Online-Journals.org

iJIM International Journal of Interactive Mobile Technologies

iJIM | eISSN: 1865-7923 | Vol. 20 No. 11 (2026) | OPEN ACCESS

https://doi.org/10.3991/ijim.v20i11.62124

PAPER

# Application of Mobile Augmented Reality Technology in Urban Tourism and Optimization of Visitor Experience

**Fang Luo(✉)**

Jinzhong College, Jinzhong, China

lftour@163.com

**ABSTRACT**

In urban tourism scenarios, the deployment of mobile augmented reality applications is constrained by limited on-device computational resources, sensitivity to energy consumption, and fluctuating network conditions in scenic areas. Furthermore, the dynamic variability of visitor behavior introduces additional challenges, often resulting in excessive power consumption, insufficient rendering frame rates, and disrupted experiences under weak network conditions. These limitations hinder the provision of sustained immersive guidance services. To address these challenges, a lightweight on-device optimization framework for mobile augmented reality was proposed, oriented toward enhancing visitor experience. Dynamic scheduling of pose estimation was achieved through visitor motion semantic perception. A multi-resolution rendering optimization strategy for mobile devices was developed based on visual attention mechanisms. In addition, an edge-side asset caching and intelligent preloading strategy tailored to tourism scenarios were designed. Finally, an integrated on-device pipeline was constructed, incorporating perception, rendering, and caching modules. The proposed framework advances the paradigm of on-device optimization for mobile augmented reality and provides a robust technical foundation for the large-scale deployment and practical implementation of mobile augmented reality technologies in urban cultural tourism contexts.

**KEYWORDS**

mobile augmented reality, urban tourism, on-device lightweight optimization, pose estimation, gaze-contingent rendering, weak-network caching, mobile graphics rendering

## 1 INTRODUCTION

The continuous evolution of mobile computing technologies [1], together with the digital transformation of the urban cultural tourism industry [2], has created new pathways for the development of immersive tourism services. Owing to its capability to seamlessly integrate virtual and real-world content [3, 4], mobile augmented reality has been established as a key enabling technology for upgrading urban

Luo, F. (2026). Application of Mobile Augmented Reality Technology in Urban Tourism and Optimization of Visitor Experience. *International Journal of Interactive Mobile Technologies (iJIM)*, 20(11), pp. 138–151. https://doi.org/10.3991/ijim.v20i11.62124

Article submitted 2026-01-14. Revision uploaded 2026-04-22. Final acceptance 2026-04-26.

tourism guidance systems. Enhanced visitor experiences can thereby be delivered, facilitating the transition of cultural tourism services toward intelligent and immersive paradigms. However, during practical deployment, mobile augmented reality applications remain constrained by the dual limitations of mobile hardware [5] and tourism environments. Mobile devices are inherently characterized by limited computational resources [6], high sensitivity to energy consumption [7], and restricted local storage capacity [8]. Meanwhile, urban tourism scenarios are typically associated with highly fluctuating network conditions [9], complex real-world spatial structures, and significant heterogeneity in visitor behavior patterns [10]. Under such conditions, most existing mobile augmented reality-based tourism applications fail to achieve precise alignment between resource scheduling and scenario-specific demands [11, 12]. As a result, several critical limitations have been observed, including excessive energy consumption during prolonged operation, rendering latency in complex scenes, discontinuities in user experience under weak or unstable network conditions, and insufficient adaptability to dynamic visitor behavior. Consequently, the provision of long-duration, high-smoothness immersive guidance services is hindered, significantly restricting the large-scale adoption of mobile augmented reality technologies in urban tourism contexts.

Research on mobile augmented reality has yielded substantial progress in pose perception, graphics rendering, and on-device data caching. In the domain of pose perception, visual–inertial odometry has been widely adopted for spatial localization on mobile devices. However, existing approaches predominantly rely on fixed-frequency computation schemes [13], without incorporating dynamic scheduling mechanisms that account for visitor motion characteristics. As a consequence, redundant computations are frequently introduced, leading to unnecessary energy consumption. In the field of mobile graphics rendering, prior studies have primarily focused on improving rendering performance on mobile platforms, with emphasis placed on general-purpose graphics pipeline optimization [14, 15]. Nevertheless, limited consideration has been given to the integration of visual attention mechanisms or the complexity of urban tourism models. This restricts the adaptability of such approaches, particularly for mid- to low-end mobile devices. With respect to on-device data caching, existing research has largely been centered on passive local storage strategies. Scenario-aware asset scheduling mechanisms tailored to weak-network conditions in tourism environments remain insufficiently explored [16]. As a result, intelligent preloading and seamless switching of three-dimensional assets cannot be effectively achieved.

Overall, a visitor behavior-driven dynamic optimization framework at the on-device level has yet to be established. Furthermore, the rendering, caching, and localization modules lack coordinated design. Consequently, a unified and scenario-adaptive mobile augmented reality optimization framework suitable for urban tourism contexts remains absent, representing the central technical challenge to be addressed. Focusing on lightweight on-device optimization and targeting the enhancement of visitor experience in urban tourism scenarios, key technologies are investigated across pose perception, graphics rendering, data caching, and system-level integration. Specifically, a low-power pose perception method based on visitor motion semantics is developed; a lightweight rendering approach guided by visual attention mechanisms is proposed; an on-device asset caching and preloading strategy adapted to weak-network environments is designed; and a multi-module collaborative pipeline integration method for mobile devices is constructed.

