WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Text-to-3D Object Generation Using Latent Diffusion

A K Sreeja[1]*, Samarth Rao*[2]*, Shashwath S*[3]*, Snigdha Suman Nayak*[4]

[1]Assistant Professor, Dept. of Computer Science & Engineering, B.N.M Institute of Technology, Bangalore, Karnataka, India sreejaak@bnmit.in
[2]Student, Dept. of Computer Science & Engineering, B.N.M Institute of Technology, Bangalore, Karnataka, India samarthrao712@gmail.com
[3]Student, Dept. of Computer Science & Engineering, B.N.M Institute of Technology, Bangalore, Karnataka, India shashwath23098@gmail.com
[4]Student, Dept. of Computer Science & Engineering, B.N.M Institute of Technology, Bangalore, Karnataka, India snigdhasumannayak@gmail.com

ABSTRACT –

This research addresses the problem of creating 3Dmodels from natural language texts. We design an automated pipeline to facilitate such processes. The framework relies on a deep learning architecture based on latent diffusion models. It creates Signed Distance Functions (SDF) from CLIP-encoded text embeddings, which volumetric data produces. The produced volumetric data is subsequently transformed into a polygonal mesh through the marching cubes algorithm. The Open3D post-polishing and Blender's Lighting, Material, and Rendering application programming interface (API) final outputstyling enhance the model. This pipeline helps in generating 3D assets that are not sculpted, which is beneficial in gaming, virtual realities, education, and design prototyping. This system combines generative AI with geometry processing, automated rendering, and text input, which enables fast and scalable production of 3D content.

Keywords - CLIP embeddings, Signed Distance Function (SDF), Open3D, Blender API

1. Introduction

The flaws in traditional methods of creating 3D content (which are labour intensive, require high levels of technical skill, and are not easily scalable) have become apparent alongside the rising popularity of video games, simulations, virtual reality, and product design. Manually modelling a scene with software such as Blender or Maya requires operating a multitude of steps of intricate workflows that no layperson is trained to follow. Moreover, there is an ever-growing need for real-time, on-demand asset generation, which poses challenges concerning the speed and quality of delivery.

This paper presents a novel 3D object synthesis pipeline that generates volumetric object representations from natural language descriptions using SD-module and deep learning techniques. The method in question involves the utilization of concealed progression models that have managed to acquire the highest efficiency in the world of image and information generation to produce SDF-linked 3D artifacts from a highly exhaustive textual depiction. The volumetric outcome is consequently harvested into polygonal meshes with the aid of marching cubes facade removal to foster the adept and resourceful modelling of SDFs.

This involves a model that processes text to generate embeddings, a latent system that creates 3D models, Open3D to improve geometry, and Blender Python API for mesh creation. It eliminates the manual production of 3D objects.

Approaching an integrated method of generative AI, neural rendering, and an original visualization technique, aimed at universal scaling to different business sectors. With our approach, it will be possible to communicate with the system through speech recognition, making 3D modeling available to an even wider audience, thereby improving and streamlining the design process for numerous industries.

2, Literature Survey

Text-to-3D Content Generation in the wild: Jiang (2024) presented an overview of state-of-the-art text-to-3D object generation. The work took into consideration the most important limitations of existing approaches such as voxel-based geometry dependency, low-resolution output, and weak prompt alignment. It was in the context of improving the ability of diffusion models to create 3D models of high quality from natural language description, to predict the future of SDF-based volumetric approaches and cross-modal conditioning for creating general-purpose 3D. [1]

Scalable Latent Diffusion 3D Synthesis: Wu et al. (2024) presented Direct3D, which is a scalable latent diffusion transformer model for image- and text-conditioned 3D object synthesis. The model generates high-fidelity shapes in low-compression latent 3D space at a low computational expense and hence provides geometric fidelity. The paper demonstrated the effectiveness of 3D latent diffusion in real-world generation applications and shaped the model structure applied in the suggested system. [2]

