Improving Rendering of Dynamic 3D Meshes in WebVR Environments with QoE-aware Level of Detail Adaptation

</> **Nguyen Thi Quynh Ly[1], Duc V. Nguyen[2], Truong Thu Huong[1], Pham Ngoc Nam[3]**

[1] Hanoi University of Science and Technology, Hanoi, Vietnam

[2] Tohoku Institute of Technology, Sendai, Japan

[3] VinUniversity, Hanoi, Vietnam



Web3D 2026

PRESENTATION OUTLINE

01 Motivation & Context

02 Problem Statement & Research Gap

03 Proposed QoE-Aware Framework

04 Experimental Evaluation

05 Conclusion & Future Work

1. MOTIVATION & CONTEXT

Dynamic 3D Mesh in WebVR: Growing Demand

WHY DYNAMIC 3D MESHES IN VR?

Education & Immersive Learning

Networked VR classrooms with animated 3D objects require real-time multi-mesh rendering per session.

Surgical & Professional Training

High-fidelity anatomical and equipment meshes demand detailed geometry and texture under motion.

Digital Cultural Heritage & Virtual Tours

Museum-scale artefact collections rendered across browser-based WebVR without native installation.

WebXR Advantage

Instant browser-based access — no platform-specific native app installation required.



A user improves her driving skills through a personalized VR driving training program. Image courtesy of Lang et al. (2018).

1. MOTIVATION & CONTEXT

Standalone VR: Severe Hardware Constraints

META QUEST 2 — RESOURCE PROFILE

Unified Mobile GPU Memory

No dedicated VRAM (Shared mobile LPDDR5 pool)

High Texture Memory Footprint

A single 4096×4096 texture buffer consumes ~100 MB of GPU memory per dynamic mesh.

Browser WebGL Overhead

JavaScript draw-call translation and WebGL driver latency impose strict per-frame compute ceilings.

Key Observation

Default Meta Quest 2 browser crashes or throttles beyond just 8 full-quality dynamic meshes.



Meta Quest 2 — standalone WebVR headset

2. PROBLEM STATEMENT & ANALYSIS

Four Interrelated Rendering Bottlenecks

ROOT CAUSE ANALYSIS — STANDALONE WEBVR

① Geometry & Texture Memory Pressure

4096×4096 textures consume ~100 MB GPU memory per mesh. High polygon counts compound geometry buffer costs further.

② WebGL Draw Call Bottleneck

JavaScript-to-WebGL translation imposes strict per-frame CPU budget. Driver overhead accumulates with each additional rendered mesh.

③ Unified Memory OOM Termination

No dedicated VRAM — OS terminates the browser process when system-wide memory threshold is exceeded.

④ Perceptually Blind LoD Allocation

Existing approaches use fixed thresholds or equal distribution — ignoring viewing distance, content saliency, and visual perception.

2. PROBLEM STATEMENT & ANALYSIS

The Research Gap: No Joint QoE + Memory-Aware LoD

EXISTING APPROACHES — LIMITATIONS

Perceptual LoD Frameworks

Prior QoE-aware LoD work [Huong 2027] optimizes perceptual quality using subjective models — but targets desktop VR and does not model browser GPU memory constraints.

Standalone VR Rendering Optimization

Existing headset optimization methods [Rendevski 2023; Sadek Hosny 2020] focus on FPS — but ignore perceptual quality and GPU memory during resource allocation.

Adaptive Streaming for XR

Network-oriented streaming approaches [Petrangeli 2019; Quang 2023] minimize bandwidth — but neglect GPU memory cost of combined geometry and texture assets.

Research Gap

No method jointly considers QoE quality, geometry cost, and GPU memory for LoD allocation in WebVR — our work addresses this gap.

