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

Virtual Reality for Immersive Visualization and Customer Interaction in 3D Architectural Design

Zhongxuan Tan[1] *and Shilong Liu*[2,*]

[1] College of Media and Art Design, Guilin University of Aerospace Technology, Guilin 541004, Guangxi Zhuang Autonomous Region, China
[2] School of Electronic Engineering and Automation, Guilin University of Electronic Technology, Guilin 541004, Guangxi Zhuang Autonomous Region, China

SUMMARY: *Given the limitations of spatial expression in two-dimensional drawings and static renderings of traditional three-dimensional architectural design, leading to spatial cognitive bias and delayed feedback from customers, this paper proposes an immersive visualization and real-time interactive collaborative framework based on virtual reality. A BIM-VR (Building Information Modeling-Virtual Reality) dynamic coupling system has been built, and two-way real-time control of professional design parameters in a virtual environment has been achieved through a three-dimensional model lightweight algorithm and multi-modal natural interaction technology. Based on the model optimisation pipeline that combines the octree spatial subdivision algorithm and progressive mesh simplification technology, the triangle face conversion from the Revit model to the Unity scene has been realised. The Unity HDRP (High Definition Render Pipeline) real-time rendering engine will be used to integrate an IFC4 (Industry Foundation Classes 4) standard metadata parsing module, set up a dynamic binding mechanism for building parameters and virtual scenes, and achieve real-time linkage of key design attributes. The Leap Motion gesture recognition framework is used to build a semantic gesture library of 8 types of building operations, and a voice command parsing system based on a fine-tuned BERT (Bidirectional Encoder Representations from Transformers) model is employed to achieve natural interaction for non-professional users. Tobii eye trackers are employed to collect data on where users look, and then LSTM (Long Short-Term Memory) networks are used to determine which areas of the design have been focused on by users and suggest parameter modifications. Modify the modification instruction in the BIM Design Platform in real time using the WebSocket protocol. Based on the above experimental results, the model uses a lightweight design that reduces geometric redundancy significantly through octree spatial subdivision and progressive mesh simplification; thus, it can be loaded in just 6.7 seconds and achieve an average frame rate of 62.2fps while boosting the component recognition accuracy for non-professional users to over 0.93. This study confirms that virtual reality technology is feasible for creating an immersive collaborative environment in architectural design and offers concrete technical paths to enhance the efficiency of design communication and the scientific basis of decision-making in the customer-designer collaboration model.*

KEYWORDS: *Virtual Reality, 3D Architectural Design, Immersive Visualization, Customer Interaction, Building Information Modeling*

*liushilong@mails.guet.edu.cn
https://doi.org/10.65102/is2026937
Published: 30 April, 2026

1 Introduction

As digital architectural design is progressing towards intelligence and immersive experiences, the traditional mode of expression for architecture based on two-dimensional drawings and static renderings has increasingly failed to meet the public's demand for spatial realism and design participation. In the course of delivering architectural plans [1, 2], non-professional customers are unable to perceive the spatial relationships and thus cannot grasp the specific intent of the design; as a result, cognitive biases occur, an extended feedback loop develops, communication costs are increased, and both accuracy and efficiency of the final design are reduced. Although three-dimensional model technology [3, 4] has been widely applied to enhance visual expression, these three-dimensional models are still in a passive receiving state due to the absence of dynamic interaction [5, 6] and an immersive feedback mechanism [7, 8]. It has a single mode of operation and cannot create a sense of place. On the other hand, although the building information model [9, 10] has a complete parameter structure and component logic, it is very large in size and complex in data. In the real-time operation of the virtual environment [11, 12], there are problems such as rendering pressure and slow response, leading to a low efficiency of design adjustment and difficulty in supporting dynamic collaborative feedback. At the same time, the mode of customer interaction is still relatively limited to mouse operations and pre-set menu selection, and thus cannot provide an intuitive, natural and convenient interactive experience; this restricts the real sense of perception and feedback for non-professional users in the architectural space [13, 14].

To address the above issues, this paper proposes an immersive visualisation and customer interaction collaborative framework based on virtual reality. By constructing a multi-level system structure, optimisations can be made to the process of three-dimensional architectural Design and expanded in terms of interaction mode. The three modules of this system are: an immersive visualization layer, a data indexing mechanism based on the octree space reorganization algorithm, and compression of redundant components to improve the loading speed of three-dimensional models; a multi-level face reconstruction mechanism that utilizes grid simplification to ensure high-quality execution of the Revit-exported model in the Unity environment. Integrate the HDRP rendering engine with the IFC parameter structure parsing module to build a dynamic mapping relationship between building components and virtual components, and thus ensure that the model performs normally in multi-resolution rendering. The natural interaction layer is designed to build a multi-modal input mechanism based on the usage needs of non-professional users, creates a gesture recognition channel using Leap Motion, and trains and recognises eight typical architectural operation semantics. At the same time, a BERT semantic parsing model is used to build a voice-input channel, obtain commands in real time, perform parameter mapping, and thus connect natural language semantics with model control. A command fusion module will be used to drive parameters with gestures and voice commands. A collaborative closed-loop layer is used to implement eye-tracking and behaviour-prediction functions, and Tobii devices are employed for gaze-trajectory collection. An LSTM prediction model will be used to analyse the sequence of changes in visual hotspots to find interested customers and recommend adjustment parameters. All interactive information is synchronised to the design platform via the WebSocket channel, and a closed-loop channel has been established between gaze behaviour and design responses to parameter feedback to provide designers with data support based on user attention. Collect parameters, understand semantics, recognize behaviour, provide design feedback and synchronize with the system in a virtual reality environment for all steps; break down information barriers between virtual models and users to create a new model for immersive collaboration and efficient decision-making in architectural design. At the same time, a

full-featured technical solution has been built in terms of model structure optimisation, construction of the natural interaction mechanism, and parameter linkage architecture design.

2 Related Work

In the development of the visualization route for three-dimensional architectural design [15, 16], architectural education [17, 18] has served as one of the initial testing grounds for virtual reality technology [19, 20]. Related research continues to explore the applications of cognition in supporting learning and design experience. Hajirasouli and others [21] have built an immersive teaching framework that combines BIM and virtual reality, and it has been validated in the architectural master's design studio of a university. Based on the above results, the framework has effectively increased students' participation, improved their cooperative abilities and mastery of cutting-edge design skills under the condition of limited hardware popularisation, thus demonstrating the initial results of integrating teaching and industrial technology. Exploration of the history of architecture classes has also strengthened this trend. Ibrahim and others [22] have found that virtual reality can be used to promote the learning of knowledge and critical thinking through virtual environment experiments, Bloom's taxonomy questionnaires and critical thinking test tools, as well as satisfaction survey data. Although some problems have been identified among different users and devices, they have also promoted the development of interactive classrooms and student-centered teaching. Elbadawy and others [23] have expanded the research scope to include the application of design practice and interdisciplinary collaboration; they have used virtual reality in conjunction with system resources, such as building information models and green building semantic data platforms, to explore how these can optimize the design process, promote multi-professional collaboration, and simulate sustainable performance, thus improving spatial awareness and design efficiency at all stages of a building's life. Although previous studies have demonstrated that many scenarios have been explored to some extent, there are generally problems such as a lack of in-depth system integration and the absence of parameter coupling mechanisms for professional application [24, 25].

In the era of immersive architectural visualisation, people are starting to use it more and more frequently, and thus, how ordinary users learn to operate these systems has become a focus of human-computer interaction (HCI) research. Based on the previous study, the users have various perceptions of architectural models without professional training. Therefore, the design of the system should introduce cognitive-friendly interaction modes, use low-threshold input methods such as natural language and gestures, and guide parameter control paths with visual attention behaviour. Immersive systems need to build a closed-loop perception system that considers information accessibility, feedback consistency and operational controllability in the interaction process; thus, users can clearly express their intentions and provide accurate feedback on their design preferences within a high-degree-of-freedom virtual environment to form an interaction design logic centered on user experience.

With the development of human-computer interaction technology [26, 27], research on architectural design and customer collaboration has gradually moved towards multimodality, high precision, and intuition, and is increasingly being applied in the form of system integration and perception accuracy. Zhang and others [28] have proposed a gesture recognition system that combines computer vision with ergonomic Design and efficient recognition algorithms. Based on the above environmental tests, the system has reached a recognition accuracy of more than 97% and shown good interaction fluidity; therefore, there has been a considerable improvement in the operating efficiency and intuitiveness of BIM

models. Zhang and others [29] focus on the interactive analysis requirements of city-level scenes and build an UrbanVR system that integrates custom parallel coordinate graphs, gesture control and viewpoint optimisation algorithms. Based on the experiment results of users and experts, it has been shown that the system can help architectural decision-making systems become more intuitive and immersive by supporting high-dimensional data analysis and three-dimensional interactive experience. Bier et al. [30] have used architectural design and robot intelligent assembly technology to develop an intelligent system for producing custom-designed furniture and dynamic space-changing parts, etc. Many disciplines have been combined to expand the flexibility of human-computer interaction and, at the same time, establish a new direction for functional diversification in intelligent buildings. Although these studies have achieved some positive results in improving recognition accuracy and system response speed and scene adaptation ability, they have not systematically investigated the participation burden on non-professional users in the process of architectural visualisation, the variety of interaction modes, or the openness of feedback mechanisms. Therefore, there is still a deficiency in connecting theory and practice for improving customer experience optimisation and design collaboration [31, 32].

3 Methods

3.1 BIM Model Data Export and Lightweight Processing

This section introduces the optimisation solution for addressing load-bearing and interactive performance bottlenecks of BIM building models in virtual reality systems. Revit is used to generate data in the IFC4 standard, and at the same time, the entire structure of semantic parameters and component information is preserved. The first filter removes non-participatory temporary components and structural layers at the export stage to reduce the volume of the original model. Given the complex spatial structure of the original IFC model, an octree spatial index is constructed with building components as nodes, and the three-dimensional space is divided into local uniform voxel areas recursively; thus, non-critical areas can be quickly pruned and marked as data blocks that do not need to be rendered.

