, with Lipschitz constant L. It is an integral over Wasserstein distances W between the optimized sample distribution S and the target (uniform) distribution μ; these distributions are filtered to only include the mass at locations where the kernel value exceeds the threshold z [Salaün et al. 2022]:

$$\mathrm{S}_{g_{i,j}>z} = \{u \in \mathrm{S} \mid g_{i,j}(u) > z\}. \tag{7}$$

We use the 2-Wasserstein distance which is defined as

$$W(\mathrm{S}, \mu) = \left(\inf_{\gamma \in \Gamma(\mathrm{S},\mu)} \int_{\Omega^2} \|x - y\|^2 \, \mathrm{d}\gamma(x, y)\right)^{1/2}. \tag{8}$$

Here $\Gamma(\mathrm{S}, \mu)$ is the set of all possible transport plans between the two distributions [Bonnotte 2013]. Since the regular Wasserstein distance is difficult to compute, we further bound it via its sliced variant which involves only easy-to-compute 1D Wasserstein distances [Pitié et al. 2005]:

$$W(\mathrm{S}, \mu) \;\leq\; SW(\mathrm{S}, \mu) = \int_{\mathbb{S}^{d-1}} W\left(\mathrm{S}^\theta, \mu^\theta\right) \mathrm{d}\theta, \tag{9}$$

where S^θ and μ^θ are the projection of the sample set and the uniform density along the 1D line θ. We provide more details on the Wasserstein error bound in Section 1 of the supplemental document.

To bound the 1-norm of our objective function (5), we sum the error bounds Eq. (6) of all pixels j in all frames i:

$$\|\epsilon(\mathrm{S})\| = \sum_{i,j} |\epsilon_{i,j}(\mathrm{S})| \;\leq\; L \sum_{i,j} \int_{\mathbb{R}} \int_{\mathbb{S}^{d-1}} W\left(\mathrm{S}^\theta_{g_{i,j}>z}, \, \mu^\theta_{g_{i,j}>z}\right) \mathrm{d}\theta \mathrm{d}z. \tag{10}$$

Gradient-descent optimization. We minimize Eq. (10) via stochastic gradient descent, using Monte Carlo integration to estimate the involved integrals. We found that the Adam optimizer works best for our case, due to the sparse support of the kernels and the rather high noise of the gradient estimates. Details on the gradient computation can be found in supplemental Section 2.

Algorithm 1. Our spatio-temporal sample optimization.

```
function OptimizeSamples($IterationCount$, $BatchSize$)
  S = InitRandom()                              ← Initialize sample set
  $Optimizer$ = InitAdamOpimizer()
  for $t = 1..IterationCount$ do
    $g = \mathbf{0}$
    for $m = 1..BatchSize$ do
      $k$ = SelectRandomKernel($M$)                 ← Fig. 3b
      $s$ = FilterSampleSet(S, $k$)                 ← Fig. 3c
      $g$ += EvaluateSWGradient(s)             ← Accumulate gradient
    Update($Optimizer$, $g/BatchSize$)
  return S
```

The process is illustrated in Fig. 3. At each optimization step, we first randomly select a kernel $g_{i,j}$ (Fig. 3b). We then sample a filtering threshold z which yields a sample subset $\mathrm{S}_{g_{i,j}>z}$ (Fig. 3c). Finally, a random slice θ is sampled to estimate the gradient of the sliced Wasserstein distance (Fig. 3d). This process is repeated multiple times to reduce the variance. The resulting multi-sample gradient estimate is then used to perform one gradient-descent update step. Algorithm 1 summarizes these optimization steps.

5 RESULTS

We evaluate the rendering performance of our method by computing ray-traced direct illumination with PBRTv3 [Pharr et al. 2016]. We compare the results to the previous approach of Wolfe et al. [2022], independent per-frame spatial-only blue noise [Salaün et al. 2022], and the baseline of independent, white-noise sampling. Animations were rendered at 60Hz. Rendering is done using 1 sample per pixel unless stated otherwise.

We compute fixed-resolution spatio-temporal sample tiles, by toroidally wrapping the kernels during the optimization to ensure that they can be seamlessly tiled in space and time during rendering. If a spatial or temporal kernel has theoretically infinite support, we truncate it at the point where its values become negligible. We observe that a tile needs to be at least an order of magnitude larger than the truncated kernel to avoid tiling artifacts (supplemental, Fig. 1). In our experiments, the kernel size is 7×7 pixels and 8 frames wide (i.e., 7×7×8 pixels). We found that a tile size of 128×128×30 pixels achieves the best trade-off between optimization cost and tiling artifacts. We use the same tile size for all methods.

We computed the blue-noise tiles of Wolfe et al. [2022] using their public code. We slightly increased their spatial Gaussian kernel to a standard deviation of 2.1 (from 1.9), to match the spatial kernel used for all other methods. We used the public code of Salaün et al. [2022] to generate 30 independently optimized 2D blue-noise sample sets.

To mimic temporal perception in a static image, the renderings presented in the following are temporally pre-filtered with the kernel of Mantiuk et al. [2021] (see Section 3). The visual quality of the results is best appreciated by referring to the supplemental video and HTML viewer.

For quantitative comparison, we compute the perceptual relative mean squared error (pRelMSE), $\epsilon_i^2(S_i)/(I_i^2+0.01)$, at the i^{th} frame. That is, we filter the rendered image and the reference according to our model (3) and compute the relative MSE of the result for the desired frame (16$^{\text{th}}$ frame, unless stated otherwise).

Table 2. Perceptual error (pRelMSE) across different scenes. The numbers are the ratio of the pRelMSE of the different methods compared to the baseline of uncorrelated sampling; lower is better. Raw error values can be found in the supplemental document. We compare the methods with and without TAA. In both cases, and on every tested scene, our method achieves the lowest perceptual error. We set the standard deviation of Gaussian kernels to $\sigma = 2.1$; for Wolfe et al. [2022] we also report results with $\sigma = 1.9$ as used by them.

