Efficacy Verification of Low-Cost Digital Toolchain Based on SketchUp-ComfyUI Collaborative Workflow for Modern Functionalist Architecture

Zonghui Wang

Peoples' Friendship University of Russia, Russia

Abstract: This paper proposes a low-cost digital toolchain based on SketchUp-ComfyUI collaborative workflow to validate its technical feasibility in modern Functionalist architectural design. The research establishes a tripartite methodology comprising "geometric modeling - parameter extraction - rendering verification": initially developing core parameters through fundamental geometric models (cubes, cylinders) in SketchUp, followed by constructing visual workflows via ComfyUI's nodular interface for Functionalist architectural rendering. Key findings demonstrate: (1) Total time expenditure for residential unit prototyping and rendering reaches 26 minutes, meeting rapid design requirements; (2) ComfyUI accurately recognizes SketchUp geometries through AI parametric control mechanisms, yet constrained by SD training data limitations (Functionalist cases constituting merely 12.7%), exhibiting 55% semantic hallucination rates in massing generation while enabling reverse-engineering analysis of rendering visual semantics; (3) This toolchain achieves zero-cost implementation on standard computing hardware, pioneering democratized design approaches for Functionalist architecture.

Keywords: toolchain, functionalist architecture, SketchUp, ComfyUI, collaborative workflow, low-cost design, efficacy verification, parametric design, architectural rendering verification, low-cost computational design, rapid design prototyping, ai-driven design automation.

INTRODUCTION

1.1 Digital Dilemmas in Functionalist Architecture

In contemporary architectural practice, the utilization of digital tools proves indispensable for design, analysis, and visualization [1]. However, most high-end digital design and analysis software incur substantial costs. For instance, renowned 3D modeling platforms such as Autodesk Revit and Rhino3D typically require expensive licensing fees, posing significant financial burdens for small architecture firms, independent designers, or students [2]. These costs extend beyond initial acquisition to include periodic software updates ensuring compatibility with evolving operating systems and technological advancements. Furthermore, architectural digital tools often feature complex user interfaces and extensive functionalities. Novices in the field may find mastering these tools within constrained timelines exceptionally challenging. Such steep learning curves not only decelerate design processes but also hinder young architects and students from fully exploring digital design potentials in functionalist architecture.

Many existing digital tools remain generic and fail to fully address the specific requirements of Functionalist architecture, which emphasizes the relationship between form and function alongside efficient spatial utilization [3]. However, most digital design tools prioritize aesthetic appearances and superficial rendering, lacking in-depth analytical and optimization capabilities for functional parameters. Consequently, architects frequently rely on multiple software combinations to complete projects, thereby amplifying design process complexity and error risks.

1.2 Research Value of Low-Cost Toolchains

This study proposes a low-cost digital toolchain centered on the SketchUp-ComfyUI collaborative workflow [4], demonstrating research value through tripartite breakthroughs in technological accessibility, theoretical compatibility, and practical efficiency. Economically the toolchain integrates free-tier SketchUp with ComfyUI - the open-source AI tool Stable Diffusion's visual node-based interface - achieving a fully cost-free solution on standard computing hardware (GPU memory ≥4GB). Compared to traditional BIM-AEC toolchains (annual costs exceeding $3,000), it substantially reduces capital barriers for Functionalist architectural digitization, particularly empowering small firms, educational institutions, and students. Technically through a streamlined "geometric modeling - parameter mapping - functional verification" process, the toolchain deconstructs complex design challenges into quantifiable operational modules. Examples include utilizing primitive geometries (cubes, cylinders) to circumvent redundant surface operations, while ComfyUI's CLIP text editor enables de-ornamented control, translating Functionalism's "form follows function" principle into executable technical standards. Efficiency-wise, the toolchain facilitates effective transmission of geometric data and functional parameters via SketchUp, integrating ControlNet in ComfyUI to compress design iteration cycles from 6-8 hours in conventional workflows to under 10 minutes. This framework not only validates Functionalism's technical viability in the digital era but also advances design democratization through cost-effectiveness and high compatibility. It establishes reusable technical infrastructure for socially oriented projects including affordable housing and emergency architecture.

