symmetry

MDPI

Review

Mathematical Modeling Techniques in Virtual Reality Technologies: An Integrated Review of Physical Simulation, Spatial Analysis, and Interface Implementation

Junhyeok Lee [1], Yong-Hyuk Kim [2] and Kang Hoon Lee [2,*]

[1] Department of Computer Science, Kwangwoon University, 20, Gwangun-ro, Nowon-gu, Seoul 01897, Republic of Korea; junhyeoklee@kw.ac.kr
[2] School of Software, Kwangwoon University, 20, Gwangun-ro, Nowon-gu, Seoul 01897, Republic of Korea; yhdfly@kw.ac.kr
* Correspondence: kang@kw.ac.kr

Abstract

Virtual reality (VR) has emerged as a complex technological domain that demands high levels of realism and interactivity. At the core of this immersive experience lies a broad spectrum of mathematical modeling techniques. This survey explores how mathematical foundations support and enhance key VR components, including physical simulations, 3D spatial analysis, rendering pipelines, and user interactions. We review differential equations and numerical integration methods (e.g., Euler, Verlet, Runge–Kutta (RK4)) used to simulate dynamic environments, as well as geometric transformations and coordinate systems that enable seamless motion and viewpoint control. The paper also examines the mathematical underpinnings of real-time rendering processes and interaction models involving collision detection and feedback prediction. In addition, recent developments such as physics-informed neural networks, differentiable rendering, and neural scene representations are presented as emerging trends bridging classical mathematics and data-driven approaches. By organizing these elements into a coherent mathematical framework, this work aims to provide researchers and developers with a comprehensive reference for applying mathematical techniques in VR systems. The paper concludes by outlining the open challenges in balancing accuracy and performance and proposes future directions for integrating advanced mathematics into next-generation VR experiences.

Keywords: virtual reality (VR); mathematical modeling; physical simulation; numerical integration; spatial transformation; human–computer interaction

Academic Editor: Calogero Vetro

Received: 18 December 2025
Revised: 24 January 2026
Accepted: 27 January 2026
Published: 30 January 2026

1. Introduction

Virtual reality (VR) imposes unusually tight mathematical and systems constraints. Dynamics must be physically plausible. Spatial estimation must be accurate. Rendering must be photometrically consistent. Interaction must be responsive. All of these must run within millisecond budgets at 90–120 Hz to sustain presence and avoid cybersickness. Even small latency spikes or implausible physics can break immersion. VR is therefore a stringent testbed for methods spanning differential equations, optimization on manifolds, real-time rendering, and human–computer interaction.

Classical foundations remain indispensable. Rigid-body dynamics and constraint methods [1,2] govern motion and contact; illumination and projection models [3–5] determine image formation; quaternions and inverse kinematics (IK) on Special Euclidean

Group (3D) ($SE(3)$) [6–9] enable natural manipulation and embodiment. Recent advances augment rather than replace these tools: NeRFs [10] and real-time variants [11,12] learn geometry and appearance for high-quality view synthesis; differentiable rendering [13] exposes image-level gradients for joint calibration; Physics-Informed Neural Networks (PINNs) [14] embed physical laws in learning.

1.1. Scope, Perspective, and Positioning

We focus on mathematical models that enable real-time VR on consumer hardware (90–120 Hz). The survey is system-oriented. It is organized around four pillars:

(i) Physics-based simulation: integration, constraints, collision, learned surrogates.
(ii) Spatial representation and geometric transforms: quaternions, Lie groups, differentiable transforms, manifold optimization.
(iii) Real-time rendering: Phong/Bidirectional Reflectance Distribution Function (BRDF), GPU pipelines, neural/differentiable rendering, perceptual strategies.
(iv) Interaction modeling: sensing/pose, IK, hand/body tracking, constraint-based manipulation, haptics, comfort-aware control.

Authoritative surveys exist for 3D user interfaces [15], classical rendering/geometry [4,5], inverse kinematics [9], and rigid-body physics [2]. Recent surveys address neural, differentiable, and foveated rendering [10–13,16,17], but typically treat these topics in isolation or outside a VR systems perspective. Our positioning reads physics, tracking/Simultaneous Localization and Mapping (SLAM), rendering, and interaction together, framed by VR-specific constraints (quality–latency–memory trade-offs, motion-to-photon, reprojection error, SSQ) and by cross-module interfaces (e.g., $SE(3)$ tracking feeding differentiable rendering and avatar controllers).

1.2. Contributions and Methodology

This article is a survey; contributions are curatorial and integrative:

(i) We organize stability-oriented practices for VR physics. We clarify when semi-implicit/symplectic integrators are preferable. We explain how constraint projection maintains plausibility under contact. We identify where perceptually gated adaptivity can safely reduce cost. We also situate differentiable simulators within this toolbox.
(ii) We connect literatures on manifold optimization, $SE(3)$ SLAM, differentiable image formation, and neural scene fields. We present an end-to-end view for tracking, calibration, and reconstruction. We include a taxonomy and typical failure modes for avatars and hand–object interaction.
(iii) We review hybrid and neural rendering for dual-eye 90/120 Hz VR. We distill rules of thumb for the quality–latency–memory trade-off (e.g., radiance caching, network factorization, perceptual scheduling).
(iv) We align methods in physics, rendering, and interaction (IK, haptics, personalized avatars) with user-facing and system metrics. We link algorithm families to time-to-grasp, slip rate, embodiment, Absolute Trajectory Error (ATE)/Relative Pose Error (RPE), reprojection error, and motion-to-photon latency.
(v) We compile checklists and a consolidated benchmark table (Appendix A). We outline a near-term (1–2 year) agenda with concrete goals. Targets include numerical energy drift $\leq 1\%$ on standard scenes. We seek average reprojection error < 1 px. We target end-to-end motion-to-photon < 20 ms. We aim for 20–30% lower SSQ under standardized protocols. To ground the discussion in practical device constraints, Appendix A.1 reports a headset specification summary (Table A1).

To provide a high-level roadmap, Figure 1 summarizes the simplified VR system pipeline considered in this survey (hardware → tracking → physics/IK → rendering →

display) and maps each block to the corresponding sections and representative equations. In particular, we later illustrate the numerical stability of the physics update in Figure 2 and detail the rendering pipeline in Figure 3.

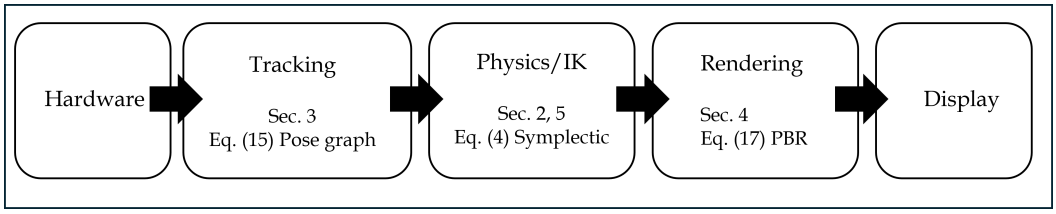

Figure 1. Simplified VR system architecture used as the organizing roadmap of this survey. The pipeline blocks (hardware, tracking, physics/IK, rendering, and display) are mapped to the corresponding sections, with one representative equation highlighted per block (e.g., tracking: Equation (15), physics: Equation (4), rendering: Equation (17)).

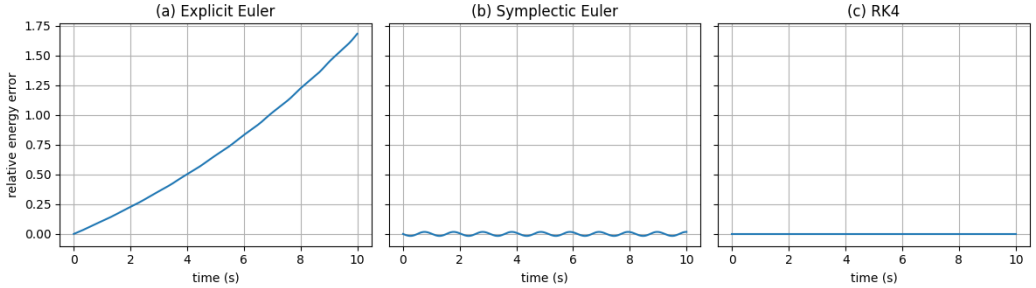

Figure 2. Integration stability for a simple pendulum (1000 steps). (**a**) Explicit Euler updates $v_{k+1} = v_k + \Delta t\, a(q_k, v_k)$ and $q_{k+1} = q_k + \Delta t\, v_k$, yielding monotonic energy drift. (**b**) Symplectic (semi-implicit) Euler follows Equation (4): $v_{k+1} = v_k + \Delta t\, a(q_k, v_k)$ and $q_{k+1} = q_k + \Delta t\, v_{k+1}$, keeping the energy error bounded. (**c**) RK4 reduces drift but is computationally more expensive.

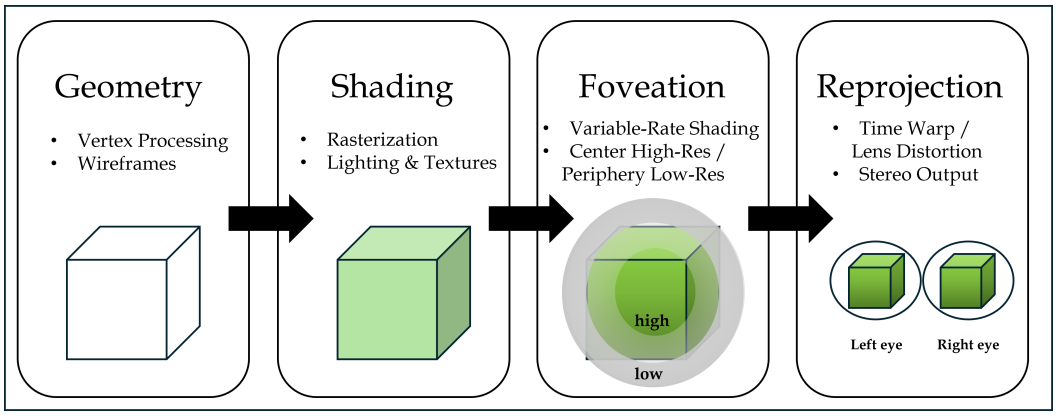

Figure 3. Overview of a modern VR rendering pipeline. Geometry processing transforms high-level primitives into screen-space meshes (Geometry). Shading applies rasterization, lighting, and texturing to generate shaded images (Shading). Foveated rendering allocates higher resolution near the gaze center while reducing detail in the periphery via variable-rate shading (Foveation). Finally, reprojection performs time warp and lens distortion to produce stereo images for the left and right eyes (Reprojection).

Selection criteria. We surveyed 2020–2025 work in SIGGRAPH,TOG, Eurographics, IEEE VR, TVCG, Virtual Reality, CHI, CVPR, ICCV, ECCV, NeurIPS, and ICRA, prioritizing: (i) real-time constraints ($\geq$90 Hz or $\leq$20 ms), (ii) explicit mathematical formulations, (iii) quantitative validation, and (iv) integration/deployment considerations. Foundational references are included where they clarify principles or baselines.

1.3. Reader's Guide and Conventions

Reader's guide. Section 2 covers numerical integration, constraint satisfaction, collision detection/response, and stability–performance trade-offs. Section 3 treats coordinate frames, rotation representations, differentiable transforms, manifold optimization, and pose-graph estimation. Section 4 spans classical to neural/differentiable rendering and perceptual strategies. Section 5 reviews sensing, IK, haptics, and comfort-aware control. Section 6 sets near-term targets; Section 7 synthesizes implications.

Notation and conventions. Bold letters denote vectors/matrices; $SO(3)/SE(3)$ denote rotations/rigid motions; unit quaternions represent orientation. We use standard reprojection/trajectory metrics (ATE/RPE), SSQ for comfort, and per-eye throughput (resolution $\times$ refresh) for system reporting.

2. Physics-Based Simulation in VR

VR immersion depends not only on visual fidelity but also on physically consistent responses under tight frame budgets. This section formalizes equations of motion and constraints, compares time integration schemes under accuracy-stability-structure trade-offs, summarizes contact/friction as non-smooth dynamics, and details perceptual/system-aware adaptivity and learning-augmented simulators relevant to real-time deployment.

Table 1 provides an at-a-glance summary of modeling priorities across representative VR application domains. The ratings indicate relative emphasis (1–5) in terms of typical system requirements rather than a universal ranking. These priorities guide the focus of the subsequent sections on tracking, physics/IK, and rendering, where we detail the corresponding mathematical formulations.

Table 1. Domain-specific modeling priorities in VR applications. Ratings (1–5) indicate relative emphasis under typical requirements; key constraints drive system-level trade-offs discussed in later sections.

Domain	Physics	Tracking	Rendering	Key Constraint
Medical	5	4	3	Haptic fidelity (kHz)
Gaming	3	4	5	90 Hz minimum
Industrial	3	5	4	Sub-cm accuracy
Social VR	2	5	3	Network latency

2.1. VR Strengths and Limitations (at a Glance)

- Strengths
  - Perceptual realism: physically consistent responses increase presence and make user action outcomes predictable.
  - Interaction stability: constraint-aware solvers reduce jitter/failure during grasp/manipulation [2].
  - Composability: shared rigid/soft formulations unify haptics, avatars, and environment logic.
- Limitations
  - Strict frame budgets: solver work must fit $\approx$ 11.1 ms @ 90 Hz (8.3 ms @ 120 Hz); physics LOD/scheduling are mandatory [18,19].
  - Stability–accuracy trade-off: Semi-implicit/projection schemes preserve stability but bias trajectories. Higher-order non-symplectic methods are costlier [2,20].
  - Contact/constraint overhead: non-smooth friction/contact induces LCP/SOCP solves; under-resolved contacts cause visible interpenetration/haptic artifacts [21,22].

– Latency coupling: physics–tracking–render clocks interact; motion-to-photon and prediction errors degrade comfort [5].

2.2. Equations of Motion and Constraints

This subsection states the equations of motion for a rigid multibody system and how equality/inequality constraints enter the model. We fix notation for the state (q, v), separate model terms M, C, f_{ext}, and introduce Lagrange multipliers for constraints; these definitions will be used throughout the section.

Classical physically based modeling formulations established the rigid-body dynamics and constraint foundations that still underpin real-time physics engines used in modern VR [23].

Let generalized coordinates $q \in \mathbb{R}^n$ and velocities $v = \dot{q}$. In the absence of constraints, the equations of motion read

$$M(q)\,\dot{v} + C(q, v) = f_{\text{ext}}(q, v, t), \tag{1}$$

with mass matrix M, Coriolis/gyroscopic term C, and external forces f_{ext}.

Intuitively, Equation (1) is the n-DOF generalization of Newton's law: $M(q)$ maps accelerations to forces, while $C(q, v)$ gathers velocity-dependent inertial effects that become prominent in coupled or fast motions.

Holonomic constraints $g(q) = 0$ (e.g., joints) and unilateral nonpenetration $g_n(q) \geq 0$ (contacts) impose algebraic conditions. Introducing Lagrange multipliers λ for the constraints yields the constrained system

$$\begin{aligned} M(q)\,\dot{v} + C(q, v) &= f_{\text{ext}}(q, v, t) + J(q)^{\top}\lambda, \\ g(q) = 0, \qquad J(q) &= \partial g/\partial q, \end{aligned} \tag{2}$$

where J is the constraint Jacobian. Constraint drift can be reduced by Baumgarte stabilization (enforcing $\dot{g} + \alpha g = 0$ in the integrator) or by post-stabilization, i.e., projecting (q, v) back to the constraint manifold after each step [1,2].