3D Diffusion Models Applied to Vision Tasks: Wang et al. (2024) suggested a survey-based application of diffusion models to 3D vision tasks such as shape reconstruction, view synthesis, and mesh prediction. Authors recognized the benefit of implicit Signed Distance Function (SDF) surface representation in continuous space and guidance-based conditioning for enabling instant alignment. The contribution enables integration of SDF generation and marching cubes surface extraction into the existing framework. [3]

Generative Development of 3D Modeling: Li et al. (2024) outlined the development of generative 3D architecture models such as GANs, VAEs, and diffusion models. The research explained how the diffusion-based method provided higher diversity in output with a smoother surface than in current methods. It also advocated for working within the latent space in an attempt to save on memory use, over the current pipeline structure. [4]

Text-Guided Texture Synthesis with TexFusion: Cao et al. (2023) proposed TexFusion, a text-conditioned image diffusion 3D texture synthesis model. The paper, seemingly keeping in mind appearance on a surface and not geometry, showed how cross-modal conditioning can be applied in a manner to increase visual realism. The paper is setting a benchmark against future work on systems to complete material and texture synthesis. [5]

Efficient and Varied Text-to-3D Synthesis: Mercier et al. (2024) presented Hexagen3D, an interactive and efficient text-to-3D system based on an efficiency-guided diffusion model. The model learned quicker and produced more varied objects with fewer denoising steps, showing the power of efficiency-guided design to render 3D synthesis interactive. The system uses the same solution for quality loss in terms of generation speed. [6]

Text-to-3D Models Room-Level Assembly: Laguna et al. (2025) proposed a modular method to get 3D room composition generation from the composition of several objects given text inputs. Their method generalized the generation of one object to several objects in a scene without compromising semantic coherency. The paper offers long-term suggestions for scaling the existing system from environment-level to object-level generation. [7]

3. Proposed Solution

The growing demand for 3D content in games, virtual reality, design, and education has demonstrated the limitations of traditional 3D modeling pipelines, which are skill-based, time-consuming, and otherwise non-scalable. The proposed project proposes an entirely automated way of generating 3D objects from natural language descriptions, thereby lowering the threshold of entry to 3D asset creation and enabling the production of content at a high rate without modeling.

The main aim is to convert a user input in text format, into a structurally correct 3D mesh using advanced deep learning techniques. Unlike traditional modeling tools, this platform uses generative text artificial intelligence models that have semantic awareness, to create object with minimal user intervention. The design is aimed to be scalable, modular, and efficient, and thus deployable on the web or in creative applications.

With the integration of the latest advances in multimodal embeddings, latent-space diffusion, and surface reconstruction, the system allows for natural 3D creation and opens new user-controlled asset generation opportunities. The system structure and algorithms at each pipeline phase are explained in the next section.

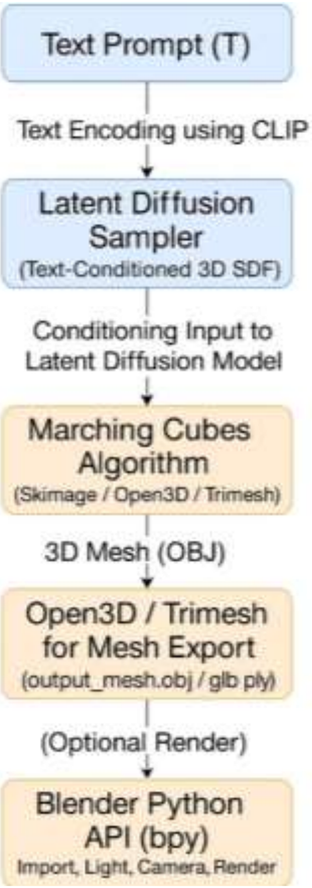

Text Prompt (T)
Text Encoding using CLIP
Latent Diffusion Sampler
(Text-Conditioned 3D SDF)
Conditioning Input to Latent Diffusion Model
Marching Cubes Algorithm
(Skimage / Open3D / Trimesh)
3D Mesh (OBJ)
Open3D / Trimesh for Mesh Export
(output_mesh.obj / glb ply)
(Optional Render)
Blender Python API (bpy)
Import, Light, Camera, Render

