DreamSurfels: Generative Gaussian Surfels for Geometrically Accurate 3D Content Creation

1st Hwan Heo
NCSOFT Research,
Graphics AI Lab
Sungnam, S. Korea,
hwanheo@ncsoft.com

2nd Sangjun Ahn*
NCSOFT Research,
Graphics AI Lab
Sungnam, S. Korea,
sjahn21@ncsoft.com

***Abstract*—The field of 3D content creation has seen significant progress through optimization-based methods, such as score distillation sampling (SDS). Recently, DreamGaussian attempted to combine SDS sampling and 3D Gaussian splatting to accelerate the generation process but faced limitations in achieving precise geometric accuracy. This paper introduces DreamSurfels, an improved 3D content generation framework that enhances efficiency and geometric accuracy. We replace the 3D Gaussian Splatting in the original DreamGaussian with geometrically accurate Gaussian Surfels. This approach achieves precise radiance fields with cleaner normal and depth reconstructions. The proposed method maintains the rapid generation capabilities of the DreamGaussian, producing high-quality textured meshes from single-view images in just 1 minute, while delivering substantial improvements in geometric fidelity. Experiments validate the performance of DreamSurfels, making it a robust solution for efficient and accurate 3D content generation. These improvements pave the way for more practical applications in industries such as game design and digital media**

***Index Terms*—Radiance Fields, Neural Rendering, Gaussian Splatting, diffusion, SDS, 3D Generation**

I. INTRODUCTION

As the demand for high-quality 3D content in various fields, including entertainment, virtual reality, gaming, and industrial applications, continues to grow, the need for more efficient and accurate methods of 3D content generation has become increasingly apparent. Traditional approaches to creating 3D models, especially those that require high fidelity and precision, have often been time-consuming and computationally expensive. The challenge lies in developing methods that not only deliver high-quality 3D representations but also do so in a time-efficient manner.

One of the most promising advancements in recent years has been the use of diffusion-based techniques, particularly in the realm of 2D image generation. These methods have inspired new approaches to 3D content creation, leveraging the power of 2D models to inform 3D geometry and appearance. A significant breakthrough in this area came with DreamFusion [1], which introduced Score Distillation Sampling (SDS). SDS distills 3D geometry and visual features from state-of-the-art 2D generative models, effectively bridging the gap between high-quality 2D and 3D content creation. This method also paved the way for optimization-based 2D lifting techniques, including the use of Neural Radiance Fields [6] (NeRF), which are capable of representing rich 3D information by capturing detailed light-field data.

Despite the advancements, NeRF-based approaches face significant challenges, particularly in terms of computational efficiency. The high computational cost of rendering and optimizing NeRF models results in long processing times, making it less practical for real-time or large-scale 3D content generation. This limitation has spurred further research into alternative methods that can maintain quality while improving efficiency.

In response to these challenges, DreamGaussian [2] emerged as a solution aimed at reducing the computational burden of 3D content generation. By integrating 3D Gaussian Splatting into the generative process, along with mesh extraction and texture refinement, DreamGaussian significantly reduced the time required to generate 3D models. However, while this approach achieved greater efficiency, it struggled with accurately capturing fine geometric details. The volumetric nature of 3D Gaussian Splatting often led to issues such as blurry outputs and problems with spatial densification, particularly in scenarios involving thin or intricate surfaces.

To address these remaining limitations, we propose **DreamSurfels**, a novel 3D content generation framework designed to enhance both the geometric accuracy and efficiency of the generation process. Inspired by recent advances in 2D Gaussian Splatting [4], DreamSurfels introduces the use of 2D Gaussian primitives—each representing an oriented elliptical disk—to represent 3D scenes. These 2D Gaussian primitives inherently include surface normals, enabling precise surface regularization through normal constraints. This allows for cleaner normal and depth reconstructions, effectively overcoming the geometric inaccuracies seen in previous methods.