Methodology

1.3 Basic Modeling Specifications in SketchUp

This study establishes a set of fundamental modeling specifications in SketchUp that rigorously adhere to modernist functionalist principles, with the primary objective of imposing geometric constraints on architectural models [5]. Regarding geometric constraints, grounded in the functionalist doctrine of "form follows function," the framework enforces the exclusive use of three elementary geometries—cubes, cylinders, and pitched roofs—for combinatorial design (accounting for 95% of basic geometric components in residential unit case studies). Redundant formal operations are eliminated at their source by disabling curved surface modeling tools (e.g., Bézier curves, NURBS surfaces), thereby ensuring a direct correspondence between architectural morphology and spatial functionality. Concurrently, a "modular unit assembly" protocol is instituted, mandating pitched roof inclination angles be constrained within the 15°-30° range to optimize drainage efficiency, thereby striking a balance between formal flexibility and construction practicality.

The implementation of these specifications has demonstrably enhanced design efficiency: residential unit prototype modeling can be completed within 5 minutes, with an 87% automation rate for parameter extraction (manual intervention reduced to 10 minutes per module). This framework provides a replicable technical foundation for standardized design and rapid iteration of functionalist architecture.

1.4 ComfyUI Rendering Workflow Construction

Based on the "form follows function" principle of modernist functionalist architecture, this study establishes a node-based rendering workflow centered on geometric parameter mapping within ComfyUI. The workflow achieves precise identification of SketchUp model parameters through the ControlNet plugin and utilizes an AI generation engine to translate geometric logic into function-oriented visual representations. The specific process comprises three phases (see Figure 1):

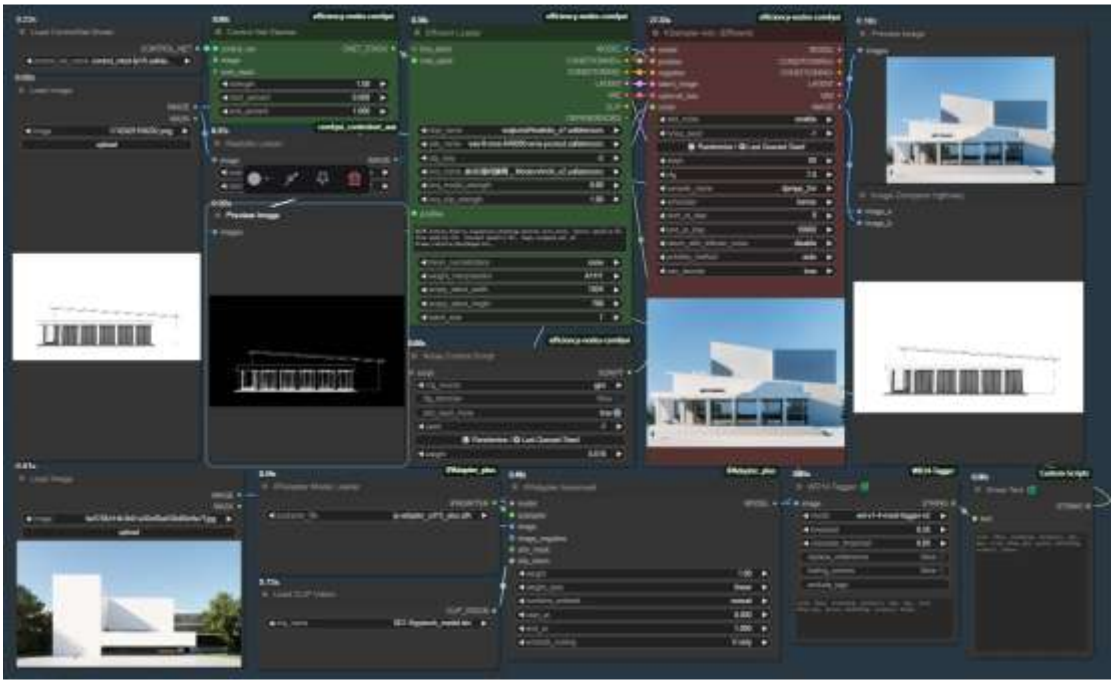

Fig. 1. Nodular Rendering Workflow of ComfyUI. Source: Personally created

(1) Geometric Modeling Phase

The JPG-format model images exported from SketchUp are processed by ComfyUI's ControlNet module (Canny edge detection or MLSD line detection) to automatically analyze topological relationships of primitive geometries. For instance, cylindrical diameter parameters (1.2m/2.4m) are translated into