In order to achieve a light-weight requirement for real-time rendering, a weight function based on the complexity of voxel boundaries is employed:

$$w(v_i) = \alpha \cdot c(v_i) + \beta \cdot s(v_i) \tag{1}$$

Among them, $c(v_i)$ represents the internal component connectivity of voxel (v_i); $s(v_i)$ represents its surface complexity evaluation value; $\alpha, \beta \in \mathbb{R}^+$ are empirical parameter weights. This function is used to hierarchically sort the octave voxels, discard the low-complexity voxels that meet $w(v_i) < \varepsilon$, achieve targeted simplification, and avoid negative impacts on the visibility and accessibility of key interaction areas.

The topology of the original IFC component needs to be reconstructed in a mesh resource format that is compatible with the Unity HDRP rendering pipeline, and a vertex cache reordering strategy is used to optimise the order in which data is loaded by the GPU (Graphics Processing Unit). In order to achieve parallel optimisation of the data structure and the spatial structure, a component voxel density matrix is employed:

$$D_{i,j,k} = \frac{n_{ijk}}{V_{ijk}} \tag{2}$$

Among them, n_{ijk} is the number of components in the i, j, and k voxels, and V_{ijk} is the

voxel volume. The density matrix is used to sort component data by spatial aggregation, so that adjacent components can be loaded continuously in the GPU, thereby improving rendering efficiency and reducing batch processing overhead.

To help verify whether the component hierarchy and volume compression ratio at each stage before and after lightweight processing have been preserved, the model geometry conversion process structure in Figure 1 is used here to clearly show the main stages and data interface conversion path from the original IFC export to the Unity pre-processed model.

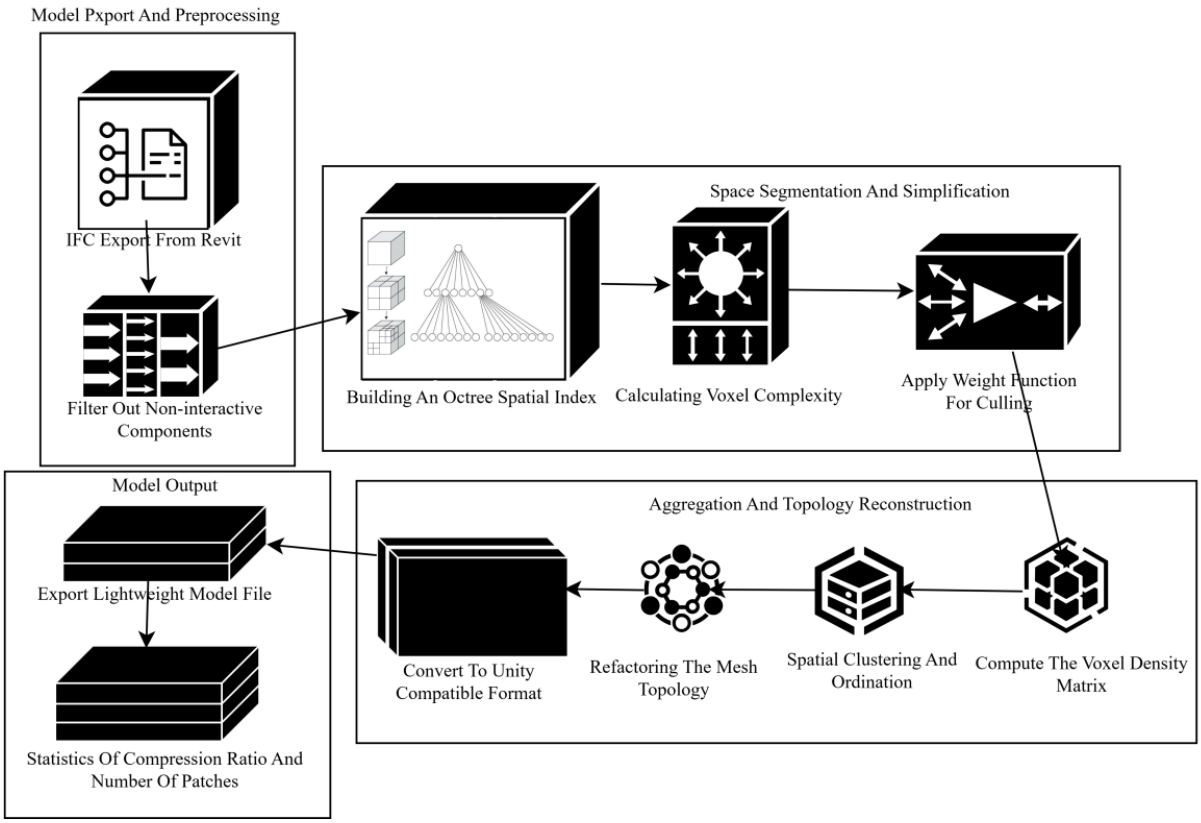

Figure 1: Model Lightweight Processing Structure Diagram

To analyse the effect of component retention strategy on spatial integrity further, the index for spatial structure retention ratio is also added:

$$R_s = \frac{V_{\text{retained}}}{V_{\text{total}}} \tag{3}$$

Among them, V_{retained} represents the visible space volume composed of the retained components, and V_{total} represents the overall volume of the original model.

3.2 Triangle Mesh Simplification and LOD Level Reconstruction

Given the problems in the Revit-exported model that appear in the Unity scene, such as a large number of triangles, high rendering load, and slow loading speed, this paper introduces a multi-resolution reconstruction method based on a progressive mesh simplification algorithm. Organise compression of the redundant geometric data to meet the real-time rendering requirements of the virtual reality platform, and at the same time maintain the integrity of essential topological structures and visual information. At the time of import, the model's mesh is divided into graph-structure representations of vertex sets and face sets, and the

objective function for mesh simplification is defined as follows:

$$\min_{\mathcal{M}'} \sum_{i=1}^{n} \| Q(v_i) \|^2 \tag{4}$$

Among them, $\mathcal{M}'$ represents the simplified target mesh. $Q(v_i)$ is the error quadratic matrix associated with the i-th vertex, which represents the estimated value of the overall geometric error after the vertex is deleted or folded. The error metric follows the Quadric Error Metrics theory of Garland-Heckbert, and local simplification is achieved by iteratively performing edge folding operations. The system automatically sets the simplification threshold according to the change of the surface normal and the perturbation of the edge length to avoid geometric damage caused by structural mutations.

To meet the performance-adaptation requirements of VR systems, the hierarchical detail-management strategy LOD (Level of Detail) is employed to build a three-level model set, and surface compression rates of 25%, 50%, and 75% of the original number are set for the near-field, mid-field, and far-field areas to cover rendering in these areas. In the loading stage, every component of a building in the scene determines its current required LOD level by calculating the Euclidean distance d and viewing angle θ to the camera, and then triggers a model switch at runtime. The function of LOD selection is as follows:

$$\mathrm{LOD}(i) = \begin{cases} L_1, \text{if } d < d_1 \wedge \theta > \theta_1 \\ L_2, \text{if } d_1 \le d < d_2 \\ L_3, \text{if } d \ge d_2 \vee \theta \le \theta_2 \end{cases} \tag{5}$$

Among them, L_1, L_2, and L_3 represent high, medium, and low levels of detail, respectively. d_1, d_2, θ_1, and θ_2 are threshold parameters set according to device performance and scene complexity. The LOD management module is bound to the Scene Graph system through C# scripts in the Unity engine to build dynamic switching logic based on observer behavior, ensuring seamless model reconstruction during the movement of the user's field of view focus, eliminating geometric flickering and loading jams.

To avoid edge jumping and texture distortion of the model during simplification, topology preservation constraints are introduced in the patch merging stage, and a texture coordinate remapping strategy is used to ensure the consistency of materials and continuity of spatial positioning before and after LOD switching.

The whole simplification and reconstruction process is finished in one go during model import and scene construction, and is decoupled from the dynamic interaction mechanism at runtime so that there is no blocking dependency between user operations and model rendering. The structure of the system for error terms is as follows:

$$E_{\text{structure}} = \sum_{i=1}^{m} \| p_i^{\text{orig}} - p_i^{\text{simp}} \|^2 \tag{6}$$

Formula (6) is used to evaluate the geometric deviation between the simplified model and the original model at the structural nodes, where p_i^{orig} and p_i^{simp} represent the spatial coordinates of the i-th key point of the original model and the simplified model, respectively.

3.3 Dynamic Binding of BIM Parameters and Unity Scenes

To achieve high-efficiency synchronous operation of BIM parameters and VR scenes, a binding mechanism has been developed at the parameter level in Unity based on the high-definition rendering pipeline. A self-developed IFC4 parsing module extracts relevant data from the exported IFC file of Revit, including materials, component sizes, node types

and building approaches. The above data are organized in a parameter metadata index tree, and thus a two-way mapping between parameters and 3D scene objects has been established. Another Indexing Strategy based on hashing has been proposed to improve query speed in runtime, thus enabling real-time parameter control and a closed-loop feedback system within the virtual environment.

At the level of data structure, the system has built a united parameter-component connection table, using component ID as the key, and it connects information about scene entities such as instance ID, position, rotation and material attributes. Dynamically reflect to safely bind the IFC parameter field to a Unity Material and transform property. For successive change quantities, such as thickness and height, they are handled by value mapping, and for separated structure types, they are controlled by state-based logic switching. To reduce the performance undulations caused by frequent parameter updates, a Bayesian smoothing-grounded state renewal mechanism has been added to stabilize parameter transmission and scene response in the system.

$$P(\theta_t|\theta_{t-1}) = \frac{P(\theta_{t-1}|\theta_t)\cdot P(\theta_t)}{P(\theta_{t-1})} \tag{7}$$

Among them, θ_t represents the parameter state at time t, and the state transition is dynamically inferred based on the state at the previous moment, which effectively alleviates the visual discontinuity problem caused by parameter mutation.

The system connects BIM parameters and Unity objects in a two-way, event-driven way while running. The component interaction trigger parses the user input and then calls the binding relationship in the mapping table. Modify the BIM parameters in real time, and the corresponding properties in the Unity scene will be updated automatically. To address the problem of synchronization delay under high-frequency interaction, an asynchronous cache queue is introduced to buffer parameter change instructions in the queue, and the synchronization batch is controlled by the change aggregation operation within a time window to improve performance stability.

