Bulletin of Electrical Engineering and Informatics
Vol. 14, No. 5, October 2025, pp. 3659~3670
ISSN: 2302-9285, DOI: 10.11591/eei.v14i5.9532

Performance analysis of 3D assets in virtual reality simulations for climate change: a case study in sustainable energy systems

Cahya Miranto[1], Ardiman Firmanda[1], Hestiasari Rante[2], Sritrusta Sukaridhoto[2], Muhammad Agus Zainuddin[2], Haolia Rahman[3]

[1]Study Program of Game Technology, Department of Informatics Engineering, Politeknik Negeri Batam, Batam, Indonesia
[2]Study Program of Multimedia Engineering Technology, Department of Informatics and Computer Engineering, Politeknik Elektronika Negeri Surabaya, Surabaya, Indonesia
[3]Study Program of Manufacturing Engineering Technology, Department of Mechanical Engineering, Politeknik Negeri Jakarta, Jakarta, Indonesia

Article Info

Article history:

Received Nov 8, 2024
Revised Jul 31, 2025
Accepted Aug 8, 2025

Keywords:

3D asset management
Climate change education
Game performance analysis
Immersive learning
Virtual reality simulations

ABSTRACT

This study investigates the performance impact of 3D assets in a virtual reality (VR) simulation designed for climate change education, aiming to balance visual fidelity and system efficiency on standalone headsets. Using a case study modeled on a sustainable energy environment, key performance metrics frames per second (FPS), triangle count, and draw calls were measured to assess the effect of object density, material transparency, and batching strategies. Experimental results show that configurations with 20 trees and 20 characters maintained 101 FPS, while denser scenes with 30 trees and 30 characters dropped to 79 FPS approaching the minimum usability threshold for VR. Transparent tree foliage with alpha-cutout materials imposed higher graphics processing unit (GPU) loads than high-triangle opaque character models, highlighting the performance cost of material complexity. These findings offer practical guidelines for optimizing asset configurations in immersive educational VR content. Future work may explore integration of artificial intelligence (AI) behavior and user interaction to assess broader system performance.

Corresponding Author:

Hestiasari Rante
Study Program of Multimedia Engineering Technology
Department of Informatics and Computer Engineering, Politeknik Elektronika Negeri Surabaya
St. Raya ITS Sukolilo Kampus PENS, Surabaya, Indonesia
Email: hestiasari@pens.ac.id

1. INTRODUCTION

Climate change remains one of the most pressing global challenges affecting ecosystems, energy systems, and public health. The integration of renewable energy technologies such as solar panels, microgrids, and sustainable water systems has been widely acknowledged as a critical step toward climate mitigation [1], [2]. However, despite technological progress, a significant gap persists in public awareness and engagement, particularly in understanding how these systems function and contribute to environmental sustainability. Traditional educational approaches often struggle to convey these concepts in a manner that is both engaging and accessible, especially for younger or non-technical audiences [3], [4].

Virtual reality (VR) offers new opportunities for immersive learning by allowing users to explore complex systems within simulated interactive environments [5]-[7]. Previous studies have shown that VR can improve motivation, comprehension, and knowledge retention in STEM-related education, including climate change [8]-[11]. Immersive simulations have also been used to visualize ecological forecasts, enabling users to explore climate-driven changes in forest environments and bridging scientific modeling with public engagement [12]. While numerous VR-based initiatives have been developed to visualize environmental phenomena or teach energy principles, most of them focus heavily on content delivery without thoroughly addressing the technical performance constraints of standalone VR devices [13], [14].

Standalone VR headsets, such as the Oculus Quest series, offer portability and accessibility, but have limited hardware capabilities compared to PC-based systems. Rendering complex 3D scenes in real time on such platforms poses substantial challenges, especially when visual fidelity must be balanced with consistent frame rates to avoid motion sickness and cognitive fatigue [15]. Maintaining stable performance, typically above 72 to 90 frames per second (FPS), is not only a technical requirement, but also a pedagogical necessity in VR learning environments [16], [17].

In response to these challenges, various optimization strategies have been explored in VR development, including mesh simplification, texture compression, and batching techniques [18], [19]. However, there is limited empirical data on how specific 3D asset configurations, such as the use of transparent materials versus high-polygon models, affect rendering performance on mobile VR devices [16]. This lack of benchmarking is particularly notable in the context of educational simulations that require high visual realism and real-time interaction [20].

This study addresses this gap by conducting a performance analysis of 3D assets in a VR simulation designed for climate change education. Using a case study set in Batam City, Indonesia, we evaluate how different asset properties, including object density, geometric complexity, and material transparency, affect performance metrics, such as FPS, triangle count, and draw calls. The goal is to identify asset configurations that maintain visual richness while ensuring stable performance on standalone VR devices. The results offer practical guidelines for educators and developers seeking to design scalable, immersive content that aligns with both technical limitations and educational goals.

2. METHOD

This study investigates the performance impact of 3D assets within a VR simulation designed to support climate change education, with the goal of identifying asset configurations that balance visual fidelity and real-time rendering performance, particularly on standalone VR devices. The study emphasizes empirical testing to evaluate how geometric complexity, material transparency, and object density affect key performance metrics. Aimed at VR developers and educators, the methodology provides practical insights by documenting asset optimization techniques, testing scenarios, and the performance monitoring setup in detail, contributing to study of reproducibility and aligning with best practices in engineering research and benchmarking.