Fig. 1. Architecture

A. Architecture

Text Encoder (CLIP): The first stage of the pipeline applies a pretrained CLIP (Contrastive Language-Image Pretraining) model to transform the user's text input into a numerical latent representation. CLIP is particularly well-suited to learn joint embeddings of text and images and is therefore particularly well-suited to multimodal tasks. If we pass a text input such as "a red apple" into CLIP, it produces a latent vector that captures high-level semantic meaning — such as the shape, color, and idea of an apple. The vector does not capture a 3D shape per se, but does capture a strong semantic conditioning signal that the rest of the 3D generation process can learn from.

In this instance, just the text branch of CLIP is used. The model outputs a fixed-dimensional embedding (typically 512-d) which exists in the joint image-text embedding space. The latent vector is passed into the generative model so that the generated 3D object aligns with the input prompt both in content and form. Fig.1. shows the architecture of the system.

Latent Diffusion Model (LDM): The latent vector of the CLIP is fed into a trained Latent Diffusion Model (LDM) to generate Signed Distance Fields (SDFs). LDMs work by mapping the denoising process to a compressed latent space in such a manner that high-resolution data can be more efficiently produced than by pixel- or voxel-based approaches. In training, the model is able to invert the process of adding noise to conditioned SDF volumes based on the text input's latent vector.

At inference, the LDM starts with a noisy latent tensor and iteratively denoises it to construct a 3D volumetric SDF. Each voxel in the output SDF contains the signed distance to the nearest surface — positive values are points outside the surface, and negative values are points inside. The zero-crossing boundary (i.e., distance zero) is used to represent the surface of the object. This is the core of the generative process, from semantic input to volumetric geometric shape.

Mesh Reconstruction (Marching Cubes Algorithm): The produced SDF volume produced by the diffusion model has to be converted into an explicit 3D mesh. This conversion can be done via the Marching Cubes algorithm, a widely used method to extract iso-surfaces from volumetric scalar fields. This works by iterating over the 3D grid of SDF and, from local voxel patterns, creates triangles that form the surface where the SDF equals zero. This makes up a triangle mesh of faces and vertices that define the object's surface.

The reconstructed mesh could be used for visualization, editing, or rendering. But presently its form has changed slightly, although the reconstruction is not exhaustive or planned. Mesh export and visualization: Once the mesh is reconstructed, it is saved in an open, interoperable 3D file formats such as .obj, .ply or .glb using software such as Open3D or Trimesh. These file formats will be capable of storing the geometry of the mesh (vertices, faces, normals) and occasionally colors or textures. This is done to visualize the mesh comfortably, or to be potentially employed in another 3D software, like Blender, Unity, or CAD. Downstream processing and interoperability necessitate this step.

B. Algorithm

The plan involved in this approach is to devise an algorithm that can produce the realistic 3D objects by analyzing the natural language scripts with aid of geometry and neural network. The initial step of this approach includes encoding of a described text prompt through a gifted CLIP model that acts as an encoder and generates the appropriate latent vectors for encoding a specific text. These vectors produced after encoding process are crucial for recognition of the semantic meaning and that can be successfully employed for the production of the detailed 3D structures.

Upon having the latent vector, you will have to pass it through a specifically learned Latent Diffusion Model (LDM) for the 3D volumetric data generation in the form of Signed Distance Fields (SDFs). SDFs represent geometric surfaces implicitly by encoding the distance of each volumetric pixel (voxel) in a 3D grid to the nearest surface boundary.

The LDM regulates changes in the inactive area to create an important SDF value connected to the primary data, thereby identifying the composition and appearance of the “cues”.