DreamSurfels improves upon the 3D Gaussian Splatting method used in DreamGaussian by replacing it with geometrically accurate Gaussian Surfels. This innovation, combined with the introduction of regularization terms such as depth distortion and normal consistency, leads to significantly improved geometric quality in 3D reconstructions. By leveraging the explicit intersection between rays and the Gaussian Surfels, DreamSurfels achieves superior accuracy in representing

* corresponding author

the geometry of complex surfaces, offering a more reliable solution for applications that require high-fidelity 3D content.

Through extensive experiments, we demonstrate that DreamSurfels not only enhances the geometric quality of 3D reconstructions but also reduces the generation time, making it a robust and efficient solution for real-world applications in 3D content creation.

Our key **contributions** are as follows:

- We adapt Gaussian Surfels as a new primitive for the accurate and efficient generation of 3D content.
- We achieve geometrically precise radiance field generation from single-view images in a matter of minutes, significantly improving the speed and quality of the process.

II. Preliminary

Before introducing our method, we review the key concepts of 3D Gaussian Splatting and discuss its challenges.

A. Gaussian Splatting

3D Gaussian Splatting [3] is a method for reconstructing a 3D scene by employing a set of anisotropic and explicit primitive kernels, known as 3D Gaussians. The foundational concept behind 3D Gaussian Splatting is the representation of spatial information through the 3D Gaussians, where the covariance matrix defines the density function of a point p in space. This function utilizes the Gaussian rotation matrix R and the scale matrix S, and is mathematically described as:

$$d(p) = \sum_{g} \alpha_g \exp\left(-\frac{1}{2}(p-\mu_g)^T \Sigma_g^{-1}(p-\mu_g)\right) \quad (1)$$

where μ is the center of the Gaussian, Σ represents the covariance matrix and α is the alpha-blending weight (*i.e.,* the opacity value). Specifically, to maintain the physical meaning of the covariance matrix, the original 3D GS's authors suggest setting the covariance matrix as follows:

$$\Sigma = RSS^{\mathrm{T}}R^{\mathrm{T}} \quad (2)$$

which ensures that the covariance matrix is always positive semi-definite.

By definition, 3D Gaussian Splatting is akin to dense point cloud reconstruction but distinguishes itself by reconstructing space as explicit radiance fields for novel view synthesis. This explicit representation offers several advantages:

1) Speed: Unlike Neural Radiance Fields (NeRF), which require querying a multi-layer perceptron (MLP) to retrieve information, 3D Gaussian Splatting explicitly stores the data of 3D Gaussians. This enables real-time scene rendering at rates exceeding 100 frames per second (fps), eliminating the need to query an MLP.

2) Portability: Since 3D Gaussian Splatting relies solely on rasterization, it is much easier to integrate into game engines and web viewers compared to NeRF. For instance, 3D Gaussian Splatting offers an alternative to SuperSplat in such applications.

3) Editability: The method allows for more direct scene editing, such as selecting, erasing, or merging specific components within a trained scene. This is a significant improvement over NeRF, which poses greater complexity in scene manipulation due to its dependence on MLPs.

B. Surface Reconstruction Problem in Gaussian Splatting

Despite its advantages, 3D Gaussian Splatting (3D GS) presents significant challenges in accurate surface reconstruction. The limitations outlined here motivate the shift towards 2D Gaussian Splatting, which overcomes these issues by improving geometric accuracy.

First, 3D GS has difficulty in learning thin surfaces. 3D GS utilizes a volumetric radiance representation that learns the three-dimensional scale. However, this method struggles with accurately representing thin surfaces, which are often essential in detailed reconstructions.

Second, it absence of surface normals. High-quality surface reconstruction typically relies on surface normals to define the orientation of surfaces in space. In contrast, 3D GS lacks surface normals, which hampers precise surface rendering. While Implicit Neural Networks (INN) address this issue through the use of Signed Distance Functions (SDF), 3D GS does not incorporate this functionality, leading to suboptimal surface fidelity.

Third, it lacks of Multi-View consistency. The rasterization process in 3D GS can introduce artifacts due to inconsistencies between different viewpoints. Specifically, as different 2D intersection surfaces are generated from various perspectives, visual discrepancies and artifacts are likely to occur.