3. PROPOSED QOE-AWARE FRAMEWORK

Proposed Architecture

1. Central HTTP Asset Server

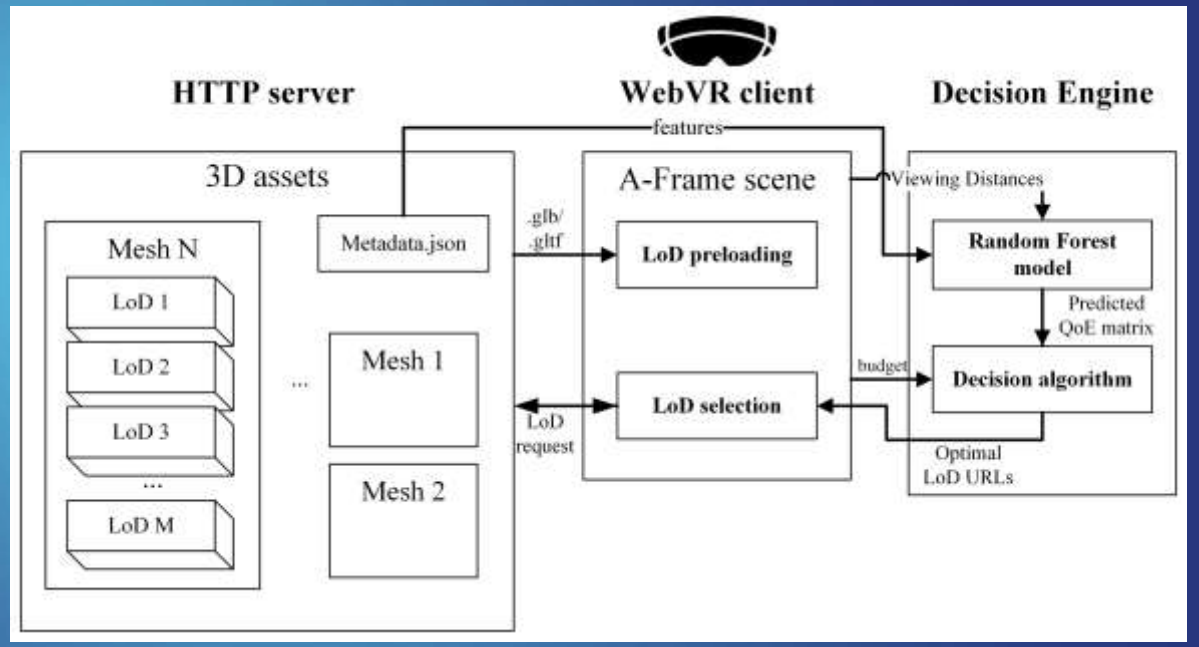
- Central repository for pre-generated 3D geometric assets.
- Hosts 8 LoD glTF/GLB models per object + textures.
- Supplies JSON metadata with glTF-Transform GPU costs.

2. Decision Engine

- Random Forest QoE Model: Estimates visual quality from mesh face count, camera distance, and spatial metrics.
- Optimization Solver: Solves constrained 0-1 BIP problem to maximize cumulative QoE within GPU memory budget CB.

3. WebVR Client (A-Frame Runtime)

- LoD Preloading & Caching: Downloads and retains models.
- Visibility Toggling: Switches active LoD dynamically without re-downloading, preserving animation timestamps.
- Reporting: Sends camera info only every 3.5s ($\Delta d > 0.5m$).



General system architecture
Illustrating interaction between HTTP Server, WebVR Client (A-Frame), and Decision Engine.

3. PROPOSED QOE-AWARE FRAMEWORK
Content Preparation & Metadata

Blender Edge-Collapse Decimation (8 LoDs)

- Generates 8 distinct LoD representations per mesh.
- Face reduction: 20%, 40%, 50%, 60%, 70%, 80%, 90%, and 95%.
- Preserves overall geometric silhouette and visual topology.
- Exported in compact binary glTF (.glb) format for WebVR rendering.