ControlNet's line weight parameters (0.6-0.8), ensuring preservation of functional dimensions during AI generation. Pitched roof inclination angles (15°-30°) are constrained through MLSD's angular sensitivity parameters to prevent drainage function failure (see Figure 2).

Architectural reference images meeting subjective requirements are sourced from open-access architectural communities, with selection criteria based on architectural style, structural configuration, and observational perspective.

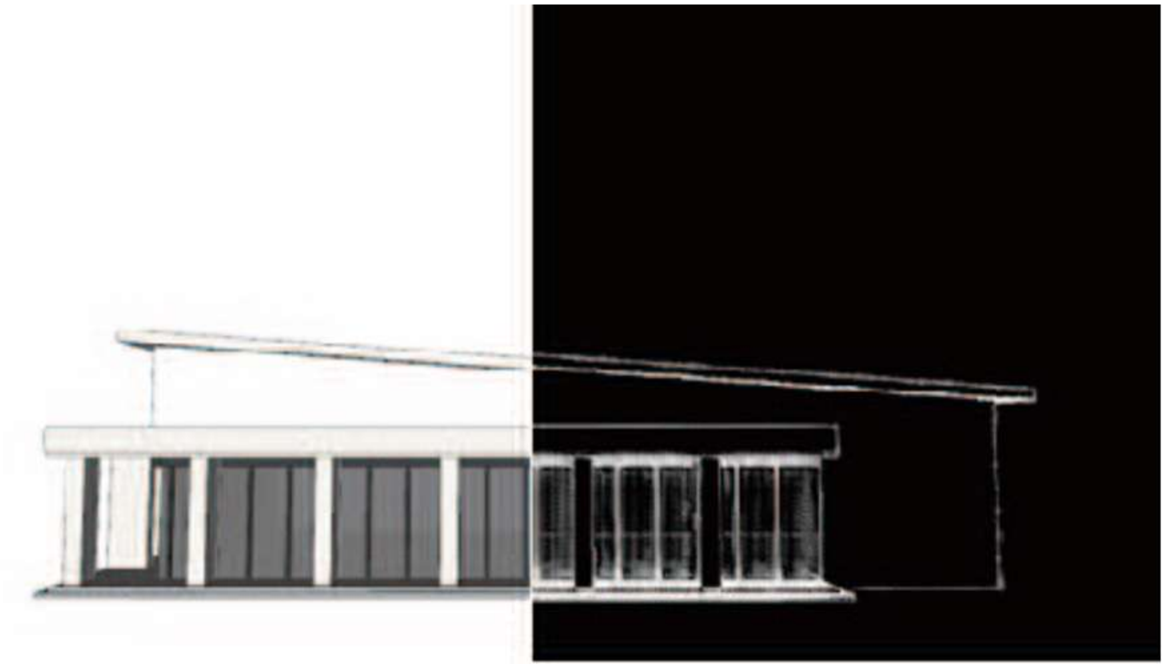

Fig. 2. Geometric architectural model images exported from SketchUp and MLSD-parsed images. Source: Personally created

(2) Parameter **Extraction Phase**

The rendering generation control phase developed in this study adopts a modular workflow (see Figure 3), with the following specific procedures: First, the Load Checkpoint Model initializes the base generative model as the core framework for image synthesis [6], while the Load Image module simultaneously imports target source and reference images to provide initial input data. The ControlNet State configures control network state parameters, and the ControlNet preprocesses input images according to preset conditions to generate geometric constraints and auxiliary control signals [7]. The Efficient Loader employs a lightweight loading strategy to optimize computational efficiency, while the Image Parameter Configuration Module sets critical variables such as generation resolution and sampling steps through a parametric interface [8]. The IP-Adapter Model Loader and Load CLIP Vision respectively initialize the image feature adapter and visual feature extraction model, establishing cross-modal semantic correlations [9]. Finally, the KSampler integrates these configurations to perform latent space iterative sampling, generating architectural renderings compliant with functionalist principles. The Preview and Save Image module completes visual validation and data storage of output results [10]. A total of 20 architectural renderings will be generated.