To support multi-scenario collaboration and scalability maintenance, the system is designed with a modular parameter binding logic, and the parser, binder, and synchronizer have been decoupled for deployment. The IFC parser independently extracts parameters and builds the structure; then, the parameter binder connects to the corresponding three-dimensional component entity by registering, and observes any changes in its attributes via the parameter synchroniser to perform synchronization. Events are distributed among the three via the publish-subscribe mode. Data interfaces between modules are in a unified format and follow the following structure definition:

$$B_i = \{ID_i, \vec{p}_i, R_i, M_i, \theta_i\} \tag{8}$$

Among them, B_i represents the i-th bound component; ID_i is the unique identification number of the scene object; $\vec{p}_i \in \mathbb{R}^3$ is the spatial position vector; $R_i \in \mathbb{R}^{3\times3}$ is the rotation matrix; M_i is the material parameter set; θ_i is the IFC attribute parameter set. All component parameters are encapsulated in this structure and passed to the rendering pipeline and interactive logic module.

To improve the response speed and accuracy of parameter control, a differential-trigger mechanism for the system is employed to compare the difference in the parameter vector between the current and previous states:

$$\Delta\theta = \| \theta_t - \theta_{t-1} \|_2 \tag{9}$$

When $\Delta\theta$ exceeds the set threshold, the state update instruction is triggered to implement the minimum necessary synchronization strategy, effectively controlling the resource load caused by parameter linkage in large-scale building models.

3.4 Multimodal Natural Interaction System Design

The construction of the multimodal natural interaction system is based on unstructured input parsing of building parameters and mapping of structured instructions to enable free operation of building models in a virtual environment by users. First, a high-frame-rate infrared capture interface based on the Leap Motion Controller is employed to obtain 25-degree-of-freedom skeletal point data of the user's hand, and a gesture dynamic recognition channel is built by modeling the temporal continuity of the trajectory sequence in three-dimensional space. A Dynamic Time Warping (DTW) algorithm is employed to match the morphological characteristics of a skeletal point sequence, and the classification task for eight types of building operation gestures is realised end-to-end by a deep residual network; at the same time, the semantic labels are output and bound to the BIM parameter controller. Given that the structural dependence of the architectural semantic operation is known, the gesture vector sequence can be defined as follows:

$$H = \{h_t \mid t = 1,2,\ldots,T\}, h_t \in \mathbb{R}^{3N} \tag{10}$$

Among them, h_t represents the three-dimensional coordinates of N skeleton points at time t. T is the time series length, which constitutes the model input space. The multi-scale convolution unit is used to extract local features to enhance the recognition robustness.

The voice interaction module builds a natural language command understanding system based on the Transformer structure, and uses the BERT pre-trained model to fine-tune the building operation corpus to construct a specific semantic space for the building scene. The voice signal is collected in real time through the WebRTC interface and transcribed into a text stream. After BERT encoding, the semantic vector representation $S = f_{\text{BERT}}(x)$ is extracted, where x is the voice transcribed text. In order to achieve accurate mapping of instructions and BIM parameters, a semantic parser based on the conditional instruction graph is constructed, and the parsing function is defined as:

$$P(y \mid x, C) = \text{softmax}(W^{\top} \cdot \text{ReLU}(U \cdot S + V \cdot C + b)) \tag{11}$$

Among them, C is the current interaction context state vector; y is the instruction category; $W^{\top}$, U, V, and b are the training parameter matrices.

In order to implement the multiple-modality cooperative scheduling mechanism, a single command arbitration module has been built in the system to receive intermediate semantic results from the gesture and voice channels, and then performs behaviour fusion according to priority rules and context conflict resolution algorithms. A confidence weighting mechanism is used to obtain the final control command output, and the fusion function is as follows:

$$\hat{y} = \arg\max_{y}(\alpha \cdot p_{\text{gesture}}(y) + \beta \cdot p_{\text{speech}}(y)), \alpha + \beta = 1 \tag{12}$$

Among them, $p_{\text{gesture}}(y)$ and $p_{\text{speech}}(y)$ are the confidence estimates of the gesture and voice for the command y, respectively. α and β are dynamically adjusted channel weights, which are adaptively updated according to the user's interaction habits.

Figure 2 shows the functional diagram of the multi-mode natural interaction system. Its structure has a two-channel input of "gesture recognition + speech instruction". The modules

of this system are gesture acquisition and recognition, speech character recording and meaning analysis, middle semantic expression, command judgment combination, and parameter control transformation. Gesture passage collects skeleton data using Leap Motion and achieves operation semantic identification by integrating with a residual network; at the same time, a BERT model is employed to process voice channel construction task commands. The two kinds of instructions are uniformly placed in the instruction combination module, and the last control signal is generated by the confidence weight mechanism; this signal is then synchronized with the parameter control level to achieve a feedback loop for user actions and model conditions.

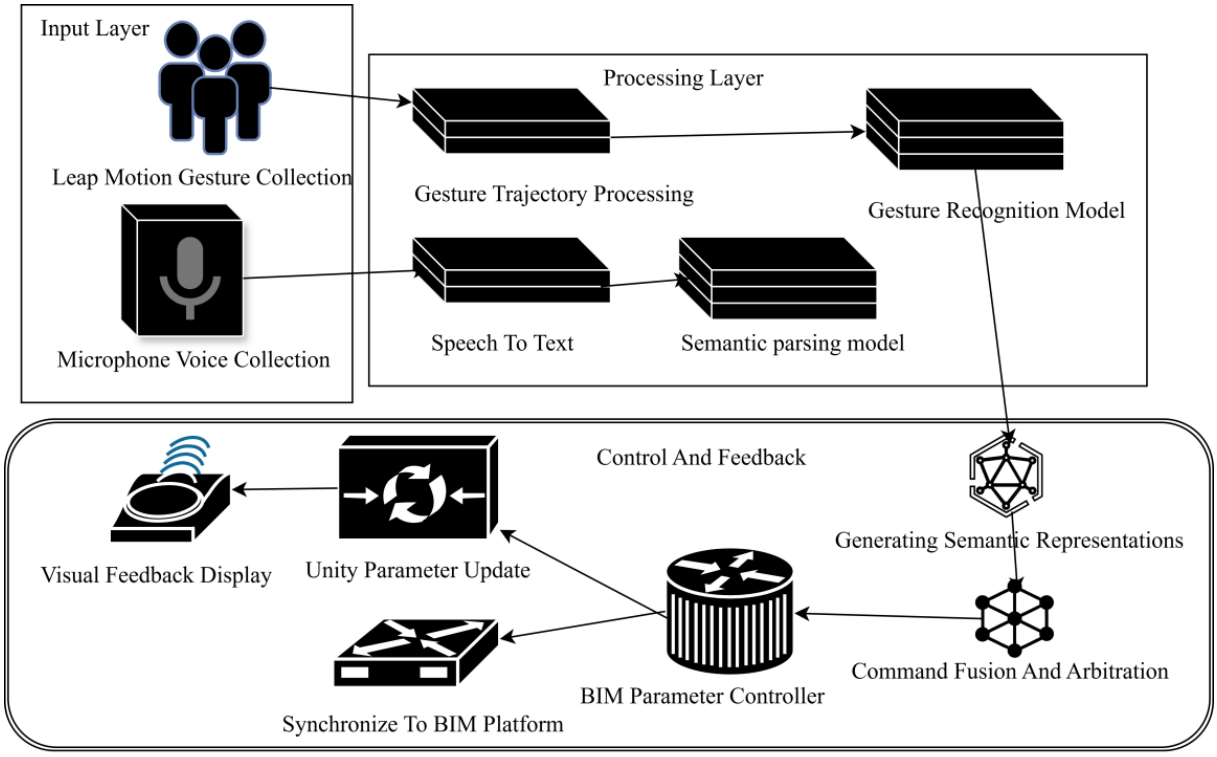

Figure 2: Architecture of Multimodal Natural Interaction System

The construction of the semantic gesture storehouse is based on the function distribution of the meanings of architecture operation movements, and eight typical behaviours have been included: wall making, window/door cutting open, material changing, model turning, part hiding, magnification controlling, view wandering, and undo/redo doing. In the training stage, the system has collected 3,200 multi-user dynamic gesture sequences and thus optimised the recognition network weights by using a weighted cross-entropy loss function. To improve the recognition ability of the model for occluded and overlapping actions, a space shelter enhancement method and a time disturbance regularization term have been added to the training process to suppress high-frequency wrong-detection noise.

Train a speech model using the Chinese building instruction dataset to reduce the average classification error via an objective function. In the fine-tuning stage, the low-level parameters of BERT are frozen, and only the top-level Transformer and classification head parameters are updated. The dimension of the output vector is the index table for the building parameter dictionary, and it directly corresponds to the model control variables. Finally, the system sends the recognition results and control commands to the BIM platform via a WebSocket channel for real-time command closed-loop control of multi-modal input.

3.5 Visual Attention Prediction and Design Parameter Control Suggestion Generation

For the purpose of promoting the active perception capability and the pertinence of design feedback in virtual reality architectural design interaction, the present study brings the Tobii eye tracker into the user experience course to gather high-frequency data about customer gaze behavior. Through the recording of visual behavior indexes which include gaze point, fixation duration, and saccade path, a hot zone distribution matrix $H(x, y, t)$ is constructed. This matrix describes the evolution trend of the user's gaze weight at different positions in the three-dimensional scene in the form of a time series. For this multi-dimensional nonlinear sequence, a two-layer LSTM is deployed. Its input is the gaze vector sequence $\mathbf{G} = \{g_1, g_2, \ldots, g_T\}$ within the time step T. Each vector $g_t \in \mathbb{R}^3$ consists of the three-dimensional coordinates of the gaze point and the dwell time. The network output is the probability distribution $\widehat{H}_{t+1}(x, y)$ of the spatial attention hot zone at the next time step t+1. In order to improve the model's ability to capture the trend of changes in the attention trajectory, a temporal attention module with a gating mechanism is introduced. The weight calculation formula is:

$$\alpha_t = \frac{\exp(\mathbf{w}^\top \tanh(\mathbf{W}_h \mathbf{h}_t + \mathbf{W}_g \mathbf{g}_t))}{\sum_{k=1}^{T} \exp(\mathbf{w}^\top \tanh(\mathbf{W}_h \mathbf{h}_k + \mathbf{W}_g \mathbf{g}_k))} \tag{13}$$

Among them, $\mathbf{h}_t$ is the LSTM hidden state; $\mathbf{g}_t$ is the gaze input of the current step; α_t is the attention coefficient of this time step.