Finally, it suffers from inaccurate affine projection problems. The affine matrix used in 3D GS to convert it into radiance fields suffers from perspective inaccuracies, especially as it deviates from the Gaussian center. This can result in noisy reconstructions and further degrade surface quality.

Moreover, 3D GS shares some of the challenges faced by Neural Radiance Fields (NeRF), particularly the difficulty in generating high-quality meshes. Methods such as Marching Cubes or Poisson Surface Reconstruction often struggle with 3D GS's volumetric opacity accumulation, leading to issues in extracting detailed and accurate surface geometry.

III. Method

We propose the novel DreamSurfel pipeline to remedy the limitations of 3D Gaussian Splatting's inaccurate normal and depth estimations. To this end, DreamSurfel integrates 2D Gaussian Splatting with 3D generation via SDS loss, while maintaining efficient generation of DreamGaussian [2].

A. Gaussian Surfel

2D Gaussian Splatting [4] (*i.e.,* Gaussian Surfels) represents a 3D scene using 2D Gaussian Splat as primitives of the scene. Each 2D Gaussian splat is defined by its central point $\mathbf{p}_k$, two principal tangential vectors $\mathbf{t}_u$ and $\mathbf{t}_v$, and a scaling vector $\mathbf{S} = (s_u, s_v)$. The normal vector of the splat, $\mathbf{t}_w = \mathbf{t}_u \times \mathbf{t}_v$, completes the orthogonal basis. The orientation can be represented by a 3×3 rotation matrix $\mathbf{R} = [\mathbf{t}_u, \mathbf{t}_v, \mathbf{t}_w]$, and the scaling factors by a 3×3 diagonal matrix $\mathbf{S}$ with the third diagonal entry as zero. A 2D Gaussian is thus defined in a local tangent plane in 3D space, parameterized as:

$$\mathbf{P}(u, v) = \mathbf{p}_k + s_u \mathbf{t}_u u + s_v \mathbf{t}_v v = \mathbf{H} \begin{pmatrix} u \\ v \\ 1 \\ 1 \end{pmatrix} \tag{3}$$

where $\mathbf{H} \in \mathbb{R}^{4\times4}$ is defined as a local tangent plane-to-world homogeneous transformation. Subsequently, the Gaussian value at point $\mathbf{u} = (u, v)$ in uv space is evaluated as:

$$G(\mathbf{u}) = \exp\left(-\frac{u^2 + v^2}{2}\right). \tag{4}$$

This approach distributes densities within a planar disk, with the normal vector indicating the steepest density change.

The parameters $\mathbf{p}_k$, (s_u, s_v), and $(\mathbf{t}_u, \mathbf{t}_v)$ are optimized during training. Each 2D Gaussian also has an opacity value α and a color feature $\mathbf{c}$ parameterized with spherical harmonics. All the above optimizable parameters are presented by Θ, where $\Theta_i = \{\mathbf{p}_i, \mathbf{S}_i, \mathbf{R}_i, \alpha_i, \mathbf{c}_i\}$ is the parameter for the i-th Gaussian Surfel.

B. Splatting

Following Huang et al. [4], we also utilize 2D-to-2D projection in our rasterization algorithm. For a given image coordinate (x, y), a ray is defined as the intersection between two homogeneous planes. These planes are represented as:

$$h_x = (-1, 0, 0, x), \quad h_y = (-1, 0, 0, y) \tag{5}$$

The ray intersection is computed by transforming the homogeneous planes h_x and h_y into uv-space. This transformation is achieved using a matrix multiplication with a transformation matrix WH, resulting in:

$$h_u = (WH)^T h_x, \quad h_v = (WH)^T h_y \tag{6}$$

By employing homography, the transformed planes h_u and h_v in uv-space are used to compute the intersection point with the 2D Gaussian splats. The intersection is determined through the following system of equations:

$$h_u \cdot (u, v, 1, 1)^T = 0, \quad h_v \cdot (u, v, 1, 1)^T = 0 \tag{7}$$

The closed-form solution for the u-coordinates and v-coordinates in uv-space is given by:

$$\begin{aligned} u(x) &= \frac{h_{2u}h_{4v} - h_{4u}h_{2v}}{h_{1u}h_{2v} - h_{2u}h_{1v}}, \\ v(x) &= \frac{h_{4u}h_{1v} - h_{1u}h_{4v}}{h_{1u}h_{2v} - h_{2u}h_{1v}} \end{aligned} \tag{8}$$