Scene	**Salaün et al. [2022]**		**Wolfe et al. [2022]**		**Ours**	
	TAA	no TAA	TAA	no TAA	TAA	no TAA
Chopper	0.61×	0.62×	0.69× (0.66×)	0.72× (0.69×)	**0.48**×	**0.55**×
Teapot	0.65×	0.63×	0.80× (0.63×)	0.78× (0.65×)	**0.56**×	**0.58**×
Modern Hall	0.90×	0.85×	0.98× (0.95×)	0.94× (0.91×)	**0.87**×	**0.83**×
Living room	0.87×	0.82×	0.89× (0.86×)	0.86× (0.82×)	**0.84**×	**0.80**×
Dragon	0.54×	0.52×	0.66× (0.62×)	0.67× (0.63×)	**0.48**×	**0.51**×
Veach MIS	0.87×	0.83×	0.99× (0.97×)	0.92× (0.90×)	**0.72**×	**0.69**×

We apply our method to animation rendering with and without temporal anti-aliasing (TAA). For our method we optimize on sample set for each variant, tailored to the filter. Table 2 summarizes our quantitative results across a diverse set of test scenes. It shows the pRelMSE of our method and previous works [Salaün et al. 2022; Wolfe et al. 2022] relative to uncorrelated (i.e., white-noise) spatio-temporal sampling. Across all scenes, our method consistently achieves better results, both with and without TAA.

Direct viewing. Figures 1 and 6 show results for animations without TAA using 1 sample per pixel. To aid interpretation of the results, the figures display the discrete Fourier transform (DFT) of different zoom-ins. Especially on the temporal slice in the bottom right of Fig. 1, these show clearly where the improvements of our method stem from: While the previous approach of Wolfe et al. [2022] explicitly optimizes for 2D blue noise in image space and 1D blue noise along the temporal domain, our method optimizes samples for an exact kernel, dictated by a perception model. Consequently, the frequency distribution of the error with our method better matches the filter, resulting in a lower perceived error.

Our algorithm can be used to optimize sample sets for any number of samples per pixel. Figure 7 shows an example using four samples. Compared to previous work, our approach achieves a better blue-noise distribution also at higher sample counts.

Temporal anti-aliasing. The behavior with explicit TAA filtering is similar to that under direct viewing. Again, our method is consistently better than uncorrelated sampling, previous work [Wolfe et al. 2022], and independent 2D blue noise [Salaün et al. 2022] across all test scenes (Table 2). Figure 9 shows the TAA (and perception) filtered frames of two scenes. For our method, we compare two different optimization objectives: our full model and a simplified version where we left out the perception filter and only optimized for TAA. Optimizing only for TAA still outperforms previous work, but yields 5-10% higher perceived error than utilizing the full model. This supports our hypothesis that optimizing for a more accurate kernel yields best results.

Table 3. Ablation test for different optimization objectives. On an animation with TAA we test sample sets optimized for three temporal kernels: a symmetric Gaussian (standard deviation 2.1), an EMA TAA kernel, and our full model incorporating the TAA kernel and the perception kernel of Mantiuk et al. [2021]. The numbers are the relative reduction in perceived noise (pRelMSE) compared to the method of Wolfe et al. [2022]; lower is better. Raw error values can be found in the supplemental document. Lowest error is achieved when the optimization is tailored to the full filter.

Scene	Gaussian	TAA	Perception + TAA
Chopper	0.75×	0.75×	**0.70**×
Teapot	0.78×	0.76×	**0.70**×
Modern Hall	0.92×	0.90×	**0.89**×
Living room	0.97×	0.96×	**0.95**×
Dragon	0.79×	0.77×	**0.72**×
Veach MIS	0.77×	0.78×	**0.74**×

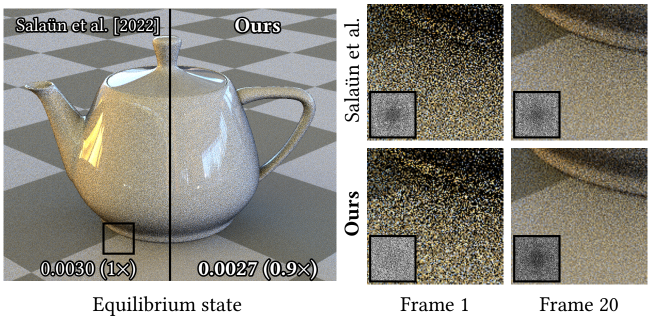


Fig. 4. Comparison between independent spatial-only optimization [Salaün et al. 2022] and our method on an animation with TAA. The image on the left is the 20th frame, the zoom-ins show the state at the 1st and 20th frame, along with the DFT spectra of their error images. Spatial-only optimization is much better in the first frame, where no temporal filtering occurs. Our method shows better blue-noise quality once the steady state is reached.

6 DISCUSSION

Impact of kernel shape. Our method differs from that of Wolfe et al. [2022] in two main aspects: the model and the optimization process. The approach of Wolfe et al. [2022] does not directly translate to a kernel in our optimization framework, since they separate the spatial and temporal dimensions. Therefore, to better understand how much of our improvements are due to the model, and how much due to the optimization itself, we performed an ablation where we optimized sample sets for kernels different from the one used for final filtering.

Table 3 summarizes the results. We report the error values for three different kernels: a symmetric Gaussian (standard deviation 2.1), the TAA kernel, and the full TAA and temporal perception kernel [Mantiuk et al. 2021]. As expected, the best result is achieved when optimizing for the full model. Since the TAA and Gaussian kernels have similar shape, their results only differ by a few percent. These results indicate that, while matching the overall shape of the final kernel is important, exact match is not critical.

Note that all models in Table 3 yield lower error than the approach of Wolfe et al. [2022]. This indicates that their separation into spatial and temporal components, while helpful for convergence in their optimizer, hampers the attainable quality.

Performance on the first frames. Our optimization assumes that a sufficient number of past frames are available to apply the full temporal kernel. This is not the case early in an animation, as shown in Fig. 4. The figure compares our result with spatial-only blue noise [Salaün et al. 2022] under environment-map illumination with TAA filtering. In the first frame, spatial-only optimization yields higher quality, since no temporal filtering can yet occur. In subsequent frames, our method performs better. This is also visible in the DFT spectra (insets) where our method has fewer low-frequency error components, i.e., a better blue-noise distribution.

We extend this analysis by comparing the evolution of perceptual error with the number of frames. Figure 8 shows this evolution on the Chopper, Dragon and Teapot scenes. The methods compared are uncorrelated sampling, Salaün et al. [2022], Wolfe et al. [2022] and ours. The results show that at the start of each curve, when few frames are accumulated, spatial-only optimization performs best. This confirms the results presented above. However, as the frames accumulate and the equilibrium state is reached, our method obtains the lowest perceptual error.

A posteriori optimization. A priori optimization is inherently limited in the achievable quality because the optimized sample set must generalize to arbitrary scenes and importance-sampling transformations. A posteriori optimization can achieve better results, but a truly practical method has yet to be found. To explore the quality achievable by an a posteriori approach using our objective, we extended the method of Chizhov et al. [2022] to the temporal domain, using our model. Specifically, we employ their "vertical" optimization which selects one out of 15 candidate samples for each pixel, to solve Eq. (4). We compare the result to our a priori optimization for the same kernel on the Chopper scene in Fig. 5. Here, a posteriori optimization yields an notable improvement of 60%. These results indicate that further research on (practical) a posteriori methods is worthwhile and can benefit from our formulation.

