ingegneria sismica
International Journal of Earthquake Engineering
Anno XLIII - Num. 2 - 2026

Interactive animation design based on virtual reality technology

Manlu Kong[1,*]

[1] School of Art, Anhui Wenda College of Information Engineering, Hefei 230000, Anhui, China

SUMMARY: *With the development of virtual reality devices and real-time rendering technology, animation design has gradually shifted from linear playback to immersive interactive generation. Focusing on the problem of interactive animation design based on virtual reality technology, this paper constructs a method framework that integrates hierarchical management of 3D scene resources, skeleton binding and skin computation, motion capture input coding, gesture recognition, finite-state maneuver painting control and adaptive rendering load adjustment. The interactive animation prototype is implemented based on Unity 3D, OpenXR and C# script, and the user pose, hand trajectory, trigger events and running log are collected through the head display, two-hand controller and spatial positioning device. In the experiment, 36 users, 4 types of virtual scenes and 12480 valid interaction logs were selected for verification. The results show that the average frame rate of low complexity scene is 91.8FPS, and high dynamic complex scene still maintains 76.9 FPS. The interactive response delay is concentrated in 18-32 ms, the highest GPU occupancy rate is 84.7%, and the average user experience score is 4.42. Research shows that this method can improve the real-time performance, stability and immersive experience of virtual reality interactive animation, which has reference significance for intelligent animation design and virtual interaction system optimization.*

KEYWORDS: *Virtual reality technology; Interactive animation design; Real-time rendering; Motion capture*

1 Introduction

With the development of computer graphics, virtual reality devices, 3D modeling engines and real-time interaction algorithms, animation design is expanding from the traditional linear playback mode to the direction of immersion, interactive and intelligent [1]. Traditional animation usually relies on fixed lens, preset timeline and one-way narrative structure. Users can only watch passively, and it is difficult to change the scene state, character behavior and animation rhythm according to their own actions [2]. Virtual reality technology provides dynamic interaction conditions in three-dimensional space for animation design through head-mounted display devices, spatial positioning systems, motion capture sensors and real-time rendering engines. Users can enter the animation scene from the first perspective, and participate in the animation process through gestures, gaze, body posture or controller input. The resulting interactive animation design not only expands the visual expression of animation, but also puts forward higher requirements for system modeling accuracy, rendering efficiency, interactive response and behavior data processing capabilities [3-5].

---

*kml198874@126.com

https://doi.org/10.65102/is2026905

In the virtual reality environment, interactive animation design involves technical aspects such as 3D scene construction, animation resource organization, user input recognition, spatial coordinate mapping, action state driving, real-time rendering optimization and feedback generation [6]. Scene models need to maintain a balance between visual accuracy and computational efficiency. Too many model faces will increase GPU rendering load, and insufficient model details will weaken immersion experience [7, 8]. Animation control also needs to get rid of the single keyframe playback logic and trigger character actions, environment changes and event feedback in real time according to user behavior. In the interactive input part, gesture trajectory, head pose, controller displacement and trigger signals should be encoded and converted into control parameters that can be executed by the animation system [9]. If the system response delay is too high, there will be a significant disconnection between the user's action and the animation feedback, which will affect the operation continuity and spatial reality.

Existing research has achieved certain results in virtual reality display, immersive narrative, 3D animation production and interactive experience evaluation, but there is still the problem of insufficient technical cohesion [10]. Some studies focus on visual effects, and the analysis of the mechanism of user behavior data-driven animation response is not enough. Some systems can achieve basic interaction, but lack the verification of the relationship between frame rate stability, interaction delay, action recognition accuracy and immersion experience. Some design processes still rely on manual experience to complete scene construction and action binding, and lack digital modeling and adaptive optimization methods for virtual reality environments [11, 12]. Therefore, interactive animation research needs to further highlight the computer technology route, system implementation logic and experimental verification process.

Based on this, this paper focuses on the interactive animation design based on virtual reality technology, and constructs an animation design method for virtual reality environment. Starting from 3D scene modeling and digital expression of animation resources, this paper designs a user-driven animation response generation and dynamic control mechanism by combining motion capture, gesture recognition, spatial coordinate mapping and real-time rendering engine. The animation trigger efficiency, response stability and immersion experience are optimized through interactive data analysis. The experimental part builds a virtual reality interactive animation prototype system, collects user operation data and system operation data, and verifies the rendering frame rate, interactive response delay, GPU occupancy rate and user experience score, which provides technical reference for virtual education, digital display, game development and intelligent animation system design.

2 Related Research

The research of interactive animation in virtual reality environment mainly focuses on character reality, social presence and user behavior mapping in the early stage. Rogers et al. believe that virtual avatars with real motion performance can enhance the sense of presence in virtual social interaction and provide basic support for the expression of body movements in interactive animation [13]. Fraser et al. further pointed out that the expression and body expression ability of real-time motion capture avatars would affect users' judgments on the realism and interaction quality of animations, indicating that animated characters are no longer just visual objects, but also assume the function of behavior feedback and emotional transmission [14]. Higgins et al. analyzed the influence of emotional valence on self-perception and social experience from the video conference scene of personalized avatar, which provided reference for the image design of virtual characters [15]. Gomes de Siqueira et al. applied avatar interaction to team formation and communication in immersive virtual environments, emphasizing the link between role interaction, task collaboration and information transmission in desktop virtual environments [16].

In terms of user experience and immersion mechanism, Vindenes and Wasson proposed a post-phenomenological framework for studying immersive virtual reality user experience, emphasizing the interactive relationship between technical medium, body perception and user behavior [17]. Fribourg et al. discussed the influence of virtual threats on embodied experience and threat response, and showed that the design of animated events would change users' psychological feelings and behavioral feedback [18]. Sinatra et al. studied the influence of social authenticity of virtual agents on the sense of presence and learning effect, indicating that there was a close relationship between the credibility of character behavior and the functional effect of interaction scenarios [19]. Fraser et al. continued to investigate the role of humanoid avatars in virtual reality social interaction in subsequent studies, and further verified the importance of realistic role modeling for interactive experience optimization [20]. Sterna et al. analyzed co-presence from the perspective of behavioral realism, photographic realism and interaction realism, and introduced psychophysical measurement methods to make the evaluation of virtual reality interaction effects more quantitative [21].

In recent years, VR interactive animation research has gradually expanded to education, inclusive design, spatial perception, and neural rendering. Kasapakis et al. constructed a virtual reality prototype of sign language learning, which showed that interactive animation can support skill learning through gesture recognition, action demonstration and immersive scenes [22]. Hartfill et al. focused on the inclusive user-centered design of two-handed interaction technology, providing method inspiration for action input adaptation under different user groups [23]. Wolf et al. analyzed the influence of self-observation distance on the embodied perception of personalized avatars, suggesting that the distance between virtual character and user space would change the animation feedback experience [24]. Mal et al. discussed the problem of realistic consistency between virtual avatars and virtual others, providing a basis for style unification and perceptual coordination in multi-character interactive animation [25]. Landeck et al. studied the relationship between object motion control and time perception, indicating that animation motion parameters would affect users' judgment of time passage [26]. The follow-up study further verified the effect of simulated motion on virtual time experience [27]. Li et al. proposed the Magic NeRF Lens method, which used neural radiance field for interactive fusion in virtual facility inspection, providing a new technical path for high-realistic 3D scene reconstruction and interactive extension of animation [28]. On the whole, the existing research provides the foundation of character modeling, motion capture, user experience and scene reconstruction for virtual reality interactive animation design. However, there is still room for further development in real-time rendering performance, behavior data-driven animation response and adaptive optimization mechanism.

3 Virtual reality technology and interactive animation design theoretical basis

Based on computer graphics, spatial localization, sensor perception and real-time rendering, virtual reality technology enables users to shift from traditional screen viewing to immersive spatial participation by constructing a perceiveable, manipulable and feed-back 3D digital environment. Interactive animation design in virtual reality environment is no longer limited to a fixed timeline, but needs to change the animation state in real time according to user posture, gesture trajectory, gaze direction and controller input. The core of the system is to convert user actions into computable data, and then use animation control logic to drive characters, objects and scenes to generate synchronous feedback.

