Crowd-Density Runtime Performance Analysis and NPC Optimisation in a Standalone Virtual Reality Tawaf Simulation

Muhammad Khuarizmi bin Mohd Rapi[1]*, Azuan bin Ahmad[2], Mohd Ilias M Shuhud[2]

[1]Faculty of Artificial Intelligence and Cyber Security, Universiti Teknikal Malaysia Melaka, Malacca, Malaysia

[2]Faculty of Science and Technology, Universiti Sains Islam Malaysia, Nilai, Negeri Sembilan, Malaysia

***Corresponding Author**

DOI: https://doi.org/10.47772/IJRISS.2026.100701062

Received: 11 August 2026; Accepted: 16 August 2026; Published: 20 August 2026

ABSTRACT

Standalone virtual reality (VR) can support portable Tawaf simulation, but its mobile-class resources limit the number of non-player characters (NPCs) that can be moved, animated, and rendered while stable frame delivery is maintained. This paper analyses crowd-density runtime performance, evaluates selected NPC update-rate and animation optimisations, and determines supported density on a PICO 4 Ultra. A quantitative device-based benchmark used six crowd levels (0, 25, 50, 75, 100, and 150 NPCs), seven baseline and optimisation conditions, and three repeated runs per combination, producing 126 runs and 44,430 clean samples. Performance was evaluated using mean frames per second (FPS), mean and percentile frame time, late-frame percentage, memory behaviour, and a predefined practical near-90 Hz rule requiring at least 89 FPS and no more than 11.20 ms mean frame time. The baseline supported 25 NPCs. Updating NPCs every three frames increased support to 75 NPCs, animation simplification supported 100 NPCs, and distance-based animation, animation culling, and combined optimisation supported the highest tested density of 150 NPCs. At 150 NPCs, combined optimisation increased mean FPS by 24.00% and reduced mean frame time by 19.37% relative to baseline. Animation culling was marginally stronger on several high-density indicators, demonstrating that optimisation effects were not fully additive. Strict 90 Hz mean and P95 stability were not achieved, and invalid CPU/GPU timing fields were excluded from processor-level attribution. The study contributes a controlled standalone-headset benchmarking framework and an evidence-based mapping between NPC optimisation configurations and supported Tawaf crowd density. These results provide practical guidance for configuring animated crowds in resource-constrained standalone VR applications intended for ritual familiarisation, training, and related immersive experiences.

Keywords: Crowd Density, NPC Optimisation, Runtime Performance, Standalone Virtual Reality, Tawaf Simulation

INTRODUCTION

Virtual reality (VR) enables spatial, procedural, and interactive experiences that are difficult or costly to reproduce consistently in physical settings. Systematic reviews have reported its use for immersive learning, industrial skills, procedural rehearsal, and professional training, while also stressing that instructional value depends on the interaction between technological design and task requirements (Hamilton et al., 2021; Jongbloed et al., 2024; Makransky & Petersen, 2021; Radhakrishnan et al., 2021; Tusher et al., 2024). Stable technical delivery is a prerequisite for these applications because rendering, tracking, simulation logic, and input processing must be completed within a fixed display interval.

This requirement is particularly demanding on standalone VR headsets. Their processors, graphics units, memory, tracking, display, battery, and thermal management are integrated into a compact mobile device. Although this architecture improves portability, it provides less sustained processing headroom than desktop-tethered VR. Reviews of enabling technologies and empirical optimisation studies identify rendering complexity, asset load, application callbacks, frame rate, latency, and sustained device performance as recurring constraints (Geris et al., 2024; Hu et al., 2021; Rendevski et al., 2023; Sadek Hosny et al., 2020; Sukaridhoto et al., 2023). Therefore, a scene that performs well in a desktop editor cannot be assumed to meet the same frame budget on the release headset.

Tawaf is a relevant crowd-based test case because participants move continuously counter-clockwise around the Kaabah in a shared space. Prior work has modelled integrated Tawaf and Sa'i movements as a mass-crowd process and has represented Hajj procedures in virtual environments (Abidin et al., 2022; Owaidah et al., 2021). The foundational Virtual Tawaf study further demonstrated that the ritual can be treated as a dense, heterogeneous crowd-simulation problem (Curtis et al., 2011). However, pilgrimage-oriented VR research has largely focused on representation, learning, and movement modelling rather than systematic standalone-headset scalability.

In a real-time Tawaf scene, each NPC can introduce route following, position and rotation updates, animation-state evaluation, skeletal transforms, visibility handling, and rendering work. These costs accumulate with crowd size. Contemporary crowd research continues to identify density, heterogeneous behaviour, validation, and computational scalability as core challenges (Dang et al., 2024; Khan & Deng, 2024). Dense trajectory datasets and agent-based forecasting studies additionally show that crowd behaviour is multiscale and sensitive to measurement definitions (Dufour et al., 2025; Makinoshima & Oishi, 2022). The present study does not attempt to reproduce the full behavioural complexity of the Mataf. Instead, it operationalises crowd density as controlled NPC counts and uses Tawaf as a domain-specific runtime workload.

NPC optimisation is motivated by the observation that every agent does not need identical processing quality on every frame. Skeletal-animation studies have used controller simplification, level of detail, culling, and adaptive update strategies to reduce crowd cost (Beacco et al., 2016; Paduraru & Paduraru, 2022; Toledo et al., 2014). Real-time crowd methods also reduce collision, position, or pose-processing cost while retaining visually plausible motion (Gomez-Nogales et al., 2024; Weiss et al., 2019). Most recently, foveated animation has shown that animation rates can be reduced for less visually important agents under tested perceptual conditions (Stancu et al., 2025). These findings support the selected update-rate and animation interventions evaluated here, but they do not establish a supported NPC threshold for a standalone VR Tawaf implementation.

The problem addressed by this paper is the absence of device-based evidence linking controlled NPC crowd density, individual optimisation techniques, frame-rate and frame-time behaviour, and supported-density decisions in a standalone Tawaf simulation. Average FPS alone is insufficient because it can conceal upper-tail instability or repeated budget violations. VR studies have shown that render resolution and frame rate affect performance and user experience (J. Wang et al., 2022, 2023), while foveated-rendering literature demonstrates the broader importance of allocating detail according to perceptual relevance (L. Wang et al., 2023). Asset and application-level optimisation studies similarly show that measured gains depend on scene content and implementation choices rather than on a universal recipe (Grande et al., 2024; Koulaxidis & Xinogalos, 2022).

This paper makes four contributions. First, it presents a controlled PICO 4 Ultra benchmark comprising six NPC densities, seven baseline and optimisation conditions, and 126 repeated runs. Second, it compares update-every-two-frames, update-every-three-frames, animation simplification, distance-based animation, animation culling, and a combined configuration under equivalent crowd loads. Third, it determines the highest supported tested density for each condition using explicit FPS and frame-time requirements. Fourth, it preserves a strict evidence boundary by reporting observable runtime limitation without making CPU-bound or GPU-bound claims when processor-timing fields are invalid. This design follows reproducible performance-evaluation principles that emphasise fixed configurations, repeated measurements, transparent preprocessing, and cautious inference (Papadopoulos et al., 2021). It also reflects evidence from software-repository studies that optimisation decisions should be tied to measurable performance changes rather than generic recommendations (Nusrat et al., 2021).