Optimization cost. Our sample sets need only be computed once per filter kernel, number of integration dimensions, sample count, and frame rate. Nevertheless, when multiple variations of these parameters are desired, computation cost may become a concern. Our CPU implementation takes about 1–2 days to optimize one sample set using 10k SGD steps with a mini-batch size of 4k. The theoretical bottleneck is the computation of the 1D optimal transport, which relies on an $n\log(n)$ sorting operation per gradient-descent iteration. In practice, computation speed is significantly affected by accessing sample subsets that are scattered in memory. Furthermore, the use of a small subset of samples with non-zero gradients per step necessitates the use of large batch sizes, resulting in higher computation costs. Despite these challenges, parallelism can be harnessed within the algorithm: across different projections of the mini-batch on the CPU, also for sorting on the GPU. We believe that, with further performance improvements, e.g., using a pre-optimized set as initialization, computation time can be reduced to at most a few hours per sample set.

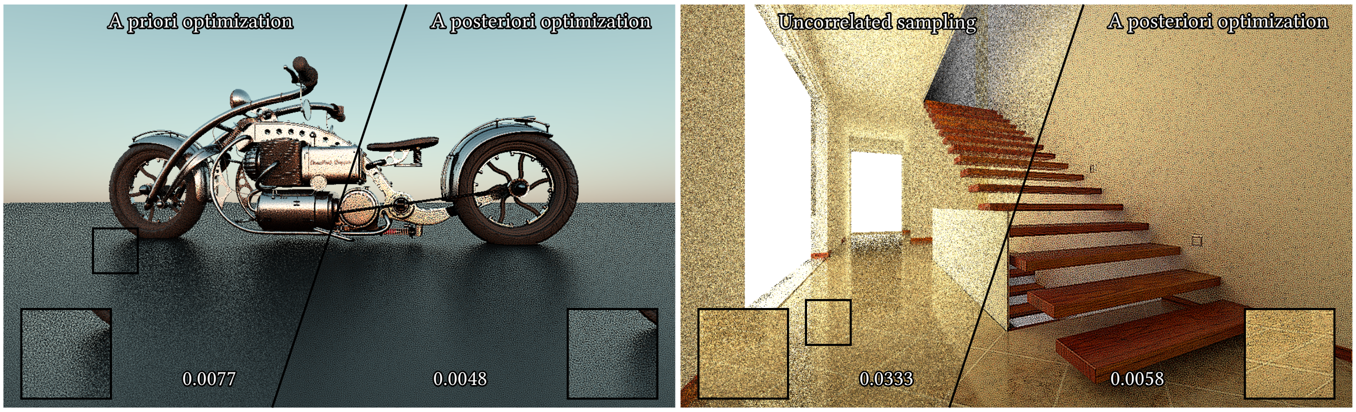


Fig. 5. Our theory can be used to extend a posteriori perceptual error optimization [Chizhov et al. 2022]. Here we show improvement in direct-illumination (2D sampling) on the left and path tracing rendering (10D sampling) on the right. Unlike a priori methods, a posteriori optimization is not sensitive to sampling dimensionality and achieves higher quality thanks to image-based optimization. All images show the 16th animation frame filtered with the temporal perception kernel of Mantiuk et al. [2021], along with the corresponding pRelMSE values.

Limitations. The signal-constancy assumptions made by a priori perceptual error optimization hold only locally and approximately. In regions of large signal variation, e.g., thin shadow penumbrae (spatially) or fast-moving objects (temporally), perceptual error increases. Temporal variations could be addressed via the use of motion vectors, which we leave for future work.

A significant limitation of a priori optimization methods lies in their applicability to more complex rendering algorithms, such as path tracing. This is due to the increased sampling dimensionality and the variation of the rendering function with longer paths. A priori optimization is not sensitive to dimensionality and can tailor the sampling to the rendered image.

The capacity of the error distribution to influence perceptual quality is also contingent upon the noise level present in the scene. When the noise level is low, the impact of the error distribution on perceptual quality diminishes.

Future work. In this work we use a basic spatio-temporal CSF model [Chizhov et al. 2022; Mantiuk et al. 2021], but our framework (Section 3) supports arbitrary filters. Exploring more advanced CSF models [Mantiuk et al. 2022] that account for display luminance (e.g., darker displays increase the HVS tolerance for contrast errors and flickering, while saving energy) and foveation (sparser sampling with increasing retinal eccentricity) could be a way to further improve visual quality. A hold-type blur of moving objects that arises in the HVS as a function of display persistence and refresh rate [Jindal et al. 2021] can also lead to increasing the HVS tolerance to rendering error. One could specifically optimize sampling by considering content-dependent visual masking [Mantiuk et al. 2021], e.g., by precomputing a texture-specific sample set.

Another promising future direction would be to find a practical approach for a posteriori optimization of sample patterns. Ideally, such a method would work in real-time applications and complex light-transport algorithms.

The relationship between sample optimization and denoising is another interesting topic. As noted by Heitz and Belcour [2019] and Chizhov et al. [2022], high-frequency error distributions can afford higher fidelity when denoising via low-pass filtering; existing denoisers may need adjustment or retraining to optimally handle such input. We believe that our high-frequency spatio-temporal distribution paves the way for devising improved, correlation-aware interactive denoising methods.

7 CONCLUSION

We have introduced a general model and a practical method for spatio-temporal sample optimization for Monte Carlo animation rendering. Our method accounts for both perceptual and explicit temporal filtering. To achieve practicality, we extend an existing a priori optimization method to support our spatio-temporal model. As a result, we can precompute scene-agnostic sample sets that yield considerable improvements over previous work in terms of perceived noise quality.

ACKNOWLEDGEMENTS

This project has received funding from the European Union's Horizon 2020 research and innovation program under Marie Skłodowska-Curie grant agreement №956585. We thank the anonymous reviewers for their feedback, and the authors of the following scenes: julioras3d (Chopper), NewSee2l035 (Modern Hall), Benedikt Bitterli (Utah Teapot), Wig42 (Living Room), and Jay Hardy (White Room).

REFERENCES

Abdalla GM Ahmed and Peter Wonka. 2020. Screen-space blue-noise diffusion of Monte Carlo sampling error via hierarchical ordering of pixels. *ACM Transactions on Graphics (TOG)* 39, 6 (2020), 1–15.

