Design and Research of an Intelligent Houdini Model Library Based on Procedural Modeling Technology

Haolong Yang, Jiayi Shen, Ning Mao, Yicong Lin, Guwei Li*
Artificial Intelligence College, Zhejiang Dongfang Polytechnic, Wenzhou, Zhejiang, China
**Corresponding Author*

Abstract: Aiming at the problems of low efficiency, poor reusability and difficulty in dynamic adjustment of traditional 3D modeling, this paper proposes a design scheme of an intelligent Houdini model library based on Procedural Content Generation (PCG) technology. This scheme integrates Houdini's node-based workflow, VEX programming language and AI-assisted generation mechanism to construct an intelligent model library system that supports parametric control, logical combination and automatic error correction. By parsing the model generation rules in fields such as architecture and games, basic components are encapsulated as procedural assets, realizing the rapid generation and dynamic iteration of complex scenes. Experiments show that this model library can shorten the production cycle of similar models by more than 60%, and has good scalability and fault tolerance, providing an efficient solution for large-scale digital content production.

Keywords: Procedural Modeling; Houdini; Parametric Design; Digital Content Generation

1. Introduction

1.1 Research Background

With the rapid development of fields such as game development, film and television production, and digital twins, higher requirements have been put forward for the quantity, complexity and iteration efficiency of 3D models. The traditional manual modeling method relies on designers to construct models point by point, which has disadvantages such as long production cycle, low reusability and high modification cost, making it difficult to meet the needs of large-scale scene generation and dynamic adjustment. Procedural Content Generation (PCG) technology automatically generates 3D content through algorithmic rules, which can significantly improve modeling efficiency and realize content diversification and customization, becoming the core technical direction to solve the above problems. As a leading procedural modeling tool in the industry, Houdini's node-based workflow has the advantages of non-destructiveness, editability and repeatability, supporting the full-process procedural construction from basic geometry to complex scenes. However, current Houdini-based modeling applications mostly focus on the generation of single scenes or specific assets, lacking systematic model library support, resulting in low asset reuse rate and difficulty in inheriting generation logic, which limits the large-scale application of procedural modeling technology. Therefore, building a Houdini model library with intelligent scheduling, parametric configuration and automatic error correction capabilities has become a key demand to improve the efficiency of digital content production.

1.2 Research Status

Scholars at home and abroad have carried out relevant research on procedural modeling and model library design. At the technical level, the Procedural Scene Programs (PSP) framework proposed by Brown University and the University of California, San Diego realizes procedural construction and automatic error correction of scenes by generating executable scripts, which significantly improves the physical rationality and controllability of 3D scenes. The 3D-GPT framework integrates Large Language Models (LLMs) and multi-agent systems to realize the automatic generation of 3D models from text instructions, providing a new paradigm for intelligent modeling. At the application level, relevant research combines Houdini with VEX language and applies it to the procedural modeling of traditional architecture. By separating architectural forms from basic components and building a parametric component library, it realizes the rapid generation and flexible adjustment of traditional residential models. In the field of digital twins, the integrated workflow of Houdini and real-time engines is used for modeling large-scale scenes such as cities and factories, realizing the digital reconstruction of the real world by parsing data such as GIS and BIM. These studies have verified the advantages of procedural modeling in efficiency improvement and scene generation, but there is still room for improvement in aspects such as intelligent scheduling of model libraries, cross-scene adaptability, and deep integration of AI and procedural logic.

2. Related Technical Foundation

2.1 Principles of Procedural Modeling Technology

Procedural modeling is a technical method to automatically generate 3D content through algorithmic rules, parameter configuration or logical instructions, and its core characteristics include automation, customizability, dynamics and infinite possibilities. This technology realizes the generation from simple geometry to complex scenes through core algorithms such as noise functions, fractal algorithms, L-systems and cellular automata. Compared with traditional manual modeling, procedural modeling does not require point-by-point construction; it only needs to adjust parameters or modify rules to generate diversified results, which greatly improves modeling efficiency and content diversity. In practical applications, the workflow of procedural modeling usually includes four stages: rule definition, parameter configuration, generation execution and result optimization. In the rule definition stage, the logical relationship of content generation is clarified through code or visual nodes; in the parameter configuration stage, key attributes of the generated content (such as size, quantity, distribution density, etc.) are set; in the generation execution stage, 3D content is automatically generated through program operation; in the result optimization stage, content quality is improved through manual adjustment or automatic error correction mechanisms [1].

