Parameter-space ReSTIR for Differentiable and Inverse Rendering: Supplemental Material

Wesley Chang
University of California San Diego
USA

Venkataram Sivaram
University of California San Diego
USA

Derek Nowrouzezahrai
McGill University
Canada

Toshiya Hachisuka
University of Waterloo
Canada

Ravi Ramamoorthi
University of California San Diego
USA

Tzu-Mao Li
University of California San Diego
USA

ABSTRACT

We provide additional ablations and comparisons of our method.

ACM Reference Format:
Wesley Chang, Venkataram Sivaram, Derek Nowrouzezahrai, Toshiya Hachisuka, Ravi Ramamoorthi, and Tzu-Mao Li. 2023. Parameter-space ReSTIR for Differentiable and Inverse Rendering: Supplemental Material. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Proceedings (SIGGRAPH '23 Conference Proceedings), August 6–10, 2023, Los Angeles, CA, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3588432.3591512

1 ABLATION: PGRIS SAMPLE REUSE

Table 1 shows comparisons that demonstrate the effect of reusing both positive and negative samples in both estimators in PGRIS, as discussed in Section 3.2 and Figure 3 in the main text. We compare reusing both samples in both estimators (Both) to reusing each sample in only the estimator with the same sign (Same Sign). Reusing both shows lower error in the reconstructed images after inverse rendering, since no samples are lost due to sign changes.

Scene	Image Reconstruction Error (10^{-2})		
	Mitsuba 3	Ours, Same Sign	Ours, Both
Chalice	6.73	1.20 (0.18x)	1.18 (0.18x)
Tire	15.08	4.45 (0.29x)	4.09 (0.27x)
Ashtray	1.41	0.40 (0.28x)	0.39 (0.28x)
Christmas Tree	1.79	1.74 (0.97x)	1.31 (0.73x)

Table 1: PGRIS Sample Reuse. Comparison of PGRIS that reuses both positive and negative samples in both estimators (Both) and PGRIS that reuses samples only in the estimator with the same sign (Same Sign). Reusing both performs better in all scenes.

2 COMPARISON: EQUAL ERROR

Table 2 shows equal-error comparisons of derivatives computed by both methods after 20 iterations. We use 1 spp for our method and increase the spp of Mitsuba 3 to reach the same error. We report the time taken to compute the derivative pass of a single iteration. Our method at 1 spp is equivalent to Mitsuba 3 at 9-13 spp, and is overall 2-5x faster due to additional overhead of resampling and reuse.

Scene	Mitsuba 3 spp	Single Iteration Time (ms)	
		Mitsuba 3	Ours
Chalice	13	174.2	31.1 (0.18x)
Tire	9	89.7	30.6 (0.34x)
Ashtray	11	91.7	38.4 (0.42x)
Christmas Tree	9	478.7	93.4 (0.19x)

Table 2: Equal Error. Equal error comparison of derivatives computed by Mitsuba 3 and our method after 20 iterations. At 1 spp, our method is equivalent to Mitsuba 3 at 9-13 spp, with 2-5x faster execution time.

3 COMPARISON: CONVERGENCE

In Figure 1, we run inverse rendering until convergence for both methods (300-900s). For all scenes, both methods converge to similar states, with minor differences due to different gradient descent steps. As discussed in the main text, our method shows slightly lower loss in the christmas tree scene, since without reuse, Mitsuba 3 computes extremely sparse gradients.

SIGGRAPH '23 Conference Proceedings, August 6–10, 2023, Los Angeles, CA, USA

ACM ISBN 979-8-4007-0159-7/23/08.
https://doi.org/10.1145/3588432.3591512

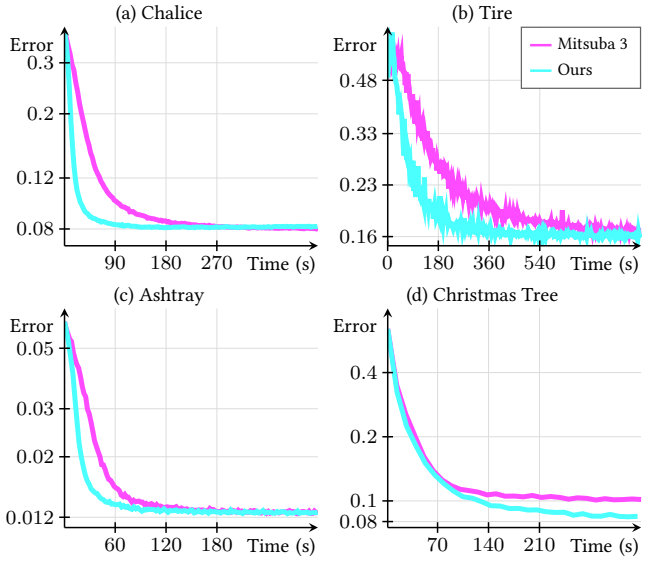

Figure 1: Convergence. Inverse rendering convergence after longer runtime. In most cases, both methods converge to similar results with minor differences.