Peter G.J. Barten. 1999. *Contrast sensitivity of the human eye and its effects on image quality.* SPIE – The International Society for Optical Engineering. https://doi.org/10.1117/3.353254

Laurent Belcour and Eric Heitz. 2021. Lessons Learned and Improvements When Building Screen-Space Samplers with Blue-Noise Error Distribution. In *ACM SIGGRAPH 2021 Talks* (Virtual Event, USA) *(SIGGRAPH '21)*. Association for Computing Machinery, New York, NY, USA, Article 9, 2 pages. https://doi.org/10.1145/3450623.3464645

Nicolas Bonnotte. 2013. *Unidimensional and evolution methods for optimal transportation.* Ph.D. Dissertation. Paris 11.

Christina A. Burbeck and D. H. Kelly. 1980. Spatiotemporal characteristics of visual mechanisms: excitatory-inhibitory model. *J. Opt. Soc. Am.* 70, 9 (Sep 1980), 1121–1126. https://doi.org/10.1364/JOSA.70.001121

Vassillen Chizhov, Iliyan Georgiev, Karol Myszkowski, and Gurprit Singh. 2022. Perceptual error optimization for Monte Carlo rendering. *ACM Trans. Graph.* 41, 3 (2022). https://doi.org/10.1145/3504002

S. Daly. 1993. The Visible Differences Predictor: An Algorithm for the Assessment of Image Fidelity. In *Digital Image and Human Vision*. 179–206.

S. Daly. 1998. Engineering observations from spatiovelocity and spatiotemporal visual models. In *Human Vision and Electronic Imaging III*. SPIE Vol. 3299, 180–191.

H. de Lange. 1958. Research into the Dynamic Nature of the Human Fovea→Cortex Systems with Intermittent and Modulated Light. I. Attenuation Characteristics with White and Colored Light. *J. Opt. Soc. Am.* 48, 11 (1958), 777–784.

Sergey Ermakov and Svetlana Leora. 2019. Monte Carlo Methods and the Koksma-Hlawka Inequality. *Mathematics* 7, 8 (2019). https://doi.org/10.3390/math7080725

Iliyan Georgiev and Marcos Fajardo. 2016. Blue-noise dithered sampling. In *ACM SIGGRAPH 2016 Talks*. 1–1.

S.T. Hammett and A.T. Smith. 1992. Two temporal channels or three? A re-evaluation. *Vision Research* 32, 2 (1992), 285–291. https://doi.org/10.1016/0042-6989(92)90139-A

Johannes Hanika, Lorenzo Tessari, and Carsten Dachsbacher. 2021. Fast Temporal Reprojection without Motion Vectors. *Journal of Computer Graphics Techniques Vol* 10, 3 (2021).

Eric Heitz and Laurent Belcour. 2019. Distributing monte carlo errors as a blue noise in screen space by permuting pixel seeds between frames. In *Computer Graphics Forum*, Vol. 38. Wiley Online Library, 149–158.

Eric Heitz, Laurent Belcour, Victor 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*. 1–2.

Akshay Jindal, Krzysztof Wolski, Karol Myszkowski, and Rafał K. Mantiuk. 2021. Perceptual Model for Adaptive Local Shading and Refresh Rate. *ACM Trans. Graph.* 40, 6, Article 281 (2021).

Leonid V. Kantorovich and Gennady S. Rubinstein. 1958. On a space of completely additive functions. *Vestnik Leningrad Univ 13* 7 (1958), 52–59.

Michael Kass and Davide Pesare. 2011. Coherent noise for non-photorealistic rendering. *ACM Transactions on Graphics (TOG)* 30, 4 (2011), 1–6.

D.H. Kelly. 1979. Motion and Vision 2. Stabilized Spatio-Temporal Threshold Surface. *Journal of the Optical Society of America* 69, 10 (1979), 1340–1349.

Rafał K. Mantiuk, Maliha Ashraf, and Alexandre Chapiro. 2022. StelaCSF: A Unified Model of Contrast Sensitivity as the Function of Spatio-Temporal Frequency, Eccentricity, Luminance and Area. *ACM Trans. Graph.* 41, 4, Article 145 (2022).

Rafał K Mantiuk, Gyorgy Denes, Alexandre Chapiro, Anton Kaplanyan, Gizem Rufo, Romain Bachy, Trisha Lian, and Anjul Patney. 2021. FovVideoVDP: A visible difference predictor for wide field-of-view video. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–19.

K. Myszkowski, P. Rokita, and T. Tawara. 1999. Perceptually-Informed Accelerated Rendering of High Quality Walkthrough Sequences. In *Proc. EGSR*. 5–18.

Lois Paulin, Nicolas Bonneel, David Coeurjolly, Jean-Claude Iehl, Antoine Webanck, Mathieu Desbrun, and Victor Ostromoukhov. 2020. Sliced optimal transport sampling. *ACM Trans. Graph.* 39, 4 (2020), 99.

Matt Pharr, Wenzel Jakob, and Greg Humphreys. 2016. *Physically based rendering: From theory to implementation.* Morgan Kaufmann.

François Pitié, Anil C. Kokaram, and Rozenn Dahyot. 2005. N-Dimensional Probablility Density Function Transfer and its Application to Colour Transfer. In *10th IEEE International Conference on Computer Vision*. IEEE Computer Society. https://doi.org/10.1109/ICCV.2005.166

John Robson. 1966. Spatial and Temporal Contrast-Sensitivity Functions of the Visual System. *Journal of The Optical Society of America* 56 (08 1966). https://doi.org/10.1364/JOSA.56.001141

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

Christoph Schied, Anton Kaplanyan, Chris Wyman, Anjul Patney, Chakravarty R Alla Chaitanya, John Burgess, Shiqiu Liu, Carsten Dachsbacher, Aaron Lefohn, and Marco Salvi. 2017. Spatiotemporal variance-guided filtering: real-time reconstruction for path-traced global illumination. In *Proceedings of High Performance Graphics*. 1–12.

Christoph Schied, Christoph Peters, and Carsten Dachsbacher. 2018. Gradient estimation for real-time adaptive temporal filtering. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 1, 2 (2018), 1–16.

Mikio Shinya. 1993. Spatial Anti-Aliasing for Animation Sequences with Spatio-Temporal Filtering. In *Proc. SIGGRAPH 93*. 289–296.

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

R.A. Ulichney. 1988. Dithering with blue noise. *Proc. IEEE* 76, 1 (1988), 56–79. https://doi.org/10.1109/5.3288

C. Villani. 2008. *Optimal Transport: Old and New.* Springer Berlin Heidelberg. https://books.google.fr/books?id=hV8o5R7_5tkC

Andrew B Watson. 2013. High frame rates and human vision: A view through the window of visibility. *SMPTE Motion Imaging Journal* 122, 2 (2013), 18–32.

