APOLLO: Agent-Powered Optimization for Local and Lightweight Rendering Operations

Oroceo Paul Angelo[§], Jae-Min Lee[†], and Dong-Seong Kim[†]
[†]Department of IT Convergence Engineering, Kumoh National Institute of Technology, Gumi, South Korea
[§]Department of Electronic Engineering, Kumoh National Institute of Technology, Gumi, South Korea
{paporoceo, ljmpaul, dskim}@kumoh.ac.kr

***Abstract*—Metaverse rendering architectures face significant scalability and latency challenges, which limit accessibility with resource-constrained devices. This paper proposes APOLLO, a decentralized framework that employs edge-based direction-aware optimization to enhance local rendering by intelligently prioritizing 3D assets based on user viewport and interaction patterns. While the complete framework encompasses several components, including Large Language Models (LLMs) for context awareness and prediction, our proof-of-concept implementation focuses on direction-aware optimization. By implementing deterministic object culling based on camera direction, our approach substantially reduces resource requirements while maintaining visual quality in the primary viewing field. Our implementation demonstrates significant reductions in rendering time and resource consumption, achieving up to 45 FPS for typical scenes on standard hardware. These results highlight the framework's potential to enhance efficiency and inclusivity compared to traditional approaches.**

***Index Terms*—Direction-aware rendering, metaverse, edge computing, decentralization, optimized rendering, large language models**

I. INTRODUCTION

The metaverse promises immersive digital experiences but faces significant technical barriers. Current implementations rely on centralized architectures. Powerful servers handle rendering and computation, creating accessibility challenges for users with resource-constrained devices [1]. These systems struggle with several critical problems. Users with smartphones and budget laptops cannot fully participate in high-fidelity metaverse environments, creating digital inequality [2]. As user numbers increase, centralized systems struggle to maintain performance, often fragmenting experiences into isolated instances with artificial user limits [3]. Server-side rendering introduces significant delays between user actions and visual feedback, diminishing the immersive experience [1].

This paper proposes APOLLO (Agent-Powered Optimization for Local and Lightweight Metaverse Operations), a framework combining direction-aware optimization, Large Language Models (LLMs) for contextual understanding, blockchain technology, and adaptive rendering techniques to enable resource-constrained devices to participate fully in immersive metaverse experiences. The framework proposes using small-scale LLMs to provide intelligent awareness of user patterns, behavioral prediction, and contextual understanding of interaction environments. While the complete framework envisions multiple components, our current implementation focuses on proving the effectiveness of direction-aware optimization for local rendering as a foundation for future LLM integration.

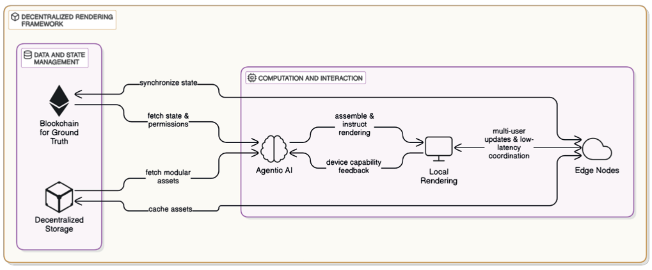

Fig. 1. APOLLO Framework Overview

II. FRAMEWORK DESIGN AND IMPLEMENTATION

The APOLLO framework enables efficient decentralized metaverse rendering through integrated components. Decentralized Storage distributes 3D assets across IPFS nodes [4], with users pre-downloading essential assets at initialization. Lightweight Blockchain State Management maintains a consistent canonical state for all participants [5].

Direction-Aware Optimization, our implementation's focus, uses quaternion-based camera detection to determine objects in the user's field of view [6]. The full framework proposes enhancing this with edge-deployed LLMs that analyze user behavior to predict likely interactions [7]. These LLMs would track movement data and interaction patterns to prioritize rendering for objects users may interact with before direct viewing.

This paper implements only the direction-aware component without LLM integration, still significantly improving performance for resource-constrained devices. The implementation includes quaternion-based orientation analysis, deterministic object culling, adjacent direction modeling, and memory tracking. Adaptive Local Rendering uses WebGL/Three.js to adjust detail levels based on importance and device capabilities [1]. In multi-user scenarios, Peer-Coordinated Interactions combine blockchain updates with WebRTC for low-latency communication [3].

