Textured 3D Model Generation in Polygon-Form Using Generative AI - Review

Max Gittel
Chair for Data Processing, Technical University of Munich
max.gittel@mytum.de
Supervisor David Fresacher

***Abstract*—3D models are integral to various domains such as video games, industrial simulations, and medical applications, and their significance is growing constantly. While generative AI models for text and image creation have seen remarkable advancements recently, the generation of 3D models remains challenged by higher complexity and limited training data. In many application areas, the capability to render 3D models in real-time is essential. Consequently, 3D models are often produced in polygonal form, which allow for efficient and fast rendering while preserving details through textures. This literature review examines the current state-of-the-art approaches for generating textured 3D meshes in polygon form using text prompts or images as inputs. Various methods are analyzed and compared, highlighting their strengths and limitations. Finally, the paper proposes a potential model for future developments in 3D model generation, addressing existing challenges and suggesting directions for further research.**

***Keywords*—3D model, generative ai, computer graphics.**

I. Introduction

Creating 3D models is a crucial task in visual applications such as films, video games, and AR/VR environments. High-quality 3D models are essential for the quality, usability, and level of immersion in virtual worlds. However, creating these models is labour-intensive and requires a high degree of artistic skill.

To simplify this process and support artists in their creative work, current research focuses on the automated generation of 3D models, either from the ground up or partly by generating textures for existing models. Recent advancements in machine learning, such as stable diffusion [23] for generating images from text prompts, offer promising solutions. Additionally, new 3D representation methods, like Neural Radiance Fields (NeRFs) [16], have been developed that might be used to generate 3D models from images. NeRFs can generate new images from unknown perspectives using only a few static images of an object from different angles.

Although current results have not yet achieved the level of quality and generation speed required for actual production use, there are promising indications of what might be possible in the coming years. With numerous research projects and approaches being explored, it is challenging to maintain a comprehensive overview.

This paper aims to consolidate, evaluate, and compare all these technologies to predict the future direction of research in 3D model generation.

Applications of such a technology are extensive, including game development, virtual reality, architectural visualization, and digital content creation. For instance, integrating automated texture design as a plugin tool in 3D modelling software can accelerate workflows for 3D artists. As demand for immersive digital experiences grows, advancing machine learning techniques for 3D model generation becomes a crucial, economically significant task that drives innovation and creativity across industries.

II. Methodology

A. Geometry Representation

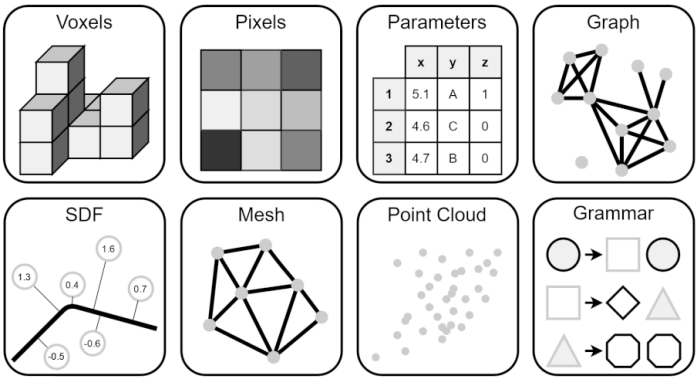

Fig. 1. Visualizations of possible geometric representations for a 3D model adapted from [21].

Mesh: Meshes are the most common representation used in the real-time industry. They are optimized for performance and are well-suited for manual modelling methods. Each mesh consists of a list of vertices, with a combination of three or more vertices forming a face, which is projected onto a 2D-pixel grid through a rasterization process. The disadvantage of meshes is that they are unordered and non-differentiable, making them less suitable for machine-learning approaches. Often, other representation methods are used and converted into a mesh in the final step.

Voxel: The Voxel representation is the 3D equivalent of a pixel grid. It is a volumetric representation where a 3D space is divided into segments called Voxels. These Voxels contain information corresponding to their specific location, such as whether the space is empty or occupied by an object.

Signed Distance Field (SDF): Similar to the Voxel representation, a SDF is a divided field where each point in the space has a value indicating how far it is from the nearest surface.

