Proceedings of CONF-SEML 2026 Symposium: Computational Analysis and Modeling in Complex Intelligent Systems
DOI: 10.54254/2755-2721/2026.GU34603

Text-to-3D Scene Generation Using Generative AI

Donghui Li

Department of Mathematics, University of California, Berkeley, USA
dli77777777777777@berkeley.edu

Abstract. The text-to-3D scene generation is one of the main directions in the study of generative artificial intelligence. It involves natural language processing, visual synthesis, geometric modeling and interactive virtual environments. Conventional development of 3D content usually demands specific modeling capabilities and high-cost software packages, as well as labor-intensive manual procedures and substantial time expenditure. Conversely, the new generative AI methods lower the costs of conventional 3D creation by enabling the user to just specify the object, room or environment that he/she wants in natural language. User instructions can be transformed into 3D objects, textured models, lighting or whole virtual scenes. A text-to-3D system enables a user to describe a scene such as a small bedroom with a desk, a bed and warm light and the AI can create a 3D version of it. This paper presents some of the most important techniques of text-to-3D generation such as diffusion models, neural radiance fields, 3D Gaussian point painting, mesh generation, and scene planning using large language models. This paper also mention the uses of these approaches in virtual reality, gaming, education and design. Despite the fact that the existing technology of 3D generation has enormous potential, there are many challenges such as lack of spatial consistency, unrealistic positioning of objects, lack of control over the users, low quality of assets and evaluation difficulties. The future text-to-3D systems could integrate generative AI, 3D engines, physics simulation, and manual editing to develop more realistic and practical virtual worlds.

Keywords: text prompt 3D generation, generative AI, diffusion models, Neural Radiance Fields, 3D Gaussian Splatting

1. Introduction

3D content is widely used in many aspects of the lives, such as games, virtual reality and augmented reality, movies, architecture, robot simulation, education, industrial design, etc. However, it is tedious to create high-quality 3D scenes manually. The creators need to master a large number of complex tools like Blender, Maya, Unity, Unreal Engine, etc. They have to model objects, add textures, layout the scene, arrange lights, ensure the consistency of the scene, etc. This process also requires a lot of creativity [1]. Text-to-3D scene generation aims to make this process easier [1]. For example, to create a bedroom containing a desk, bed, lamp, and window, only a few textual descriptions of the required scene are needed. This paper would like to create a bedroom with a desk, bed, lamp, and window. Then, it is possible to generate a 3D scene based on these

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2026.GU34603

descriptions. It will save time for users to learn how to use various professional 3D software packages. Users can be technical modelers or senior creative directors without learning any 3D modeling skills. There are many works on text-to-image generation [2] that can help us to understand how to generate 3D scenes from texts. Recently, some researchers proposed to train diffusion models on image-text pairs and then generated images from natural language prompts [1]. In addition, they also tried to generate 3D scenes from natural language prompts [3]. Compared with text-to-image generation, generating 3D scenes is more challenging because it is required to ensure that the generated 3D scenes are consistent with the input texts. A good 3D scene should not only look good but also have good geometry, depth, textures, object sizes, and spatial relations [4, 5]. The chair should not be too big or too small. It should also have appropriate colors and shapes. The space between the chairs and the walls should also be suitable. The floor should not be too dark or too bright. All these details should be considered carefully. To achieve this goal, there are several methods that can be used. Some people try to optimize their results by using 2D diffusion models as visual priors [6, 7]. Others try to optimize their results by using 3D representations [8, 9]. DreamFusion [1] demonstrates that pre-trained text-to-image diffusion models can be combined with neural radiance fields (NeRFs) [10] to improve 3D generation quality.