In terms of technical implementation, virtual reality interactive animation relies on spatial coordinate mapping mechanism. The user's head position, hand action and controller displacement in the real space need to be mapped to the corresponding object in the virtual space after sensor acquisition, coordinate transformation and scene matching. This process can be expressed as follows:

$$\mathrm{P_v} = \mathrm{M_{w \to v}} \cdot \mathrm{P_r} \tag{1}$$

where, P_v represents the interaction point coordinates in the virtual space, P_r represents the collected user action coordinates in the real space, and $\mathrm{M_{w \to v}}$ represents the coordinate transformation matrix from the real space to the virtual space. Through this mapping, the system can maintain the spatial consistency between the user's action and the virtual animation response, and provide the data basis for subsequent animation triggering and motion control.

Interactive animation design also needs to meet the real-time requirements. In the virtual reality scene, there is a continuous processing link between user action, system recognition, animation calculation and screen rendering. Once the response delay is too high, users will feel the screen lag, action disconnection, and even dizziness. Therefore, the interactive response efficiency can be described by the total delay:

$$\mathrm{T_{total}} = \mathrm{T_s} + \mathrm{T_r} + \mathrm{T_a} + \mathrm{T_g} \tag{2}$$

where, T_s is the sensor acquisition delay, T_r is the action recognition and interaction judgment delay, T_a is the animation state update delay, T_g is the graphics rendering output delay. Interactive animation systems need to reduce the computational overhead in each step to keep the total delay within the acceptable range of users, so as to ensure the continuity of animation feedback and operation stability.

From the perspective of design theory, virtual reality interactive animation should pay attention to visual realism, behavior consistency and interaction controllability at the same time. Visual realism is mainly determined by model accuracy, material mapping, illumination calculation and rendering frame rate. Behavior consistency is reflected by the logical matching between user input and animation feedback. Interactive controllability requires that the system can recognize different user behaviors and generate corresponding animation results according to the scene state. The three jointly determine the user's immersion experience in the virtual space.

4 Interactive animation design method based on virtual reality technology

4.1 Virtual reality 3D scene modeling and digital expression of animation resources

The starting point of virtual reality interactive animation design is the standardized modeling of 3D scene resources and the digital expression of animation assets. Due to the free viewing

Angle, spatial movement and real-time operation behavior of users in the virtual reality environment, the scene resources need to form a searchable, callable and interactive resource hierarchy in the 3D engine. In this paper, the virtual reality animation scene is divided into four categories: environment resources, interactive object resources, character animation resources and interface feedback resources. Environment resources mainly include terrain, lighting, background props and space boundaries. Interactive object resources include controller triggers, gripper objects and dynamic response components. Character animation resources are composed of mesh model, material map, skeleton structure, skin weight and animation segment. Interface feedback resources are used to display task prompts, interaction states, and operation results. Through hierarchical organization, the system can quickly locate resource nodes in the running phase and trigger corresponding animation feedback according to user input.

In the process of 3D modeling, the model resources need to go through the steps of geometric modeling, mesh simplification, UV expansion, material mapping, bone binding and animation segment import. Virtual reality devices are sensitive to frame rate and latency, and the number of model faces, texture resolution, skeleton nodes and animation segment scale will directly affect GPU rendering load. Therefore, this paper estimates the scenario complexity before resource import, and expresses the running load of a single resource as follows:

$$\mathrm{L_i = \alpha F_i + \beta R_i + \gamma B_i + \delta A_i} \tag{3}$$

where, L_i represents the comprehensive running load of the ith 3D resource, F_i represents the number of triangular faces of the model, R_i represents the resolution coefficient of the tile, B_i represents the number of bone nodes, A_i represents the number of animation segments, α, β, γ, δ represent the influence weight of different factors on the system load respectively. This formulation is used to identify high-load resources in the modeling phase, which provides a basis for subsequent surface reduction, tile compression, and bone simplification.

For characters, props and complex scene models, mesh optimization is an important part to ensure the smooth operation of virtual reality animation. In this paper, mesh compression ratio is used to describe the lightweight degree of the model:

$$\mathrm{C_m = \frac{N_o - N_p}{N_o} \times 100\%} \tag{4}$$

where, C_m represents the mesh compression rate of the model, N_o represents the number of triangular faces of the original model, and N_p represents the number of triangular faces of the optimized model. When the compression rate is too low, the model will still occupy high rendering resources. When the compression ratio is too high, the edges of characters, the details of costumes and the contours of interactive objects may be distorted. Therefore, the mesh optimization needs to combine the model purpose and the horizon level. The close interactive objects retain more geometric details, and the distant background resources use a higher compression ratio to reduce the overall rendering pressure. Figure 1 shows the diagram of the resource hierarchy of the virtual reality 3D scene and the skeleton binding.

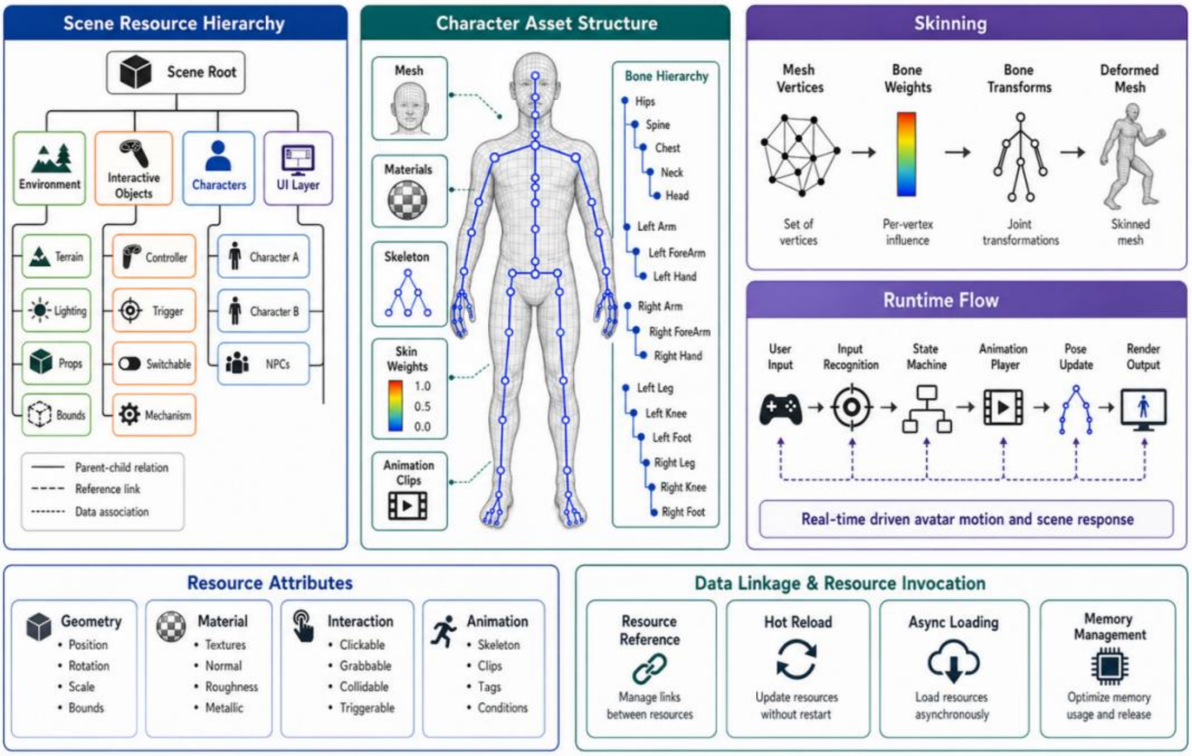

Figure 1: Schematic diagram of resource hierarchy and bone binding in VR 3D scene

As shown in Figure 1, virtual reality 3D scene resources are composed of scene hierarchy, character assets, skeleton binding, resource attributes and resource calling mechanism. Scene resources organize the environment, interactive objects, characters and interface layers with the root node as the center. The role assets further contain mesh, material, skeleton, skin weight and animation segments. The skeleton binding module realizes the character grid driven by vertex influence relationship and skeleton transformation, and the resource attribute and call mechanism provide the data basis for subsequent real-time loading, animation playback and interactive feedback.