Andrew B Watson and Albert J Ahumada. 2016. The pyramid of visibility. *Electronic Imaging* 2016, 16 (2016), 1–6.

Andrew B. Watson, Albert J. Ahumada, and Joyce E. Farrell. 1986. Window of visibility: a psychophysical theory of fidelity in time-sampled visual motion displays. *J. Opt. Soc. Am. A* 3, 3 (1986), 300–307.

Alan Wolfe, Nathan Morrical, Tomas Akenine-Möller, and Ravi Ramamoorthi. 2022. Spatiotemporal Blue Noise Masks. In *Eurographics Symposium on Rendering*. https://doi.org/10.2312/sr.20221161

Sophie Wuerger, Maliha Ashraf, Minjung Kim, Jasna Martinovic, María Pérez-Ortiz, and Rafał K. Mantiuk. 2020. Spatio-chromatic contrast sensitivity under mesopic and photopic light levels. *Journal of Vision* 20, 4 (04 2020), 23–23. https://doi.org/10.1167/jov.20.4.23

Lei Yang, Shiqiu Liu, and Marco Salvi. 2020. A Survey of Temporal Antialiasing Techniques. *Computer Graphics Forum* 39, 2 (2020), 607–621. https://doi.org/10.1111/cgf.14018 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.14018

Hector Yee, Sumanita Pattanaik, and Donald P. Greenberg. 2001. Spatiotemporal Sensitivity and Visual Attention for Efficient Rendering of Dynamic Environments. *ACM Trans. Graph.* 20, 1 (2001), 39–65.

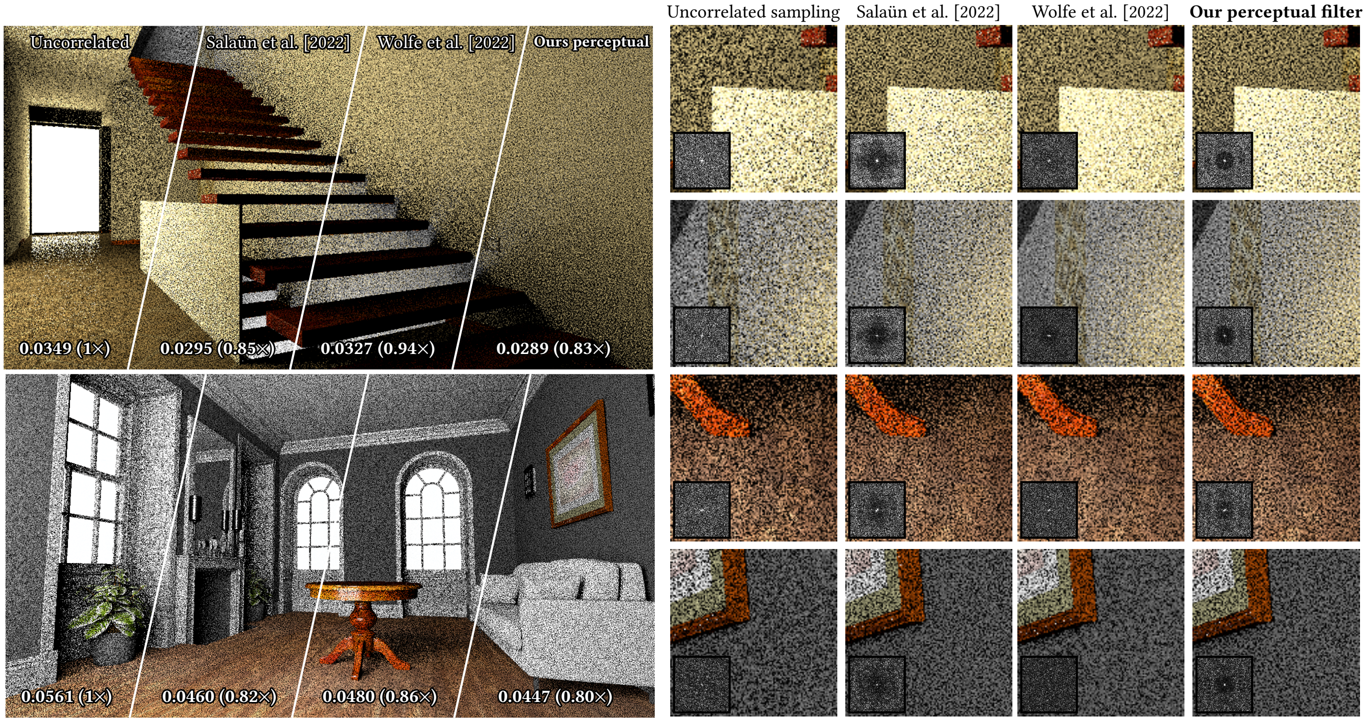


Fig. 6. Comparison of the 16th animation frame, without TAA. To mimic human perception, for display we apply the temporal filter of Mantiuk et al. [2021]. We compare our method to uncorrelated sampling and the methods of Salaün et al. [2022] and Wolfe et al. [2022]. The insets in each crop show the DFT of the error image, and the numbers on the left are the pRelMSE of each method (lower is better). We achieve visible improvements over previous work on all scenes, and a more pronounced blue-noise distribution in the DFT spectrum. Please refer to the supplemental HTML viewer to better appreciate the differences.