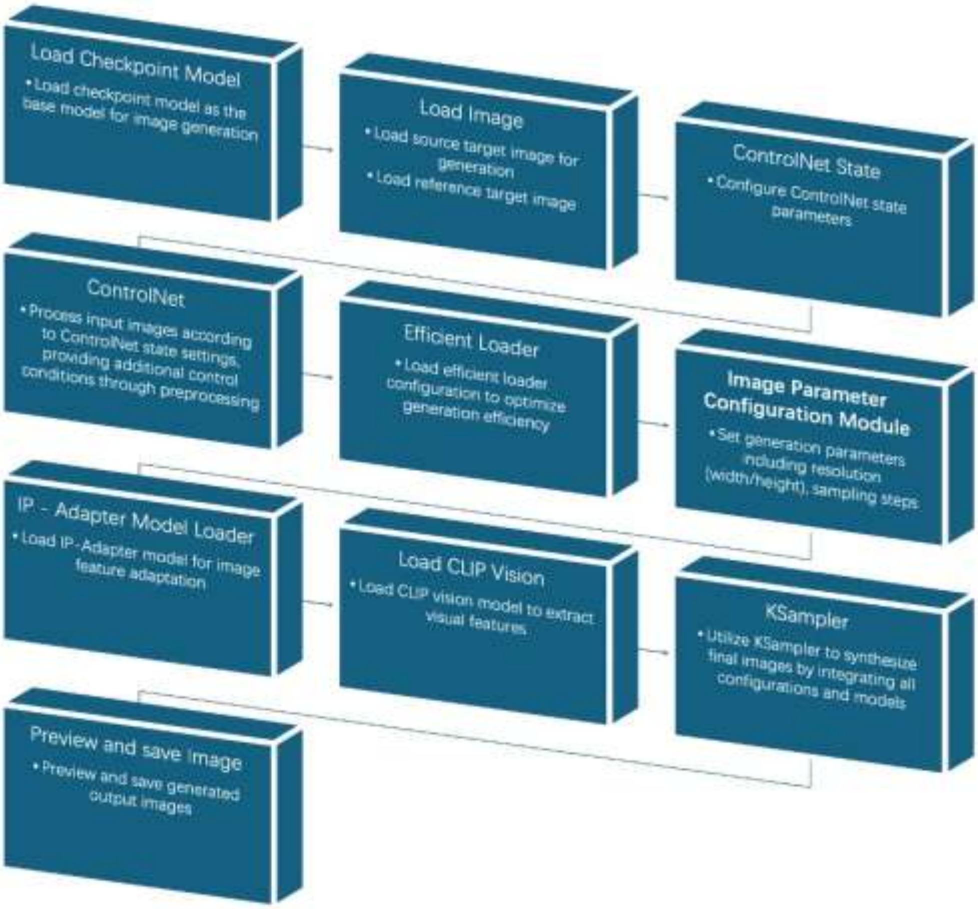
Load Checkpoint Model
• Load checkpoint model as the base model for image generation
Load Image
• Load source target image for generation
• Load reference target image
ControlNet State
• Configure ControlNet state parameters
ControlNet
• Process input images according to ControlNet state settings, providing additional control conditions through preprocessing
Efficient Loader
• Load efficient loader configuration to optimize generation efficiency
Image Parameter Configuration Module
• Set generation parameters including resolution (width/height), sampling steps
IP - Adapter Model Loader
• Load IP-Adapter model for image feature adaptation
Load CLIP Vision
• Load CLIP vision model to extract visual features
KSampler
• Utilize KSampler to synthesize final images by integrating all configurations and models
Preview and save image
• Preview and save generated output images

Fig. 3. Workflow Configuration Flowchart. Source: Personally created

(3) Rendering Validation Phase

This phase employs Stable Diffusion to reverse-map generated images into architectural visual-semantic parameters for consistency verification against original SketchUp architectural prototypes. The validation encompasses both structural accuracy verification and de-ornamentation verification.

