

A High-Performance Neural Rendering Accelerator Based on Novel Multi-Level Ray Scheduling and Dual-Process Backend

Wenkai Zhou*, Yuefeng Zhang*, Cheng Zhang*, Binzhe Yuan*, Junsheng Chen*, Luntian Zhang*, Xiangyu Zhang*, Pingqiang Zhou*, Jingyi Yu* and Xin Lou*[†]

*School of Information Science and Technology, ShanghaiTech University

[†]GGU Technology Co., Ltd, China

Abstract—Neural rendering enables photorealistic scene reconstruction but remains difficult to deploy on edge devices due to intensive computation, redundant sampling, and memory bandwidth constraints. This work presents a high-performance neural rendering accelerator for real-time embedded rendering. The proposed design integrates: (1) a dual-process backend with fused micro-MLPs to significantly improve sample processing efficiency, (2) multi-resolution spatial partitioning with adaptive ray clustering to exploit sparsity and achieve over 95% cache hit rate, and (3) a multi-level scheduling framework with proactive prefetching to reduce MLP stalls. Implemented on FPGA, the prototype achieves 94.7 FPS at 800×800 resolution with 6.4 W power consumption. An ASIC implementation in 28 nm technology sustains 440 FPS at 268 mW. Experimental results demonstrate state-of-the-art performance and energy efficiency while preserving rendering quality above 30 dB PSNR.

Index Terms—Neural Rendering, Multi-Level Ray Scheduling, Dual-Process Backend, High-Performance, Energy-Efficient

I. INTRODUCTION

Photorealistic neural rendering is a key enabler for immersive spatial computing applications such as augmented and virtual reality. Since the introduction of Neural Radiance Fields (NeRF) [1], implicit neural representations have achieved superior visual fidelity by modeling scenes as continuous functions. However, dense per-ray sampling, repeated multilayer perceptron (MLP) inference, and bandwidth-intensive feature access make real-time deployment on edge platforms challenging. Existing accelerators targeting NeRF-style pipelines [2], [3] or dense MLP computation [4], [5] struggle to sustain real-time performance under tight power constraints. Recent approaches such as Instant-NGP [6] and 3D Gaussian Splatting (3DGS) [7] improve throughput but still require substantial resources.

This work presents a high-performance and energy-efficient neural rendering accelerator for real-time edge deployment. The proposed architecture integrates a dual-process backend, multi-resolution spatial partitioning, and multi-level scheduling with prefetching, achieving 94.7 FPS on FPGA and 440 FPS at 268 mW in 28 nm CMOS, demonstrating state-of-the-art efficiency.

II. HARDWARE ARCHITECTURE

A. Overall Architecture

Fig. 1 illustrates the overall architecture of the proposed accelerator, comprising a front-end and a back-end processing pipeline. To manage the dataflow efficiently, all modules utilize

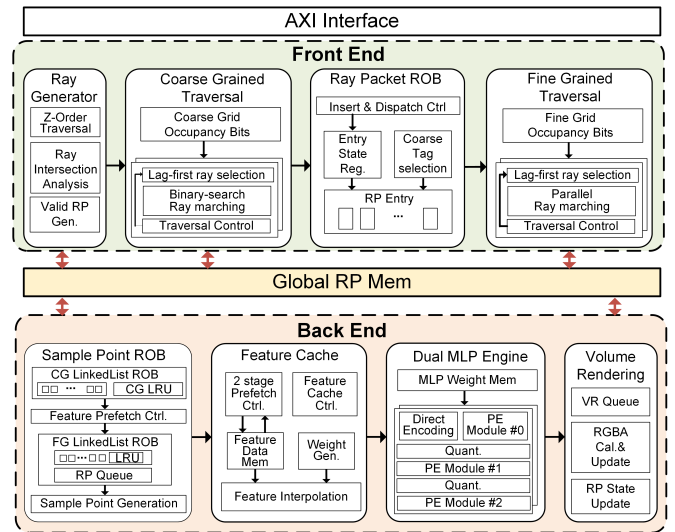


Fig. 1. The Overall Architecture of the Proposed Accelerator

handshake-based stream interfaces to transmit ray data or memory pointers. In the front-end, the Ray Generator produces rays following a Z-order curve and filters out non-intersecting rays. Valid rays are grouped into Ray Packets (RPs) and dispatched to the Coarse-Grained Traversal Unit (CTU) to identify the first intersected Coarse Voxel (CV). A Ray Packet Reorder Buffer (RPROB) then clusters RPs by CV and forwards them to the Fine-Grained Traversal Unit (FTU), where ray marching is performed within Fine Voxels (FVs) to determine sample point locations. In the back-end, generated sample points are first organized by the Sample Point Reorder Buffer (SPROB) to enable coarse-level clustering and feature data prefetching. After prefetching, RPs are further grouped at the fine grid level and sent to the Feature Cache (FC), where dual-sample interpolation is executed in parallel. The interpolated features are then processed by parallel MLP inference followed by Volume Rendering (VR), completing the rendering pipeline.