2.2 Houdini Procedural Modeling Features

Houdini's procedural modeling workflow is realized based on node networks. Each modeling step is recorded in the form of nodes, supporting full-process non-destructive editing. Users can backtrack to any step for modification at any time without rework, which significantly improves the process fault tolerance rate. Houdini takes nodes as the basic operation unit, supports data input/output and attribute preservation, constructs complex modeling logic through node connections, and all operation processes can be displayed visually [2]. It supports multi-dimensional modeling, covering functions such as basic geometric modeling, topology reconstruction and point cloud data processing, and supports the import, cleaning and geometric transformation of LIDAR data, which can meet the modeling needs of different scenes. It has a built-in VEX programming language, allowing users to customize modeling logic through code to realize parametric control and automatic generation of complex components. It supports encapsulating complex node networks into Houdini Digital Assets (HDAs) to realize the reuse and cross-project sharing of modeling logic. These features make Houdini an ideal platform for building intelligent model libraries, capable of supporting the full-process procedural generation and management from basic components to complex scenes.

2.3 AI-Assisted Procedural Modeling Technology

In recent years, the integration of AI technology and procedural modeling has become a research hotspot. The PSP framework realizes procedural construction of scenes by generating executable scripts, and its core includes the Procedural Scene Description Language (PSDL) and the Program Search error correction module [3]. The former is used to define scene generation logic, and the latter realizes automatic error correction through geometric consistency detection, requiring only 7 program modifications on average to fix most errors. The 3D-GPT framework uses LLMs to build a multi-agent system, which realizes the conversion from text instructions to Python code through the collaborative work of task scheduling, conceptualization and modeling agents, and then calls 3D software APIs to generate models. These technologies provide core support for intelligent model libraries: LLMs are used to parse natural language instructions and extract key parameters and generation rules; automatic error correction modules are used to optimize the geometric consistency and physical rationality of models; multi-agent systems realize the automatic decomposition and execution of modeling tasks, providing technical guarantee for the intelligent scheduling of model libraries [4].

3. Architecture Design of the Intelligent Houdini Model Library

3.1 Overall Architecture

The intelligent Houdini model library designed in this paper adopts a "three-layer architecture", including the basic component layer, the rule engine layer and the intelligent interaction layer, and each layer achieves efficient collaboration through data interfaces. Among them, the basic component layer is the core asset of the model library, including procedurally encapsulated basic geometric components, industry-specific components (such as building components, game props) and composite scene templates, and all components support parametric control. The rule engine layer is responsible for component scheduling, combination and optimization, including classification management module, parameter mapping module, logical combination module and automatic error correction module, realizing intelligent matching of components and reasonable generation of scenes; the intelligent interaction layer provides a variety of user interaction interfaces, including parameter configuration interface, node-based editing window and natural language interaction module, supporting efficient operation of users with different technical backgrounds [5].

3.2 Core Module Design

3.2.1 Design of basic component library

The basic component library adopts the design idea of "classified storage - parametric encapsulation", and is divided into two categories: general components and special components according to application fields: general components include geometric bodies (cubes, spheres, surfaces, etc.) and basic structures (frames, supports, connecting parts, etc.), supporting the adjustment of basic parameters such as size, material and subdivision degree; special components are designed for specific needs in fields such as architecture, games and digital twins, such as plinths, beams and roof components in the architectural field, props and terrain components in the game field, equipment models and pipeline components in the digital twin field, etc. All components are encapsulated through Houdini's HDA technology, and each component includes a visual parameter panel and a node network core [6]. Taking building components as an example, by separating component forms from control parameters, parameters such as the diameter, height and decorative patterns of plinths are set as adjustable parameters, and parameters such as the length, cross-sectional size and texture style of beams are bound through VEX code to realize flexible customization of components. The component library supports version management and incremental updates, and users can expand custom components according to their needs.

