Proceedings of CONF-SEML 2026 Symposium: Computational Analysis and Modeling in Complex Intelligent Syste (CONF-SEML 2026)
DOI: 10.54254/2755-2721/2026.GU35001

The Application of Large Language Models in 3D Modeling: Current Workflows and Future Prospects

Haomin Gao

Interactive Media Art, New York University Shanghai, Shanghai, China
Hg3170@nyu.edu

Abstract. Demands for 3D contents across different industries are increasing such as gaming, architecture and visual arts. Recently with the development of Large Language Models (LLMs) new avenues have opened for automating and simplifying this process.This paper presents a review of current progress and applications of LLMs in 3D modeling. They can be basically categorized as two approaches: code-driven procedural 3D modeling and multimodal direct 3D synthesis. Code-driven methods like 3D-GPT and BlenderLLM, uses LLMs to translate natural language into executable scripts, which enables automated scene generation and asset editing. However these methods often lack spatial awareness. Despite significant progress, challenges such as execution errors and limited spatial reasoning remain. Finally, this paper explores future prospects, highlighting the potential of Vision-Language-Action (VLA) feedback loops, native 3D tokenization, and the shift towards synergistic human-AI co-creation workflows. These advancements may greatly improve the efficiency, accessibility, and creativity of future 3D content generation systems.

Keywords: Large language models, 3d modeling, procedural generation, multimodal synthesis, human-AI co-creation

1. Introduction

3D modeling software in computer graphics such as Blender has become irreplaceable in multiple industries, like game development, architectural visualization and visual art. Although it has extensive capabilities and robust toolsets that is able to model everything from reality or virtual, traditional manual 3D modeling is a labor-intensive working process and costs lots of time and effort. Besides, learning multiple kinds of tools in Blender takes a long time for beginners to start with. At this point with the rising need for 3D modeling in multiple fields, there are more needs in finding a way to simplify 3D modeling process, reducing human labor and lowering the barrier for beginners.

In recent years, the rapid development of Large Language Models (LLMs)—such as ChatGPT, Gemini, and Grok—has demonstrated remarkable capabilities in natural language understanding, complex mathematical reasoning, and code generation [1-3]. As a result, some researchers had begun working on exploring using Large Language models in 3D modeling and Computer-aided Design aspect. There are several existing outcomes that had completed a try by using LLM's text to code generation. For example, the BlenderLLM translates user's text prompts into python codes that

DOI: 10.54254/2755-2721/2026.GU35001

Blender can read, which then executes basic geometric operations inside Blender. These kinds of assisting tools allow users to create simple shapes and manage basic scene hierarchies through natural language instructions, which offers a promising chance in automatic 3D modeling.

However, though there are some inspiring outcomes and advancements in AI-assisted 3D modeling aspect, there are still significant limitations still remains. Current Large Language Models are actually text-based, they are unable to understand the three-dimensional world and executed 3D modeling directly. Most existing methods are based on basic CAD tasks and are unable to automatically execute complex geometric manipulations and create surface materials. Furthermore, directly using a Large Language Model to generate a Python script often leads to execution errors or misleading visual outputs. Besides lacking spatial awareness and the absence of iterative, vision-based feedback mechanisms during the script execution phase. It is still unclear about the mechanism it needs to transform complicated requirements and intension from user into precise models that can be applied to usage.

2. Current application of large language models on 3D modeling

The application of Large Language Models into 3D modeling has reached far more beyond simple text-to-shape generation, it had been into complicated multi-agent frameworks. These frameworks are able to interact with professional APIs. Based on current existing method, this essay introduces seven iconic in most distinct works. These works can be separated into code-based code-driven procedural modeling and multimodal direct 3D synthesis.

2.1. Code-driven procedural 3D modeling

This method uses LLMs' ability of generating code to produce whole scripts like python. These scripts directly invoke the Application Programming Interfaces (APIs) of traditional 3D software, like Blender or 3D max to make the 3D modeling process partly or even completely automatic.

2.1.1. 3D-GPT: multi-agent procedural generation

In 2023, Sun et al. proposed 3D-GPT. This is an advanced framework that makes LLMs into a problem-solving agent for instruction-driven 3D modeling [4]. Instead of directly trying to generate complex 3D mesh data from scratch, 3D-GPT takes down and the overarching modeling task and decomposes into smaller and manageable small, separated tasks. Its workflow relies on multi-agent system comprising a task-dispatch agent, a conceptualization agent, and a modeling agent. For example, when users input a prompt in human language, the conceptualization agent first starts to edit the prompt, making it more detailed, then translating vague concepts into specific visual parameters. The modeling agent consults the official documentation of procedural generators (like Blender's geometry nodes) to infer the correct function calls and parameters. In the end it will generate a python script that can be read and run by the program. The most significant innovation of 3D-GPT is its ability that uses existing procedural generators through language, which ensures high geometric accuracy and topological correctness. Besides, it can reduce the human effort required to complex modeling at the same time.

