Correction to the Timings in Neural Layered BRDFs

1 CORRECTION TO THE INFERENCE TIME

In the main paper, we reported a 5 ms inference time for our represnetation network in CUDA implementation on an RTX 2080Ti GPU. However, according to our review and further experiments, we find that the reported time was wrong. In Table 1, we provide the accurate inference time of different batch sizes in our CUDA implementation for the readers' reference.

Table 1: The corrected inference time of our representation network with CUDA kernels for different batch sizes.

Batch size	Inference time (ms)
1024	1
4096	2
8192	4
65,536	23
131,072	45
262,144	88

2 CORRECTION TO THE GPU SHADING TIME

In this section, following the correction in Sec 1, we report the corrected GPU timings for all the scenes we used from the main paper in Table 2. With the corrected timings, most of our conclusions stay true, except the comparison with Belcour [2018], which actually implies *a longer shading time* for our method. However, with equal time, our method still produces lower MSE in Figure 1.

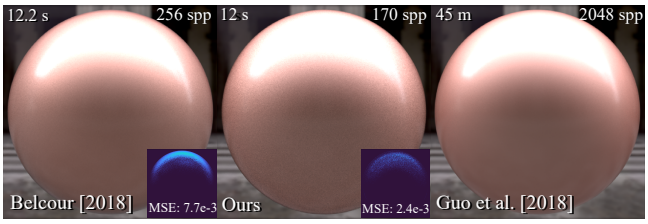

Figure 1: Comparison between our method and Belcour [2018] with equal time. Our method produces a closer match to the reference, even with fewer samples.

For CUDA timings, we pre-compiled multiple CUDA kernels for different batch sizes (as we reported in Table 1). In rendering, we divide all neural layered BRDF pixels into two largest possible batches that can be consumed by those pre-computed kernels. For example, for the 336,519 neural layered BRDF pixels in the Still Life scene, they will be rendered in two batches: 262,144 and 131,072. At last, the final GPU time will be the sum of these two batches.

Table 2: The corrected GPU rendering time for all scenes in the main paper. Note that all CPU timings were correct, and our conclusions in comparisons to Guo et al. [2018] still hold true.

Scene	Resolution	#Neural pixels per spp	GPU Time per spp (ms)
Still Life	1024×1024	336,519	113
Ball	512×512	187,928	68
Shoe	888×500	68,591	25
Globe	1024×1024	262,842	89
Teapot	720×480	69,680	27

REFERENCES

Laurent Belcour. 2018. Efficient Rendering of Layered Materials Using an Atomic Decomposition with Statistical Operators. *ACM Trans. Graph.* 37, 4, Article 73 (July 2018), 15 pages.

Yu Guo, Miloš Hašan, and Shuang Zhao. 2018. Position-Free Monte Carlo Simulation for Arbitrary Layered BSDFs. *ACM Trans. Graph.* 37, 6, Article 279 (Dec. 2018), 14 pages.

SIGGRAPH '22 Conference Proceedings, August 7–11, 2022, Vancouver, BC, Canada

ACM ISBN 978-1-4503-9337-9/22/08…$15.00
https://doi.org/10.1145/3528233.3530732