Texture Resolution Downscaling

- Original meshes: 4096×4096 textures (~89.5 MB VRAM each).
- LoD 1–3: 2048×2048 | LoD 4–6: 1024×1024 | LoD 7–8: 512×512.
- **Substantially cuts GPU memory with negligible perceptual loss.**

Spatial Metrics & glTF Metadata

- Blender Rendering: Separates "shading without texture" and "texture without shading" renders to isolate surface features.
- Sobel Edge Filtering: Computes geometry spatial info (SI_geo) and color spatial info (SI_col) based on Nehmé et al. (2023).
- glTF-Transform Profiling: Reports exact texture and buffer memory
sizes (MB), stored in JSON metadata sidecar (Listing 1).

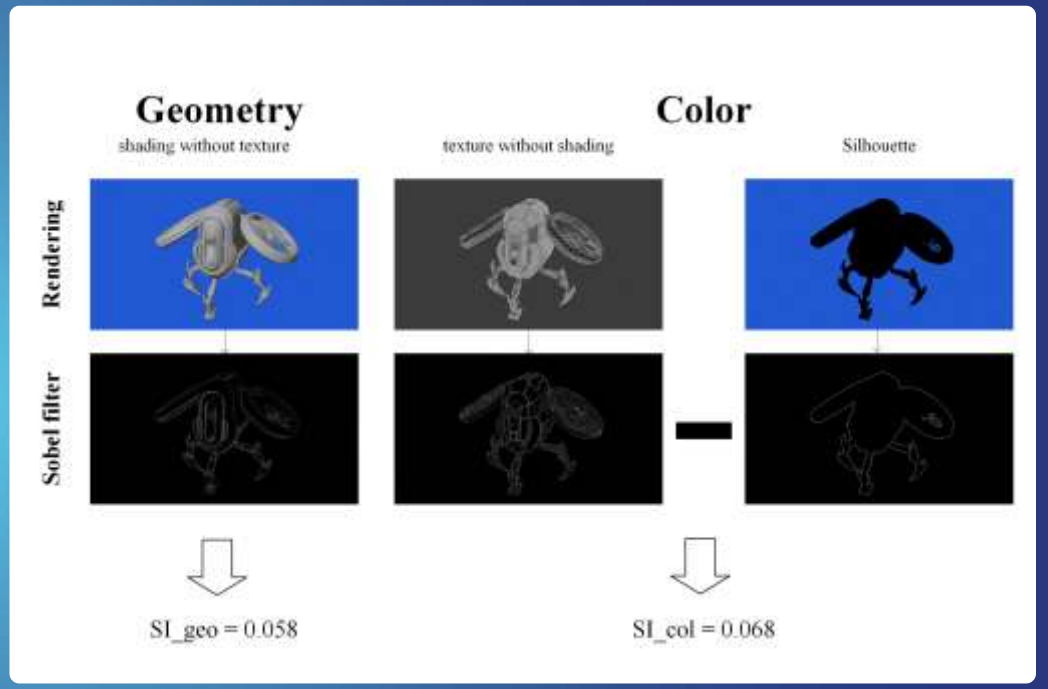


Illustration of the geometry and color spatial feature extraction process
Workflow computing SI_geo and SI_col metrics from dynamic 3D meshes using Blender and Sobel edge filtering (Nehmé et al. 2023).

3. PROPOSED QOE-AWARE FRAMEWORK

Perceptual Modeling: Random Forest QoE Regression

OBJECTIVE QUALITY PREDICTION FORMULATION

$$QoE_{nm} = f_{RF}(f_{nm}, d_n, l_m, s_{geo,\mathrm{n}}, s_{col,n})$$

f_{nm} : face count | d_n : viewing distance (m) | l_m : LoD level (1–8) | $s_{geo,\mathrm{n}}$: geometry spatial info | $s_{col,n}$,n: color spatial info

Subjective Dataset & Methodology

- **Comprehensive Test Stimuli Pool:**

8 dynamic 3D meshes × 8 LoD levels × 5 viewing distances
= 320 total evaluated test stimuli conditions.

Distances tested: 4 m, 8 m, 12 m, 16 m, and 20 m.

- **Standardized Human Rating Protocol:**
Evaluated by 20 human participants via DSIS methodology in immersive VR.