A common post-stabilization step applies a minimal correction that projects the tentative configuration $\tilde{q}$ back onto the constraint manifold:

$$q \leftarrow \tilde{q} + \Delta q, \qquad \Delta q = -M(\tilde{q})^{-1} J(\tilde{q})^{\top} \Big(J(\tilde{q})\, M(\tilde{q})^{-1} J(\tilde{q})^{\top} \Big)^{-1} g(\tilde{q}), \tag{3}$$

where $J(q) = \partial g/\partial q$.

2.3. Time Integration: Accuracy, Stability, and Structure

This subsection contrasts commonly used time-stepping schemes for interactive VR physics and graphics. We focus on how order of accuracy, stability, and structure preservation trade off under real-time budgets. Let Δt be the fixed timestep and let $a(q, v)$ denote the instantaneous acceleration implied by forces and constraints.

Explicit Euler. The explicit Euler integration is formulated as follows:

$$v_{k+1} = v_k + \Delta t\, a(q_k, v_k), \qquad q_{k+1} = q_k + \Delta t\, v_k. \tag{4}$$

It is first-order accurate and cheap, but it tends to drift in energy over long horizons.

Symplectic (semi-implicit) Euler. The symplectic Euler integration is formulated as:

$$v_{k+1} = v_k + \Delta t\, a(q_k, v_k), \qquad q_{k+1} = q_k + \Delta t\, v_{k+1}. \tag{5}$$

This scheme preserves a discrete symplectic structure and often exhibits bounded long-horizon energy error for Hamiltonian systems. To illustrate the long-horizon stability of Equation (4), Figure 2 compares the relative energy error of a simple pendulum using explicit Euler, symplectic (semi-implicit) Euler (Equation (4)), and RK4 under the same step size Δt.

Position Verlet. This simple equation gives the position Verlet method:

$$q_{k+1} = 2q_k - q_{k-1} + \Delta t^2 \, a(q_k). \tag{6}$$

It is second-order accurate and symplectic for conservative forces and is widely used for cloth and soft-constraint models.

High-order non-symplectic (RK4). Classical RK4 is fourth-order accurate per step but non-symplectic and costlier; it suits small, nonstiff subsystems (e.g., camera oscillators) when local accuracy outweighs structure preservation [20].

In practice, semi-implicit updates are often coupled with constraint projection methods such as position-based dynamics (PBD) and extended position-based dynamics (XPBD) to maintain stability under contact bursts [2,24,25].

2.4. Contact, Friction, and Non-Smooth Dynamics

This subsection summarizes the discrete-time treatment of rigid contact and Coulomb friction used in real-time VR simulators. We adopt an impulse and velocity level view, which handles intermittent contact, sticking and sliding, and naturally leads to complementarity or cone projection problems.

Normal contact. Let g_n denote the end-of-step normal gap (non-penetration requires $g_n \geq 0$) and let $\lambda_n \geq 0$ be the normal impulse delivered over the step. Normal contact is enforced by the complementarity condition

$$0 \leq \lambda_n \;\perp\; g_n \geq 0, \tag{7}$$

meaning either a positive gap with zero impulse, or zero gap with a compressive impulse [1,21].

Tangential friction. With coefficient μ, the Coulomb cone is

$$\|\lambda_t\| \leq \mu \lambda_n, \tag{8}$$

with sticking if the post-impact tangential velocity $v_t = 0$ lies strictly inside the cone, and sliding on the boundary with v_t opposite the friction direction (maximum dissipation).

Discrete formulations and solvers. Approximating the circular cone by a friction pyramid yields a linear complementarity problem (LCP); keeping the cone gives a second-order cone program (SOCP). Common real-time solvers include projected Gauss–Seidel and cone projection (PGS/CP), proximal or alternating direction method of multipliers (ADMM) variants, and interior-point methods

Proximity queries. Accurate contact generation requires robust distance queries and feature pairs. For rigid models, Gilbert–Johnson–Keerthi (GJK) algorithm with Expanding Polytope Algorithm (EPA) refinement and bounding volume hierarchies (BVHs) are standard; for deformables, signed distance fields or volumetric proxies are widely used [26–28].

2.5. Perceptual- and System-Aware Adaptivity

This subsection formalizes adaptivity for real-time VR as four ingredients: error-controlled timestepping, contact-aware substepping, gaze-driven physics LOD, and frame-

budget scheduling. The goal is to meet a per-frame budget of B (e.g., $B \approx 11.1$ ms at 90 Hz) without perceptible artifacts [2,18,19,22].

(i) Error-controlled timestepping.

Let the integrator have local order p, and let $\hat{e}_k$ be an estimate of the local truncation error at step k (from an embedded pair or a residual). For a user or device tolerance τ_{phys},

$$\hat{e}_k \le \tau_{\text{phys}} \quad \Rightarrow \quad \Delta t_{k+1} = \Delta t_k \,\text{safety}\left(\frac{\tau_{\text{phys}}}{\hat{e}_k}\right)^{\frac{1}{p+1}}, \tag{9}$$

with safety $\in (0,1)$ to prevent step rejection cascades. This keeps integration error below a perceptual threshold while exploiting slack when motion is slow.

(ii) Contact-aware substepping.

For a candidate contact with normal gap $g_n > 0$ and closing speed $v_n^- \ge 0$, choose the number of substeps m inside the frame step Δt so that predicted closure per substep is a fraction $\eta \in (0,1)$ of the gap:

$$m \;\ge\; \left\lceil \frac{v_n^- \,\Delta t}{\eta\, g_n + \varepsilon} \right\rceil, \qquad \Delta t_{\text{sub}} = \Delta t / m, \tag{10}$$

with small ε to avoid division by zero. This reduces penetrations without globally shrinking Δt during brief contact bursts.

(iii) Gaze-driven physics LOD.

Let $\pi : \mathbb{R}^3 \to \mathbb{R}^2$ be the rendering projection and $J_i = \partial\pi/\partial x$ evaluated at object i. To bound screen-space motion by an eccentricity-dependent tolerance $\tau_{\text{pix}}(\theta_i)$ (from eye-tracking),

$$\|J_i\, \delta x_i\| \;\le\; \tau_{\text{pix}}(\theta_i), \qquad \delta x_i \approx v_i\, \Delta t_i, \tag{11}$$

which yields a per-object update interval

$$\Delta t_i \;\le\; \frac{\tau_{\text{pix}}(\theta_i)}{\|J_i\, v_i\|}. \tag{12}$$

Equivalently, define a perceptual weight $w_i = w(\theta_i)$ (e.g., $w(\theta) = \exp(-\beta\theta)$) and set per-object iteration counts $I_i \propto w_i$ or update rates $r_i = 1/\Delta t_i$ that decrease with eccentricity [19].

(iv) Frame-budget scheduling.

Let $\mathcal{T}$ be tasks in the current frame (constraint batches, cloth solver iterations, background rigid updates, etc.). Each task $t \in \mathcal{T}$ has cost c_t and perceptual utility u_t (higher near fovea, for large screen-space motion, or for audible haptic coupling).

We allocate work by

$$\max_{\{x_t\}} \sum_{t\in\mathcal{T}} u_t\, x_t \quad \text{subject to} \quad \sum_{t\in\mathcal{T}} c_t\, x_t \le B, \;\; x_t \in \mathbb{N}, \tag{13}$$

a knapsack-style problem. A greedy policy using ratio u_t/c_t is effective in real time; engine backends (e.g., DOTS, Chaos, Flex) realize this by prioritizing foreground constraints and deferring low-u_t tasks to worker queues [2].

Clock Alignment

Let $T_{\text{sim}} + T_{\text{ren}} + T_{\text{queue}} \leq B$. We reduce perceived latency by predicting the presented state to the next vsync time t_v:

$$\hat{x}(t_v) \;=\; x_k + v_k\,(t_v - t_k), \tag{14}$$

and by choosing Δt via (9) so that jitter in T_{sim} does not spill over B.

2.6. Learning- and Gradient-Based Simulation

Differentiable physics layers embed simulation inside gradient-based pipelines for system identification, control, and inverse problems; Recent differentiable solvers that explicitly handle contact (and, when available, frictional effects) improve gradient quality for learning-based control and inverse problems [29]. Such advances make physically grounded optimization more practical within real-time VR pipelines.

Moreover, contact is often regularized (compliance/soft constraints) to ensure stable gradients [30,31]. Physics-Informed Neural Networks incorporate PDE residuals into the loss,

$$\mathcal{L}_{\text{PINN}} = \mathcal{L}_{\text{data}} + \lambda\,\|\mathcal{N}(q, v, \dot{v}) - f\|^2, \tag{15}$$

serving as surrogates for expensive substeps (e.g., local solves in reduced FEM) or for environment response fields in digital twins [14,32,33]. These hybrids can reduce substep counts while meeting comfort constraints when validated against baselines.

Deployment in real-time VR must additionally account for tail latency and out-of-distribution failures; practical guardrails and fallback strategies are discussed in Section 6.8.

2.7. Practical Guidance and Trade-Offs

Based on Table 2 and the preceding discussion, several practical guidelines emerge for real-time VR: structure-preserving (symplectic) schemes are generally preferable for long-horizon stability; adaptive substepping near impacts mitigates penetrations without shrinking the global step; per-frame constraint projection helps maintain feasibility under contacts; and perceptual gating reduces workload by lowering update rates for off-gaze or low-saliency bodies [18,19].

Table 2. Integration schemes under real-time VR constraints. The typical use cases reflect common VR workloads, ranging from low-priority background props to articulated characters and constraint-rich soft objects. The local error and energy behavior indicate long-horizon stability (e.g., drift versus bounded oscillations), while per-step cost determines achievable update rates under stereo frame budgets. See Section 2.7 for stability-oriented discussion and practical selection guidelines.

Method	Local Error	Energy Behavior	Per-Step Cost	Typical Use in VR
Explicit Euler	$\mathcal{O}(\Delta t^2)$ (1st order global)	Drift (non-symplectic)	Very low	Background props, cheap particles; avoid long horizons
Symplectic Euler	$\mathcal{O}(\Delta t)$	Bounded drift (symplectic)	Very low	Rigid stacks, ragdolls with constraint projection [2]
Verlet (position)	$\mathcal{O}(\Delta t^3)$ (2nd global)	Good for conservative forces (symplectic)	Low	Cloth/strings, soft constraints with XPBD-style projection [24,25]
RK4	$\mathcal{O}(\Delta t^5)$ (4th global)	Non-symplectic	Medium–high	Small ODEs requiring accuracy (camera oscillators, effects) [20]

To ground the above dual-rate scheduling principles in a concrete VR medical scenario, we summarize a representative deployment in Box 1 (see Appendix A.2 for full details).

Box 1. Illustrative Deployment: Medical VR Surgical Simulator.

Context: A neurosurgery training system requires haptic feedback at 1 kHz while rendering at 90 Hz on Quest 3 (Table A1).
Mathematical solution:

- *Physics* (Section 2): Reduced-order FEM with 50 modal basis functions (Equation (1), $\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C} = \mathbf{f}_{\text{ext}}$). Modal reduction: 10K tetrahedral elements $\rightarrow$ 50 modes, achieving 0.8 ms update at 90 Hz.
- *Haptics* (Section 5.4): Passivity controller (Equations (66) and (67)) with adaptive damping maintains $E[k] \geq 0$ over 5-min procedures, preventing instability under variable rendering latency.
- *Rendering* (Section 4): Per-vertex shader skinning from modal weights; budget allocation: 3 ms physics + 6 ms rendering + 2 ms margin = 11 ms at 90 Hz.

Measured outcomes ($n = 12$ surgeons, 20 procedures):

Metric	Target	Achieved	Ref.
Haptic rate	1 kHz	950–1000 Hz	Section 5.4
Visual rate (P95)	90 Hz	88–90 Hz	Section 6
Force fidelity	<10% error	8.1% RMSE	Equation (68)

Key lesson: Modal reduction (Section 2.6) is essential—full FEM cannot meet dual refresh rates. Decoupled visual/haptic threads (Section 5.4) prevent mutual interference.
See Appendix A.2 for full deployment details including design rationale and measured performance breakdown.

2.8. Summary of Section 2

Table 3 summarizes the core formulations, practical implications, and canonical references introduced in Section 2 under real-time VR constraints.

Table 3. Section 2 at-a-glance: core physics formulations, their practical implications for real-time VR, and representative references. Each row summarizes the mathematical form, typical stability/performance trade-offs, and the system-level consequences that matter at VR update rates (e.g., feasibility under constraints and long-horizon energy behavior). Cross-references point to detailed derivations and implementation notes.

Topic	Core Formulation	Practical Implications	Canonical Refs.
Dynamics & constraints	$M\dot{v} + C = f_{\text{ext}} + J^{\top}\lambda$; algebraic constraints $g(q) = 0$, $g_n(q) \geq 0$; stabilization via Baumgarte or post-step projection.	Fixes state/force notation used throughout. Projection keeps feasibility and reduces drift under real-time steps.	[1,2]
Time integration	Semi-implicit/symplectic, Verlet, RK4 (accuracy vs. structure).	Use symplectic + projection for long-horizon stability; reserve RK4 for small, non-stiff subsystems when accuracy dominates.	[20,24,25]
Contact & friction	Complementarity $0 \leq \lambda_n \perp g_n \geq 0$; Coulomb cone $\|\lambda_t\| \leq \mu\lambda_n$; LCP/SOCP formulations; collision via GJK/EPA, BVHs; deformables via SDF/volumetric proxies.	Choose PGS/CP for speed or ADMM/interior-point for tighter solves on small sets. Robust proximity queries are critical for stable contacts.	[21,26–28]
Perceptual/system adaptivity	Saliency/occlusion/gaze-gated LOD and update-rate scheduling to meet frame budget B (e.g., $B \approx 11.1$ ms @ 90 Hz).	Prioritize foreground constraints; throttle off-gaze/low-utility tasks to cut misses/jitter without visible artifacts.	[2,18,19,22]
Integrator trade-offs	See Table 2.	Compare accuracy, stability, and cost of representative schemes under VR budgets.	—

3. Spatial Representation and Geometric Transformations in VR

Classical spatial modeling—coordinate frame definitions, transform chains, and rotation operators—remains the backbone of VR systems. What has changed since 2020 is how these primitives are organized, differentiated, and learned in modern pipelines. This section reviews the minimal classical tools needed for VR, then focuses on recent trends that make these tools robust and learnable at scale.

3.1. Coordinate Frames and Transform Chains

This subsection recalls the minimal coordinate-frame conventions and transform chains used throughout VR pipelines. We focus on composing and inverting rigid transforms for head, hand, and world frames, and on bookkeeping that avoids drift in long chains.

Scene graphs in VR arrange objects by hierarchical frames; correctness depends on composing translation, rotation, and scale with numerically stable conventions. Practical recipes for transform order, handedness, and matrix conditioning are well-documented in graphics texts and VR geometry primers [5,34]. In practice, systems favor right-handed world frames with explicit documentation of pre-/post-multiplication and column-/row-major storage to avoid ambiguity during engine integration [34].

3.2. Rotations: From Euler Angles to Quaternions and Lie Groups

We review rotation representations with an eye to interactive stability: Euler angles are compact but suffer from singularities, while quaternions and Lie-group updates provide smooth composition and interpolation for head, hand, and camera control. We outline when each is preferable in real-time systems and prepare the notation used later.

Euler angles are compact but suffer from gimbal singularities; quaternions support smooth interpolation (SLERP) and numerically robust composition for head–pose, hand–pose, and camera control [6,7]. Surveys on 3D orientation (attitude) representations compare rotation matrices, quaternion/Euler-parameter forms, and exponential maps and frame rotations as elements of the Lie group $\mathrm{SO}(3)$ with minimal updates in its tangent space [35].