ISSN No. 2454-6186 | DOI: 10.47772/IJRISS | Volume X Issue VII July 2026

Related Works

VR simulation and standalone constraints

Immersive VR combines stereoscopic rendering, low-latency tracking, interaction, audio, animation, and application logic. The Cognitive Affective Model of Immersive Learning separates technological characteristics from the cognitive and affective mechanisms they may support (Makransky & Petersen, 2021). Although this study does not measure learning, the distinction is technically important: a well-designed Tawaf lesson can still be undermined by unstable frame delivery.

Standalone deployment intensifies this concern. The device must execute the application and the XR runtime within a shared mobile power and thermal envelope. Hu et al. (2021) identify computation, latency, resource allocation, and quality of experience as central VR-system issues. Sadek Hosny et al. (2020) specifically discuss optimisation for standalone headsets, while Sukaridhoto et al. (2023) report that asset optimisation improved average FPS on Quest-class devices. Rendevski et al. (2023) analyse factors affecting standalone FPS, and Geris et al. (2024) connect FPS, latency, and batching with performance and comfort considerations. Collectively, this literature supports target-device measurement rather than reliance on editor-side results.

A refresh rate of 90 Hz provides a nominal frame interval of

$$B90 = 1000 / 90 = 11.111 \text{ ms} \quad (1)$$

where B90 is the strict frame-time reference. Averages close to this value can still coexist with late frames, so the present benchmark reports mean, P95, and P99 frame time together with FPS.

Tawaf and pilgrimage simulation

Tawaf differs from a simple point-to-point pedestrian task because agents repeatedly circulate around a fixed central landmark. The workload persists throughout a benchmark rather than declining as agents exit the scene. Owaidah et al. (2021) modelled integrated Tawaf and Sa’i operations using discrete-event simulation, providing evidence that crowd accumulation and flow are central pilgrimage-management concerns. Abidin et al. (2022) developed a Virtual Hajj learning environment, demonstrating the suitability of interactive 3D media for ritual familiarisation. Curtis et al. (2011) modelled dense, heterogeneous Virtual Tawaf behaviour, establishing an important computational precedent.

These studies address valuable behavioural, operational, or educational questions, but they do not determine how many animated NPCs a contemporary standalone headset can sustain under controlled update-rate and animation configurations. The gap is therefore intersectional: existing pilgrimage studies establish the context, whereas standalone performance and crowd-animation studies establish the technical methods. The present work integrates these areas into a single device-based benchmark.

Crowd and NPC optimisation

Agent-based crowd systems repeatedly execute movement, navigation, collision, animation, and rendering tasks. Reviews by Dang et al. (2024) and Khan and Deng (2024) show that model detail, heterogeneity, contact behaviour, and scalability are closely coupled. Dufour et al. (2025) illustrate the multiscale nature of dense pedestrian trajectories, while Makinoshima and Oishi (2022) demonstrate sequential parameter estimation for agent-based crowd forecasting. For this paper, these findings reinforce two design choices: crowd conditions must be explicitly defined, and conclusions must remain bounded to the implemented model. Animation can become a significant cost when the same skeletal controller is evaluated for many agents. Paduraru and Paduraru (2022) review skeletal techniques for massive crowds. Hierarchical and appearance-based level-of-detail methods reduce detail as perceptual importance decreases (Beacco et al., 2016; Toledo et al., 2014). Position-based crowd simulation reduces large-crowd computation while maintaining real-time behaviour (Weiss et al., 2019), and recent contact-aware work improves dense 3D animation quality (Gomez-Nogales et al., 2024). Stancu et al. (2025) directly show the value of foveated animation-rate control for crowd simulation. The current

ISSN No. 2454-6186 | DOI: 10.47772/IJRISS | Volume X Issue VII July 2026

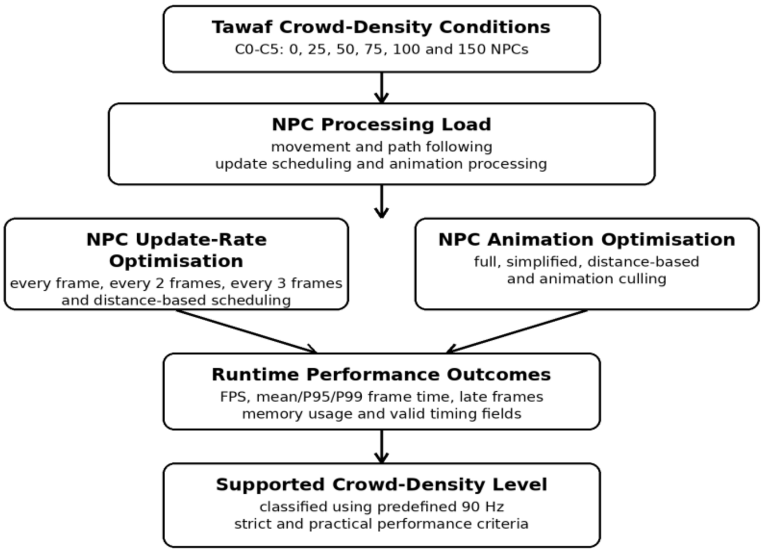
study translates these general principles into four practical Unity conditions: simplified animation, distance-based selection, culling, and combined control.

Performance measurement and research gap

VR performance should be interpreted using more than a single average. J. Wang et al. (2022) report that render resolution affects gameplay performance and simulator sickness, and their later study identifies frame rate as an important factor in user experience and performance (J. Wang et al., 2023). L. Wang et al. (2023) review foveated rendering as a means of reducing peripheral rendering cost. Grande et al. (2024) and Koulaxidis and Xinogalos (2022) demonstrate that model and engine-level optimisation can produce measurable benefits, but the size of those benefits is scene-dependent.

The literature therefore supports device-side profiling, controlled variables, repeated runs, and explicit acceptance criteria. It does not provide a Tawaf-specific mapping from optimisation condition to supported NPC count. Figure 1 summarises the technical framework derived from this gap.

Figure 1: Technical Framework Linking Crowd Density, NPC Optimisation, Runtime Outcomes, and Supported Density



METHODOLOGY

Research design

A quantitative experimental performance-benchmarking design was used. The VR system was the unit of analysis; no human participants were involved. The independent variables were NPC crowd level and optimisation condition. The primary dependent variables were mean FPS and mean frame time, supported by P95/P99 frame time, late-frame percentage, memory behaviour, and pass/fail status.

The design contained six crowd levels and seven optimisation conditions. Each of the 42 combinations was executed three times, producing 126 benchmark runs. The scene, route, NPC prefab, device, build, refresh-rate target, warm-up duration, measurement duration, and logger configuration were held constant. This full matrix allowed each optimisation to be compared at identical density levels.

Testbed and runtime configuration

The simulation was developed in Unity 6.0.31f1 LTS and deployed as an Android ARM64 IL2CPP build using the Universal Render Pipeline and PICO Unity XR SDK. The target device was a PICO 4 Ultra running firmware version 5.14.4.U. The headset was configured for a 90 Hz target. Each run contained a 30-s warm-up followed by 180 s of measurement. Table 1 lists the fixed configuration.