2. Related work

2.1. General process of text-to-3D scene generation

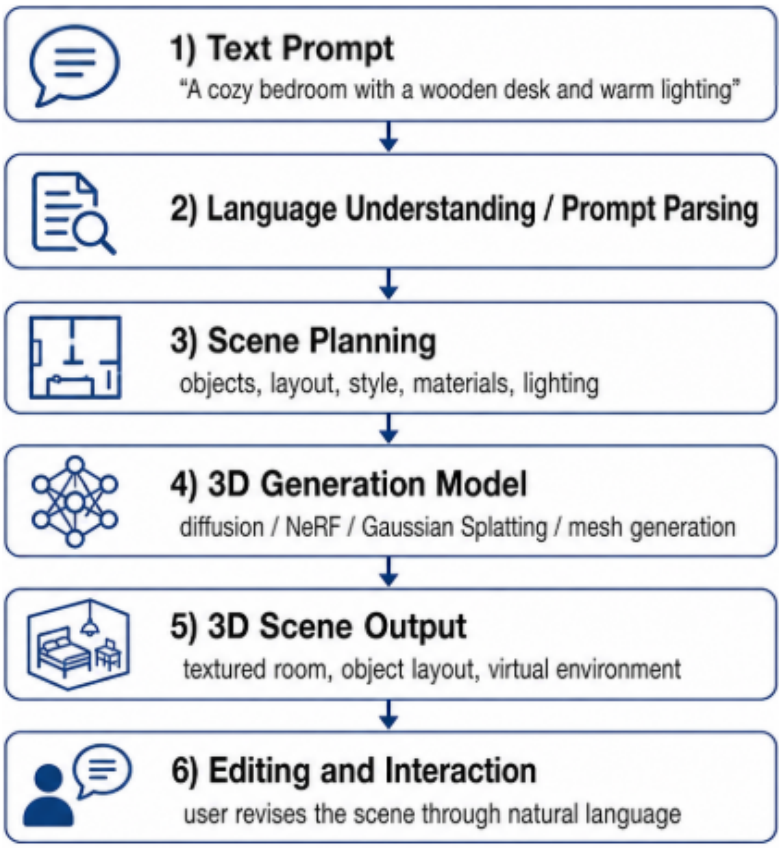

Figure 1. General workflow of text-to-3D scene generation using generative AI (picture credit: original)

DOI: 10.54254/2755-2721/2026.GU34603

The first and the most significant step in the text-to-3D generation process is the user prompt. The system gets commands by the user on what to generate. As an example, a prompt can be as follows: A small classroom with a desk, a chair, a whiteboard, and bright lighting. The system then processes the prompt. It initially identifies the style and setting, like a small classroom and bright lighting. After that, the system creates the scene plan. The plan may consist of one or more objects, their spatial relations, and the materials used [5-7]. Indicatively, the chair has to be placed close to the desk. The whiteboard must be hung on the wall. The light should be installed above the room. This is a critical phase since the 3D scene does not exist as a set of individual objects; it is an environment where these objects are supposed to cooperate [5]. The system then creates the scene through a 3D generation model. Depending on the method, the output could be a mesh, a NeRF, a Gaussian point cloud scene, or any other 3D representation [1-4]. Lastly, the user has the option of editing the output. As an example, the user might instruct the system to shift the desk, alter the color of the bed, or enhance the brightness of the room [6, 7] as shown in Figure 1.

2.2. Diffusion models in text-to-3D generation

The generative AI has a diffusion model as one of the most valuable tools [1]. It is already being applied in text-to-image generation [1]. A diffusion process is a noise-based image that is slowly transformed into a useful image. The prompt will be used to direct the model and the resultant image will reflect the description by the user [1]. Diffusion models can be employed in generating 3D images since they are able to perceive the visual style and object appearance [1]. Nevertheless, it is generally not possible to produce a three-dimensional object using diffusion models [1]. To address this issue, researchers apply diffusion models to direct the construction of a 3D form [1]. In other words, the machine learning system constructs an object in 3 dimensions and then takes photos of the object with various cameras and compares these photos to the text instructions [1]. There is one case in point, namely DreamFusion. DreamFusion is a pretrained text-to-image diffusion model that helps optimize the shape of a 3D object [1]. The system does not require training on a large set of labelled 3D models [1]. This is accomplished through the information provided in the two dimensional diffusion model [1]. This is also helpful because there are more image-text data than 3d data. It is the benefit of this technique that even when 3D training data is limited, text-to-3D creation is feasible [1]. But it is also problematic. The 3D structure is not necessarily accurate since the model was trained based on the 2D images [1]. Sometimes it appears nice at a certain perspective but bizarre at another side. This is an issue that arises during the process of text-to-3D generation.