The overall research process is summarized in Figure 1, which outlines the sequential stages of the experiment from 3D asset preparation and optimization, virtual scene setup, testing deployment on Oculus Quest 3, and collection of performance metrics. This diagram provides a high-level view of the workflow used to evaluate the impact of asset complexity on real-time VR performance.

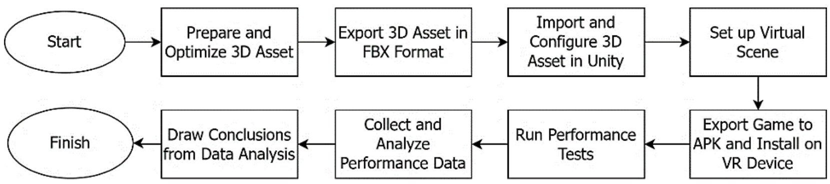

Figure 1. Flow diagram of the research process

2.1. Research objective and system platform

This study aims to investigate how key 3D asset parameters, namely triangle count, draw calls, material transparency, and object density, affect real-time rendering performance in VR environments. The testbed is a virtual simulation of a sustainable energy system set in Batam City, Indonesia, designed for climate change education. The goal is to determine optimal asset configurations that balance visual richness with stable frame rates on standalone VR headsets.

Three performance indicators guide this evaluation:

- FPS, which must exceed 72 to ensure comfort and prevent motion sickness.
- Triangle count, a measure of geometric complexity affecting graphics processing unit (GPU) load.
- Draw calls, which indicate the number of rendering operations per frame, heavily influenced by material variety and batching strategies.

These metrics are well-established benchmarks in VR performance studies, particularly relevant for mobile devices with limited hardware resources [16], [21]. The simulation was developed using Unity 2022.3 LTS, configured with the universal render pipeline (URP), a lightweight rendering solution optimized for mobile and VR applications [22]. All 3D assets were modeled and optimized in Blender 3.6, exported in FBX format, and imported into Unity. Texture maps were compressed using efficient formats, such as JPEG and PNG, to reduce the memory overhead [23].

Testing was conducted on the Oculus Quest 3, a standalone VR headset powered by the Snapdragon XR2 Gen 2 chipset, with 8 GB of RAM and a 120 Hz native refresh rate. This device was selected for its growing popularity in educational and industrial applications, where cost, mobility, and standalone functionality are key. Unlike PC-tethered systems, it presents real-world constraints in rendering performance, making it an ideal platform for assessing practical deployment scenarios for immersive learning [24], [25].

Runtime performance was monitored using Unity Profiler and a custom in-game UI that displayed real-time statistics for FPS, triangle count, and draw calls. These indicators were logged to CSV files at one-second intervals for offline analysis. The simulation also utilized Unity’s rendering statistics panel during APK builds to ensure measurement accuracy during sideloaded deployment.

Figure 2 illustrates the system architecture of the VR environment, including both the virtual simulation workflow in Unity and the integration with real-world IoT data sources. While this study focuses exclusively on rendering performance within the VR environment, the broader system design contextualizes its real-time educational applications.

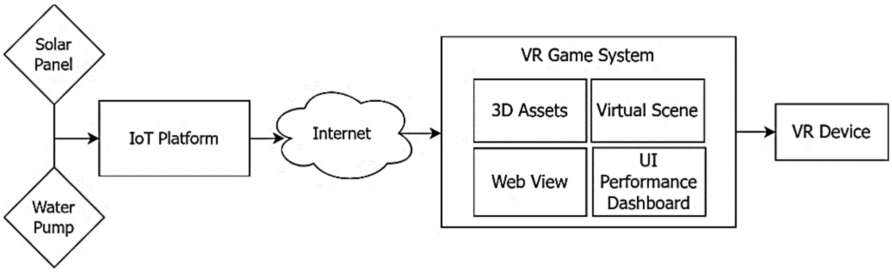

Figure 2. System design of the VR simulation environment

2.2. Asset optimization and scene assembly

To ensure optimal performance on standalone VR devices, all 3D assets were systematically optimized through a multistage process aimed at reducing polygon complexity, material overhead, and draw calls. The assets comprising static and dynamic models such as terrain, vegetation, buildings, and animated characters were created in Blender 3.6, simplified, and then integrated into Unity using the URP.

Polygon reduction was achieved through both manual decimation and automated level of detail (LOD) generation. Trees, in particular, were configured with four LOD stages, ranging from detailed geometry (LOD 0) to billboard-style representations (LOD 3). This approach ensured high visual fidelity up close and minimal rendering cost at greater distances [26]-[28]. Figure 3 shows the visual transition between LOD stages.

Figure 3. Tree model appearance with LOD stages

Material optimization was a critical part of the workflow. Transparent tree foliage required diffuse and alpha-cutout materials, while character models used only three shared URP Lit opaque materials. Texture maps were compressed using Unity's built-in tools, converting them to efficient formats like adaptive scalable texture compression (ASTC) for Android deployment. The use of shared materials reduced rendering state changes and improved batching efficiency.

Batching strategies were employed to further enhance performance. Static batching was applied to terrain and infrastructure, while GPU instancing was enabled for frequently repeated assets such as trees. This allowed the rendering engine to group similar objects during draw calls, significantly reducing CPU overhead. Table 1 summarizes the primary game objects used in the virtual Batam environment, along with their triangle counts, material types, and batching configurations.