Point Cloud: A collection of unconnected points in 3D space, representing parts of the external surface of an object. Point clouds are mostly acquired through 3D scanning of real environments and objects.

Neural Radiance Fields (NeRFs): A novel method for representing and synthesizing 3D scenes from a limited set of 2D images. By utilizing a deep learning model, NeRFs create a continuous function in 3D space, where each point contains colour radiance information that varies with the viewing angle. To generate an image from a NeRF, traditional volumetric rendering techniques are employed [16].

Other representations that might be used include Grammar, Graphs and Parameterizations [21]. All mentioned representations can generally be converted between each other, though not always without a loss of information.

B. Textures Representation

Textures Mapping: Texture Mapping projects a 2D image texture onto a 3D mesh surface by unwrapping the model into a 2D space. The albedo-map represents the base color without lighting effects and the material map defines properties like reflectivity and glossiness. Texture maps oftentimes specifically depend on the geometry of the given model and cannot be reused on other models easily [7].

PBR Materials: Physically-Based Rendering (PBR) [26] Materials simulate real-world material properties for more accurate light interaction. Key components include metalness (indicating if the surface is metallic), Roughness (defining surface texture detail), and Base (Albedo) Color. PBR materials ensure realistic and consistent visuals across various lighting conditions. The advantage compared to normal texture maps is that they are often independent of the geometry and can be easily reused across different models. However, this can also be a disadvantage, as it may cause the texture to appear misaligned or unnatural in certain areas.

Texture Field: A continuous 3D function parameterized with a neural network which avoids limiting factors like shape discretization and parameterization since it is independent of the shape representation of the 3D object. [18].

Neural Radiance Fields: Neural Radiance Fields (NeRFs), use a neural network to encode volumetric density and color for each point in space. These fields enable realistic rendering by varying colour based on viewing direction and using volumetric rendering techniques. Radiance fields can capture fine details and complex light interactions, synthesizing novel views from sparse input images [16].

C. Generation Methods

Generative Adversarial Network (GAN): A generative method that involves the simultaneous training of two models: a generator G, which is designed to produce synthetic data, and a discriminator D, which is tasked with distinguishing between real data from the training set and data generated by G. The generator is iteratively trained to deceive the discriminator, resulting in progressively improved synthetic data that increasingly mimic the original data sets [6].

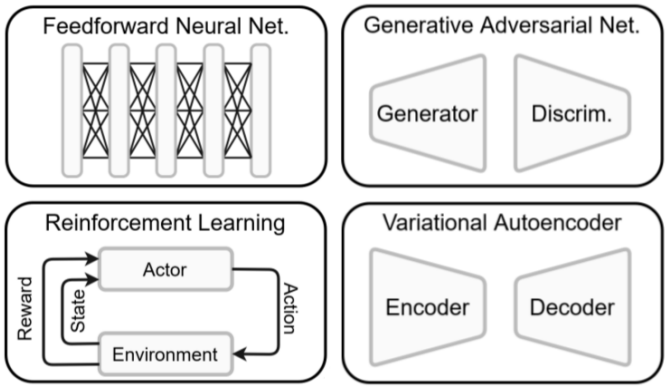

Fig. 2. Visualizations of possible machine learning methods commonly used for generative AI adopted from [21].

Variational autoencoder (VAEs): An autoencoder is an artificial neural network architecture that consists of two components: the encoder and the decoder. The task of the encoder is to map the input data points to a latent space that has a much lower dimensionality than the input data but still retains all the important information from the input data. The decoder is simultaneously trained to generate the original input data as accurately as possible from these compressed data points. While simple autoencoders are not designed to create novel content, variational autoencoders (VAEs) can generate new content through the usage of a parameterization of a probability distribution as its internal representation [12].

Reinforcement Learning (RL): Reinforcement Learning (RL) differs from other machine learning methodologies in that it does not rely on pre-existing datasets for the learning process. Instead, it involves an actor that interacts with an environment or state space, receiving rewards for actions that contribute to predefined goals. The actors's objective is to determine the optimal sequence of actions that maximizes the cumulative reward over time. While a key advantage of RL is its independence from large datasets, its effectiveness heavily depends on the design of a well-suited reward function and the structure of the environment and action space [10].