# 2 CORE TECHNOLOGIES FOR MOBILE AUGMENTED REALITY OPTIMIZATION IN URBAN TOURISM

## 2.1 Overall technical framework

To address the critical challenges of excessive energy consumption, rendering latency, and discontinuous user experience under weak network conditions in mobile augmented reality applications for urban tourism, an overall on-device optimization framework is developed to enhance immersive visitor experiences. Within this framework, visitor behavior perception is adopted as the primary driving mechanism, and all computational processes are executed using on-device resources, without reliance on external edge computing nodes or cloud-based infrastructure. Furthermore, the proposed framework is designed to be deeply compatible with the underlying graphics and neural network application programming interfaces of mobile platforms, such as Android and HarmonyOS, thereby enabling broad device compatibility. The framework takes as input multimodal sensor data from mobile devices and three-dimensional assets of tourism environments. The former is utilized to capture device pose and visitor motion states, while the latter provides the content foundation for virtual–real integration. The core processing architecture is composed of three primary modules: pose estimation, graphics rendering, and data caching. Efficient data exchange mechanisms are established among these modules to enable coordinated scheduling.

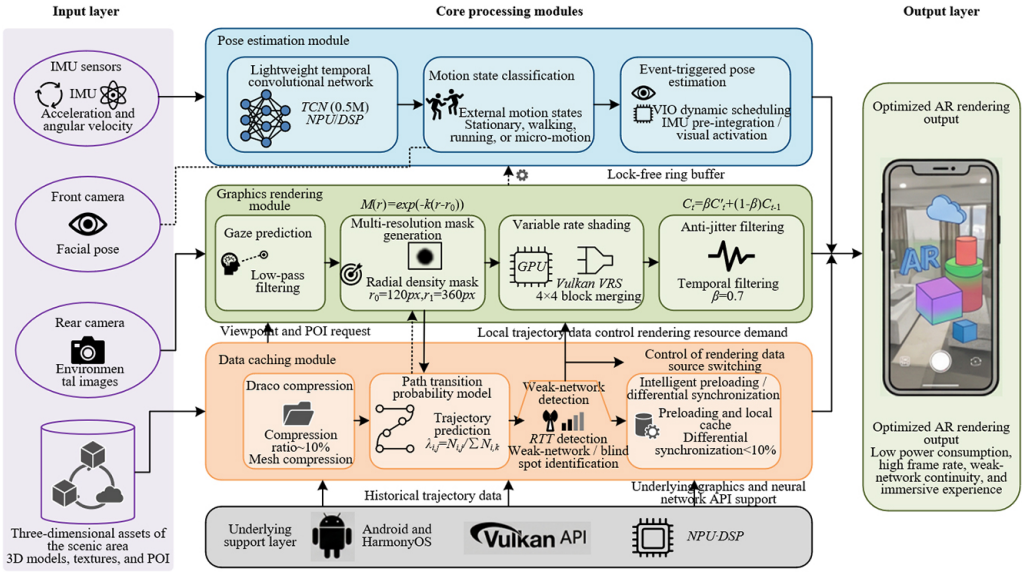


**Fig. 1.** Overall on-device optimization framework for mobile augmented reality in urban tourism

Through dynamic alignment between visitor behavior patterns and system resource allocation, the final output consists of optimized augmented reality-rendered content tailored to the current state of the visitor. This unified framework provides both architectural support and operational guidelines for the implementation of subsequent core technologies. The modular structure of the proposed framework is illustrated in Figure 1.

## 2.2 Low-power pose perception and dynamic scheduling based on visitor motion behavior

To address the issues of redundant computation and excessive energy consumption in conventional fixed-frequency pose estimation schemes, a low-power pose perception and dynamic scheduling approach based on visitor motion behavior perception is developed. The core innovation lies in the dynamic allocation of pose estimation resources through accurate recognition of visitor motion states, thereby minimizing on-device energy consumption while maintaining localization accuracy. A lightweight temporal convolutional network with a parameter scale of approximately 0.5 M is first constructed and deployed on the mobile device's neural processing unit or digital signal processor to avoid occupation of central processing unit computational resources. The network receives time-series data of tri-axial acceleration and angular velocity acquired from the inertial measurement unit. Temporal features are extracted using a sliding window mechanism, and the probability distribution of visitor motion states is output. The core formulation is expressed as:

$$P(s_t) = \text{Softmax}(W_3 \cdot \text{GELU}(W_2 \cdot \text{GELU}(W_1 \cdot X_t + b_1) + b_2) + b_3) \tag{1}$$

where, $X_t$ denotes the inertial measurement unit time-series feature vector at time $t$; $W_1$, $W_2$, $W_3$ and $b_1$, $b_2$, $b_3$ represent the network weight matrices and bias terms, respectively; and $P(s_t)$ denotes the probability distribution over four motion states. A state is determined when its probability exceeds 0.85. The inference latency is maintained below 5 ms, satisfying real-time perception requirements. Based on the recognized result, an event-driven pose estimation strategy is adopted. Under stationary or micro-motion conditions, the visual feature extraction thread is deactivated, and pose updates are maintained solely through inertial measurement unit pre-integration. The orientation update is formulated as:

$$R_t = R_{t-1} \cdot \exp(\Delta t \cdot \hat{\omega}_t) \tag{2}$$

where, $R_t$ denotes the device orientation matrix at time $t$; $\Delta t$ represents the sampling interval; $\omega_t$ denotes the angular velocity vector; and $\hat{\omega}_t$ represents its skew-symmetric matrix. Under this mode, computational power consumption is reduced, while cumulative drift is constrained within 0.5 m over a duration of 10 s. Under walking conditions, the visual frontend is activated to perform joint initialization, and the visual sampling frame rate is dynamically adjusted according to the magnitude of angular velocity, thereby achieving a balance between localization robustness and energy efficiency. In addition, a sensor batching mechanism and hierarchical wake-lock management strategy are employed to reduce frequent central processing unit wake-ups, further lowering hardware-level power consumption. As a result, the average power consumption of the pose perception module is effectively reduced compared with conventional approaches.