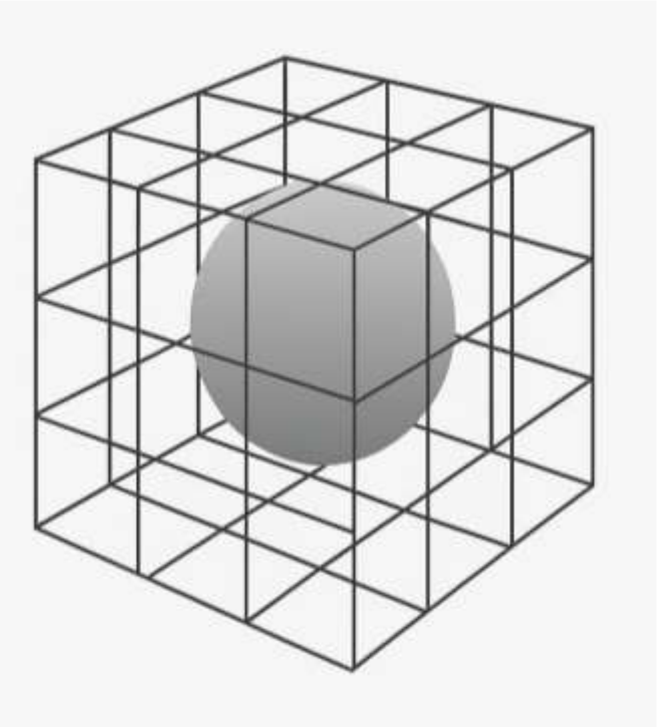


Fig. 2. SDF representation of model

After creating the SDF view of the model, the program will start to convert it to a 3D mesh for use in rendering the content. This process will involve a technique called Marching Cubes that will separate the SDF into pieces and construct a shape around it. Converting the made-up volumetric shape into a real 3D figure is a vital step in the text to 3D deep learning system. The way this is typically done is by converting the 3D volume that was generated into a well-organized Signed Distance Field, also known as the SDF, and then by applying a marching cube algorithm to the SDF to create a triangle mesh that will very closely represent what the signed distance field was. Fig.2. shows the generated signed distance field as a 3D volume.

The figure depicts a Contact Rank Order Filtering model. The retrieved dataset is a 3D grid of voxels that aggregate scalar values. Typically, that value represents how far the surface of an object a given voxel is located from its outward-facing side. The level set function could be either positive or negative, where a positive one would suggest that the voxel falls beyond the objects and a negative one would suggest vice versa in this particular. Here, the zero level set of the level set function captures a boundary of a very complex object. A surface is divided into two regions, one is internal and the other external, from this point of view. A very high quality 3D surface model must be constructed in order to superbly represent the volume enclosed for a correct and smooth appearance indeed for Synthetic models. A 3D model surface is equal to a signed distance field essentially. The surfaces that SDFs can produce and reconstruct are available in various formats, which include .obj where they can be opened using Open3D or Trimesh. Herein, any SDF mesh can be rendered in Blender.