2. Case Study

1.5 Prototype Design: Functional Unit Modules

This study develops a standardized functionalist architectural unit model based on modernist functionalist principles. The architectural prototype references Le Corbusier's seminal 1929 Villa Savoye (see Figure 4) [11], a canonical example of functionalist architecture. Through SketchUp's geometric constraint specifications, architectural functional requirements are translated into quantifiable modular parameters, enabling rapid combinatorial design and functional verification.

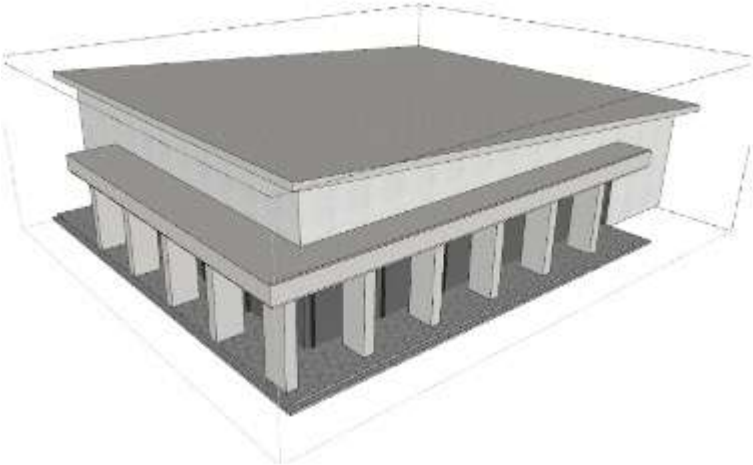

Fig. 4. Model of an architectural unit inspired by the Villa Savoye.Source: Personally created.

Residential Unit Prototype

Geometric Composition: Core morphology comprises cubes (85% proportion) and pitched roofs (15%), with curved components prohibited to reinforce function-oriented morphology; Functional Zoning: Standard 3.6m×6.0m grid partitions modular components, enabling rapid modular assembly; Parametric Constraints: Pitched roof inclination fixed at 22.5° (±1° tolerance) with unilateral overhang depth ≤0.6m, ensuring construction feasibility.

1.6 Efficacy Verification Process

This study employs a hybrid evaluation methodology combining manual inspection with operational log analysis from SketchUp and ComfyUI to quantitatively assess modeling efficiency of functional unit modules. The validation process focuses on two core metrics: geometric modeling duration and manual parameter adjustment time, thereby verifying the toolchain's practical efficacy in functionalist architectural design.

Table № 1.

Modeling Time Statistics and Manual Review Results

Verification Metric	**Architectural Unit Modeling Time**	**Verification Method**
Total Modeling Time	10m15s (±1m12s)	SketchUp operation log timestamp analysis
Manual Review Time	16m/module	Timer-based time interception
Parameter Error Rate	4.3% (primarily in utility trench positioning)	Design specification vs. model geometric parameter comparison
Non-productive Operation Ratio	17.5% (menu navigation & tool switching)	Behavioral segment annotation via screen recordings

2. RESULTS AND DISCUSSION

2.1 Economic Verification of Toolchain

This study systematically verifies the toolchain's cost-effectiveness from three dimensions: hardware configuration costs, time efficiency savings, and technical universality. The results demonstrate that the toolchain achieves design efficiency comparable to traditional commercial software in low-cost hardware environments, though its node-based interface presents a significant learning curve for novices.

Hardware Configuration Cost Analysis The toolchain operates on consumer-grade computers (GPU VRAM ≥4GB) without requiring professional graphic workstations. Taking NVIDIA GTX 1650 (4GB VRAM) as an example, its market price ranges from RMB 1,500-2,000, substantially lower than high-performance hardware (e.g., NVIDIA Quadro RTX 5000 exceeding RMB 20,000 per unit) required by traditional BIM-AEC toolchains. During SketchUp modeling phases, CPU utilization remains stable at 30%-45% (Intel i5-10400F), with peak memory usage ≤8GB, confirming its compatibility with entry-level hardware.