This Lie-group view is advantageous for VR tracking and control because it enforces orthogonality and unit norm by construction, avoids parameter singularities and redundancy, yields consistent Jacobians for EKF/SLAM/ICP, and reduces long–chain drift via manifold retraction [5].

Consequently, modern systems favor quaternion updates or Lie–algebra increments ($R \leftarrow R \exp([\omega]_\times \Delta t)$) within $\mathrm{SO}(3)$ and $\mathrm{SE}(3)$ to maintain stability at interactive rates [5,6].

3.3. Differentiable Transforms and Inverse Graphics

Modern VR stacks expose camera and object transforms to gradient-based optimization. This subsection formalizes differentiable parameterizations for extrinsics/intrinsics and shows how they plug into photometric objectives without breaking geometric constraints.

A major shift is differentiable transforms that permit gradients to flow through rendering and geometry. Early differentiable rendering frameworks established approximate yet useful Jacobians for camera and lighting parameters [13]. Neural implicit scene models (e.g., NeRF and its fast variants) embed pose and calibration variables into optimization, enabling view synthesis and on-the-fly relighting for VR telepresence and scene capture [10,12].

These pipelines rely on stable pose/warp parameterizations (quaternions, axis–angle) so that gradient-based solvers remain well-conditioned during joint optimization of geometry and appearance [10,13].

3.4. Learning on Manifolds and Non-Euclidean Domains

Many geometry spaces in VR are non-Euclidean (poses on SO(3)/SE(3), shapes on low-dimensional manifolds). We summarize practical manifold-aware tools used for tracking, retargeting, and avatar personalization, emphasizing what matters at interactive rates.

Geometric deep learning extends classical signal processing to meshes, graphs, and point clouds—data domains common in VR avatars, environments, and props. Convolutional operators on manifolds support curvature-aware feature extraction for registration and morphable modeling [36]; recent surveys generalize these ideas to groups, graphs, and gauges with principled handling of symmetry and coordinate choices [37]. In human-centric VR, kinematic consistency is enforced by putting learning inside a model-based loop (e.g., SMPL/SMPL-X fitting), improving 3D pose/shape reconstruction from monocular or sparse sensors [38,39].

3.5. Pose Graphs, SLAM, and Online Estimation for VR

We introduce pose-graph optimization for keeping content world-locked: nodes are 6-DoF poses and edges are relative measurements with covariances. We then give the standard SE(3) objective and explain how these estimators stabilize mixed-reality overlays in real time.

Interactive VR scenes benefit from real-time pose-graph optimization and differentiable tracking to keep virtual content world-locked (anchored in a fixed world frame so objects do not move with the user's head), rather than head-locked (attached to the display frame). A pose graph is a sparse graph whose nodes are 6-DoF poses $T \in \mathrm{SE}(3)$ and whose edges encode relative-pose measurements with covariances.

A standard objective uses Lie-group residuals via the logarithm map:

$$\min_{\{T_i \in \mathrm{SE}(3)\}} \sum_{(i,j)\in\mathcal{E}} \rho\left(\left\| \Sigma_{ij}^{-1/2} \log\left(Z_{ij}^{-1} T_i^{-1} T_j \right) \right\|_2^2 \right), \tag{16}$$

Equation (15) minimizes weighted relative-pose residuals over edges $(i,j) \in \mathcal{E}$ by mapping the group discrepancy $Z_{ij}^{-1} T_i^{-1} T_j$ to a 6D tangent-space error via $\log(\cdot)$, and scaling it by the uncertainty $\Sigma_{ij}^{-1/2}$ while robustifying with $\rho(\cdot)$. In VR, this is used to suppress drift and keep anchors world-locked by enforcing global consistency across many local motion constraints (e.g., loop closures and multi-sensor factors) in Section 3.5.

Recent deep SLAM systems couple learned features with geometric bundle adjustment and robust $\mathrm{SE}(3)$ updates, yielding accurate trajectories under fast motion and low texture [40]. These estimators integrate with differentiable rendering and neural scene fields (Sections 3 and 4), enabling joint refinement of poses, intrinsics/extrinsics, scene fields, and rendering parameters at runtime.

3.6. Summary of Section 3

Table 4 summarizes the main spatial representations and differentiable image-formation tools from Section 3, emphasizing their practical impact on VR tracking, calibration, and avatar control.

Table 4. Section 3 at-a-glance: spatial representations, differentiable image formation, and manifold-aware estimation relevant to VR tracking and calibration. The summary emphasizes structure-preserving pose updates and differentiable objectives that enable online refinement under noisy sensing. These elements support world-locked content, stable anchors, and reproducible evaluation in Section 6.

Topic	Core Idea/Formulation	Practical Implication for VR	Canonical Refs
Transforms & frames	On-manifold pose updates ($R \leftarrow R\,\exp([\delta\phi]_\times)$), unit-quaternion orientation with normalization; stable $\mathrm{SE}(3)$ chaining.	Reduces drift and gimbal issues at 90/120 Hz; improves numerical stability in long transform chains and resampling.	[6,34,35]
Differentiable image formation & neural fields	Photometric losses with Jacobians w.r.t. $\mathrm{SE}(3)$ pose/intrinsics; NeRF-style radiance fields for view synthesis and calibration.	Enables gradient-based tracking/relocalization and online calibration; ties rendering directly to estimation quality.	[10,12,13]
Manifold-aware learning (meshes, avatars)	Geometric DL on surfaces/graphs (e.g., Laplace–Beltrami, spectral/mesh operators); differentiable body/face models.	Respects non-Euclidean structure; better generalization and lower retargeting error for hand-object and avatar tasks.	[36–39]
SLAM & pose-graph optimization	On-manifold bundle adjustment and dense–sparse fusion for robust pose graphs under motion.	Tighter online estimates for world-locked content; reduced drift for stable anchoring in interactive scenes.	[40]

4. Real-Time Rendering and Mathematical Models in VR

Rendering is the terminal stage of the VR pipeline that translates a mathematically defined scene into the visual stimuli perceived by the user. Unlike offline graphics, VR imposes strict constraints on latency, frame rate, and stability, since delays or visual artifacts can degrade presence or induce discomfort. This section organizes the mathematics behind real-time rendering and highlights how recent developments extend classical models [4,5].

4.1. Classical Illumination Models and Shading

This subsection summarizes core shading models and shows how the rendering integral is evaluated efficiently on the GPU.

4.1.1. Phong Shading

The Phong model decomposes shading into ambient, diffuse, and specular terms [3]:

$$I \;=\; k_a I_a \;+\; k_d (L \cdot N)\, I_\ell \;+\; k_s (R \cdot V)^n\, I_\ell, \tag{17}$$

where k_a, k_d, k_s are material coefficients, L is the light direction, N the surface normal, R the reflection of L about N, and V the view direction.

4.1.2. Rendering Equation (Physically Based Rendering)

Physically based rendering (PBR) adopts energy-conserving microfacet BRDFs [41]:

$$L_o(p, \omega_o) \;=\; \int_{\Omega(n)} f_r(p, \omega_i, \omega_o)\, L_i(p, \omega_i)\; \max(0,\, n \cdot \omega_i)\, d\omega_i. \tag{18}$$

Here L_o is the outgoing radiance toward ω_o at point p with normal n, L_i is incident radiance from ω_i over the hemisphere $\Omega(n)$, f_r is the BRDF, and $\max(0,\, n \cdot \omega_i)$ is the cosine term.

4.1.3. Monte Carlo Evaluation of the Hemisphere Integral

On the GPU, the integral is approximated by Monte Carlo with importance sampling [42]:

$$\hat{L}_o = \frac{1}{N}\sum_{k=1}^{N} \frac{f_r(p,\omega_k,\omega_o)\, L_i(p,\omega_k)\, \max(0,\, n\cdot\omega_k)}{p(\omega_k)}, \tag{19}$$

where samples $\omega_k \sim p(\omega)$ are drawn from a proposal p matched to the integrand. Practical choices: cosine-weighted sampling for diffuse and sampling the microfacet normal distribution (e.g., GGX (Ground Glass × unknown)/Trowbridge–Reitz) for glossy lobes [43]. For scenes with both emitters and glossy BRDFs, multiple importance sampling blends light- and BRDF-based proposals to reduce variance [42].

4.1.4. Real-Time Approximations in Engines

Typical pipelines combine (i) next-event estimation for direct lighting, (ii) a few BRDF-importance samples for specular response, and (iii) denoising or temporal accumulation for residual noise. Preintegrated approximations (e.g., split-sum environment BRDF via small LUTs for geometry and Fresnel terms) amortize cost to near constant time in PBR engines [4,5].

4.2. Camera Models and Projection Matrices

This subsection formalizes the pinhole and off-axis (asymmetric) stereo cameras used in VR, derives per-eye projection from headset geometry, and gives the math for lens predistortion and timewarp.

4.2.1. Symmetric Perspective

With intrinsics f_x, f_y, and principal point (c_x, c_y),

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \qquad P_{\text{sym}} = \begin{bmatrix} \frac{2n}{w} & 0 & 0 & 0 \\ 0 & \frac{2n}{h} & 0 & 0 \\ 0 & 0 & \frac{f+n}{n-f} & \frac{2fn}{n-f} \\ 0 & 0 & -1 & 0 \end{bmatrix}. \tag{20}$$

where n, f are near/far distances and w, h are the near-plane width/height.

4.2.2. Off-Axis (Asymmetric) per-Eye Frustum

With near-plane bounds l, r, b, t in camera space,

$$P_{\text{off}} = \begin{bmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & \frac{f+n}{n-f} & \frac{2fn}{n-f} \\ 0 & 0 & -1 & 0 \end{bmatrix}. \tag{21}$$

Place the screen/lens image plane at distance d with physical rectangle $[L, R] \times [B, T]$ (meters). Then

$$l = n\,\frac{L}{d}, \quad r = n\,\frac{R}{d}, \quad b = n\,\frac{B}{d}, \quad t = n\,\frac{T}{d}. \tag{22}$$

If the per-eye horizontal optical center shift is s_e (IPD and lens center), with panel half-width $W/2$, then

$$L = -\frac{W}{2} - s_e, \qquad R = \frac{W}{2} - s_e, \qquad s_e = \pm\tfrac{\text{IPD}}{2} - c_x^{(\text{lens})}, \tag{23}$$

where the sign $+$ is for the right eye and $-$ for the left eye.

4.2.3. Lens Distortion (Predistortion for VR)

Let (x, y) be normalized image-plane coordinates and $r^2 = x^2 + y^2$. The Brown–Conrady model writes [44,45]

$$\begin{aligned} x_d &= x\left(1 + k_1 r^2 + k_2 r^4 + k_3 r^6\right) + 2p_1 xy + p_2\left(r^2 + 2x^2\right), \\ y_d &= y\left(1 + k_1 r^2 + k_2 r^4 + k_3 r^6\right) + p_1\left(r^2 + 2y^2\right) + 2p_2 xy. \end{aligned} \tag{24}$$

VR runtimes render to a *predistorted* image using $(x, y) \mapsto (x_d, y_d)$ (or its inverse via a LUT) so that post-lens appearance is undistorted [5].

4.2.4. Timewarp (Orientation-Only and Depth-Aware)

Let K be intrinsics, R_{pred} the head rotation used for rendering, and R_{now} the most recent rotation. Orientation-only timewarp:

$$\tilde{\mathbf{u}} \;\propto\; K\left(R_{\text{now}} R_{\text{pred}}^{-1}\right) K^{-1}\,\mathbf{u}, \tag{25}$$

for homogeneous pixel $\mathbf{u} = (u, v, 1)^\top$; $\tilde{\mathbf{u}}$ samples the rendered color buffer. With per-pixel depth $D(\mathbf{u})$, reconstruct $X = D\,K^{-1}\mathbf{u}$, apply $\Delta T = T_{\text{now}} T_{\text{pred}}^{-1}$, and reproject:

$$\mathbf{u}' \;\propto\; K\left(\Delta T\, X\right). \tag{26}$$

Both warps reduce apparent motion-to-photon latency; use the same per-eye P_{off} in the original render and the warp.

4.3. GPU Pipelines and Shader Mathematics

This subsection isolates shader-side math that most directly affects VR image quality: perspective-correct interpolation, depth mapping/precision, normal-space transforms, and texture filtering with MIP selection.

4.3.1. Perspective-Correct Interpolation

Let a triangle have screen-space barycentrics α, β, γ with $\alpha + \beta + \gamma = 1$ and clip-space w_i at vertices $i \in \{1, 2, 3\}$. For an attribute a,

$$a_{\text{persp}} \;=\; \frac{\alpha\,\dfrac{a_1}{w_1} + \beta\,\dfrac{a_2}{w_2} + \gamma\,\dfrac{a_3}{w_3}}{\alpha\,\dfrac{1}{w_1} + \beta\,\dfrac{1}{w_2} + \gamma\,\dfrac{1}{w_3}}, \tag{27}$$

which avoids the distortion of affine (screen-linear) interpolation [4,46].

4.3.2. Depth Mapping and Precision

With eye-space $z_e > 0$ and near/far n, f, the projection in Section 4.2 gives

$$z_{\text{ndc}} \;=\; \frac{f+n}{n-f} \;+\; \frac{2fn}{(n-f)\,z_e}, \qquad d \;=\; \tfrac{1}{2}\,z_{\text{ndc}} + \tfrac{1}{2}. \tag{28}$$

Since $\partial d / \partial z_e \propto 1/z_e^2$, precision concentrates near n; pushing n outward reduces z-fighting in VR [4].

4.3.3. Normals: Inverse-Transpose and TBN

For linear object transform $M \in \mathbb{R}^{3\times3}$,

$$N' \propto (M^{-1})^{\top} N, \qquad N' \leftarrow \frac{N'}{\|N'\|}. \tag{29}$$

With normal maps, decode tangent-space normal n_t and map via orthonormal basis T, B, N:

$$n_{\text{world}} = [\,T\ B\ N\,]\, n_t. \tag{30}$$

This eliminates shading skew under non-uniform scales and deformations [4].

4.3.4. Texture Filtering and MIP Selection

Given screen-space differentials $\partial u/\partial x$, $\partial u/\partial y$, $\partial v/\partial x$, $\partial v/\partial y$, GPUs choose

$$\lambda = \log_2\!\left(\max\left\{\sqrt{(\tfrac{\partial u}{\partial x})^2 + (\tfrac{\partial v}{\partial x})^2}, \sqrt{(\tfrac{\partial u}{\partial y})^2 + (\tfrac{\partial v}{\partial y})^2}\right\}\right), \tag{31}$$

then apply bilinear/trilinear sampling;anisotropic/Elliptical Weighted Average (EWA) filters approximate the projected footprints ellipse for grazing angles [46,47].

4.3.5. Linear Color and Fast Fresnel

Lighting is computed in linear space; encode to display with the standard RGB (sRGB) transfer function only at output. A common GPU Fresnel approximation is Schlick's

$$F(\cos\theta) = F_0 + (1 - F_0)\,(1 - \cos\theta)^5, \tag{32}$$

which closely tracks conductor/dielectric behavior at a fraction of the cost [48,49].

4.4. Neural and Differentiable Rendering: Models and Math

Neural/differentiable rendering provides learnable scene representations and differentiable image formation suitable for VR.

4.4.1. Neural Radiance and Density Fields

A scene is represented by a network

$$F_\theta(\mathbf{x}, \mathbf{d}) = \big(\sigma(\mathbf{x}),\, c(\mathbf{x}, \mathbf{d})\big), \tag{33}$$

where $\mathbf{x} \in \mathbb{R}^3$ (meters), $\mathbf{d} \in \mathbb{S}^2$ (unit), $\sigma \geq 0$ has units 1/m (attenuation per unit length), and $c \in [0,1]^3$ is RGB; θ are MLP weights [10]. Range conventions: σ typically uses softplus to enforce nonnegativity; c uses sigmoid to stay in gamut.