Based on the prediction of the focus area, the control suggestions of the design parameters are generated through a multi-objective evaluation mechanism. The system maps the three-dimensional spatial position of the predicted hot zone coverage with the IFC4 building component metadata to construct a hot zone-component parameter mapping tensor $\mathcal{P}_{ijk}$, where index i corresponds to the hot zone number; j is the component ID; k is the parameter attribute item. The parameter suggestion generation process is implemented in parallel through clustered frequency back-analysis and gradient attribution model. The former counts the most frequently modified parameter items in the components corresponding to high-frequency gaze hotspots, and the latter constructs an influence map between gaze behavior and parameter regulation results. The contribution of component parameter p_k to the target hot zone prediction response function f(H) is taken as the core indicator. The first-order Taylor expansion is introduced to approximate its local sensitivity, forming the following attribution expression:

$$\Delta f(H) \approx \sum_{k=1}^{K} \frac{\partial f}{\partial p_k} \Delta p_k \tag{14}$$

Among them, $\Delta f(H)$ represents the change amplitude of the hot zone response; $\frac{\partial f}{\partial p_k}$ is the gradient of the function to the k-th parameter, reflecting its regulation sensitivity; Δp_k is the candidate adjustment amplitude of the parameter. The system selects the parameter corresponding to the maximum attribution degree for recommendation generation, and attaches the impact factor score to assist users in decision-making.

To enhance the practical value of the parameter recommendation and the generalisation ability of the system, a constrained optimisation mechanism is used to select candidate parameter combinations and an objective function is set as follows:

$$\min_{\mathbf{p}} \mathcal{L}(\mathbf{p}) = \lambda_1 \parallel \mathbf{p} - \mathbf{p}_0 \parallel^2 + \lambda_2 \mathcal{C}(\mathbf{p}) \tag{15}$$

Among them, $\mathbf{p}_0$ is the current component parameter vector; $\mathcal{C}(\mathbf{p})$ represents the compliance constraint of the parameter combination to the building code; λ_1 and λ_2 are the control weight coefficients to ensure that the modification plan recommended by the system takes into account both the original design intention and compliance.

Real-time handling of visual attention behaviour is packed as a running module in the Unity plug-in and placed within the interactive main thread loop. The update frequency of the forecast module will be 10 times per second or more to promptly reflect any changes in the user's gaze behaviour and avoid any interruption of the interactive experience due to a delay in feedback. All the generated parameter proposals are displayed in real time by a graphic user interface, and then they need to be manually approved and pushed to the BIM platform by the designer. To maintain consistency of data and establish a feedback mechanism for all parts of the system, instructions can be formatted as structured JSON messages over the WebSocket channel, and in this message, include component ID, parameter name, modification scope, and confidence degree. After obtaining the above information in the design platform, this information will be immediately added to the Revit model by the platform, and the corresponding rendering state inside the virtual scene will be updated; thus, a parameter-level closed-loop control mechanism can be achieved between VR and the design platform.

3.6 Construction of Real-time Synchronization Mechanism for Interactive Feedback

In the process of building a real-time synchronization mechanism for interactive feedback, a full-duplex communication agreement based on WebSocket is employed to achieve low-delay two-way data synchronization of user operations in the virtual reality environment and the BIM design platform. All events during the operation of parameters in the virtual scene are monitored in real time by the monitoring module and then structured. A custom unified data protocol header is used to identify the type of operation and a corresponding parameter ID for high-precision synchronization mapping at the parameter level. To reduce the communication load in concurrent operations, an asynchronous message queue management mechanism and a timestamp sequence control tactic have been adopted; thus, data consistency and timing reliability under multi-user cooperation can still be guaranteed.

Aiming at the high-dimensional parameter structure of the BIM design platform, a parameter state hash table is constructed to map the state difference of the parameter modification involved in each virtual interaction, and the sparse matrix compression technology is used to optimize the transmission data volume. Each interaction operation is recorded in the form of a four-tuple as $O_i = (p_i, v_i, t_i, u_i)$, where p_i is the parameter ID; v_i is the modification value; t_i is the operation timestamp; u_i is the user ID. During the transmission process, this structure is transmitted via the binary format encoding embedded in the WebSocket data frame. After receiving it, the server will synchronously map it to the BIM model parameter space through the parameter decoupling module, and automatically trigger the change instruction in the Autodesk Revit API to form a closed-loop feedback.

A delay estimation function based on an exponential moving average has been added to increase the robustness and real-time performance of instruction synchronisation in the system.

$$\hat{d}_t = \alpha \cdot d_t + (1 - \alpha) \cdot \hat{d}_{t-1} \tag{16}$$

Among them, $\hat{d}_t$ represents the current estimated communication delay; d_t is the actual observed delay; $\alpha \in (0,1)$ is the smoothing factor, which is set to 0.35 in the experiment.

After generating the operation instruction on the client side, the frame listener caches it in the local operation stack and triggers the WebSocket communication thread. Bandwidth-adaptive sliding window protocol is used to regulate the frame rate and dynamically set an upper bound. At the time of high user activity, for the sake of rendering stability, the instruction frame rate will be restricted to 25fps. The server uses a thread pool to process the parameter update request concurrently, builds the dependency graph topology of the parameters, and decides whether to trigger cascade updates based on the coupling relationship among parameters to prevent data rollback due to invalid synchronization operations.

For multi-user collaborative interaction environment, the system builds a user role permission matrix to limit the scope of modification of model parameters by different users, and embeds it in the synchronization instruction structure through the role field $r \in R$. The server performs access verification according to the permission graph, rejects unauthorized operations and records logs. In order to support operation backtracking and parameter status comparison, a state vector snapshot mechanism is introduced, and the operation sequence window length is set to n. After each n parameter updates, a vector representation $S_t = [v_1, v_2, \ldots, v_k]$ of the current parameter state is automatically generated. The system maintains a sliding snapshot history queue for difference detection and error recovery.

The rate of parameter modification is not directly related to the demand for real-time synchronisation. According to the above analysis, it has the following form:

$$\lambda_s = \beta \cdot \ln(f_u + 1) \tag{17}$$

Among them, λ_s is the synchronization load per unit time; f_u is the average user interaction frequency; β is the platform load sensitivity constant. To further improve system stability and load elasticity, Nginx reverse proxy and Redis memory cache strategy are deployed on the server side to cache the latest interaction parameter status and support breakpoint retransmission and delay recovery mechanism.

4 Experiment

4.1 Experimental Environment and Hardware Configuration

The experimental platform is a high-performance computing facility that supports multi-mode feel technology, model loading, and real-time mutual operation and precise vision tracking in VR. The working station of this system has an Intel Core i9 CPU, an NVIDIA RTX 4080 GPU, 128 GB of DDR5 memory and a Windows 11 Pro operating system. Revit 2023 is used to create the BIM model and handle data; an IFC4 output file is generated, and in Unity 2022.3 LTS, the HDRP pipeline builds the virtual surroundings for high-fidelity picture drawing.

Immersion-type mutual operation is available for the Varjo XR-3 helmet, Leap Motion hand position tracking, a row microphone for sound input, and Tobii Pro Fusion eye movement tracking at a frequency of 120 Hz. One language module based on BERT will be used to carry out command explanation and parameter mapping. To keep the VR client and the BIM platform in synchronisation at any time, this system uses asynchronous multi-process communication with WebSockets to establish a closed-loop channel for parameter updates and status feedback.

The experimental hardware and system composition are shown in Table 1, and the corresponding configuration of each module and the tasks performed are as follows:

Table 1: Hardware and Software Configuration of the System Experimental Platform

System Module	Hardware/Software Version	Key Specifications	Assigned Function
Computing Platform	Intel Core i9 + RTX 4080	128GB RAM, 2TB SSD	Model Computation And Real-time Rendering
Modeling And Parameter Processing	Revit 2023 + IFC4 Standard	Support for 200+ Parameter Entities	BIM Data Export And Structural Mapping
Virtual Reality Development Engine	Unity 2022.3 HDRP	Integrated IFC4 Parsing Module	Scene Construction And Real-time Visual Rendering
Immersive Display And Interaction Devices	Varjo XR-3 + Leap Motion	2880x2720eye, Six Degrees Of Freedom Hand Tracking	Immersive Display And Gesture Interaction Parsing
User Attention Tracking System	Tobii Pro Fusion	120Hz Binocular Tracking	Visual Focus Capture And Hotspot Analysis

Table 1 is a summary of the hardware and software layout of this system, including the calculation platform, model construction tools, development engine, mutual-action devices, and perception modules. Together, these parts of the Constitution affect the stability of the system, the loading efficiency of the 3D model, and the quality of interaction. During the period of experimentation, various modules are employed to help build the model for calculating work, analysing data related to work, drawing vision pictures, capturing movement, and recording attention levels. Together, they are able to respond to changes in real time and have a large working capacity for this system. The selected configuration will offer a stable operating environment and reliable data support for a high-demand building model and multi-modal interaction.

4.2 Dataset Preparation and Model Selection

Five real-life design project data from different construction categories have been selected as the origin of the experiment data groups to ensure that the system can work in all circumstances, covering dwelling houses, public constructions, business compounds, factory workshops, and multi-floor office buildings. All of the above construction projects used Autodesk Revit for the completion of the early-stage model. The model contains many professional components, such as architecture, structure, electromechanical equipment, etc., and is exported in the IFC4 standard format for subsequent use. To enhance the interaction efficiency of BIM models in a virtual reality environment, all models are now preprocessed uniformly to remove redundant components and standardise material name rules for consistent parameter parsing and binding.

