HZB-RPU: A 168fps 0.396frame/mJ End-to-End Pipeline 3D Gaussian Splatting Processor for Real-time and Efficient 3D Scene Rendering

Xi Chen[1,2], Wentao Zheng[1], Tianhui Jiao[1], Yuhui Shi[1], Yingxuan Zhou[1],
Shaochen Li[1], Xiaodong Li[1], Jun Yang[1,2], Xin Si[1]
[1] Southeast University, Nanjing, China
[2] National Center of Technology Innovation for EDA, Nanjing, China
Email: {xichen, xinsi}@seu.edu.cn

***Abstract*—This paper presents HZB-RPU, an end-to-end hot-zone based render processing unit tailored for edge 3D gaussian splatting (3DGS) acceleration. The proposed hot-zone based rendering (HZB-R) technique eliminates irregular external memory access (EMA) and reduces on-chip cache requirements by 99.43%. Furthermore, the multi-level skipping (MLS) and intra-tile assisting (ITA) schemes are introduced to reduce 48% redundant computations and improve processing element (PE) utilization by 1.73×. The effectiveness of HZB-RPU is verified in silicon across various rendering tasks. It demonstrates state-of-the-art performance, achieving 1.18× higher throughput, 4.37× better area efficiency and 1.72× higher energy efficiency while maintaining comparable rendering accuracy.**

***Index Terms*—3D scene rendering, 3D gaussian splatting, real-time, efficient, accelerator**

I. Introduction

The rapid advancement of artificial intelligence (AI) and immersive technologies has significantly increased the demand for real-time, high-quality 3D scene rendering. Recently, 3D gaussian splatting (3DGS) has been emerged as a state-of-the-art 3D scene representation technique [6]–[10], proving critical for applications with substantial market potential such as world models, VR/AR, and robotics. However, deploying 3DGS on resource-constrained edge devices poses significant hardware challenges [1]–[5], as illustrated in (Fig. 1&2): 1) Inconsistency between storage order and rendering order (RO): the vanilla algorithm generates tile-depth pair (TDP) lists by traversing DRAM in storage order, which necessitates huge on-chip SRAM cache. During the rendering stage, 2D gaussian projections (2D-GPs) are fetched in RO, resulting in massive irregular external memory access (EMA). 2) Redundant spatiotemporal computations: a significant number of computations are inefficient both spatially (for tiles and pixels outside the influence range of a 2D-GP) and temporally (due to high frame-to-frame similarity). 3) Irregular early-stop phenomenon: the inconsistent per pixel rendering completion time, further exacerbated by various skipping schemes, leads to severe underutilization of processing element (PE) arrays.

II. Hot-Zone Based Render Processing Unit

HZB-RPU comprehensively tackles the above challenges with three key features: 1) a hot-zone based rendering (HZB-R) to eliminate irregular EMA and reduce on-chip cache. 2) a multi-level skipping (MLS) to reduce redundant computations. 3) an intra-tile assisting (ITA) to sustain high PE utilization.

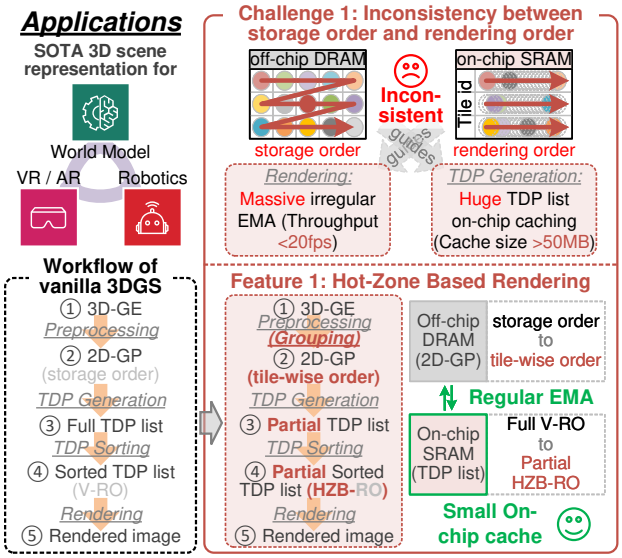


Fig. 1: Applications and workflow of vanilla 3DGS, the 1st challenge and Feature 1.