Table 1. List of game objects used in the virtual Batam environment

Model	Tris count	Material type	Batches
Character	16.780	3 URP Lit (Opaque) material	8
Terrain	41.055	1 URP Lit (Opaque) material	10
Ocean	8.985	1 custom shader	2
Tree LOD 0	16.322	2 diffuse material and 1 diffuse with alpha cutout for the leaves	6
Tree LOD 1	8.134	2 diffuse material and 1 diffuse with alpha cutout for the leaves	4
Tree LOD 2	3.482	2 diffuse material and 1 diffuse with alpha cutout for the leaves	4
Tree LOD 3	18	1 diffuse material with alpha cutout for the leaves	2

LOD transitions were enforced during placement to minimize overdraw from distant assets, and batching was manually reviewed to ensure effectiveness across vegetation and terrain elements. A visual overview of the scene is presented in Figure 4, and scene metrics including total triangle count and draw calls are outlined in Table 2, which served as the baseline configuration for performance testing.

Figure 4. Virtual Batam City environment scene

Table 2. Overview of virtual Batam environment

City	Tris count	Draw calls	Scene complexity
Batam	55.370	65	100 trees, terrain, ocean, barelang bridge, and solar panels

2.3. Performance testing and scenario setup

To evaluate how 3D asset complexity affects rendering performance on standalone VR devices, a series of controlled experiments was conducted using the Oculus Quest 3. The VR simulation was exported as an APK and sideloaded for native execution, ensuring that all results reflect real-world performance without tethered GPU acceleration [16].

A custom user interface (UI) was developed to visualize real-time performance metrics FPS, triangle count, and draw calls alongside interactive controls for dynamically adjusting scene complexity. Using a slider-based input system, researchers could incrementally spawn trees or animated characters and monitor performance fluctuations during runtime. Metrics were also logged every second to CSV files for post-test analysis. Figure 5 shows the in-game UI used during the test procedures.

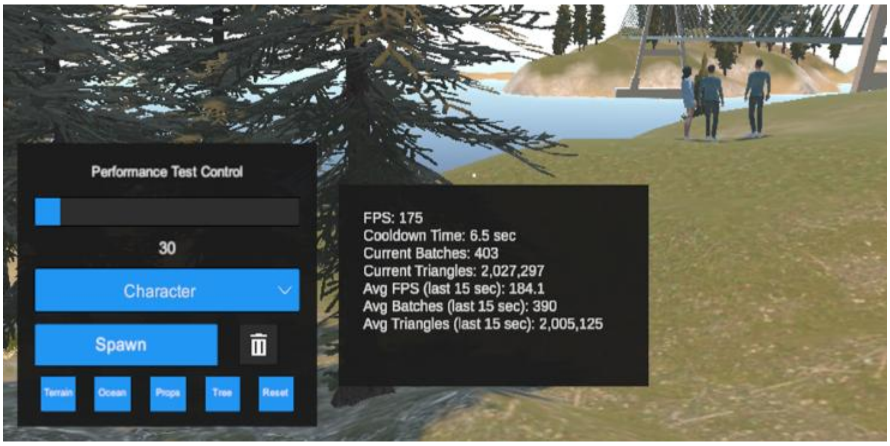

Figure 5. UI for controlling and monitoring game performance

Object placements followed structured spatial rules to minimize inconsistency. Tree models were evenly distributed across both islands, while character models were confined to one island and performed idle looping animations without AI logic or navigation, thereby isolating rendering cost from behavioral overhead. An automated script controlled the player avatar's path across eight predefined waypoints, enabling uniform scene traversal across all scenarios. Each test lasted one minute, ensuring consistency in timing and view exposure. The avatar's camera remained at eye-level with a forward-facing orientation to replicate typical VR perspective. Figure 6 illustrates the virtual Batam City scene, showing Figure 6(a) as the eight designated points for player navigation and Figure 6(b) as the corresponding camera view used during data capture.

(a) (b)

Figure 6. Virtual Batam City scene; (a) eight designated points for player navigation and (b) eye-level camera view

Six distinct test scenarios were constructed by varying object counts and types. Static assets (trees) used alpha-cutout materials to simulate transparent foliage, while dynamic assets (characters) featured high triangle counts and opaque materials. Test conditions incremented trees and characters in groups of ten from 10 to 100 trees and 10 to 200 characters, respectively. Additionally, hybrid configurations (e.g., 20 trees+20 characters, 30 trees+30 characters) were created to examine combined effects of transparency and geometry-heavy rendering.

FPS, triangle count, and draw calls were selected as core performance indicators, following VR development best practices. FPS directly relates to user comfort, with sustained rates above 72 FPS considered necessary for immersive VR use [29]. Triangle count and draw calls indicate rendering load on GPU and CPU respectively, with transparency and material diversity known to significantly affect batching efficiency [12]. Each configuration was tested using the same movement and logging procedure to ensure data comparability across scenarios.