This solution provides the projection of the screen pixel from uv-space, where the depth z is determined using a previously defined equation. This equation accurately computes the projection of 2D Gaussian splats onto the image space, which is crucial for rendering geometrically precise surfaces.

C. DreamSurfel

To generate **DreamSurfels**, we build upon the foundational optimization framework introduced in DreamGaussian, incorporating several key modifications to improve the accuracy and efficiency of 3D content generation. While the core optimization pipeline remains similar, a major difference lies in the way we regularize the Gaussian Surfels to better capture geometric details.

1) Initialization and Optimization: At the outset, we initialize the Gaussian Surfels as a set of randomly distributed points within a spherical volume. Each Surfel is assigned a random position inside this sphere, with its initial scaling set to unity and no rotational transformation applied. During the optimization process, these Surfels are periodically densified—meaning that more Surfels are added to increase the resolution of the 3D model as necessary. This densification strategy is crucial for maintaining the detail and accuracy of the scene geometry as the optimization progresses.

2) Score Distillation Sampling for 3D Content Generation: As in previous works, we employ Score Distillation Sampling (SDS) to guide the optimization of Surfels. SDS serves as the primary tool for translating 2D image priors into meaningful 3D geometries. For the task of generating 3D content from an input image, we assume the availability of a reference image, denoted as $\tilde{I}_r$, along with a corresponding foreground mask, $\tilde{I}_r^A$, to help distinguish the object of interest from the background.

To leverage the rich information encapsulated in 2D diffusion models, we utilize Zero-1-to-3 XL [5] as our 2D diffusion prior. This diffusion model helps generate plausible 3D structures by predicting noise within the image space, which is then back-propagated to refine the 3D Surfels.

The SDS loss function, central to this optimization, is formulated as follows:

$$\nabla_\Theta \mathcal{L} = \mathbb{E}_{t,p,\epsilon}\left[w(t)\left(\epsilon_\phi(I_p; t, \tilde{I}_r, \Delta p) - \epsilon\right)\frac{\partial I_p}{\partial \Theta}\right] \tag{9}$$

In this equation, $w(t)$ is a weighting function that modulates the importance of different time steps during the optimization. $\epsilon_\phi(\cdot)$ denotes the predicted noise by the 2D diffusion prior, which helps estimate the difference between the current image prediction and the reference image, guiding the Surfels' refinement. Δp refers to the relative change in camera pose from the reference camera, a crucial factor in ensuring the Surfels' 3D alignment with the input image's perspective. Θ represents the parameters of the Surfels that are being optimized.

3) Normal-Consistency Regularization: In addition to the SDS-based optimization, we incorporate a normal-consistency regularization term. This regularization is inspired by techniques used in 2D Gaussian Splatting and aims to produce

Surfels with smoother surface normals and more accurate depth information. By enforcing normal consistency, we reduce artifacts such as noisy surface normals, which can lead to jagged or unrealistic 3D reconstructions.

The normal-consistency loss is given by:

$$\mathcal{L}_n = \sum_i \omega_i (1 - \mathbf{n}_i^\top \mathbf{N}) \tag{10}$$

Here, ω_i is a weighting factor that adjusts the importance of each Surfel based on its contribution to the overall scene. $\mathbf{n}_i$ represents the surface normal of the i-th Surfel. $\mathbf{N}$ is the ground-truth or predicted normal from the global 3D scene structure. This term encourages each Surfel to align its surface normal with the overall scene geometry, leading to cleaner and more realistic 3D shapes.