Fig. 3 introduces the concept of the proposed HZB-R. The hot-zone of a specific tile and the corresponding hot-zone 2D-GP are defined in DEF.1 and DEF.2. During the preprocessing stage, the RPU computes 2D-GPs and stores them into DRAM in a tile-wise order (Grouping with dynamic address mapping, DAM). During the sorting and rendering stage, the RPU loads hot-zone 2D-GPs (regular EMA) and generate a hot-zone based rendering order (HZB-RO), avoiding the costly caching of the full TDP list. Compared to the vanilla RO (V-RO), HZB-RO on average calculates 12% less 2D-GPs, with a PSNR decrease of only 0.082dB, while achieving a 99.43% reduction in on-chip cache and a 13.13× throughput enhancement. Fig. 4 shows the overall architecture of the proposed HZB-RPU. It comprises a top controller, an interconnect network, a DRAM interface, global memory, a preprocessor, a sorter, and a renderer.

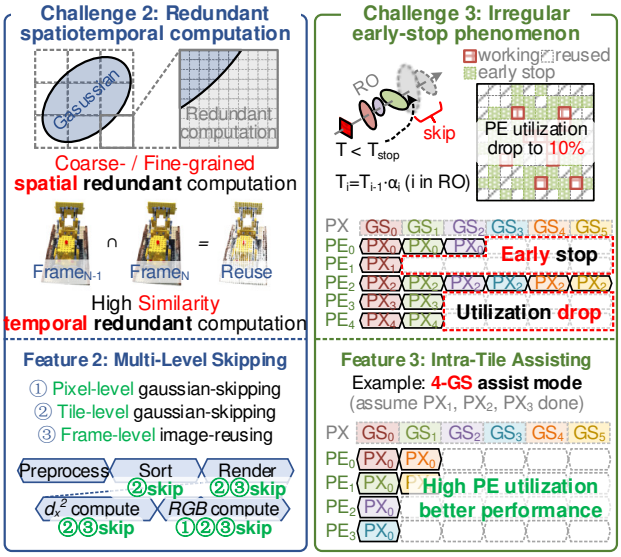


Fig. 2: 2nd and 3rd challenges and Feature 2 and Feature 3.

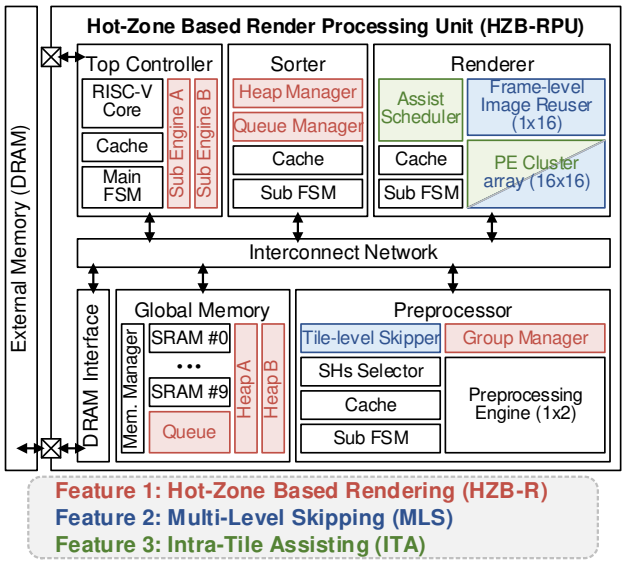


Fig. 4: Overall architecture of the HZB-RPU.

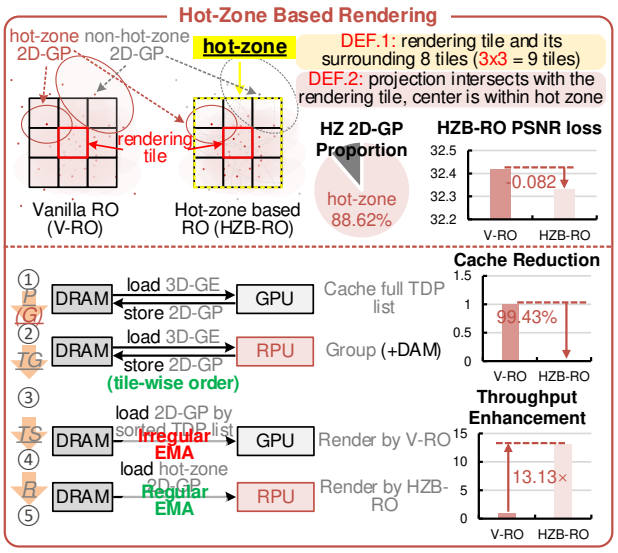


Fig. 3: The concept of the hot-zone based rendering.

