# Supervised Surface Aligned Gaussian Splatting

**Youbo Shao**
IIIS
Tsinghua University
2023011438
shao-yb23@mails.tsinghua.edu.cn

**Kaixin Zhang**
IIIS
Tsinghua University
2023011430
zhang-kx23@mails.tsinghua.edu.cn

**Yuyang Wu**
IIIS
Tsinghua University
2023010817
yy-wu23@mails.tsinghua.edu.cn

## Abstract

3D Gaussian Splatting (3DGS) [12] has recently become a powerful approach for multiview 3D reconstruction and novel-view synthesis, offering photorealistic rendering and efficient training through differentiable rendering of 3D Gaussians. While 3DGS excels at generating high-quality images, extracting accurate surface meshes remains a significant challenge due to the sparsity of the underlying density field and the lack of geometric priors. Recent methods such as SuGaR [7] address this by introducing surface-aligned regularization, but their reliance on self-supervised normal estimation can lead to suboptimal performance in complex scenes. In this work, we propose a hybrid regularization framework that enhances 3DGS meshability by combining regularization from SuGaR [7] with external normal and depth supervision. Our training strategy gradually transitions from external guidance to self-supervision, improving geometric consistency while preserving rendering fidelity. Experimental results demonstrate that our approach produces smoother and more accurate surface meshes, bridging the gap between photometric reconstruction and geometric understanding. Our code will be released in https://github.com/shaoyoubo/Supervised_Sugar.

## 1 Introduction

3DGS has emerged as a prominent method for multiview 3D reconstruction in recent years. Leveraging the novel representation structure of 3D Gaussians, it enables fast, differentiable rendering on GPUs, introducing an analysis-by-synthesis approach to 3D reconstruction: differentiable rendering serves as the analysis step, comparison with input images as the synthesis step, and backpropagation on Gaussian parameters as the update mechanism. This method not only achieves rapid training and low computational costs but also generates highly photorealistic rendered images.

A primary application of 3DGS is novel-view synthesis—a technique for generating unseen viewpoints of a 3D scene from a set of 2D images. Since its inception, it has inspired numerous extensions, including 4D Gaussian splatting for dynamic scene rendering [28], 3D content creation [22], human avatar modeling [9], and text-to-3D scene generation [4]. Through view synthesis, 3DGS-generated scenes also offer a valuable source of photorealistic synthetic images for training generative AI models, such as diffusion models and GANs. This capability allows practitioners to create virtually unlimited training datasets spanning diverse viewpoints and scene variations.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

Meanwhile, surface mesh extraction represents another critical direction for 3DGS applications. Beyond novel-view synthesis, surface mesh extraction facilitates scene manipulation, enabling editing, sculpting, and animation while preserving temporal consistency. The explicit 3D representation supports object manipulation, viewpoint adjustments, and lighting modifications while maintaining occlusion handling. Geometrically accurate rendering is particularly relevant in AR/VR applications and gaming, where precise 3D scene reconstruction is essential.

However, compared to novel-view synthesis, 3DGS faces greater challenges in surface mesh extraction. The first issue lies in the difficulty of converting 3DGS into a point set. Conventional approaches extract surfaces by applying the Marching Cubes algorithm to select a level set of the density function, followed by Poisson reconstruction to generate a mesh. However, to achieve high-fidelity scene reconstruction, 3DGS significantly increases the number of 3D Gaussians, many of which become extremely small to capture fine details. This results in an overly sparse density function, making it challenging for marching cubes to identify meaningful level sets [7]. The second issue stems from the training process: 3DGS relies solely on photometric supervision from images, lacking geometric priors from the real world. Consequently, it struggles to reconstruct planar surfaces (e.g., walls, floors) and idealized curved structures (e.g., tires, disks), complicating subsequent surface reconstruction.