B. Dual-Process Back-end Acceleration Architecture

We introduce a dual-process back-end pipeline that executes feature interpolation, MLP inference, and volume rendering (VR) in parallel. Paired samples originate from consecutive points along a ray, ray endpoints, or ray packets, enabling effi-

TABLE I
PERFORMANCE COMPARISON WITH STATE-OF-THE-ART FPGA-BASED AND SYNTHESIZED ACCELERATORS

	Edge GPU	CICC'24 [8]	ASSCC'24 [9]	Ours (FPGA)	ISSCC'25 [10]	DATE'24 [11]	Ours (ASIC)
Application	NeRF	NeRF	3DGS	NeRF	3DGS	3DGS	NeRF
Evaluation Method	/	FPGA	FPGA	FPGA	Measurement	Synthesis	Synthesis
Frequency (MHz)	200	200	200	200	700	1000	1000
Memory (KB)	-	2560	1341.4	329	187.4	320	329
Area (mm ²)	-	-	-	-	2.43	3.88	2.46
Frames per Second	2.9	7.8	66.6	94.7	373	117	440
Power (W)	15	5.8	3.6	6.5	0.664	1.039	0.268
Energy Efficiency (nJ/Pixel)	8030	620	84.47	107	2.78	13.87	0.76
Implement Tech. or Platform	-	Xilinx VU19P	Intel Cyclone V	Xilinx VU37P	28nm	28nm	28nm

cient dual-sample processing. By leveraging preload operations and memory partitioning, the Feature Cache (FC) and VR units complete computations for different rays within a single cycle using conventional single-port SRAM. Within the MLP engine, partial reuse of direction-encoding vectors allows simultaneous evaluation of two samples with minimal overhead. Moreover, the design replaces monolithic MLPs with fully fused micro-MLPs assigned to populated coarse voxels. Overall, the dual-process architecture improves sample transport efficiency by 98% and increases the average frame rate by 37%.

C. Multi-level Scheduling Framework

The proposed multi-level scheduling framework with prefetch support coordinates coarse- and fine-grained RP flows to maximize data locality and throughput. At the coarse level, the Coarse-Grained Reorder Buffer (CGROB) clusters RPs by their current coarse voxel (CV) and selects voxels based on RP occupancy and recency to balance progress. A Feature Prefetch Interface (FPI) interacts with the Feature Cache (FC) to ensure that required features and on-chip resources are available before dispatching RPs; otherwise, RPs are deferred for later service. At the fine level, the Fine-Grained ROB (FGROB) further groups RPs by fine voxels (FVs) and flexibly issues entries to exploit spatial locality. This dual-ROB scheduling strategy increases the average frame rate by 49%, reduces MLP weight swapping by 83%, and shortens MLP stall time by 77%, significantly improving back-end pipeline utilization.

III. EVALUATION

We evaluate the proposed accelerator using the widely adopted Synthetic NeRF dataset at a resolution of 800×800 . As shown in Fig. 2, the prototype is implemented on a Xilinx Virtex UltraScale+ VU37P FPGA operating at 200 MHz, where frame rate, power consumption, and on-chip memory usage are directly measured on hardware. For fair comparison with prior ASIC-based accelerators, we further synthesize the design using Synopsys Design Compiler in a 28 nm CMOS technology at 1 GHz, following common evaluation practices.

Rendering quality is measured by PSNR, while performance is reported in frames per second (FPS). Power consumption and energy efficiency are evaluated using nJ/pixel to normalize performance across resolutions. We additionally report chip area and on-chip memory capacity, which are critical metrics

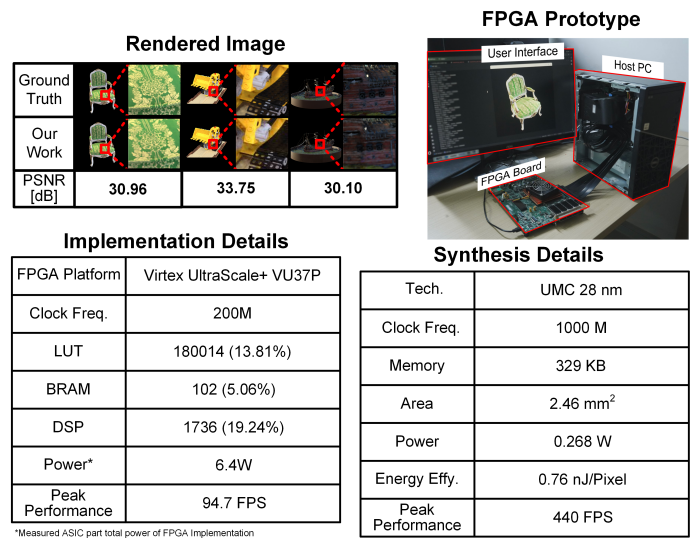


Fig. 2. FPGA Prototype System and Implementation Details.

for edge accelerators. The following results demonstrate the advantages of the proposed design across these dimensions.