The core of digital expression of character animation is skeleton binding and skinning calculation. Each mesh vertex is affected by one or more skeleton nodes, and the influence of different bones is determined by the weight. When the user triggers actions such as grasping, turning, waving, or walking, the animation controller updates the skeleton pose and calculates the deformed character grid position by skimming, which is expressed as:

$$V_i' = \sum_{j=1}^{k} w_{ij} M_j V_i \tag{5}$$

where, V_i'represents the position of the ith mesh vertex after deformation, V_i represents the original vertex position under binding posture, w_{ij} represents the influence weight of the JTH bone on the vertex, M_j represents the transformation matrix of the JTH bone, and k represents the number of bones involved in the deformation of the vertex.

At the resource management level, 3D resources are encapsulated as data objects with geometric attributes, material attributes, interactive attributes and animation attributes. Geometric properties describe the spatial position, rotation Angle and scaling ratio of the model. Material properties describe textures, normals, roughness, and transparency. Interaction properties record clickable, crawlable, collisible, and triggerable states. The animation property holds the skeleton hierarchy, animation segments, status labels, and trigger conditions. Through this digital representation, virtual reality interactive animation can complete resource call, action drive and feedback output in real-time rendering environment, which provides stable data basis for subsequent motion capture coding, animation state control and adaptive optimization of user behavior.

4.2 Interactive input encoding method for motion capture and gesture recognition

The real-time response of virtual reality interactive animation depends on stable input encoding mechanism. In this paper, the user behavior collected by the motion capture device, the handle controller and the gesture recognition module is uniformly transformed into temporal features, so that the animation system can recognize the user's intention and trigger the corresponding role action. The raw input data mainly includes head position, head rotation, left and right hand spatial coordinates, hand pose, controller keystrokes, trigger intensity and collision events. In order to facilitate the subsequent animation control, this paper uses time frame as the basic unit to construct interactive input data:

$$D_t = \{p_h^t, q_h^t, p_l^t, q_l^t, p_r^t, q_r^t, u_t\} \tag{6}$$

where, D_t represents the interactive input data in frame t; $\mathrm{p_h^t}$, $\mathrm{p_l^t}$ and $\mathrm{q_r^t}$ represent the spatial position of head, left hand and right hand respectively; $\mathrm{q_h^t}$, $\mathrm{q_l^t}$ and $\mathrm{q_r^t}$ represent the corresponding rotation posture; u_t represent the controller keys and trigger signals. The data structure can simultaneously describe the user's spatial position, body orientation and operation intention, and provide a unified input for interactive animation control.

In order to reduce the scale differences caused by different users' heights, arm spans and operating habits, the system needs to normalize the motion trajectory:

$$\hat{\mathrm{p}}_\mathrm{t} = \frac{\mathrm{p_t} - \mu_\mathrm{p}}{\sigma_\mathrm{p} + \varepsilon} \tag{7}$$

where, $\hat{\mathrm{p}}_\mathrm{t}$ represents the normalized action position, pt represents the original acquisition position, μ_p represents the mean position in a time window, σ_p represents the position standard deviation, and ε is the stability term to prevent the denominator from being zero. After normalization, the system can recognize interactive actions such as grasping, waving, pointing, and push-pull more stably.

The dynamic change of the action trajectory is reflected by the velocity and acceleration. In this paper, the position change of consecutive frames is transformed into motion features:

$$\mathrm{v_t} = \frac{\mathrm{p_t} - \mathrm{p_{t-1}}}{\Delta\mathrm{t}}, \quad \mathrm{a_t} = \frac{\mathrm{v_t} - \mathrm{v_{t-1}}}{\Delta\mathrm{t}} \tag{8}$$

where, $\mathrm{v_t}$ represents the motion velocity in frame t, $\mathrm{a_t}$ represents the motion acceleration in frame t, and Δt represents the time interval between adjacent frames. This feature can be used to distinguish action states such as fast waving, slow pointing, sudden stopping, and continuous dragging.

In the gesture recognition stage, the system inputs the hand trajectory, attitude Angle, fingertip distance and controller signal into the classification model to obtain the recognition probability of different interaction actions:

$$P(y = c|x_t) = \frac{e^{z_c}}{\sum_{m=1}^{M} e^{z_m}} \tag{9}$$

where $P(y = c|x_t)$ is the probability that the input feature x_t belongs to the CTH gesture, z_c is the output value of the classification model for the CTH gesture, and M is the total number of gesture categories. The system selects the category with the highest probability and exceeding the threshold as the animation trigger instruction, so as to reduce the interference of false recognition on the animation state switching. Table 1 shows the encoding way of motion capture and gesture input features, which is used to illustrate how different input data are transformed into animation control parameters.

Table 1: Motion capture and gesture input feature encoding table

Input Category	Raw Data	Encoded Features	Data Type	Animation Control Function
Head Posture	HMD position and rotation angle	Viewing direction, observation height, turning angle	Continuous	Controls camera view and scene observation direction
Left and Right Hand Trajectories	3D hand coordinate sequences	Displacement, velocity, acceleration	Continuous	Recognizes waving, dragging, pushing, and pulling actions
Gesture Form	Fingertip distance and palm orientation	Grasping state, open-hand state, pointing state	Categorical	Triggers character hand animation and object interaction
Controller Input	Buttons, joystick, trigger value	Click, selection, confirmation, release	Mixed	Controls animation clip activation and state switching
Collision Contact	Collider contact signal	Contact object ID, contact duration	Discrete	Determines grabbable, triggerable, and collidable events
Gaze Direction	Head orientation and target ray	Gaze target, dwell time	Continuous	Triggers gaze feedback and interface prompt animation
Interaction Confidence	Recognition model output probability	Action category probability, recognition threshold	Continuous	Suppresses false triggering of low-confidence actions

As can be seen from Table 1, motion capture and gesture recognition integrates spatial position, pose change, controller events, and recognition probabilities into animation control features. Through unified coding, the system can convert user behavior into computable animation instructions, which provides stable input for subsequent animation state transition, real-time rendering response and interactive experience optimization.

4.3 Animation response generation and dynamic control method based on real-time rendering engine

The core of virtual reality interactive animation is to quickly transform user input into visible and perceivable animation feedback. In this paper, the real-time rendering engine is used as the running platform, and the motion capture features, gesture recognition results and controller events obtained in 4.2 are input into the animation control layer. After state judgment, animation segment matching, pose mixing and rendering output, continuous interactive animation response is formed. Different from traditional animations that play on a fixed timeline, virtual reality animation needs to adjust the character state in real time according to user input to keep the animation behavior synchronized with scene events.

In the animation control layer, finite state machine is used to describe the running process of character animation. The character states mainly include the types of idle, mobile, interactive, attack, expression feedback and abnormal recovery. When the system receives the user input event, it will make a state transition based on the current state, input category and priority condition, which is expressed as:

$$S_{t+1} = \delta(S_t, E_t, \lambda_t) \tag{10}$$

where S_t represents the animation state of frame t, E_t represents the current input event, λ_t represents the event priority parameter, and $\delta(\cdot)$ represents the state transition function. The formula is used to control the switching logic of animation state, so that the input such as moving, grasping, clicking and waving can enter the corresponding animation branch, and avoid the state conflict caused by multiple actions triggered at the same time. Through priority control, the system can ensure the priority response of key interactive actions. The animated state machine transition driven by interactive input is shown in Figure 2.