3. RESULTS AND DISCUSSION

This section presents the results of performance testing across different 3D asset configurations in the virtual Batam City environment using metrics such as FPS, triangle count, and draw calls. The analysis explores how varying the number of static and dynamic assets, specifically trees and animated characters, affects rendering performance on Oculus Quest 3. Beyond presenting raw data, the discussion focuses on interpreting the impact of asset characteristics (e.g., material type, transparency, and batch count) and identifying thresholds critical for maintaining a usable frame rate in educational VR applications. The findings are also compared across asset types to provide design recommendations for developers working on immersive content for standalone devices.

3.1. Performance impact of tree density

The first test assessed how increasing the number of tree models with alpha-cutout materials used for foliage impacted rendering performance. Despite having relatively modest triangle counts, the use of transparency significantly increased GPU workload due to overdraw and reduced batching efficiency.

As shown in Table 3, the simulation maintained 120 FPS with up to 20 trees. However, performance declined sharply beyond this point: 30 trees dropped FPS to 102, 40 trees to 78, and 60 trees to 58 falling below the 72 FPS threshold commonly recommended for VR comfort. At the highest density of 100 trees, the frame rate dropped to 39 FPS, far below usability standards. These performance trends are visualized in Figure 7, which plots FPS, triangle count, and batch count against the number of trees.

Table 3. Game performance metrics (FPS, batches, and triangle count) for each 10-tree spawn increment

Number of trees	FPS	Batches	Triangle count (tris)
10	120	40	44.347
20	120	41	44.214
30	102	47	45.113
40	78	52	45.474
50	67	55	46.023
60	58	58	46.478
70	54	60	47.062
80	46	68	49.542
90	40	78	53.217
100	39	80	55.370

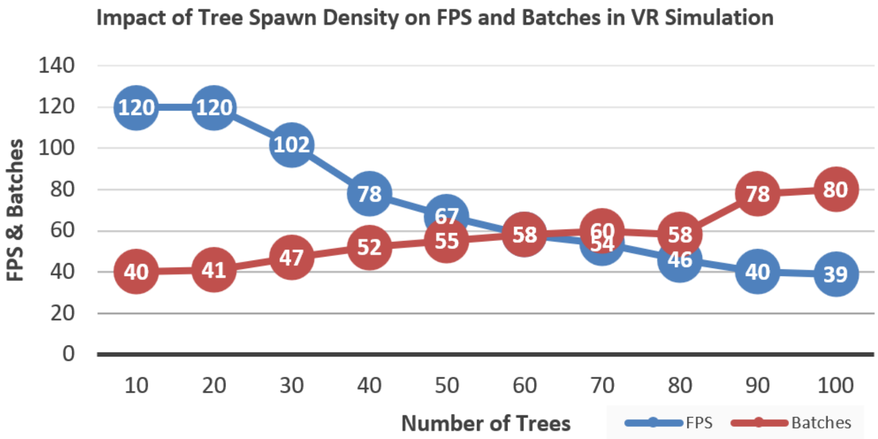

Figure 7. FPS and batch values for each incremental tree spawn (in groups of 10)

While triangle count increased modestly from 44,000 to 55,000 across test cases, draw calls and overdraw from transparent materials had a disproportionately larger impact on FPS. For instance, doubling the tree count from 30 to 60 raised triangle count by only ~1,300 but caused a drop from 102 FPS to 58. The inability to batch transparent objects effectively due to material diversity and sorting requirements was the primary cause of this degradation. These results indicate that transparency and material complexity have a stronger negative impact on standalone VR performance than geometric density alone. For environments rich in vegetation, developers are encouraged to minimize transparency, limit material variety, and consider opaque or simplified foliage to preserve frame rates [30], [31].

3.2. Performance impact of character density

The second experiment evaluated how animated character models, which featured high triangle counts but used opaque materials, affected performance. Each character exceeded 16,000 triangles and performed idle looping animations without artificial intelligence (AI) or physics to isolate rendering cost.

Unlike the tree tests, character models demonstrated stronger performance stability despite higher geometric complexity. As shown in Table 4, the simulation sustained 120 FPS up to 40 characters. Even with 60 characters, FPS remained at 86, and only dropped below the 72 FPS threshold at 80 characters (61 FPS). The lowest performance recorded was 43 FPS at 200 characters. Trends in FPS, draw calls, and triangle counts are visualized in Figure 8.

Table 4. Game performance metrics (FPS, batches, and triangle count) for each 10-character spawn increment

Number of characters	FPS	Batches	Triangle count (tris)
10	120	64	104.172
20	120	107	194.328
30	120	148	280.377
40	120	186	358.816
50	109	223	438.049
60	86	261	516.553
70	67	293	584.038
80	61	329	659.166
90	60	371	746.871
100	60	408	825.610
200	43	693	1.423.328