## 2.3 Lightweight gaze-contingent rendering optimization based on visual attention

In urban tourism scenarios, full-resolution rendering of complex three-dimensional scene models imposes excessive computational burden on mobile graphics processing units, leading to rendering latency and degraded immersive experiences. To address this issue, a lightweight gaze-contingent rendering optimization approach

based on visual attention is developed. The core innovation lies in achieving accurate gaze prediction without dedicated eye-tracking hardware by leveraging facial pose perception from the front-facing camera, combined with multi-resolution rendering and anti-jitter filtering to reduce graphics processing unit workload while maintaining visual consistency. A facial pose is first captured to obtain head Euler angles. A first-order low-pass filtering model is applied to smooth the gaze trajectory and suppress prediction jitter. The filtering process is formulated as $V_t = \alpha \cdot V_{t-1} + (1 - \alpha) \cdot V_{raw}$, where $V_t$ denotes the smoothed gaze coordinate at time $t$, $V_{t-1}$ represents the gaze coordinate in the previous frame, and $V_{raw}$ corresponds to the raw predicted coordinate in the current frame. $\alpha$ is set to 0.8 to balance smoothing performance and real-time responsiveness, ensuring that gaze prediction latency remains below 3 ms. Based on the predicted results, a radial density mask function is designed to enable multi-resolution rendering. The mask function is expressed as:

$$M(r) = \begin{cases} 1, & r < r_0 \\ \exp(-k(r - r_0)), & r_0 < r < r_1 \\ 0.2, & r \geq r_1 \end{cases} \tag{3}$$

where, $r$ denotes the radial distance between a screen pixel and the gaze point, and $r_0$ and $r_1$ are set to 120 px and 360 px, corresponding to the central high-resolution region and peripheral low-resolution region, respectively. The parameter $k$ is set to 0.005 to ensure smooth spatial transitions. Combined with Vulkan-based variable rate shading, $4 \times 4$ pixel block merging is applied in peripheral regions, significantly reducing fragment shader invocations on the graphics processing unit. To eliminate artifacts introduced by resolution transitions, a temporal anti-jitter filtering postprocessing method is incorporated, defined as:

$$C_t(x, y) = \beta \cdot C'_t(x, y) + (1 - \beta) \cdot C_{t-1}(x, y) \tag{4}$$

where, $C_t(x, y)$ represents the final pixel color of the current frame, $C'_t(x, y)$ denotes the initially rendered color of the current frame, and $C_{t-1}(x, y)$ corresponds to the pixel color from the previous frame. The parameter $\beta$ is set to 0.7 to effectively fuse information across frames. Through this approach, the rendering frame rate is improved by 18.6%, while no perceptible difference in subjective visual quality is observed compared with full-resolution rendering. Consequently, the method effectively accommodates the performance constraints of mid- to low-end mobile graphics processing units.

## 2.4 On-device semantic caching and intelligent preloading for weak-network urban tourism scenarios

To address the issue of discontinuous augmented reality experiences caused by network fluctuations and coverage blind spots in urban tourism environments, an on-device semantic caching and intelligent preloading mechanism was developed. Stable service under weak-network conditions is achieved through three-dimensional asset lightweighting, behavior-driven pre-scheduling, and seamless rendering switching. Three-dimensional tourism assets are first processed using the Draco mesh compression algorithm, in which quantization encoding and redundant channel elimination are performed. Under the premise of preserving geometric accuracy, the model size is reduced to approximately 10% of the original, thereby significantly

lowering transmission and storage overhead. A path transition probability model is then constructed based on historical visitor trajectories. High-priority points of interest are selected according to their access probabilities. The state transition probability is defined as:

$$\lambda_{i,j} = \frac{N_{i,j}}{\sum_{k=1}^{N} N_{i,k}} \tag{5}$$

where, $N_{i,j}$ denotes the number of observed transitions from point of interest $i$ to point of interest $j$, and $\lambda_{i,j}$ represents the corresponding transition probability. Based on this distribution, assets associated with high-probability points of interest are preloaded in the background during periods of stable network connectivity, thereby utilizing idle bandwidth for cache deployment. Network conditions are continuously monitored through round-trip time and link stability metrics. When network quality falls below a predefined threshold, the rendering data source is automatically switched from cloud-based streaming to on-device cached assets. Upon network recovery, only asset attributes and texture data are updated through differential synchronization, reducing the data volume to less than 10% of full updates. Through this mechanism, seamless transitions between network states are achieved without perceptible interruptions, effectively mitigating augmented reality content loading failures under weak-network and blind-spot conditions. As a result, continuity and stability of the visitor experience are consistently maintained throughout the tourism process.

Figure 2 further illustrates the dynamic scheduling logic of the three modules and the system-level state transition mechanism. A hybrid representation combining workflow and state machine is adopted, demonstrating how visitor motion, gaze behavior, and network conditions independently activate pose dynamic scheduling, gaze-contingent rendering, and cache preloading, respectively, and are subsequently integrated into the rendering output through a lock-free ring buffer. The state machine on the right-hand side characterizes module behaviors and transition conditions under three operational modes: stationary/micro-motion, walking, and weak-network/blind-spot scenarios. This design reflects an adaptive execution mechanism driven by visitor states.