Table 1: Fixed Hardware, Software, and Measurement Configuration

Item	Configuration	Item	Configuration
Headset	PICO 4 Ultra	Firmware	V5.14.4.U
Engine	Unity 6.0.31f1 LTS	Rendering	Universal Render Pipeline
XR stack	PICO Unity XR SDK	Build	Android, IL2CPP, ARM64, ASTC
Target refresh rate	90 Hz	Timing reference	11.111 ms per frame
Warm-up	30 s	Measurement	180 s per run
Profiling	Unity Profiler and device tools	Logging	Custom condition-labelled CSV logger

The environment represented a controlled Kaabah-centred Tawaf area. NPCs followed the same circular route and used the same movement parameters across conditions. Figure 2 shows the test environment, while Figure 3 presents the main component interaction. The Tawaf environment is shown in colour, while the architecture diagram uses line art for clarity.

Figure 2: Controlled Standalone VR Tawaf Environment

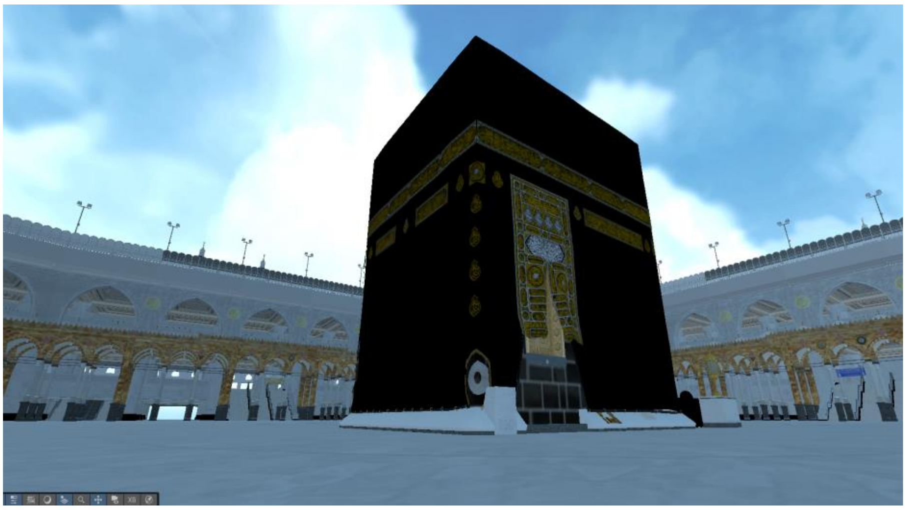
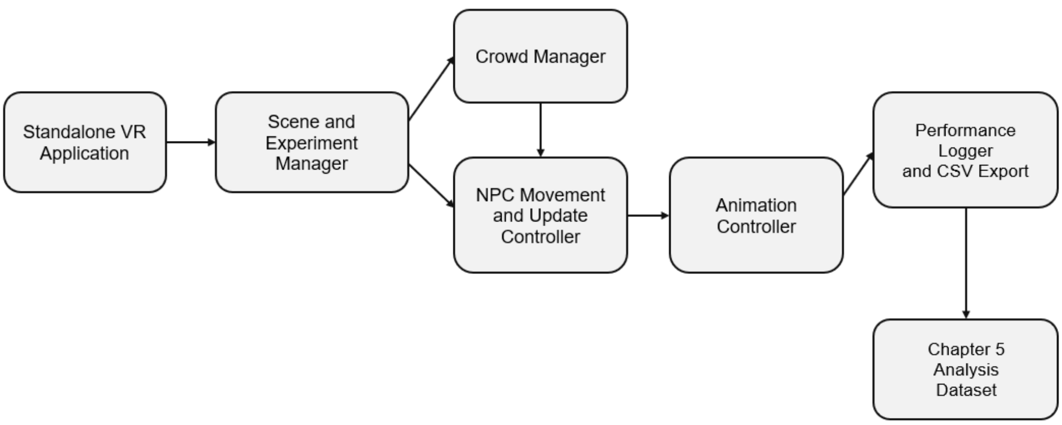


Figure 3: System Architecture of the Standalone VR Tawaf Benchmark



ISSN No. 2454-6186 | DOI: 10.47772/IJRISS | Volume X Issue VII July 2026

Crowd levels and optimisation conditions

Crowd density was operationalised as an active NPC count rather than pilgrims per square metre. The levels were C0=0, C1=25, C2=50, C3=75, C4=100, and C5=150 NPCs. C0 provided the environment-only reference, and C5 was the maximum stress-test level included in the thesis experiment.

Table 2 defines the optimisation conditions. O0 was the every-frame, full-animation baseline. O1 and O2 altered update frequency while retaining full animation. O3–O5 retained the baseline movement-update mode while changing animation processing. O6 integrated adaptive update and animation control.

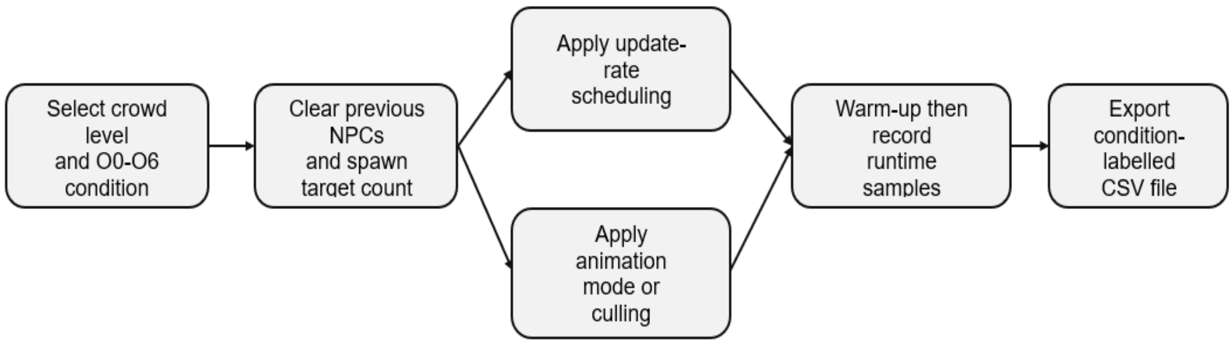
Table 2: Implemented Optimisation Conditions

Code	Condition
O0	Every-frame update with full animation (baseline)
O1	NPC movement and logic updated every two frames
O2	NPC movement and logic updated every three frames
O3	Simplified Animator controller
O4	Distance-based animation selection
O5	Animation culling
O6	Combined adaptive update and animation optimisation

For O1 and O2, elapsed time accumulated across skipped frames was applied during the next selected update. This preserved intended movement speed and prevented a misleading performance gain caused merely by slower NPC motion. O3 reduced controller states and transitions for repetitive walking. O4 applied animation quality according to distance thresholds of 6 m and 14 m. O5 used animation culling to avoid unnecessary off-screen transform updates. O6 combined adaptive movement processing with distance-aware animation control.

Figure 4 illustrates the controlled sequence from condition selection through crowd generation, processing, frame measurement, and CSV export.

Figure 4: Controlled NPC Processing and Measurement Pipeline



Metrics and decision criteria

Mean FPS measured delivered frame rate. Mean frame time measured the average duration of a sample window. P95 and P99 characterised upper-tail instability, and late-frame percentage represented the share of samples above the strict 11.111 ms reference. Memory was treated as a supporting stability indicator.