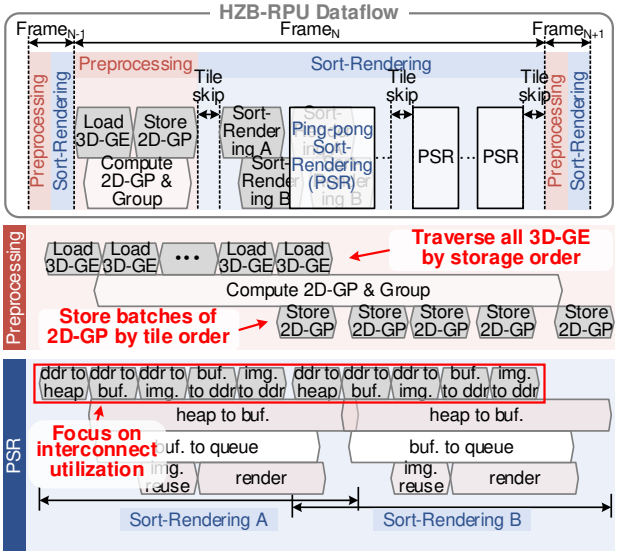


Fig. 5: HZB-RPU dataflow.

Fig. 5&6 presents the proposed HZB-RPU dataflow and detailed implementation and dataflow of ping-pong sort-rendering system. The processing of each frame is divided into two stages: the preprocessing stage and the sort-rendering stage. (1) Preprocessing stage: This stage involves (a) loading and converting 3D gaussian ellipsoids (3D-GEs) into 2D-GPs; and (b) grouping these 2D-GPs based on the tiles where the centers of the 2D-GPs are located, and then storing them in tile order to external memory. (2) Sort-rendering stage: This stage performs (a) loading (via regular EMA, enabled by the group operation in the preprocessing stage) and sorting the grouped/sorted 2D-GPs within the hot-zone; and (b) rendering with the HZB-RO. Given the sorter's moderate output bandwidth requirement and principle of minimizing hardware cost and first-output latency, heapsort is selected as the foundational algorithm for the sorter. Heap in and heap out are required to obtain the HZB-RO from grouped 2D-GPs (tile#8) and sorted 2D-GPs (tile#0-7). To maximum the utilization of bandwidth, a ping-pong sort-rendering (PSR) flow is proposed by allowing the sort-rendering of the next tile (sort-rendering B) to commence before the current tile (sort-rendering A) completes, thereby further boosting the system performance.

Fig. 7 demonstrates the MLS to refrain from redundant spa-

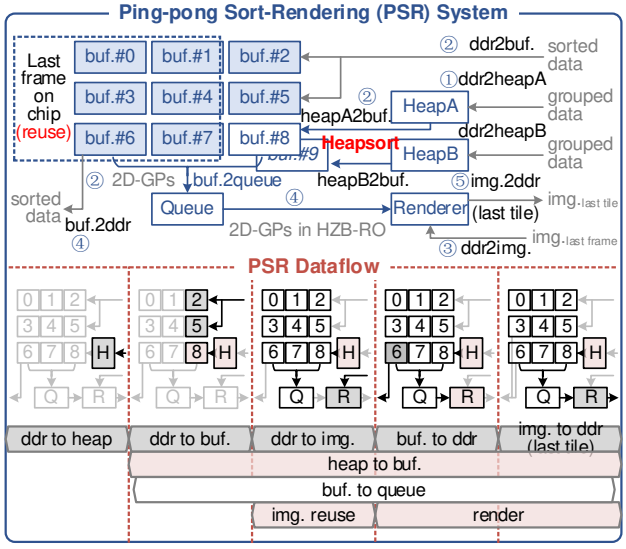


Fig. 6: Detailed implementation and dataflow of ping-pong sort rendering system.