Diffusion models: Diffusion models are machine learning techniques designed to generate high-quality synthetic data by progressively adding noise to input data, creating a sequence of increasingly noisy versions. The model learns to predict the noise introduced at each step and then reverses the process, gradually denoising the noisy data to reconstruct the original input. These models can also be conditioned on specific inputs, allowing them to generate outputs that follow desired patterns or attributes, making them highly versatile in controlled generation tasks [23].

III. Contributions

A. Model Generation

1) 3D Model From Text: A significant portion of research initiatives in this domain focuses on developing methodologies for generating 3D models from text prompts. A widely adopted technique in this area is the **Zero-shot** approach, where a model can generalize to new tasks or data it has never encountered before. In the context of 3D model generation, generative models often rely on large-scale image data sets for training, rather than using actual 3D models or data. This approach allows the models to learn from extensive visual information, which can then be applied to create accurate and realistic 3D representations from text descriptions. A key model frequently used in these research projects is **CLIP (Contrastive Language–Image Pretraining)**, a pre-trained model that interprets the relationship between images and text descriptions. In many of the studies that follow, CLIP is used to assess the correlation between the generated image and the input text prompt, effectively measuring the quality of the generative output.

Approaches like **CLIP-Mesh** [11] utilize the zero-shot method, relying solely on textual input without the need for explicit 3D supervision. This technique deforms a subdivided control mesh, along with its associated texture and normal maps. CLIP is then used to evaluate the correlation between the rendered images and the input text prompt. By treating this as an inverse problem, the shape and texture of the mesh can be optimized to maximize the CLIP score, ensuring that the generated 3D assets closely align with the text descriptions. Although the initial results from this method often produce rough outputs, characterized by distorted geometries and unnatural color schemes, the foundational technique has been highly influential and has been further developed in numerous subsequent studies.

Similarly, **Dream Fields** [8] is a zero-shot method that employs simple geometric priors, generating the geometry and colour of various objects without the need for 3D supervision. Unlike CLIP-Mesh this does not allow the direct generation of a mesh but instead trains a neural radiance field [16]. Once again a pre-trained CLIP model is used to measure the alignment, which enhances both fidelity and visual quality. The large training data set allows Dream Field to generate more diverse and creative models. However, the results are often times very abstract and only very loosely resemble the intended objects in the prompt.

Another zero-shot method, incorporating explicit 3D shape priors is seen in **Dream3D** [29], which also combines these priors with CLIP-guided 3D optimization. Dream3D begins by generating a high-quality 3D shape from an input text, using this shape as a starting point for initializing a neural radiance field. This field is then further optimized based on the full-text prompt, allowing for a more effective translation of complex textual descriptions into 3D shapes. Compared to the previously mentioned approaches, Dream3D generation results reflect a good understanding of symmetry, proportions and straight clear lines, however, they still struggle to create natural or aesthetically pleasing textures and contain strong noise or artefacts.

Many current approaches take a significant amount of time to generate models, even on modern hardware. **LATTE3D** [28] stands out for its significantly reduced generation times compared to its competitors, aiming to achieve real-time text-to-3D synthesis. LATTE3D enhances quality and robustness by using 3D data during training, incorporating 3D-aware diffusion priors, applying regularization loss, and initializing model weights through pretraining with 3D reconstruction techniques. Compared to the previously mentioned methods, LATTE3D's strength lies also in its consistency and variety. However, while it generates more natural colours than prior methods, it still often incorporates unwanted lighting information from the training data, resulting in uneven and unnatural texturing. Additionally, LATTE3D tends to overly round sharp edges, sometimes leading to wonky silhouettes.

Techniques like **Magic3D** [13] use a combination of diffusion priors and efficient rendering strategies to refine coarse 3D models into high-quality textured meshes. This process is faster than many related methods, such as **DreamFusion** [20], and yields higher resolution results. Magic3D makes use a low-resolution diffusion prior alongside a sparse 3D hash grid structure, followed by high-resolution refinement using a latent diffusion model. With that Magic3D manages to create highly detailed meshes, which are not easy to achieve with other methods. However, like many other works it struggles with unnatural colors and unwanted lighting artefacts.