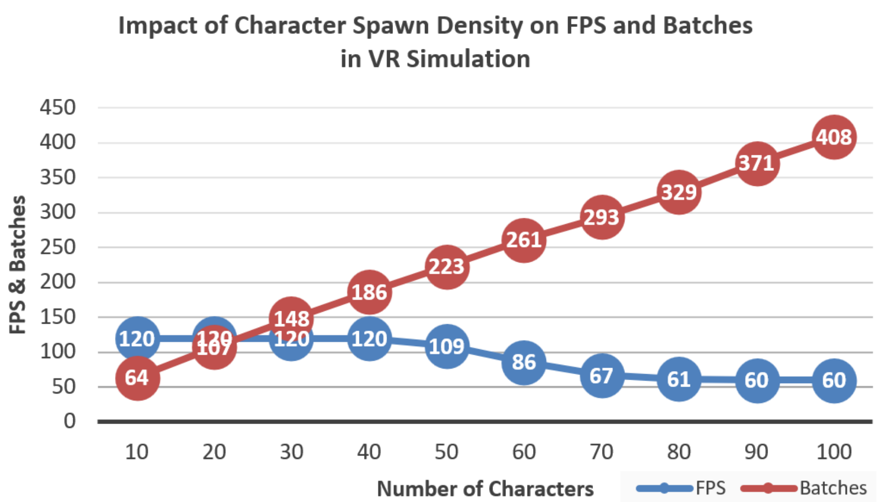

Figure 8. FPS and batch values for each incremental character spawn (in groups of 10)

Despite triangle counts exceeding 800,000 in the 100-character scenario, performance degradation was moderate compared to tree tests. This efficiency is largely attributed to the use of opaque URP Lit materials and shared shaders, which enabled effective batching and minimized overdraw. Additionally, the uniform material configuration across characters helped reduce draw call fragmentation.

These findings suggest that standalone VR systems can handle relatively dense, geometry-heavy animated assets if material complexity is minimized and batching is leveraged [32]. For educational simulations requiring multiple avatars, up to 60–70 animated characters can be rendered while maintaining smooth frame rates, provided transparency is avoided.

3.3. Comparative analysis of trees vs. characters

A direct comparison of tree and character performance reveals that material properties have a greater influence on frame rate than geometric complexity. Although tree models had significantly lower triangle counts, they caused earlier and more severe FPS degradation than the high-polygon character models. At 60 objects, for example, trees reduced FPS to 58, while characters maintained 86 FPS despite exceeding 500,000 triangles.

This performance gap is primarily due to alpha-cutout materials used in tree foliage, which increase overdraw and prevent efficient GPU batching. In contrast, characters used opaque shaders and shared materials, enabling the rendering pipeline to batch draw calls effectively and minimize rendering overhead. Additionally, draw calls rose sharply with trees due to material variation and transparency, whereas characters maintained lower draw call counts relative to object quantity. While characters included skeletal animations, their lack of AI and logic kept CPU load minimal. These results reinforce that transparency, not triangle count, is the more critical constraint for standalone VR devices. Optimizing material use and batching strategy is essential for preserving performance, especially when designing dense, immersive environments [33].

3.4. Optimal asset configuration for usable frame rate

Based on the test results, several asset configurations were identified that maintain frame rates within acceptable VR usability thresholds on standalone devices. In general, configurations that sustain ≥90 FPS are considered optimal, while ≥72 FPS is the minimum acceptable standard for user comfort in immersive environments like the Oculus Quest 3 [17].

For static assets, scenes with up to 30 trees consistently maintained FPS above 90, with performance dropping to 78 FPS at 40 trees approaching the usability threshold. For dynamic assets, up to 70 animated characters sustained performance above 72 FPS, with 50 characters averaging above 100 FPS. In hybrid scenarios, balanced configurations such as 20 trees+20 characters sustained 101 FPS, while 30 trees+30 characters dropped to 79 FPS still within acceptable limits. These mixed conditions reflect realistic VR scenes that combine environmental elements and interactive agents. Table 5 summarizes the detailed metrics for these optimal configurations, presenting the scenarios that achieve target frame rates of 90 FPS and 72 FPS or higher.

Table 5. Ideal configuration scenarios for achieving target FPS (90 and 72 FPS)

Configuration	FPS	Batches	Triangle count (tris)
20 trees and 20 characters	101	124	206.270
30 trees and 30 characters	79	124	197.722

These benchmarks offer practical guidance for VR developers working on educational simulations or digital twin applications. By adhering to these limits, content creators can ensure a balance between visual fidelity and real-time responsiveness on mobile VR platforms.

3.5. Key findings and design implications

This study highlights several critical insights for designing performance-efficient VR simulations on standalone platforms such as the Oculus Quest 3. First, material properties, particularly transparency, have a more pronounced impact on performance than geometric complexity. Transparent assets with alpha-cutout shaders, such as tree foliage, caused significant frame rate drops due to overdraw and poor batching efficiency. In contrast, high-polygon character models using opaque materials sustained smoother performance because they allowed better rendering optimization.

Second, batching and material uniformity proved essential for maintaining frame stability at scale. Character assets that shared a small number of opaque materials could be efficiently grouped during rendering, which minimized draw call overhead even when object counts were high. This strategy reduced the GPU workload and maintained usability even with dense populations of animated models.

Third, the test results aligned with previously established usability thresholds, where 90 FPS is considered optimal and 72 FPS is the minimum recommended for comfort in mobile VR [17], [26]. The asset configurations tested in this study demonstrated that maintaining performance above these thresholds supports smoother experiences, which is especially beneficial in educational simulations requiring sustained immersion.

These findings emphasize the importance of performance-aware asset design in immersive VR development. Developers targeting mobile or standalone systems should prioritize opaque shader use, limit material variations, and implement LOD and batching strategies early in the design process. By following these principles, it is possible to create visually rich, scalable VR experiences that remain responsive and accessible. Future work will expand the performance framework to include AI behavior, real-time physics, and user interaction, enabling a more comprehensive assessment of full-system demands in immersive learning environments.