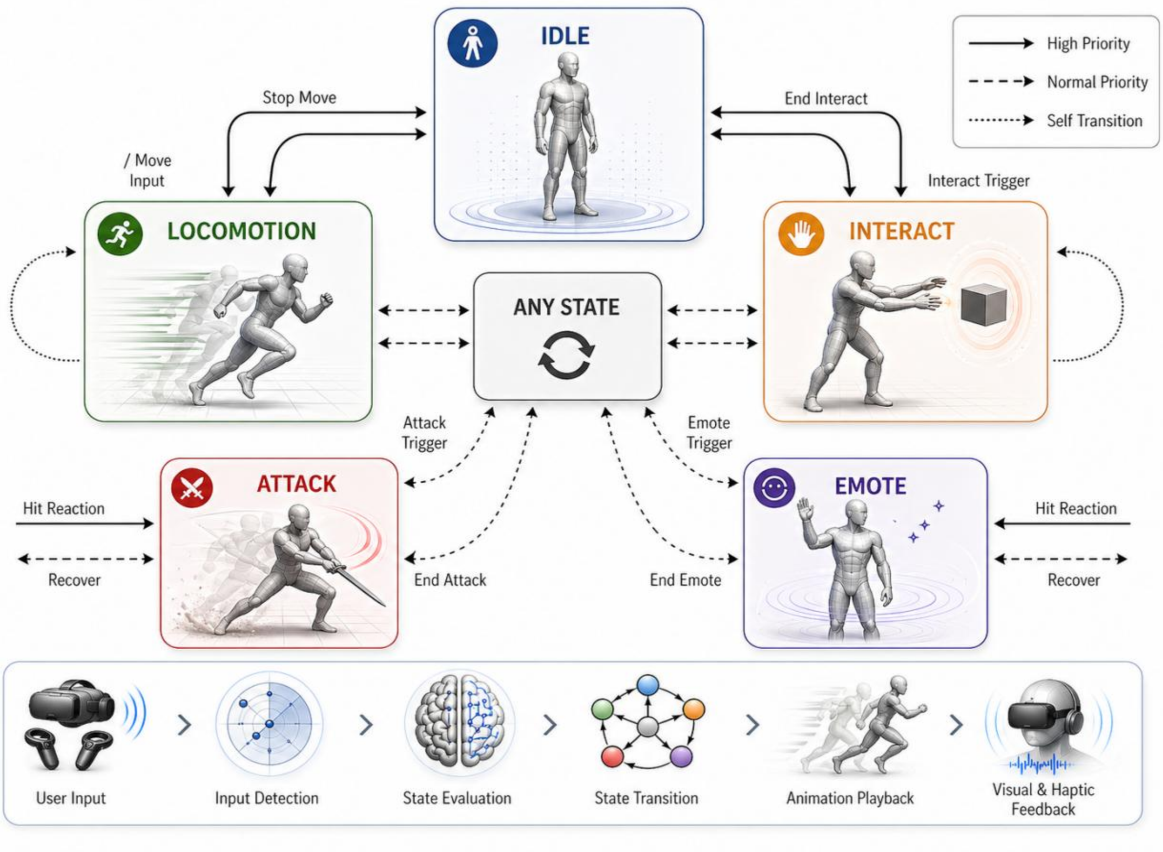

Figure 2: Animated state machine transition diagram driven by interactive input

As shown in Figure 2, the animation state machine driven by interactive Input is based on the Idle state and enters different animation states according to events such as Move Input, Interact Trigger, Attack Trigger and Emote Trigger. The system handles cross-state transitions through the Any State node and utilizes priority rules to distinguish high-priority actions, normal actions and self-cycling actions. The bottom running response link further shows that the user input needs to go through input detection, state evaluation, state transition, animation playback and visual haptic feedback, and finally forms a real-time animation response in the virtual reality environment.

In the process of playing the animation segment, the direct switching is easy to produce abrupt action, so the system adopts the weighted blending method to achieve smooth transition. If there are multiple animation segments that can participate in blending at the same time, the current pose can be expressed as follows:

$$P_t = \sum_{i=1}^{n} \omega_i (t) P_i(t) \tag{11}$$

where, P_t represents the final output character pose of frame t, $P_i(t)$ represents the pose data of the ith animation segment in this frame, $\omega_i(t)$ represents the blending weight of this segment, and n represents the number of animation segments involved in blending. The formula can support the natural cohesion of actions such as standing to walking, walking to grasping, and grasping to releasing, thereby reducing animation jumps.

Animation weights need to change continuously over time. In this paper, a linear transition function is used to describe the weight change of the animation segment during the transition time:

$$\omega_i(t) = \frac{t - t_s}{t_e - t_s} \tag{12}$$

where, t_s represents the start time of the animation transition, t_e represents the end time of the animation transition, and t represents the current rendering time. Through the weight calculation, the system can complete the motion in and out in a short time, so that the animation switch is smoother. In practice, the transition time can be set according to the type of action. The quick click action adopts a shorter transition time, and the body movement action adopts a longer transition time.

The real-time rendering engine also needs to control the temporal relationship between the animation update and the screen output. Virtual reality scenes usually require stable frame rate. If the output of animation calculation, physical detection and rendering exceeds the target frame time, it will cause lag or interaction delay. In this paper, frame time offset is used to describe the rendering performance state:

$$\Delta T_t = T_t - T_{target} \tag{13}$$

where, ΔT_t represents the frame time deviation of the TTH frame, T_t represents the actual time consumption of the current frame, and T_{target} represents the time consumption of the target frame. When ΔT_t is continuously positive, it means that the system is under high operating pressure, and it is necessary to reduce the frequency of animation update, simplify the vision model, or reduce non-critical particle effects. When ΔT_t is close to zero, it means that the animation response and rendering output are in a relatively stable state. This metric can be used to support subsequent real-time rendering performance verification.

In complex interactive scenarios, the system also needs to dynamically adjust the resource level according to the frame time deviation. In this paper, the adaptive adjustment of animation and rendering resources is expressed as follows:

$$Q_{t+1} = \begin{cases} Q_t - 1, \Delta T_t > \theta_h \\ Q_t, -\theta_l \le \Delta T_t \le \theta_h \\ Q_t + 1, \Delta T_t < -\theta_l \end{cases} \tag{14}$$

where, Q_t represents the current resource quality level, θ_h represents the high load threshold, and θ_l represents the low load threshold. When the system load was too high, the resource quality level was reduced to ensure the frame rate. When the system load is low, the resource quality is appropriately improved to enhance the screen performance. Through this dynamic control method, the virtual reality interactive animation can maintain a relatively stable operation effect under different hardware loads and user operation intensity.

In summary, the animation response generation method for real-time rendering engine is established through state machine transition, animation segment blending, transition weight control and rendering load adjustment. This method can transform user interaction input into continuous animation feedback, and control frame rate and response delay while ensuring visual performance, which provides technical support for the stable operation of virtual reality interactive animation.

4.4 Adaptive Optimization strategy for interactive animation with user behavior data

Interactive virtual reality animation continuously generates user behavior data during operation, including gaze retention, gesture triggering, path movement, misoperation, task completion time, animation response delay, and feedback rating. In this paper, these data are used as input for animation adaptive optimization, which enables the system to adjust the interaction sensitivity, animation trigger threshold, and resource loading strategy according to user operation habits. Compared with the fixed parameter control method, the behavior data-driven method can better adapt to the operation rhythm of different users, and improve the accuracy and immersion experience of animation feedback.

In this paper, the behavior state of the u-th user in the t-th time window is expressed as follows:

$$B_{u,t} = [g_{u,t}, f_{u,t}, m_{u,t}, r_{u,t}, d_{u,t}, s_{u,t}] \tag{15}$$

where, $B_{u,t}$ represents user behavior feature vector, gu,t represents fixation dwell time, $f_{u,t}$ represents interaction trigger frequency, $m_{u,t}$ represents misoperation rate, $r_{u,t}$ represents system response delay, $d_{u,t}$ represents task completion time, $s_{u,t}$ represents user experience score. The vector can unify the subjective experience and objective operation data into the same analysis space, and provide a data basis for the adjustment of animation parameters.