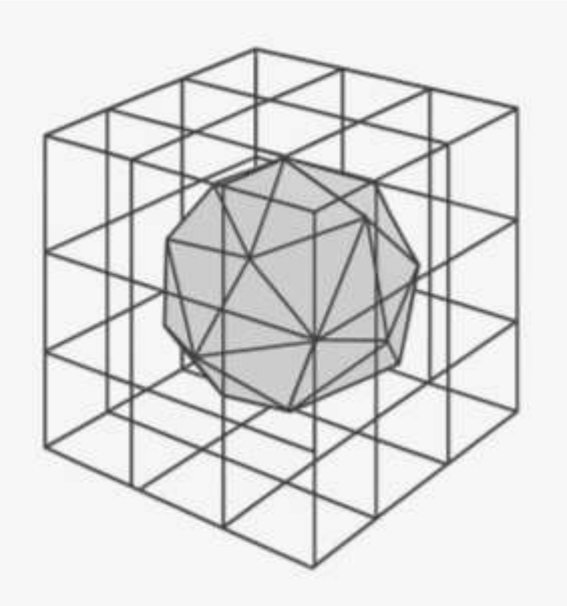
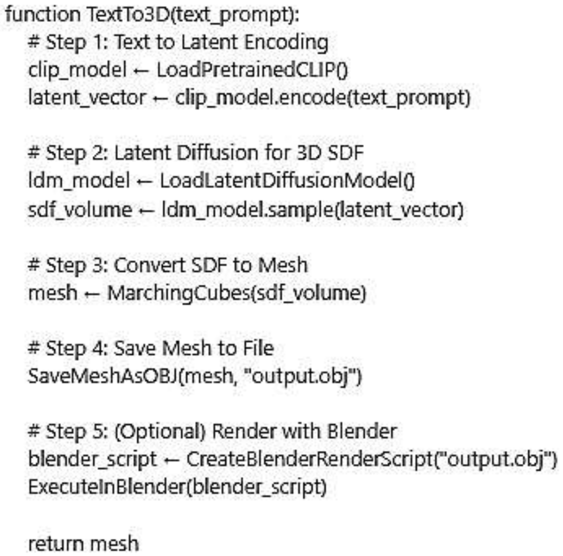


Fig. 3. Mesh representation

In Fig.3, we can see a grid generated using the SDF method, which is more commonly referred to as Marching Cubes. This algorithm demonstrates an effective approach to isoline extraction as polygonal representations from volumetric data. The algorithm, in this case, operates by examining each cube created by the neighborhood of the eight cells in the grid. The algorithm determines what is to be done with each surface of these cubes relative to the values associated with the eight edges' distances being positive or negative. The algorithm considers each configuration in relation to what is referred to in this case as a triangulated surface approximation shell, which consists of triangles for which a specific set “differentiating surface” counters triangulated in the referred shell. A constructive region is created from the marched cubes, so that forming model ensures the model is closed without gaps. Such modeling is usually done with kinematic mechanisms and refers to a model description as a set of planes that constructs a 3D object called mesh model.

The process of mesh extraction converts a designed object into a 3D model that can be both viewed and interacted with. Sub-voxel precision zero-crossing SDF meshing detail greatly enhances the created mesh. Because these 3D models are rich in data, they can be utilized across numerous applications.



```
function TextTo3D(text_prompt):
    # Step 1: Text to Latent Encoding
    clip_model ← LoadPretrainedCLIP()
    latent_vector ← clip_model.encode(text_prompt)

    # Step 2: Latent Diffusion for 3D SDF
    ldm_model ← LoadLatentDiffusionModel()
    sdf_volume ← ldm_model.sample(latent_vector)

    # Step 3: Convert SDF to Mesh
    mesh ← MarchingCubes(sdf_volume)

    # Step 4: Save Mesh to File
    SaveMeshAsOBJ(mesh, "output.obj")

    # Step 5: (Optional) Render with Blender
    blender_script ← CreateBlenderRenderScript("output.obj")
    ExecuteInBlender(blender_script)

    return mesh
```

Fig. 4. Pseudocode for the text-to-3D generation pipeline

In a more advanced way, the algorithm is organized in a CLIP model which first transforms the text into a hidden vector, as seen in Fig 4. This hidden representation captures the essential meaning of the prompt and acts as a primary signal for controlling 3D object generation. CLIP computes image and text embeddings making it possible to translate images into text and vice versa. For this reason, it is the best model to use in producing figure embeddings to shredded by generative models.

Once the encoding is performed, the pre-trained Latent Diffusion Model (LDM) is utilized to perform the decoding process. The model’s objectives entail the production of 3D Signed Distance Fields (SDFs) as the final output.