The proportion of the built model directly affects the operating effect of the virtual reality mutual-action system. To conduct quantification of model complexity and composition structure, the total number of components, the size of the IFC file, and the initial quantity of triangles in each project are all counted. Table 2 lists the data composition parameters for the

five representative building models. Among them, the residential structure has a relatively low component density. The industry factory has a relatively simple structure but is a large covering area and has many distribution points. Commercial composite buildings and multi-story office buildings have a large number of elements and are complex in terms of IFC file size and the number of triangular faces. Public building models have many functions and complex structures; therefore, they are generally large in scale.

Table 2: Statistics of Parameters for Different Construction Project Datasets

Project Type	Number of Components	IFC File Size (MB)	Initial Triangle Count (Ten Thousand)
Residential	798	42.3	65.4
Public Building	2,384	138.7	174.6
Commercial Complex	2,137	129.5	161.2
Industrial Plant	1,425	93.6	118.8
Multi-Story Office	1,746	101.9	126.5

The model of interactive behaviour in this gesture-recognition system for non-professional users' natural hand movements in virtual environments is based on Leap Motion Controllers. Gesture data sets are obtained by using three-dimensional tracking data of 20 volunteers performing a set of building operation tasks in an immersive environment. All the participants finished 8 kinds of basic construction interaction tasks, and every kind of gesture was recorded ten times consecutively. One thousand six hundred high-accuracy moving path samples were collected by us to build a multi-gesture training set. The speech recognition module collects voice instruction audio data in the same task, unifies specialised word groups and operation parameter noun collections, and carries out BERT model fine-tuning training work to improve semantic comprehension ability.

A Tobii Pro eye tracker was used to acquire the gaze trajectories of the experiment subjects in all virtual environments for a visual attention prediction model. Convert the original gaze data into a two-dimensional attention area map through spatial thermal mapping, record the current model parameter status, and construct a time series sample pair for training the LSTM prediction network. To improve the generalisation ability of the model for different types of buildings, the training data set has been evenly divided based on building type and user ID to ensure that the visual hotspot prediction results are representative.

In terms of the model for interactive behaviour, a gesture recognition system is used to collect natural hand movements by non-professional users in a virtual environment through a Leap Motion Controller. A gesture data set has been generated by collecting the 3D tracking data of 20 volunteers performing fixed building operation tasks in a virtual environment. All participants have completed the eight categories of basic construction interaction activities, and each gesture has been recorded ten times. One thousand six hundred high-accuracy moving path samples were collected by us to build a multi-gesture training set. The speech recognition module collects voice instruction audio data in the same task, unites specialised word groups and operation parameter noun collections, and performs BERT model fine-tuning training to improve semantic understanding ability.

4.3 Experimental Process and Interactive Verification Method

After the initial Revit model is complete, all parts of the BIM model, including structure, MEP and inside sub-systems, are exported as an IFC4 file. Redundant geometry is reduced by

spatial segmentation based on an octree, and thus progressive mesh simplification and LOD reconstruction from LOD0 to LOD3 can be performed to meet the HDRP real-time rendering requirements in Unity. A light-weight model is used in Unity, and a self-made IFC parser is built to map BIM parameters to scene objects dynamically updateable. To ensure a relatively smooth picture and reduce the visual impact of high-frequency changes, parameter synchronous threshold values and a cache-queue mechanism have been added.

In the test of the interactive function, a Leap Motion device is placed over the area where users make gestures, and eight kinds of pre-recorded building operation instructions can be recognised by the embedded gesture recognition module. At the time of testing, the recognition response speed and accuracy of various users operating different gestures in multiple building scenes are recorded, matched with and verified against a standard gesture database. The voice interaction module is connected to the debugged Chinese BERT fine-tuning model, receives natural language instructions from users, and outputs corresponding parameter control instructions; then, it calls the parameter update module in Unity to achieve a state change. A set of preset voices will be used for several rounds of command parsing verification in the voice system test phase. Each set of commands should be divided into three steps: intent recognition, parameter mapping and instruction execution, and their recognition success rates and system response delays under various noise conditions should all be recorded.

In the process of collecting eye-tracking data, the Tobii Pro system continuously records where a user looks and for how long inside a virtual scene, as well as the weight of fixation at these locations. The above data are converted into a spatial attention matrix by means of hotspot mapping and then used as input for an LSTM model to predict future attention distribution. In the forecasting stage, the output of the model is compared with the actual gaze overlap and dynamically drawn; therefore, it can be known how users' attention is shifting and corresponding adjustment suggestions offered by designers. A connection between visual heat points and recommendation system parameters is constructed by matching based on distance and historical interaction data, and thus a recommendation threshold controls when and where parameter reminder information is displayed visually.

The interactive operations at any time in the whole process are recorded by the Unity scene control script. All user-input data, system-response information, and visual trajectory data are synchronously transmitted back to the BIM platform via the local WebSocket channel to ensure the state consistency of the interactive operation and the original model data. The interactive command call log, response delay and synchronisation delay will be written to a standard test log in the experimental recording module for subsequent analysis of the stability and accuracy of all interactive subsystems under complex building model conditions.

5 Result Analysis

5.1 Model Loading Time and Running Frame Rate Evaluation

Five typical models (Model A-E) were chosen as the performance analysis subjects for a systematic evaluation of the loading efficiency and real-time rendering performance of architectural 3D models in a virtual reality environment, including residential buildings, public buildings, commercial complexes, industrial plants and multi-story offices. The models have different spatial hierarchies, various degrees of component complexity and geometric detail accuracy, and different scales of BIM data and rendering loads for different types of design. All models were exported from Revit in the IFC4 standard format, and two sets of parallel processing paths were established: one was a lightweight strategy based on octree

spatial stripping and patch reconstruction to enhance the speed of geometric data loading and reduce GPU load; the other was a multi-resolution LOD mechanism that divided each model into five levels from LOD0 to LOD4, corresponding to the rendering strategy levels of high-fidelity to low-fidelity, and dynamically adjusted the running performance. To quantify the performance of the system under different models and optimisation strategies, the following three indices were set: model loading time to indicate the system's response speed from import to visualization; average frame rate to show the graphic smoothness of the model in a virtual reality scene; and patch compression ratio to assess the geometric data compression capability of lightweight processing. Figure 3(a) shows the loading time and frame rate differences of various models before and after optimisation, and a dual-axis chart is used to display the change in response time and rendering performance simultaneously; Figure 3(b) shows the number of original and lightweight facets, as well as the compression ratio, and demonstrates the compression effect of different models after structural redundancy stripping; Figure 3(c) compares the average frame rate changes of all models at five levels of LOD to show how well the multi-resolution strategy can adjust rendering efficiency.

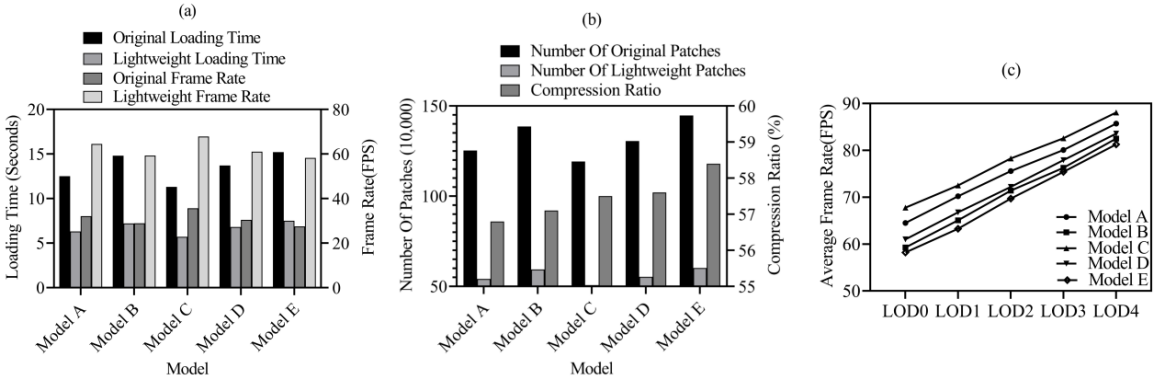

Figure 3(a) Comparison of Model Loading Time and Average Frame Rate
Figure 3(b): Number of Triangle Faces and Compression Ratio
Figure 3(c) Average frame rate at different LOD levels.
Figure 3: Analysis of Loading Performance, Facet Compression Ratio and Running Frame Rate for Different Building Models Under Lightweight and Multi-level LOD Strategies

Lightweighting is for reducing both model size and running cost. The initial import speed of the original model was 13.5 seconds, and after optimisations for a reduced weight, it has now fallen to 6.7 seconds. The loading time of Model E has been reduced from 15.2 seconds to 7.5 seconds, representing the largest decrease, and it has a high complexity and redundant data volume in its original form. In terms of frame rate performance, the running frame rate of the unprocessed model is generally less than 35.6 fps; on the other hand, after lightweighting, the average frame rate reaches 62.2 fps, and both Model D and Model E have increased from 30.4 fps and 27.5 fps to 61 fps and 58.2 fps, respectively; thus, face compression has shown a more substantial improvement in reducing the rendering pressure of geometrically dense models. The middle sub-figure shows that the average number of faces of the original model is 1,316,600, which is reduced to 559,400 after lightweighting, with an average compression ratio of 57.5%; Model E reaches a compression ratio of 58.4%, and thus it can be inferred that its industrial plant structure has a higher simplification potential for the stripping algorithm. The right sub-graph shows that the hierarchical impact of the LOD mechanism on operating efficiency is as follows. LOD0 has a mean of 62.2 frames per second, and it increases to as high as 84.2 fps at LOD4. The frame rate of Model E increases by 23.1 fps, and thus its

multi-layer office structure is more feasible for LOD level switching. A light-weight model and an LOD mechanism can be used to improve the loading speed and real-time display of the architectural VR system in multi-dimensional cooperation.