2.1.2. LL3M: agentic collaboration for editable assets

In 2025, Lu proposed LL3M [5], this is a multi-agent collaboration system that changes the process of traditional 3D modeling. In its process, when it receives a text prompt, the model will form a

DOI: 10.54254/2755-2721/2026.GU35001

"specialized team" of different LLM agents. The most significant innovation of LL3M is its emphasis on modularity and asset editability. Since the entire output is code-based, it deeply relies on Blender's native B-mesh structures and shader node networks. This means what it generates is now only abstract meshes, they're readable content for human that can edit and put into use in multiple scenes.

2.1.3. Self-optimized LLMs for CAD and blender (BlenderLLM)

Though generally used LLMs like GPT-4 and Gemini are powerful in most aspects, they often suffer from syntax hallucinations when engaging in these highly specific APIs like Blender's bpy module. To deal with this problem, some research are focusing on fine tuning in specific domains. For example, the BlenderLLM project and related studies utilize custom instruction-tuning datasets like the BlendNet to assist trian models for Computer-Aided Design (CAD) [6]. Its workflow consists of collecting thousands of human instructions and corresponding error-free Blender Python scripts. It uses a self-optimizing loop that will be rewarded for generating scripts that compile without errors. This special LLMs has a significantly higher success rate in script generation. This approach highlights the necessity of domain-specific fine-tuning to overcome the instability inherent in using generalized models for precise engineering and design tasks.

2.1.4. LLM-driven large-scale scene generation

Except for individual objects, LLMs are also being used to generate a whole 3D environment. Frameworks like Holodeck use a modular LLM pipeline to construct fully interactive virtual environment inside game engine and blender. This method contains 3 steps including spatial planning, asset retrieval, and layout execution. This kind of LLMs act as an interior designer, it tries to understand the spatial relationship, how the room layout is design, etc. Then it uses a large 3D library to find suitable models and generates the code to locate all assists within the entire 3D space. Its key innovation is to use Retrieval-Augmented Generation (RAG) combined with spatial reasoning constraints, which makes the outcome from LLM not only makes sense in visual, but also in human logic.

2.2. Multimodal LLMs and direct 3D synthesis

Though the previous code-driven approach is highly editable, they are still litited by APIs they're underlying. As a result, researchers are expanding LLM architecture to process and understand 3D spatial representations in a more native way.

2.2.1. 3D-LLM

In 2023, Hong invented 3D-LLM, aiming to endow language models with a direct cognitive understanding of the physical 3D world [7]. Traditional LLMs are more based on 2D vision and lack intrinsic spatial perception. This 3D-LLM achieves 3D vision by using 3D point clouds and their extracted features as direct neural inputs. Its framework can extract 3D features and then put these features together with textual sentences and 3D coordination together. This new kind of workflow gives the model an ability to work on highly complicated tasks with understanding to spatial understand. This advance means that Artificial Intelligence now have the ability to really "feel" and "perceive" the native modality.

DOI: 10.54254/2755-2721/2026.GU35001

2.2.2. PointLLM: fine-grained understanding of complex geometries

To wrap up all 3D data, PointLLM introduced multimodal Large Language Models [8]. This kind of LLM is specialized in color processing without relying on 2D image rendering. In traditional 3D processing, previous tools tend to change 3D models into 2D images and then input these into Artificial Intelligence. This complicated process often causes loss of information and depth ambiguity. PointLLM used a point encoder to extract geometric and color structure to solve this problem. Then, the tool projects this data into the latest space of Large Language Models. The biggest innovation of which is its ability to understand object types and then capture geometrical details.

2.2.3. Text2Room: extracting textured 3D meshes from 2D models

In 2023, Höllein created Text2Room [9]. This system provided a new method to generate textured 3D meshes as large as a room scale through descriptive text. The outcome not only has a complete 3D model but also has meshes. The system uses lots of pre-trained 2D text-to-image generated content, it syncs the object from multiple simulated viewpoints. The Text2Room uses monocular depth estimation and continuous alignment just to transform these separated and flat 2D images into a single 3D space. The system uses images to fill places where the 2D camera can't be seen and then integrate these with previous viewpoints. Its view selection and depth-fusion techniques allow users to create a complete and seamless virtual environment.