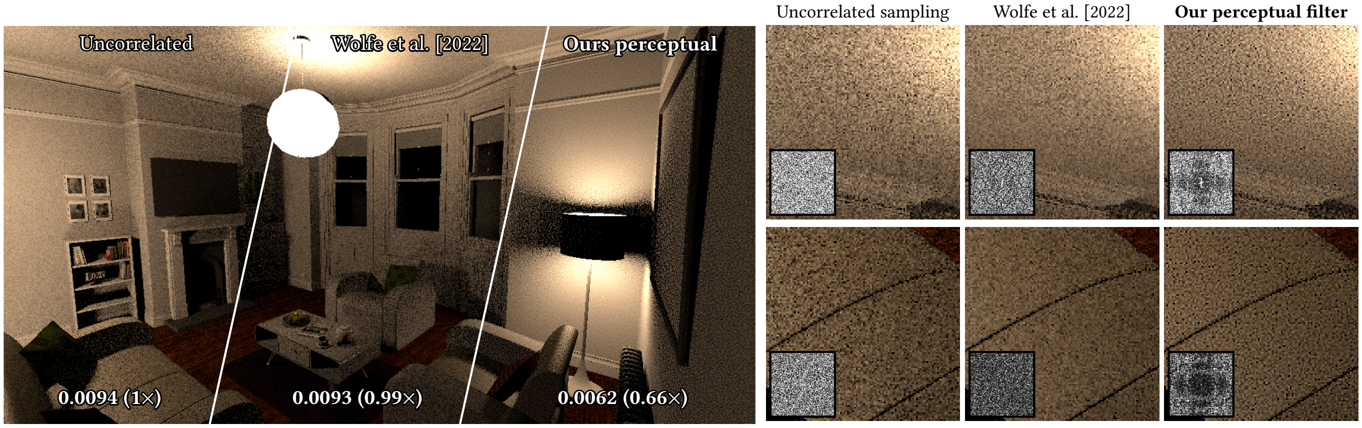


Fig. 7. Rendering comparison on the 10th animation frame, rendered with 4 samples per pixel. To mimic human perception, for display we apply the temporal filter of Mantiuk et al. [2021]. We compare our method to uncorrelated sampling and the method of Wolfe et al. [2022]. The insets in each crop show the DFT of the error image, and the numbers on the left are the pRelMSE of each method (lower is better). Our method preserves the desirable blue-noise error distribution also at higher sample counts.

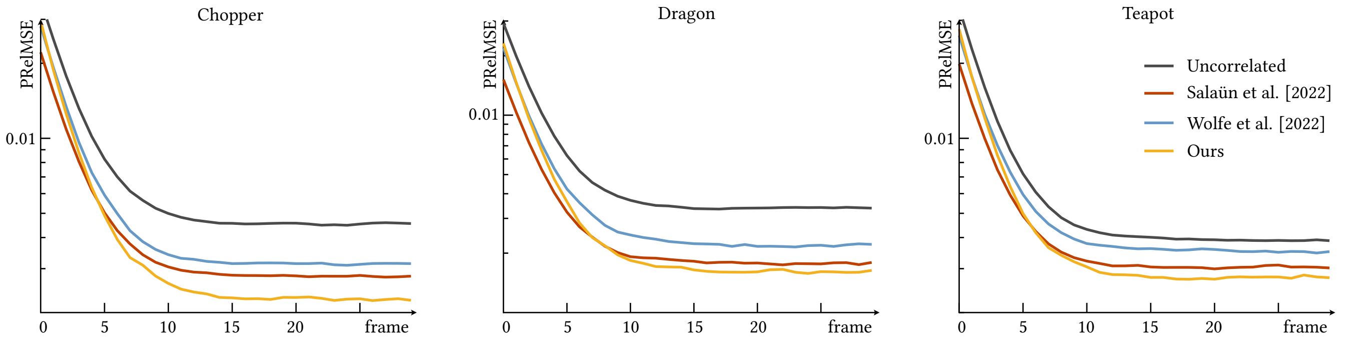


Fig. 8. Perceptual error across the first 30 animation frames (without motion) for three scenes rendered with TAA. The error initially reduces for all methods, as more frames are included in the temporal filter; until frame 15, where the full support of the kernel is reached. Spatial-only blue noise [Salaün et al. 2022] performs best for the first few frames, where not much temporal filtering yet occurs. Our optimization achieves the lowest perceptual error at the steady state.

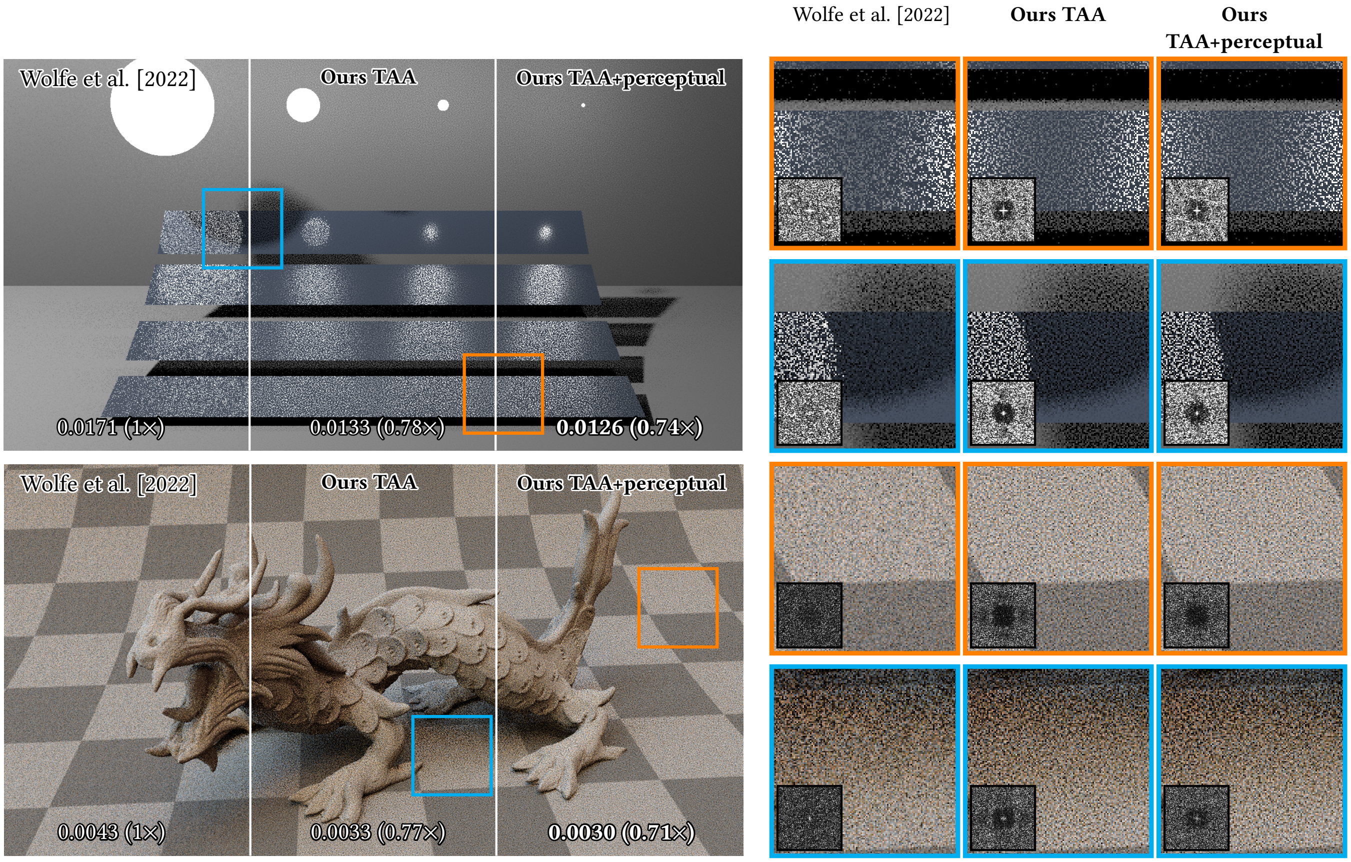


Fig. 9. Rendered images with temporal anti-aliasing (TAA). We mimic human perception by applying the temporal filter of Mantiuk et al. [2021]. As an ablation, we compare our method optimized only for the TAA kernel (center) and the full result (right) to previous work. We provide 2 crops for each scene associated with the DFT of the error in the region. The numbers at the bottom are the pRelMSE for 16^{th} frame for each method, lower is better. Optimizing only for TAA already performs well, but optimizing for both TAA and perception yields best results.