ISSN No. 2454-6186 | DOI: 10.47772/IJRISS | Volume X Issue VII July 2026

The primary supported-density classification used a practical near-90 Hz criterion. A condition passed only when

$$\bar{F} \geq 89 \text{ FPS} \;\wedge\; \bar{T} \leq 11.20 \text{ ms} \quad (2)$$

where F is mean FPS and T is mean frame time. The strict reference remained 90 FPS with mean and P95 frame time no greater than 11.111 ms. The practical criterion was used to identify supported density; it was not interpreted as perfect strict compliance.

Percentage improvement relative to baseline was calculated as

$$\Delta F = ((F_o - F_0) / F_0) \times 100\% \quad (3)$$

while frame-time reduction was

$$\Delta T = ((T_0 - T_o) / T_0) \times 100\% \quad (4)$$

where subscript 0 denotes O0 and o denotes an optimised condition at the same crowd level.

CPU and GPU frame-timing fields were retained in the logger but returned invalid -1 values on the final device build. They were excluded from processor-level attribution. The analysis therefore identifies observable runtime limitation and improvement without claiming that a particular processor stage was the sole cause.

Data collection and quality control

Every run used the same preparation sequence: load the target condition, reset and spawn the required NPC count, allow 30 s for warm-up, record for 180 s, and export a uniquely labelled CSV file. Startup rows were removed consistently. Metadata fields were checked against filename labels, and duplicate or missing condition files were assessed before analysis.

The raw dataset contained 44,556 rows. Removal of one startup row per run removed 126 rows and produced 44,430 clean samples. All 42 condition combinations and 126 files were present, with no duplicate condition files. Sample rows described temporal behaviour within a run, whereas the run remained the independent replication unit. The results are therefore descriptive and decision-based; they do not treat all 44,430 rows as statistically independent observations.

Table 3: Dataset Structure and Quality-Control Summary

Item	**Value**
Crowd levels	6
Optimisation conditions	7
Condition combinations	42
Repeated runs per combination	3
CSV files	126
Raw sample rows	44,556
Startup rows removed	126
Clean sample rows	44,430

ISSN No. 2454-6186 | DOI: 10.47772/IJRISS | Volume X Issue VII July 2026

Item	Value
Missing / duplicate combinations	0 / 0

EVALUATION AND RESULTS

Baseline crowd-density performance

Table 4 shows the O0 relationship between NPC count and runtime performance. C0 and C1 passed the practical criterion, whereas failure began at C2. The baseline therefore supported 25 NPCs. At C2, mean FPS remained 89.24, but mean frame time reached 11.208 ms and exceeded the 11.20 ms boundary. This divergence demonstrates why FPS and frame time must be interpreted together.

Performance deteriorated more strongly from C3 onward. Mean FPS decreased from 89.98 at C0 to 72.51 at C5, while mean frame time increased from 11.114 to 13.793 ms. At C5, P95 and P99 were 14.078 and 14.230 ms, respectively, and all samples exceeded the strict 90 Hz budget. The environment-only condition was therefore not the principal observed limitation; degradation emerged as movement, animation, object management, and rendering-related work accumulated across the NPC population.

Table 4: Baseline Runtime Performance Across Crowd-Density Levels

Lvl	NPC	FPS	FT (ms)	Status
C0	0	89.98	11.114	P
C1	25	89.99	11.113	P
C2	50	89.24	11.208	F
C3	75	84.56	11.828	F
C4	100	80.90	12.362	F
C5	150	72.51	13.793	F

Update-rate optimisation

O1 did not increase supported density. It passed at C0–C1 and failed from C2, matching the 25-NPC baseline capacity. At C3, O1 contained a shorter run and an extreme recorded frame-time event, producing a mean of 21.054 ms. The run was retained because no predefined exclusion rule classified it as invalid, but the anomaly was not treated as representative of the general O1 trend.

O2 provided a clear but partial improvement. It passed at C2 and C3, raising supported density to 75 NPCs. At C4 and C5, however, its 86.49 and 78.25 FPS values remained outside the practical criterion. At 150 NPCs, O2 improved FPS by 7.92% and reduced mean frame time by 7.35% relative to O0, but all samples still exceeded the strict budget. Update reduction alone therefore redistributed movement and logic work without eliminating the high-density animation and rendering burden.

Animation optimisation

Animation-focused conditions produced larger gains at medium and high densities. O3 passed through C4 and supported 100 NPCs. At C5, it increased mean FPS from 72.51 to 88.11 and reduced mean frame time from 13.793 to 11.355 ms, but still failed both practical requirements.

ISSN No. 2454-6186 | DOI: 10.47772/IJRISS | Volume X Issue VII July 2026

O4 and O5 passed the full tested range. At C5, distance-based animation achieved 89.68 FPS and 11.152 ms. Animation culling achieved 89.94 FPS and 11.118 ms, making it the strongest individual condition on the principal high-density indicators. The improvement grew with NPC count: at low density, the baseline was already near the target, leaving little measurable headroom, whereas at C5 a large amount of unnecessary animation work could be avoided.

Figures 5 and 6 compare mean FPS and mean frame time across the principal conditions. Distinct colours, markers, and line styles are used so that the figures remain distinguishable in colour and interpretable when printed in greyscale.

Figure 5: Mean FPS Across Baseline and Principal Optimisation Conditions

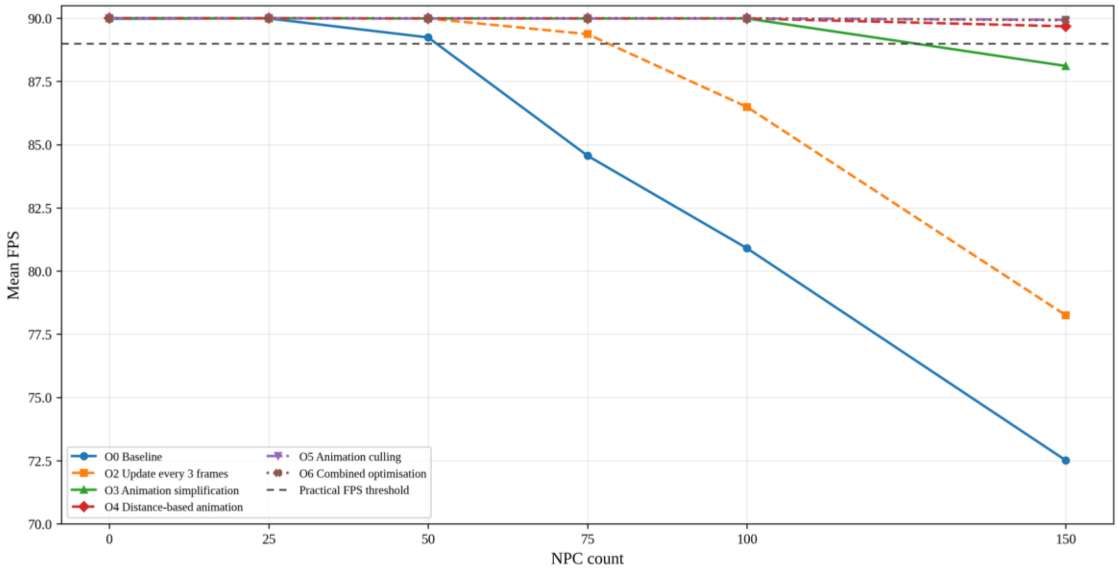
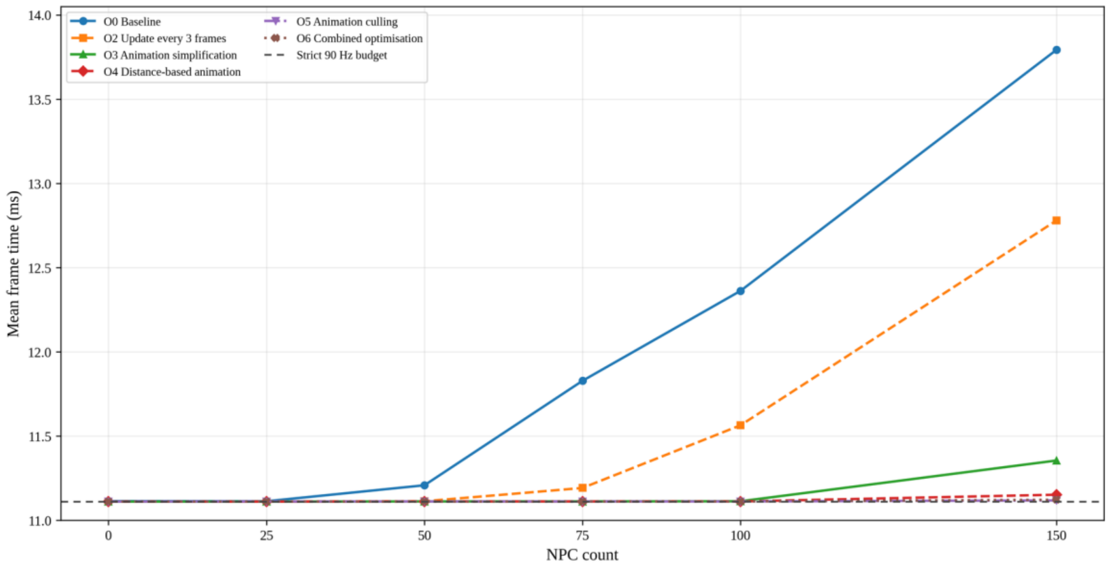