3. Future prospects

3.1. Multimodal Vision-Language-Action (VLA) feedback loops

The most promising direction in AI-assisted 3D modeling is to mix Vision-Language Models (VLMs) into the code execution loop [10]. In this future workflow, after the LLM generates a Blender script, the software renders the final 3D viewport outcome. The VLM then acts as an automated director, analyzing the rendered image to detect visual anomalies. This visual critique is translated back into text prompts, allowing the LLM to rewrite and refine the generated script. This new type of workflow can significantly increase the quality of AI-generated 3D contents.

3.2. Native 3D tokenization and unified architectures

There's a limit for content window limitations when AI generates Python scripts. To bypass this limitation, AI models in the future may even give up code generation and turn to native 3D tokenization. By tokenizing 3D structures into a whole content including text and images, LLMs will be able to process and generate 3D spaces naturally as they generate words. This new chain can bypass API constraints and generate high-quality scenes.

3.3. Human-AI co-creation workflows

When Artificial Intelligence keep making progress, these mentioned will dramatically change their workflow but will not take the place of 3D artists. Future prospects include deep integration of LLMs into node-based editors. This method will use natural language to control the AI to transform complicated tasks, they won't request an entire finished model. This kind of co-creation paradigm, it

DOI: 10.54254/2755-2721/2026.GU35001

will keep artistic control in user's hand while utilizing the LLM for rapid prototyping and labor reduction.

4. Conclusion

This paper reviewed the recent development of Large Language Models in the field of 3D modeling, including both code-driven procedural modeling and multimodal direct 3D synthesis approaches. Current studies have shown that LLMs can simplify many traditional modeling tasks through natural language interaction, automated script generation, and intelligent scene construction. Methods such as 3D-GPT, BlenderLLM, and PointLLM demonstrate the strong potential of combining language understanding with 3D content creation. In the future, the integration of multimodal feedback mechanisms, native 3D tokenization, and human-AI collaborative workflows may further improve the quality and efficiency of 3D modeling systems.

References

[1] Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., ... & McGrew, B. (2023). Gpt-4 technical report. arXiv preprint arXiv: 2303.08774.
[2] Team, G., Anil, R., Borgeaud, S., Alayrac, J. B., Yu, J., Soricut, R., ... & Blanco, L. (2023). Gemini: a family of highly capable multimodal models. arXiv preprint arXiv: 2312.11805.
[3] Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H. P. D. O., Kaplan, J., ... & Zaremba, W. (2021). Evaluating large language models trained on code. arXiv preprint arXiv: 2107.03374.
[4] Sun, C., Han, J., Deng, W., Wang, X., Qin, Z., & Gould, S. (2025, March). 3d-gpt: Procedural 3d modeling with large language models. In 2025 International Conference on 3D Vision (3DV) (pp. 1253-1263). IEEE.
[5] Lu, S., Chen, G., Dinh, N. A., Lang, I., Holtzman, A., & Hanocka, R. (2025). Ll3m: Large language 3d modelers. arXiv preprint arXiv: 2508.08228.
[6] Du, Y., Chen, S., Zan, W., Li, P., Wang, M., Song, D., ... & Wang, B. (2024). Blenderllm: Training large language models for computer-aided design with self-improvement. arXiv preprint arXiv: 2412.14203.
[7] Hong, Y., Zhen, H., Chen, P., Zheng, S., Du, Y., Chen, Z., & Gan, C. (2023). 3d-llm: Injecting the 3d world into large language models. Advances in Neural Information Processing Systems, 36, 20482-20494.
[8] Xu, R., Wang, X., Wang, T., Chen, Y., Pang, J., & Lin, D. (2024). Pointllm: Empowering large language models to understand point clouds. In European Conference on Computer Vision (pp. 131-147). Cham: Springer Nature Switzerland.
[9] Höllein, L., Cao, A., Owens, A., Johnson, J., & Nießner, M. (2023). Text2room: Extracting textured 3d meshes from 2d text-to-image models. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 7909-7920).
[10] Ji, H., Aditya, K., Escalante, S., & Qiu, Y. (2026). CodeGen-3D: A Benchmark for Evaluating LLMs in Zero-Shot and Iterative 3D Modeling in Blender. IEEE Access.