4.4.2. Frequency Encodings (Positional/SIREN)

To mitigate spectral bias and resolve detail, inputs are lifted by positional encodings

$$\gamma(\mathbf{x}) = \big[\sin(2^0\pi\mathbf{x}), \cos(2^0\pi\mathbf{x}), \ldots, \sin(2^{L-1}\pi\mathbf{x}), \cos(2^{L-1}\pi\mathbf{x})\big], \tag{34}$$

applied per-coordinate; $\gamma(\mathbf{d})$ uses fewer bands due to smoother view dependence [10,50]. An alternative uses periodic activations (SIREN),

$$\mathbf{h}_{\ell+1} = \sin\big(\omega_0 W_\ell \mathbf{h}_\ell + \mathbf{b}_\ell\big), \tag{35}$$

which supplies a Fourier-like basis intrinsically [51]. Bandwidth note:larger L or ω_0 increases representable frequencies but raises sampling demands.

4.4.3. Differentiable Volume Rendering

Along a camera ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$,

$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\,\sigma(\mathbf{r}(t))\,c(\mathbf{r}(t),\mathbf{d})\,dt, \quad T(t) = \exp\Big(-\int_{t_n}^{t}\sigma(\mathbf{r}(s))\,ds\Big). \tag{36}$$

$T(t)$ is transmittance (survival probability to depth t). With stratified samples $\{t_i\}$, $\delta_i = t_{i+1} - t_i$,

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} \underbrace{\exp\Big(-\sum_{j=1}^{i-1}\sigma_j\,\delta_j\Big)}_{T_i}\ \underbrace{\big(1-\exp(-\sigma_i\,\delta_i)\big)}_{\alpha_i}\ c_i, \tag{37}$$

In Equation (36), $T_i = \exp(-\sum_{j<i}\sigma_j\delta_j)$ is transmittance and $\alpha_i = 1 - \exp(-\sigma_i\delta_i)$ is opacity, so $w_i = T_i\alpha_i$ yields front-to-back compositing along the ray [10]. Intuitively, higher density σ_i increases opacity and attenuates contributions from later samples, which is the key mechanism enabling differentiable view synthesis in real-time VR pipelines.

Gradients: $\partial\hat{C}/\partial c_i = w_i$; $\partial\alpha_i/\partial\sigma_i = \delta_i e^{-\sigma_i\delta_i}$; autodiff handles the dependence of later T_k ($k > i$) on earlier σ_i.

4.4.4. Differentiable Rasterization

(i) Forward model.

Vertices $X_w \in \mathbb{R}^3$ map to camera space $X_c = R\,X_w + t$ and to pixels via the pinhole projection

$$u \;=\; f_x\frac{X_c}{Z_c} + c_x, \qquad v \;=\; f_y\frac{Y_c}{Z_c} + c_y. \tag{38}$$

Shading produces color $S(u, v; \theta_{\text{sh}})$ (BRDF/texture/lighting). The per-image loss is

$$\mathcal{L} \;=\; \sum_{\mathbf{p}} \rho\big(C(\mathbf{p}) - C^*(\mathbf{p})\big), \tag{39}$$

with robust penalty $\rho(\cdot)$ [13]. Purpose: Equations (38) and (39) define the forward pass and objective.

(ii) Projection Jacobians (pose). With $X_c = (X_c, Y_c, Z_c)^\top$, the image Jacobian is

$$J_\pi(X_c) \;=\; \begin{bmatrix} \frac{f_x}{Z_c} & 0 & -f_x\frac{X_c}{Z_c^2} \\ 0 & \frac{f_y}{Z_c} & -f_y\frac{Y_c}{Z_c^2} \end{bmatrix}. \tag{40}$$

For an SE(3) twist $\delta\xi = (\delta t, \delta\phi)$,

$$\frac{\partial X_c}{\partial \xi} \;=\; \begin{bmatrix} I & -[X_c]_\times \end{bmatrix} \in \mathbb{R}^{3\times 6}, \qquad \frac{\partial(u,v)}{\partial\xi} \;=\; J_\pi(X_c)\,\frac{\partial X_c}{\partial\xi}. \tag{41}$$

Manifold-consistent updates keep $R \in \mathrm{SO}(3)$:

$$R \;\leftarrow\; R\,\exp([\delta\phi]_\times), \qquad [\delta\phi]_\times = \begin{bmatrix} 0 & -\delta\phi_z & \delta\phi_y \\ \delta\phi_z & 0 & -\delta\phi_x \\ -\delta\phi_y & \delta\phi_x & 0 \end{bmatrix}. \tag{42}$$

Purpose: Equations (40) and (42) give pose sensitivities and a stable pose update [13].

(iii) Shape and intrinsics derivatives. Linear shape bases B_k give

$$X_w(\alpha) \;=\; X_w^{(0)} + \sum_k \alpha_k B_k, \qquad \frac{\partial(u,v)}{\partial \alpha_k} \;=\; J_\pi(RX_w + t)\, R\, B_k. \tag{43}$$

Intrinsics gradients are

$$\frac{\partial u}{\partial f_x} = \frac{X_c}{Z_c}, \quad \frac{\partial v}{\partial f_y} = \frac{Y_c}{Z_c}, \quad \frac{\partial u}{\partial c_x} = 1, \quad \frac{\partial v}{\partial c_y} = 1. \tag{44}$$

Purpose: Equations (43) and (44) provide shape/camera sensitivities.

(iv) Smooth visibility/occlusion. Soft coverage (edge SDF $s_i(\mathbf{p})$) and soft z-buffer give nonzero gradients near silhouettes:

$$w_i(\mathbf{p}) \;=\; \sigma\big(-\kappa\, s_i(\mathbf{p})\big), \qquad \sigma(z) = \frac{1}{1+e^{-z}},\ \kappa > 0, \tag{45}$$

$$\omega_i(\mathbf{p}) \;=\; \frac{\exp(-\tau\, z_i)}{\sum_j \exp(-\tau\, z_j)}, \qquad \tau > 0. \tag{46}$$

Blended color:

$$C(\mathbf{p}) \;=\; \sum_i \big(w_i(\mathbf{p})\, \omega_i(\mathbf{p})\big)\, S_i(\mathbf{p}; \theta_{\text{sh}}). \tag{47}$$

Purpose: Equations (45)–(47) relax discrete visibility for differentiability (cf. OpenDR [13]).

(v) Shading derivatives (chain rule). For texture sampling with UVs (u_t, v_t),

$$\frac{\partial S}{\partial X_c} \;=\; \frac{\partial S}{\partial (u_t, v_t)}\, \frac{\partial (u_t, v_t)}{\partial (u, v)}\, \frac{\partial (u, v)}{\partial X_c}, \tag{48}$$

and analogously for $\xi, \alpha_k, f_x, f_y, c_x, c_y$. Note: Include TBN for normal maps and any material channels in $\partial S / \partial(\cdot)$.

(vii) Loss-gradient assembly. With residual $r(\mathbf{p}) = C(\mathbf{p}) - C^*(\mathbf{p})$ and $\rho'(r)$,

$$\frac{\partial \mathcal{L}}{\partial \theta} \;=\; \sum_{\mathbf{p}} \rho'(r(\mathbf{p})) \left(\sum_i \frac{\partial C}{\partial S_i} \frac{\partial S_i}{\partial \theta} \;+\; \sum_i \frac{\partial C}{\partial \mathbf{u}_i} \frac{\partial \mathbf{u}_i}{\partial \theta} \right), \tag{49}$$

for $\theta \in \{\xi, \alpha_k, f_x, f_y, c_x, c_y, \theta_{\text{sh}}\}$. Note: $\frac{\partial C}{\partial \mathbf{u}_i}$ accumulates coverage and depth terms from Equations (45) and (46).

(viii) Stability practices (concise checklist).

- Coarse→fine schedules: solve pose/intrinsics first, then shape/appearance.
- Clamp softness: choose κ, τ to avoid vanishing/exploding gradients at edges.
- Use robust penalties ρ and multi-view priors for regularization.
- For avatars, pair with learned face/body models for stronger priors [38,39,52].

4.4.5. Training Objective and Regularization

Given calibrated multi-view targets $C^*(\mathbf{r})$,

$$\min_\theta \sum_{\mathbf{r} \in \mathcal{R}} \big\| \hat{C}_\theta(\mathbf{r}) - C^*(\mathbf{r}) \big\|_2^2 \;+\; \lambda\, \mathcal{L}_{\text{reg}}(\theta), \tag{50}$$

where common regularizers include sparsity on σ and distortion/smoothness on accumulated density to reduce floaters [10]. Differentiability: the loss backpropagates through sampling, transmittance, and network layers by the chain rule.

4.4.6. Acceleration for Interactive VR

To approach dual-eye 90/120 Hz, systems (i) factorize fields into many tiny MLPs for parallel evaluation [11], (ii) amortize multi-bounce lighting via neural radiance caching inside real-time path tracing [12], and (iii) reduce memory pressure in large scenes through streamable, memory-efficient radiance field representations that support partitioning and on-demand streaming [53]. Budgeting tip: couple per-eye foveation with coarser sampling/offline caches in the periphery to respect VR frame budgets. In practice, learning-based accelerations should be paired with runtime monitoring and classical fallbacks to remain robust under latency spikes and out-of-distribution content (Section 6.8).

4.5. Hybrid and Perceptual Rendering for VR

VR must sustain high frame rates with limited budgets. Perceptual strategies modulate computation where the human visual system is most sensitive. Foveated rendering concentrates resolution and shading near the gaze point, guided by eye tracking; recent surveys organize algorithms and hardware support [16,17,19]. Temporal reprojection/upsampling and dynamic level-of-detail further trade accuracy for stability.

Hybrid renderers mix rasterization (for primary visibility and nearby geometry) with selective ray/path tracing (for glossy reflections, soft shadows, GI), allocating work to regions that most impact perceived quality. Perceptual error metrics and attention-aware policies help decide when to refine shading or geometry [18,22]. Together, these trends show a shift from purely deterministic shading to hybrid, learned, and perceptually guided rendering, tailored to the stringent constraints of immersive VR.

4.6. Summary of Section 4

Section 4 reviewed how classical shading, camera and projection models, GPU math, and newer neural or differentiable methods are combined into practical VR rendering pipelines. Table 5 summarizes the core ideas and their impact on VR systems.

Table 5. Section 4 at-a-glance: rendering foundations and modern extensions for VR. The summary spans physically based shading, camera and projection models for stereo, GPU-precision considerations, and neural/differentiable methods for view synthesis. Perceptual strategies such as foveation allocate computation where users are most sensitive, supporting real-time performance without sacrificing perceived quality.

Topic (Section)	Core Math/Formulation	VR Impact
Classical shading (Section 4.1)	Physically based microfacet BRDFs with energy-conserving parameters; Schlick-style Fresnel; compute lighting in linear space and apply the sRGB transfer function only at the final output.	Gives physically plausible highlights and stable tone reproduction at real-time rates.
Cameras & projection (Section 4.2)	Per-eye off-axis frusta from IPD and lens geometry using (l, r, b, t); lens pre-distortion; orientation- and depth-aware timewarp/spacewarp that reuse the same per-eye projection as the main render.	Produces correct stereo geometry and reduces apparent latency without resubmitting full frames.
GPU math essentials (Section 4.3)	Perspective-correct interpolation realised by dividing attributes by clip-space w, interpolating, then renormalizing; depth precision concentrated near the near plane; normal transform via the inverse-transpose of the model matrix; MIP level estimated from screen-space derivatives, with anisotropic or EWA filtering when needed.	Avoids affine warping, mitigates z-fighting by pushing the near plane, fixes shading skew, and preserves texture detail at grazing angles.
Neural & differentiable rendering (Section 4.4)	Neural radiance fields for view synthesis with volumetric integration along rays; differentiable rasterization that exposes derivatives with respect to pose and intrinsics, updating rotations on the SO(3) manifold.	Enables gradient-based tracking, relocalization, and online calibration; real-time use becomes feasible via network factorization and radiance caching.
Hybrid & perceptual (Section 4.5)	Eye-tracked foveated rendering; temporal reprojection and upsampling; hybrid pipelines where rasterization handles primary visibility while selected reflections, soft shadows, and global illumination use ray or path tracing.	Concentrates computation where perception is most sensitive and improves quality-per-watt under 90/120 Hz stereo constraints.

Practical Takeaways

- Do all shading computations in linear space, and apply the sRGB transfer function only once at the final color write.
- Use the same per-eye off-axis projection in both the main render and the time-warp/spacewarp paths to avoid extra distortion.
- Push the near plane outward as far as content allows, and transform normals with the inverse-transpose of the model matrix to reduce depth and shading artifacts.
- In NeRF-style or neural field training, make physical units explicit (for example, treat density as "per meter") and monitor the accumulated opacity weights that actually scale image residuals and gradients.
- For 90/120 Hz stereo, allocate resolution and shading budget to the fovea, and rely on cheaper sampling, models, or caches in the periphery.

5. Interaction Modeling and User Interface Dynamics

User interaction in VR is a closed-loop process spanning sensing, state estimation, kinematic/physics modeling, rendering, and human motor response. This section consolidates foundational models for pose and manipulation, then organizes advance during the recent five years (2021–2025) that fuse learning-based perception, differentiable optimization, and perceptual constraints into the interaction loop.

5.1. Foundations: Sensing, Pose, and Kinematic Mapping

This subsection formalizes the sensing→pose pipeline and the kinematic mapping used to drive hands, tools, and avatars in VR under real-time constraints. Contemporary VR interaction follows the classic 3D UI taxonomy (selection, manipulation, travel, system control) [15]. Rigid-body poses evolve on $SE(3)$; orientations are commonly represented by unit quaternions for numerically stable interpolation and composition [7,34,35]. End-effector goals (e.g., a hand or tool tip) are mapped to joint angles via FK/IK with joint limits and comfort costs [8].

Foundational work on modeling and controlling virtual humans provides enduring principles for avatar kinematics, posture control, and animation pipelines, complementing the FK/IK formulations used in VR embodiment [54].

Social VR further couples embodiment with multisensory feedback. For example, even simple collision haptics during human–virtual human interaction can measurably influence presence and user experience, motivating careful feedback design alongside stability constraints [55]. In practice, low-latency controllers rely on Jacobian-based updates.

(i) Problem setup and notation. Given a task-space target $x^\star \in \mathbb{R}^k$ and forward kinematics $f(\theta) \in \mathbb{R}^k$ with joint angles $\theta \in \mathbb{R}^m$, define the task error

$$e \;=\; x^\star - f(\theta). \tag{51}$$

Here $k \in \{3, 6\}$ denotes position/pose, and m is the number of degrees of freedom.

(ii) Geometric Jacobian. With the geometric Jacobian $J = \partial f / \partial \theta \in \mathbb{R}^{k \times m}$, small variations satisfy the linear approximation

$$\delta x \;\approx\; J\,\delta\theta. \tag{52}$$

In real time, J is updated numerically or from analytic expressions.

(iii) DLS (damped least-squares) update. To remain stable near singularities, use the damped update

$$\Delta\theta \;=\; \eta\, J^\top \Big(JJ^\top + \lambda^2 I \Big)^{-1} e, \tag{53}$$

Here the damping term $\lambda^2 I$ regularizes the inverse near singular configurations by keeping $(JJ^\top + \lambda^2 I)$ well-conditioned, so $\Delta\theta$ remains bounded even when J is ill-conditioned. In VR hand/avatar control, this trades accuracy for stability under tight frame budgets, and λ is typically tuned to avoid jitter while maintaining responsive convergence.