Our work investigates how to regularize 3D Gaussians to produce smooth planar surfaces that can be converted into meshes while preserving reconstruction fidelity. SuGaR [7] addresses this by introducing a surface-aligned regularization term that encourages Gaussians to align with scene surfaces, facilitating efficient mesh extraction via Poisson reconstruction. However, SuGaR's reliance on self-supervised normal estimation can lead to inaccuracies, especially in texture-less or complex regions.

We combine SuGaR's surface-aligned regularization with external normal supervisions. We propose a hybrid training strategy that initially relies heavily on external normal maps and depth maps, then gradually shifts towards self-supervised learning, balancing the strengths of both approaches. Through systematic comparisons with existing methods, we design a robust regularization framework that balances rendering quality with geometric flatness.

## 2 Related Work

**3D Reconstruction from Multi-view Images.** Recovering 3D geometry of a scene from multi-view images has been a fundamental challenge in computer vision for decades. Traditional multi-view stereo (MVS) techniques [1, 3, 15, 19, 20] typically rely on either feature correspondence for depth prediction [3, 19] or volumetric representations for shape modeling [15, 19]. Modern learning-based MVS approaches often enhance specific components of conventional pipelines, such as feature matching [8, 16], depth integration [5], or direct depth estimation from multi-view inputs [11, 31]. Unlike classical MVS techniques that employ explicit geometric representations, modern neural methods [18, 26, 32, 17, 2, 24, 12] model surfaces implicitly using continuous functions parameterized by multilayer perceptrons (MLPs). These include Signed Distance Fields (SDF) [18, 26, 32], Neural Radiance Fields (NeRF) [17, 2, 24], and 3DGS [12]. Although trained solely on posed 2D images and capable of producing high-quality reconstructions without discretization artifacts, these approaches often exhibit limitations in recovering precise surface geometry. In this study, we demonstrate that integrating monocular depth priors substantially improves reconstruction accuracy in these frameworks.

**Monocular Geometry Estimation.** Monocular Geometry Estimation [6, 27, 30] is a fundamental task in computer vision that involves extracting 3D geometric information—such as depth, surface normals, and sometimes camera pose—from a single 2D image. Because it infers 3D structure from limited information, it is inherently ill-posed and relies heavily on learned priors or constraints. We utility the models for such task to provide external depth supervisions and normal supervisions.

These models typically employ large-scale datasets comprising diverse real-world images or artificially processed 3D models for supervised training. When the dataset is sufficiently extensive and diverse, the model can develop strong generalization capabilities for real-world images, thereby enabling its application to our problem.

**Incorporating Priors into 3DGS.** Several works [14, 7, 23, 25, 29] leverage depth priors and normal priors generated by pre-trained models to supervise the training process of 3DGS. Supervision with

depth is generally straightforward, as depth values can be obtained by computing the weighted depth of Gaussians during rasterization. In contrast, incorporating normal supervision is more challenging. For instance, [14] computes a SDF from the depth map and derives normals by taking its gradients. Other approaches, such as [7, 23], introduce regularization terms to explicitly force the Gaussians to form disc-like surfel representations, which makes estimating normals more tractable. Additionally, some methods add regularization terms into the optimization objective to integrate human geometric intuition, such as surface smoothness and edge sharpness. For example, [7, 23, 10] all propose techniques to enforce Gaussians to adopt disc-like surfel structures. GausSurf [25] segments the scene and applies supervision from depth and normal priors, obtained via classical multi-view stereo (MVS) methods, in texture-rich regions. GSRec [29] locally reconstructs an SDF using 3DGS and introduces regularization terms to enforce smoothness and accuracy of the SDF. Building upon these insights, our work utilizes monocular priors to enhance the accuracy of 3D Gaussian Splatting representations while simultaneously improving surface smoothness in reconstruction.