4) Final Optimization Rule: The total loss function for our method is a combination of the SDS loss and the normal consistency regularization loss. The final objective is:

$$\mathcal{L} = \mathcal{L}_{\text{SDS}} + \lambda \mathcal{L}_n \tag{11}$$

where λ is a hyperparameter that balances the influence of the normal-consistency loss relative to the SDS loss.

By combining these loss components, our method optimizes the positions, scales, and orientations of the Gaussian Surfels, ensuring that they not only conform to the image priors provided by the diffusion model but also produce geometrically accurate and visually coherent 3D structures. This approach allows DreamSurfels to achieve both high geometric precision and computational efficiency, making it a powerful tool for rapid and high-quality 3D content generation from single images.

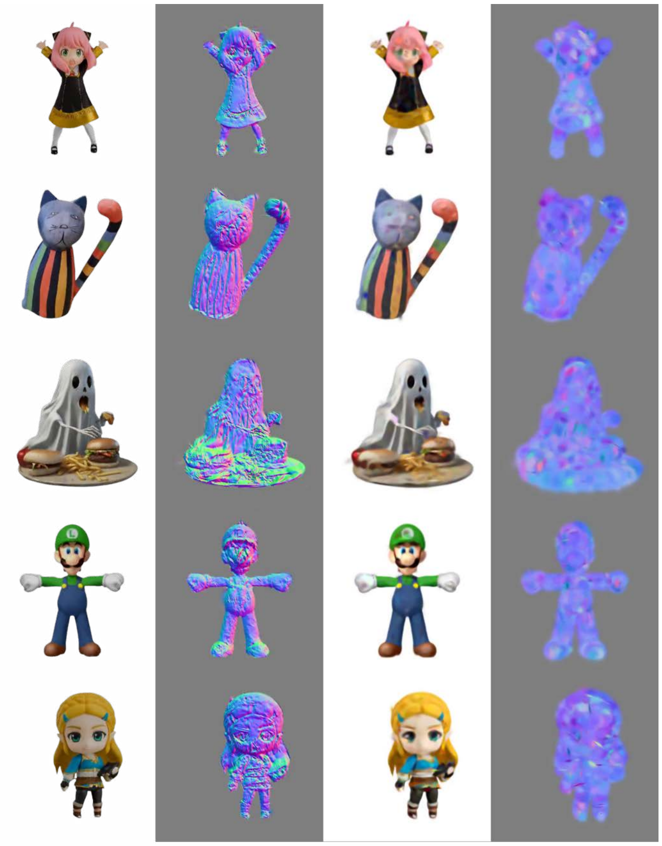
IV. Experiment

Our codebase is built upon the DreamGaussian [2] framework, with most of the experimental settings and hyperparameters aligned with the original implementation. Apart from the changes in the primitive kernel definition and the rasterization module to support Gaussian Surfels, all other aspects, including hyperparameters, remain consistent across both methods.

A. Qualitative Result

We provide qualitative comparisons of image-to-3D generation results in Figure 1. The left side of the figure shows RGB renderings, while the right side displays normal renderings derived from the median depth values.

As shown in Figure 1, our proposed DreamSurfels method demonstrates notable improvements in both appearance fidelity and geometric reconstruction quality compared to DreamGaussian. The enhanced surface details and more accurate normal estimations result in cleaner geometry and a higher-quality 3D asset representation. Hence, It is important to note that the results shown are *before* the refinement stage suggested in DreamGaussian. We are comparing the generated Gaussian scene directly, without any further post-processing or mesh



(a) DreamSurfel (b) DreamGaussian

Fig. 1. Visual comparison between DreamSurfel and DreamGaussian. The left side shows RGB renderings, while the right side displays normal renderings. All results are from Stage 1, *i.e.,* before the refinement stage.

refinement, ensuring a fair comparison between the initial outputs of both methods.

Additionally, our method maintains the rapid generation speed of DreamGaussian, achieving 3D content creation in just one minute, but with significantly improved geometric accuracy. The method can also be extended to fast and accurate TSDF mesh extraction, enabling more efficient mesh refinement in 3D content generation workflows.