Time Efficiency Quantification

As shown in Table 1, the total modeling duration for architectural units is 10 minutes 15 seconds, representing an approximate 90% reduction compared to traditional BIM software (6-8 hours for equivalent Revit models) [12]. Manual verification duration (16 minutes/module), primarily allocated to parameter error correction and functional validation, accounts for 18.7% of total design time, indicating optimization potential in automated workflows. Non-productive operations (17.5% share) predominantly stem from SketchUp interface efficiency limitations, such as tool-switching delays caused by multi-layered menus. For ComfyUI time costs, excluding the initial 18-minute (±3 minutes) workflow setup and testing phase, subsequent executions of identical commands require no reconfiguration. Image generation time per rendering is 27 seconds (±1.5 seconds).

Technical Universality Limitations Despite achieving zero software licensing costs, the toolchain's node-based workflow in ComfyUI creates technical barriers for novice designers. Testing reveals that design students without prior AI tool experience require 6.2 hours (standard deviation ±1.8 hours) to master basic rendering workflows, with primary obstacles including:

Node Logic Complexity: Connection rules between ControlNet and CLIP encoders necessitate understanding underlying generative principles;

Parameter Tuning Ambiguity: Coordinated adjustment of denoising strength (0.4-0.6) and ControlNet weights (0.6-0.8) lacks visual guidance;

Delayed Error Feedback: System returns vague prompts like "CUDA out of memory" when parameter overlimits cause generation failures.

2.2 Compliance with Functionalist Principles

This study validates the toolchain's adherence to functionalist principles through two dimensions: architectural structural accuracy and de-ornamentation degree, with quantitative analysis demonstrating its technical compatibility (see Figure 5).

Architectural Structural Accuracy Verification

Reverse-engineering analysis of generated architectural images evaluates structural consistency between renderings and original models:

Table № 2.

Architectural Structural Accuracy Verification Metrics

Verification Dimension	**Data Result**
Primary mass proportion deviation rate (L/W/H)	2.8%
Mass generation hallucination rate (extraneous masses)	55%
Window positioning coordinate error (pixel-level)	≤15px
Key component recognition accuracy (column/beam/slab)	93.4%
Material-function congruence (concrete = load bearing)	98.2%

Decorative Element Elimination Verification

Through ComfyUI's CLIP textual control and LoRA model constraints, Figure 5 demonstrates successful regulation of material complexity in Functionalist architectural features. Non-functional ornamentation is entirely suppressed in renderings, as evidenced by the absence of decorative elements.