4. CONCLUSION

This study examined the impact of 3D asset configurations on rendering performance in a VR simulation designed for climate change education, targeting standalone platforms such as the Oculus Quest 3. By systematically testing both static (trees with transparent materials) and dynamic (animated characters with opaque shaders) assets, the results showed that material complexity especially transparency has a more substantial effect on frame rates than polygon count alone.

Transparent foliage caused early and significant performance drops due to overdraw and batching limitations, while high-triangle character models sustained higher FPS through optimized material use and batching efficiency. Performance thresholds of 90 FPS (optimal) and 72 FPS (minimum) were used to identify viable asset configurations. Based on these limits, the study recommends design constraints such as 30 trees, 60–70 characters, or a balanced mix of 30 trees and 30 characters for ensuring a usable frame rate.

Beyond raw performance data, the findings highlight the importance of early asset design decisions in achieving scalable, immersive experiences on mobile VR systems. Future work will incorporate AI behavior, user interaction, and real-time physics to extend performance benchmarking across broader system demands in full-featured educational simulations.

ACKNOWLEDGMENTS

The authors would like to thank APTV under the Directorate of Vocational Higher Education, Ministry of Education, Culture, Research, and Technology of the Republic of Indonesia for supporting this consortium-based research initiative. The authors also acknowledge the collaborative contributions from all participating institutions involved in the project.

FUNDING INFORMATION

This research described in this article was supported by a research grant from APTV under the Directorate of Vocational Higher Education, Ministry of Education, Culture, Research, and Technology of the Republic of Indonesia through the Applied Assignment Research Scheme, under Contract No. 67/SPK/D.D4/PPK.01.APTV/III/2024, dated 19 March 2024. The project is titled "Platform Real-time Monitoring Kualitas Air dan Pengukuran Carbon Emissions berbasis VR terintegrasi AI dan IoT sebagai Perwujudan Climate Action."

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Cahya Miranto	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
Ardiman Firmanda	✓	✓		✓	✓	✓			✓	✓	✓			
Hestiasari Rante	✓	✓		✓			✓	✓	✓	✓		✓	✓	✓
Sritrusta Sukaridhoto	✓	✓	✓	✓			✓	✓		✓				✓
Muhammad Agus Zainuddin	✓			✓						✓				✓
Haolia Rahman	✓			✓			✓	✓		✓			✓	

C	:	**C**onceptualization	I	:	**I**nvestigation	Vi	:	**Vi**sualization
M	:	**M**ethodology	R	:	**R**esources	Su	:	**Su**pervision
So	:	**So**ftware	D	:	**D**ata Curation	P	:	**P**roject administration
Va	:	**Va**lidation	O	:	Writing - **O**riginal Draft	Fu	:	**Fu**nding acquisition
Fo	:	**Fo**rmal analysis	E	:	Writing - Review & **E**diting			

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author, [HR], upon reasonable request.

REFERENCES

[1] S. Gkalonaki and K. Karatzas, "Assessing the environmental impacts of renewable energy sources with emphasis on wind energy," *IOP Conference Series: Earth and Environmental Science*, vol. 1123, no. 1, pp. 1–10, Dec. 2022, doi: 10.1088/1755-1315/1123/1/012053.

[2] E. Uğurlu, "No Renewale Energy Sources and Climate Change Mitigation," *Energy Policy Advancement*, pp. 69–92, Dec. 2021, doi: 10.1007/978-3-030-84993-1_4.

[3] G. Huang and H. Wang, "Traditional Energy and Renewable Energy: The Differences and Transitions," in *Proceedings of the 2021 International Conference on Public Art and Human Development (ICPAHD 2021)*, 2022, pp. 1100–1105, doi: 10.2991/assehr.k.220110.207.

[4] W. N. A. W. Mohamad and K. Osman, "A Systematic Literature Review on Citizen Awareness of Energy," *Malaysian Journal of Social Sciences and Humanities (MJSSH)*, vol. 7, no. 10, p. e001803, Oct. 2022, doi: 10.47405/mjssh.v7i10.1803.

[5] R. R. Ravichandran and J. Mahapatra, "Virtual Reality in Vocational Education and Training: Challenges and Possibilities," *Journal of Digital Learning and Education*, vol. 3, no. 1, pp. 25–31, Apr. 2023, doi: 10.52562/jdle.v3i1.602.

[6] J. Choi, C. E. Thompson, J. Choi, C. B. Waddill, and S. Choi, "Effectiveness of Immersive Virtual Reality in Nursing Education: Systematic Review," *Nurse Educator*, vol. 47, no. 3, pp. E57–E61, 2022, doi: 10.1097/NNE.0000000000001117.

[7] Y. J. Kim and S. Y. Ahn, "Factors influencing nursing students' immersive virtual reality media technology-based learning," *Sensors*, vol. 21, no. 23, pp. 1–12, Dec. 2021, doi: 10.3390/s21238088.

[8] I. Maheshwari and P. Maheshwari, "Effectiveness of Immersive VR in STEM Education," in *2020 Seventh International Conference on Information Technology Trends (ITT)*, Nov. 2020, pp. 7–12, doi: 10.1109/ITT51279.2020.9320779.