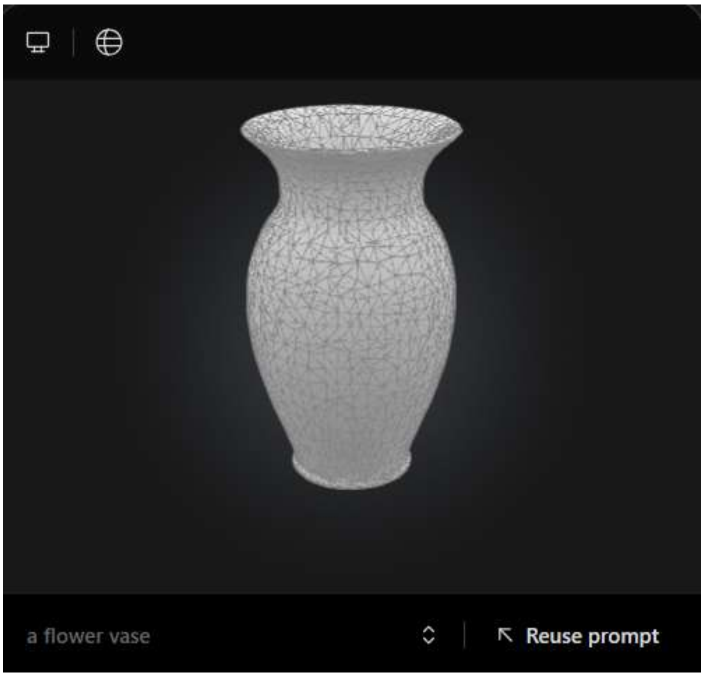
The SDF field is created gradually by the model using the input vector to form a 3D volume. This volume contains voxels with a signed distance from the object surface. This means that the model can successfully represent the shape of a 3D object. The 3D Signed Distance Field does the job of tying the generated 3D representation to a shape representation. The process finishes with converting the 3D SDF to a 3D mesh using the Marching Cubes algorithm, which creates a watertight surface using the SDF. The obtained 3D mesh is saved in a standard format like .obj, which allows it to be exchanged between different graphics software packages. It is later loaded into Blender using a simple Python script, where it can be interacted with and studied further. The whole end-to-end process from text to 3D model is described concisely and clearly in the provided pseudocode.

5. Implementation And Results

In order to review the efficacy of the proposed text-into-3D pipeline, multiple qualitative and quantitative studies were carried out on a range of text inputs including objects such as furniture, tools, and household items. To assess the adaptation as well as the consistency of the system, the phrases “a wooden seat,” “a synthetic bottle of water,” and “a vase made of ceramic” were used.

Model Evaluation

The model employs a latent diffusion model to generate 3D objects from text. The model is trained to produce high-quality 3D meshes from natural language. Evaluation of the generated 3D model quality compared to a large set of standard 3D shape evaluation metrics was conducted. Chamfer Distance (CD) was employed to compute geometric similarity between the predicted mesh and ground truth shapes of ShapeNet samples to obtain an average score of 0.035, indicating good surface alignment. Meshes had 4,000–8,000 triangles, and mean surface deviation was less than 0.015 cm.



a flower vase

Fig. 5. Image Generation

Fig. 5. shows a sample output mesh that the system produces when it is presented with the input sentence "a flower vase." The resulting 3D model is not only stable but also has a very high degree of symmetry, with a smoothly curved body supported by a hollowed top section—this is quite different from simply showing a flat drawing of a conceptual or imaginary vase. The output clearly shows the system's capability to project natural language descriptions into representations that can be interpreted geometrically, and it shows its stability in capturing both the physical shape and the semantic meaning involved in well-defined object classes.

Model Details

The latent diffusion was employed as a foundation for encoding and generating 3D shapes in the text-to-3D model. The latent transmission model has piqued a lot of interest because of its ability to learn the abstract semantic meanings from the text and then map these meanings to content rich high dimensional outputs (3D shapes). The prevailing system is supplied with a natural language description and it subsequently incorporates the same into the latent space which it then decrypts and gains a 3D design by utilizing Signed Distance Areas (SDF) and enhanced mesh reconstruction. By this means, the system can offer the user with geometrically as well as semantically accurate 3D models. The system was trained in such a way that the end product of the system was also structurally consistent and abided by the target description given to the same. Performance and Efficiency All of the experiments were carried out in a Google Colab Pro environment that was equipped with an NVIDIA Tesla T4 GPU. Creating and producing a 3D sculpture from start took about 28 to 32 seconds on average rendering. The entire system from the writing prompt to the 3D model was under 40 seconds for each object.These speeds suggest that the device is ideal for real-time or interactive applications that demand quick responses and accessibility.