5.2 Parameter Interaction Response Speed Evaluation

To quantify the system's response efficiency for different types of natural interaction in typical architectural scenarios, five representative user interaction tasks have been selected as test cases; through a standardised experimental process, the average response time of gesture and voice interaction has been collected, and their effects on the real-time performance of virtual reality architectural visualisation systems have been investigated. Task types are divided according to the complexity of the operation and different system feedback chains. Perspective switching tasks correspond to moving the virtual camera in the scene, and these mainly depend on the view reconstruction efficiency of the rendering engine; they are low-latency operations. Material replacement involves loading and applying model material resources. The processing process needs to parse the corresponding component information and refresh the rendering resources, so it has a medium response time; size adjustment and parameter panel operation involve parsing, calling and real-time feedback of design parameters, and are dependent on the BIM parameter binding module and multi-modal command mapping mechanism; the system response path is long; the spatial component switch task involves switching the display status of the component and is dependent on the model state synchronization mechanism, but this delay is relatively controllable. In terms of interaction methods, gesture interaction directly invokes the control logic, has a short command trigger path, and the sensor recognition delay is stable; voice interaction, on the other hand, needs to use semantic parsing and text mapping processes, and there are time-consuming parsing steps and fluctuations in the accuracy of fault-tolerant recognition. Figure 4 shows the average response time comparison results of the two interaction methods for the above five types of tasks.

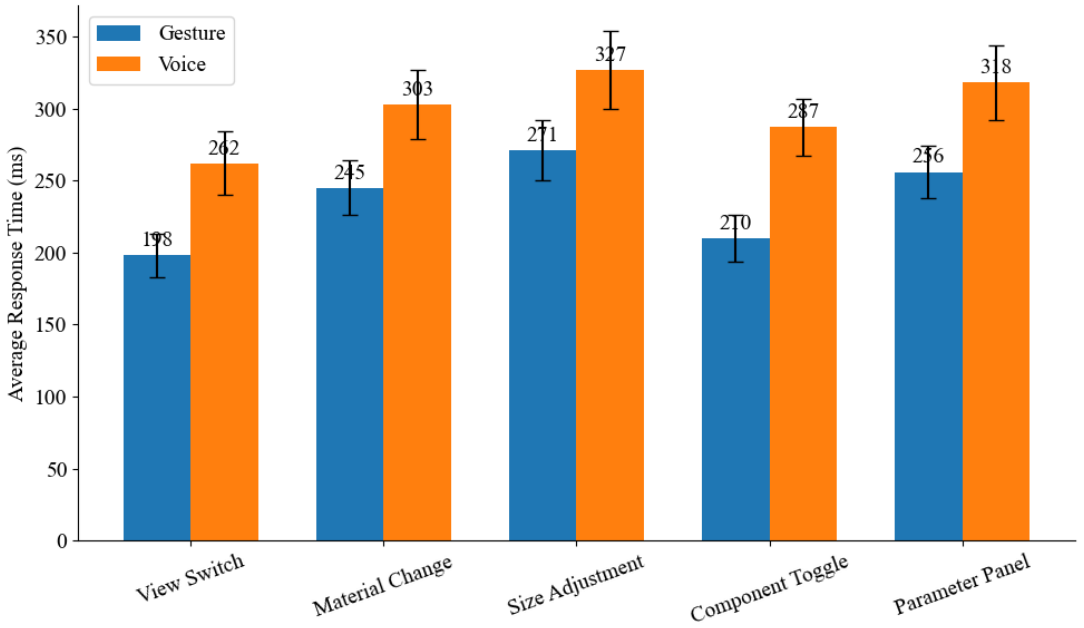

Figure 4: Comparison of Average Response Time for Natural Interaction Methods in Different Interactive Tasks

Voice interaction has a longer system response time for all task types, and the average response time of 318ms in the parameter-setting task (T5) is significantly higher than that of gesture interaction at 256ms. The main reason for this difference is that voice input needs to pass through a multi-stage processing path of voice recognition, semantic parsing, and command mapping, and thus has a higher structural complexity than the direct mapping path of gesture. The average response time for voice in the size adjustment (T3) and material replacement (T2) tasks was 327 ms and 303 ms, respectively, while that of gesture interaction was 271 ms and 245 ms; thus, voice showed a relatively large response-time increase in tasks that involved parameter linkage and geometric transformation. It can be seen from this that language input has structural limitations in conveying specific instructions; thus, a longer processing path and delayed responses occur under complicated circumstances. In the perspectives of switching (T1) and component switching (T4), the voice interaction response time was 262ms and 287ms, respectively, and the gesture response time was 198ms and 210ms. There is still an absolute gap between the two types of interactions, but the overall voice response is within an acceptable range; that is to say, in cases of low task complexity or clear feedback paths, the response time control of voice interaction is feasible. The effect of different types of interaction modes on the system's response speed generally varies according to the kind of task. The voice method has a longer response time for complex tasks because of a multi-stage processing flow (e.g., both the T3 and T5 response times exceed 300ms), and it also shows a relatively high standard deviation (e.g., the T3 voice standard deviation is 27ms, compared to 21ms for gestures), indicating a lack of stability; on the other hand, the gesture method uses a direct-mapping mechanism to keep the time fluctuations during interaction-intensive design relatively small and is therefore more suitable for real-time applications.

Based on the average response times of the different interaction modes measured in the previous stage, this stage will be divided into four sub-stages: recognition, binding, feedback and rendering, and the composition differences and trend changes of response time at each stage for various tasks will be examined. The Interaction Processes of the various task types are not the same. View switching and component switching are relatively low-complexity operations with short command mapping paths, simple feedback parameters, and distinct boundaries between stages. Size adjustment and parameter setting are multi-dimensional parameter analyses, local model reconstructions, and state refreshes of the graphics engine. Interactive response shows a close connection between the calculation of feedback and the update of rendering; as a result, there is a non-linear increase in response time with higher task complexity. Under the two input modes of gesture and voice, the processing mechanisms in the recognition and analysis stages differ; thus, the total response delay will be affected directly, and command binding and feedback stages are more dependent on task complexity and parameter structure. Figure 5 shows the response time of each task in the four interaction stages with broken lines, and separately plots the gesture and voice results to show how the interaction method and task difficulty affect the response structure of the system.

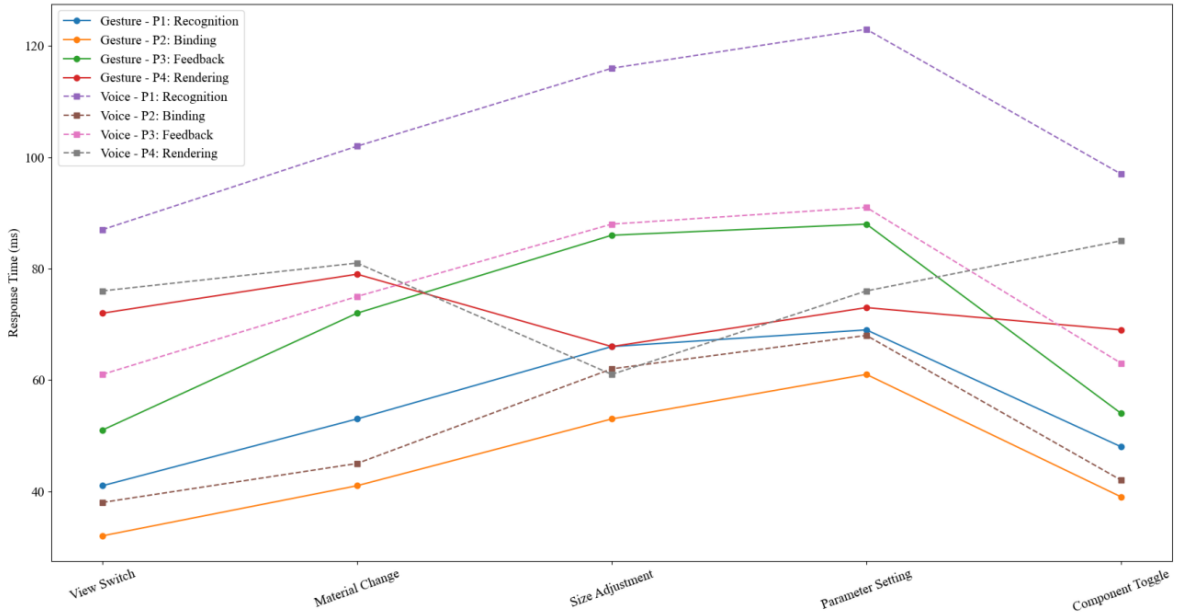

Figure 5: Stacked bar chart of time consumption analysis in interaction process stages

The two types of interaction modes, gestures and voices, have substantially different recognition and analysis characteristics for different tasks (P1). For high-complexity tasks (e.g., size adjustment and parameter setting), the time taken by voice interaction in this stage generally exceeds 100 milliseconds (123 ms for parameter setting tasks), and the time spent on parameter setting for gesture interaction under the same task is 69 ms. The reason for the above is that voice commands are subject to acoustic modelling and semantic decoding at multiple levels, while gesture path analysis only requires image sensor recognition and gesture matching. With an increase in the difficulty of the task, both the duration for command binding (P2) and parameter feedback (P3) will also be extended. The total time taken by P2+P3 for size adjustment and parameter setting of voice interaction is 150ms (62+88) and 159ms (68+91), respectively; this is significantly longer than that for low-complexity tasks (e.g., perspective switching P2+P3=99ms), and a mapping logic complexity amplification effect can be observed in the execution path. The rendering refresh (P4) stage has a range of 60-85 milliseconds (66-79 milliseconds for gestures and 61-85 milliseconds for voice), and the difference between voice and gestures in the same task is less than 6% (73ms for gestures vs. 76ms for voice in parameter setting tasks); it shows no dependence on the method of interaction and indicates that the graphics system has a certain decoupling capability for the source of interactive input.

5.3 Comparison of Customer Spatial Cognition Accuracy

Two recognition experimental processes were set up in this study: the traditional rendering environment and a virtual reality immersive scene, and the accuracy difference of component spatial recognition was investigated. Traditional Renderings are generally static in perspective and two-dimensional projection. Due to a lack of spatial imagination, they are often unable to grasp recognition dislocation, component occlusion or scale perception deviations. VR immersive environments offer a three-dimensional walking and perspective-switching system to provide users with a more realistic experience of the real-scale construction system. The recognition difficulty of the two environments is not uniform for all parts. Wall and column components are easily confused because they have linear ductility and similar materials.