[9] H. S. Qorbani, A. Arya, N. Nowlan, and M. Abdinejad, "ScienceVR: A Virtual Reality Framework for STEM Education, Simulation and Assessment," in *2021 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*, Nov. 2021, pp. 267–275, doi: 10.1109/AIVR52153.2021.00060.

[10] M. Bagci *et al.*, "Simulation-Based Learning in Virtual Reality: Three Use Cases from Social Science and Technological Foundations in Terms of Va.Si.Li-Lab," *Technology, Knowledge and Learning*, pp. 1–40, Apr. 2025, doi: 10.1007/s10758-025-09837-7.

[11] A. Chen, "Leveraging Virtual and Augmented Reality to Boost Student Engagement in STEM Disciplines," *Region - Educational Research and Reviews*, vol. 6, no. 12, pp. 61–66, 2024, doi: 10.32629/rerr.v6i12.2999.

[12] J. Huang, M. S. Lucash, R. M. Scheller, and A. Klippel, "Walking through the forests of the future: using data-driven virtual reality to visualize forests under climate change," *International Journal of Geographical Information Science*, vol. 35, no. 6, pp. 1155–1178, Jun. 2021, doi: 10.1080/13658816.2020.1830997.

[13] M. Rana, U. Norbisrath, E. Vainikko, and B. Rossi, "Transforming Affordable Virtual Reality Headsets into Effective Learning Environments," in *2023 International Conference on Intelligent Metaverse Technologies & Applications (iMETA)*, Sep. 2023, pp. 1–8, doi: 10.1109/iMETA59369.2023.10294604.

[14] J. Chen, F. Qian, and B. Li, "An interactive and immersive remote education platform based on commodity devices," in *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, May 2021, pp. 1–2, doi: 10.1109/INFOCOMWKSHPS51825.2021.9484492.

[15] J. C. Servotte *et al.*, "Virtual Reality Experience: Immersion, Sense of Presence, and Cybersickness," *Clinical Simulation in Nursing*, vol. 38, pp. 35–43, Jan. 2020, doi: 10.1016/j.ecns.2019.09.006.

[16] N. Rendevski, K. Veljanovski, and N. Emini, "FPS Performance Factors in Standalone Virtual Reality Applications," in *2023 58th International Scientific Conference on Information, Communication and Energy Systems and Technologies (ICEST)*, IEEE, Jun. 2023, pp. 349–352, doi: 10.1109/ICEST58410.2023.10187389.

[17] Y. Chen and Z. Wu, "A review on ergonomics evaluations of virtual reality," *Work*, vol. 74, no. 3, pp. 831–841, Mar. 2023, doi: 10.3233/WOR-205232.

[18] I. Fatima and K. Sooda, "Enhancing Immersive Virtual Reality Product Experiences: Strategies for Graphics Quality, Performance Optimization, and User-Centric Interfaces," in *2023 IEEE Engineering Informatics*, Nov. 2023, pp. 1–5, doi: 10.1109/IEEECONF58110.2023.10520405.

[19] M. Tytarenko, "Optimizing Immersion: Analyzing Graphics and Performance Considerations in Unity3D VR Development," *Asian Journal of Research in Computer Science*, vol. 16, no. 4, pp. 104–114, Oct. 2023, doi: 10.9734/ajrcos/2023/v16i4374.

[20] J. Hogan, A. Salo, D. E. Rzig, F. Hassan, and B. Maxim, "Analyzing Performance Issues of Virtual Reality Applications," *arXiv preprint*, 2022, doi: 10.48550/arXiv.2211.02013.

[21] R. Grande, J. Albusac, D. Vallejo, C. Glez-Morcillo, and J. J. Castro-Schez, "Performance Evaluation and Optimization of 3D Models from Low-Cost 3D Scanning Technologies for Virtual Reality and Metaverse E-Commerce," *Applied Sciences*, vol. 14, no. 14, pp. 1–33, Jul. 2024, doi: 10.3390/app14146037.