Figure 6: Mean Frame Time Across Baseline and Principal Optimisation Conditions



Combined optimisation and high-density comparison

O6 integrated adaptive update and animation control. It passed every tested crowd level and achieved 89.92 FPS with an 11.122 ms mean frame time at C5. Relative to the C5 baseline, this was a 24.00% FPS improvement and a 19.37% mean frame-time reduction.

Table 5 compares the principal C5 conditions. O6 was highly effective but was not uniformly superior to O5. Animation culling produced a marginally higher mean FPS, lower mean frame time, lower P95/P99, and lower

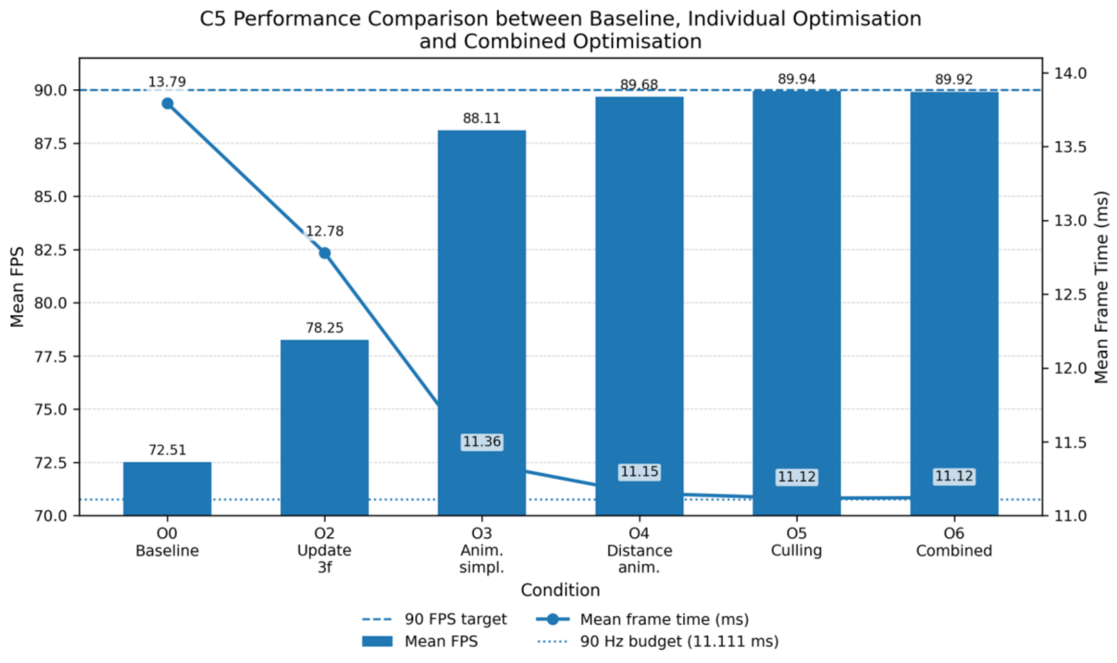
ISSN No. 2454-6186 | DOI: 10.47772/IJRISS | Volume X Issue VII July 2026

late-frame percentage. The differences were small, but they show that optimisation gains were not fully additive. Once culling had removed a substantial share of unnecessary animation work, additional update control produced little measurable benefit near the device's practical performance ceiling.

Table 5: Performance Comparison at the Highest Tested Density (C5, 150 NPCs)

Code	Condition	Mean FPS	Mean FT	P95	P99	Late (%)	Status
O0	Baseline	72.51	13.793	14.078	14.230	100.00	F
O2	Update every 3 frames	78.25	12.780	12.951	13.014	100.00	F
O3	Animation simplification	88.11	11.355	11.869	12.080	85.79	F
O4	Distance-based animation	89.68	11.152	11.383	11.649	62.56	P
O5	Animation culling	89.94	11.118	11.164	11.351	56.87	P
O6	Combined optimisation	89.92	11.122	11.180	11.372	57.95	P

Figure 7: C5 Mean FPS and Mean Frame-Time Comparison



Supported crowd-density determination

Table 6 presents the full practical pass/fail matrix. All conditions passed at C0 and C1. O0 and O1 failed from C2, O2 failed from C4, O3 failed at C5, and only O4–O6 passed the highest tested level.