In a different approach, **Shap·E** [9] directly generates parameters for implicit functions that can be rendered as textured meshes and neural radiance fields. This method involves using an encoder to map 3D assets to parameters, which are then processed by a conditional diffusion model. By conditioning the diffusion prior on images or text descriptions from a large data set of 3D assets, Shap·E can quickly generate complex 3D objects. The final results tend to struggle with high-frequency objects with thin, small details and often displaying unnatural proportions. Compared to other approaches, the given results of Shap·E indicate that the textures are free of lighting artefacts.

Lastly, **SDF-Diffusion** [24] introduces a novel method for 3D shape generation using denoising diffusion models that represent shapes with signed distance fields (SDF). This approach separates the generation process into two stages: first, a diffusion-based model creates a low-resolution SDF, which is then refined to high-resolution using a second diffusion model. According to the authors, this use of SDFs enhances memory efficiency and allows for direct mesh reconstruction, making the method suitable for detailed 3D shape generation. The results of this paper show simple textureless geometry and overly rounded edges, which might be improved by increasing the SDF resolution.

2) 3D Model from Images: Instead of using text prompts as input for 3D model generation, it is also possible to use images to guide the process. One such method is **Pixel2Mesh** [27], which proposes an end-to-end deep learning architecture that generates a 3D shape in the form of a triangular mesh from a single colour image. This model utilizes a graph-based convolutional neural network to represent the 3D mesh and claims to achieve accurate geometry by progressively deform-

ing an ellipsoid based on perceptual features extracted from the input image. To ensure stability during deformation, a coarse-to-fine strategy is employed, along with various mesh-related loss functions to capture different properties. Pixel2Mesh does not produce textures and can result in shapes with wonky edges and strong rounding, often requiring high polygon counts even for simple shapes.

Another approach, **GET3D** [5], introduces a generative model that directly produces explicit textured 3D meshes with complex topology, rich geometric details, and high-fidelity textures from images. GET3D combines advancements in differentiable surface modelling, differentiable rendering, and 2D Generative Adversarial Networks and is trained using 2D image collections. This model can generate high-quality textured meshes for a wide range of objects, including cars, chairs, animals, motorbikes, human characters, and buildings, significantly outperforming other methods. A standout feature of GET3D is its ability to also generate material properties, like adjusting the reflectivity of a car's window. However, while it creates realistic textures and shapes, it occasionally produces models with unclean edges.

Similarly, the method described in **Convolutional Generation of Textured 3D Meshes** [19] uses only supervision from single-view natural images. A key innovation of this approach is encoding both the mesh and texture as 2D representations that are semantically aligned and effectively modelled by a 2D convolutional **GAN**. This technique is effective in generating good textures but struggels to create high detail 3d models.

The Fine Detailed Texture Learning for 3D Meshes with Generative Models [4] approach focuses on reconstructing high-quality textured 3D models from both multi-view and single-view images. This method frames the reconstruction process as an adaptation problem that progresses through two stages: first, it learns accurate geometry, and then it uses a generative adversarial network to learn the texture. Key enhancements in this pipeline include an attention mechanism based on learnable pixel positions to ensure spatial alignment of textures, and an augmented input for the discriminator with a learnable embedding to improve feedback to the generator. While this method achieves good texture quality, it can also introduce lighting artefacts and distorted shapes.

3) View Reconstruction: **Zero-1-to-3** [15] is introduced as a framework for changing the camera viewpoint of an object based on a single RGB image. To achieve novel view synthesis in this setting, the framework uses geometric priors learned by large-scale diffusion models from natural images. A conditional diffusion model, trained on a synthetic data set, learns to control the relative camera viewpoint, enabling the generation of new images of the same object under specified camera transformations. Additionally, this viewpoint-conditioned diffusion approach can be applied to 3D reconstruction from a single image.

4) Model like Human: A common drawback of generative 3D modelling methods is the disorganized and inefficient topology of the generated meshes, which often complicates manual post-processing by human modellers. Additionally, poor mesh topology can create problems during animations, resulting in undesirable artefacts. Research initiatives such as **Modeling 3D Shapes by Reinforcement Learning** [14] seek to address this issue by enabling machines to model 3D shapes more like human modellers, leveraging deep reinforcement learning (RL). This approach mimics the human process by first identifying a set of primitives that approximate the target object and then refining these primitives to generate detailed geometry that aligns with the intended design. To train the modelling agents effectively, the study introduces a novel algorithm that combines heuristic policies, imitation learning, and reinforcement learning. The results show that agents can learn to produce structured, regular, and topology-aware mesh models, underscoring the feasibility and effectiveness of the RL framework.