(iv) Task weighting and conditioning. If task components have different priorities, introduce $W \succ 0$:

$$\Delta\theta \;=\; \eta\, J^\top \Big(JJ^\top + \lambda^2 W\Big)^{-1} e, \tag{54}$$

which improves conditioning and allows per-axis tolerances.

(v) Secondary objectives via null space. To address posture comfort or joint-limit avoidance without disturbing the primary task,

$$\Delta\theta \;\leftarrow\; \Delta\theta \;+\; \big(I - J^{+}J\big)\,\big(-\nabla_\theta C(\theta)\big), \tag{55}$$

where $C(\theta)$ is a secondary cost and J^+ the pseudoinverse.

(vi) Lightweight alternatives. When matrix solves are too expensive, use the Jacobian-transpose heuristic

$$\Delta\theta \;=\; \eta\, J^\top e \tag{56}$$

or cyclic coordinate descent (CCD). These are approximations but very fast.

(vii) Closing the loop (tracking and full-body drive). Visual–inertial (inside-out) tracking estimates head/hand 6DoF poses; SLAM back-ends stabilize world anchors over long interactions [40]. For detailed hands/faces, learned regressors provide strong priors for retargeting and animation [39]. Empirically, controller input can outperform vision-only hand tracking on some tasks, motivating context-aware switching policies [56].

5.2. Hand, Body, and Object Interaction

Direct manipulation (grasp, pinch, poke) hinges on accurate articulated tracking with temporal smoothing and constraint projection to suppress jitter near contact [57]. Whole-body interaction blends controller priors, optical cues, and IK to maintain reachability and avoid self-collision in tight spaces [58]. Robust contact and proximity tests rely on efficient collision detection, broad-phase culling, and convex distance queries (e.g., GJK), which also inform haptic proxies and constraint solvers [26,28].

A growing trend is controller-less tracking from headset-mounted egocentric cameras, where self-occlusion and motion blur become dominant failure modes. Large-scale egocentric datasets enable such body/hand tracking under realistic occlusions, strengthening avatar embodiment pipelines [59].

5.3. Constraint-Based Manipulation and Differentiable Calibration

This subsection formalizes contact handling for VR manipulation at real-time rates using three complementary views–impulse/velocity, convex QP, and position-level projection–and then shows how differentiable sensing/graphics enable online calibration.

5.3.1. Impulse-/Velocity-Level Contact Resolution

Let q, v be configuration and velocity, $M \succ 0$ the generalized mass, J the contact Jacobian, and f_{ext} external forces. Over a step Δt,

$$M\,(v_{t+1} - v_t) \;=\; \Delta t\, f_{\text{ext}} \;+\; J^\top \lambda, \tag{57}$$

where λ are contact impulses. With normal gap $g_n(q) \geq 0$ (signed distance), nonpenetration is enforced by complementarity

$$g_n(q_{t+1}) \;\geq\; 0, \quad \lambda_n \;\geq\; 0, \quad \lambda_n^\top g_n(q_{t+1}) \;=\; 0, \tag{58}$$

and tangential (Coulomb) friction by a cone $\|\lambda_t\| \leq \mu\, \lambda_n$ or a friction pyramid. A velocity-level variant uses

$$J_n v_{t+1} + b_n \;\geq\; 0, \quad \lambda_n \geq 0, \quad \lambda_n^\top (J_n v_{t+1} + b_n) = 0, \tag{59}$$

with b_n collecting Baumgarte/restition terms. These yield a mixed complementarity problem (MCP) in λ (or in (v_{t+1}, λ)) [2,21].

5.3.2. Convex QP Time Stepping

A common real-time scheme solves a convex projection in velocity space:

$$\min_{\Delta v,\, \lambda} \; \frac{1}{2} \|\Delta v - \Delta v_{\text{free}}\|_M^2 \quad \text{subject to} \quad J_n\, \Delta v + b_n \;\geq\; 0, \;\; \text{friction constraints}, \tag{60}$$

where $\Delta v_{\text{free}} = M^{-1} \Delta t\, f_{\text{ext}}$. Stationarity gives the KKT condition $M\, \Delta v - J^\top \lambda = 0$; together with the inequalities and the complementarity

$$\lambda_n^\top (J_n \Delta v + b_n) = 0, \tag{61}$$

(60) is equivalent to an LCP when a friction pyramid is used, or to a SOCP with a circular cone $\|\lambda_t\| \leq \mu \lambda_n$. This framing unifies projected Gauss–Seidel/cone-projection and interior-point solvers under one objective [2].

5.3.3. Geometric (Position-Level) Projection

Position-level methods correct constraint violations directly:

$$\min_{\Delta q} \; \frac{1}{2} \|\Delta q\|^2 \quad \text{subject to} \quad C(q + \Delta q) = 0, \tag{62}$$

where $C(q) = 0$ encodes joints/contacts. Linearizing,

$$\Delta q \;\approx\; -\, J_C^\top (J_C J_C^\top)^{-1}\, C(q), \tag{63}$$

with $J_C = \partial C / \partial q$. In practice, constraints are processed iteratively (Gauss–Seidel) within the frame budget; small compliance or damping in C regularizes contact jitter [2].

5.3.4. Differentiable Calibration and Identification

Online calibration estimates unknown frames/offsets (e.g., controller/tool pose offsets or contact frames) by minimizing visual reprojection or photometric residuals through differentiable graphics/kinematics. With calibrated intrinsics K, 3D features X_i, image points u_i, and pose $T \in \mathrm{SE}(3)$,

$$\min_{T,\, \phi} \; \sum_i \left\| \; \pi\big(K\, T(\phi)\, X_i\big) \;-\; u_i \; \right\|_2^2 \;+\; \lambda\, \mathcal{L}_{\text{photo}}, \tag{64}$$

where ϕ parameterizes the pose increment and π is projection. Updates are applied on the manifold,

$$T \;\leftarrow\; T\, \mathrm{Exp}(\hat{\phi}), \tag{65}$$

which preserves group structure and improves stability [13]. For contact-rich skills, differentiable physics adds residuals on contact consistency (e.g., nonpenetration, frictional stick/slide), enabling recovery of latent states/forces from video or tactile surrogates and improving data efficiency in policy learning [30].

5.4. Haptics and Force-Feedback for Action–Perception Coupling

Haptic channels tighten the loop by reflecting contact and stiffness cues. Lightweight devices and exoskeletons typically use PID/impedance targets with passivity-minded gains to remain stable under network and rendering latency; classic PHANToM-style designs and control envelopes are well documented [60,61]. A canonical proxy-based approach is the constraint-based god-object formulation, which enforces non-penetration while maintaining stable force feedback [62]. In soft-tissue or tool-mediated tasks, reduced-order FEM or proxy models support high-rate (kHz) force updates while the visual path runs at display refresh, sustaining stability and realism [63].

Discrete-Time

Passivity (PO/PC) Let $f_h[k], v_h[k]$ be device-port force/velocity at sample k and T_s the servo period. A passivity observer tracks energy

$$E[k{+}1] \;=\; E[k] \;+\; T_s\, f_h[k]^\top v_h[k] \;-\; T_s\, f_v[k]^\top v_h[k], \quad E[0] \geq 0, \tag{66}$$

where f_v is the virtual environment force. If $E[k{+}1] < 0$ (incipient non-passivity), a passivity controller injects adaptive damping

$$f_v[k] \;\leftarrow\; f_v[k] \;+\; \alpha[k]\, v_h[k], \qquad \alpha[k] \;=\; \frac{-E[k{+}1]}{T_s\, \|v_h[k]\|^2 + \varepsilon}, \tag{67}$$

Equation (65) monitors the net energy exchanged at the device port; if $E[k{+}1] < 0$, the coupled loop is injecting energy (non-passive), which can trigger unstable oscillations under delay or stiff virtual contacts. Equation (66) restores passivity by adding the minimum adaptive damping $\alpha[k]\, v_h[k]$ needed to dissipate the excess energy, making kHz haptic updates stable while visuals run at lower refresh rates.

Which guarantees $E[k{+}2] \geq 0$ and restores passivity while minimally perturbing the nominal interaction. Energy-tank variants enforce $E[k] \in [0, E_{\max}]$ to budget aggressive transients (tool impacts) before throttling feedback. These controllers are compatible with the proxy/FEM pipelines above and preserve stability across rendering-latency fluctuations [60,61,63]. Comparative studies further show that visuo-haptic versus visual-only feedback can systematically change presence, underscoring that modeling priorities depend on the dominant feedback channel [64].

5.5. Perception-Aware Interaction: Foveation, Workload, and Cybersickness

Perceptual limits bound what must be simulated for believable action. Eye-tracked foveation concentrates shading and sampling near the gaze point, freeing budget for physics/IK during action onsets; algorithmic trade-offs and system design are summarized in recent surveys [16,17,19]. Perceptual LOD throttles distant/occluded dynamics without degrading agency. For comfort, ML-based cybersickness predictors leverage demographics and behavioral/physiological signals; Personalized cybersickness prediction models can improve forecasting accuracy at the individual level, enabling comfort-aware adaptation policies in real time [65]. These models can drive online mitigation (e.g., locomotion gains, vignette strength) during interaction [66–68].

5.6. Learning-Augmented Interfaces

Interfaces increasingly fuse geometric cores with learned scene/function representations. NeRF-style encodings assist occlusion-aware queries and grasp target prediction; factorized fields accelerate spatial lookups for interaction [10,11]. Graph/mesh operators enable spatial reasoning over layouts (affordances, collision margins) in shared spaces [36,37]. End-to-end policies co-train perception and control through differentiable kinematics/rendering losses for low-drift hand–object tasks [39]. Personalized, animatable avatars reduce retargeting error and improve ownership in multi-user scenes [31,52].

Emerging LLM-driven agents in social VR suggest a path toward more natural dialogue and interactive behavior; however, they introduce new constraints on latency, safety, and evaluation [69]. These trends motivate future benchmarking that jointly measures interaction quality and real-time system performance.

Despite their promise, learning-augmented interfaces should be paired with deployment guardrails (latency margins, validation, fallbacks) to remain robust in diverse real-time VR settings (Section 6.8).

5.7. Robustness, Latency, and System Practices

Interaction quality depends on stable 6DoF state, contact resolution, and motion-to-photon latency. Practical engines prioritize foreground constraints, defer background dynamics, and align sensing/physics/render clocks to minimize jitter [15]. Physics choices (semi-implicit/symplectic steps for energy behavior, adaptive Δt for contact bursts) directly influence grasp stability and selection accuracy [2].

5.8. Summary of Section 5

Section 5 collected the main ingredients for physically grounded, yet comfortable VR interaction: manifold-aware pose updates, IK for arm and hand control, contact and constraint handling, online calibration, and perceptual allocation of computation and haptics. The main design rules are:

(i) Maintain poses on $\mathrm{SE}(3)$ with quaternion/Lie updates; fuse visual–inertial sensing to stabilize world-locked anchors.
(ii) Map end-effector goals with DLS/transpose/CCD under joint limits; use null-space terms to encourage comfort and good posture.
(iii) Resolve contacts via impulse/QP or position-level projection, iterating within the frame budget and regularizing to suppress jitter.
(iv) Calibrate online through differentiable kinematics and rendering with manifold-consistent pose updates.
(v) Allocate work perceptually (eye-tracked foveation, perceptual LOD); keep haptics passive under latency using passivity observer/controller (PO/PC) style damping.

As a compact reference for reporting and reproducible evaluation, Table 6 summarizes the interaction metrics used throughout Section 5.

Table 6. Interaction metrics glossary for Section 5. The listed metrics quantify manipulation accuracy, contact feasibility, calibration quality, and responsiveness in ways that are directly measurable in VR experiments. Units and measurement contexts are included to support reproducible reporting, and the associated formulas provide precise definitions used later in benchmarks and case studies (Section 6).

Metric	Symbol	What It Measures	Notes/Units
EE position RMSE	Equation (68)	Average Euclidean error of the end-effector position relative to a reference trajectory.	Meters, in the world frame.
EE orientation error	Equation (69)	Angular misalignment between predicted and reference orientations (geodesic angle on SO(3)).	Degrees or radians.
Contact infeasibility	Equation (70)	Largest remaining penetration depth over all active contacts.	Meters; should be close to zero.
Complementarity residual	Equation (71)	Violation of the normal-contact complementarity $0 \le \lambda_n \perp g_n \ge 0$.	Impulse·meter; zero means perfectly feasible contacts.
Reprojection RMSE	Equation (72)	Pixel-wise error when reprojecting tracked visual features into the image.	Pixels in the image plane.
Calibration drift	Equation (73)	Accumulated change of the estimated pose on SE(3) over time.	Geodesic distance, e.g., $\|\log(\Delta T)\|$.
Control latency	Equation (74)	Delay from sensing to the corresponding actuation reaching the user.	Milliseconds.
Slip rate	Equation (75)	How often a grasp unintentionally fails during interaction.	Fraction in $[0,1]$ (events per grasp).
Passivity margin	Equations (76) and (77)	Energy balance in the haptic loop (how far the system is from violating passivity).	Nonnegative is desired; larger margins are safer.

Reporting Metrics

Notation

$\hat{\mathbf{x}}_i$ (prediction), $\mathbf{x}_i^*$ (reference); $\angle(R) = \arccos((\operatorname{tr} R - 1)/2)$; $g_+(q) = \max(g(q), 0)$; $\odot$ denotes Hadamard product; $\pi(\cdot)$ is the projection with intrinsics K; $\|\Delta T\| = \|\log(\Delta T)\|_2$ is the geodesic norm on SE(3).

$$\textbf{EE position RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \left\|\hat{\mathbf{x}}_i - \mathbf{x}_i^*\right\|_2^2} \tag{68}$$

$$\textbf{EE orientation error} = \angle\left(\hat{R}_i R_i^{*\top}\right) \tag{69}$$

$$\textbf{Contact infeasibility} = \left\|g_+(q)\right\|_\infty \tag{70}$$

$$\textbf{Complementarity residual} = \left\|\lambda_n \odot g_n(q)\right\|_\infty \tag{71}$$

$$\textbf{Reprojection RMSE} = \sqrt{\frac{1}{|\mathcal{P}|}\sum_{p\in\mathcal{P}} \left\| \mathbf{u}_p - \pi\left(K\,\hat{T}_p\,\mathbf{X}_p\right)\right\|_2^2} \tag{72}$$

$$\textbf{Calibration drift} = \|\Delta T\| = \left\|\log\left(\hat{T}_t^{-1} T_t\right)\right\|_2 \tag{73}$$

$$\textbf{Control latency} = t_{\text{pose out}} - t_{\text{sensor in}} \tag{74}$$

$$\textbf{Slip rate} = \frac{\#\{\text{unintended releases}\}}{\#\{\text{grasps}\}} \tag{75}$$

$$\begin{aligned} E[k{+}1] &= E[k] + T_s\, f_h[k]^\top v_h[k] - T_s\, f_v[k]^\top v_h[k], \\ \textbf{Passivity margin} &= \min_{0\le j\le k} E[j] \quad (\ge 0 \text{ desired}). \end{aligned} \tag{76}$$

$$\text{Enforce passivity by:} \quad f_v[k] \leftarrow f_v[k] + \alpha[k]\, v_h[k], \quad \alpha[k] = \frac{-\min(0, E[k{+}1])}{T_s\, \|v_h[k]\|^2 + \varepsilon}. \tag{77}$$

6. Future Direction of Mathematical Modeling in VR

This section charts near-term (1–2 year) opportunities for VR that are (i) geometry-aware–numerics and estimation on appropriate manifolds with constraint/contact handling [2,5,21], (ii) perception-aware–budgets and error bounded by human sensitivity via foveation and comfort models [16,17,19,22], and (iii) learning-augmented–neural modules used where they offer the largest cost–benefit while preserving structure [10–14,40]. Subsections detail: physics and time stepping (Section 6.1), spatial estimation/SLAM (Section 6.2), neural/differentiable rendering (Section 6.3), avatars and IK (Section 6.4), haptics and collision (Section 6.5), and cybersickness-aware scheduling (Section 6.6). Common metrics/targets and a concise reporting template appear in Section 6.7.