Table 6: Practical Near-90 Hz Pass/Fail Matrix

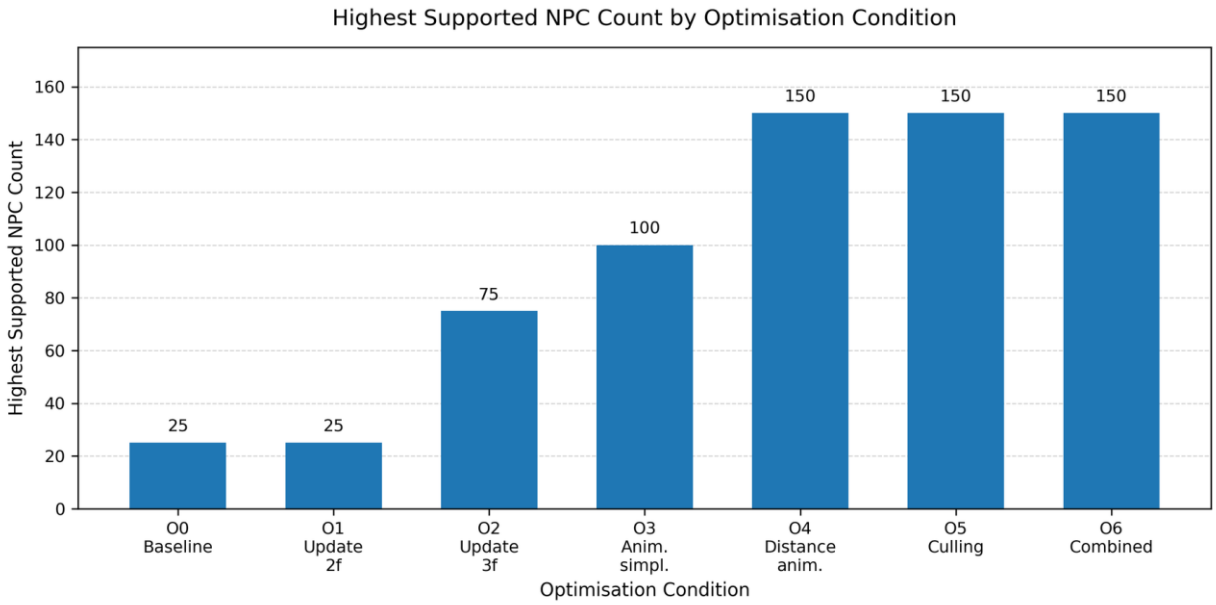
Lvl	NPC	O0	O1	O2	O3	O4	O5	O6
C0	0	P	P	P	P	P	P	P
C1	25	P	P	P	P	P	P	P

ISSN No. 2454-6186 | DOI: 10.47772/IJRISS | Volume X Issue VII July 2026

Lvl	NPC	O0	O1	O2	O3	O4	O5	O6
C2	50	F	F	P	P	P	P	P
C3	75	F	F	P	P	P	P	P
C4	100	F	F	F	P	P	P	P
C5	150	F	F	F	F	P	P	P

The highest supported densities were 25 NPCs for O0 and O1, 75 for O2, 100 for O3, and 150 for O4–O6. Figure 8 visualises this progression. Relative to the baseline, the best update-rate-only condition tripled supported capacity, simplification increased it fourfold, and the strongest animation and combined conditions reached six times the baseline capacity.

Figure 8: Highest Supported NPC Count by Optimisation Condition



None of the conditions met both strict mean and strict P95 90 Hz criteria. The correct interpretation is therefore practical near-90 Hz support for up to 150 NPCs on the tested configuration, not perfect strict 90 Hz stability.

DISCUSSION

Crowd density as an observable runtime load

The baseline findings establish a clear density-dependent performance boundary. Adding 25 NPCs produced little average degradation, but failure began at 50 NPCs and became pronounced from 75 onward. This pattern is consistent with crowd-simulation literature that identifies scalability as a cumulative consequence of per-agent movement, animation, contact, and object-management work (Dang et al., 2024; Khan & Deng, 2024). It also supports the methodological decision to use a controlled, persistent Tawaf route: because agents did not leave the environment, workload remained active throughout every run.

The C2 baseline result is particularly instructive. Mean FPS of 89.24 appeared close to the target, yet mean frame time of 11.208 ms failed the practical rule. A decision based only on rounded FPS could therefore have classified the condition as acceptable. Reporting both metrics produced a more defensible boundary and aligns with evidence that frame rate, render resolution, and temporal stability influence VR performance and experience (J. Wang et al., 2022, 2023).

The evidence supports an association between NPC processing and the observed limitation, but not processor-level causality. Performance changed systematically as the NPC population increased, and targeted reductions in update and animation work improved the same output metrics. However, invalid CPU/GPU timing values prevent a claim that the baseline was definitively CPU-bound, GPU-bound, or animation-thread-bound. This evidence boundary is important because performance studies can over-attribute a result when a profiler field is unavailable or unreliable.

Relative effectiveness of the techniques

Update-rate optimisation provided partial scalability. O2 increased supported density from 25 to 75 NPCs, confirming that recurring movement and logic callbacks contributed measurable cost. O1 did not increase capacity, showing that optimisation strength was not monotonic with a small interval change. The result illustrates why engine advice must be validated experimentally rather than assumed to transfer directly to every scene, a point also reflected in empirical studies of Unity optimisation commits (Nusrat et al., 2021).

Animation optimisation was more influential at high density. O3 supported 100 NPCs, and O4/O5 supported 150. This ranking agrees with skeletal crowd literature: controller complexity, bone transforms, off-screen updates, and uniform animation quality can become expensive when multiplied across many characters (Beacco et al., 2016; Paduraru & Paduraru, 2022; Toledo et al., 2014). The distance-based and culling results also align with perceptual allocation principles in foveated rendering and animation (Stancu et al., 2025; L. Wang et al., 2023). Less visually important agents do not always require the same processing fidelity as near or visible agents.

The non-additive O5/O6 comparison is a second important finding. Combined optimisation might be expected to outperform every individual technique, but O5 was slightly stronger at C5. The likely system-level interpretation is that animation culling had already removed enough unnecessary work for the application to approach its practical display ceiling. Remaining differences were then small and influenced by scheduling overhead, condition logic, or other scene costs. This is consistent with optimisation literature showing that gains depend on interactions among scene assets, draw calls, scripts, and device constraints (Grande et al., 2024; Koulaxidis & Xinogalos, 2022; Sukaridhoto et al., 2023).

Contribution to pilgrimage VR

Existing Hajj and Tawaf work demonstrates the value of virtual environments and crowd modelling (Abidin et al., 2022; Curtis et al., 2011; Owaidah et al., 2021). The present contribution is different: it treats the Tawaf crowd as a measurable standalone runtime workload and maps individual NPC-processing strategies to supported density. The result complements educational and behavioural applications by providing a technical precondition for deployment. A participant evaluation of learning, presence, or comfort should be conducted only after the system has achieved a defensible level of runtime stability.

The reported NPC counts are not a reconstruction of real-world pilgrims per square metre. They are computational load levels for one scene, route, prefab, build, and headset. The highest result means that 150 NPCs passed the practical rule under O4, O5, and O6; it does not establish an absolute upper limit or a universal value for all Tawaf applications. This bounded interpretation protects the paper from conflating technical capacity with real crowd density.

Engineering implications

The benchmark suggests four practical implications for standalone VR developers. First, define an intended crowd range early and test it on the final device. Second, evaluate animation simplification, distance-based control, and culling before assuming that update throttling alone will solve a high-density problem. Third, combine FPS with frame-time distribution and late-frame indicators. Fourth, benchmark each combined configuration because individual gains may overlap rather than accumulate.

The method is transferable beyond Tawaf. Any standalone VR application with numerous animated agents can adopt the same sequence: establish a device baseline, define controlled workload levels, isolate candidate interventions, repeat equivalent runs, validate logged fields, and determine supported capacity using explicit criteria. The numerical thresholds and supported counts remain implementation-specific, but the decision process is reusable.

Benchmark reproducibility and decision protocol

The benchmark was designed so that a future implementation can reproduce the decision process even when its numerical result differs. The essential artefact is not a universal claim that a specific headset supports a fixed number of characters; it is a traceable sequence that connects a defined workload to an acceptance decision. Six elements should be locked before a replication begins.