# 3 Method

## 3.1 3D Gaussian Splatting

To provide a comprehensive overview, we outline the foundational principles of the 3D Gaussian Splatting technique. The scene is modeled as a collection of 3D Gaussians, where each Gaussian $g$ is defined by its position $\mu_g \in \mathbb{R}^3$ and a covariance matrix $\Sigma_g$, which is decomposed into a scaling vector $s_g \in \mathbb{R}^3$ and a rotation quaternion $q_g \in \mathbb{R}^4$. Additionally, every Gaussian has an opacity value $\alpha_g \in [0, 1]$ and spherical harmonics coefficients that encode view-dependent color properties.

Rendering an image from a specific viewpoint involves projecting these 3D Gaussians onto the 2D image plane as splatted 2D Gaussians. This rasterization process is highly efficient, enabling real-time rendering—a significant advantage over slower ray-marching techniques used in Neural Radiance Fields (NeRFs).

The optimization process begins with an initial point cloud from Structure-from-Motion (SfM) [21]. Through gradient descent, the Gaussian parameters—including position, rotation, scale, opacity, and spherical harmonics coefficients—are refined to minimize the difference between rendered and input images. Dynamic adjustments, such as adding, removing, or merging Gaussians, help improve geometric accuracy. As a result, the final representation often consists of millions of finely detailed Gaussians.

## 3.2 Surface-Aligned Gaussian Splatting: SuGaR

Simply put, SuGaR [7] seeks to optimize Gaussian distributions by making them as flat as possible. This optimization bridges the gap between 3DGS representations and explicit surface representations while facilitating surface extraction.

Our methodology incorporates these regularization terms from SuGaR [7]:

**Opacity Entropy Regularization.** This term enforces binary opacity values by minimizing the entropy of each Gaussian's opacity parameter $\alpha_g$. The entropy loss is formulated as $-(\alpha_g \log \alpha_g + (1 - \alpha_g) \log(1 - \alpha_g))$ per Gaussian.

**Flatness Control.** For flattened Gaussians, one scaling factor should be close to zero, and the expected density field is related to the projection of $p$ to the smallest axis $n_{g^*}$ of the nearest gaussian $g^*$: $\overline{d}(p) = \exp(-\frac{1}{2s_{g^*}^2}\langle p - \mu_{g^*}, n_{g^*}\rangle)$, here $s_{g^*}$ is the scaling factor, and $\mu_{g^*}$ is the mean of the Gaussian. Then, this regularization minimizes the discrepancy between the expected density field $\overline{d}(p)$ and the actual field $d(p)$ obtained through rasterization, or the difference between the expected SDF $\overline{f}(p)$ calculated by $\overline{d}(p)$ and the actual SDF obtained through depth map. These differences all happen near the gaussians, i.e. $p$ is close to the nearest Gaussian $g$.

**Normal Continuity Regularization.** This term ensures smooth normal transitions by minimizing differences between neighboring normals. Using K-Nearest Neighbors (KNN), each Gaussian computes a weighted average normal (incorporating distance and rotation metrics) and minimizes its deviation from this expected normal.

### 3.3 Normal Prior Regularization

In SuGaR, the Normal Continuity Regularization ensures smooth normal transitions, but it does not consider corner conditions and produce inaccurate results. In constrast, external supervision by normal priors can provide promising results.

We add an external normal supervision term:

$$R_{\text{ext}} = \frac{1}{|P|} \sum_{p \in P} \|n_p - n_{\text{ext}}(p)\|_2^2$$

here $n_p$ is the smallest axis of the closest Gaussian $g$, and $n_{\text{ext}}(p)$ is the external supervised normal at point $p$.

### 3.4 Depth Prior Regularization

In SuGaR [7], depth regularization is implemented by minimizing the discrepancy between the estimated signed distance function (SDF) $\hat{f}(p)$ and the ideal SDF $f(p)$:

$$R_{\text{SDF}} = \frac{1}{|P|} \sum_{p \in P} \left| \hat{f}(p) - f(p) \right|$$

Here, $\hat{f}(p)$ is derived from the difference between the depth of point $p$ and the depth obtained from the rendered depth map at the projection of $p$, while $f(p)$ represents the ideal SDF computed based on the Gaussian distribution assumptions.