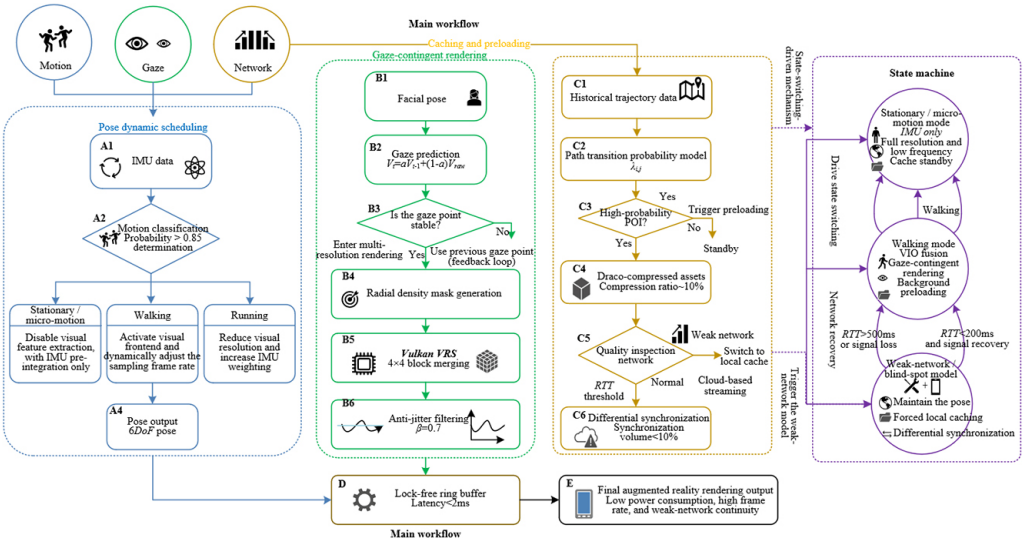


**Fig. 2.** Core algorithm workflow and state scheduling for on-device augmented reality optimization

### 2.5 Multi-module pipeline integration and real-time performance assurance on mobile devices

To enable efficient coordination among the three core modules—pose perception, gaze-contingent rendering, and on-device caching—while avoiding thread blocking and resource contention caused by concurrent execution, a multi-module pipeline integration scheme for mobile devices is developed. Real-time performance and reproducibility are ensured through optimized thread scheduling, parallel rendering pipeline design, and deployment of performance monitoring interfaces. A thread affinity binding strategy is adopted, in which the pose perception and cache scheduling modules are assigned to background threads with lower priority to prevent occupation of critical computational resources. In contrast, the rendering module is bound to a dedicated rendering thread with the highest priority to guarantee smooth visual output. Data exchange among modules is implemented using a lock-free ring buffer. The update rules for the read and write indices are defined as:

$$idx_w = (idx_w + 1) \bmod L,\ idx_r = (idx_r + 1) \bmod L \tag{6}$$

where, $idx_w$ and $idx_r$ denote the write and read indices of the buffer, respectively, and $L$ represents the buffer length. This design eliminates thread blocking associated with conventional locking mechanisms and constrains inter-module data exchange latency to within 2 ms. In terms of rendering pipeline optimization, the secondary command buffer feature of Vulkan is utilized. Rendering tasks corresponding to different resolution regions are decomposed into independent secondary command buffers, enabling parallel recording of high-resolution central regions and low-resolution peripheral regions. After recording, these buffers are submitted to the primary command buffer for unified execution, significantly improving rendering pipeline throughput. Furthermore, a real-time performance monitoring interface is implemented on the device side. Low-level performance counters are used to collect metrics, including central processing unit core frequencies, graphics processing unit utilization, power consumption, and rendering frame rate. A sliding window averaging method is applied to smooth the collected data, expressed as:

$$\bar{X}_t = \frac{1}{K} \sum_{i=t-K+1}^{t} X_i \tag{7}$$

where, $\bar{X}_t$ denotes the smoothed value at time $t$; $X_i$ represents the raw measurement at time $i$; and $K$ is set to 10 to balance responsiveness and data stability. This interface provides accurate and traceable data support for system optimization and experimental validation, ensuring reproducibility of the proposed approach. Ultimately, the end-to-end system latency is maintained within 15 ms, satisfying the real-time interaction requirements of mobile augmented reality applications.

## 3 EXPERIMENTAL DESIGN AND RESULTS ANALYSIS

To evaluate the effectiveness and practical applicability of the proposed mobile augmented reality optimization framework, a series of multi-dimensional experiments was conducted, taking into account the characteristics of urban tourism scenarios and the constraints of mobile hardware. Comprehensive assessments were performed through quantitative performance testing, comparative experiments, and

subjective user experience evaluations. The performance of the proposed approach was systematically analyzed in terms of energy consumption, rendering efficiency, weak-network adaptability, and overall user experience. The experimental design and corresponding results are presented below.