First, the workload definition must specify the scene, active NPC count, route, movement speed, animation controller, viewing condition, and build configuration. A label such as "100 NPCs" is insufficient when a different prefab, skeleton, shader, route, or visibility pattern can change the cost substantially. In the present experiment, C0–C5 were therefore treated as condition codes whose meaning remained constant across O0–O6.

Second, each intervention must change a known processing factor while preserving the intended task. For update-rate conditions, accumulated elapsed time was applied during the next selected tick so that reduced callback frequency did not reduce travel speed. For animation conditions, the same model and route were retained while controller complexity, distance selection, or culling behaviour changed. This isolation is necessary because an apparent optimisation can otherwise result from a reduced workload rather than more efficient processing.

Third, the replication unit must be distinguished from the sample unit. The logger produced many temporal samples inside each run, which were useful for mean, percentile, and late-frame calculations. These rows did not become independent experimental replications. The three runs per condition remained the independent repeated executions, and the 44,430 clean rows were interpreted as within-run temporal evidence. This distinction prevents artificial inflation of sample size.

Fourth, preprocessing rules should be specified before condition comparison. The current dataset removed one startup row from every file, validated filename and metadata agreement, retained the disclosed O1 anomaly, and did not remove a run merely because it produced an unfavourable value. A future study may adopt a longer warm-up or a different outlier policy, but the rule must be applied consistently and reported with the final file inventory.

Fifth, supported capacity should require an explicit conjunction of criteria. Equation (2) prevented a condition from passing on FPS while materially exceeding the frame-time boundary. Strict mean and P95 status were reported separately so that an operational tolerance was not presented as exact display-budget compliance. For other refresh rates, the strict budget can be recalculated using Equation (1), while any practical tolerance should be justified and labelled separately.

Finally, conclusions must follow the validity of the recorded fields. The present logger returned valid FPS, frame-time, percentile, late-frame, and memory evidence but invalid processor timings. The analysis therefore stopped at observable runtime limitation. A replication that obtains validated CPU and GPU timings may extend the conclusion to bottleneck attribution; one that does not should preserve the same evidence boundary. The resulting protocol is to lock the platform and workload, isolate the intervention, preserve the run as the replication unit, predeclare cleaning rules, apply joint performance criteria, and limit claims to valid metrics.

This protocol also supports cross-device comparison. A second headset can use the same condition matrix and analysis rules while reporting its own supported-density mapping. The relative ordering of O0–O6 can then be compared without assuming that absolute NPC counts transfer across hardware. In this way, the framework separates repeatable methodology from device-specific capacity.

Threats to validity and limitations

Construct validity was strengthened by using both FPS and frame time, percentile indicators, late-frame percentage, memory observation, and explicit pass/fail rules. Nevertheless, the practical criterion contains a small tolerance around the strict 90 Hz reference. A practical pass must not be interpreted as every frame meeting 11.111 ms.

Internal validity was supported by fixed hardware, scene, route, build, viewpoint, duration, and NPC parameters. Each condition was repeated three times. The limited repetition count constrains inferential analysis, and the paper consequently reports descriptive and decision-based results. One anomalous O1 run was retained and disclosed rather than removed after observing its outcome.

Measurement validity is constrained by unavailable CPU and GPU frame-timing fields. FPS, frame-time distribution, late frames, memory, and supported status remained valid, but processor-level bottleneck attribution was not possible. Future work should validate multiple device profiling sources before full collection.

External validity is limited to the PICO 4 Ultra, Unity 6.0.31f1 LTS, PICO XR stack, implemented NPC model, controlled Tawaf route, and tested range. No PCVR or cross-headset comparison was performed. The maximum density was 150 NPCs, so the study demonstrates support at 150 rather than the absolute system ceiling. NPC behaviour was intentionally simplified and did not include full local avoidance, group dynamics, contact forces, or participant interaction. Shader, mesh-LOD, GPU instancing, dynamic resolution, object pooling, and other whole-scene optimisations were outside the experimental scope.

No participant study was included. The paper therefore makes no claims about presence, learning effectiveness, religious instruction quality, simulator sickness, or perceptual acceptability of the reductions. These outcomes require separate ethics-approved research after technical stability is established.

CONCLUSION

This paper investigated crowd-density runtime performance and selected NPC optimisations in a standalone VR Tawaf simulation. The 126-run benchmark showed that the every-frame, full-animation baseline supported 25 NPCs under the practical near-90 Hz criterion. Updating every three frames increased support to 75 NPCs, animation simplification supported 100, and distance-based animation, animation culling, and combined optimisation supported the highest tested level of 150 NPCs.

At 150 NPCs, combined optimisation improved mean FPS by 24.00% and reduced mean frame time by 19.37% relative to the baseline. Animation culling was marginally better on several high-density indicators, demonstrating that optimisation effects were not automatically additive. Animation-related intervention was therefore the most effective category in the implemented system. None of the conditions achieved complete strict 90 Hz mean and P95 stability, and invalid CPU/GPU timings prevented processor-level attribution.

The principal contribution is a controlled, device-based framework that connects NPC crowd levels, individual optimisation strategies, valid runtime evidence, and supported-density decisions in a Tawaf context. Future research should validate processor timings, test longer runs and densities above 150, replicate the matrix across standalone headsets, and evaluate mesh/skeletal LOD, instancing, dynamic resolution, spatial activation, and perceptual crowd techniques. Once stable technical delivery is achieved, participant studies can examine whether the optimised animation remains acceptable for presence, comfort, ritual familiarisation, and learning.

ACKNOWLEDGEMENTS

The authors acknowledge Universiti Sains Islam Malaysia and Universiti Teknikal Malaysia Melaka for academic and institutional support. The authors also acknowledge the technical collaborators who supported development and device testing of the VR Tawaf benchmark.

Data Availability, Ethics And Conflict Of Interest

The study analysed system-generated benchmark logs and involved no human participants; participant ethics approval was therefore not required for the reported experiment. Data supporting the findings are available from the corresponding author upon reasonable request, subject to institutional and thesis-release conditions. The authors declare no conflict of interest.