Doors and windows are more difficult to identify precisely because they have small boundary details and no interactive feedback. In order to show the recognition performance differences of the two scenarios systematically, the component recognition heat map shown in Figure 6 is used to present the distribution of misjudgments for different component categories during recognition. In order to improve the contrast of the recognition performance differences, these values are not strictly standardised, and there are small discrepancies in the sum of some rows.

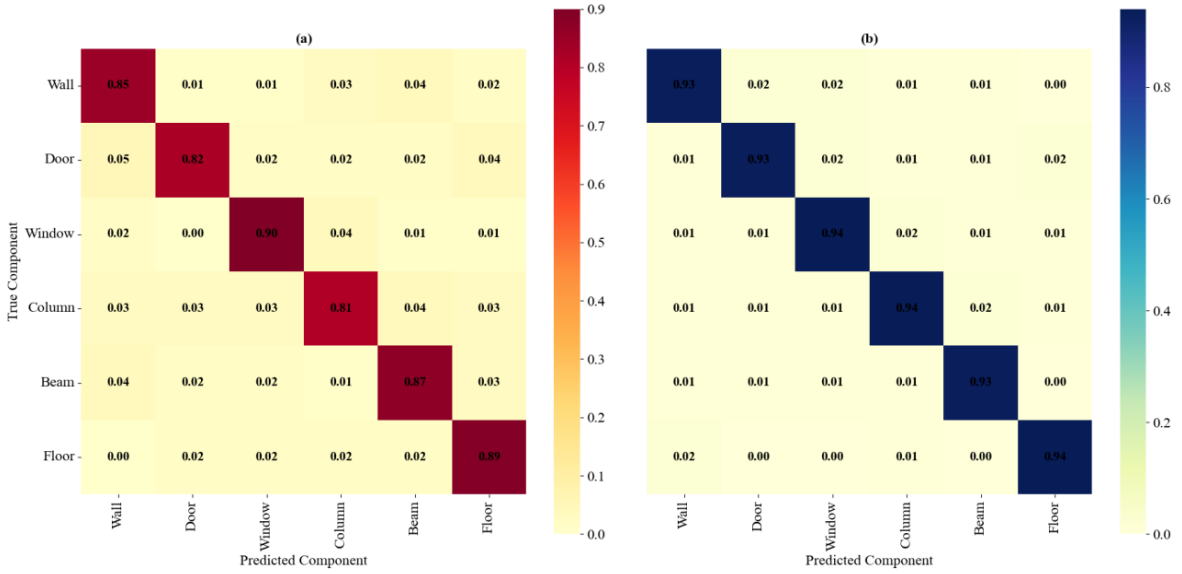

Figure 6(a) Traditional Rendering
Figure 6(b) VR immersive Scenes

Figure 6: Comparison of Component Recognition Error Rates in Traditional Rendering and VR Immersive Scenes.

In the traditional rendering scene, the recognition accuracy of the wall is 0.85, and the probability of misidentification as a column and beam is 0.03 and 0.04, respectively; this indicates a spatial judgment deviation for the linear contour components. The actual recognition rate of the column is only 0.81, and there are some relatively high-proportion misidentifications, such as walls (0.03) and doors (0.03); that is to say, it is easy to confuse the vertical extension and plane projection features of the column grid structure with those of other components in the absence of perspective. Although the door and window components have their own features, in a two-dimensional environment without interaction, the recognition accuracy of the door is 0.82 and that of the window is 0.90. Among them, the door is misidentified as a wall 5% of the time due to unclear boundary information; therefore, static perspectives are not suitable for conveying the three-dimensional characteristics of opening components. At the same time, the entire recognition performance of the VR immersive scene is significantly better than that of the traditional method, and the recognition accuracy of all components is above 0.93. For example, the recognition rate of window components increased to 0.94, and the proportion of column components in the misidentified items was the highest (0.02); however, the errors were also distributed among the categories of walls, doors and beams (each 0.01), indicating that although the overall recognition accuracy is high, some components near adjacent spaces or with similar features may still cause confusion. Based on the experiment, an immersive interactive environment can be employed to enhance the accuracy of spatial recognition and structural understanding by

dynamically observing component volumes and connection relationships along real-scale observation paths and multi-perspective exploration.

5.4 Customer Operation Behavior Coverage

In the immersive virtual reality architectural Design system, the user's natural interactive behaviour directly affects the efficiency of design parameter manipulation and the responsiveness of the system. Based on the modules of Leap Motion and voice recognition, eight types of gesture operations and six types of voice commands were established in this experiment, and the interaction behaviour frequency of 10 non-professional experimenters during the entire task was recorded. Gesture operations generally have highly intuitive control tasks, such as spatial construction and view adjustment, for example, wall creation, opening windows and doors, perspective roaming, hiding components, etc.; they are strong in physicality and immediate response, and suitable for users to achieve their intentions for spatial operations through actions. Most of the voice commands are for state switching or information query purposes, such as controlling lights, switching materials, scaling models, etc., and they do not require precise location or complex operation paths. Figure 7 is the frequency and proportion of the two types of natural interaction, gestures and voice, across all operation categories.

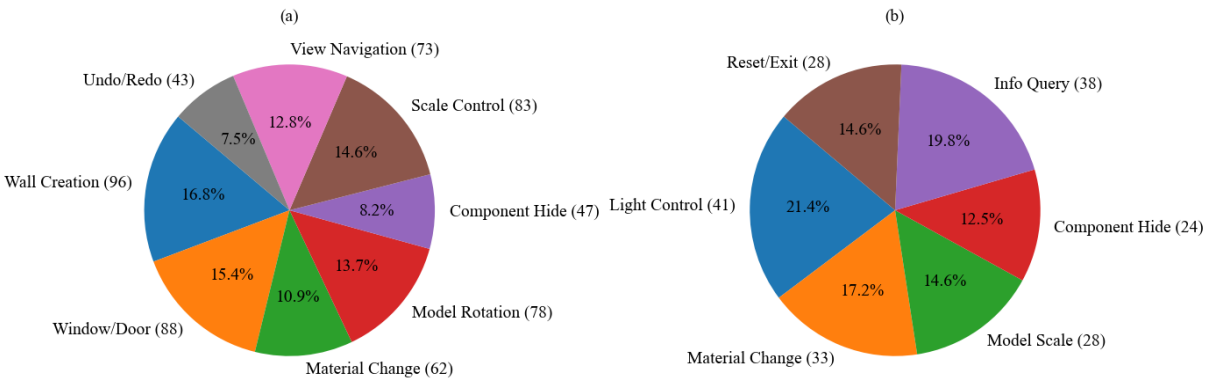

Figure 7: Frequency distribution of gesture and voice commands in the virtual reality building interaction system.

Among the distribution of user operating behaviour, gesture interaction occurred much more frequently than voice interaction; that is, 570 were started by gestures and 192 by voice. Therefore, action-type interaction in virtual reality is more suitable for the use habits of non-professional users. Among the gesture commands, wall building and zoom control were used most frequently, occurring 96 and 83 times, respectively; thus, users rely heavily on the basic model construction and vision adjustment functions. By comparison, the concealment of components and the functions of undo and redo were used less frequently; thus, these operations likely rely more heavily on judgment related to the context. Voice, light switches and information queries are generally the most frequently used functions for voice interaction; thus, voice is likely more suitable for tasks such as system state control and does not require spatial manipulation. Generally speaking, the results show that there are different functional areas for natural interaction patterns in architectural VR environments; thus, the system design should be in line with the mode of interaction and the type of task or user expectation to improve operational consistency and feedback efficiency.

5.5 Hot Zone Prediction Accuracy and Parameter Recommendation Acceptance Rate

Two main parts of the system's good performance in the process of immersive design feedback are the correct prediction of users' visual attention in the virtual scene and the quality of parameter advice generated based on this prediction. This study has built an LSTM-based visual hotspot model from eye-tracking movement data and, by means of a hotspot-parameter connection mechanism, generated directional design suggestion schemes. The experiment we have conducted shows that the users' gazes are not the same. The focus and direction of people's line of sight directly affect the stability and accuracy of the prediction; therefore, the functional category and display mode of the parameters will also impact whether designers are willing to accept the proposed adjustment schemes. Figure 8(a) shows the matching accuracy between actual gaze areas and predicted hot spots for all users, along with prediction stability error bars. Figure 8(b) shows how frequently proposals of different kinds are used, and thus how popular they are in practice.

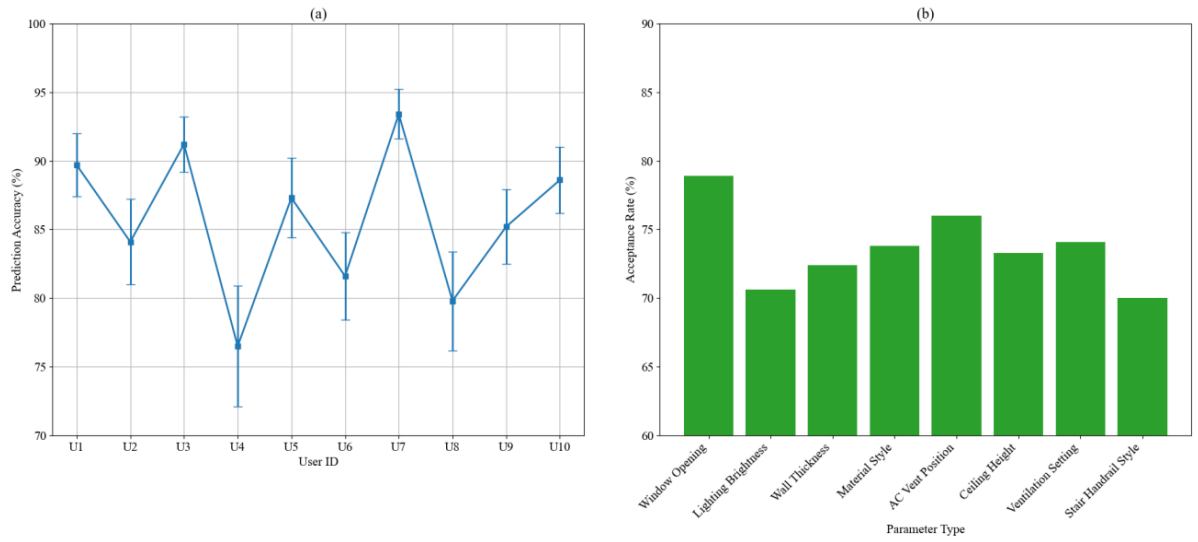

Figure 8(a) Hotspot Prediction Accuracy

Figure 8(b) Acceptance Rate of Parameter Recommendations

Figure 8: Performance Evaluation of Parameter Feedback Mechanism Based on Hotspot Prediction

Users U3 and U7 have achieved the highest prediction accuracy, at 91.2% and 93.4%, respectively. Their gaze moved between areas where the structure changed and zones of joint action, so this offered more attention points for the LSTM model to recognise. U4 was only 76.5 per cent; therefore, it changed hot spots frequently and repeatedly shifted to non-key areas, reducing prediction stability. Generally speaking, the accuracy is around 76% - 94%, so the model can perform somewhat differently according to how eyes are moved, but it is not stable in adverse circumstances either.