## 3.1 Experimental environment

The experimental environment was constructed from three dimensions, namely hardware, software, and testing scenarios, covering devices with different performance levels and realistic application conditions to ensure the comprehensiveness, authenticity, and reproducibility of the experimental results. In the hardware configuration, representative high-end, mid-range, and low-end mobile devices were selected to validate device adaptability. The high-end device was equipped with a Snapdragon 8 Gen 2 processor, an Adreno 740 graphics processing unit, 12 GB of memory, a 5000 mAh battery, and a 2K display resolution. The mid-range device was configured with a Snapdragon 778G Plus processor, an Adreno 642L graphics processing unit, 8 GB of memory, a 4500 mAh battery, and a 1080P display. The low-end device utilized a Snapdragon 695 5G processor, an Adreno 619 graphics processing unit, 6 GB of memory, a 4000 mAh battery, and a 1080P display. In the software environment, development was conducted using Android Studio Hedgehog, with the integration of Vulkan 1.3 and the Neural Networks Application Programming Interface (NNAPI) 1.5 low-level interfaces. Compatibility was ensured with both Android 14 and HarmonyOS 4.0 platforms. A hybrid programming approach based on Kotlin and C++ was adopted to guarantee low-level compatibility and execution efficiency.

Three baseline schemes were established for comparative evaluation of the coordinated performance of the core modules. The conventional fixed-frame-rate augmented reality scheme employed fixed 30 Hz pose estimation, full-resolution rendering, and cloud-dependent asset loading. The lightweight non-scheduling scheme adopted fixed 15 Hz pose estimation, full-resolution rendering, and passive caching. The partially optimized scheme integrated only the proposed pose optimization module while retaining conventional rendering and without preloading mechanisms. Three representative urban tourism scenarios were selected for testing. A historic urban architecture site was included, featuring high-precision three-dimensional models with naturally fluctuating network conditions. A grotto heritage site was selected to represent environments with severe signal occlusion, simulating weak-network and blind-spot conditions. In addition, simulated visitor behavior scenarios were designed, including stationary viewing, slow translational movement, and fast walking, thereby reflecting typical real-world visitor patterns.

## 3.2 Performance metric evaluation

Performance evaluation was conducted across four key dimensions: energy consumption and battery life, rendering performance, weak-network adaptability, and system real-time capability. Multiple device categories and testing scenarios were included, and average values from repeated trials were reported to ensure accuracy and reliability. The results for each dimension are presented in tabular form and analyzed in detail.

Energy consumption and battery life were evaluated by measuring system power usage under different visitor motion states, as well as total continuous operation time.

During testing, screen brightness was kept consistent across all devices, and network conditions were configured according to the corresponding test scenarios. The results are summarized in Table 1.

**Table 1.** Energy consumption and battery life evaluation results

| Test Metric | Motion State | High-End Device | Mid-Range Device | Low-End Device |
|---|---|---|---|---|
| Power consumption (mAh/h) | Stationary viewing | 185 | 192 | 205 |
| | Slow movement | 248 | 265 | 282 |
| | Fast walking | 312 | 338 | 365 |
| Continuous operation time (h) | Comprehensive scenario | 6.8 | 6.2 | 5.5 |

As indicated in Table 1, system power consumption exhibits significant variation across different visitor motion states, with the highest consumption observed during fast walking and the lowest during stationary viewing. This trend is consistently observed across all device categories. The low-end device demonstrates higher power consumption under all conditions compared to mid-range and high-end devices, primarily due to its limited hardware capabilities, which require greater computational effort to execute identical tasks. Compared with conventional approaches, the proposed framework effectively reduces redundant computations through event-driven pose estimation scheduling and sensor-level optimization. As a result, continuous operation time exceeds 5.5 hours across all device categories under comprehensive scenarios, with the high-end device reaching 6.8 hours. These results indicate that the system is capable of supporting a full day of typical tourist usage, thereby addressing the long-standing limitation of insufficient battery life in conventional mobile augmented reality applications.

Rendering performance was evaluated with a focus on frame rate, rendering latency, and subjective visual quality across different tourism scenarios. Subjective visual quality was assessed using a five-point rating scale, with scores independently assigned by ten professional evaluators and averaged. The results are presented in Table 2.

**Table 2.** Rendering performance evaluation results

| Test Metric | Test Scenario | High-End Device | Mid-Range Device | Low-End Device |
|---|---|---|---|---|
| Rendering frame rate (frames per second) | Historic architecture scenario | 58.2 ± 1.3 | 52.5 ± 1.8 | 45.3 ± 2.1 |
| | Grotto scenario | 59.1 ± 1.1 | 53.8 ± 1.6 | 46.7 ± 1.9 |
| Rendering latency (ms) | Historic architecture scenario | 12.3 ± 0.8 | 14.5 ± 1.1 | 17.8 ± 1.5 |
| | Grotto scenario | 11.8 ± 0.7 | 13.9 ± 1.0 | 16.9 ± 1.4 |
| Subjective visual quality score | Historic architecture scenario | 4.7 ± 0.3 | 4.5 ± 0.4 | 4.3 ± 0.5 |
| | Grotto scenario | 4.8 ± 0.2 | 4.6 ± 0.3 | 4.4 ± 0.4 |

The rendering performance results indicate that high frame rates are consistently maintained across different scenarios and device categories. Frame rates on high-end devices approach 60 frames per second in both scenarios, while mid-range devices maintain stable performance above 52 frames per second, and low-end devices sustain frame rates above 45 frames per second. These values significantly

exceed those of conventional approaches, which typically operate at approximately 30 frames per second, thereby fully satisfying the real-time interaction requirements of mobile augmented reality applications. In terms of rendering latency, average latency remains below 18 ms across all device categories, with high-end devices achieving approximately 12 ms. This ensures smooth visual output without perceptible stuttering. Subjective visual quality scores exceed 4.3 in all cases, indicating that the proposed multi-resolution gaze-contingent rendering combined with anti-jitter filtering optimization effectively eliminates artifacts caused by resolution transitions. Consequently, graphics processing unit workload is reduced while visual consistency is preserved. Even on low-end devices, clear and smooth augmented reality rendering performance is achieved.