In the optimization process, the system needs to consider interaction accuracy, response efficiency and quality of experience simultaneously. The comprehensive optimization objective function is constructed as follows:

$$J = \eta_1 A + \eta_2 U - \eta_3 R - \eta_4 M \tag{16}$$

where, J represents the comprehensive optimization score of interactive animation, A represents the accuracy of action recognition, U represents the user experience score, R represents the average response delay, M represents the error operation rate, and η_1, η_2, η_3, and η_4 are the weight coefficients. The objective function enables the system to improve the action recognition and immersion experience while reducing the response delay and the risk of false trigger, ensuring that the optimization direction is consistent with the operation goal of interactive animation.

In the parameter update stage, the system dynamically adjusted the animation trigger threshold and resource quality level according to the user behavior feedback. Its update process can be expressed as follows:

$$\theta_{t+1} = \theta_t + \rho(J_t - J_{t-1})\nabla_\theta B_{u,t} \tag{17}$$

where θ_t represents the current animation control parameter, ρ represents the update step, $J_t - J_{t-1}$ represents the change in the comprehensive optimization score in adjacent time Windows, and $\nabla_\theta B_{u,t}$ represents the influence of behavioral characteristics on the direction of parameter adjustment. The formula is used to dynamically modify the interaction threshold, animation transition time and haptic feedback strength, so that the system can gradually approach the optimal interaction state according to the actual use process.

In the specific implementation, the system performs sliding window statistics on continuous interaction logs and marks behaviors such as abnormal clicks, long stays, and repeated grasping failures as optimization signals. When the false operation rate increased, the system increased the action trigger threshold and extended the confirmation decision time. When the response delay increases, the system reduces the quality of non-critical animation details and vision resources. When the user rating is high and the task completion time is shortened, the current parameter is kept as a stable configuration. In this way, the interactive animation can form an adaptive closed-loop of "behavior collection-feature analysis-parameter update-feedback verification", so that the virtual reality animation system can maintain good real-time performance, accuracy and experience stability under different users and different task scenarios.

5 Experimental verification and effect analysis of interactive animation design based on virtual reality technology

5.1 Experimental platform construction and interactive data collection process

In order to verify the feasibility of the proposed interactive animation design method in the real virtual reality environment, this paper builds an experimental platform including hardware acquisition layer, interactive control layer and rendering execution layer. The hardware end consists of a head-mounted display device, a two-hand controller, a spatial positioning base station and a high-performance workstation. The workstation is configured with Intel Core i7 processor, 32 GB memory and RTX 4070 graphics card. On the software side, Unity 3D engine, OpenXR interaction framework and C# script module are used to realize 3D scene loading, gesture input analysis, animation state scheduling and running data recording. The experimental tasks are set to five categories: scene browsing, target grasping, object triggering, action feedback and continuous path interaction, so as to cover typical animation states such as movement, interaction, attack and expression feedback. The data acquisition process is based on frame-level recording, synchronously sampling head pose, hand trajectory, key event, gaze stay, interaction delay and animation playback state, and alignment of multi-source data by timestamp. Finally, an experimental data set can be formed for performance analysis and experience evaluation. In order to reduce the interference of abnormal operation on the results, the interruption samples and obvious mismatch samples are eliminated in the experiment, and the trajectory data are normalized.

In order to illustrate the experimental environment and acquisition process more intuitively, the experimental platform and the interface of data acquisition are shown in Figure 3.

Time (s)	Event Type	Object	Hand	Result
12.532	Grasp	Red Cube	Right	Success
15.781	Move	Red Cube	Right	Success
18.942	Release	Red Cube	Right	Success
22.106	Select	Blue Cube	Left	Success
25.365	Press	Yellow Sphere	Left	Success
...				

Figure 3: Virtual reality interactive animation experimental scene and data acquisition interface diagram

As can be seen from Figure 3, the experimental platform can synchronously complete user motion capture, interaction state recognition, animation playback control and running log recording, providing a stable data source for subsequent real-time rendering performance verification and experience analysis. The experimental samples and evaluation index configurations are shown in Table 2.

Table 2: Experimental samples and evaluation index configuration table

Configuration Category	Specific Content	Value / Description	Technical Function
Number of Participants	Test users	36 users	Ensures the representativeness of interactive behavior samples
Gender Composition	Male / Female	20 / 16	Reduces bias caused by a single user group
Age Range	User age	21–30 years	Maintains relatively stable user operation ability
Number of Experimental Scenes	Virtual reality scenes	4 scenes	Covers animation interaction environments with different complexity levels
Number of Interaction Tasks	Typical task types	5 types	Corresponds to movement, grasping, triggering, feedback, and path interaction
Number of Valid Samples	Cleaned log records	12,480 records	Serves as the data basis for subsequent experimental analysis
Sampling Frequency	Input sampling rate	90 Hz	Ensures the continuity of motion trajectory recording
Evaluation Indicators	System performance indicators	Frame rate, response latency, GPU utilization	Verifies the real-time performance and stability of the method
Experience Indicators	Subjective evaluation indicators	Immersion, operation fluency, animation realism	Evaluates the effect of interactive animation
Statistical Method	Result analysis method	Mean, standard deviation, boxplot distribution	Supports performance comparison and result interpretation

It can be seen from Table 2 that the experiment has completed a unified configuration from four aspects of sample size, task setting, sampling frequency and evaluation index, which can not only support system performance verification, but also provide data basis for subsequent user immersion experience and analysis of animation interaction effect.

5.2 Performance verification of real-time rendering of virtual reality interactive animation

In order to verify the real-time running ability of the proposed method in virtual reality interactive animation, experiments are carried out from three aspects: frame rate stability, interactive response delay and animation resource load. The experimental scenes are set as low complexity scenes, medium complexity scenes, high complexity scenes, and high dynamic complexity scenes, corresponding to different numbers of 3D models, dynamic lights, skeletal animated characters, and interactive objects, respectively. The system runs continuously for 120 s in the same hardware environment, and the average frame rate per second, the response delay of different interactive actions, and the changes of GPU occupancy rate are recorded. In the experiment, the low complexity scene contains about 48,000 triangular faces and 6 interactive objects, the medium complexity scene contains about 126,000 triangular faces and 14 interactive objects, and the high complexity scene contains about 234,000 triangular faces and 26 interactive objects. On this basis, multi-character skeleton animation, particle feedback and dynamic shadow are added to the high dynamic complex scene. To test the stability of the system under high load conditions. Figure 4 shows the real-time rendering frame rate variation curves for different scene complexities.

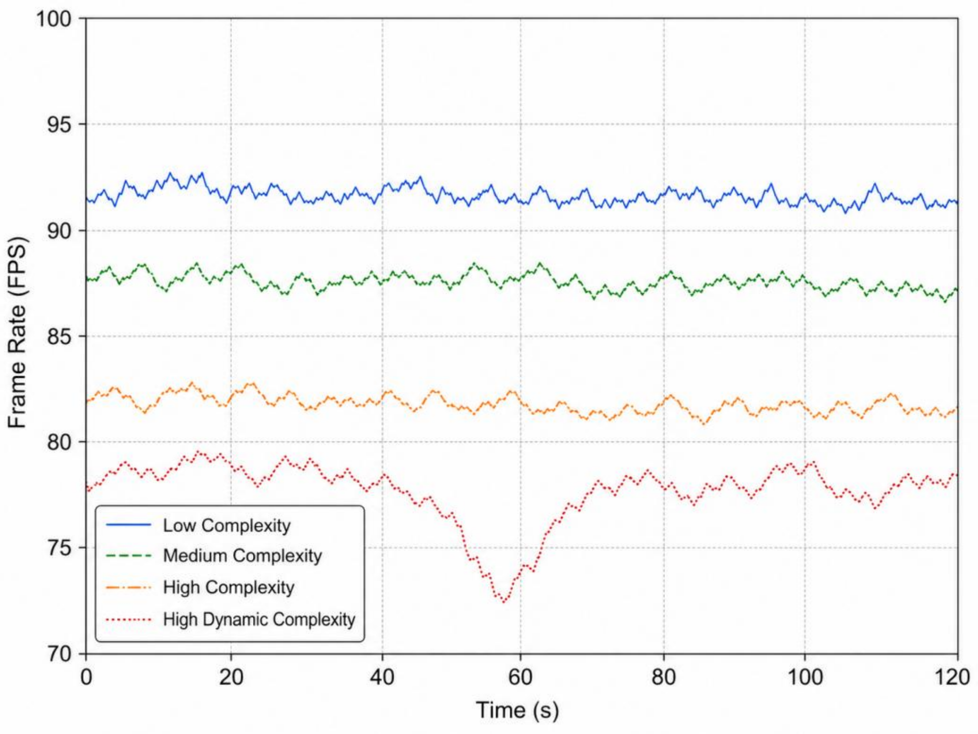

Figure 4: Real-time rendering frame rate variation curves for different scene complexities

