Supplementary Materials for "CityGaussian: Real-time High-quality Large-Scale Scene Rendering with Gaussians"

Yang Liu[1,2], Chuanchen Luo[4], Lue Fan[1,2], Naiyan Wang[5], Junran Peng[6] ✉, and Zhaoxiang Zhang[1,2,3] ✉

[1] NLPR, MAIS, Institute of Automation, Chinese Academy of Sciences
[2] University of Chinese Academy of Sciences
[3]Centre for Artificial Intelligence and Robotic [4]Shandong University
[5]TuSimple [6]University of Science and Technology Beijing
{liuyang2022, lue.fan, zhaoxiang.zhang}@ia.ac.cn,
chuanchen.luo@sdu.edu.cn, winsty@gmail.com, jrpeng4ever@126.com

A Additional Experimental Results

A.1 Additional Quantitative Comparison

The quantitative comparisons across datasets *MatrixCity*, *Residence*, *Rubble*, *Building*, and *Sci-Art* are presented in Table S1. We not only provide the performance of the CityGS with no LoD as done in Sec. 4.2 of our main paper, but also presents the standard CityGS for reference. It can be observed that our approach outperforms others in terms of **SSIM** and **LPIPS** among all the datasets, and achieves the highest **PSNR** on *MatrixCity*, *Rubble*, and *Building*. The relatively weaker **PSNR** of *Sci-Art* and *Residence* is mainly attributed to the appearance variations across views in these dataset. We leave solving this issue for future works. Thanks to the superior efficiency of 3DGS, we have achieved much faster speed than previous state-of-the-art even without LoD.

Table S1: Quantitative Comparison on five large-scale scene datasets. The '-' symbol indicates Mega-NeRF [2] and Switch-NeRF [4] were not evaluated on *MatrixCity* due to difficulties in adjusting its training configurations beyond the provided, resulting in poor performance on this dataset. The best results of each metric are in **bold**.

	MatrixCity				Residence				Rubble				Building				Sci-Art			
Metrics	SSIM↑	PSNR↑	LPIPS↓	FPS↑	SSIM↑	PSNR↑	LPIPS↓	FPS↑	SSIM↑	PSNR↑	LPIPS↓	FPS↑	SSIM↑	PSNR↑	LPIPS↓	FPS↑	SSIM↑	PSNR↑	LPIPS↓	FPS↑
MegaNeRF [2]	-	-	-	-	0.628	22.08	0.489	<0.1	0.553	24.06	0.516	<0.1	0.547	20.93	0.504	<0.1	0.770	25.60	0.390	<0.1
Switch-NeRF [4]	-	-	-	-	0.654	**22.57**	0.457	<0.1	0.562	24.31	0.496	<0.1	0.579	21.54	0.474	<0.1	0.795	**26.61**	0.360	<0.1
GP-NeRF [3]	0.611	23.56	0.630	0.15	0.661	22.31	0.448	0.31	0.565	24.06	0.496	0.40	0.566	21.03	0.486	0.42	0.783	25.37	0.373	0.34
3DGS† [1]	0.735	23.67	0.384	35.9	0.791	21.44	0.236	**62.1**	0.777	25.47	0.277	47.8	0.720	20.46	0.305	**45.0**	0.830	21.05	0.242	**72.2**
CityGS(no LoD)	**0.865**	**27.46**	**0.204**	21.6	**0.813**	22.00	**0.211**	32.7	**0.813**	**25.77**	**0.228**	43.9	**0.778**	21.55	**0.246**	24.3	**0.837**	21.39	**0.230**	56.1
CityGS	0.855	27.32	0.229	**53.7**	0.805	21.90	0.217	41.6	0.785	24.90	0.256	**52.6**	0.764	**21.67**	0.262	37.4	0.833	21.34	0.232	64.6

Besides, as shown in Table S1, LoD significantly improves efficiency, especially for extremely large-scale scenes such as *MatrixCity*. Compared with our CityGS, the 3DGS† [1] possesses faster speed but significantly lower rendering

quality. The main reason is that the original 3DGS requires sufficiently large iterations and memory to optimize the whole scene with thousands of images. Bounded by computation resources, the capacity of the trained original 3DGS is too limited to represent the whole large-scale scene well.

A.2 Additional Ablations

We also explored the influence of hyper-parameters in training, namely block number and data assignment threshold ε mentioned in Sec. 3.2 of main paper. Here, we control the overall finetuning iterations to be 9×32000. As shown in Table S2, as the block number grows, the average data assigned to blocks decreases. We achieve optimal results around 3×3 partitions. The ε also controls data assignment. As ε grows, the average poses assigned decreases. And if it is too high, many necessary training data will be lost, thus leading to lower PSNR performance.