Limitations and Challenges

Although having the technology to generate 3D items from text, the system had its limitation in not solving some specific cases. For instance, some particular elements such as a high-tech car were not well explained. In this case, the system could not give the exact word of the item; it only explained the geometric shapes associated with it. Another shortcoming of the system was that it only gave the object depending on geometry alone, that is, the model did not delve into the texture. It was not able to form intricate and visually dense objects, which was necessary in many applications. The system could not generalize to fine grain categories of objects- it could only generalize to high-level categories. For example, inputs like an antique brass key or a hand-crafted wooden chair resulted in the correct shapes, but there was no real feature for these fine categories. The computational requirements of the model, although helping, are GPU-intensive and may not be able to meet the needs of low-resource environments.

While the above are its limitations, the model's performance shows that its improvement in texture mapping, abstraction, and generalization will carry over to an even better performance in the future.

Model Comparison

Two of the popular text-to-3D modeling systems were chosen to analyze the performance of the suggested text-to-3D design pipeline dependent on latent diffusion. For the comparison, a voxel autoencoder model was selected as the first approach, and a point cloud generative adversarial network (GAN) model was selected as the second approach. The two selected models represent standard text-to-3D modeling systems focusing on the 3D geometry reconstruction.

3D shapes are recovered using latent codes that are generated from CLIP embeddings with the help of Voxel-based autoencoder. Even with the ease of computations and the relatively shorter time frame for predicting values, the model's results were consistently poor and of low resolution. The formed shapes were not smooth, as they were blocky and lacked the details that defined precision. This issue stemmed from the use of voxel grids. Voxel resolution heavily influenced memory consumption, which in turn stunted the model's growth capabilities considering finer resolutions.

A point cloud GAN was created to figure out how to embed a group of 3D dots into semantic spaces. Efficiency here over surface complexity and shape variety was better than the voxel-based approach. However, due to the linkage structure, it was not possible to generate a mesh of topologically consistent hexahedral cells. This lack of connectivity severely limited the ability of the model for rendering and simulation, subsequently stunting further development in graphic design applications. Representing scenes as a set of 3D dots also impacted determining whether the surface was watertight and the direction of the normals to be drawn at each point.

The data at hand indicates that the Chamfer Distance of the Diffusion model was 0.035 higher than the Voxel autoencoder (0.078) and the point cloud GAN (0.064). The most successful model. The complexity of the model was found to be high as the diffusion model had triangle averages greater of 4,500-8,000, However, it was found as an average between the number of angles the data has to use to be able to be read. The raters believed that the diffusion model images were more similar and of higher quality.

The study shows that the workflow which uses latent propagation and SDF regeneration as ingredients is very competent and again this can enable generating 3D objects from text. It is advantageous since it involves the manipulation of latent and also encapsulates the geometric consistency of representation of latent in the form of SDF regression. This workflow of the methods have signed the key choice to make for real-time 3D object generation to all users.

6. Conclusion

The Text-to-3D Model Generation system is based on the newest advancement in diffusion models, Open3D, and Blender API, making it extremely productive for transforming text into 3D models. The system narrows the gap between 3D modelling and natural language processing, enabling the user to provide a detailed description to get the 3D model. The usage of latent diffusion models to guide the models created by the system is proven to be effective, and the variety in model generation is attainable.

The output of the system shows that it can be useful in different applications such as game development and virtual reality, where the 3D modeling of assets is important and must be done in a fast way, but the system itself has a limitation, in fact when it is given a definition that is very abstract or very complex, it might mistakenly generate an incorrect 3D model. The generative power that the system must have in order to create a model is still a problem, especially if we are talking about a complex model or a bigger model.