Predictive Accuracy & Validation

0.98
PLCC

0.19
RMSE

3. PROPOSED QOE-AWARE FRAMEWORK
Mathematical Formulation

CONSTRAINED 0-1 BINARY INTEGER PROGRAMMING MODEL

Objective: $\max \sum_{n=1}^{N} \sum_{m=1}^{M} QoE_{nm} \cdot x_{nm}$ (n = 1 … N meshes, m = 1 … M LoD levels)

Subject to: $\sum \sum c_{nm} \cdot x_{nm} \leq C_B$(Budget) | $\sum_m x_{nm} = 1, \ \forall n$ (Single LoD) | $x_{nm} \in 0,1$(Binary)

Decision Variables & Objective

- Binary Decision Indicator (x_{nm}:(

$x_{nm} = 1$ if LoD level m is selected for dynamic mesh n
$x_{nm} = 0$ otherwise.

- Global QoE Maximization:

Evaluates total predicted visual MOS across all N active objects, **favoring higher LoDs for meshes with high perceptual impact.**

N: number of dynamic meshes
M: 8 candidate LoD levels.

Hardware & Model Constraints

- Hard GPU Memory Budget (C_B)

c_{nm} denotes estimated GPU memory (MB) from glTF-Transform.
Total memory of active meshes cannot exceed budget C_B.

- Single LoD Assignment:

Sum over m of $x_{nm} = 1$ guarantees exactly one LoD representation is rendered per dynamic 3D object.

Mathematical Structure: NP-hard 0-1 Knapsack problem variant.

3. PROPOSED QOE-AWARE FRAMEWORK
Real-time Decision Engine

Solver Scaling Challenge

- Exact Branch-and-Bound (BnB) runtime escalates exponentially:
  - N = 32 meshes: ~30 seconds
  - **N = 40 meshes: > 17 minutes (exceeds 3.5s budget!)**
- Unchecked combinatorial explosion prevents real-time VR use.

Proposed Hybrid Decision Strategy

- N ≤ 32: Exact BnB Solver (globally optimal configuration).

- N > 32: Greedy Approximation (highest-QoE within budget).

- Bounded Runtime: Requires only ~10s at maximum N=64 scale.

A-Frame Client-Side Runtime

- Dynamic Entity Caching: Once loaded, models remain cached; transitions simply toggle entity visibility (avoiding re-download).
- Animation Timestamp Restoration: Records timestamp prior to LoD swap and restores it to preserve motion continuity.
- Throttled Network Calls: Distance sent every 3.5s if Δd > 0.5m.



Example of the WebVR scene built using A-Frame
Virtual scene containing multiple animated dynamic 3D meshes running in Meta Quest 2 browser during real-time adaptation.

4. EXPERIMENTAL EVALUATION
Scalability

FULL-QUALITY RENDERING
8 Meshes (Baseline limit)

EQUAL DISTRIBUTION
40 Meshes (5.0× increase)

PROPOSED QOE-AWARE LOD
64 Meshes (8.0× increase)

Table 1: Rendering Strategy	Max Supported Meshes	Resource Allocation Principle	Relative Gain
Full-quality Rendering	**8**	Unreduced 4K textures and complete polygon meshes. Meta Quest 2 browser runs out of GPU memory beyond 8 meshes.	1.0× (Base)
Equal Distribution	**40**	Uniformly divides memory budget across all meshes. Ignores viewing distance, content saliency, and human perception.	**5.0× Gain**
Proposed QoE-aware LoD	**64**	**BIP optimization jointly allocating geometry and texture memory.** Prioritizes salient foreground objects; bounded ~10s decision time.	**8.0× Gain**

Experimental setting: Meta Quest 2 standalone VR browser, A-Frame WebXR environment, memory budget $C_B = 2800$ MB.