[22] R. F. Murray, K. Y. Patel, and E. S. Wiedenmann, “Luminance calibration of virtual reality displays in Unity,” *Journal of Vision*, vol. 22, no. 13, pp. 1-9, Dec. 2022, doi: 10.1167/jov.22.13.1.
[23] S. Kim, J. Do, J. Kang, and H. Y. Kim, “Rate-Rendering Distortion Optimized Preprocessing for Texture Map Compression of 3D Reconstructed Scenes,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 34, no. 5, pp. 3138–3155, May. 2024, doi: 10.1109/TCSVT.2023.3310522.
[24] C. E. Chu *et al.*, “Enhancing Biology Laboratory Learning: Student Perceptions of Performing Heart Dissection with Virtual Reality,” *IEEE Access*, vol. 12, pp. 76682–76691, 2024, doi: 10.1109/ACCESS.2024.3394713.
[25] Z. Naz, A. Azam, M. U. G. Khan, T. Saba, S. Al-Otaibi, and A. Rehman, “Development and evaluation of immersive VR laboratories of organic chemistry and physics for students education,” *Physica Scripta*, vol. 99, no. 5, p. 056101, May 2024, doi: 10.1088/1402-4896/ad3024.
[26] A. Geris, B. Cukurbasi, M. Kilinc, and O. Teke, “Balancing performance and comfort in virtual reality: A study of FPS, latency, and batch values,” *Software - Practice and Experience*, vol. 54, no. 12, pp. 2336–2348, Dec. 2024, doi: 10.1002/spe.3356.
[27] S. Ye, T. Wu, M. Jarvis, and Y. Zhu, “Digital Reconstruction of Elmina Castle for Mobile Virtual Reality via Point-based Detail Transfer,” *IS and T International Symposium on Electronic Imaging Science and Technology*, vol. 34, no. 1, pp. 401–409, Jan. 2022, doi: 10.2352/EI.2022.34.1.VDA-409.
[28] G. Koulaxidis and S. Xinogalos, “Improving Mobile Game Performance with Basic Optimization Techniques in Unity,” *Modelling*, vol. 3, no. 2, pp. 201–223, Mar. 2022, doi: 10.3390/modelling3020014.
[29] R. Rouhani *et al.*, “Towards benchmarking VR sickness: A novel methodological framework for assessing contributing factors and mitigation strategies through rapid VR sickness induction and recovery,” *Displays*, vol. 84, pp. 1–22, Sep. 2024, doi: 10.1016/j.displa.2024.102807.
[30] R. Wang, W. Hua, Y. Huo, and H. Bao, “Real-time Rendering and Editing of Scattering Effects for Translucent Objects,” *arXiv preprint*, 2022, doi: 10.48550/arXiv.2203.12339.
[31] X. Zhang *et al.*, “Tree Branch Level of Detail Models for Forest Navigation,” *Computer Graphics Forum*, vol. 36, no. 8, pp. 402–417, Dec. 2017, doi: 10.1111/cgf.13088.
[32] Y. Dong and C. Peng, “Real-Time Large Crowd Rendering with Efficient Character and Instance Management on GPU,” *International Journal of Computer Games Technology*, pp. 1–15, Mar. 2019, doi: 10.1155/2019/1792304.
[33] F. Nusrat, F. Hassan, H. Zhong, and X. Wang, “How developers optimize virtual reality applications: A study of optimization commits in open source unity projects,” in *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*, May 2021, pp. 473–485, doi: 10.1109/ICSE43902.2021.00052.

BIOGRAPHIES OF AUTHORS

Cahya Miranto earned his Master’s degree in Informatics and Computer Engineering from Politeknik Elektronika Negeri Surabaya. His research interests lie in game development, virtual reality (VR), game asset development and gamification, with a particular focus on the applications of immersive technology for educational and interactive experiences. He can be contacted at email: cahya@polibatam.ac.id.

Ardiman Firmanda received a Master’s degree in Informatics and Computer Engineering from Politeknik Elektronika Negeri Surabaya. His research interests lie in game development, augmented reality, with a particular focus on the applications of immersive technology for education. He can be contacted at email: ardiman@polibatam.ac.id.

Hestiasari Rante earned her doctorate in Computer Science from the University of Bremen, Germany, in 2020. She is currently a senior lecturer at Politeknik Elektronika Negeri Surabaya, Indonesia, and a visiting scholar at Université Clermont Auvergne, France. As an IEEE Senior Member, she actively contributes to the global academic community, serving as a reviewer for leading international journals and research funding proposals. Her work bridges technology and education, with research interests spanning immersive technologies (AR/VR/XR), interaction design, and the use of games and movies as innovative learning media. She can be contacted at email: hestiasari@pens.ac.id.

Sritrusta Sukaridhoto received the B.E. degree in electrical engineering, computer science program from Sepuluh Nopember Institute of Technology, Indonesia, in 2002 and the Ph.D. degree in Communication Networks Engineering from Okayama University, Japan, in 2013. He joined PENS, Indonesia, as a lecturer in 2002, and he became an Assistant Professor in 2011, respectively. He stayed at Tohoku University, Japan, in 2004, as a visiting researcher. From 2017, he becomes Head of Human Centric Multimedia Lab, received several research grants, and also has several collaborations with government, and industries. He is a technology enthusiast, his research interests include computer networks, human-centric, immersive multimedia, and industrial internet of things. He has received several academic awards, best paper awards, and IEEE Young Researcher Award in 2009. He is a member of IEEE. Further info on his homepage: http://dhoto.lecturer.pens.ac.id/. He can be contacted at email: dhoto@pens.ac.id.

Muhammad Agus Zainuddin earned his doctorate in Informatic, Universite de France-Comte, France, in 2017. He is a lecturer at Politeknik Elektronika Negeri Surabaya (PENS), Indonesia, and currently assigned to be the Head of Study Program in Broadcasting Multimedia Technology. He also serves as reviewer in several international journals. His research interests include multimedia networks, nanocommunications, immersive technology (AR/VR), and blockchain technology. He can be contacted at email: magusz@pens.ac.id.

Haolia Rahman earned his Ph.D. in Mechanical Engineering from Kookmin University, South Korea, in 2019, with a focus on energy efficiency and sustainable technologies. He is a lecturer at the Department of Mechanical Engineering, Politeknik Negeri Jakarta (PNJ), Indonesia, where he has been actively involved in teaching, research, and academic development. His research interests encompass building energy systems, renewable energy technologies, refrigeration and air conditioning, and energy conservation strategies for industrial and residential applications. In addition to his academic roles, he frequently engages in collaborative research projects, industry-based innovation programs, and scholarly publications aimed at advancing knowledge in mechanical and energy engineering. He can be contacted at email: haolia.rahman@mesin.pnj.ac.id.