Figure shows that with the increase of scene complexity, the overall frame rate of the system shows a downward trend, but still maintains a relatively stable operating state. The average frame rate of low complexity scene is 91.8FPS, medium complexity scene is 87.4FPS, and high complexity scene is 81.6FPS. After adding dynamic shadows, particle feedback and multi-character animation, the average frame rate of the high dynamic complex scene still reaches 76.9 FPS, and the lowest frame rate is 72.4 FPS. There is no continuity lag, which indicates that the resource compression, animation segment scheduling and rendering quality adaptive strategy can effectively control the real-time rendering pressure.

Interactive animation not only requires smooth screen, but also requires user input to be quickly converted into animation feedback. To further analyze the response stability under different action types, the response delay distributions of five types of interactive actions, such as moving, clicking, releasing, grasping and waving, are statistically analyzed, as shown in Figure 5.

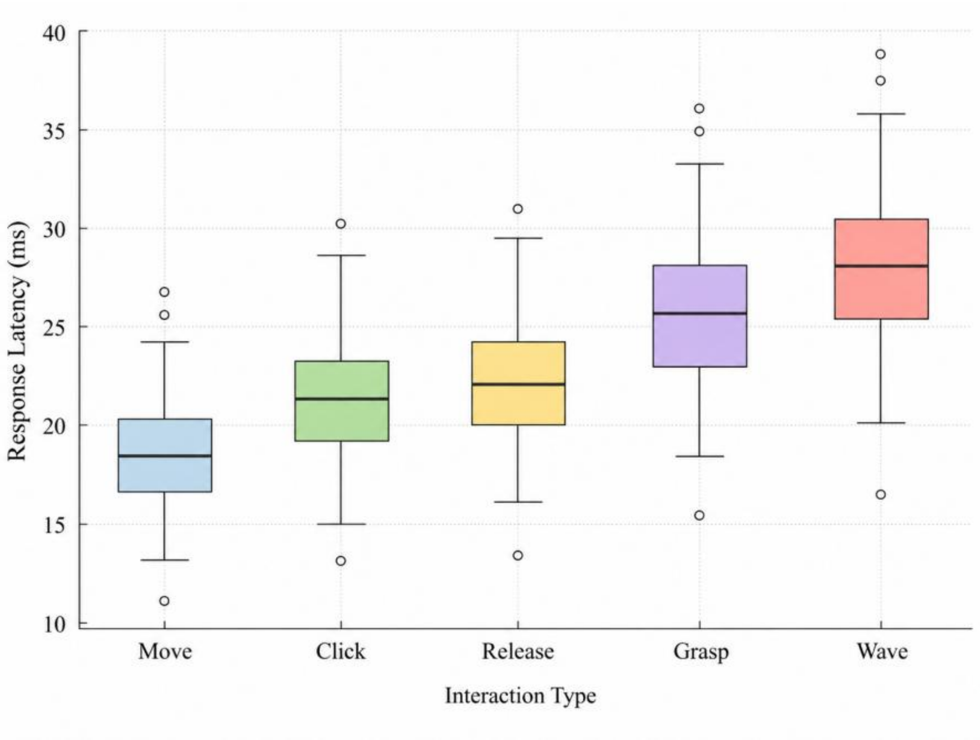

Figure 5: Box plot of response delay distribution for interactive animations

It can be seen from Figure 5 that the response delay of various interactive actions is generally concentrated in the range of 18-32 ms, among which the median delay of movement action is 18.6 ms, click action is 21.4 ms, release action is 22.1 ms, and grasp action is 25.7 ms. The waving action involves trajectory recognition and action category determination, and the median delay is relatively high. It took 28.3 ms. The upper quartile delay of all actions is less than 35 ms, and the number of abnormal high values is small, indicating that the connection between motion capture coding, state machine transformation and animation hybrid calculation is smooth, which can meet the needs of real-time interactive feedback in virtual reality environment.

In order to judge the impact of 3D resource load on graphics computing resources, this paper further analyzes the relationship between the number of model faces, tile resolution, number of bones and number of animation segments and GPU occupancy, as shown in Figure 6.

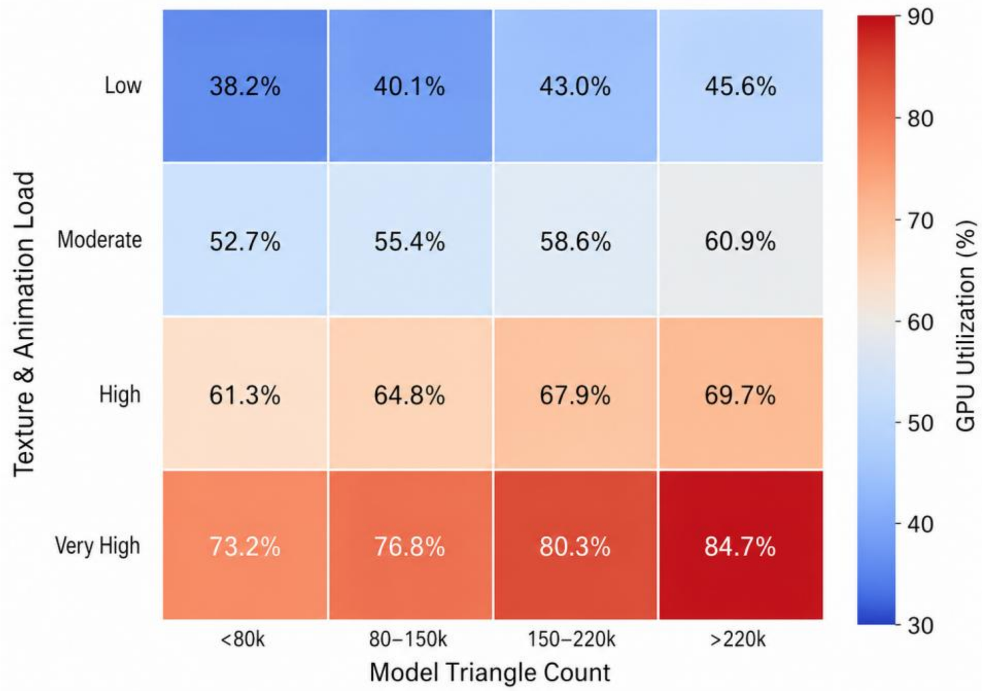

Figure 6: Heat map of animation resource load versus GPU occupancy

It can be seen from Figure 6 that the GPU occupancy rate increases significantly with the rise of resource complexity. When the number of model faces is less than 80,000 and the tile resolution is controlled within 1K, the GPU occupancy rate is mainly concentrated at 38%-46%. When the number of model faces is increased to more than 150 thousand, and the mapping and animation load is High or above, the GPU occupancy rate increases to 61%-70%. When the number of model faces exceeds 220,000 and the load of mapping and animation reaches Very High level, the GPU occupancy reaches 84.7%. After resource hierarchical loading and animation update frequency adjustment, the high-load area does not exceed 90%, indicating that the proposed method can reduce the peak pressure of rendering resources while maintaining animation performance, and provide support for the stable operation of virtual reality interactive animation.