The accepting mode also appeared in the recommendation. Proposals for the parameters of functional space, such as window opening and ventilation arrangement, have received a relatively large number of approvals compared with more subjective suggestions on handrail or material style. Thus, it can be seen that the designers are relatively more willing to accept modifications in the areas of space function and comfort. Therefore, we will give more weight to the above parameters in the recommendation module to enhance its general appeal. Frequent submissions of function proposals in practice have sped up the plan approval

process and reduced unnecessary communication; thus, users have grown more trusting of the Design. Therefore, the maximum effect of the feedback mechanism can be achieved by fine-grained modelling of visual attention and function-driven parameter suggestions.

6 Conclusion

The present paper proposes an immersive visual display and customer mutual activity coordination framework based on virtual reality technology. Through the construction of a BIM-VR dynamic parameter coupling system, combined with octree space division, progressive mesh simplification and multimodal natural interaction technology, we have achieved efficient and lightweight processing of building models and real-time parameter-level control. Based on the above experiments, the model loading time has been reduced from the original average of 13.5 seconds to 6.7 seconds, the average frame rate has been raised from 35.6 fps to 62.2 fps, and the recognition accuracy for non-professional users in virtual reality scenes has been improved from 0.81 (the lowest value) for traditional renderings to more than 0.93. The system can achieve low time-delay interaction through Leap Motion gesture recognition and BERT fine-tuning of speech analysis. In the size adjustment task, the reaction time for gesture operation is 271ms, and it is relatively high compared to the 327ms of voice operation. The visual hot spot forecast module achieved 91.2% and 93.4% accuracy for users U3 and U7, respectively, but due to scattered look behaviour, the forecast accuracy for user U4 dropped to 76.5%. Limitations of this study include predictive stability under extreme gaze conditions, the complexity of time-delay control for voice-interactive work, and adaptive capabilities of multi-mode cooperation mechanisms. In the future work, we need to improve the generalisation ability of the model for dynamic gaze-moving paths, optimise the response speed of semantic analysis and parameter mapping, study data synchronisation and authority management mechanisms in cross-platform cooperation scenarios, and thus further promote the deep application of immersive design feedback throughout the entire lifecycle of construction.

Funding

This work was supported by the research foundation ability improvement project for young and middle-aged teachers in Guangxi colleges and universities in 2022: "Research on the virtual display of Baibuyao cultural tourism scenic spots in Guangxi under the rural revitalization strategy", Project number: 2022KY0778.

References

[1] Ehab, A., Burnett, G., & Heath, T. (2023). Enhancing public engagement in architectural design: A comparative analysis of advanced virtual reality approaches in building information modeling and gamification techniques. Buildings, 13(5), 1262-1285.

[2] Gomez-Tone, H. C., Alpaca Chávez, M., Vásquez Samalvides, L., et al. (2022). Introducing immersive virtual reality in the initial phases of the design process—Case study: Freshmen designing ephemeral architecture. Buildings, 12(5), 518-537.

[3] Ploennigs, J., & Berger, M. (2023). AI art in architecture. AI in Civil Engineering, 2(1), 8-19.

[4] Segovia, M., & Garcia-Alfaro, J. (2022). Design, modeling and implementation of digital twins. Sensors, 22(14), 5396-5426.

[5] Lin, S., & Chen, S. (2021). 3D design of gravity dam based on virtual reality CAD dynamic interactive system. Computer-Aided Design and Applications, 19(S5), 11-20.

[6] Li, X. (2021). An experimental study: Textual information driven spatial understanding and representation for user interface design of 3D modeling tools. Proceedings of the Design Society, 1(1), 437-446.

[7] Yu, R., Gu, N., Lee, G., et al. (2022). A systematic review of architectural design collaboration in immersive virtual environments. Designs, 6(5), 93-116.

[8] Chamusca, I. L., Cai, Y., Silva, P. M. C., et al. (2024). Evaluating design guidelines for intuitive, therefore sustainable, virtual reality authoring tools. Sustainability, 16(5), 1744-1769.

[9] Rajaratnam, D., Weerasinghe, D., Abeynayake, M., et al. (2022). Potential use of augmented reality in pre-contract design communication in construction projects. Intelligent Buildings International, 14(6), 661-678.

[10] Wijerathna, A., Perera, S., Nanayakkara, S., et al. (2024). Developing a workflow for transforming BIM models into immersive virtual reality experiences. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 48(1), 485-492.

[11] Zhou, M., Wang, J., Yu, B., et al. (2024). A quality management method for prefabricated building design based on BIM and VR-integrated technology. Applied Sciences, 14(4), 1635-1657.

[12] Chen, J., & Laokhongthavorn, L. (2024). Application of BIM and virtual reality of system integration design and development in medical building projects: A case study in China. Engineering and Technology Horizons, 41(3), 410301.

[13] Safikhani, S., Keller, S., Schweiger, G., et al. (2022). Immersive virtual reality for extending the potential of building information modeling in architecture, engineering, and construction sector: Systematic review. International Journal of Digital Earth, 15(1), 503-526.

[14] Chan, C. S., Bogdanovic, J., & Kalivarapu, V. (2022). Applying immersive virtual reality for remote teaching architectural history. Education and Information Technologies, 27(3), 4365-4397.

[15] Žujović, M., Obradović, R., Rakonjac, I., et al. (2022). 3D printing technologies in architectural design and construction: A systematic literature review. Buildings, 12(9), 1319-1343.

[16] Liu, Z., Gu, X., Dong, Q., et al. (2021). 3D visualization of airport pavement quality based on BIM and WebGL integration. Journal of Transportation Engineering, Part B: Pavements, 147(3), 04021024-04021051.

[17] Li, T., Hu, H., Ma, H., et al. (2025). Using virtual reality to enhance learning performance and address educational resource disparities in architectural history courses. Sustainability, 17(3), 866-888.

[18] Ummihusna, A., & Zairul, M. (2022). Investigating immersive learning technology intervention in architecture education: A systematic literature review. Journal of Applied Research in Higher Education, 14(1), 264-281.

[19] Latif Rauf, H., S. Shareef, S., & Najim Othman, N. (2021). Innovation in architecture education: Collaborative learning method through virtual reality. Journal of Higher Education Theory and Practice, 21(16), 33-40.

[20] Ummihusna, A., Zairul, M., Ab Jalil, H., et al. (2025). Immersive virtual reality in experiential learning for architecture design education: An action research. Journal of Applied Research in Higher Education, 17(2), 738-758.

[21] Hajirasouli, A., Banihashemi, S., Sanders, P., et al. (2024). BIM-enabled virtual reality (VR)-based pedagogical framework in architectural design studios. Smart and Sustainable Built Environment, 13(6), 1490-1510.

[22] Ibrahim, A., Al-Rababah, A. I., & Bani Baker, Q. (2021). Integrating virtual reality technology into architecture education: The case of architectural history courses. Open House International, 46(4), 498-509.

[23] Elbadawy, H., & Farouk, A. (2025). Implementation of virtual reality technology in architecture field, and education: A review. Archives of Computational Methods in Engineering, 32(1), 1-9.

[24] Podkosova, I., Reisinger, J., Kaufmann, H., et al. (2022). Bimflexi-vr: A virtual reality framework for early-stage collaboration in flexible industrial building design. Frontiers in Virtual Reality, 3(1), 782169-782182.

[25] Maksoud, A., Hussien, A., Mushtaha, E., et al. (2023). Computational design and virtual reality tools as an effective approach for designing optimization, enhancement, and validation of Islamic parametric elevation. Buildings, 13(5), 1204-1259.

[26] Shehadeh, A., & Alshboul, O. (2025). Enhancing engineering and architectural design through virtual reality and machine learning integration. Buildings, 15(3), 328-352.

[27] Zhang, Y., Wang, Z., Zhang, J., et al. (2023). A survey of immersive visualization: Focus on perception and interaction. Visual Informatics, 7(4), 22-35.

[28] Zhang, T., Wang, Y., Zhou, X., et al. (2025). Intelligent human–computer interaction for building information models using gesture recognition. Inventions, 10(1), 5-28.

[29] Zhang, C., Zeng, W., & Liu, L. (2021). UrbanVR: An immersive analytics system for context-aware urban design. Computers & Graphics, 99(1), 128-138.

[30] Bier, H., Hidding, A., Brancart, S., et al. (2024). AI-supported approach for human-building interaction implemented at furniture scale. Frontiers in Computer Science, 6(1), 1295014-1295023.

[31] Mejia-Puig, L., & Chandrasekera, T. (2023). The presentation of self in virtual reality: A cognitive load study. Journal of Interior Design, 48(1), 29-46.

[32] Díaz González, E. M., Belaroussi, R., Soto-Martín, O., et al. (2025). Effect of interactive virtual reality on the teaching of conceptual design in engineering and architecture fields. Applied Sciences, 15(8), 4205-4228.