James-Stein Gradient Combiner for Inverse Monte Carlo Rendering (Supplemental Report)

JEONGMIN GU, Gwangju Institute of Science and Technology, South Korea
BOCHANG MOON, Gwangju Institute of Science and Technology, South Korea

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

Additional Key Words and Phrases: James-Stein estimator, gradient combiner, inverse Monte Carlo rendering, gradient denoising

ACM Reference Format:
Jeongmin Gu and Bochang Moon. 2025. James-Stein Gradient Combiner for Inverse Monte Carlo Rendering (Supplemental Report). In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers (SIGGRAPH Conference Papers '25), August 10–14, 2025, Vancouver, BC, Canada.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3721238.3730714

Table 1. Relative MSEs of our inference under different values of α (in Eq. 8 of the paper) and various Adam learning rates. The number of samples used was two for the Tray scene and eight for the Plane and Frame scenes.

Scene	Unbiased	Biased	Ours α=0.0	Ours α=0.7	Ours α=0.8	Ours α=0.9
	learning rate = 0.04					
Tray	0.00800	0.00623	0.00445	0.00446	0.00444	**0.00435**
Plane	0.02823	0.01313	0.01296	0.01112	0.01068	**0.00943**
Frame	0.01906	0.00332	0.00925	0.00400	0.00328	**0.00262**
	learning rate = 0.02					
Tray	0.03985	0.00713	0.00433	**0.00430**	0.00432	0.00437
Plane	0.01256	0.01301	0.00896	0.00709	0.00661	**0.00592**
Frame	0.01038	0.00300	0.00367	0.00211	0.00190	**0.00172**
	learning rate = 0.01					
Tray	0.12691	0.00941	0.00666	**0.00648**	0.00657	0.00686
Plane	0.01137	0.01426	0.00726	0.00556	0.00534	**0.00507**
Frame	0.00992	0.00354	0.00191	0.00157	**0.00155**	0.00157

1 Analysis of our method under different learning rates

We use a consistent learning rate of 0.02 for Adam in all BSDF optimization experiments presented in the main paper. In this section, we analyze how the inference accuracy varies when different learning rates are used.

Authors' Contact Information: Jeongmin Gu, jeong755@gm.gist.ac.kr, Gwangju Institute of Science and Technology, Gwangju, South Korea; Bochang Moon, bmoon@gist.ac.kr, Gwangju Institute of Science and Technology, Gwangju, South Korea.

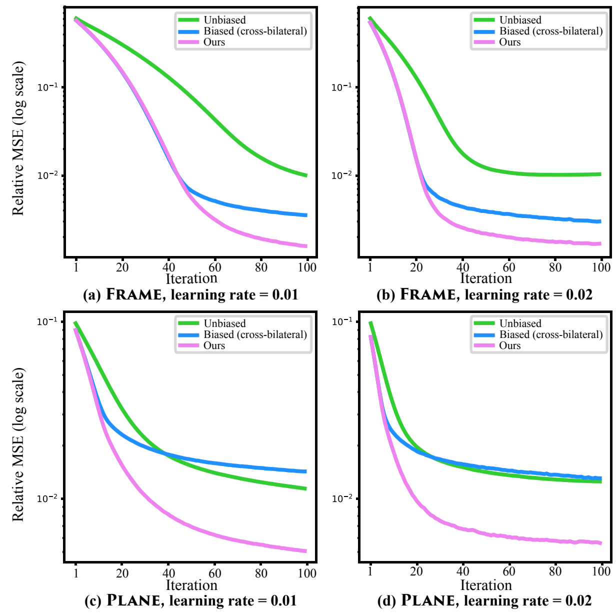

Fig. 1. Relative MSE convergence of scene inference using different gradients (unbiased [Vicini et al. 2021], biased [Chang et al. 2024], and ours) under two learning rates (0.01 and 0.02). We used eight samples for the Plane and Frame scenes.

Analysis of α across different learning rates. Table 1 reports the BSDF inference errors for varying values of α (in Eq. 8 of the paper) and learning rates. The parameter α controls the degree of temporal aggregation of unbiased derivatives in our error estimation (see Sec. 4.2 of the paper). A higher value of α places more weight on temporally accumulated unbiased derivatives, which makes the error estimation less sensitive to noise in the derivatives.

As shown in Table 1, the optimal value of α that minimizes inference error depends on both the Adam learning rate and the specific scene. However, the variation in error across different α remains moderate, except when $\alpha = 0$, which leads to the noisiest error estimates. Based on this observation, we choose $\alpha = 0.9$, as it produces better results than the tested other values in most cases.

