LapisGS: Layered Progressive 3D Gaussian Splatting for Adaptive Streaming

Supplementary Material

6. View-Adaptive Rendering

Distance-aware foveated rendering is a technique that optimizes rendering performance by adjusting the LOD in a 3D scene based on the viewer's gaze and the distance to objects. This method focuses computational resources on rendering high-quality images in the viewer's direct line of sight (foveal region), while reducing detail in peripheral areas, as shown in Fig. 8.

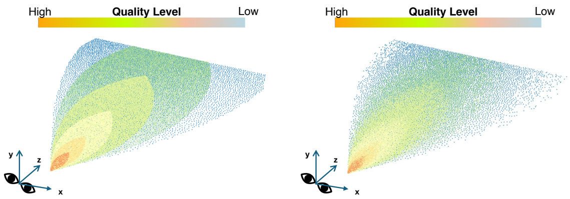
Space-based hierarchical structures commonly used in 3D rendering present challenges due to their discrete nature [9, 12, 22]. The structure often results in chunk-based representation, which can lead to inefficient alignment with arbitrarily oriented scene elements such as walls, pillars, or stairs. This misalignment causes the detailed rendering of the scene center to be less effective [23]. Although smooth transitions can be achieved within individual chunks, the chunk-wise rendering approach leads to popping artifacts, resulting in a fundamentally discrete quality transition, as shown in Fig. 8a.

Our proposed layered structure addresses these limitations by using a cumulative stacking approach, where layers progressively merge to form higher-quality levels. This architecture allows for the sharing of visual information across different layers, making it naturally suitable for adaptive quality rendering. Unlike the chunk-wise approach of tree structures, our method evaluates and adjusts the opacity of Gaussian splats on a splat-wise basis with Eq. (4), enabling smoother, continuous transitions. As illustrated in Fig. 8b, our method supports view-adaptive rendering, providing seamless quality transitions and enhanced visual fidelity. We also show an example of view-adaptive rendering from our model in Fig. 9.

7. Qualitative Results

We show the qualitative results of LapisGS alongside comparison methods on a variety of scenes, including *Drjohnson* and *Playroom* in Deep Blending dataset [8], *Train* and *Truck* in Tank&Temples dataset [13], and *Room* and *Treehill* in Mip-NeRF360 dataset [2], as shown in Figs. 10 to 15.

As observed, LapisGS demonstrates superior performance in preserving intricate scene details while eliminating common rendering artifacts across various environments. Our method matches the visual quality of the Multiscale approach, which is the upper bound of reconstruction quality, while achieving a substantially reduced computational footprint.



(a) Discrete quality adaptation. (b) Continuous quality adaptation.

Figure 8. Illustration of discrete and continuous quality adaptation.

(a) Ground truth rendering. (b) Continuous adaptive rendering.

Figure 9. The sample rendering results of *Garden*.

8. Per-Scene Quantitative and Qualitative Ablation Study

To provide a comprehensive comparison of the effectiveness of our dynamic opacity optimization both qualitatively and quantitatively, we present per-scene results and rendering samples for our method and the ablation model in Tabs. 6 to 9 and Figs. 16 to 19, respectively. The ablation model is designed according to the experimental setup described in the main paper, where all parameters of prior layers, including opacity, are kept fixed during the training of enhancement layers.

This comparison reveals that without dynamic opacity optimization, the model becomes significantly more redundant as additional splats are required to compensate for deficiencies in the lower layers. This results in larger model size and a decline in visual quality, particularly in areas that demand high-frequency detail. In contrast, our method dynamically refines the representation by adjusting the opacity of lower layers, ensuring that only essential splats contribute to the final output. This approach not only reduces model size but also enhances visual fidelity, especially in complex scenes. The results underscore the efficiency and scalability of our method across various datasets.

Figure 10. Sample renderings of *Drjohnson* from Deep Blending dataset[8] at different scales.