Perceptual error optimization for Monte Carlo animation rendering: Supplemental document

MIŠA KORAĆ [*], Saarland University, DFKI, Germany
CORENTIN SALAÜN [*], Max Planck Institute for Informatics, Germany
ILIYAN GEORGIEV, Adobe, UK
PASCAL GRITTMANN, Saarland University, Germany
PHILIPP SLUSALLEK, Saarland University, DFKI, Germany
KAROL MYSZKOWSKI, Max Planck Institute for Informatics, Germany
GURPRIT SINGH, Max Planck Institute for Informatics, Germany

In this document we present additional details regarding the Wasserstein error bound for Monte Carlo integration, offering insights into the computation of the Filtered Sliced Wasserstein derivative for the gradient-based optimizer, and show further results from our study.

1 WASSERSTEIN ERROR BOUND

In their work, Paulin et al. [2020] employed a variation of the Koksma-Hlawka inequality based on Optimal transport, specifically the Wasserstein distance.

$$\left|Q(\mathrm{S}) - I\right| \leq L_{w \cdot f} W(\mathrm{S}, \rho) \tag{1}$$

Where f is the integrated function and w an arbitrary kernel function. This bound can be derived from the Lipschitz inequality. Unlike the Discrepancy, the Wasserstein distance offers the advantage of being differentiable, making it more suitable for certain applications and optimization tasks.

The p-Wasserstein distance is defined as

$$W_p(\nu, \rho) = \left(\inf_{\gamma \in \Gamma(\nu,\rho)} \int_{\mathcal{X}^2} |x - y|^p \, \mathrm{d}\gamma(x, y) \right)^{1/p}, \tag{2}$$

where p can be arbitrary integer greater or equal to 1. ν and ρ are two density distribution. In practice we use the 2-Wasserstein distance. For convenience, we will simplify the notation as $W(\nu)$.

Initially defined for a single pixel, Wasserstein error bound has been extended by Salaün et al. [2022] to a Filtered bound. This extended bound allows for the inclusion of filters that overlap multiple pixels, transforming the problem into a multi-class one.

$$\left|Q_w(\mathrm{S}) - I_w\right| \leq \; L_f \int_{\mathbb{R}} W(\mathrm{S}_{w>z}) \, \mathrm{d}z \tag{3}$$

Note that in Eq. (3) the kernel w is part of the Wasserstein distance. This makes it a tailored distance metric wrt the kernel function. Minimizing the multi-class error bound is equivalent to solving the following barycentric optimization problem:

$$\mathrm{S} \; = \; \underset{\mathrm{S}}{\arg\min} \int_0^M \int_{\mathbb{R}} W\big(\mathrm{S}_{w_m>z}\big) \, \mathrm{d}z \, \mathrm{d}m \tag{4}$$

with M the number of kernel/pixel in the image sequence.

Sliced Wasserstein distance. The Wasserstein distance can be bounded by a variation called the Sliced Wasserstein distance, which in turn is bounded by the Wasserstein distance at a factor determined solely by the dimensionality.

$$W(\mathrm{S}) \leq SW(\mathrm{S}) \leq C^d \cdot W(\mathrm{S}) \tag{5}$$

This property makes the Sliced Wasserstein distance an excellent metric to substitute for the Wasserstein distance. Moreover, the Sliced Wasserstein distance is often more practical to compute, making it a favorable choice for many applications.

2 FILTERED SLICED WASSERSTEIN DERIVATIVE

Our optimization algorithm is based on an iterative gradient-based method, which is defined by the following equation:

$$\mathrm{S}_{i+1} = \mathrm{S}_i - \eta \frac{d}{d\mathrm{S}_i} \sum_{j=1}^{M} \int_{\mathbb{R}} SW(\mathrm{S}_{g_j>z}) dz \tag{6}$$

where S_i represents the sample-set at iteration i, and η is the learning rate. This iterative approach allows us to update the parameters in each iteration, progressively improving the optimization process. The details of the derivative are defined as follows:

$$\frac{d}{d\mathrm{S}_i} \sum_{j=1}^{M} \int_{\mathbb{R}} SW(\mathrm{S}_{g_j>z}, \rho_{g_j>z}) dz$$
$$= \frac{d}{d\mathrm{S}_i} \sum_{j=1}^{M} \int_{\mathbb{R}} \int_{\mathbb{S}^{d-1}} W(\mathrm{S}^{\theta}_{g_j>z}) d\theta dz \tag{7}$$

This equation represents the gradient of the objective function with respect to the parameters S_i at iteration t. It captures the direction and magnitude of the steepest ascent or descent in the parameter space during the optimization process.

In the given equation, the Wasserstein distance is computed in 1D by projecting both the sample set and the target density onto a 1D line. This projection operation is known as the Sliced Wasserstein

[*] These authors have contributed equally to this work.

SA Conference Papers '23, December 12–15, 2023, Sydney, NSW, Australia

distance. By reducing the dimensionality, it becomes feasible to derive a straightforward solution for the derivative using the inverse cumulative distribution function of the target density. If we extract a sample x_i from the sample set, the derivative of its position with respect to the Sliced Wasserstein distance is given by:

$$\frac{d}{dx_i} W(\mathrm{S}^{\theta}_{g_j>z}) = \frac{m}{n}\left[2\frac{x^{\theta}_i}{m} - 2\int_{\frac{i-1}{m}}^{\frac{i}{m}} F^{-1}_{\rho^{\theta}}(x)\,\mathrm{d}x\right] \qquad (8)$$

Here $F^{-1}_{\rho^{\theta}}(x)$ denotes the inverse cumulative distribution function of the projected target density, and the integral $\int_{\frac{i-1}{m}}^{\frac{i}{m}}$ represents the quantile associated with the position of i-th sample sorted by projected position x_i. x^{θ}_i is the position of the projection of the point x_i along the direction θ. m represent the number of point in the $\mathrm{S}_{g_j>z}$ and n the total number of points of the sample set. The computation of the inverse cumulative distribution function (CDF) and quantiles can be efficiently solved by sorting the list of projected samples for the quantile calculation and sorting the discretized projected target density for the inverse CDF. By relying solely on sorting operations, this process becomes efficient for an optimal transport-based methods. For more details about the derivation we refer to Salaün et al. [2022] as the proof on the error bound and derivatives have been made for any arbitrary kernel including spatio-temporal ones.