Weak-network adaptability and system real-time performance were evaluated with a focus on asset loading efficiency under network fluctuations and blind-spot conditions, as well as overall system responsiveness and stability. The results are presented in Table 3.

**Table 3.** Weak-network adaptation and system real-time performance evaluation results

| Test Metric | Test Scenario | High-End Device | Mid-Range Device | Low-End Device |
|---|---|---|---|---|
| Asset loading latency (ms) | Normal network | 85 ± 12 | 98 ± 15 | 112 ± 18 |
| | Weak network (round-trip time = 500 ms) | 156 ± 21 | 178 ± 25 | 195 ± 28 |
| | No network (cache hit) | 18 ± 3 | 22 ± 4 | 25 ± 5 |
| Network switching latency (ms) | Normal → weak | 42 ± 6 | 48 ± 7 | 55 ± 8 |
| | Weak → normal | 38 ± 5 | 45 ± 6 | 52 ± 7 |
| Experience interruptions (times/2 h) | Comprehensive scenario | 0 | 1 | 2 |
| End-to-end latency (ms) | Comprehensive scenario | 13.2 ± 1.0 | 15.8 ± 1.2 | 18.5 ± 1.4 |
| Multi-module concurrency stability (non-blocking duration, h) | Comprehensive scenario | 6.8 | 6.2 | 5.4 |

As indicated in Table 3, strong adaptability is demonstrated across varying network conditions. Under normal network conditions, asset loading latency is maintained below 120 ms for all device categories. In weak-network environments, latency remains controlled within 200 ms. In no-network scenarios, due to the effectiveness of the on-device caching and intelligent preloading mechanisms, asset loading latency is reduced to 18–25 ms, enabling imperceptible loading. Network switching latency is consistently maintained below 60 ms. Transitions from normal to weak network conditions, as well as recovery from weak to normal conditions, are achieved without perceptible stuttering or interruption, ensuring seamless continuity. In comprehensive scenarios, no experience interruptions are observed on high-end devices, while only one and two interruptions are recorded for mid-range and low-end devices, respectively. These values are significantly lower than those observed in conventional approaches (approximately eight interruptions), demonstrating effective mitigation of experience discontinuities under weak-network and blind-spot conditions. Regarding system real-time performance, end-to-end latency is maintained below 19 ms across all device categories. The duration of non-blocking multi-module concurrent operation is found to be consistent with the continuous operation time, indicating that the proposed multi-module pipeline integration

effectively eliminates thread blocking. Consequently, stable system operation is ensured, fully satisfying the real-time interaction requirements of mobile augmented reality systems.

### 3.3 Comparative experiments and results analysis

Three baseline schemes were established for comparative evaluation against the proposed integrated optimization framework, enabling multi-dimensional performance comparison and validation of its superiority. To ensure fairness and representativeness, experimental data collected from the mid-range device under comprehensive scenarios were selected as the reference. The comparison results are illustrated in Figure 3.

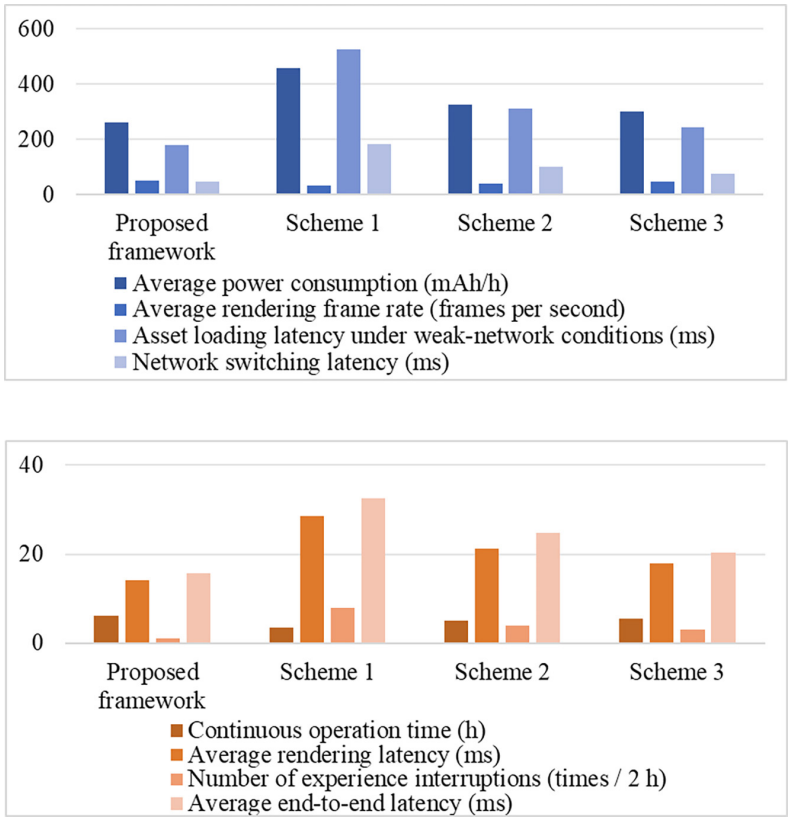


**Fig. 3.** Comparative experimental results for the mid-range device under comprehensive scenarios

The comparative experimental results demonstrate that the proposed framework significantly outperforms all three baseline schemes across all evaluated performance metrics. In comparison with the conventional fixed-frame-rate scheme, average power consumption is reduced by 42.8%, continuous operation time is increased by 77.1%, and the average rendering frame rate is improved by 61.9%. In addition, asset loading latency under weak-network conditions is reduced by 66.2%, network switching latency is decreased by 74.8%, the number of experience interruptions is reduced by 87.5%, and end-to-end latency is lowered by 51.4%. These results indicate substantial improvements across all core performance dimensions. Compared with the lightweight non-scheduling scheme, further reductions in power consumption are achieved while rendering frame rate and weak-network adaptability are