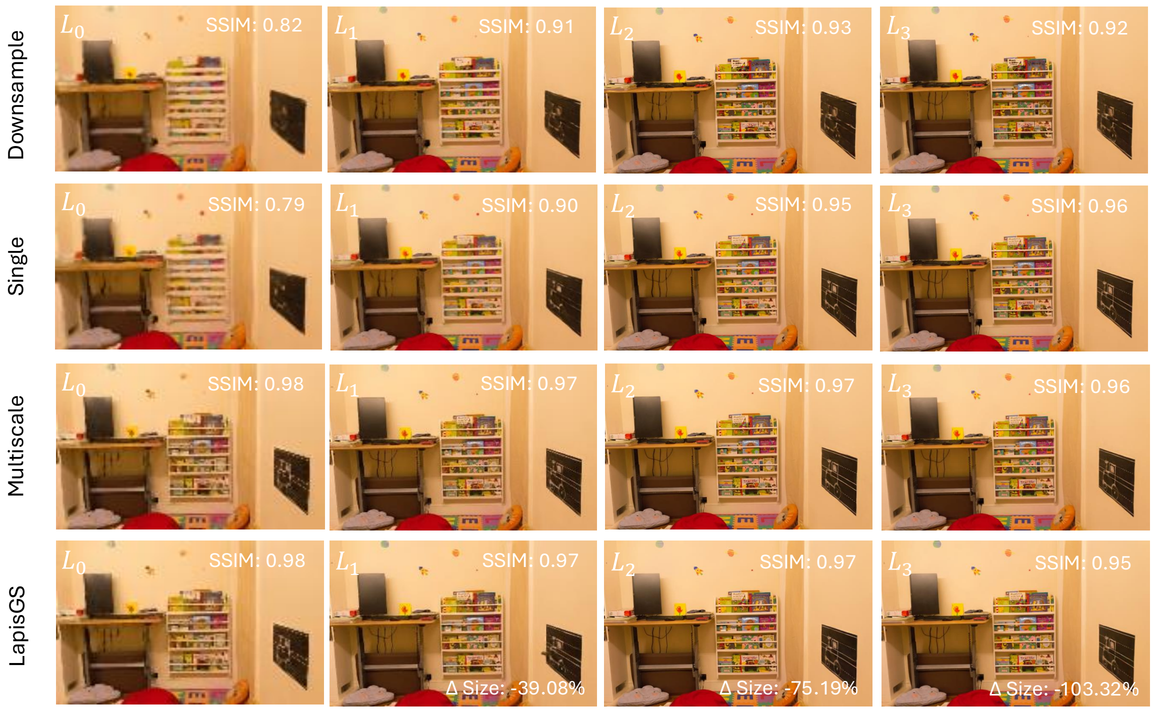


Figure 11. Sample renderings of *Playroom* from Deep Blending dataset[8] at different scales.

Figure 12. Sample renderings of *Train* from Tank&Temples dataset[13] at different scales.

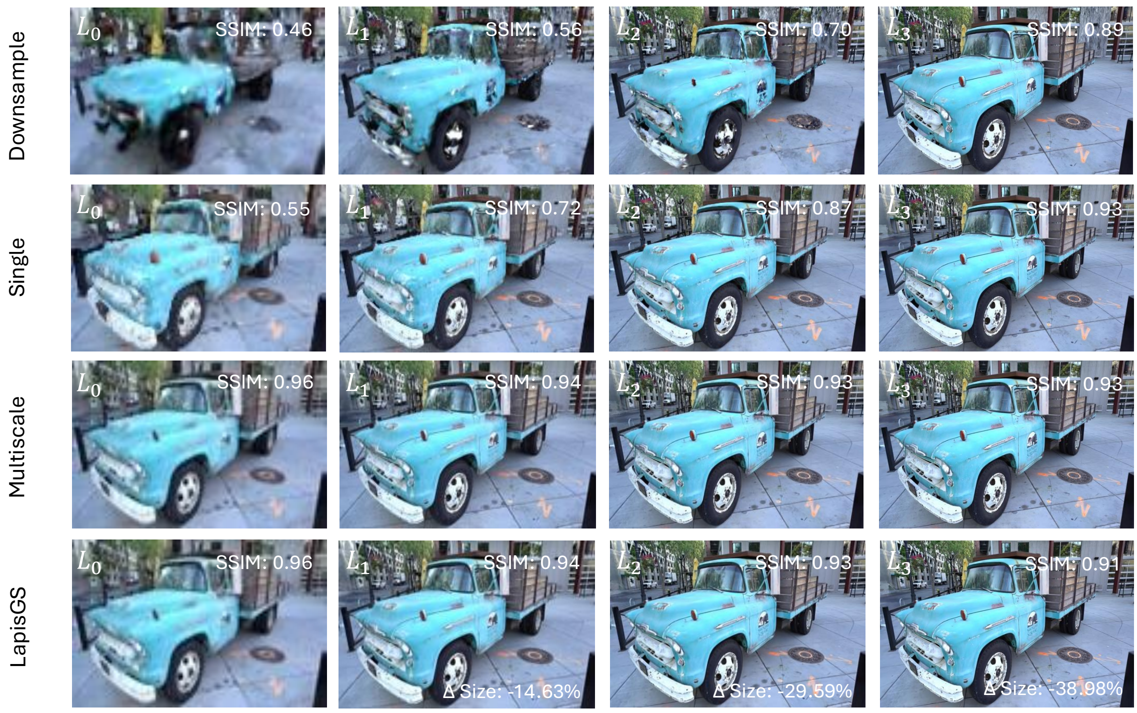


Figure 13. Sample renderings of *Truck* from Tank&Temples dataset[13] at different scales.

