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:

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

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:

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

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_i^{θ} 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.

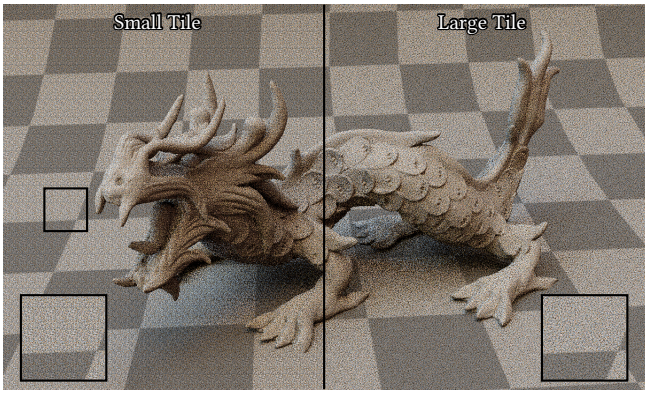

Fig. 1. Rendering with a tile size of 32×32×10 pixels shows noticeable tiling artifacts (left). Increasing the tile size to 128×128×30 pixels avoids such artifacts (right).

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**