3.2.2 Rule engine module

The rule engine module is the core to realize the intelligence of the model library, responsible for parsing user needs, scheduling component resources and generating reasonable scenes. It mainly includes: classification management module, which can realize fast retrieval and positioning of components through label matching (such as "Architecture - Wooden Structure - Beam", "Game - Prop - Weapon"); parameter mapping module, which can establish the mapping relationship between user needs and component parameters, support extracting key parameters from text instructions or visual configurations (such as "Generate a three-bay and four-column porch", "Create a cylinder with a height of 5 meters"), and automatically match corresponding components and configure parameters; logical combination module, which defines the combination relationship between components based on domain rules, such as modular assembly rules of building components and connection logic of mechanical parts, and automatically realizes the spatial layout and assembly of components through node networks; automatic error correction module, which integrates the Program Search mechanism of the PSP framework, defines geometric consistency indicators (such as overlap rate, support relationship, occlusion situation), automatically detects generated scenes, and repairs geometric errors by adjusting component parameters or replacing expressions without re-invoking the model generation process [7].

3.2.3 Intelligent interaction module

To improve the usability of the model library, a multi-mode interaction interface is designed, providing: a visual panel, allowing users to directly adjust parameters such as component size, quantity and distribution density, and preview generation results in real time; native Houdini node network, allowing advanced users to customize modeling logic and expand component generation rules; integrated LLM model, supporting users to generate needs through natural language descriptions (such as "Generate a flickering lamp on the dock at dusk"), the system automatically parses text instructions, extracts scene elements, spatial relationships and attribute parameters, and calls corresponding components to generate scenes.

4. Implementation and Application Verification of the Model Library

4.1 Development Environment and Technology Stack

This model library is developed based on Houdini 19.5, and the core technology stack includes: Houdini node network and VEX programming language for procedural implementation of components; Python for integrating LLMs and automatic error correction algorithms, calling Houdini API to realize data interaction [8]; JSON format for storing component attributes and rule configurations, supporting fast reading and modification; Houdini's Python States for building custom parameter panels and natural language input windows.

4.2 Core Function Implementation

4.2.1 Procedural encapsulation of components

In terms of core function implementation, the procedural encapsulation of components is represented by plinth components in the architectural field, and its encapsulation process is as follows: first, use Houdini nodes such as Poly Extrude and Bevel to construct the basic form of the plinth [9]; then define the generation rules of decorative patterns through VEX code, supporting parametric control of pattern types (such as meander patterns, scroll patterns) and density; then encapsulate it as an HDA asset, exposing adjustable parameters such as diameter, height, pattern type and subdivision degree; finally, add the label "Architecture - Wooden Structure - Plinth" and record the component attribute information into the model library. Components realized through this process support real-time parameter adjustment; after modifying parameters such as diameter and height, the model can be updated automatically while maintaining the integrity of the topological structure.

4.2.2 Intelligent scene generation

Intelligent scene generation is exemplified by "generating Jianghuai patio-style dwellings", and the specific process is as follows: the user inputs "Generate a three-bay and two-story Jianghuai patio-style dwelling" through natural language; the natural language interaction module parses this instruction and extracts key parameters—the building type is Jianghuai patio-style dwelling, the number of bays is 3, and the number of stories is 2; then the rule engine module retrieves the architectural form rule library, calls corresponding spatial layout templates and basic components (columns, beams, roof, etc.); then according to modular assembly rules, automatically realizes the spatial layout and assembly of components through node networks to generate the main structure of the dwelling; then the automatic error correction module detects problems such as component overlap and support relationship, fine-tunes component positions and sizes to ensure geometric consistency; finally outputs the complete dwelling model, supporting users to further adjust parameters or modify details [10].

4.2.3 Implementation of automatic error correction mechanism

The automatic error correction mechanism is realized based on the Program Search mechanism of the PSP framework, and the specific process is as follows: first define geometric consistency detection indicators, including component overlap rate (threshold ≤ 5%), support relationship (the bottom of vertical components must contact the support surface), and occlusion rationality (key functional components cannot be completely occluded); after generating the scene, write detection scripts through Houdini's Geometry Wrangle node to calculate various indicators; when anomalies are detected (such as excessive overlap rate), the system automatically retrieves the parameter space, adjusts component positions and sizes or replaces conflicting generation logic; then repeats the detection and adjustment process until geometric consistency requirements are met, with the average number of adjustments not exceeding 7 times.