As previously mentioned, many prior approaches do not generate polygon meshes directly, instead using alternative 3D representations that are more compatible with generative models. The polygon meshes are then converted in a post-processing step, often resulting in suboptimal topology. **PolyGen** [17] tackles the challenges of generating polygon meshes with clean topology directly by predicting mesh vertices and faces sequentially using a Transformer-based architecture. PolyGen is composed of two key components: a vertex model, which unconditionally generates mesh vertices, and a face model, which predicts mesh faces conditioned on the vertices. The results of this paper demonstrate a clean and efficient topology, similar to what one would expect from a human modeller.

B. Texture Generation

There are multiple approaches to generating textures for 3D models. Most approached begin with an untextured 3D model and use either an input text prompt or an image-based description to define the desired texture. Achieving state-of-the-art performance in text-driven texture synthesis involves overcoming two critical challenges. The first challenge is ensuring broad generalization across various objects, whether guided by diverse prompts or images. The second challenge is eliminating unwanted coupled artefacts like unwanted illumination in the generated textures, which often results from pre-training.

A common method for generating textures involves rendering a 3D mesh from a specific viewpoint and using depth data to generate and inpaint the texture. **Text2Tex** [3] is such a text-driven texture synthesis technique that uses depth-aware diffusion models. The view-dependent textures produced are then back-projected onto the texture map, allowing for the progressive synthesis of high-resolution partial textures from multiple viewpoints. By iterating this process from different angles, new regions are covered, and stretched textures are corrected. Closely related to this **TEXTure** [22] uses the same depth-aware diffusion model generating the texture from different viewpoints with segmentions they call "generate", "refine" and "keep". Additionally, it also enables editing and refining existing textures using either a text prompt or user-provided scribbles.

Another approach that utilizes a publicly available Stable-Diffusion model with depth conditioning is **TexFusion** [2],

which proposes using latent diffusion models that autoencode images into a latent space to generate images within that space. Their pipeline uses Score Distillation Sampling (SDS) to distill, or optimize, a 3D representation such that its renders are encouraged to be high-likelihood under the image prior. This approach claims to produce textures with a more natural tone, stronger view consistency, and significantly faster sampling times compared to other similar methods.

A GAN-based method that directly operates on the surface of 3D objects without requiring 3D colour supervision or correspondence between shape geometry and images for texturing is **Texturify** [25]. Given a shape geometry, Texturify can generate diverse textures by sampling from a latent texture space. By using a latent texture space this method avoids relying on 2D texture parameterization with texture maps, which enables texture generation that respects the 3D structural neighbourhood relations, minimizing distortion.

Most approaches to texture generation focus solely on producing the so-called albedo colour, neglecting material properties such as roughness and metalness. This oversight reduces the overall quality of the results. One project addressing this issue is **PaintIt** [30], which synthesizes high-fidelity texture maps and material information for a given mesh and text description through a synthesis-through-optimization process. Specifically, PaintIt introduces Deep Convolutional Physically-Based Rendering (DC-PBR) parameterization, which replaces traditional pixel-based parameterization of PBR texture maps with convolutional neural kernels that are randomly initialized. During optimization, DC-PBR re-parameterizes the texture maps, optimizing the neural surrogate of the PBR texture maps rather than directly optimizing pixel values.

Paint3D [31] addresses the challenge of generating high-quality textures without embedded illumination, allowing for re-lighting and re-editing within external graphics pipelines. It starts by using a depth-aware 2D diffusion model to create an initial coarse texture map through view-conditional images and multi-view texture fusion. To overcome the limitations of 2D models in fully representing 3D shapes and removing lighting effects, texture map inpainting and UVHD diffusion models are used for refining incomplete areas and eliminating illumination artefacts. This coarse-to-fine approach enables Paint3D to produce high-quality, consistent 2K texture maps without lighting artefacts, advancing 3D object texturing. The diffusion models are trained in texture map space, using feasible 3D objects and their high-quality illumination-free textures as supervision.