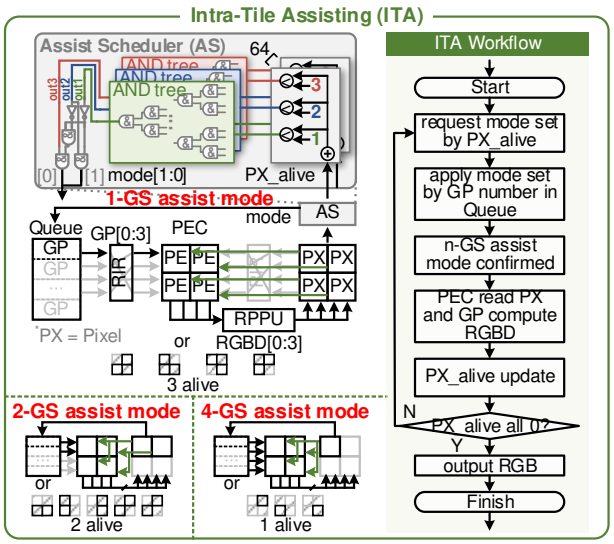


Fig. 8: Implementation and workflow of intra-tile assisting.

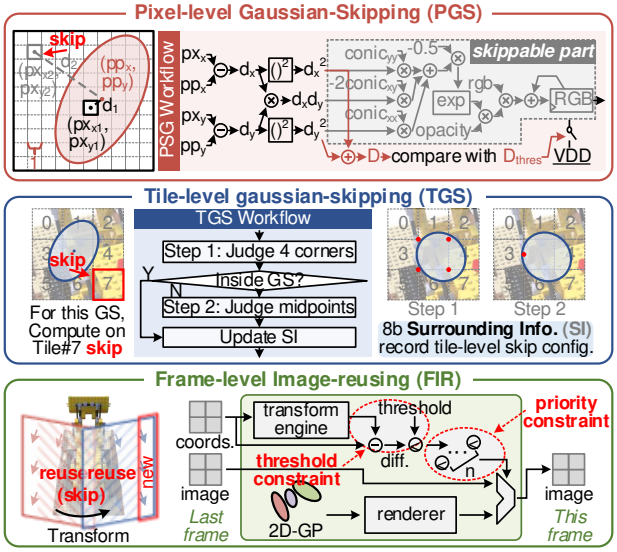


Fig. 7: Implementation and workflow of multi-level skipping.

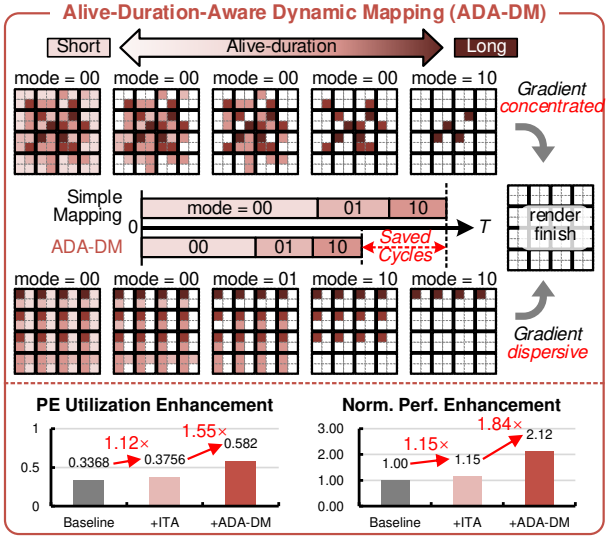
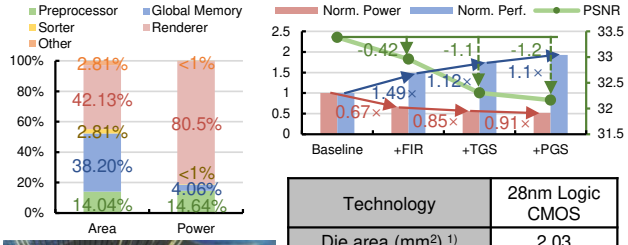
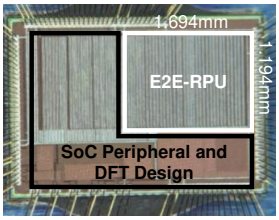