5.3 Analysis of user immersion Experience and animation interaction effect

After the system performance verification, this paper further evaluates the effect of interactive animation design from the perspective of user subjective experience. The experiment uses a 5-point questionnaire to score the five indicators of immersion, operation fluency, animation realism, feedback timeliness and interaction naturalness, and makes a comprehensive analysis combined with the interview results after the completion of the task. To facilitate the presentation of evaluation results of different dimensions, the statistical results of user immersion experience and interaction effect are shown in Figure 7.

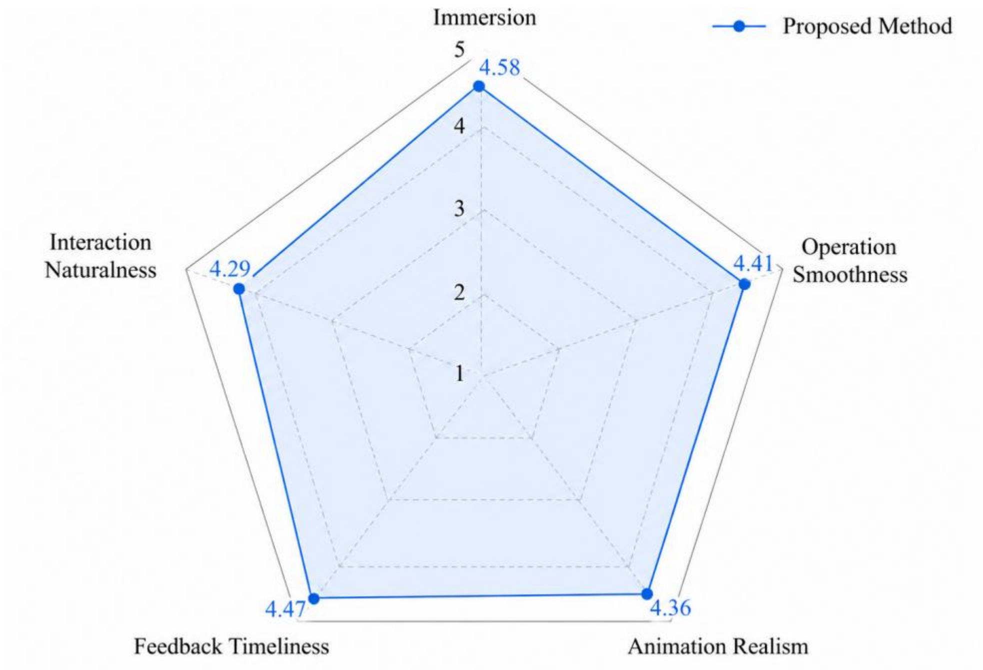

Figure 7: Radar chart of user immersion experience and interaction effect evaluation

As can be seen from Figure 7, the proposed method achieved high scores on a number of experience indicators. Among them, the average score of immersion is 4.58, operation fluency is 4.41, animation realism is 4.36, feedback timeliness is 4.47, interaction naturalness is 4.29, and the overall average score is 4.42. Most users believe that the action response of characters in virtual reality scenes is more coherent, and the matching degree between interactive feedback and their own operations is high, which can form a more obvious sense of spatial participation. Combined with the above real-time rendering results, it can be seen that the average frame rate of most scenes remains above 81 FPS, the high dynamic and complex scenes still maintain 76.9 FPS, and the median response delay of five types of interactive actions is controlled within 28.3 ms, which indicates that the user subjective experience has good stability. It shows that the interactive animation design method proposed in this paper can not only ensure the running performance, but also achieve ideal experimental results in terms of immersion experience and animation interaction effect.

6 Discussion

This paper focuses on the construction and experimental verification of interactive animation design supported by virtual reality technology. The overall results show that the hierarchical management of 3D scene resources, motion capture input coding, animation state machine control and adaptive adjustment of rendering load can better support real-time animation response in virtual reality environment. The average frame rate of low complexity scene can reach 91.8FPS, and high dynamic complex scene can still maintain 76.9FPS. It shows that resource compression, animation segment scheduling and dynamic quality control have practical effects on reducing GPU pressure. The response time of basic actions such as moving, clicking and releasing is faster, and the delay of grasping and waving is relatively increased. The main reason is that this kind of action needs to combine hand trajectory, posture change and action category judgment, and the calculation link is longer. This phenomenon also suggests that the subsequent system optimization should continue to strengthen the lightweight design of gesture recognition model.

From the user experience results, the immersion, operation fluency, animation realism, feedback timeliness and interaction naturalness have achieved high scores, indicating that the computer technology processing effect can directly affect the user's subjective judgment on the quality of animation. The experience improvement of virtual reality interactive animation is not only from the fineness of the screen, but also closely related to the accuracy of input recognition, the stability of state switching, the naturalness of animation transition and the timeliness of haptic feedback. When the user's actions can be quickly captured by the system and converted into coherent animation feedback, it is easier for the virtual character to form a real spatial response relationship with the scene objects, and the user is more likely to have a sense of participation and control.

However, the proposed method still has some room for improvement. The experimental platform is mainly based on fixed hardware environment to complete the verification, and has not covered low-performance mobile devices, cloud rendering scenes and multi-person collaborative interaction scenes. The current animation state machine can handle common actions such as moving, grabbing, clicking and feedback. However, when facing complex continuous behaviors, the state rules may still have the problem of excessive branch expansion. In the future, reinforcement learning or behavior prediction models can be introduced to automatically adjust the animation state switch according to the user's operation habits. Meanwhile, the experience evaluation in this paper is mainly based on questionnaire scores, which can reflect the subjective feelings of users, but pay insufficient attention to physiological indicators such as visual fatigue, space pressure and physical load. In the future, eye tracking, heart rate change, EEG signal or posture stability data can be further combined to build a more complete immersive experience evaluation system, so as to improve the intelligence level and engineering applicability of virtual reality interactive animation design.

7 Conclusion

Aiming at the problems of decentralized resource organization, unstable input response and delayed animation feedback in interactive animation in virtual reality environment, this paper proposes a design method of interactive animation for real-time rendering engine. Starting from the digital expression of 3D resources, this paper manages the environment, interactive objects, characters and interface resources in a hierarchical way, and realizes the character action driving through skeleton binding, skin weight and animation segment scheduling. At the interactive control level, motion capture and gesture recognition coding methods are introduced to transform head posture, hand trajectory, controller signals and collision events into animation control features, and the finite state machine is used to complete state switching such as moving, grasping, attacking and expression feedback. The experimental results show that the system maintains good frame rate stability under different scene complexity. The average frame rate of high dynamic complex scene is 76.9 FPS, and the lowest frame rate is 72.4 FPS. The median time delay of interactive actions was controlled within 28.3 ms, and the average score of user immersion experience was 4.42. In general, the proposed method can take into account animation performance, real-time rendering and user interaction experience. In the future, eye tracking, cloud rendering and behavior prediction models can be further combined to improve the intelligent adaptation ability in complex scenes.

Author's Profile

Manlu Kong was born in Huainan，Anhui Province, P.R. China, in 1988. She graduated with a master's degree in Anhui University of Technology and Science Design and art in 2014. She is now working in the Art College of Anhui Wenda Information Engineering College. Her main research direction is art design.

Funding

Anhui Province Quality Engineering Project: Serving the Top Ten Emerging Industry Characteristic Majors (Micro Majors), AI Hypermedia Technology, Project Number: 2024cywzy065

Anhui Province University Young and Middle aged Teacher Training Action Project, Excellent Young Teacher Training Project: Kong Manlu, Project Number: YQYB2025072

References

[1] Ahn D K, Bae B C, Kim Y. User Experience of a Digital Fashion Show: Exploring the Effectiveness of Interactivity in Virtual Reality[J]. Applied Sciences, 2023, 13(4): 2558. DOI: 10.3390/app13042558.

[2] Veličković P, Milovanović M. Improvement of the Interaction Model Aimed to Reduce the Negative Effects of Cybersickness in VR Rehab Applications[J]. Sensors, 2021, 21(2): 321. DOI: 10.3390/s21020321.

[3] Makled E, Gerhardt C, Schwandt T, Weidner F, Broll W. Evaluating Behavioral Realism in AR and VR: A Comparison of Single-Point IK and Full-Body Motion Capture Virtual Humans[J]. The Visual Computer, 2025, 41: 11721-11733. DOI: 10.1007/s00371-025-03934-5.