Finally, we consider **Mesh2Tex** [1], which introduces a novel hybrid mesh-neural-field texture representation. This approach enables diverse and realistic texture generation on object mesh geometries by linking a neural texture field to the barycentric coordinate system of the mesh faces. While the generated textures are of high quality with minimal distortions, they still exhibit unwanted illumination information embedded within the textures.

IV. Challenges and Future Work

Despite the enormous progress in 3D model generation in recent years, several challenges still need to be addressed. The following section outlines some of these issues.

A. Challenges

Usability: A significant challenge will be elevating the generated 3D models to an industry-usable level. To achieve this, the models must be user-friendly, high-quality, fast, and controllable. Existing methods often rely on volume rendering or neural rendering and fail to produce quality content suitable for the rasterization graphics pipeline.

Generation times: For generative AI to be viable in 3D model production, it must be fast enough to integrate smoothly into workflows, support rapid iterations, and scale to meet high-volume demands, ensuring efficiency and cost-effectiveness. Some of the works presented here take up to 30 minutes to generate a model, which is too slow for producing probabilistic results. However, we have also seen some models that can generate models in near real-time.

Unwanted Lighting-Information in textures: In many texture generation approaches, the final results exhibit undesirable lighting effects inherited from the training images. However, for the use of 3D models in a production environment, it is crucial that the textures are free of lighting information, representing only the so-called albedo color. Like mentioned there are already first research attempts that address this challenge, focusing on developing methods to produce lighting-independent textures.

Unnatural textures: In a notable number of results, textures exhibit very high saturation, which significantly reduces the quality of the outcomes and makes them appear unnatural. Addressing this issue is essential to improve the realism and visual appeal of the generated textures.

Unnoptimized Mesh-Topology: A clean topology is crucial for maintaining and dynamically editing meshes, and it can also impact the final quality, such as in vertex base animations. In this paper, we discussed two research works that focus on modeling 3D models as cleanly as a human would. However, most generative models still rely on topology- unaware generation methods.

Limited Datasets: Although the available 3D model datasets are now of considerable size, they still lag significantly behind the datasets used for image generation. This is especially true since the style of the available 3D data is often very similar, limiting diversity. One solution to this problem is to rely more on existing 2D datasets to generate 3D models.

B. Future work

Future work will focus on improving the overall visual quality of 3D models. More importantly, enhancing the stability of 3D models is crucial; the topology must be clean and optimized for further processing and applications, making the generated models suitable for animations. Improving the diversity of generation will also be significant, likely utilizing the large 2D datasets available. Perhaps the most critical area for development is the controllability of the generations. Accurately representing the desired input is essential, but so is the ability to control additional high-precision parameters like aspect, bounds, and feature points.

V. Conclusion

This review has comprehensively examined the current landscape of textured 3D model generation in polygon form using machine learning techniques. Various techniques, including GANs, reinforcement learning, and diffusion models, have been explored to address the challenges associated with 3D modeling, such as generating high-quality textures and detailed geometries.

The integration of neural networks with traditional polygon-based methods has shown promising results, producing models with improved fidelity and complexity. However, challenges remain, particularly in achieving seamless, lighting-less texture mapping and optimizing the balance between computational efficiency and model accuracy.

Future research should focus on refining these machine learning approaches to enhance the quality and realism of 3D models further. Additionally, there is a need for more robust training algorithms that can handle diverse and complex datasets, ensuring that generated models are both accurate and versatile for various applications in computer graphics, robotics, and game development.

Overall, the advancements in machine learning provide a robust foundation for the continued evolution of textured 3D model generation, offering exciting possibilities for future innovations and applications in the field.

References

[1] Alexey Bokhovkin, Shubham Tulsiani, and Angela Dai. Mesh2tex: Generating mesh textures from image queries, 2023.

[2] Tianshi Cao, 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 (ICCV)*, 2023.

[3] Dave Zhenyu Chen, Yawar Siddiqui, Hsin-Ying Lee, Sergey Tulyakov, and Matthias Nießner. Text2tex: Text-driven texture synthesis via diffusion models. *arXiv preprint arXiv:2303.11396*, 2023.