Fig. 9: Alive-Duration-Aware Dynamic Mapping and effectiveness of ITA and ADA-DM.

tiotemporal computations. The pixel-level gaussian-skipping (PGS) helps skip the fine-grained spatial redundant computations by comparing an intermediate variable D against a configurable threshold Dthres to determine whether the skippable part can be skipped. The tile-level gaussian-skipping (TLS) helps skip the coarse-grained spatial redundant computations. Each 2D-GP contains an 8-bit surrounding information (SI) which is obtained by a two-step process to describe whether the compute of this 2D-GP on corresponding tiles can be skipped. The frame-level image-reusing (FIR) helps skip the computations of pixels that are highly similar to the last frame. Two constraints (threshold and priority constraint) are proposed to decide reuse eligibility. Fig. 8&9 describes the intra-tile assisting (ITA) and alive-duration-aware dynamic mapping (ADA-DM). ITA improves PE utilization and accelerates the rendering process. Each PE in the PE cluster (PEC) tracks its pixel's completion status, while an assist scheduler (AS) determines the assist mode based on completion status across all PECs. Since the PECs are fixed in hardware, to address scenarios where fixed PEC mapping limit ITA activation, ADA-DM dynamically determines the mapping between PEs and pixels (PX) based on their alive-durations (transparency)





Technology	28nm Logic CMOS
Die area (mm²) [1]	2.03
Data precision	FP16
On-chip SRAM (kB)	283.5
Supply Voltage (V)	0.7 - 0.9
Max frequency (MHz)	200
PSNR (dB) [2]	32.166
Throughput (fps) [3]	168
Power (mW)	105.92
Norm. Area efficiency (fps/mm²) [4]	30.65
Norm. Energy efficiency (frame/mJ) [4]	0.396

[1] Preprocessor excluded. 8×8 PEC array in renderer.
[2] Measured under NeRF synthetic dataset.
[3] Measured under 0.9V, 200MHz.
[4] Normalized to 16×16 PEC for 800×800 pixels per frame. Preprocessor included.

Fig. 10: Measurement results, die micrograph and chip summary table.

	Jetson ONX[4]	ASPLOS [3] Lee	ISSCC25 [2] Feng	ISSCC25 [1] Song	This work
Technology	8nm	28nm	28nm	28nm	28nm
Die area (mm²)	450	3.95	2.43	20.25	2.03 [1]
Data precision	FP32	FP16	FP16	FP4/8/16	FP16
On-chip SRAM	4MB	272kB	187.4kB	N/A	283.5kB
Max frequency	918MHz	1GHz	700MHz	200MHz	200MHz
End-to-End [2]	Yes	Yes	No (①+③)	Yes	Yes
Silicon verified	Yes	No	Yes	Yes	Yes
PSNR (dB) [3]	33.32	34.4	33.98	25.8	32.166
Throughput (fps)	12.8	118.06 [4]	373 [5]	142.1	168 [6]
Power (mW)	15000	870	664	554-616.5	105.92
Area efficiency (fps/mm²)	0.03	29.89	20.97 [7]	7.02	30.65 [8]
Energy efficiency (frame/mJ)	0.00085	0.136	0.13 [7]	0.230	0.396 [8]

[1] Preprocessor excluded. 8×8 PEC array in renderer. [2] End to end process includes ①preprocess ②sort ③render. [3] Measured under NeRF synthetic dataset (Lego). [4] Normalized to the same DRAM bandwidth. We assume that DMA accounted for 60% of total runtime in the vanilla algorithm. [5] Not end-to-end throughput. [6] Measured under 0.9V, 200MHz. Considering preprocessor and delay of EMA (DDR4 25.6GB/s). [7] Normalized end-to-end area/energy efficiency, considering the overhead of Sort and EMA. [8] Normalized to 16×16 PEC for 800×800 pixels per frame. Preprocessor included.

Fig. 11: Comparison table.

from the last frame dispersively, thereby enhancing the ITA utilization and efficiency.