While integrating depth information from pre-trained models could potentially enhance depth supervision, monocular depth estimations often suffer from scale ambiguities and proportional inaccuracies. These inconsistencies can introduce errors when directly used for regularization, unlike normal vectors, which are more reliably estimated.

To mitigate this issue, we propose a loss function that computes the variance of the logarithmic difference between the predicted depth map $\hat{f}$ and the pre-trained depth map $f_{pre}$. This loss dynamically adjusts the scale of the pre-trained depth to match the true scene depth, enabling the model to adaptively correct scale misalignments during training.

$$R_{\text{Depth}} = \frac{1}{|P|} \sum_{p \in P} \left| \log \hat{f}(p) - \log \cdot f_{\text{pre}}(p) \right|^2$$

### 3.5 Total Loss Function

The total loss function combines photometric loss, Flatness Cotrol, Normal Continuity Regularization, Normal Prior Regularization and Depth Prior Regularization:

$$\begin{aligned} \mathcal{L} = \quad & \mathcal{L}_{\text{photometric}} \\ + \quad & \alpha_{\text{flatness}} \lambda_{\text{flatness}}(t) \mathcal{L}_{\text{flatness}} + \alpha_{\text{NC}} \lambda_{\text{NC}}(t) \mathcal{L}_{\text{NC}} \\ + \quad & \alpha_{\text{NP}} \lambda_{\text{NP}}(t) \mathcal{L}_{\text{NP}} + \alpha_{\text{DP}} \lambda_{\text{DP}}(t) \mathcal{L}_{\text{DP}} \end{aligned}$$

Here NC, NP, DP means Normal Continuity Regularization, Normal Prior Regularization and Depth Prior Regularization, respectively. $\alpha_\cdot$ are the weights combine them together. Some of these regularization terms have time-dependent weighting functions $\lambda_\cdot(t)$, because $\mathcal{L}_{\text{flatness}}, \mathcal{L}_{\text{NC}}$ are self-supervised, but $\mathcal{L}_{\text{NP}}$ and $\mathcal{L}_{\text{DP}}$ are external supervision, so the time-dependent factors allow the model to transition from reliance on external supervision to self-supervised learning. This adaptive integration of pre-trained depth information aims to enhance the geometric accuracy of the reconstructed surfaces without being adversely affected by the inherent limitations of monocular depth estimation.

# 4 Experiments

## 4.1 Training Strategy

Our training strategy builds upon the methodology introduced in SuGaR [7], incorporating additional supervision from pre-trained normal and depth maps from MoGe [27] to enhance surface reconstruction. The training process is divided into three distinct phases:

1. **Initial Optimization (7,000 iterations):** We commence by optimizing the 3D Gaussians without any regularization for 7,000 iterations. This phase allows the Gaussians to position themselves based solely on photometric cues, facilitating an initial alignment with the scene geometry.
2. **Opacity Binarization (2,000 iterations):** Subsequently, we introduce an entropy loss on the opacities $\alpha_g$ of the Gaussians for 2,000 iterations. This encourages the opacities to become binary, effectively distinguishing between occupied and free space.
3. **Regularized Optimization (6,000 iterations):** In the final phase, we remove Gaussians with opacity values below 0.5 and perform 6,000 iterations incorporating our proposed regularization terms. These include both self-supervised and supervised components for normal and depth alignment.

## 4.2 Training Details

**Datasets.** Our experiments utilize multiple scenes from the Tanks & Temples benchmark [13], which offers high-resolution video sequences accompanied by precise ground-truth 3D scans obtained via industrial laser scanning systems.