2.3. Neural radiance fields

Neural Radiance Fields, or NeRFs [1] are also applicable in 3D generation. The neural network is used to encode a scene of 3D [1]. It does not save the scene as a mesh which is common. Rather, it is taught to learn what colour and light ought to appear at various points in space in 3D [1]. When the camera rotates the NeRF has the ability to render the same image with the new position of the camera [1]. NeRFs are helpful since they can create realistic images of a given situation [1]. They have an excellent performance in novel view synthesis, i.e., the process of producing images out of camera positions that were not included before [1]. NeRFs may be tuned in text-to-3D generation to fit the rendered views to the user text prompt [1]. However, there are no ideal NeRFs. One of them is their sluggishness in training and rendering [2]. Another one is that they cannot be easily modified [2]. In conventional 3D software, a chair object can be directly selected, repositioned, resized, and textured through interactive editing tools.This type of direct editing is not easy with a NeRF because

DOI: 10.54254/2755-2721/2026.GU34603

it is represented by a neural network [2]. This is why NeRFs are so good when it comes to photo-realistic work but are less useful when it comes to the daily life of 3D design. Such restriction implies that certain subsequent systems attempt to transform the results of the NeRF into meshes, or any other 3D representation that is easier to edit and control by users [2].

2.4. Mesh-based generation

The traditional computer graphics has the most common 3D representation, which is meshes. Vertices, edges and faces constitute a mesh. The majority of 3D modelling programs and game engines employ meshes due to their simplicity in rendering, editing, animating and exporting. In-text-to-3D generation and mesh-based output are significant since they are more practical [2]. When an AI system produces a chair as a mesh, it can be imported into Blender, Unity or Unreal Engine. They are able to modify its size, alter its material or put it within a bigger scene. One such system that enhances text-to-3D generation is Magic3D, which generates better quality 3D mesh models [2]. It is a two-stage process [2]. Initially, it forms a rough 3D shape [2]. Then it refines the form and texture during the second stage [2]. This renders the end product more intricate and functional [2]. Nevertheless, there are issues with mesh generation. A generated mesh can have broken geometry, messy surfaces or strange textures. It can appear nice on a preview picture, but will not be easy to implement in a real game or VR application. To use it professionally, the mesh must be clean, structured and editable shown in Figure 2.

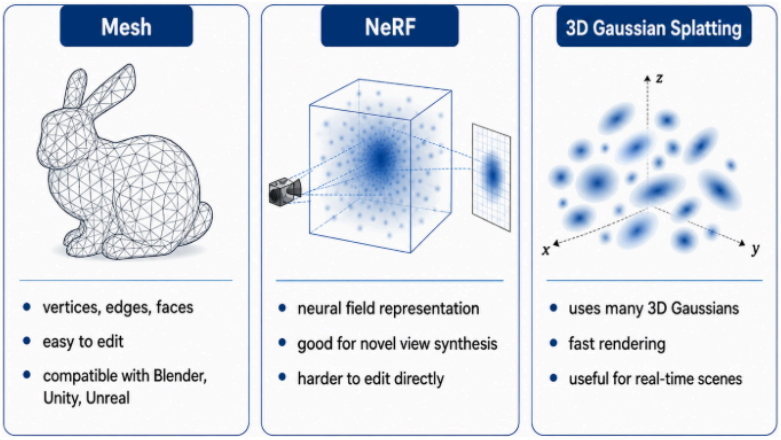

Figure 2. Comparison of common 3D representations used in text-to-3D generation (picture credit: original)

2.5. 3D gaussian splatting

Gaussian splatting is a rather recent method of modeling 3D scenes [4]. The approach does not rely on mesh-based models or neural field approaches to represent a scene but instead uses numerous small points in the form of 3D Gaussians [4]. Information such as position, color, size and transparency are stored by each Gaussian [4]. They can be combined and rendered to produce a realistic picture of a three-dimensional space [4]. It is evident that 3D Gaussian splatting is fast to render [4]. Its ability to create the view of a 3D object in a short time makes it suitable to use in real-time situations like virtual and augmented reality and interactive 3D editing [4]. This has therefore made it one of the significant ways in the modern generation of 3D research [4]. Text-to-3D generation is promising in the case of Gaussian splatting because people will often need to create and edit their texts in real time. As an illustration, a person could request the computer to create a

DOI: 10.54254/2755-2721/2026.GU34603

living room and then say, make the sofa larger, add a bookshelf behind the table, etc. Such a kind of interaction is easier and more efficient with the help of a quicker 3D representation [4]. However, there are certain drawbacks associated with the Gaussian splatting. One specific thing in the scene is not easy to edit. All objects are represented using a large number of Gaussian points which may require other tools to isolate them, alter the materials, or only part of the scene. The future of Gaussian splatting may be connected with scene graphs or LLM planning and allow generated scenes to be manipulated and edited [4-7].