REFERENCES

1. Abidin, A. R. Z., Masmuzidin, M. Z., Hussien, N. S., & Mohaidin, M. (2022). Virtual Hajj: The development of virtual environment for learning about Hajj and Umrah. AIP Conference Proceedings, 040005. https://doi.org/10.1063/5.0119708
2. Beacco, A., Pelechano, N., & Andújar, C. (2016). A survey of real-time crowd rendering. Computer Graphics Forum, 35(8), 32–50. https://doi.org/10.1111/cgf.12774
3. Curtis, S., Guy, S. J., Zafar, B., & Manocha, D. (2011). Virtual Tawaf: A case study in simulating the behavior of dense, heterogeneous crowds. 2011 IEEE International Conference on Computer Vision Workshops, 128–135. https://doi.org/10.1109/ICCVW.2011.6130234
4. Dang, H.-T., Gaudou, B., & Verstaevel, N. (2024). A literature review of dense crowd simulation. Simulation Modelling Practice and Theory, 134, 102955. https://doi.org/10.1016/j.simpat.2024.102955
5. Dufour, O., Dang, H.-T., Cordes, J., Korbmacher, R., Chraibi, M., Gaudou, B., Nicolas, A., & Tordeux, A. (2025). Dense crowd dynamics and pedestrian trajectories: A multiscale field dataset from the festival of lights in Lyon. Scientific Data, 12(1), 718. https://doi.org/10.1038/s41597-025-04732-3
6. Geris, A., Cukurbasi, B., Kilinc, M., & Teke, O. (2024). Balancing performance and comfort in virtual reality: A study of FPS, latency, and batch values. Software: Practice and Experience, 54(12), 2336–2348. https://doi.org/10.1002/spe.3356
7. Gomez-Nogales, G., Prieto-Martin, M., Romero, C., Comino-Trinidad, M., Ramon-Prieto, P., Olivier, A.-H., Hoyet, L., Otaduy, M. A., Pettré, J., & Casas, D. (2024). Resolving collisions in dense 3D crowd animations. ACM Transactions on Graphics, 43(5), 163:1–163:14. https://doi.org/10.1145/3687266
8. Grande, R., Albusac, J., Vallejo, D., Glez-Morcillo, C., & Castro-Schez, J. J. (2024). Performance evaluation and optimization of 3D models from low-cost 3D scanning technologies for virtual reality and metaverse e-commerce. Applied Sciences, 14(14), 6037. https://doi.org/10.3390/app14146037
9. Hamilton, D., McKechnie, J., Edgerton, E., & Wilson, C. (2021). Immersive virtual reality as a pedagogical tool in education: A systematic literature review of quantitative learning outcomes and experimental design. Journal of Computers in Education, 8(1), 1–32. https://doi.org/10.1007/s40692-020-00169-2
10. Hu, M., Luo, X., Chen, J., Lee, Y. C., Zhou, Y., & Wu, D. (2021). Virtual reality: A survey of enabling technologies and its applications in IoT. Journal of Network and Computer Applications, 178, 102970. https://doi.org/10.1016/j.jnca.2020.102970
11. Jongbloed, J., Chaker, R., & Lavoué, E. (2024). Immersive procedural training in virtual reality: A systematic literature review. Computers & Education, 221, 105124. https://doi.org/10.1016/j.compedu.2024.105124
12. Khan, S., & Deng, Z. (2024). Agent-based crowd simulation: An in-depth survey of determining factors for heterogeneous behavior. The Visual Computer, 40(7), 4993–5004. https://doi.org/10.1007/s00371-024-03503-2
13. Koulaxidis, G., & Xinogalos, S. (2022). Improving mobile game performance with basic optimization techniques in Unity. Modelling, 3(2), 201–223. https://doi.org/10.3390/modelling3020014
14. Makinoshima, F., & Oishi, Y. (2022). Crowd flow forecasting via agent-based simulations with sequential latent parameter estimation from aggregate observation. Scientific Reports, 12(1), 11168. https://doi.org/10.1038/s41598-022-14646-4
15. Makransky, G., & Petersen, G. B. (2021). The cognitive affective model of immersive learning (CAMIL): A theoretical research-based model of learning in immersive virtual reality. Educational Psychology Review, 33(3), 937–958. https://doi.org/10.1007/s10648-020-09586-2
16. Nusrat, F., Hassan, F., Zhong, H., & Wang, X. (2021). How developers optimize virtual reality applications: A study of optimization commits in open source Unity projects. 43rd IEEE/ACM

International Conference on Software Engineering (ICSE), 473–485. https://doi.org/10.1109/ICSE43902.2021.00052

17. Owaidah, A. A., Olaru, D., Bennamoun, M., Sohel, F., & Khan, R. N. (2021). Modelling mass crowd using discrete event simulation: A case study of integrated Tawaf and Sayee rituals during Hajj. IEEE Access, 9, 79424–79448. https://doi.org/10.1109/ACCESS.2021.3083265
18. Paduraru, C., & Paduraru, M. (2022). Techniques for skeletal-based animation in massive crowd simulations. Computers, 11(2), 21. https://doi.org/10.3390/computers11020021
19. Papadopoulos, A. V., Versluis, L., Bauer, A., Herbst, N., von Kistowski, J., Ali-Eldin, A., Abad, C. L., Amaral, J. N., Tuma, P., & Iosup, A. (2021). Methodological principles for reproducible performance evaluation in cloud computing. IEEE Transactions on Software Engineering, 47(8), 1528–1543. https://doi.org/10.1109/TSE.2019.2927908
20. Radhakrishnan, U., Koumaditis, K., & Chinello, F. (2021). A systematic review of immersive virtual reality for industrial skills training. Behaviour & Information Technology, 40(12), 1310–1339. https://doi.org/10.1080/0144929X.2021.1954693
21. Rendevski, N., Veljanovski, K., & Emini, N. (2023). FPS performance factors in standalone virtual reality applications. 58th International Scientific Conference on Information, Communication and Energy Systems and Technologies (ICEST), 349–352. https://doi.org/10.1109/ICEST58410.2023.10187389
22. Sadek Hosny, Y., Salem, M. A.-M., & Wahby, A. (2020). Performance optimization for standalone virtual reality headsets. 2020 IEEE Graphics and Multimedia (GAME), 13–18. https://doi.org/10.1109/GAME50158.2020.9315059
23. Stancu, F.-V., Weiss, T., & Kuffner dos Anjos, R. (2025). Foveated animations for efficient crowd simulation. Proceedings of the ACM on Computer Graphics and Interactive Techniques, 8(1), 1–14. https://doi.org/10.1145/3728306
24. Sukaridhoto, S., Haz, A. L., Fajrianti, E. D., & Budiarti, R. P. N. (2023). Comparative study of 3D assets optimization of virtual reality application on VR standalone device. International Journal on Advanced Science, Engineering and Information Technology, 13(3), 999–1008. https://doi.org/10.18517/ijaseit.13.3.18375
25. Toledo, L., De Gyves, O., & Rudomín, I. (2014). Hierarchical level of detail for varied animated crowds. The Visual Computer, 30(6–8), 949–961. https://doi.org/10.1007/s00371-014-0975-9
26. Tusher, H. M., Mallam, S., & Nazir, S. (2024). A systematic review of virtual reality features for skill training. Technology, Knowledge and Learning, 29(2), 843–878. https://doi.org/10.1007/s10758-023-09713-2
27. Wang, J., Shi, R., Xiao, Z., Qin, X., & Liang, H.-N. (2022). Effect of render resolution on gameplay experience, performance, and simulator sickness in virtual reality games. Proceedings of the ACM on Computer Graphics and Interactive Techniques, 5(1), 1–15. https://doi.org/10.1145/3522610
28. Wang, J., Shi, R., Zheng, W., Xie, W., Kao, D., & Liang, H.-N. (2023). Effect of frame rate on user experience, performance, and simulator sickness in virtual reality. IEEE Transactions on Visualization and Computer Graphics, 29(5), 2478–2488. https://doi.org/10.1109/TVCG.2023.3247057
29. Wang, L., Shi, X., & Liu, Y. (2023). Foveated rendering: A state-of-the-art survey. Computational Visual Media, 9(2), 195–228. https://doi.org/10.1007/s41095-022-0306-4
30. Weiss, T., Litteneker, A., Jiang, C., & Terzopoulos, D. (2019). Position-based real-time simulation of large crowds. Computers & Graphics, 78, 12–22. https://doi.org/10.1016/j.cag.2018.10.008