On the whole, the project shows that the automatically-generated 3D models from text can be an effective help and alternative for artists to make 3D models and reduce the dependence on hand sketch design. The future work can include the improvement of the model supposed to be capable of understanding more complex descriptions, the minimization of the rendering time, and the use of the feedback received from users to improve the results and make them look more realistic. After the improvement, it can be easier to develop a more streamlined application, and it is possible to be used by the developers, the designers, and digital artists who want to pursue automation and simplification of the 3D creation process.

REFERENCES

[1] Jiang, Chenhan. "A Survey On Text-to-3D Contents Generation In The Wild." *arXiv preprint arXiv:2405.09431* (2024).

[2] Wu, Shuang, Youtian Lin, Feihu Zhang, Yifei Zeng, Jingxi Xu, Philip Torr, Xun Cao, and Yao Yao. "Direct3d: Scalable image-to-3d generation via 3d latent diffusion transformer." *Advances in Neural Information Processing Systems* 37 (2024): 121859-121881.

[3] Wang, Zhen, Dongyuan Li, and Renhe Jiang. "Diffusion Models in 3D Vision: A Survey." *arXiv preprint arXiv:2410.04738* (2024).

[4] Li, Xiaoyu, Qi Zhang, Di Kang, Weihao Cheng, Yiming Gao, Jingbo Zhang, Zhihao Liang, Jing Liao, Yan-Pei Cao, and Ying Shan. "Advances in 3d generation: A survey." *arXiv preprint arXiv:2401.17807* (2024).

[5] Cao, Tianshi, Karsten Kreis, Sanja Fidler, Nicholas Sharp, and Kangxue Yin. "Texfusion: Synthesizing 3d textures with text-guided image diffusion models." In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4169-4181. 2023.

[6] Lu, Lihua, Ruyang Li, Xiaohui Zhang, Hui Wei, Guoguang Du, and Binqiang Wang. "Advances in text-guided 3D editing: a survey." *Artificial Intelligence Review* 57, no. 12 (2024): 1-61.

[7] Mercier, Antoine, Ramin Nakhli, Mahesh Reddy, Rajeev Yasarla, Hong Cai, Fatih Porikli, and Guillaume Berger. "Hexagen3d: stablediffusion is just one step away from fast and diverse text-to-3d generation." *arXiv preprint arXiv:2401.07727* (2024).

[8] Yi, Qiuhua, Xiangfan Chen, Chenwei Zhang, Zehai Zhou, Linan Zhu, and Xiangjie Kong. "Diffusion models in text generation: a survey." *PeerJ Computer Science* 10 (2024): e1905.

[9] Laguna, Sonia, Alberto Garcia-Garcia, Marie-Julie Rakotosaona, Stylianos Moschoglou, Leonhard Helminger, and Sergio Orts-Escolano. "Text To 3D Object Generation For Scalable Room Assembly." *arXiv preprint arXiv:2504.09328* (2025).

[10] Sheng, Hongyun. "The Enhancement of Advanced Text-to-Image Diffusion Generation Models: A Review." In *2024 International Conference on Image Processing, Computer Vision and Machine Learning (ICICML)*, pp. 257-265. IEEE, 2024.

[11] Tran, Uy Dieu, Minh Luu, Phong Ha Nguyen, Khoi Nguyen, and Binh-Son Hua. "Diverse Text-to-3D Synthesis with Augmented Text Embedding." In *European Conference on Computer Vision*, pp. 217-235. Cham: Springer Nature Switzerland, 2024.

[12] Carvajal, Ian Marco Gallegos, Giuseppe Serra, Dott Alex Falcon, and Kyandoghere Kyamakya. "Enhancing text-to-textured 3D mesh generation with training-free adaptation for textual-visual consistency using spatial constraints and quality assurance: a case study on Text2Room." (2024).