2.6. Large language models as scene planners

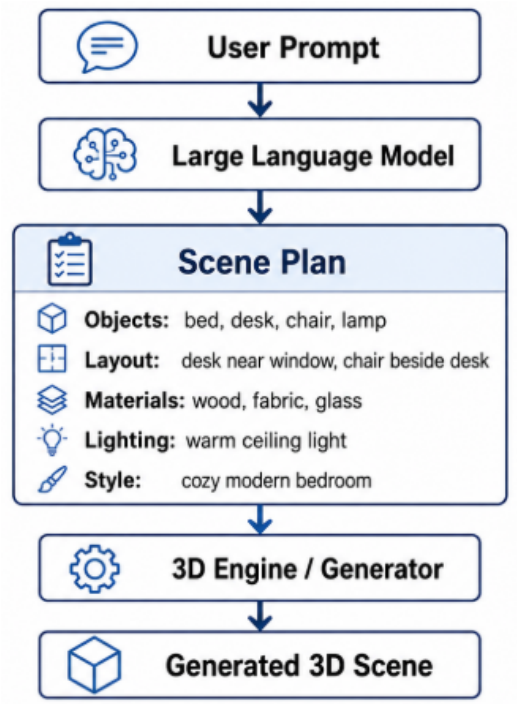

Figure 3. Role of large language models as high-level planners in 3D scene generation (picture credit: original)

The large language models or LLMs are artificial intelligence systems that can be trained to recognize and produce human languages. Some of the examples of such models are ChatGPT. LLMs in 3D scene generation are helpful since they have the ability to comprehend complex prompts and arrange information [5-7]. The final 3D geometry is not necessarily created by an LLM. It could serve as a planner instead [5-7]. In case the user writes, this paper would like to make a modern bedroom, which should be cozy, then the LLM may understand that the room has to contain a bed, a desk, a chair, a lamp, a window, a rug and maybe decorations. It also can deduce the connections: the desk might be at the side of the window, the chair must be close to the desk, and the lamp might be next to the bed [5]. This is so because most of the text prompts lack all the details. People tend to omit what they know because it is assumed to be common sense. When one requests a kitchen, he/she tends to assume a sink, stove, cabinets, and refrigerator although he/she does not mention them all. These missing details can be filled with the help of LLMs [5]. There are interactive 3D environments generated by using language models, e.g., Holodeck [5]. These systems will not only generate a single object but will generate whole spaces, including apartments, offices, museums, or

DOI: 10.54254/2755-2721/2026.GU34603

classrooms [5]. LLM assists in the floor plan, choice of objects, materials used in rooms and objects arrangement [5] shown in Figure 3.

3. Applications of text-to-3D scene generation

3.1. Games and virtual worlds

One obvious use of text-to-3D generation is in the development of games. The worlds of games are generally characterized by numerous objects, buildings, rooms, landscapes and minor decorative elements. This can be a very time consuming and tedious process to developers when done manually. Text-to-3D systems may assist with this by rapidly generating rough drafts of game environments. Indicatively, a designer may write: Make a medieval village with little wooden houses, a market place, and a stone road. Then AI could make an approximate version of that scene, which the designer would be able to improve on later. It does not mean that AI will replace the job of game artists. Rather, it is more of an assistant which assists artists to build early prototypes quicker. Independent game developers who do not have big art teams might also find text-to-3D generation useful. A small group might employ AI to create background objects, play around with visual styles or try out various scene concepts before selecting one. LLM-based tools are particularly effective in this regard as they are capable of linking natural language prompts with 3D engines, allowing quicker prototyping [6, 7].

3.2. Virtual reality and education

Besides, the text-to-3D generation may be used to create a virtual reality teaching. VR is also very effective as it will help students learn in a more engaging way. As an example, they could even take a walk around a 3D Roman city and not just read a book about ancient Rome. Rather than simply observing a diagram of the biological system, he or she can step into a cell and observe its components directly. However, most of the time, creation of educational settings in VR is extremely difficult. Most teachers do not have the capacity or ability to produce a 3D scene that is complex on their own. It can be made easier by text-to-3D production. The teacher would only have to specify the type of environment of learning he/she requires, and AI would be able to generate a prototype. A science lab with tables and microscopes and safety posters and bottles of chemicals, wrote a teacher. The AI would be able to create a classroom or a laboratory which the teacher would then modify to suit the lesson. This is particularly pertinent because the Holodeck is a system designed to provide language based descriptions of embodied interactive 3D environments [5]. It is almost related to procedural content generation. In both cases, the amount of manual work involved in producing the virtual contents is reduced. The significant distinction is that the process of generating 3D images from natural language is taken as input; hence, it might be less complicated to the users who are non-technical [5-7].