**Baselines.** We adopt the original SuGaR framework [7] as our primary baseline for comparative analysis.

**Experiments.** We systematically evaluate various time-dependent weighting functions for the regularization terms, with all experiments maintaining consistent hyperparameters: $\alpha_{\text{flatness}} = \alpha_{\text{NC}} = \alpha_{\text{NP}} = \alpha_{\text{DP}} = 0.2$. Five distinct weighting configurations are investigated:

- **Baseline:** $\lambda_{\text{flatness}}(t) = \lambda_{\text{NC}}(t) = 1, \lambda_{\text{NP}}(t) = \lambda_{\text{DP}}(t) = 0$.
- **Linear Increase:** $\lambda_{\text{flatness}}(t) = \lambda_{\text{NC}}(t) = 1 - \lambda(t), \lambda_{\text{NP}}(t) = \lambda_{\text{DP}}(t) = \lambda(t)$. $\lambda(t)$ linearly increases from 0 to 1 in $9000 \sim 11000$ iterations, and stay 1 in $11000 \sim 15000$ iterations.
- **Linear Decrease:** $\lambda_{\text{flatness}}(t) = \lambda_{\text{NC}}(t) = 1 - \lambda(t), \lambda_{\text{NP}}(t) = \lambda_{\text{DP}}(t) = \lambda(t)$. $\lambda(t)$ linearly decreases from 1 to 0 in $9000 \sim 11000$ iterations, and stay 0 in $11000 \sim 15000$ iterations.
- **Linear Decrease 2:** $\lambda_{\text{flatness}}(t) = \lambda_{\text{NC}}(t) = 1 - \lambda(t), \lambda_{\text{NP}}(t) = \lambda_{\text{DP}}(t) = \lambda(t)$. $\lambda(t)$ linearly decreases from 1 to 0 in $9000 \sim 13000$ iterations, and stay 0 in $13000 \sim 15000$ iterations.
- **All On:** $\lambda_{\text{flatness}}(t) = \lambda_{\text{NC}}(t) = \lambda_{\text{NP}}(t) = \lambda_{\text{DP}}(t) = 1$.

These functions provide flexibility in controlling the rate at which the model transitions from supervised to self-supervised learning.

**Evaluaton Metrics.** We employ three established metrics for rendering quality assessment: Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), and Learned Perceptual Image Patch Similarity (LPIPS).

## 4.3 Results

Table 1 presents our quantitative results on the resized Truck dataset (251 images, $360 \times 640$ resolution). Notably, our baseline values differ from those reported in the original SuGaR publication, as we only compare the initial three pipeline stages: initial 3DGS training (7000 iterations), coarse refinement (8000 iterations), and mesh extraction.

Our analysis reveals two key findings: First, as demonstrated in Figure 1, our method consistently outperforms the baseline in preserving fine geometric details, particularly evident in the wooden truck bed's row spacing. Second, the Linear Decrease variants extract much flatter surface compared to other configurations.

Table 1: Results on the Tanks&Temples dataset.

| Name | PSNR ↑ | SSIM ↑ | LPIPS ↓ |
|---|---|---|---|
| Baseline (SuGaR) | 23.60 | 0.82 | 0.19 |
| Ours (Linear Increase) | 24.23 | 0.84 | 0.17 |
| Ours (Linear Decrease) | 23.80 | 0.83 | 0.19 |
| Ours (Linear Decrease 2) | 23.84 | 0.83 | 0.18 |
| Ours (All On) | 24.25 | 0.84 | 0.17 |


Figure 1: Render results of 3D Gaussians for different weighting functions.

Figure 2 provides further evidence that while both the baseline and Linear Decrease approaches produce similarly flat surfaces, our method better preserves geometric fidelity. This improvement stems from our two-phase optimization strategy: initial external supervision for Gaussian positioning followed by self-supervised regularization for surface flattening.

The normal maps in Figure 3 offer additional insight: the baseline and Linear Decrease produce coherent surfaces, while Linear Increase and All On exhibit irregular normal distributions. This observation confirms our hypothesis regarding the importance of properly scheduled supervision transitions.