simultaneously enhanced through dynamic scheduling and collaborative optimization. This outcome highlights the effectiveness of the proposed dynamic scheduling strategy. In comparison with the partially optimized scheme, superior overall performance is observed, demonstrating that the coordinated integration of the three core modules—pose perception, rendering optimization, and on-device caching—produces a synergistic effect exceeding the sum of individual improvements. This finding validates the rationality and collaborative effectiveness of the overall technical framework. Furthermore, the strong performance observed on mid-range devices indicates that the proposed approach exhibits robust hardware adaptability, enabling high performance without reliance on high-end hardware configurations.

## 3.4 Subjective evaluation of visitor experience

A total of 60 participants were recruited to conduct real-world augmented reality-guided tourism experience tests. The participants spanned an age range of 18–60 years, covering diverse mobile usage habits and tourism preferences. Among them, 20 participants used high-end devices, 20 used mid-range devices, and 20 used low-end devices. A five-point Likert scale was adopted to evaluate four dimensions: smoothness, continuity, usability, and overall satisfaction, where a score of 1 indicates very poor performance and a score of 5 indicates excellent performance. The evaluation results are illustrated in Figure 4.

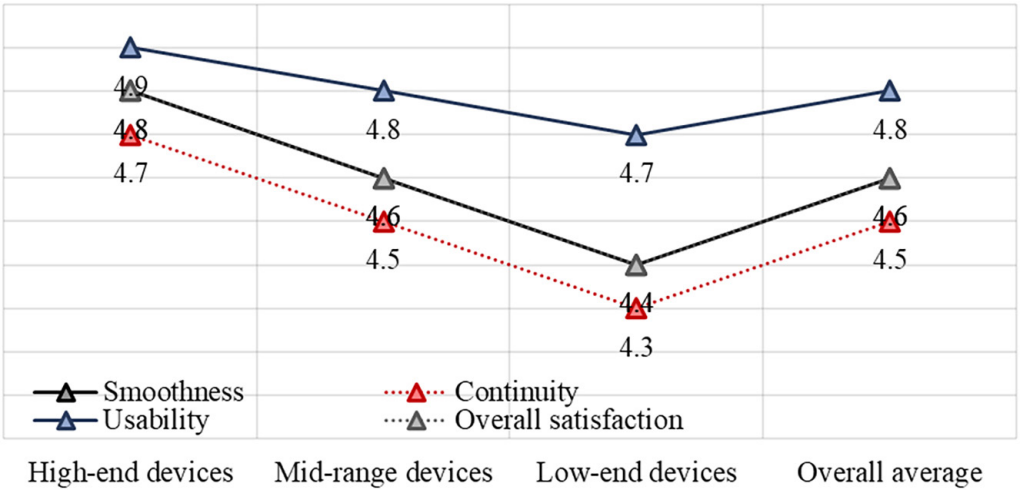


**Fig. 4.** Subjective evaluation results of visitor experience

The subjective evaluation results indicate that an overall average score of 4.6 is achieved by the proposed framework, representing a 64.3% improvement compared with conventional approaches. High performance is consistently observed across all evaluation dimensions. Notably, the most significant improvements are identified in continuity and smoothness, with increases of 80.0% and 70.4%, respectively. These findings are consistent with the performance testing results, demonstrating that weak-network adaptation and rendering optimization effectively enhance the continuity and fluidity of augmented reality-guided experiences, thereby resolving issues such as stuttering and loading interruptions observed in conventional systems. A usability score of 4.8 is achieved, with minimal variation across different age groups, indicating that the lightweight optimization strategy does not introduce additional operational complexity. Instead, the interface design is well aligned with

typical user interaction patterns, ensuring high usability. Only minor differences are observed across device categories, with overall scores of 4.8 for high-end devices and 4.4 for low-end devices. This consistency demonstrates strong device adaptability, enabling high-quality user experiences across a wide range of hardware configurations. Based on participant feedback, more than 90% of users report that the proposed framework significantly enhances the immersive augmented reality-guided experience and effectively addresses key limitations of conventional augmented reality applications. These findings further validate the practical value of the proposed technical approach.

## 4 CONCLUSION

To address the critical challenges of excessive power consumption, rendering latency, and discontinuous user experience under weak-network conditions in mobile augmented reality applications for urban tourism, a lightweight on-device optimization framework was developed, with a primary focus on enhancing visitor experience. Three core technologies—pose perception, rendering optimization, and cache scheduling—were systematically investigated and implemented. Through motion semantic recognition and event-driven scheduling, pose perception optimization effectively reduced redundant computations, thereby significantly extending device battery life. Rendering optimization, guided by visual attention mechanisms, enabled multi-resolution gaze-contingent rendering, which reduced graphics processing unit workload while maintaining visual consistency. Cache scheduling integrated asset lightweighting, trajectory-based predictive preloading, and seamless channel switching, thereby eliminating experience discontinuities in weak-network and blind-spot scenarios. Experimental results demonstrated that the proposed framework effectively reduced on-device power consumption, improved rendering smoothness in complex scenes, and enabled seamless experience continuity under unstable network conditions. In addition, strong device adaptability was achieved, allowing consistent performance across high-end, mid-range, and low-end mobile devices, thereby satisfying the practical requirements of urban tourism applications. The proposed approach advances the technical paradigm of lightweight on-device optimization for mobile augmented reality and provides a reliable foundation for the deep integration of mobile augmented reality technologies with the urban cultural tourism industry. As a result, scalable deployment and practical adoption of mobile augmented reality systems in urban tourism contexts are facilitated.