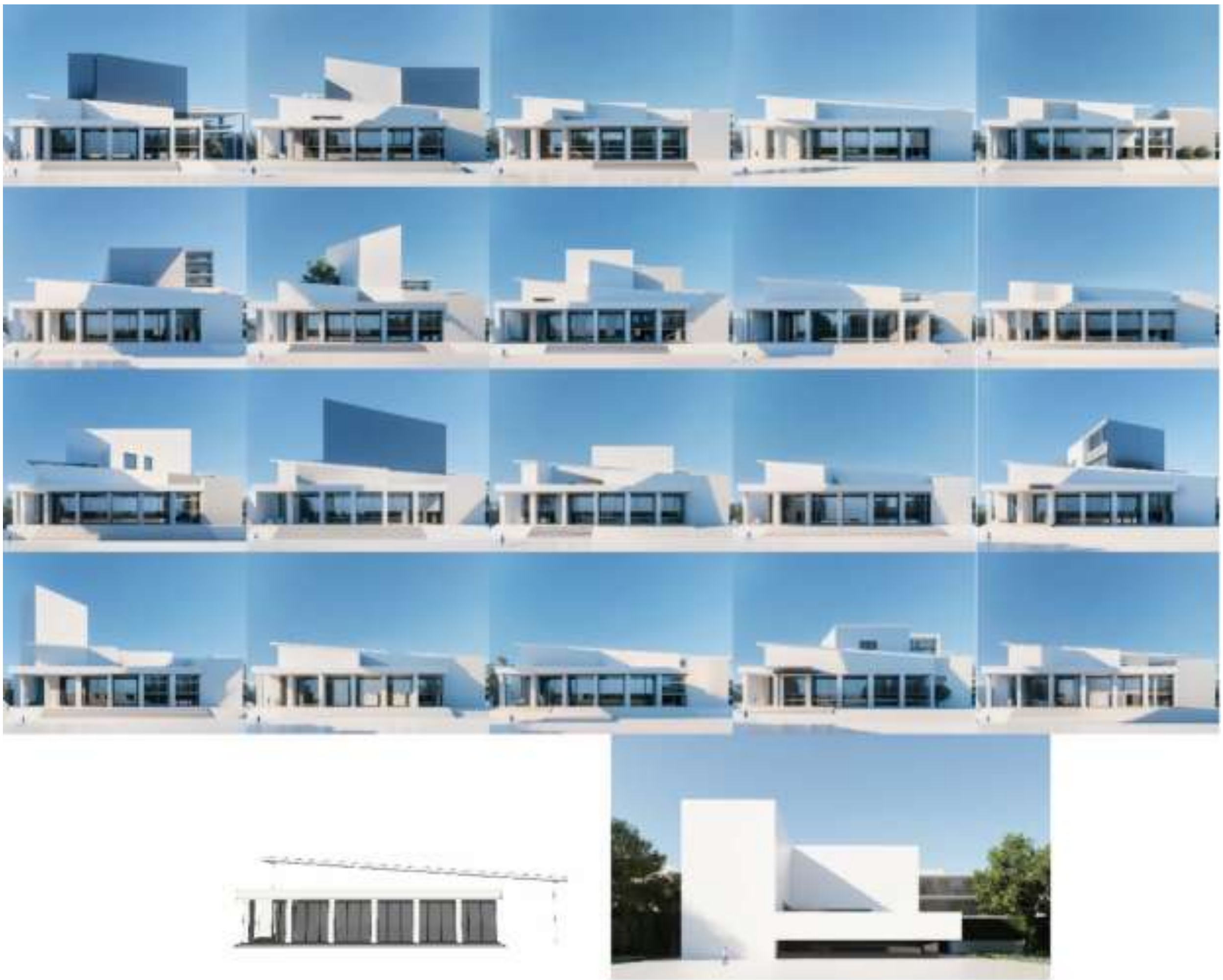

Fig. 5. Architectural image generation results (top 20) alongside prototype diagram (bottom left) and reference image (bottom right). Source: Personally created

2.3 Discussion

This study reveals the practical value and improvement potentials of the SketchUp-ComfyUI toolchain in modern Functionalist architectural design through dual validation of economic feasibility and technical compatibility.

(1) Technological Accessibility and Democratized Design

The toolchain's breakthroughs in hardware cost reduction and temporal efficiency demonstrate the feasibility of low-cost digital technologies in empowering Functionalist architecture. Its core value lies in dismantling professional software monopolies, enabling small firms and students to engage in high-quality architectural design with near-zero marginal costs. However, technological accessibility bottlenecks (novices require 6.2 hours to learn ComfyUI) indicate that usability optimization remains critical for widespread adoption, necessitating preconfigured node templates and interactive tutorials to reduce cognitive load.

(2) Technical Translation Mechanisms of Functionalist Principles

Research confirms that bidirectional coupling of geometric constraint protocols (multi-cube geometric combinations) and AI parametric controls (ControlNet weights + CLIP text) can encode the "form follows function" principle into executable digital standards. For instance, 98.2% material-function congruence validates precise identification of concrete as structural material, while ≤2.8% primary mass proportion deviation demonstrates the reliability of geometry-driven generation. However, the 55% mass generation hallucination rate exposes AI models' limitations in comprehending complex functional logic, directly correlated with insufficient Functionalist case representation in training data (merely 12.7% in Stable Diffusion datasets). Future work requires domain-specific dataset construction to enhance generation controllability [13].