3.3. Architecture and interior design

The architecture and the interior design can also be assisted by text-to-3D generation. The designers usually require to have several copies of a room in order to select one of them. It would be possible to test various styles with generative AI, which is much faster. In one case, the user may request a living room that has white walls, grey sofa, wooden table and large windows. They could then tell the system to alter it, e.g., change the color of the sofa to blue, add a bookshelf, etc. This natural

DOI: 10.54254/2755-2721/2026.GU34603

language based editing process can be more efficient as well as easier. Text2Room is an example of such systems that are used in this application since they are aimed at creating full indoor scenes instead of individual objects [3]. But the accuracy of the architecture is required. A produced space should be in accordance with the real dimensions and construction regulations. Thus, the use of the text-to-3D software in the field of architecture is more accurate than the creative image-based software.

3.4. Robotics and simulation

Robotics is another key application domain. Robots must be trained and tested in a variety of environments; for instance, a home robot may have to learn to navigate kitchens, bedrooms, and living rooms. Constructing such simulation environments by hand can be time-consuming. Text-to-3D generation could aid in rapidly generating a large number of training environments. A researcher could generate a variety of room layouts filled with objects and clutter. This would lead to more varied robot training. Holodeck also follows this direction by generating environments for embodied AI in which an agent can interact with the scene [5]. However, for robotics, the scene must be physically plausible; objects cannot be floating or overlapping, and the robot must be navigable through space. This implies that text-to-3D systems for robotics necessitates strong spatial reasoning and physical constraints.

4. Discussion: Challenges and limitations

Despite the potential of text-to-3D scene generation, it has numerous limitations. These issues are significant as a 3D scene should not only be visually appealing but also be consistent, functional, editable and realistic.

4.1. Spatial consistency

Spatial consistency refers to the ability of the objects not to change and appear logical in different perspectives. It is a challenging task since 2D image models are used by most text-to-3D systems [1, 3]. A 2D model can be able to produce images that look well but may not know what the real 3D structure is. As an illustration, one chair might be normal when viewed on the front but damaged when viewed at the side. One window can be seen in a room but another window cannot be seen in the same room. The final 3D scene becomes unreliable due to these issues.

4.2. Physical plausibility

The scene should be physically plausible. The objects must not float in the air (unless it is explicitly requested in the prompt). Objects must not be intersected with each other (e.g., furniture cannot overlap). Doors must not be blocked by any object. A bed can only be placed on the ground and a lamp on a table or ceiling. This is particularly crucial for 3D scenes, such as Text2Room [5] and Holodeck [3], since the user or an AI agent may have to move through the space.

4.3. Controllability

The users would normally desire more control of the produced output. An easy instruction can produce a simple scenario, but it might not give the precise configuration that the user wants. As an example, the user may prefer the bed against the left wall, the chair facing the desk and the desk by

DOI: 10.54254/2755-2721/2026.GU34603

the window. Some of these instructions are obeyed by existing AI systems but not others. This inconsistency is a problem in professional design. Future systems will require improved editing tools whereby individuals will be able to edit the errors made by the AI. In this direction, there are LLM-based models like 3D-GPT and SceneCraft which employ language as a procedural modeling interface and scene manipulation [6, 7].

4.4. Asset quality

It is required to produce 3D objects with high quality of assets. A scene can be beautiful when it is far away and the viewer approaches closer to it, they may notice that the object is not well rendered, with a texture or jumbled geometry. As a result, this makes the situation less applicable in practice. To take an example, a chair will be made generally as a chair, but its mesh could be over complicated, non-uniform, or damaged. At times the texture can appear to be stretched, blurred or even unrealistic. These restrictions reduce the application of the final product on design work, virtual reality worlds, games. The problem can be addressed by using mesh-based solutions like Magic3D which generate 3D material with more texture and resolution [2].

4.5. Evaluation

Evaluation is also challenging. In the case of text-to-image generation, humans can judge if an image is good and satisfies the task. A 3D scene is more complex - a high-quality 3D environment has to be relevant to the problem, realistic, geometrically accurate, consistent from different viewpoints, and interactive. There is no single ideal metric for evaluating text-to-3D scenes. Human evaluation, despite being subjective and labor-intensive, is useful. The question of automatic evaluation remains open shown in Figure 4.