6.1. Learning-Augmented Physics and Stable Time Stepping

6.1.1. Structure-Preserving Time Stepping

Let (q, p) be generalized coordinates and momenta with Hamiltonian $H(q, p) = T(p) + V(q)$ and holonomic constraints $c(q) = 0$ with Jacobian $G = \partial c/\partial q$. A symplectic Euler step with multipliers λ is

$$\begin{aligned} p_{k+1} &= p_k - \Delta t\, \nabla V(q_k) - G(q_k)^\top \lambda_k, \\ q_{k+1} &= q_k + \Delta t\, M^{-1} p_{k+1}, \qquad c(q_{k+1}) = 0, \end{aligned} \tag{78}$$

which preserves a discrete symplectic form and yields bounded long-horizon energy drift when constraint residuals are controlled [2,20]. Here $M \succ 0$ is the mass matrix and λ_k enforces $c(q_{k+1})$=0.

6.1.2. Learning Insertion Without Breaking Structure

A learned module predicts parameters ϕ_θ and a conservative correction $U_\theta(q)$ so that $F_\theta(q) = -\nabla U_\theta(q)$, replacing $V \leftarrow V + U_\theta$ while keeping the step symplectic. If only a black-box force $\tilde{F}_\theta$ is available, project it to a conservative field via a Poisson solve

$$\psi_\theta \;=\; \arg\min_{\psi} \int_\Omega \|\nabla\psi(q) - \tilde{F}_\theta(q)\|^2\, dq, \quad F_\theta \leftarrow \nabla\psi_\theta, \tag{79}$$

so the update still derives from a potential [14,32,33].

6.1.3. Nonsmooth Contact and Friction

Normal complementarity and frictional cones (or pyramids) act on impulses λ:

$$g_n(q_{k+1}) \ge 0, \quad \lambda_n \ge 0, \quad \lambda_n^\top g_n(q_{k+1}) = 0, \qquad \|\lambda_t\| \le \mu\, \lambda_n, \tag{80}$$

yielding an MCP/LCP/SOCP depending on the friction model [2,21]. Here g_n is signed gap, μ the friction coefficient, and (λ_n, λ_t) the normal/tangential impulses.

6.1.4. Stability and What to Report

If $\nabla^2(V+U_\theta) \preceq LI$ and constraints are projected each step, discrete Grönwall bounds give

$$|H(q_k,p_k) - H(q_0,p_0)| \;\le\; C\,\Delta t \sum_{j<k} \big(\|c(q_j)\| + \text{friction viol.}\big), \tag{81}$$

so energy drift is controlled by residuals [2,20]. Report energy drift (%), max nonpenetration/equality violation, and fallback/substep rate as in Section 6.7 [18,22].

6.2. Spatial Estimation on Manifolds and Differentiable SLAM

6.2.1. Factor-Graph Objective on SE(3)

Given relative pose measurements Z_{ij} and per-pixel matches (u_{ip}, X_p), estimate poses $T_i \in \mathrm{SE}(3)$, depths/structure D by

$$\min_{\{T_i\},D} \sum_{(i,j)} \big\|\Sigma_{ij}^{-\frac{1}{2}} \log\!\big(Z_{ij}^{-1} T_i^{-1} T_j\big)\big\|_2^2 \;+\; \sum_{(i,p)} \rho\Big(\big\|u_{ip} - \pi(K\,T_i\,X_p(D))\big\|_2^2\Big), \tag{82}$$

where $\log(\cdot)$ is the Lie log on SE(3) and ρ a robustifier. This unifies geometric (between-frames) and photometric (reprojection) terms [40].

6.2.2. On-Manifold Updates and Gauge

Optimize with $T_i \leftarrow T_i \exp(\hat{\xi}_i)$ to remain on SE(3). Fix one pose (and global scale for monocular) to remove gauge freedoms; otherwise the normal equations are rank-deficient [5,40].

6.2.3. Metrics and Alignment Choices

Use ATE/RPE and per-frame reprojection RMSE with alignment stated explicitly (Sim(3) for monocular; SE(3) for stereo/RGB-D), following Section 6.7 [5,40].

6.3. Neural and Differentiable Rendering for Real-Time XR

6.3.1. Cached Estimators and Perceptual Budgeting

For per-pixel radiance, combine a fresh unbiased estimate $\tilde{L}^{(t)}$ and a cached running estimate $\hat{L}^{(t-1)}$ by

$$\hat{L}^{(t)} \;=\; \alpha\,\tilde{L}^{(t)} + (1-\alpha)\,\hat{L}^{(t-1)}, \tag{83}$$

whose MSE decomposes into bias $\propto (1-\alpha)^2$ and variance $\propto \alpha^2/N$. Choose (α, N) by minimizing samples under a CSF-derived weight $w(\theta, f)$ such that $\mathrm{MSE}\cdot w \le \varepsilon_{\mathrm{JND}}$ [16,19]. Neural radiance caching and tiny-MLP factorization supply $\tilde{L}^{(t)}$ at VR rates [10–12].

6.3.2. Foveated Sampling as a Constrained Program

Let $\Lambda(\mathbf{x})$ be spp and $w(\mathbf{x})$ the gaze-weight. Solve

$$\min_{\Lambda \in \mathbb{N}} \sum_{\mathbf{x}} \Lambda(\mathbf{x}) \quad \text{subject to} \quad \mathrm{MSE}(\Lambda(\mathbf{x}))\, w(\mathbf{x}) \le \varepsilon, \tag{84}$$

to allocate samples where the HVS is most sensitive [16,19].

6.4. Avatars, IK, and Differentiable Bodies

6.4.1. Shape–Pose Estimation with Physics Priors

Estimate shape β, pose θ, and global T by

$$\min_{\beta,\theta,T} \sum_i \big\|u_i - \pi\big(K\,T\,X_i(\beta,\theta)\big)\big\|_2^2 + \lambda_{\mathrm{contact}}\|g(q(\theta))_+\|_2^2 + \lambda_{\mathrm{prior}} R(\beta,\theta), \tag{85}$$

where $g(\cdot)$ penalizes interpenetration and R encodes learned priors. This couples differentiable IK/rendering for stable personalization [38,39,52,70]. Here X_i is the skinned vertex, $q(\theta)$ joint angles, and u_i image points.

6.4.2. Identifiability and Sensing

Local identifiability requires the stacked Jacobian w.r.t. (β, θ, T) to have full column rank after gauge fixing; with HMD+controllers, regularization via R and contact terms reduces ambiguity [39,52].

6.5. Haptics, Collision, and Contact Modeling

6.5.1. Discrete Passivity with PO/PC

With sample period T_s and port variables $(f_h[k], v_h[k])$, passivity requires

$$\sum_{k=0}^{N-1} T_s \, f_h[k]^\top v_h[k] \;\geq\; -E_0. \tag{86}$$

A passivity controller injects adaptive damping

$$f_v[k] \leftarrow f_v[k] + \alpha[k]\, v_h[k], \qquad \alpha[k] = \frac{\max(0, -E[k+1])}{T_s \|v_h[k]\|^2 + \varepsilon}, \tag{87}$$

to restore $E[k] \geq 0$ while minimally altering the virtual interaction [60,61,63]. Here f_v is the virtual environment force and $E[k]$ the observed stored energy.

6.5.2. Deterministic Collision Bounds

For convex A, B, GJK/EPA yields $d(A, B) = \min_{a \in A, b \in B} \|a - b\|_2$ with BVH LOD. Guarantee $|d - \tilde{d}| \leq \delta$ by bounding node radii, giving predictable haptic-step timing and stable force updates [26,28].

6.6. Cybersickness-Aware Scheduling and Perceptual Control

Predict-and-Allocate (MPC View)

Let a learned proxy $\hat{s}_t = h_\varphi(x_t)$ predict instantaneous sickness from signals x_t (gaze, motion, frame-time, head kinematics). Allocate budgets $b_t = (b_t^{\text{render}}, b_t^{\text{phys}}, b_t^{\text{anim}})$ by

$$\min_{b_t \geq 0} \; J_t \;=\; \alpha\, \hat{s}_t + \beta\, \text{MTP}_t^2 + \gamma\, \text{miss}_t, \quad \text{subject to} \quad b_t^\top \mathbf{1} \leq B. \tag{88}$$

with horizon-H MPC and device budget B. Train h_φ on SSQ proxies and behavioral/physiological features [66–68] and gate foveation/LOD as in [16,17]. Here MTP_t is motion-to-photon, and miss_t the missed-frame ratio.

6.7. Summary of Section 6

6.7.1. Guiding Principles

(i) Preserve geometric structure in integration and estimation (symplectic/semi-implicit updates; on-manifold pose updates).
(ii) Use learned priors/modules to cut cost, while keeping constraints/guarantees in the outer loop.
(iii) Balance quality and performance with perceptual limits (eye-tracked/foveated budgets; just-noticeable thresholds).

To facilitate reproducible evaluation and cross-paper comparison, we summarize a concise set of common VR metrics in Table 7.

Table 7. Common metrics across physics, estimation, rendering/systems, and comfort for VR evaluation. Physics-oriented measures track stability and constraint satisfaction; estimation measures capture tracking drift and consistency; system measures capture missed-frame behavior and latency; and comfort measures summarize user impact. Together, they provide a compact, reproducible checklist for reporting and cross-paper comparison.

Domain	Metric (Symbol)	What It Measures	Units/Notes
Physics	Energy drift (Equation (89))	Change of total energy over a window	% (lower is better)
	Constraint infeasibility (Equation (90))	Residuals for non-penetration/equality/velocity-level constraints	m or unitless
Estimation	ATE (Equation (91))	Absolute trajectory error after alignment	m (RMSE)
	RPE (Equation (92))	Relative pose error over gap Δ	m/deg (RMSE)
	Reprojection RMSE (Equation (93))	Pixel-space residual per frame	pixels
Rendering/Systems	Missed-frame ratio (Equation (94))	Budget violations per frame	fraction (0–1)
	Visual rate (P95)	95th-percentile achieved display rate (or equivalently, 95th-percentile frame time)	Hz (higher is better); report alongside missed-frame ratio
	Motion-to-photon (MTP)	Sensor-to-photon latency (median/p95)	ms
	Per-eye throughput	Sustained res. and refresh	e.g., $2 \times (2160 \times 2160)$ @ 90/120 Hz
Comfort	SSQ (Total/subscales)	Simulator Sickness Questionnaire	scalar scores
	Time-to-discomfort (Equation (95))	Exposure time until threshold	s or min
	Dropout rate (Equation (96))	Participants unable to complete exposure	fraction (0–1)

6.7.2. Definitions and Formulas

Notation

$\angle(R) = \arccos((\operatorname{tr} R - 1)/2)$; $g_+(q) = \max(g(q), 0)$; $\pi(\cdot)$ is the projection with intrinsics K; $\mathrm{RMSE}(\cdot)$ denotes root-mean-square error; $S \in \mathrm{SE}(3)$ or $\mathrm{Sim}(3)$ aligns estimates to ground truth.

Physics

$$\mathrm{Drift}(\%) \;=\; 100\,\frac{|E(T) - E(0)|}{\max\big(E(0), \varepsilon\big)}, \quad E(t) = T(t) + V(t). \tag{89}$$

$$\big\|g_+(q)\big\|_\infty \;\text{(non-penetration)}, \qquad \|g(q)\|_\infty \;\text{(equality)}, \qquad \|J(q)\,v\|_\infty \;\text{(velocity-level)}. \tag{90}$$

Spatial Estimation

$$\mathrm{ATE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \big\|\mathbf{x}_i - \big(S\,\hat{\mathbf{x}}_i\big)\big\|_2^2}. \tag{91}$$

$$R_i = \big(T_i^{-1} T_{i+\Delta}\big)^{-1}\big(\hat{T}_i^{-1}\hat{T}_{i+\Delta}\big), \quad \mathrm{RPE}_{\mathrm{trans}} = \mathrm{RMSE}\big(\mathrm{trans}(R_i)\big), \;\; \mathrm{RPE}_{\mathrm{rot}} = \mathrm{RMSE}\big(\angle(R_i)\big). \tag{92}$$

$$\mathrm{RMSE}_{\mathrm{repr}} = \sqrt{\frac{1}{|\mathcal{P}|}\sum_{p\in\mathcal{P}} \big\|\,\mathbf{u}_p - \pi\big(K\,\hat{T}_p\,\mathbf{X}_p\big)\big\|_2^2}. \tag{93}$$

Rendering and Systems

$$\text{Missed-frame ratio} \;=\; \frac{\#\{\,t_{\mathrm{frame}} > \mathrm{budget}\,\}}{\#\{\mathrm{frames}\}}. \tag{94}$$

Report MTP as median and 95th percentile (ms) from sensor timestamp to photons-on-panel.

Comfort

$$\text{Time-to-discomfort} \;=\; \min\{\, t \mid \text{SSQ}_{\text{Total}}(t) \geq \tau \,\}. \tag{95}$$

$$\text{Dropout rate} \;=\; \frac{\#\{\text{participants who stop early}\}}{\#\{\text{participants}\}}. \tag{96}$$

6.7.3. Reporting Template (Concise)

(i) Task/dataset and total budget (per-eye resolution/refresh, frame budget).
(ii) Metrics from Table 7, with window sizes and alignment choices.
(iii) Foveation policy and salient system settings (e.g., reprojection, timewarp).
(iv) Statistical summaries (median/p90/p99 with n and confidence intervals).
(v) Ablations isolating learned vs. structure-preserving components.

6.8. Learning-Based Components: Practical Limitations Under Real-Time VR Constraints

6.8.1. Computational Overhead and Latency Variability

Learning-based components can reduce modeling burden or enable new capabilities, but they also introduce inference overhead and latency variability that are difficult to reconcile with strict VR frame budgets. Neural rendering and learned surrogates typically add extra stages (network evaluation, feature fetching, cache management) whose costs are not always amortized across frames, especially when scene content, view-dependent effects, or contact states change rapidly. Even when the median cost appears acceptable, occasional spikes caused by content-dependent workloads, thermal throttling, driver/runtime effects, or concurrent processes can trigger missed deadlines and visible judder. This is particularly problematic in stereo rendering, where small scheduling slips can accumulate into motion-to-photon latency increases and discomfort. Consequently, evaluation should emphasize tail behavior (e.g., percentile frame times) and include explicit fallback policies that preserve interaction continuity when learned components exceed budget.

6.8.2. Generalization and Out-of-Distribution Failures

A second limitation is robustness under distribution shift. Models trained on curated datasets can fail under novel lighting, motion patterns, occlusions, sensor noise, or uncommon user morphologies. In tracking and world-locking pipelines, such failures manifest as drift, sudden pose jumps, or loss of tracking, which directly undermines spatial stability and can produce strong discomfort. Similarly, learned avatar or body/hand models may degrade for users whose proportions or motion styles are underrepresented in training data, increasing retargeting error and reducing embodiment. Learned physics surrogates face comparable risks: contact-rich edge cases (multi-contact, high-velocity impacts, or frictional stick–slip transitions) can violate learned assumptions and yield non-physical behavior. For real-time VR deployment, these failure modes argue for hybrid designs in which learned outputs are continuously validated by geometric/physical consistency checks and are automatically reverted to classical estimators/solvers when confidence drops.