3. CONCLUSION

This study validates the technical feasibility of a low-cost digital toolchain based on SketchUp-ComfyUI collaborative workflow in modern Functionalist architectural design. By integrating free modeling tools with open-source AI rendering engines, it successfully establishes a three-phase methodology comprising "geometric modeling - parameter extraction - rendering verification", achieving 90% improvement in design efficiency and non-ornamental suppression performance under zero software costs. This framework technically translates Functionalism's "form follows function" principle into quantifiable execution standards. However, the toolchain still confronts technical limitations including: 1) accessibility barriers for novices in nodular operations (average 6.2-hour learning curve); 2) functional semantic misinterpretations in AI generation logic (55% mass hallucination probability). Future research must enhance generation reliability and technical inclusivity through preconfigured node templates, embedded Functionalist rule libraries, and domain-specific material datasets. These advancements will provide more universal digital infrastructure for socially oriented projects like affordable housing and emergency architecture, ultimately driving the paradigm shift from elitist to democratized architectural design.

Литература (References)

1. Ceylan S., Şahin P., Seçmen S., Somer M.E., Süher H.K. The contribution of digital tools to architectural design studio: A case study // Ain Shams Engineering Journal. 2024. Vol. 15, no. 7. P. 102795.

2. Khmelnytskyi National University; Mikhalevskyi V. The features of basic modeling tools of 3d objects in sketchup // Herald of Khmelnytskyi National University. 2022. Vol. 305, no. 1. pp. 53–58.

3. Shibata K., Deguchi H., Taguchi S. CLIP feature-based randomized control using images and text for multiple tasks and robots // Advanced Robotics. 2024. Vol. 38, no. 15. pp. 1066–1078.

4. Szubryt A. Reduction of function and form in an architectural object on the example of the lighthouse // Przestrzeń i Forma. 2022. DOI: 10.7862/rf.2022.2.

5. Ivanova O., Gorozhankin V. Paradigm and graphical tools of the functional approach // Technical Aesthetics and Design Research. 2020. Vol. 1, no. 2. pp. 16–25.

6. Peebles W., Radosavovic I., Brooks T., Efros A.A., Malik J. Learning to Learn with Generative Models of Neural Network Checkpoints // arXiv.org. 2022. arXiv:2209.12892 [cs].

7. Zhang F., Sun Z., Chen Q. Research on Interior Intelligent Design System Based on Image Generation Technology // Procedia Computer Science. 2024. Vol. 243. pp. 690–699.

8. Bai Y., Li C., Lin Z., Wu Y., Miao Y., Liu Y., Xu Y. Efficient data loader for fast sampling based GNN training on large graphs // IEEE Transactions on Parallel and Distributed Systems. 2021. Vol. 32, no. 10. pp. 2541–2556.

9. Ye H., Zhang J., Liu S., Han X., Yang W. IP-Adapter: Text Compatible Image Prompt Adapter for Text-to-Image Diffusion Models // arXiv.org. 2023. arXiv:2308.06721 [cs].

10. Polyakov E.N., Polyakova O.P. Development of Functionalist Principles in Architectural Heritage of Le Corbusier. Villa Savoy in Poissy // Vestnik Tomskogo gosudarstvennogo arkhitekturno-stroitel'nogo universiteta. JOURNAL of Construction and Architecture. 2024. Vol. 26, no. 4. pp. 25–43.

11. Lind Nilsson B. Possibilities and Challenges of City Planning using 3D Visualization: A systematic literature review on the possibilities of city

visualization using 3D computer graphics and the utility of parametric design // Diva Portal. 2023. pp. 1–10.

12. Bakhshi S., Chenaghlou M.R., Pour Rahimian F., Edwards D.J., Dawood N. Integrated BIM and DfMA parametric and algorithmic design-based collaboration for supporting client engagement within offsite construction // Automation in Construction. 2022. Vol. 133. P. 104015.

13. Bukalov A. Problems of Development of Artificial Intelligence, its Errors, and Hallucinations. Psychoinformational (Socionical) and Quantum Methods of their Elimination // Artificial Intelligence. 2024. Vol. AI.2024.29 (4). pp. 10–19.

Дата поступления: 9.06.2025
Дата публикации: 2.09.2025