[4] Aysegul Dundar, Jun Gao, Andrew Tao, and Bryan Catanzaro. Fine detailed texture learning for 3d meshes with generative models. *arXiv preprint arXiv:2203.09362*, 2022.

[5] Jun Gao, Tianchang Shen, Zian Wang, Wenzheng Chen, Kangxue Yin, Daiqing Li, Or Litany, Zan Gojcic, and Sanja Fidler. Get3d: A generative model of high quality 3d textured shapes learned from images. In *Advances In Neural Information Processing Systems*, 2022.

[6] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[7] Paul Heckbert. Survey of texture mapping. *Computer Graphics and Applications, IEEE*, 6:56–67, 12 1986.

[8] Ajay Jain, Ben Mildenhall, Jonathan T. Barron, Pieter Abbeel, and Ben Poole. Zero-shot text-guided object generation with dream fields. 2022.

[9] Heewoo Jun and Alex Nichol. Shap-e: Generating conditional 3d implicit functions, 2023.

[10] L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: A survey, 1996.

[11] Nasir Mohammad Khalid, Tianhao Xie, Eugene Belilovsky, and Popa Tiberiu. Clip-mesh: Generating textured meshes from text using pre-trained image-text models. *SIGGRAPH Asia 2022 Conference Papers*, December 2022.

[12] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022.

[13] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[14] Cheng Lin, Tingxiang Fan, Wenping Wang, and Matthias Nießner. Modeling 3d shapes by reinforcement learning, 2020.

[15] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object, 2023.

[16] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

[17] Charlie Nash, Yaroslav Ganin, S. M. Ali Eslami, and Peter W. Battaglia. Polygen: An autoregressive generative model of 3d meshes, 2020.

[18] Michael Oechsle, Lars Mescheder, Michael Niemeyer, Thilo Strauss, and Andreas Geiger. Texture fields: Learning texture representations in function space. In *International Conference on Computer Vision*, October 2019.

[19] Dario Pavllo, Graham Spinks, Thomas Hofmann, Marie-Francine Moens, and Aurelien Lucchi. Convolutional generation of textured 3d meshes. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 870–882. Curran Associates, Inc., 2020.

[20] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion, 2022.

[21] Lyle Regenwetter, Amin Heyrani Nobari, and Faez Ahmed. Deep generative models in engineering design: A review, 2022.

[22] Elad Richardson, Gal Metzer, Yuval Alaluf, Raja Giryes, and Daniel Cohen-Or. Texture: Text-guided texturing of 3d shapes, 2023.

[23] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022.

[24] Jaehyeok Shim, Changwoo Kang, and Kyungdon Joo. Diffusion-based signed distance fields for 3d shape generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20887–20897, June 2023.

[25] Yawar Siddiqui, Justus Thies, Fangchang Ma, Qi Shan, Matthias Nießner, and Angela Dai. Texturify: Generating textures on 3d shape surfaces, 2022.

[26] Rosana Montes Soldado and Carlos Ureña Almagro. An overview of brdf models. 2012.

[27] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *ECCV*, 2018.

[28] Kevin Xie, Jonathan Lorraine, Tianshi Cao, Jun Gao, James Lucas, Antonio Torralba, Sanja Fidler, and Xiaohui Zeng. Latte3d: Large-scale amortized text-to-enhanced3d synthesis. *arXiv preprint arXiv:2403.15385*, 2024.

[29] Jiale Xu, Xintao Wang, Weihao Cheng, Yan-Pei Cao, Ying Shan, Xiaohu Qie, and Shenghua Gao. Dream3d: Zero-shot text-to-3d synthesis using 3d shape prior and text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20908–20918, 2023.

[30] Kim Youwang, Tae-Hyun Oh, and Gerard Pons-Moll. Paint-it: Text-to-texture synthesis via deep convolutional texture map optimization and physically-based rendering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[31] Xianfang Zeng, Xin Chen, Zhongqi Qi, Wen Liu, Zibo Zhao, Zhibin Wang, Bin Fu, Yong Liu, and Gang Yu. Paint3d: Paint anything 3d with lighting-less texture diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4252–4262, 2024.