B. Quantitative Result

Additionally, we present quantitative performance metrics in Table I. The 1-stage result of our approach achieves a CLIP similarity score of 0.734 and generates content in 1 minute, outperforming several state-of-the-art methods in terms of both efficiency and quality.

The results indicate that our method offers a competitive balance between accuracy and efficiency, providing high-quality textured meshes from single-view images within a minute.

V. Conclusion and Limitations

In this paper, we present DreamSurfels, a novel framework for 3D content generation that enhances both geometric accuracy and efficiency compared to existing methods like

Method	Type	CLIP-Similarity ↑	Time ↓
One-2-3-45	Inference	0.594	45 seconds
Point-E	Inference	0.587	78 seconds
Shap-E	Inference	0.591	27 seconds
Zero-1-to-3	Optimization	0.647	20 minutes
Zero-1-to-3*	Optimization	0.778	30 minutes
DreamGaussian	Optimization	0.678	1 minute
Ours	Optimization	0.734	1 minute

TABLE I

TABLE 1: QUANTITATIVE COMPARISON OF 3D CONTENT GENERATION METHODS. ZERO-1-TO-3* DENOTES THE ORIGINAL ZERO-1-TO-3 WITH A MESH FINE-TUNING STAGE. WE REPORT ONLY THE STAGE 1 RESULT OF OUR RESULT AGAINST TO THE STAGE 1 RESULT OF DREAMGAUSSIAN.

DreamGaussian. By incorporating Gaussian Surfels, our approach enables faster and more precise normal and depth reconstructions, achieving high-quality textured mesh generation in just one minute from single-view inputs. Our experiments show significant improvements in both appearance fidelity and geometric detail, making DreamSurfels a robust solution for 3D content creation.

Despite these advancements, our method still shares some limitations with other optimization-based 3D content generation methods that rely on single-view inputs and do not utilize multi-view diffusion techniques. Specifically, these methods commonly encounter issues such as:

- Multi-Face Janus Problem: Our approach, similar to other single-view optimization-based methods, struggles with accurately handling scenes with multiple faces or complex geometry, often leading to incomplete or inconsistent reconstructions.
- Oversaturated Textures: Due to the reliance on single-view images, there can be oversaturation in texture details, which affects the overall appearance and fidelity of the generated 3D content.
- Baked Lighting: The generated scenes may exhibit baked lighting artifacts, where the lighting is not accurately represented across different viewpoints, resulting in unrealistic lighting effects and shadows.

These limitations are intrinsic to methods that do not leverage multi-view diffusion or other advanced techniques for comprehensive scene understanding and accurate texture representation.

Future work may address this issue by integrating multi-view diffusion models or leveraging advanced generative techniques for depth and normal estimation to further enhance the clarity and sharpness of the generated assets.

VI. ACKNOWLEDGEMENTS

This research was supported by Culture, Sports and Tourism R & D Program through the Korea Creative Content Agency grant founded by the Ministry of Culture, Sports and Tourism in 2024 (Project Name: Development of Copilot Technology for Auto-completion of 3D Webtoon Based on Generative AI, Project Number: RS-2024-00400004, Contribution Rate: 100%)