3 ADDITIONAL RESULTS

In this section we present additional results to those in the main paper. This includes illustration of artifacts due to insufficiently large tile size in Fig. 1, perceptual error tables comparing different optimizations for our method in Table 1 and the comparison of our method with uncorrelated sampling, [Wolfe et al. 2022] and [Salaün et al. 2022] for spatio-temporal rendering in Table 2.

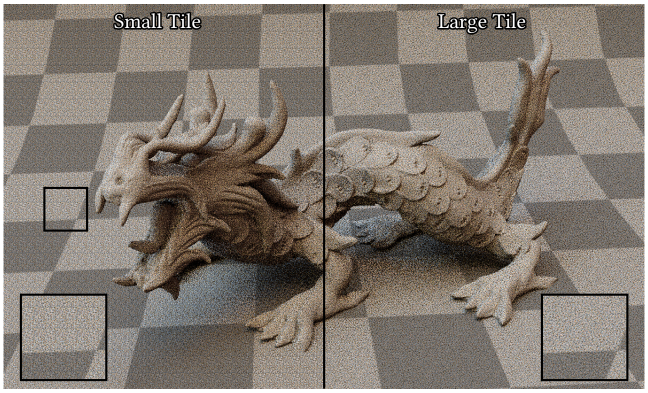


Fig. 1. Rendering with a tile size of 32×32×10 pixels shows noticeable tiling artifacts (left). Increasing the tile size to 128×128×30 pixels avoids such artifacts (right).

REFERENCES

Rafał K Mantiuk, Gyorgy Denes, Alexandre Chapiro, Anton Kaplanyan, Gizem Rufo, Romain Bachy, Trisha Lian, and Anjul Patney. 2021. FovVideoVDP: A visible difference predictor for wide field-of-view video. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–19.

Lois Paulin, Nicolas Bonneel, David Coeurjolly, Jean-Claude Iehl, Antoine Webanck, Mathieu Desbrun, and Victor Ostromoukhov. 2020. Sliced optimal transport sampling. *ACM Trans. Graph.* 39, 4 (2020), 99.

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

Alan Wolfe, Nathan Morrical, Tomas Akenine-Möller, and Ravi Ramamoorthi. 2022. Spatiotemporal Blue Noise Masks. In *Eurographics Symposium on Rendering*. https://doi.org/10.2312/sr.20221161

Table 1. Ablation test for different optimization objectives. On an animation with TAA, we test sample sets optimized for three different temporal kernels: a symetric Gaussian akin to Wolfe et al. [2022], an EMA TAA kernel, and our full model inccorporating the TAA kernel and the perception of [Mantiuk et al. 2021]. We compare the perceived noise (pRelMSE) using Wolfe et al. [2022] as the baseline; lower is better. Lowest error is achieved when the optimization is tailored to the full filter.

Scene	**Uncorrelated**	**Ours Gaussian**	**Ours TAA**	**Ours Mantiuk et al. [2021] + TAA**
Chopper	0.0061	0.0046	0.0046	**0.0043**
Teapot	0.0063	0.0049	0.0048	**0.0044**
Modern Hall	0.0264	0.0242	0.0238	**0.0234**
Living Room	0.0375	0.362	0.0359	**0.0355**
Dragon	0.0043	0.0034	0.0033	**0.0031**
Veach MIS	0.0171	0.0131	0.0133	**0.0126**

Table 2. Perceptual error (pRelMSE) across different scenes. We compare uncorrelated sampling, Salaün et al. [2022], Wolfe et al. [2022] and our method optimized for perceptual temporal filter; lower is better. Comparison are done with and without the TAA filtering. In the upper half of the table, we report the results using common standard deviation of 2.1 for g_s, and in the bottom half we used standard deviation of 1.9 proposed by Wolfe et al. [2022]. Note that in the upper half we report also the results for [Wolfe et al. 2022] with standard deviation proposed by the authors. In both cases, and on every tested scene, our method achieves the lowest perceptual error.

Scene	**Uncorrelated**		**Salaün et al. [2022]**		**Wolfe et al. [2022]**		**Ours perceptual**	
	TAA	no TAA	TAA	no TAA	TAA σ =2.1|1.9	no TAA σ =2.1|1.9	TAA	no TAA
Chopper	0.0088	0.0140	0.0054	0.0087	0.0061 | 0.0058	0.0101 | 0.0096	**0.0043**	**0.0077**
Teapot	0.0079	0.0126	0.0051	0.0079	0.0063 | 0.0050	0.0099 | 0.0082	**0.0044**	**0.0073**
Modern Hall	0.0270	0.0346	0.0244	0.0295	0.0264 | 0.0257	0.0327 | 0.0315	**0.0234**	**0.0288**
Living room	0.0423	0.0561	0.0366	0.0460	0.0375 | 0.0363	0.0480 | 0.0462	**0.0355**	**0.0447**
Dragon	0.0065	0.0107	0.0035	0.0056	0.0043 | 0.0040	0.0072 | 0.0067	**0.0031**	**0.0055**
Veach MIS	0.0174	0.0263	0.0152	0.0218	0.0171 | 0.0168	0.0242 | 0.0238	**0.0126**	**0.0182**
Chopper	0.0094	0.0151	0.0061	0.0098	0.0063	0.0104	**0.0053**	**0.0083**
Teapot	0.0084	0.0137	0.0057	0.0088	0.0055	0.0090	**0.0050**	**0.0076**
Modern Hall	0.0280	0.0365	0.0257	0.0315	0.0266	0.0328	**0.0238**	**0.0292**
Living room	0.0442	0.0589	0.0202	0.0288	0.0208	0.0303	**0.0181**	**0.0249**
Dragon	0.0071	0.0117	0.0039	0.0063	0.0044	0.0073	**0.0036**	**0.0057**
Veach MIS	0.0186	0.0278	0.0164	0.0231	0.0177	0.0250	**0.0137**	**0.0194**