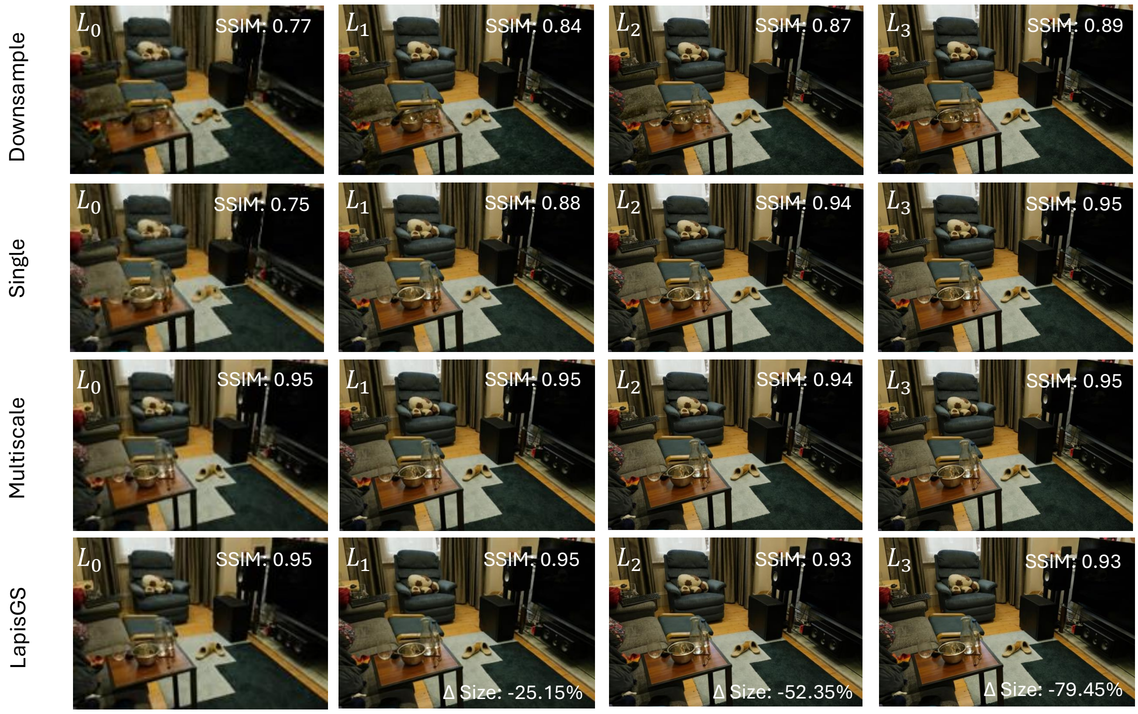


Figure 14. Sample renderings of *Room* from Mip-NeRF360 dataset [2] at different scales.

Figure 15. Sample renderings of *Treehill* from Mip-NeRF360 dataset [2] at different scales.

Table 6. Per-scene size and quality difference in percentage of LapisGS on synthetic Blender dataset [18] at different quality levels, compared to the Freeze method.

Scene	L_1			L_2			L_3		
	Δ SSIM ↑	Δ LPIPS ↓	Δ Size ↓	Δ SSIM ↑	Δ LPIPS ↓	Δ Size ↓	Δ SSIM ↑	Δ LPIPS ↓	Δ Size ↓
Lego	0.21%	-16.43%	-113.78%	0.65%	-40.72%	-121.51%	1.19%	43.6%	-232.51%
Hotdog	0.25%	-7.87%	-128.50%	0.31%	-9.39%	-62.31%	0.18%	8.85%	-87.35%
Ship	0.32%	-7.67%	-68.57%	0.75%	-10.25%	-106.61%	3.88%	6.74%	-146.29%
Materials	0.35%	-28.27%	-96.70%	0.84%	-39.19%	-135.39%	1.30%	29.62%	-233.48%
Ficus	0.62%	-59.32%	-58.96%	1.59%	-100.36%	-96.93%	2.59%	108.65%	-126.38%
Mic	0.12%	-34.28%	-59.83%	0.18%	-36.87%	-68.31%	0.32%	42.37%	-142.97%
Chair	0.10%	-23.60%	-135.32%	0.28%	-31.62%	-110.90%	0.61%	28.68%	-103.74%
Drums	0.42%	-13.39%	-77.88%	1.06%	-27.87%	-96.12%	1.65%	32.42%	-144.60%

Table 7. Per-scene size and quality difference in percentage of LapisGS on Mip-NeRF360 dataset [2] at different quality levels, compared to the Freeze method.