Comparison of scene optimization convergence under different learning rates. Fig. 1 shows the relative MSEs of BSDF optimization results obtained using different gradient types under two learning rates of the Adam optimizer. With a relatively high learning rate (0.02), the three tested methods converge quickly to local minima compared to when a lower learning rate (0.01) is used. However, a lower learning rate can be more desirable when a long optimization time (i.e., more iterations) is feasible, as it may continue to reduce

Table 2. Relative MSEs of scene inference results for the Tray scene using 2 and 32 samples per pixel (spp) with different learning rates.

spp	Learning rate	Unbiased	Biased	Ours
2	0.01	0.12691	0.00941	0.00686
	0.02	0.03985	0.00713	0.00437
	0.04	0.00800	0.00623	0.00435
32	0.01	0.00217	0.00883	0.00180
	0.02	0.00211	0.00568	0.00173
	0.04	0.00248	0.00408	0.00209

Table 3. Relative MSEs of our BSDF optimization results using different window sizes.

Scene	Unbiased	Biased	Ours		
			5×5	7×7	9×9
Tray	0.03985	0.00713	**0.00381**	0.00437	0.00482
Plane	0.01256	0.01301	0.00680	0.00592	**0.00551**
Frame	0.01038	0.00300	0.00204	0.00172	**0.00165**

the error over time, unlike the high learning rate. While the choice of learning rate influences convergence behavior, our method consistently achieves better optimization performance than using either unbiased or biased gradients alone for the tested cases.

Analysis of our results with different learning rates and sample counts. Table 2 presents the numerical accuracy of our BSDF optimization results using different Adam learning rates, tested with 2 and 32 samples per pixel (spp), respectively. We used 100 optimization iterations for the Tray scene. The results show that the optimal learning rate depends on the sample count. For instance, a higher learning rate (0.04) yields the lowest error when using 2 spp, whereas a moderate rate (0.02) performs best with 32 spp. While the choice of learning rate affects the optimization outcomes across all tested methods, including ours, combining unbiased and biased gradients consistently achieves lower errors than using either gradient type alone.

2 Analysis of the James-Stein gradient combiner with different window sizes

In the James-Stein gradient combiner, we use a 7×7 window for 2D parameters (e.g., textured BSDFs) to define the spatial neighborhood around each parameter index c in parameter space (see Sec. 4.1 of the main paper). To assess the impact of window size on our inference performance, we report our errors using three different window configurations (5×5, 7×7, and 9×9) in Table 3. While varying the window size influences our inference results, the changes in error are relatively minor compared to the error reduction of our method over the alternatives that rely solely on unbiased or biased gradients.

3 Comparison with variants of the cross-bilateral filter

In the main paper, we use the cross-bilateral filter to generate a biased gradient by denoising unbiased gradients while relying on the original Adam optimizer. As presented in the original paper [Chang et al. 2024], an alternative approach is to apply the cross-bilateral filter to denoise the moment estimates of Adam (post-filtering) rather than denoising unbiased gradients (pre-filtering). Additionally, both pre-filtering and post-filtering can be applied together.

Table 4. Relative MSEs of the optimization results using different configurations of the cross-bilateral filter [Chang et al. 2024]. We evaluate three variants: pre-filtering (pre), which denoises the unbiased gradients; post-filtering (post), which filters the moment estimates in Adam; and a combined version (both), which applies both pre- and post-filtering.

Scene	Unbiased	Biased (pre)	Biased (post)	Biased (both)	Ours (pre)	Ours (both)
Tray	0.03985	0.00713	0.01018	0.00656	**0.00437**	0.00442
Plane	0.01256	0.01301	0.00906	0.01346	0.00592	**0.00524**
Frame	0.01038	0.00300	0.00254	0.00312	0.00172	**0.00155**

For the post-filtering variant, we employ a smaller filtering window than pre-filtering, as the moment estimates (i.e., inputs to post-filtering) are generally less noisy than unbiased gradients (inputs to pre-filtering). Specifically, we reduce the number of iterations of the Á-trous cross-bilateral filter [Chang et al. 2024].

As shown in Table 4, the three filtering variants produce comparable errors. For instance, post-filtering yields the lowest errors for the Plane and Frame scenes but performs worse on the Tray scene compared to the other two configurations.

Moreover, it is worth exploring the integration of post-filtered Adam into our method. Since our approach only modifies the input to the optimizer, the original Adam can be easily replaced with its post-filtered variant. The final column in Table 4 reports results from this configuration, which exploits pre-filtering to generate our biased input and passes our combined gradients to post-filtered Adam. While this setup results in a slightly higher error than our chosen configuration using the original Adam on the Tray scene, it achieves lower errors on the Plane and Frame scenes.

References

Wesley Chang, Xuanda Yang, Yash Belhe, Ravi Ramamoorthi, and Tzu-Mao Li. 2024. Spatiotemporal Bilateral Gradient Filtering for Inverse Rendering. In *SIGGRAPH Asia 2024 Conference Papers (SA '24)*. Association for Computing Machinery, Article 70, 11 pages.

Delio Vicini, Sébastien Speierer, and Wenzel Jakob. 2021. Path replay backpropagation: differentiating light paths using constant memory and linear time. *ACM Trans. Graph.* 40, 4, Article 108 (2021), 14 pages.