6.8.3. Deployment Guardrails and Recommended Practice

From a deployment perspective, the most reliable pattern is to treat learning as an augmentation layer rather than a single point of failure. Classical pipelines remain attractive because they provide predictable runtime, clearer debugging signals, and graceful degra-

dation modes (reduced quality rather than catastrophic loss). Learning is most appropriate when its scope is well-bounded (e.g., accelerating a specific query, providing priors for optimization, or improving personalization) and when runtime monitors can detect anomalies early. Practically, systems should (i) profile performance under worst-case content and device states, (ii) reserve explicit margins for variability, (iii) implement fallback paths that maintain tracking stability and interaction safety, and (iv) report these guardrails as part of reproducibility. These considerations complement the broader limitations discussed later (Section 7.4) and motivate hybrid architectures that combine learned priors with geometric and physical constraints.

7. Conclusions

7.1. Synthesis: Toward Geometry-Aware, Perception-Aware, and Learning-Augmented VR

This survey has argued that the clearest path for real-time VR is a synthesis of three complementary lenses: geometry-aware models that preserve structure in dynamics and pose; perception-aware scheduling that allocates computation where it matters for users; and learning–augmented components that accelerate or calibrate core loops without breaking invariants. In practice, constraint-consistent updates and semi-implicit/symplectic stepping support physically plausible motion [2], perceptual tolerances gate adaptive effort across physics and rendering [16,17,22], and differentiable layers couple estimation and inverse problems to image formation and scene priors [10,13].

7.2. Domain-Specific Advances and Challenges

Physics. Progress stems from stable time stepping. Data-driven parameter identification and projection-based constraints further improve robustness. Nonsmooth frictional contact remains the main obstacle for differentiable pipelines and learned controllers [21]. Estimation. On-manifold optimization fused with differentiable rendering unifies tracking and calibration. This approach reduces drift and improves relocalization under fast motion [13,40]. Rendering. Hybrid rasterization–neural methods are pushing dual-eye 90/120 Hz. Representative techniques include radiance caching and factorized tiny networks. These maintain photometric fidelity [10–12]. Embodiment and Haptics. Implicit neural avatars with differentiable IK reduce retargeting error from sparse sensors [39,52,70]. Passivity-aware impedance control and reduced-order models sustain kHz haptics alongside visual refresh. Device/scene co-design and stability margins under latency remain open challenges [60,61,63]. Comfort-aware controllers that close the loop with cybersickness predictors are promising, but they need broader validation [66–68,71].

7.3. Methodological Recommendations

To make advances comparable and cumulative, we recommend a common reporting battery. For physics, include long-horizon energy drift, momentum/constraint violations, and contact stability traces [2]. For estimation, report ATE/RPE and per-frame reprojection RMSE with defined train/test splits and motion profiles [13,40]. For rendering/systems, provide frame-time percentiles (P50/P90/P99), motion-to-photon latency, and binocular synchronization, alongside image metrics; for interaction/comfort, include standardized cybersickness outcomes with pre-registered thresholds and time-to-discomfort analyses, plus ablations of perception-informed policies [16,17,66,67]. Where learning is used, ablate the surrogate's role, characterize failure modes, and disclose memory/bandwidth profiles relevant to headsets.

7.4. Limitations and Future Outlook

Three cross-cutting limitations persist. First, memory/bandwidth ceilings constrain neural scene/shape representations in large or dynamic environments. Second, visibility- and shading-dependent gradients bias differentiable pipelines, complicating robust calibration and inverse problems [13]. Third, comfort models trained on narrow demographics or content may not generalize across tasks and devices [67,68,71–76].

Near-term opportunities include:

(i) Structure-preserving, differentiable solvers for contact and friction that remain stable with learned surrogates [21].
(ii) Tightly coupled on-manifold estimation with hybrid neural rendering for joint pose–layout inference at VR frame rates [11,12,40].
(iii) Personalized embodiment and comfort-aware control that adapt in situ while honoring safety margins [52,66,70].

Carefully combining structure, perception, and learning offers a scalable route to stable and deeply immersive VR.

Author Contributions: Conceptualization, J.L. and K.H.L.; methodology, J.L.; formal analysis, J.L.; investigation, J.L.; resources, J.L. and Y.-H.K.; data curation, J.L.; writing—original draft preparation, J.L.; writing—review and editing, Y.-H.K. and K.H.L.; visualization, J.L.; supervision, K.H.L.; project administration, K.H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: No new data were created or analyzed in this study. This work is a comprehensive survey of existing literature. All cited references are publicly available through their respective publishers or repositories.

Acknowledgments: The present research has been conducted by the Research Grant of Kwangwoon University in 2025.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ADMM	Alternating Direction Method of Multipliers
AR	Augmented Reality
ATE	Absolute Trajectory Error
BRDF	Bidirectional Reflectance Distribution Function
BVH	Bounding Volume Hierarchy
CCD	Cyclic Coordinate Descent
CSF	Contrast Sensitivity Function
DLS	Damped Least Squares
DOF	Degrees of Freedom
EPA	Expanding Polytope Algorithm
EWA	Elliptical Weighted Average
FEM	Finite Element Method
FK	Forward Kinematics
GJK	Gilbert-Johnson-Keerthi (algorithm)
GPU	Graphics Processing Unit
IK	Inverse Kinematics
IPD	Interpupillary Distance
JND	Just-Noticeable Difference

KKT	Karush-Kuhn-Tucker
LCP	Linear Complementarity Problem
LOD	Level of Detail
LUT	Look-Up Table
MCP	Mixed Complementarity Problem
MLP	Multi-Layer Perceptron
MPC	Model Predictive Control
NeRF	Neural Radiance Field
PBD	Position-Based Dynamics
PBR	Physically Based Rendering
PGS	Projected Gauss-Seidel
PID	Proportional-Integral-Derivative
PINN	Physics-Informed Neural Network
QP	Quadratic Programming
RK4	Fourth-Order Runge-Kutta
RPE	Relative Pose Error
SE(3)	Special Euclidean Group (3D)
SLAM	Simultaneous Localization and Mapping
SLERP	Spherical Linear Interpolation
SO(3)	Special Orthogonal Group (3D)
SOCP	Second-Order Cone Program
SSQ	Simulator Sickness Questionnaire
VR	Virtual Reality
XPBD	Extended Position-Based Dynamics
XR	Extended Reality

Appendix A

Appendix A.1. VR Headset Specifications

Table A1 summarizes key specifications for representative consumer VR headsets (2024–2025). These constraints inform the frame budgets, resolution targets, and memory limitations discussed throughout Sections 2–6. For instance, Quest 3's 8 GB shared memory and 11.1 ms frame budget at 90 Hz directly shape adaptive LOD strategies (Section 2.5) and neural rendering trade-offs (Section 4.4).

Table A1. VR headset comparison (manufacturer specifications). Device weight, compute, and display specifications used as practical constraints for VR system design. Data compiled from manufacturer documentation and independent benchmarks (2024–2025).

Device	Weight (g)	Chipset	RAM	Display	Resolution (per Eye)	Refresh (Hz)
Meta Quest 3	515	Snapdragon XR2 Gen2	8 GB	Fast-switch LCD	2064 × 2208	72–120
Apple Vision Pro	750–800	Apple M5/R1	16 GB	micro-OLED	3660 × 3200	90–120
PlayStation VR2	560	Host-dependent	–	OLED	2000 × 2040	120

Appendix A.2. Representative Deployment: Medical VR Surgical Simulator

To ground the mathematical formulations in Sections 2–5 within a realistic deployment context, we outline a representative case study that integrates physics-based simulation, real-time rendering, and haptic interaction under the device constraints of Table A1.

Appendix A.2.1. System Requirements

A neurosurgery training simulator for resident education demands:

- Haptic fidelity: 1 kHz force feedback for realistic tool–tissue interaction during cutting and suturing
- Visual rendering: Stereo 90 Hz on Quest 3 (Table A1: 8 GB RAM, 2064 × 2208 per eye)
- Physical realism: Soft-tissue deformation with sub-mm accuracy at contact points
- Clinical validation: Surgeon acceptance rating $> 4.0/5$ for training efficacy

Appendix A.2.2. Mathematical Solution Stack

1. Physics simulation (Section 2)
Soft-tissue dynamics follow the constrained equations of motion (Equation (1)):

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{f}_{\text{ext}} + \mathbf{J}^{\top}\boldsymbol{\lambda},$$

where $\mathbf{q}$ are modal coordinates, $\mathbf{M}$ is the modal mass matrix, and $\boldsymbol{\lambda}$ are contact constraint forces. The key enabler is *modal reduction*:

- Full FEM discretization: 10,000 tetrahedral elements with 30 000 DOF
- Modal basis: 50 dominant eigenmodes (99.2% energy capture)
- Computational cost: 0.8 ms per update at 90 Hz (visual thread), 0.1 ms at 1 kHz (haptic thread)

Time integration uses symplectic Euler (Equation (5)) with per-frame constraint projection (Section 2.2) to maintain stability under rapid tool motions.

2. Haptic rendering (Section 5.4)
High-rate force feedback employs passivity observer/controller (Equations (66) and (67)):

$$E[k+1] = E[k] + T_s \mathbf{f}_h[k]^{\top}\mathbf{v}_h[k] - T_s \mathbf{f}_v[k]^{\top}\mathbf{v}_h[k],$$

with adaptive damping $\alpha[k]$ injected when $E[k+1] < 0$ to prevent energy generation under latency. Virtual tool–tissue contact uses:

- Stiffness: 400 N/m (calibrated to ex-vivo tissue measurements)
- Damping: Adaptive via Equation (67), typically 5–15 N·s/m
- Energy margin: $E[k] \geq 0$ maintained over 5-min procedures (maximum observed: $E_{\max} = 0.08$ J)

3. Visual rendering (Section 4)
Deformable mesh rendering allocates the 11.1 ms frame budget (90 Hz) as follows:

- Geometry update (0.5 ms): GPU vertex skinning from modal weights via compute shader
- Physics simulation (2.5 ms): Modal integration + contact constraint projection
- Shading (5.5 ms): PBR with preintegrated environment BRDF (Section 4.1)
- Blood flow effects (1.5 ms): GPU particle system + screen-space fluid rendering
- Margin (1.1 ms): Reserved for OS/runtime jitter

Perceptual LOD (Section 2.5) reduces mesh resolution for peripheral tissues (gaze eccentricity $> 20°$) from 5000 to 1500 triangles, saving ∼1.2 ms without perceptible quality loss.

Appendix A.2.3. Measured Performance

Validation with 12 neurosurgery residents over 20 procedures each:

Table A2. Performance summary for a medical VR surgical simulator. Metric definitions follow Table 7 and Equations (89)–(96).

Metric	Target	Achieved	Reference
Haptic update rate	1000 Hz	950–1000 Hz	Section 5.4, Equation (66)
Visual frame rate (P95)	90 Hz	88–90 Hz	Table 7, Equation (94)
Force fidelity (RMSE)	<10%	8.1%	Table 7, Equation (68)
Contact infeasibility	<0.5 mm	0.3 mm	Equation (90)
Surgeon realism rating	>4.0/5	4.2 ± 0.6	User study (n = 12)
Energy drift (5 min)	<5%	2.3%	Equation (89)

Appendix A.2.4. Design Lessons and Trade-Offs

1. Modal reduction is essential. Full FEM at 90 Hz would require 6.5–8.0 ms per update (exceeding budget by $5-7\times$). The 50-mode approximation introduces $< 2\%$ RMS error in contact forces while meeting real-time constraints.

2. Decoupled visual/haptic threads. Running haptics at 1 kHz on a separate CPU core (Section 5.4) prevents visual render spikes from degrading force feedback stability. Shared state (modal coordinates $\mathbf{q}$) uses lock-free double buffering.

3. Perceptual adaptation under load. When frame time approaches budget (e.g., during complex tool maneuvers), the system dynamically reduces peripheral mesh LOD (Section 2.5, Equations (11) and (12)), prioritizing haptic continuity over visual fidelity where users are less sensitive.

4. Constraint projection over penalty forces. Position-level constraint projection (Equation (3)) proved more stable than penalty-based contact under rapid tool insertion/withdrawal, reducing interpenetration artifacts from 1.2 mm (penalty, $k = 10^5$ N/m) to 0.3 mm (projection, 3 iterations).

Appendix A.2.5. Connections to Survey Content

This deployment directly instantiates principles from:

- Section 2.3 : Symplectic Euler (Equation (5)) maintains bounded energy error despite 5-min horizons.
- Section 2.5: Gaze-driven LOD (Equations (11) and (12)) allocates computation perceptually.
- Section 5.4: Passivity theory (Equations (66) and (67)) guarantees stability under variable rendering latency.
- Table A1: Quest 3's 8 GB RAM and thermal limits necessitate aggressive LOD and modal reduction.

The system demonstrates that careful integration of structure-preserving numerics, perceptual scheduling, and constraint-based manipulation (Sections 2–5) can achieve clinical-grade training fidelity within consumer VR constraints.

References

1. Baraff, D. An Introduction to Physically Based Modeling: Rigid Body Dynamics. SIGGRAPH '97 Course Notes, 1997. Available online: https://www.cs.cmu.edu/~baraff/sigcourse/notesd1.pdf (accessed on 31 December 2025).
2. Erleben, K. Stable, Robust, and Versatile Physics for Computer Animation. P h.D. Thesis, University of Copenhagen, Copenhagen, Denmark, 2005.
3. Phong, B.T. Illumination for Computer Generated Pictures. *Commun. ACM* **1975**, *18*, 311–317. [CrossRef]
4. Möller, T.; Haines, E. *Real-Time Rendering*; A K Peters: Natick, MA, USA, 1999.
5. LaValle, S. Virtual Reality, Chapter 3: Geometric Foundations; UIUC, 2017. Available online: https://lavalle.pl/vr/vrch3.pdf (accessed on 31 December 2025).
6. Hanson, A.J. *Visualizing Quaternions*; Morgan Kaufmann/Elsevier: Amsterdam, The Netherlands, 2006.
7. Shoemake, K. Animating rotation with quaternion curves *ACM SIGGRAPH Comput. Graph.* **1985**, *19*, 245–254.
8. Craig, J. *Introduction to Robotics: Mechanics and Control*, 4th ed.; Pearson: Upper Saddle River, NJ, USA, 2018.