III. Measurement Results

Fig. 10 shows the area and power breakdown of the HZB-RPU and quantifies the effectiveness of MLS. MLS saves 48% of power and enhances performance by 1.93× with only a 1.2dB PSNR loss. The die micrograph and a chip summary table are presented as well. Fabricated in 28nm LOGIC CMOS technology, HZB-RPU can work under 0.7V - 0.9V with a max frequency of 200MHz, occupying a 2.03mm^2 die area and 283.5kB on-chip SRAM. Evaluated on the NeRF synthetic dataset in FP16 precision, it achieves a peak 168fps end-to-end rendering with 32.166dB PSNR and a 105.92mW power consumption at 0.9V, 200MHz. Compared with GPU and state-of-the-art 3DGS accelerators [1]–[4] (Fig. 11), HZB-RPU achieves a 168fps (1.18×) end-to-end throughput ([2] excluded), a 30.65fps/mm^2 (1.03×) area efficiency and an 0.396 frame/mJ (1.72×) energy efficiency.

IV. Conclusion

The IEEEtran class file is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (Grant No. 92264203); by the National Key R&D Program of China (Grant No. 2022ZD0118902); by the National Natural Science Foundation of China (Grant Nos. 62522403, 92464202, 92464302); by the Key R&D Program of Jiangsu Province (Grant No. BE2023020-1); in part by the Fundamental Research Funds for the Central Universities; and in part by Xiaomi Xuanjie.

References

[1] S. Song, S. Kim, W. Park, J. Park, S. An, G. Park and et al., "IRIS: A 8.55mJ/frame Spatial Computing SoC for Interactable Rendering and Surface-Aware Modeling with 3D Gaussian Splatting," IEEE ISSCC, USA, 2025, pp. 56-57.
[2] X. Feng, H. Wang, C. Tang, T. Wu, H. Yang and Y. Liu, "1.78mJ/Frame 373fps 3D GS Processor Based on Shape-Aware Hybrid Architecture Using Earlier Computation Skipping and Gaussian Cache Scheduler," IEEE ISSCC, USA, 2025, pp. 54-55.
[3] H. Lee, G. Park, W. Park, W. Jo, J. Park and H. Yoo, "A 66.6 FPS High Quality Gaussian Splats Rendering FPGA Processor with Reconfigurable Computation Architecture," IEEE A-SSCC, Japan, 2024, pp. 1-3.
[4] Z. Ye, Y. Fu, J. Zhang, L. Li, Y. Zhang, S. Li and et al., "Gaussian Blending Unit: An Edge GPU Plug-in for Real-Time Gaussian-Based Rendering in AR/VR," IEEE HPCA, USA, 2025, pp. 353-365.
[5] J. Lee, S. Lee, J. Lee, J. Park and J. Sim, "GSCore: Efficient Radiance Field Rendering via Architectural Support for 3D Gaussian Splatting," ACM ASPLOS, USA, 2024, vol. 3, pp. 497-511.
[6] NVIDIA, "Jetson Orin NX Series Modules Data Sheet," https://developer.nvidia.com/downloads/jetson-orin-nx-module-series-data-sheet, (released on 2025/05/28).
[7] J. Ryu, H. Kwon, W. Park, Z. Li, B. Kwon, D. Han and et al., "NeuGPU: A 18.5mJ/Iter Neural-Graphics Processing Unit for Instant-Modeling and Real-Time Rendering with Segmented-Hashing Architecture," IEEE ISSCC, USA, 2024, pp. 372-373.
[8] D. Han, J. Ryu, S. Kim, S. Kim and H. Yoo, "MetaVRain: A 133mW Real-Time Hyper-Realistic 3D-NeRF Processor with 1D-2D Hybrid-Neural Engines for Metaverse on Mobile Devices," IEEE ISSCC, USA, 2023, pp. 50-51.
[9] B. Kerbl, G. Kopanas, T. Leimkuehler and G. Drettakis, "3D Gaussian Splatting for Real-Time Radiance Field Rendering," ACM Trans. Graph., vol. 42 no. 4, 2023.
[10] J. Wang, M. Chan, N. Karaev, A. Vedaldi, C. Rupprecht and D. Novotny, "VGGT: Visual Geometry Grounded Transformer," IEEE CVPR, USA, 2025, pp. 5294-5306.