4.3 Application Verification and Performance Analysis

4.3.1 Verification scene design

In the application verification and performance analysis, two typical application scenarios are selected: architectural modeling and game scene generation. In the architectural modeling scenario, 3 different types of traditional residences in central Anhui are generated (Jianghuai patio-style, Jianghuai courtyard-style, and weibao-style), with 5 instances of different parameter configurations for each type; in the game scene generation scenario, a forest scene in an open-world game is generated, including 10 types of assets such as trees, rocks and vegetation, with a scene scale of 1km × 1km.

4.3.2 Performance indicators and test results

Performance evaluation adopts three core indicators: modeling efficiency, model quality and scalability. In terms of modeling efficiency, compared with traditional manual modeling, the modeling time of architectural scenes is reduced from an average of 8 hours per model to 2.5 hours per model, with an efficiency improvement of 68.75%; the modeling time of game scenes is reduced from an average of 12 hours to 3.8 hours, with an efficiency improvement of 68.33%. In terms of model quality, through the automatic error correction module, the component overlap rate is controlled within 3%, the compliance rate of support relationships and physical rationality is 95%, and the model topological structure is complete, which can be directly used for animation and rendering. In terms of scalability, the model library supports the rapid integration of new components, with the average time-consuming of adding a new type of component not exceeding 1 hour, and does not affect the normal use of existing components. Test results show that this intelligent Houdini model library can significantly improve modeling efficiency, ensure model quality, and has good scalability, meeting the application needs of different fields.

5. Conclusion

This paper designs and implements an intelligent Houdini model library based on procedural modeling technology. Through the three-layer architecture of "basic component layer - rule engine layer - intelligent interaction layer", integrating Houdini's node-based workflow, VEX programming language and AI-assisted generation technology, it realizes parametric encapsulation, intelligent scheduling of basic components and rapid generation of complex scenes. Research results show that the deep integration of procedural modeling technology and Houdini can effectively improve the reusability and generation efficiency of models, and shorten the modeling cycle; the introduction of rule engine and automatic error correction module significantly improves the rationality and stability of model generation, and reduces modification costs; the design of multi-mode interaction interface enhances the usability of the model library, meeting the operation needs of users with different technical backgrounds. This model library provides an efficient solution for large-scale digital content production, and is applicable to multiple fields such as architecture, games and digital twins.

References

[1] Meta Reality Labs. “Varifocal Display Latency Optimization.” SID Symposium Digest, 54(1), 112-115, 2023.
[2] DeepMind. “Reinforcement Learning for Adaptive Fire Scenarios.” Nature Machine Intelligence, 6(7), 732-744, 2022.
[3] Yang Dai, Zhiyuan Luo. “Review of Unsupervised Person Re-Identification.” Journal of New Media, 3(4), 129-136, 2021.
[4] Meskat Jahan, Manajir Hassan, Sahadat Hossin, et al. “Unsupervised person Re-identification: A review of recent works,” Neurocomputing, 572(1), 127193–127196, 2024.
[5] Zhenping Xia, Qishuai Han, Yuning Zhang, et al. “Objective quantification of dynamic spatial distortions for enhanced realism in virtual environments.” Virtual Reality, 29(1), 39-39, 2025.
[6] Mohammad BaniSalman, Mohammad Aljaidi, Najat Elgeberi, et al. “VRDeepSafety: A Scalable VR Simulation Platform with V2X Communication for Enhanced Accident Prediction in Autonomous Vehicles.” World Electric Vehicle Journal, 16(2), 82-82, 2025.

[7] R. Wazirali, “Aligning education with vision 2030 using augmented reality,” Computer Systems Science and Engineering, 36(2), 342–351, 2021.
[8] Menzemer Leo Willem, Ronchi Enrico, Karsten Mette Marie Vad, et al. “A scoping review and bibliometric analysis of methods for fire evacuation training in buildings.” Fire Safety Journal, 136(1), 15–17, 2023.
[9] Ali Haider, Ilya Morev, Aleksi Rintanen, et al. “Accelerated numerical simulations of hydrogen flames: Open-source implementation of an advanced diffusion model library in OpenFOAM,” International Journal of Hydrogen Energy, 189(1), 152115-152115, 2025.
[10] Zekai Liu, Genming Lai, Yunxing Zuo, et al. “AI-driven next-generation lithium-ion battery design automation (BDA) software,” National Science Open, 4(6), 10–15, 2025.