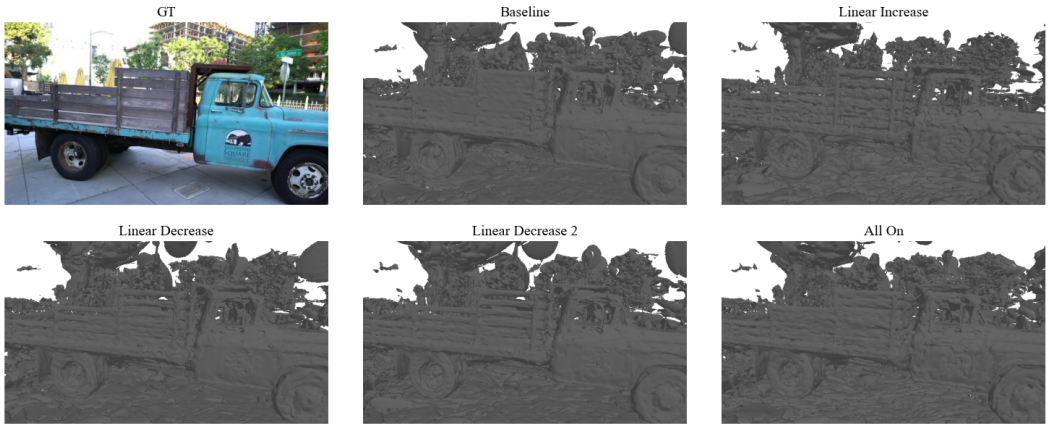


Figure 2: Mesh without texture for different weighting functions.

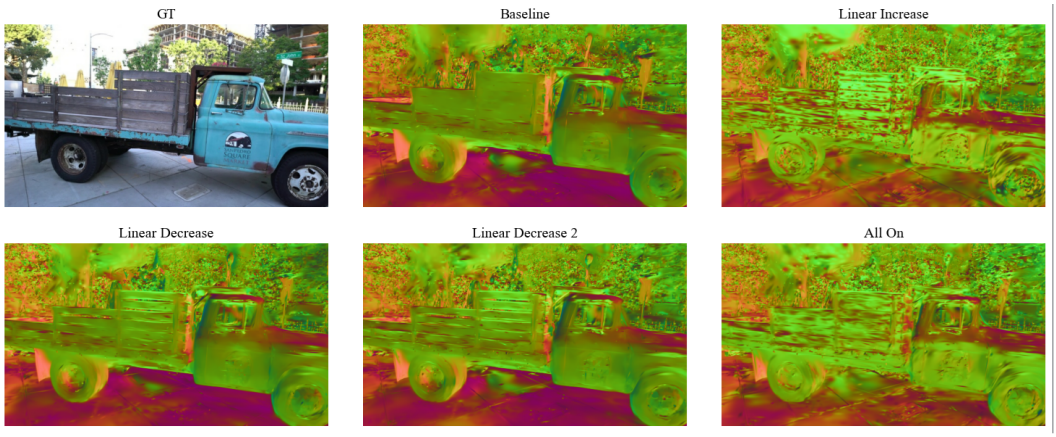


Figure 3: Normal map of mesh for different weighting functions.

## 5 Conclusion

To sum up, we enhance SuGaR's surface-aligned Gaussian splatting framework by introducing additional regularization terms for improved geometric fidelity. While SuGaR optimizes Gaussian flatness and enforces opacity sparsity and normal smoothness, our extensions incorporate external normal supervision and a scale-invariant depth prior to address ambiguities in monocular depth estimation. These refinements further bridge the gap between 3DGS and explicit surface reconstruction, enabling more accurate and robust surface extraction. Future work may explore adaptive fusion of multi-view geometric cues to strengthen generalization.

## References

[1] Motilal Agrawal and Larry S Davis. A probabilistic framework for surface reconstruction from multiple images. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 2, pages II–II. IEEE, 2001.

[2] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5855–5864, 2021.

[3] Michael Bleyer, Christoph Rhemann, and Carsten Rother. Patchmatch stereo-stereo matching with slanted support windows. In *Bmvc*, volume 11, pages 1–11, 2011.