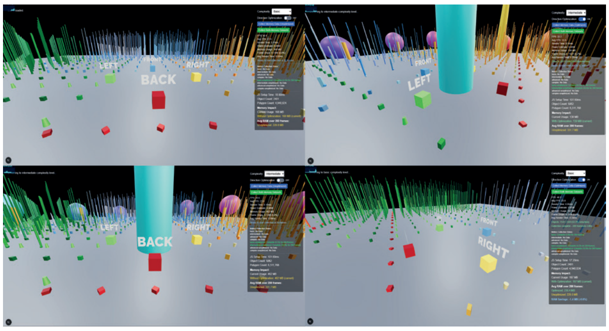

Fig. 2. Direction-Aware Optimization Implementation

TABLE I
COMPREHENSIVE PERFORMANCE COMPARISON

Metric	APOLLO	Traditional
Scene Specifications		
Simple Scene	5,062 objects 6,511,768 polygons	5,062 objects 6,511,768 polygons
Complex Scene	18,500 objects 20,000,000+ polygons	18,500 objects 20,000,000+ polygons
Rendering Performance		
FPS (Intermediate)	47.6 (peak) / 26.1 (avg)	<30
Render Time (ms)	0.25 (avg)	>0.5
Memory Usage (MB)	147	333
Objects Rendered	2023/5062 (60% reduction)	5062 (100%)
Frame Drops	19/7779 (0.2%)	>5%

III. SYSTEM EVALUATION

We evaluated our optimization approach using a proof-of-concept implementation with Next.js and Three.js across various device categories, focusing solely on the rendering optimization aspect of the APOLLO framework. As shown in Table I, our implementation achieved faster rendering times, maintained higher frame rates, and demonstrated substantially lower resource utilization compared to traditional platforms. Most notably, the optimization reduced the number of rendered objects by approximately 60% while maintaining visual quality in the user's field of view. When testing with our intermediate complexity scene, the optimized implementation achieved peak frame rates of 47.6 FPS with an average of 26.1 FPS, significantly outperforming the unoptimized version.

Our optimization implementation yielded significant improvements in resource utilization. The system reduced rendered objects from 5,062 to 2,023 (60% reduction) while maintaining visual integrity by preserving key landmarks in the viewing direction. Memory usage decreased from 333 MB to 147 MB (56% reduction), and frame drops fell to just 0.2% compared to over 5% in traditional implementations. These results provide a promising baseline for future integration of LLM-based user pattern recognition, which we expect to further reduce resource requirements by predicting interaction targets beyond the current field of view.

To ensure robust results, we warmed up the with 90 initial frames before collecting data across 200 frames for statistical significance. We measured optimized vs. unoptimized states and benchmarked multiple complexity levels from basic to complex scenes. This approach ensured our metrics accurately reflect real-world usage scenarios, as illustrated in Fig. 2.

IV. CONCLUSION

The implementation of APOLLO's direction-aware optimization demonstrates significant performance improvements for metaverse applications on resource-constrained devices. By reducing rendered objects by 60% and memory usage by 56% while maintaining visual quality, this approach enables more accessible metaverse experiences. The quaternion-based camera direction detection and deterministic object culling show the potential for contextual awareness in improving 3D web application performance. Our framework proposes integrating small-scale LLMs to further enhance this awareness by understanding and predicting user interaction patterns, potentially achieving greater optimization beyond simple direction-based culling. This represents a path toward democratizing metaverse access through intelligent optimization, decentralized architecture, and adaptive techniques for entry-level devices.

Future work will implement additional framework components including LLM integration for user pattern recognition, decentralized storage, blockchain state management, and peer-coordinated interactions. By demonstrating the significant performance improvements from even a partial implementation focusing solely on direction awareness, we establish a foundation for a fully integrated system that can make immersive metaverse experiences accessible without requiring high-end hardware.

ACKNOWLEDGEMENT

This work was partly supported by Innovative Human Resource Development for Local Intellectualization program through the IITP grant funded by the Korea government(MSIT) (IITP-2025-RS-2020-II201612, 33%) and by Priority Research Centers Program through the NRF funded by the MEST(2018R1A6A1A03024003, 33%) and by the MSIT, Korea, under the ITRC support program(IITP-2025-RS-2024-00438430, 34%).

REFERENCES

[1] N. Aung, S. Dhelim, L. Chen *et al.*, "Edge-enabled metaverse: The convergence of metaverse and mobile edge computing," *Tsinghua Science and Technology*, vol. 29, no. 3, pp. 795–805, 2024.

[2] T. Baidya *et al.*, "Comprehensive survey on resource allocation for edge-computing-enabled metaverse," *Journal of Network and Computer Applications*, vol. 236, p. 103649, 2024.

[3] K. Smolorz, "How edge computing can unlock the metaverse," *STL Partners*, 2024. [Online]. Available: https://stlpartners.com/articles/edge-computing/edge-computing-metaverse/

[4] T. R. Gadekallu *et al.*, "Blockchain for the metaverse: A review," *arXiv preprint arXiv:2203.09738*, 2022.

[5] Q. Yang, Y. Zhao, H. Huang, Z. Xiong, J. Kang, and Z. Zhang, "Fusing blockchain and ai with metaverse: A survey," *IEEE Open Journal of the Computer Society*, vol. 3, pp. 122–136, 2022.

[6] S. Shah, "Efficient resource management with small language models (slms) in edge computing," *InfoQ*, 2024. [Online]. Available: https://www.infoq.com/articles/efficient-resource-management-small-language-models/

[7] Z. Wan, Z. Zhou, M. Xu *et al.*, "A review on edge large language models: Design, execution, and applications," *ACM Computing Surveys*, 2024.