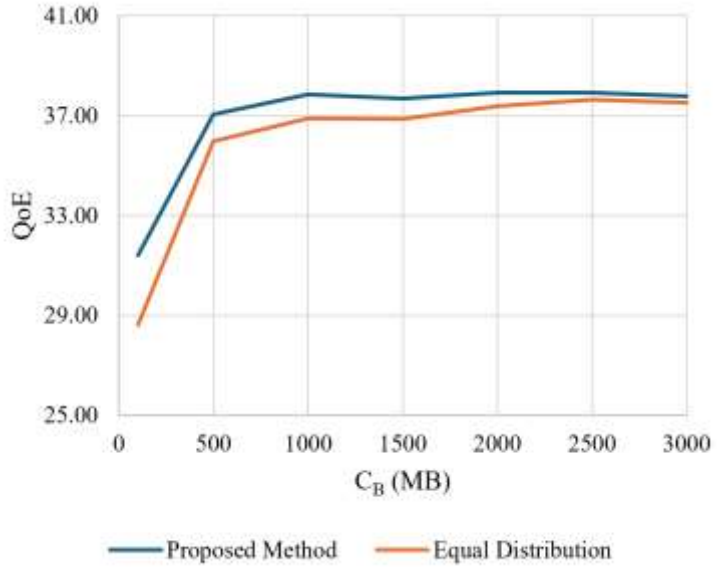
4. EXPERIMENTAL EVALUATION

Runtime Performance & QoE Across Memory Budgets

Performance Metric	Equal Distribution	QoE-aware LoD
Average Frame Rate (FPS)	31.27	**31.34**
Average Draw Calls	361.68	**344.40**
Avg. Texture Count	148.95	**169.98**
Avg. Geometry Count	284.22	**302.05**
Scene Loading Time	2.32 s	**3.14 s**

Table 2: Benchmark with N = 8 meshes, C_B = 2800 MB on Meta Quest 2 browser.

QoE vs. Memory Budget (100 MB – 3000 MB) Summed MOS (N=8)



RUNTIME OVERHEAD FINDINGS (TABLE 2)

- **Zero Frame Rate Penalty**

Maintains 31.34 FPS (baseline 31.27), ensuring no VR motion stutter.

- **Reduced GPU Bottlenecks**

344.40 vs 361.68 draw calls (-4.8%), mitigating WebVR driver overhead.

- **Perceptual Detail Retention**

Retains +14.1% textures and +6.3% geometries on salient models.

QOE BUDGET SENSITIVITY INSIGHTS

- **Strict Perceptual Superiority**

Proposed method dominates Equal Distribution across 100–3000 MB.

- **Maximum Gain in Tight Memory**

At $C_B = 100MB$, QoE rises from 28.63 to 31.42 (+9.7% quality jump).

- **Asymptotic Convergence**

Both curves saturate above 2000 MB (~35.5 MOS); values sum N=8 meshes.

5. CONCLUSION & FUTURE WORK

KEY SCIENTIFIC CONTRIBUTIONS

✓ First Joint QoE-Memory Optimization Model

Formulated LoD allocation as a Binary Integer Programming (BIP) problem optimizing perceptual quality under strict GPU limits.

✓ Demonstrated 8× Mesh Scalability

Scaled capacity from 8 to 64 dynamic meshes.
Maintained stable 31.34 FPS (evaluated at N=8 setup).

✓ Bounded Decision Runtime & Open Source

BnB/greedy solver completes in ~10s at N=64
code released: **github.com/qnhly/webvr-adaptive-lod**

FUTURE RESEARCH AGENDA

Dynamic GPU Compute Workload Adaptation

Incorporate real-time shader execution, geometry vertex processing load, and mobile thermal limits into LoD selection.

User Gaze & Interaction-Aware Saliency

Leverage headset eye-tracking and controller focus to dynamically prioritize foveated LoD allocation for actively inspected objects.

Large-Scale Multi-Room Virtual Environments

Extend framework from single scenes to complex, multi-room virtual environments with continuous user navigation.

Predictive Spatial Asset Streaming

Combine user trajectory prediction with prefetching to eliminate model load delays during dynamic scene transitions.



Thank You!

Web3D

2026

Doha, Qatar