Table S2: Ablation on block numbers and SSIM threshold ε. The experiment is conducted on the *Rubble* dataset. The first row is the performance of coarse global Gaussians prior mentioned in Sec. 3.2 of main paper, and thus has no block number or ε setting. MEAN denotes an average number of assigned training poses among divided blocks. The best performances are in **bold**.

ε	#Blocks	SSIM↑	PSNR↑	LPIPS↓	MEAN
0.1	2×2	0.762	24.48	0.285	835
0.1	3×3	0.806	25.41	0.238	552
0.1	4×4	0.804	25.45	0.241	432
0.12	3×3	**0.813**	**25.77**	0.231	469
0.14	3×3	0.812	25.43	**0.229**	415

Table S3: Detailed Parameter Setting. In **training**, the foreground area for contraction is bounded by $p_{\min} = (x_{\min}, y_{\min}, z_{\min})$ and $p_{\max} = (x_{\max}, y_{\max}, z_{\max})$. Here we take the height dimension as z. From the bird's eye view, the longest side is set as the x-axis, while the shortest is the y-axis. z bound is set as the minimum and maximum position of all Gaussians, and thus not included. The block dimension along the x-axis and y-axis is denoted as #Blocks, and ε is SSIM threshold. In **rendering**, the Distance Interval decides detail level assignment.

Dataset	$x_{\min}(m)$	$y_{\min}(m)$	$x_{\max}(m)$	$y_{\max}(m)$	#Blocks	ε	Distance Interval (m)
MatrixCity	-350	-400	450	200	6×6	0.05	[0,200],[200,400],[400,∞]
Rubble	-50	-5	50	-135	3×3	0.12	[0,100],[100,200],[200,∞]
Building	-140	250	-10	0	5×4	0.1	[0,100],[100,200],[200,∞]
Residence	-270	-25	60	175	5×4	0.08	[0,250],[250,500],[500,∞]
Sci-Art	-205	-110	90	55	3×3	0.05	[0,250],[250,500],[500,∞]

B Detailed Parameter Setting

We present the specific hyper-parameter configurations for each dataset in Table S3. The roles of training parameters are detailed in Sec. 3.2 of the main paper, while the role of rendering parameter Distance Interval is specified in Sec. 3.3 of the main paper. Note that for LoD on datasets except for *MatrixCity*, we use three detail levels of compression rate 60%, 50%, and 40%.

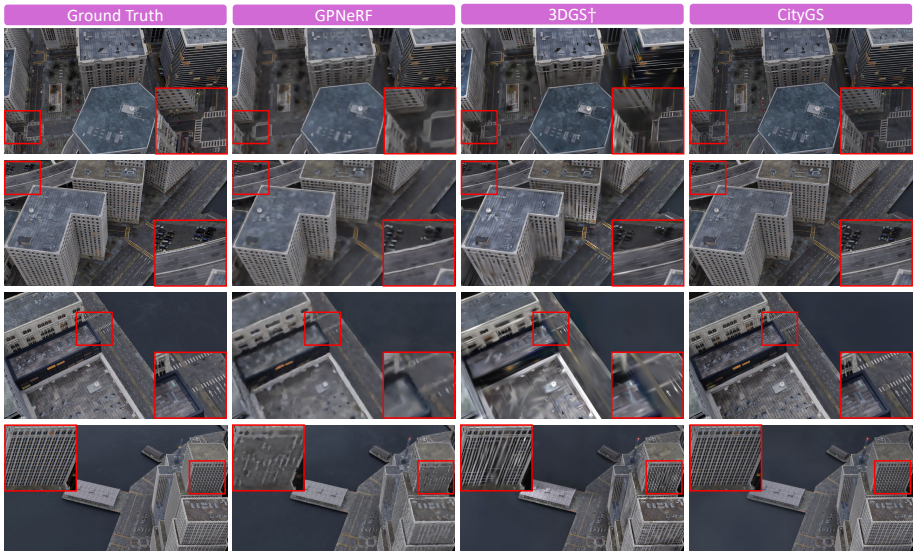

Fig. S1: More qualitative comparison with SOTA methods on *MatrixCity* dataset.

C More Visualization on MatrixCity Dataset

In this section, we provide additional qualitative comparisons using the *MatrixCity* dataset, depicted in Fig. S1. Our results showcase the superior reconstruction quality of intricate details, including crowded cars and crosswalks. The remarkable enhancement in visual fidelity compared with other methods sufficiently illustrates the superiority of our CityGS.

D Qualitative Validation on Concatenated Fusion

To validate the effectiveness of the concatenated fusion strategy, we perform rendering at the viewpoints where the visible area spans multiple blocks. The Gaussians utilized here come from direct concatenation of fine-tuned Gaussians

of corresponding blocks. As depicted in Fig. S2, rendering from a specific viewpoint may involve four or more blocks. Despite that, the rendered images exhibit no discernible discontinuities, showcasing smooth boundary transitions facilitated by our coarse global Gaussian prior, as discussed in Sec. 3.2 of our main paper.