[4] Z. Chen, F. Wang, Y. Wang, and H. Liu. Text-to-3d using gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21401–21412, 2024.

[5] Simon Donne and Andreas Geiger. Learning non-volumetric depth fusion using successive reprojections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7634–7643, 2019.

[6] Ainaz Eftekhar, Alexander Sax, Jitendra Malik, and Amir Zamir. Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3d scans. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10786–10796, 2021.

[7] Antoine Guédon and Vincent Lepetit. Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5354–5363, 2024.

[8] Wilfried Hartmann, Silvano Galliani, Michal Havlena, Luc Van Gool, and Konrad Schindler. Learned multi-patch similarity. In *Proceedings of the IEEE international conference on computer vision*, pages 1586–1594, 2017.

[9] L. Hu, H. Zhang, Y. Zhang, B. Zhou, B. Liu, S. Zhang, and L. Nie. Gaussianavatar: Towards realistic human avatar modeling from a single video via animatable 3d gaussians. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 634–644, 2024.

[10] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 conference papers*, pages 1–11, 2024.

[11] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. Deepmvs: Learning multi-view stereopsis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2821–2830, 2018.

[12] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023.

[13] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 36(4), 2017.

[14] Meenakshi Krishnan, Liam Fowl, and Ramani Duraiswami. 3d gaussian splatting with normal information for mesh extraction and improved rendering. *arXiv preprint arXiv:2501.08370*, 2025.

[15] Kiriakos N Kutulakos and Steven M Seitz. A theory of shape by space carving. *International journal of computer vision*, 38:199–218, 2000.

[16] Vincent Leroy, Jean-Sébastien Franco, and Edmond Boyer. Shape reconstruction using volume sweeping and learned photoconsistency. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 781–796, 2018.

[17] 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. *Communications of the ACM*, 65(1):99–106, 2021.

[18] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174, 2019.

[19] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*, pages 501–518. Springer, 2016.

[20] Steven M Seitz and Charles R Dyer. Photorealistic scene reconstruction by voxel coloring. *International journal of computer vision*, 35:151–173, 1999.

[21] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: Exploring photo collections in 3d. In *ACM SIGGRAPH*, 2006.

[22] J. Tang, Z. Chen, X. Chen, T. Wang, G. Zeng, and Z. Liu. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. In *European Conference on Computer Vision*, pages 1–18. Springer, 2025.

[23] Matias Turkulainen, Xuqian Ren, Iaroslav Melekhov, Otto Seiskari, Esa Rahtu, and Juho Kannala. Dn-splatter: Depth and normal priors for gaussian splatting and meshing. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2421–2431. IEEE, 2025.

[24] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T Barron, and Pratul P Srinivasan. Ref-nerf: Structured view-dependent appearance for neural radiance fields. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5481–5490. IEEE, 2022.

[25] Jiepeng Wang, Yuan Liu, Peng Wang, Cheng Lin, Junhui Hou, Xin Li, Taku Komura, and Wenping Wang. Gaussurf: Geometry-guided 3d gaussian splatting for surface reconstruction. *arXiv preprint arXiv:2411.19454*, 2024.

[26] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021.

[27] Ruicheng Wang, Sicheng Xu, Cassie Dai, Jianfeng Xiang, Yu Deng, Xin Tong, and Jiaolong Yang. Moge: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5261–5271, 2025.

[28] G. Wu, T. Yi, J. Fang, L. Xie, X. Zhang, W. Wei, W. Liu, Q. Tian, and X. Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20310–20320, 2024.

[29] Qianyi Wu, Jianmin Zheng, and Jianfei Cai. Surface reconstruction from 3d gaussian splatting via local structural hints. In *European Conference on Computer Vision*, pages 441–458. Springer, 2024.

[30] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10371–10381, 2024.

[31] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *Proceedings of the European conference on computer vision (ECCV)*, pages 767–783, 2018.

[32] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. *Advances in neural information processing systems*, 35:25018–25032, 2022.