REFERENCES

[1] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall, "DreamFusion: Text-to-3D using 2D Diffusion", in ICLR, 2022.
[2] J. Tang, J. Ren, H. Zhou, Z. Liu, and G. Zeng, "DreamGaussian: Generative Gaussian Splatting for Efficient 3D Content Creation", in ICLR, 2023.
[3] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3D Gaussian Splatting for Real-Time Radiance Field Rendering," in SIGGRAPH, 2023.
[4] B. Huang, Z. Yu, A. Chen, A. Geiger, and S. Gao, "2D Gaussian Splatting for Geometrically Accurate Radiance Fields," in SIGGRAPH, 2024.
[5] R. Liu, R. Wu, B. Van Hoorick, P. Tokmakov, S. Zakharov, and C. Vondrick, "Zero-1-to-3: Zero-shot One Image to 3D Object," in ICCV, 2023.
[6] B. Mildenhall and P. P. Srinivasan and M. Tancik and J. T. Barron and R. Ramamoorthi and R. Ng, "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis", in ECCV, 2020.
[7] C., Luxi and W., Zhengyi and Li, C. and Gao, T. and Su, Hang and Z., Jun, "MicroDreamer: Zero-shot 3D Generation in 20 Seconds by Score-based Iterative Reconstruction", 2024.
[8] J. Ho, A. Jain, and P. Abbeel, "Denoising Diffusion Probabilistic Models," in *NeurIPS*, 2020.
[9] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-Resolution Image Synthesis with Latent Diffusion Models," in *CVPR*, 2022.
[10] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, *et al.*, "Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding," in *NeurIPS*, 2022.
[11] H. Jun, A. Nichol, and P. Dhariwal, "Shap-E: Generating Conditional 3D Implicit Functions," *arXiv preprint arXiv:2305.02463*, 2023.
[12] A. Nichol, P. Dhariwal, and M. Khadilkar, "Point-E: A System for Generating 3D Point Clouds from Complex Prompts," *arXiv preprint arXiv:2212.08751*, 2022.
[13] A. Jain, B. Mildenhall, J. T. Barron, P. Abbeel, and B. Poole, "Zero-Shot Text-Guided Object Generation with Dream Fields," in *CVPR*, 2022.
[14] C.-H. Lin, J. Gao, L. Tang, T. Takikawa, X. Zeng, *et al.*, "Magic3D: High-Resolution Text-to-3D Content Creation," in *CVPR*, 2023.
[15] I. Ntavelis, T. Vu, A. Osman Ulusoy, O. Hilliges, and T. Birdal, "SDFusion: Multimodal 3D Shape Completion, Reconstruction, and Generation," in *CVPR*, 2023.
[16] Z. Wang, P. Han, Z. Lin, and S. Jiang, "Score Jacobian Chaining: Lifting Pretrained 2D Diffusion Models for 3D Generation," *arXiv preprint arXiv:2304.03442*, 2023.
[17] Z. Yu, J. Wang, Z. Cao, Y. Chen, and S. Shan, "Holistic 3D Scene Generation from Text Prompt," in *ICCV*, 2023.
[18] Z. Huang, J. Bao, D. Chen, F. Wen, and J. Guo, "Layered Neural Atlas: 3D Scene Reconstruction by Fusing Images with Neural Fields," in *CVPR*, 2023.
[19] G. Metzer, E. Tretschk, R. Gal, D. Freedman, and K. Kreis, "Latent-NeRF for Shape-Guided Generation of 3D Shapes and Textures from Images," *arXiv preprint arXiv:2210.04628*, 2022.
[20] Y. Zhuang, Z. Wang, J. Wang, and S. Jiang, "Efficient Text-to-3D Rendering with Layered Depth Priors," in *ICCV*, 2023.
[21] U. Singer, S. Yariv, and T. Dekel, "Text2NeRF: Zero-shot Text-to-Image Translation to Neural Radiance Fields," *arXiv preprint arXiv:2303.12305*, 2023.
[22] A. Raj, J. T. Barron, B. Poole, T. Chan, and S. Fidler, "DreamBooth3D: Subject-Driven Text-to-3D Generation," in *CVPR*, 2023.
[23] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant Neural Graphics Primitives with a Multiresolution Hash Encoding," in *ACM TOG*, 2022.
[24] S. Fridovich-Keil, A. Yu, M. Tancik, Q. Chen, B. Recht, and A. Kanazawa, "Plenoxels: Radiance Fields without Neural Networks," in *CVPR*, 2022.
[25] J. Luiten, G. Kopanas, B. Leibe, and D. Ramanan, "Dynamic 3D Gaussians: Tracking by Persistent Dynamic View Synthesis," in *arXiv preprint arXiv:2308.09713*, 2023.