Figure 4. Major challenges in current text-to-3D scene generation systems (picture credit: original)

5. Future directions

The text-to-3D scene development in the future will probably be based on a combination of different methods. The most advanced visual generation is diffusion models [1]. NeRFs and Gaussian Splatting are applicable to 3D rendering [1, 4]. Manufacturing and editing are done using meshes [2]. LLMs could prove useful in scenario planning and prompt understanding [5-7]. All these pieces

DOI: 10.54254/2755-2721/2026.GU34603

can be significant in a good system. Multimodal AI may be enhanced as one way forward. A multi-modal model has the ability to interpret text, images, depth, point clouds, models and layouts of scenes. This shall enhance the 3D spatial reasoning capability of the system. Another possibility is human-in-the-loop editing. The user can communicate with the AI step by step, instead of requesting it to offer an ideal solution at once. The user builds a room and then requests the AI to reorganize, switch materials, decorate or correct errors. This is more realistic because human designers also make changes to their work several times. The practicality of this method is increased by the fact that LLM-based systems allow users to modify scenes with natural language [6, 7]. The third alternative is closer connection with 3D software. When text-to-3D systems are able to directly plug into Blender, Unity, and Unreal Engine, they will become more practical in the real world. The output must be editable, compatible with existing tools and appear good. SceneCraft and 3D-GPT are examples of such strategies which integrate language models with generative 3D modeling or Blender-like scene generation [6, 7]. Constraints of physics and layout should eventually be incorporated into future systems. This would minimize the probability of blocked passageways, overlapping furniture and floating things. However, physical realism does have its place in practical uses such as VR and robotics [5].

6. Conclusion

Text-to-3D scene generation is one of the most recent breakthroughs in generative AI. The 3D environments and objects are built by the user with the help of natural-language commands. This might be a way to reduce the level of entry to non-expert modelers to produce 3D materials. It also helps academics, lecturers, game designers and designers to create virtual spaces faster. A number of key methods of text-to-3D development have been studied such as diffusion models, NeRFs, mesh-based generation, 3D Gaussian Splatting, and large-language-model design. There are advantages to all the approaches. Visual knowledge is abundant in diffusion models. The ability to view in 3D is possible through NeRFs . Meshes are suitable to edit and make. Splats of Gaussian allow fast rendering . Cue interpretation and scenario preparation can be aided by LLMs. However, there exist some problems in the existing technology of text-to-3D. They are usually difficult to measure, control, realistic in terms of physics, spatially consistent, and quality of assets. These challenges become especially crucial when it comes to creating sceneries that are used in the gaming industry, virtual reality, education, architecture or robotics. Text-to-3D generation has an infinite potential at the end of the day. Although it cannot replace the 3D artists who work professionally or human artist, it can be a useful tool. The future direction would involve using natural language as a tool to design and edit the virtual environment so that it will be more accessible, rapid and interactive to use in 3D creation.

References

[1] Poole, B., Jain, A., Barron, J. T., & Mildenhall, B. (2022). Dreamfusion: Text-to-3d using 2d diffusion. arXiv preprint arXiv: 2209.14988.

[2] Lin, C. H., Gao, J., Tang, L., Takikawa, T., Zeng, X., Huang, X., ... & Lin, T. Y. (2023). Magic3d: High-resolution text-to-3d content creation. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 300-309).

[3] 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).

[4] Kerbl, B., Kopanas, G., Leimkühler, T., & Drettakis, G. (2023). 3d gaussian splatting for real-time radiance field rendering. ACM Trans. Graph., 42(4), 139-1.

DOI: 10.54254/2755-2721/2026.GU34603

[5] Yang, Y., Sun, F. Y., Weihs, L., VanderBilt, E., Herrasti, A., Han, W., ... & Clark, C. (2024). Holodeck: Language guided generation of 3d embodied ai environments. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 16227-16237).
[6] 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.
[7] Hu, Z., Iscen, A., Jain, A., Kipf, T., Yue, Y., Ross, D. A., ... & Fathi, A. (2024, March). Scenecraft: An llm agent for synthesizing 3d scenes as blender code. In Forty-first International Conference on Machine Learning.
[8] Karnewar, A., Vedaldi, A., Novotny, D., & Mitra, N. J. (2023). Holodiffusion: Training a 3d diffusion model using 2d images. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 18423-18433).
[9] Wang, C., Peng, H. Y., Liu, Y. T., Gu, J., & Hu, S. M. (2025). Diffusion models for 3D generation: A survey. Computational Visual Media, 11(1), 1-28.
[10] Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., & Ng, R. (2021). Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM, 65(1), 99-106.