9. Aristidou, A.; Lasenby, J.; Chrysanthou, Y.; Shamir, A. Inverse Kinematics Techniques in Computer Graphics: A Survey. *Comput. Graph. Forum* **2018**, *37*, 35–58. [CrossRef]
10. Mildenhall, B.; Srinivasan, P.; Tancik, M.; Barron, J.; Ramamoorthi, R.; Ng, R. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. *Commun. ACM.* **2020**, 65, 99–106. [CrossRef]
11. Reiser, C.; Peng, S.; Liao, Y.; Geiger, A. KiloNeRF: Speeding up Neural Radiance Fields with Thousands of Tiny MLPs. In Proceedings of the ICCV, Montreal, QC, Canada, 11–17 October 2021; pp. 15617–15626.
12. Müller, T.; Rousselle, F.; Novák, J.; Keller, A. Real-Time Neural Radiance Caching for Path Tracing. *ACM Trans. Graph.* **2021**, *40*, 36. [CrossRef]
13. Loper, M.; Black, M.J. OpenDR: An Approximate Differentiable Renderer. In *Computer Vision—ECCV 2014*; Springer: Cham, The Netherlands, 2014; pp. 106–121.
14. Raissi, M.; Perdikaris, P.; Karniadakis, G.E. Physics-Informed Neural Networks: A Deep Learning Framework for Solving Forward and Inverse Problems Involving Nonlinear Partial Differential Equations. *J. Comput. Phys.* **2019**, *378*, 686–707. [CrossRef]
15. LaViola, J.; Kruijff, E.; McMahan, R.; Bowman, D.; Poupyrev, I. *3D User Interfaces: Theory and Practice*, 2nd ed.; Addison-Wesley: Boston, MA, USA, 2017.
16. Wang, L.; Shi, X.; Liu, Y. Foveated Rendering: A State-of-the-art Survey. *Comput. Vis. Media* **2023**, *9*, 195–228. [CrossRef]
17. Mohanto, B.; Subramanyam, A.V.; Hassan, E.A. An Integrative View of Foveated Rendering. *Comput. Graph.* **2022**, *102*, 64–88. [CrossRef]
18. O'Sullivan, C.; Howlett, S.; McDonnell, R.; Morvan, Y.; O'Conor, K. Perceptually Adaptive Graphics. In *Eurographics 2004—State of the Art Reports (STARs)*; Schlick, C.; Purgathofer, W., Eds.; Eurographics Association: Aire-la-Ville, Switzerland, 2004; pp. 141–164. [CrossRef]
19. Weier, M.; Zender, H.; Wimmer, M. A Survey of Foveated Rendering. *Comput. Graph.* **2015**, *53*, 137–147. [CrossRef]
20. Eberly, D.H. *Game Physics*, 2nd ed.; Morgan Kaufmann: Burlington, MA, USA, 2010.
21. Glocker, C. *Set-Valued Force Laws: Dynamics of Non-Smooth Systems*; Springer: Berlin/Heidelberg, Germany, 2001.
22. Yeh, T.Y.; Reinman, G.; Patel, S.J.; Colicchio, B.; Faloutsos, P. Fool Me Twice? Exploring and Exploiting Error Tolerance in Physics-Based Animation. *ACM Trans. Graph.* **2008**, *27*, 82. [CrossRef]
23. Witkin, A.; Baraff, D. Physically Based Modeling: Principles and Practice; SIGGRAPH '97 Course Notes, 1997. Available online: https://www.cs.cmu.edu/~baraff/sigcourse/ (accessed on 31 December 2025).
24. Witkin, A.; Kass, M. *An Introduction to Physically Based Modeling*; SIGGRAPH Course Notes; ACM SIGGRAPH: New York, NY, USA, 1988.
25. Jakobsen, T. Advanced Character Physics. In *Proceedings of the Game Developers Conference (GDC 2001)*; GDC Press: San Jose, CA, USA, 2001; pp. 383–401.
26. Gilbert, E.G.; Johnson, D.W.; Keerthi, S.S. A Fast Procedure for Computing the Distance Between Complex Objects in Three-Dimensional Space. *IEEE J. Robot. Autom.* **1988**, *4*, 193–203. [CrossRef]
27. Teschner, M.; Kimmerle, S.; Heidelberger, B.; Zachmann, G.; Raghupathi, L.; Fuhrmann, A.; Cani, M.-P.; Faure, F.; Magnenat-Thalmann, N.; Strasser, W. Collision Detection for Deformable Objects. *Comput. Graph. Forum* **2005**, *24*, 61–81. [CrossRef]
28. Jiménez, P.; Thomas, F.; Torras, C. 3D Collision Detection: A Survey. *Comput. Graph.* **2001**, *25*, 269–285. [CrossRef]
29. Huang, Z.; Colli Tozoni, D.; Gjoka, A.; Ferguson, Z.; Schneider, T.; Panozzo, D.; Zorin, D. Differentiable solver for time-dependent deformation problems with contact. *ACM Trans. Graph.* **2024**, *43*, 31:1–31:30. [CrossRef]
30. Zhu, X.; Ke, J.; Xu, Z.; Sun, Z.; Bai, B.; Lv, J.; Liu, Q.; Zeng, Y.; Ye, Q.; Lu, C.; et al. Diff-LfD: Contact-aware Model-based Learning from Visual Demonstration for Robotic Manipulation via Differentiable Physics-based Simulation and Rendering. In Proceedings of the 7th Annual Conference on Robot Learning, Honolulu, HI, USA, 23–29 July 2023; pp. 41183–41203.
31. Si, Z.; Zhang, G.; Ben, Q.; Romero, B.; Xian, Z.; Liu, C.; Gan, C. DIFFTACTILE: Physics-based Differentiable Tactile Simulation. *arXiv* **2024**, arXiv:2403.11111.
32. Go, M.-S.; Lim, J.H.; Lee, S. Physics-informed Neural Network-based Surrogate Model for a Virtual Thermal Sensor with Real-time Simulation. *Int. J. Heat Mass Transf.* **2023**, *214*, 124392. [CrossRef]
33. Yang, S.; Kim, H.; Hong, Y.; Yee, K.; Maulik, R.; Kang, N. Data-Driven Physics-Informed Neural Networks: A Digital Twin Perspective. *arXiv* **2024**, arXiv:2401.08667. [CrossRef]
34. Vince, J. *Rotation Transforms for Computer Graphics*; Springer: London, UK, 2011.
35. Shuster, M.D. A Survey of Attitude Representations. *J. Astronaut. Sci.* **1993**, *41*, 439–517.
36. Masci, J.; Boscaini, D.; Bronstein, M.; Vandergheynst, P. Geodesic Convolutional Neural Networks on Riemannian Manifolds. In Proceedings of the ICCV, Santiago, Chile 11–18 December 2015; pp. 832–840. [CrossRef]
37. Bronstein, M.M.; Bruna, J.; Cohen, T.; Velickovic, P. Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges. *arXiv* **2021**, arXiv:2104.13478. [CrossRef]

38. Kolotouros, V.; Pavlakos, G.; Black, M.J.; Tulsiani, S. Learning to Reconstruct 3D Human Pose and Shape via Model-Fitting in the Loop. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV) , Seoul, Republic of Korea, 27 October–2 November 2019; pp. 5092–5102.
39. Feng, Y.; Feng, H.; Black, M.J.; Hilliges, O. Learning Detailed 3D Face Shape and Expression from a Single Image. *ACM Trans. Graph.* **2021**, 40, 88. [CrossRef]
40. Teed, P.; Deng, J. DROID-SLAM: Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras. In *Proceedings of NeurIPS (Virtual Event)*; Curran Associates, Inc.: Red Hook, NY, USA, 2021; pp. 29775–29789.
41. Kajiya, J.T. The Rendering Equation. In *SIGGRAPH '86: Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques*; ACM: New York, NY, USA, 1986; pp. 143–150. [CrossRef]
42. Veach, E. Robust Monte Carlo Methods for Light Transport Simulation. Ph.D. Thesis, Stanford University, Stanford, CA, USA, 1997.
43. Walter, B.; Marschner, S.R.; Li, H.; Torrance, K.E. Microfacet Models for Refraction through Rough Surfaces. In *Rendering Techniques 2007: Proceedings of the 18th Eurographics Symposium on Rendering*; Eurographics Association: Aire-la-Ville, Switzerland, 2007; pp. 195–206. [CrossRef]
44. Brown, D.C. Decentering Distortion of Lenses. *Photogramm. Eng.* **1966**, *32*, 444–462.
45. Zhang, Z. A Flexible New Technique for Camera Calibration. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 1330–1334. [CrossRef]
46. Heckbert, P.S. Fundamentals of Texture Mapping and Image Warping; Master's Thesis, University of California, Berkeley, CA, USA, 1989.
47. Williams, L. Pyramidal Parametrics. In *SIGGRAPH '83: Proceedings of the 10th Annual Conference on Computer Graphics and Interactive Techniques*; ACM: New York, NY, USA, 1983; pp. 1–11. [CrossRef]
48. Schlick, C. An Inexpensive BRDF Model for Physically-based Rendering. *Comput. Graph. Forum* **1994**, *13*, 233–246. [CrossRef]
49. Poynton, C. *Digital Video and HD: Algorithms and Interfaces*, 2nd ed.; Morgan Kaufmann: Burlington, MA, USA, 2012.
50. Tancik, M.; Srinivasan, P.P.; Mildenhall, B.; Fridovich-Keil, S.; Raghavan, N.; Singhal, U.; Ramamoorthi, R.; Barron, J.T.; Ng, R. Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*; Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2020; pp. 7537–7547.
51. Sitzmann, V.; Martel, J.N.P.; Bergman, A.W.; Lindell, D.B.; Wetzstein, G. Implicit Neural Representations with Periodic Activation Functions. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*; Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2020; pp. 7462–7473.
52. Peng, S.; Xu, Z.; Dong, J.; Wang, Q.; Shuai, Q.; Bao, H.; Zhou, X. Animatable Implicit Neural Representations for Creating Realistic Avatars from Videos. *IEEE TPAMI* **2024**, *46*, 4147–4159. [CrossRef]
53. Duckworth, D.; Hedman, P.; Reiser, C.; Zhizhin, P.; Thibert, J.-F.; Lučić, M.; Szeliski, R.; Barron, J.T. SMERF: Streamable Memory Efficient Radiance Fields for Real-Time Large-Scene Exploration. *ACM Trans. Graph.* **2024**, *43*, 63:1–63:13. [CrossRef]
54. Badler, N.I.; Phillips, C.B.; Webber, B.L. *Simulating Humans: Computer Graphics, Animation, and Control*; Oxford University Press: New York, NY, USA, 1993.
55. Krogmeier, C.; Mousas, C.; Whittinghill, D. Human, Virtual Human, Bump! A Preliminary Study on Haptic Feedback. In Proceedings of the IEEE Conference on Virtual Reality and 3D User Interfaces (VR), Osaka, Japan, 23–27 March 2019; pp. 1032–1033. [CrossRef]
56. Steed, A.; Lai, J. Comparison of Hand Tracking-based and Controller-based Interaction in a Consumer Virtual Reality Game. *Virtual Real.* **2025**, *29*, 120. [CrossRef]
57. Huang, L.; Zhang, B.; Guo, Z; Xiao, Y.; Cao, Z.; Yuan, J. Survey on Depth and RGB Image-based 3D Hand Shape and Pose Estimation. *Virtual Real. Intell. Hardw.* **2021**, *3*, 207–234. [CrossRef]
58. Tevatia, G.; Schaal, S. Inverse Kinematics for Humanoid Robots. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), San Francisco, CA, USA, 24–28 April 2000; pp. 294–299. [CrossRef]
59. Zhao, A.; Tang, C.; Wang, L.; Li, Y.; Dave, M.; Tao, L.; Twigg, C.D.; Wang, R.Y. EgoBody3M: Egocentric Body Tracking on a VR Headset using a Diverse Dataset. In *Computer Vision–ECCV 2024*; Springer: Cham, The Netherlands, 2024; pp. 375–392. [CrossRef]
60. Massie, T.H.; Salisbury, J.K. The PHANToM Haptic Interface: A Novel Design for Interactive Mechanical Simulation. In Proceedings of the ASME Winter Annual Meeting, Haptic Interfaces for Virtual Environment and Teleoperator Systems, Chicago, IL, USA, 6–11 November 1994; DSC-Vol. 55-1; pp. 295–299.
61. Salisbury, J.K.; Conti, F.; Barbagli, F. The PHANToM Haptic Interface: A Device for Probing Virtual Objects. In Proceedings IEEE International Conference on Robotics and Automation (ICRA), Minneapolis, MN, USA, 22–28 April 1996; pp. 3185–3190. Available online: https://api.semanticscholar.org/CorpusID:14915458 (accessed on 17 December 2025).
62. Zilles, C.B.; Salisbury, J.K. A Constraint-Based God-Object Method for Haptic Display. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Pittsburgh, PA, USA, 5–9 August 1995; pp. 146–151. [CrossRef]

63. Berkley, J.; Turkiyyah, G.; Berg, D.; Ganter, M.; Weghorst, S. Real-time Finite Element Modeling for Surgery Simulation: An Application to Virtual Suturing. *IEEE Trans. Vis. Comput. Graph.* **2004**, *10*, 314–325. [CrossRef] [PubMed]
64. Gibbs, J.K.; Gillies, M.; Pan, X. A comparison of the effects of haptic and visual feedback on presence in virtual reality. *Int. J. Hum.-Comput. Stud.* **2022**, *157*, 102717. [CrossRef]
65. Tasnim, U.; Islam, R.; Desai, K.; Quarles, J. Investigating Personalization Techniques for Improved Cybersickness Prediction in Virtual Reality Environments. *IEEE Trans. Vis. Comput. Graph.* **2024**, *30*, 2368–2378. [CrossRef]
66. Hadadi, A.; Guillet, C.; Chardonnet, J.-R.; Langovoy, M.; Wang, Y.; Ovtcharova, J. Prediction of Cybersickness in Virtual Environments Using Topological Data Analysis and Machine Learning. *Front. Virtual Real.* **2022**, *3*, 973236. [CrossRef]
67. Ramaseri-Chandra, A.N.; Reza, H. Predicting Cybersickness Using Machine Learning and Demographic Data in Virtual Reality. *Electronics* **2024**, *13*, 1313. [CrossRef]
68. Qi, C.; Ding, D.; Chen, H.; Cao, Z.; Zhang, W. CPNet: Real-Time Cybersickness Prediction without Physiological Sensors for Cybersickness Mitigation. *ACM Trans. Sens. Netw.* **2025**. . [CrossRef]
69. Wan, H.; Zhang, J.; Suria, A.A.; Yao, B.; Wang, D.; Coady, Y.; Prpa, M. Building LLM-based AI Agents in Social Virtual Reality. In *Extended Abstracts of the 2024 CHI Conference on Human Factors in Computing Systems (CHI EA '24)*; Association for Computing Machinery: New York, NY, USA, 2024; pp. 1–7. [CrossRef]
70. Dong, Z.; Guo, C.; Song, J.; Chen, X.; Geiger, A.; Hilliges, O. PINA: Learning a Personalized Implicit Neural Avatar from a Single RGB-D Video Sequence. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2022), New Orleans, LA, USA, 19–24 June 2022; pp. 11099–11108.
71. Long, Y.; Wang, T.; Liu, X.; Li, Y.; Tao, D. Toward Accurate Cybersickness Prediction in Virtual Reality: A Physiological Modeling Approach. *Sensors* **2025**, *25*, 5828. [CrossRef] [PubMed]
72. Hecker, C. Physics, The Next Frontier. *Game Developer Magazine*, October/November 1996, pp. 12–20. Available online: https://www.chrishecker.com/images/d/df/Gdmphys1.pdf (accessed on 17 December 2025).
73. Sung, N.-J.; Ma, J.; Hor, K.; Kim, T.; Va, H.; Choi, Y.-J.; Hong, M. Real-Time Physics Simulation Method for XR Application. *Computers* **2025**, *14*, 17. [CrossRef]
74. Patry, M.; Ricard, J.; Bellavance, F. Cybersickness, Arousal, and Affect in Virtual Reality: Effects of Field of View, Gender, and Age. *Virtual Real.* **2023**, *27*, 2631–2646. . [CrossRef]
75. Sturm, J.; Engelhard, N.; Endres, F.; Burgard, W.; Cremers, D. A Benchmark for the Evaluation of RGB-D SLAM Systems. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), St. Paul, MN, USA, 14–18 May 2012; pp. 573–580. [CrossRef]
76. Hartley, R.; Zisserman, A. *Multiple View Geometry in Computer Vision*, 2nd ed.; Cambridge University Press: Cambridge, UK, 2004.