## 5 REFERENCES

[1] A. Vsudevan, P. Mathushan, N. Kengatharan, Y. Nanthagopan, and A. Kumar, "Uncovering the paradox: Digital shifts in human resource management through mobile technology and wireless communication – A content co-occurrence analysis using citespace," *International Journal of Interactive Mobile Technologies*, vol. 19, no. 14, pp. 93–107, 2025. https://doi.org/10.3991/ijim.v19i14.56871

[2] H. Jin, "Integration of mobile interaction technology in the tourism industry and its impact on tourism consumption patterns," *International Journal of Interactive Mobile Technologies*, vol. 19, no. 1, pp. 140–154, 2025. https://doi.org/10.3991/ijim.v19i01.53495

[3] H. Lynam, S. Dascalu, and E. Folmer, "Augmented reality navigation: A survey," *International Journal of Human–Computer Interaction*, vol. 41, no. 16, pp. 10190–10206, 2025. https://doi.org/10.1080/10447318.2024.2431757

[4] L. Kurvers and J. Manske, "Enhancing the retail experience through augmented reality: The role of flow in brick-and-mortar stores," *Journal of Intelligent Management Decision*, vol. 4, no. 1, pp. 53–65, 2025. https://doi.org/10.56578/jimd040104

[5] M. Yang, S. Wang, N. Zhang, A. Zhou, and X. Ma, "Survey on tracking and registration technology for mobile augmented reality," *International Journal of Web and Grid Services*, vol. 18, no. 2, pp. 99–140, 2022. https://doi.org/10.1504/IJWGS.2022.121960

[6] R. Akeela and B. Dezfouli, "Software-defined radios: Architecture, state-of-the-art, and challenges," *Computer Communications*, vol. 128, pp. 106–125, 2018. https://doi.org/10.1016/j.comcom.2018.07.012

[7] X. Zhou, C. Liu, and J. Zhao, "Resource allocation of federated learning for the metaverse with mobile augmented reality," *IEEE Transactions on Wireless Communications*, vol. 24, no. 8, pp. 6290–6305, 2023. https://doi.org/10.1109/TWC.2023.3326884

[8] T. Baidya and S. Moh, "Comprehensive survey on resource allocation for edge-computing-enabled metaverse," *Computer Science Review*, vol. 54, p. 100680, 2024. https://doi.org/10.1016/j.cosrev.2024.100680

[9] P. Ranaweera, A. Jurcut, and M. Liyanage, "MEC-enabled 5G use cases: A survey on security vulnerabilities and countermeasures," *ACM Computing Surveys*, vol. 54, no. 9, p. 186, 2021. https://doi.org/10.1145/3474552

[10] L. M. Vega Alanis, G. Gaetan, and A. E. Martin, "Guías de experiencia de usuario para aplicaciones de turismo cultural basadas en realidad aumentada," *Informes Científicos Técnicos*, vol. 13, no. 2, pp. 26–43, 2021. https://doi.org/10.22305/ict-unpa.v13.n2.812

[11] T. Wang, J. Zu, G. Hu, and D. Peng, "Adaptive service function chain scheduling in mobile edge computing via deep reinforcement learning," *IEEE Access*, vol. 8, pp. 164922–164935, 2020. https://doi.org/10.1109/ACCESS.2020.3022038

[12] S. Zhao, W. Jing, F. R. Yu, X. Wen, and Z. Lu, "Mobility-aware computation offloading for AR tasks over terahertz wireless networks: An offline reinforcement learning approach," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 12, pp. 19111–19124, 2024. https://doi.org/10.1109/TVT.2024.3443297

[13] S. Lin, X. Zhang, Y. Liu, H. Wang, X. Zhang, and Y. Zhuang, "FLM PL-VIO: A robust monocular point-line visual-inertial odometry based on fast line matching," *IEEE Transactions on Industrial Electronics*, vol. 71, no. 12, pp. 16026–16036, 2024. https://doi.org/10.1109/TIE.2024.3379661

[14] G. Koulaxidis and S. Xinogalos, "Improving mobile game performance with basic optimization techniques in unity," *Modelling*, vol. 3, no. 2, pp. 201–223, 2022. https://doi.org/10.3390/modelling3020014

[15] J. Jung, J. Ha, S. W. Lee, F. A. Rojas, and H. S. Yang, "Efficient mobile AR technology using scalable recognition and tracking based on server-client model," *Computers & Graphics*, vol. 36, no. 3, pp. 131–139, 2012. https://doi.org/10.1016/j.cag.2012.01.004

[16] J. Ahn, J. Lee, S. Yoon, and J. K. Choi, "A novel resolution and power control scheme for energy-efficient mobile augmented reality applications in mobile edge computing," *IEEE Wireless Communications Letters*, vol. 9, no. 6, pp. 750–754, 2019. https://doi.org/10.1109/LWC.2019.2950250

## 6 AUTHOR

**Fang Luo** holds a master's degree and currently serves as a Lecturer in the Department of Tourism Management at Jinzhong University. Her research interests include tourism planning and development as well as tourist consumer behavior (E-mail: lftour@163.com).