Scene	L_1			L_2			L_3		
	Δ SSIM ↑	Δ LPIPS ↓	Δ Size ↓	Δ SSIM ↑	Δ LPIPS ↓	Δ Size ↓	Δ SSIM ↑	Δ LPIPS ↓	Δ Size ↓
Treehill	0.93%	-10.51%	-72.26%	2.53%	-21.85%	-128.95%	17.78%	-50.71%	-124.24%
Room	1.24%	-17.90%	-95.75%	2.94%	-30.95%	-180.00%	7.42%	-50.87%	-265.26%
Bonsai	0.47%	-3.20%	-109.87%	0.86%	-17.66%	-247.38%	1.16%	-27.92%	-400.09%
Counter	0.71%	-3.57%	-103.37%	1.52%	-11.85%	-208.83%	2.12%	-24.54%	-332.97%
Kitchen	0.97%	-7.92%	-88.17%	1.98%	-19.65%	-181.91%	1.97%	-16.39%	-256.25%
Flowers	1.64%	-11.20%	-79.76%	10.22%	-20.21%	-139.81%	26.00%	-36.98%	-154.45%
Garden	0.58%	-5.74%	-87.92%	1.73%	-13.61%	-152.35%	12.22%	-56.77%	-195.91%

Table 8. Per-scene size and quality difference in percentage of LapisGS on Deep Blending dataset [8] at different quality levels, compared to the Freeze method.

Scene	L_1			L_2			L_3		
	Δ SSIM ↑	Δ LPIPS ↓	Δ Size ↓	Δ SSIM ↑	Δ LPIPS ↓	Δ Size ↓	Δ SSIM ↑	Δ LPIPS ↓	Δ Size ↓
Playroom	0.51%	3.15%	-126.67%	0.90%	-10.87%	-272.72%	1.29%	-14.90%	-378.33%
Drjohnson	-0.75%	7.00%	-95.71%	3.04%	-21.03%	-168.14%	5.35%	-27.28%	-240.81%

Table 9. Per-scene size and quality difference in percentage of LapisGS on Tank&Temples dataset [13] at different quality levels, compared to the Freeze method.

Scene	L_1			L_2			L_3		
	Δ SSIM ↑	Δ LPIPS ↓	Δ Size ↓	Δ SSIM ↑	Δ LPIPS ↓	Δ Size ↓	Δ SSIM ↑	Δ LPIPS ↓	Δ Size ↓
Train	1.61%	-15.55%	-74.58%	6.05%	-26.54%	-116.69%	11.00%	-29.56%	-145.92%
Truck	1.75%	-25.31%	-75.40%	3.74%	-68.70%	-144.47%	5.90%	-38.85%	-164.20%

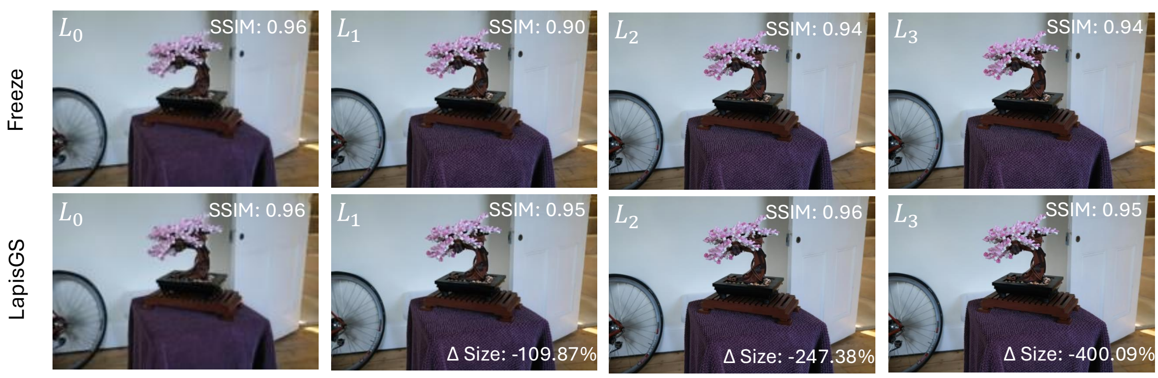


Figure 16. Sample renderings of *Bonsai* from Mip-NeRF360 dataset[2] at different scales.

Figure 17. Sample renderings of *Counter* from Mip-NeRF360 dataset[2] at different scales.

Figure 18. Sample renderings of *Drjohnson* from Deep Blending dataset[8] at different scales.

Figure 19. Sample renderings of *Truck* from Tank&Temples dataset[13] at different scales.