[4] Mangalam M, Oruganti S, Buckingham G, Borst C W. Enhancing Hand-Object Interactions in Virtual Reality for Precision Manual Tasks[J]. Virtual Reality, 2024, 28: 166. DOI: 10.1007/s10055-024-01055-3.

[5] Valmorisco S, Raya L, Sanchez A. Enabling Personalized VR Experiences: A Framework for Real-Time Adaptation and Recommendations in VR Environments[J]. Virtual Reality, 2024, 28: 128. DOI: 10.1007/s10055-024-01020-0.

[6] Morales-Vega J C, Raya L, Rubio-Sánchez M, Sanchez A. A Virtual Reality Data Visualization Tool for Dimensionality Reduction Methods[J]. Virtual Reality, 2024, 28: 41. DOI: 10.1007/s10055-024-00939-8.

[7] Linares-Vargas B G P, Cieza-Mostacero S E. Interactive Virtual Reality Environments and Emotions: A Systematic Review[J]. Virtual Reality, 2025, 29: 3. DOI: 10.1007/s10055-024-01049-1.

[8] Jangra S, Singh G, Mantri A, Ahmed Z, Liew T W, Ahmad F. Exploring the Impact of Virtual Reality on Museum Experiences: Visitor Immersion and Experience Consequences[J]. Virtual Reality, 2025, 29: 84. DOI: 10.1007/s10055-025-01140-1.

[9] Gkoutzios S, Vrigkas M. A Multi-User Virtual Reality Game Based on Gesture Recognition From RGB Information[J]. Computer Animation and Virtual Worlds, 2025, 36(5): e70063. DOI: 10.1002/cav.70063.

[10] Dzardanova E, Nikolakopoulou V, Kasapakis V, Vosinakis S, Xenakis I, Gavalas D. Exploring the Impact of Non-Verbal Cues on User Experience in Immersive Virtual Reality[J]. Computer Animation and Virtual Worlds, 2024, 35(1): e2224. DOI: 10.1002/cav.2224.

[11] Moutafidou A, Toulatzis V, Fudos I. Deep Fusible Skinning of Animation Sequences[J]. The Visual Computer, 2024, 40: 5695-5715. DOI: 10.1007/s00371-023-03130-3.

[12] Petrescu D, Warren P A, Montazeri Z, Strain G, Pettifer S. The Effects of Visual Behavior and Ego-Movement on Foveated Rendering Performance in Virtual Reality[J]. SN Computer Science, 2025, 6: 386. DOI: 10.1007/s42979-025-03885-7.

[13] Rogers S L, Broadbent R, Brown J, Fraser A, Speelman C P. Realistic Motion Avatars Are the Future for Social Interaction in Virtual Reality[J]. Frontiers in Virtual Reality, 2022, 2: 750729. DOI: 10.3389/frvir.2021.750729.

[14] Fraser A D, Branson I, Hollett R C, Speelman C P, Rogers S L. Expressiveness of Real-Time Motion Captured Avatars Influences Perceived Animation Realism and Perceived Quality of Social Interaction in Virtual Reality[J]. Frontiers in Virtual Reality, 2022, 3: 981400. DOI: 10.3389/frvir.2022.981400.

[15] Higgins D, Fribourg R, McDonnell R. Remotely Perceived: Investigating the Influence of Valence on Self-Perception and Social Experience for Dyadic Video-Conferencing With Personalized Avatars[J]. Frontiers in Virtual Reality, 2021, 2: 668499. DOI: 10.3389/frvir.2021.668499.

[16] Gomes de Siqueira A, Feijóo-García P G, Stuart J, Lok B. Toward Facilitating Team Formation and Communication Through Avatar Based Interaction in Desktop-Based Immersive Virtual Environments[J]. Frontiers in Virtual Reality, 2021, 2: 647801. DOI: 10.3389/frvir.2021.647801.

[17] Vindenes J, Wasson B. A Postphenomenological Framework for Studying User Experience of Immersive Virtual Reality[J]. Frontiers in Virtual Reality, 2021, 2: 656423. DOI: 10.3389/frvir.2021.656423.

[18] Fribourg R, Blanpied E, Hoyet L, Lécuyer A, Argelaguet F. Does Virtual Threat Harm VR Experience? Impact of Threat Occurrence and Repeatability on Virtual Embodiment and Threat Response[J]. Computers & Graphics, 2021, 100: 125-136. DOI: 10.1016/j.cag.2021.07.017.

[19] Sinatra A M, Pollard K A, Files B T, Oiknine A H, Ericson M, Khooshabeh P. Social Fidelity in Virtual Agents: Impacts on Presence and Learning[J]. Computers in Human Behavior, 2021, 114: 106562. DOI: 10.1016/j.chb.2020.106562.

[20] Fraser A D, Branson I, Hollett R C, Speelman C P, Rogers S L. Do Realistic Avatars Make Virtual Reality Better? Examining Human-Like Avatars for VR Social Interactions[J]. Computers in Human Behavior: Artificial Humans, 2024, 2(2): 100082.

DOI: 10.1016/j.chbah.2024.100082.

[21] Sterna R, Cybulski A, Igras-Cybulska M, Pilarczyk J, Segiet N, Kuniecki M. How Behavioral, Photographic, and Interactional Realism Influence the Sense of Co-Presence in VR: An Investigation with Psychophysiological Measurement[J]. International Journal of Human-Computer Interaction, 2024, 40(24): 8574-8589. DOI: 10.1080/10447318.2023.2285641.

[22] Kasapakis V, Dzardanova E, Vosinakis S, Agelada A. Sign Language in Immersive Virtual Reality: Design, Development, and Evaluation of a Virtual Reality Learning Environment Prototype[J]. Interactive Learning Environments, 2024, 32(10): 6657-6671. DOI: 10.1080/10494820.2023.2277746.

[23] Hartfill J, Hajahmadi S, Schmidt S, Marfia G, Steinicke F. Embracing Differences in Virtual Reality: Inclusive User-Centered Design of Bimanual Interaction Techniques[J]. Frontiers in Virtual Reality, 2025, 6: 1586875. DOI: 10.3389/frvir.2025.1586875.

[24] Wolf E, Döllinger N, Mal D, Wenninger S, Bartl A, Botsch M, Latoschik M E, Wienrich C. Does Distance Matter? Embodiment and Perception of Personalized Avatars in Relation to the Self-Observation Distance in Virtual Reality[J]. Frontiers in Virtual Reality, 2022, 3: 1031093. DOI: 10.3389/frvir.2022.1031093.

[25] Mal D, Döllinger N, Wolf E, Wenninger S, Botsch M, Wienrich C, Latoschik M E. Am I the Odd One? Exploring Congruencies and Incongruencies in the Realism of Avatars and Virtual Others in Virtual Reality[J]. Frontiers in Virtual Reality, 2024, 5: 1417066. DOI: 10.3389/frvir.2024.1417066.

[26] Landeck M, Unruh F, Lugrin J L, Latoschik M E. Object Motion Manipulation and Time Perception in Virtual Reality[J]. Frontiers in Virtual Reality, 2024, 5: 1390703. DOI: 10.3389/frvir.2024.1390703.

[27] Landeck M, Alvarez Igarzábal F, Unruh F, Habenicht H, Khoshnoud S, Wittmann M, Lugrin J L, Latoschik M E. Journey Through a Virtual Tunnel: Simulated Motion and Its Effects on the Experience of Time in Virtual Reality[J]. Frontiers in Virtual Reality, 2023, 3: 1059971. DOI: 10.3389/frvir.2022.1059971.

[28] Li K, Schmidt S, Rolff T, Bacher R, Leemans W, Steinicke F. Magic NeRF Lens: Interactive Fusion of Neural Radiance Fields for Virtual Facility Inspection[J]. Frontiers in Virtual Reality, 2024, 5: 1377245. DOI: 10.3389/frvir.2024.1377245.