As summarized in Table I, the proposed accelerator demonstrates strong performance across both FPGA and ASIC implementations. On a Xilinx VU37P FPGA operating at 200 MHz, the prototype achieves 94.7 FPS with 6.4 W power consumption, corresponding to an energy efficiency of 105.6 nJ/pixel, while using only 329 KB of on-chip memory and preserving high rendering quality. Compared with prior FPGA-based neural rendering accelerators, our design improves frame rate by 42.2% and reduces on-chip memory usage by 63.3%.

For ASIC evaluation, the design synthesized in 28 nm CMOS at 1 GHz sustains 440 FPS with only 0.268 W power consumption, achieving 0.76 nJ/pixel energy efficiency. Relative to state-of-the-art synthesized neural rendering accelerators under the same evaluation methodology, our design delivers a 276% FPS improvement while reducing power consumption by 74.2%.

ACKNOWLEDGMENT

Wenkai Zhou and Yuefeng Zhang contributed equally to this work. The corresponding author of this paper is Xin Lou (louxin@shanghaitech.edu.cn). The authors thank Binzhe Yuan for his valuable technical guidance.

REFERENCES

- [1] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: representing scenes as neural radiance fields for view synthesis," *Commun. ACM*, vol. 65, no. 1, p. 99–106, Dec. 2021. [Online]. Available: <https://doi.org/10.1145/3503250>
- [2] C. Rao, H. Yu, H. Wan, J. Zhou, Y. Zheng, M. Wu, Y. Ma, A. Chen, B. Yuan, P. Zhou, X. Lou, and J. Yu, "ICARUS: A specialized architecture for neural radiance fields rendering," *ACM Trans. Graph.*, vol. 41, no. 6, Nov. 2022. [Online]. Available: <https://doi.org/10.1145/3550454.3555505>
- [3] Y. Wang, Y. Li, H. Zhang, J. Yu, and K. Wang, "Moth: A hardware accelerator for neural radiance field inference on fpga," in *2023 IEEE 31st Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*. IEEE, 2023, pp. 227–227.
- [4] K. Long, C. Rao, Y. He, Z. Yuan, P. Zhou, J. Yu, and X. Lou, "Analysis and design of precision-scalable computation array for efficient neural radiance field rendering," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 70, no. 11, pp. 4260–4270, 2023.
- [5] H. Wan, L. Ma, A. Li, P. Zhou, J. Yu, and X. Lou, "ZeroTetris: A spacial feature similarity-based sparse MLP engine for neural volume rendering," in *Proceedings of the 61st ACM/IEEE Design Automation Conference*, 2024, pp. 1–6.
- [6] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Trans. Graph.*, vol. 41, no. 4, Jul. 2022. [Online]. Available: <https://doi.org/10.1145/3528223.3530127>
- [7] 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, Jul. 2023. [Online]. Available: <https://doi.org/10.1145/3592433>
- [8] Z. Yuan, B. Yuan, Y. Gu, Y. Zheng, Y. He, X. Wang, C. Rao, P. Zhou, J. Yu, and X. Lou, "A 0.59 $\mu\text{J}/\text{pixel}$ high-throughput energy-efficient neural volume rendering accelerator on FPGA," in *2024 IEEE Custom Integrated Circuits Conference (CICC)*, 2024, pp. 1–2.
- [9] H. Lee, G. Park, W. Park, W. Jo, J. Park, and H.-J. Yoo, "A 66.6 FPS high quality Gaussian Splats rendering FPGA processor with reconfigurable computation architecture," in *2024 IEEE Asian Solid-State Circuits Conference (A-SSCC)*. IEEE, 2024, pp. 1–3.
- [10] X. Feng, H. Wang, C. Tang, T. Wu, H. Yang, and Y. Liu, "1.78 mJ/Frame 373 FPS 3D GS processor based on shape-aware hybrid architecture using earlier computation skipping and Gaussian cache scheduler," in *2025 IEEE International Solid-State Circuits Conference (ISSCC)*, vol. 68. IEEE, 2025, pp. 1–3.
- [11] J. Jo and J. Park, "PS-GS: Group-wise parallel rendering with stage-wise complexity reductions for real-time 3D Gaussian Splatting," in *2025 Design, Automation & Test in Europe Conference (DATE)*. IEEE, 2025, pp. 1–7.