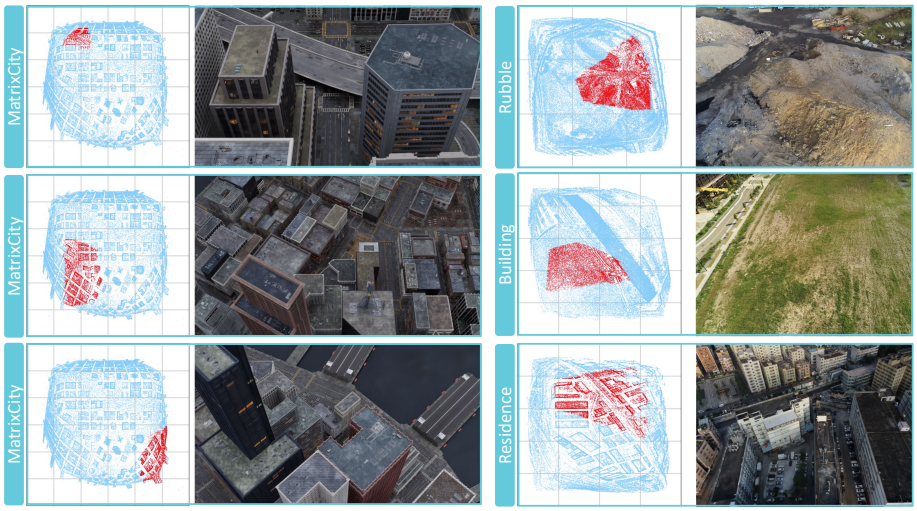

Fig. S2: Qualitative validation of boundary continuity on both synthetic and real datasets when visible Gaussians across multiple blocks. Each subfigure illustrates point distribution under contracted space on the left and rendered image on the right. For the point distribution, blue points denote overall Gaussians, while the red points denote visible Gaussians. The grey grid depicts block partition under contracted space.

E Scene Manipulation

For the implicit representation of NeRF-based methods, it is hard to explain the correspondence between network parameters and scene structure. However, since we can reconstruct the explicit city representation with relatively high geometric precision in CityGS, the geometric and appearance distribution can be manipulated as desired. The demos are shown in Fig. S3. The appearance of a specified part of a building can be transformed to the desired style. It is also possible to delete a building and replace it with another one. By placing cars or pedestrians, the pre-defined traffic conditions can be simulated in the city. These demos indicate potential real-time and interactive application of CityGS.

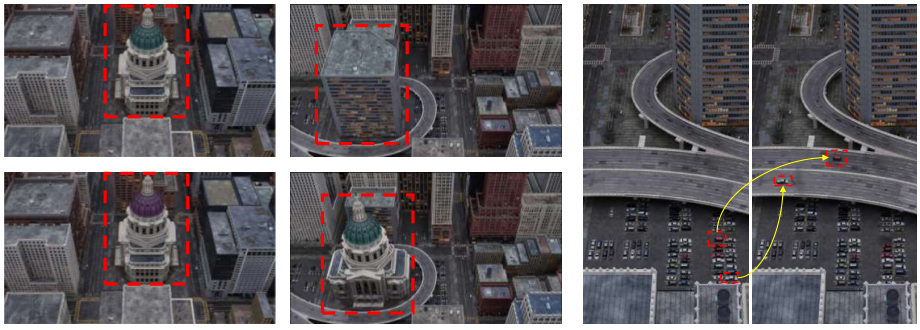

Fig. S3: Illustration of city scene manipulation driven by explicit representation of CityGS. In Part (a), the dome of the original building in the first row is repainted to the desired color shown in the second row. In Part (b), the building of the first row is removed and replaced with the one shown in the second row. In Part (c), the cars parked at locations shown in the left image are moved to the positions shown in the right image, so as to simulate the required traffic conditions. NeRF-based methods struggle to realize such manipulation.

References

1. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics **42**(4) (2023)
2. Turki, H., Ramanan, D., Satyanarayanan, M.: Mega-nerf: Scalable construction of large-scale nerfs for virtual fly-throughs. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12922–12931 (2022)
3. Xu, L., Xiangli, Y., Peng, S., Pan, X., Zhao, N., Theobalt, C., Dai, B., Lin, D.: Grid-guided neural radiance fields for large urban scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8296–8306 (2023)
4. Zhenxing, M., Xu, D.: Switch-nerf: Learning scene decomposition with mixture of experts for large-scale neural radiance fields. In: The Eleventh International Conference on Learning Representations (2022)