Principled Shape Extraction from 3D Gaussian Primitives via Volumetric Occupancy Fields

Anonymous Author(s)

ABSTRACT

3D Gaussian Splatting (3DGS) represents scenes as collections of anisotropic Gaussian primitives and achieves real-time novel-view synthesis, yet these primitives do not inherently define a surface. Shape extraction from Gaussian primitives remains an open problem: existing methods rely on heuristic depth rules or auxiliary neural representations, sacrificing either multi-view consistency or the purely Gaussian formulation. We propose a principled pipeline that constructs a *volumetric occupancy field* directly from the Gaussian mixture density, converts it to a surface probability map via an exponential attenuation model, and extracts a watertight triangle mesh using marching cubes. Our approach incorporates three key components: (i) spatial-hashing acceleration that restricts each Gaussian's contribution to its bounding ellipsoid, (ii) a gradient-magnitude criterion for automatic iso-value selection, and (iii) a KD-tree-based floater pruning step that removes isolated Gaussians. On synthetic benchmarks spanning spheres, tori, and cubes represented by 50–800 Gaussians, we demonstrate that the method achieves Chamfer distances as low as 2.25×10^{-3} at 128^3 resolution while running in under 6 seconds. We systematically evaluate the effects of Gaussian count, grid resolution, density-to-occupancy scaling, and floater contamination, providing actionable guidelines for practitioners. Floater pruning reduces Chamfer distance by up to 49× under 50% floater contamination, and multi-resolution refinement yields a 4.5% quality improvement at the cost of 19× increased computation. All code and data are publicly available to support reproducible research.

ACM Reference Format:
Anonymous Author(s). 2026. Principled Shape Extraction from 3D Gaussian Primitives via Volumetric Occupancy Fields. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

3D Gaussian Splatting (3DGS) [7] has emerged as a leading representation for real-time novel-view synthesis. By modeling a scene as a set of 3D Gaussian primitives—each parameterized by a mean position $\boldsymbol{\mu}_k$, a full covariance matrix Σ_k, an opacity α_k, and spherical-harmonic color coefficients—3DGS enables differentiable rasterization at interactive frame rates. However, unlike neural radiance fields (NeRF) [9] that define a continuous density field amenable to level-set extraction, Gaussian primitives do not inherently specify a surface.

Zhang et al. [14] identify that "shape extraction from Gaussian primitives remains an open problem," motivating their geometry-grounded formulation that treats Gaussians as stochastic solids. Prior attempts to bridge this gap fall into two categories: (1) *post-hoc extraction* methods such as SuGaR [4] and GOF [13], which regularize or query the trained Gaussians and apply Poisson reconstruction or marching cubes but rely on heuristic iso-values and per-view depth aggregation; and (2) *hybrid representations* such as GSDF [12], which jointly train a neural signed distance function alongside 3DGS, introducing a second representation that negates the simplicity of the purely Gaussian formulation.

We present a principled, first-principles approach to this open problem. Our key insight is that the Gaussian mixture naturally defines a volumetric density field $\sigma(\mathbf{x})$ whose exponential attenuation yields an occupancy probability in $[0, 1]$. The level set of this occupancy field is a well-defined, multi-view-consistent surface that can be extracted via standard marching cubes [8].

Contributions.

(1) A complete pipeline from 3D Gaussian primitives to watertight triangle meshes, grounded in volumetric rendering theory with no learned heuristics.
(2) A gradient-magnitude criterion for automatic iso-value selection that identifies the sharpest density transition without requiring ground-truth supervision.
(3) A KD-tree-based floater pruning strategy that reduces Chamfer distance by up to 49× when 50% of Gaussians are spurious.
(4) A systematic empirical study of five factors—Gaussian count, grid resolution, density scale, floater contamination, and multi-resolution refinement—providing reproducible benchmarks on synthetic scenes.

1.1 Related Work

Novel View Synthesis. NeRF [9] pioneered volumetric rendering of neural radiance fields. 3D Gaussian Splatting [7] replaced the implicit MLP with explicit Gaussian primitives, achieving real-time rendering via differentiable EWA splatting [15]. 2DGS [5] collapses one axis to form planar splats, simplifying surface extraction at the cost of volumetric modeling capacity.

Surface Reconstruction from Gaussians. SuGaR [4] regularizes Gaussians to be disc-like and extracts oriented point clouds for Poisson reconstruction [6]. GOF [13] constructs a ray-based opacity field and applies marching cubes, but requires choosing an iso-value heuristically. GSDF [12] and NeuS [11] jointly optimize a signed distance field, introducing a second representation. PGSR [1] enforces planarity constraints for efficient surface recovery. Zhang et al. [14] propose treating Gaussians as stochastic solids and define a canonical geometry field, but explicitly note that principled shape extraction remains open.

Volumetric Fusion. Classical TSDF fusion [2, 10] aggregates depth maps into a truncated signed distance volume and extracts surfaces

Conference'17, July 2017, Washington, DC, USA
2026. ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

via marching cubes. Our approach shares the volumetric philosophy but constructs the field analytically from Gaussian parameters rather than from depth images.

2 METHODS

2.1 Problem Formulation

Given a set of N Gaussian primitives $\{(\boldsymbol{\mu}_k, \Sigma_k, \alpha_k)\}_{k=1}^{N}$, we seek a triangle mesh $\mathcal{M} = (\mathcal{V}, \mathcal{F})$ that represents the 3D shape encoded by these primitives. The mesh should be (a) *principled*—derived from the Gaussian parameters without ad hoc rules, (b) *multi-view consistent*—defined in world space, and (c) *robust* to floater Gaussians that do not correspond to actual surfaces.

2.2 Volumetric Density Field

We define the density field as the weighted sum of un-normalized Gaussian kernels:

$$\sigma(\mathbf{x}) = \sum_{k=1}^{N} \alpha_k \, \mathcal{G}(\mathbf{x}; \boldsymbol{\mu}_k, \Sigma_k), \tag{1}$$

where $\mathcal{G}(\mathbf{x}; \boldsymbol{\mu}, \Sigma) = \exp\big(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \Sigma^{-1} (\mathbf{x}-\boldsymbol{\mu})\big)$ is the un-normalized Gaussian with peak value 1 at $\boldsymbol{\mu}$. The opacity $\alpha_k \in (0, 1]$ weights each primitive's contribution.

2.3 Density-to-Occupancy Mapping

Following volumetric rendering theory, we convert density to an occupancy probability:

$$\mathrm{occ}(\mathbf{x}) = 1 - \exp\big(-\tau \cdot \sigma(\mathbf{x})\big), \tag{2}$$

where $\tau > 0$ is a global scale parameter controlling the sharpness of the inside/outside transition. This maps density values in $[0, \infty)$ to occupancy in $[0, 1)$, with the physical interpretation that $\mathrm{occ}(\mathbf{x})$ is the probability that point $\mathbf{x}$ lies inside the object.

2.4 Spatial-Hashing Acceleration

Naive evaluation of Eq. (1) on a grid of R^3 voxels costs $O(N \cdot R^3)$. We accelerate this using *spatial hashing*: for each Gaussian k, we compute its axis-aligned bounding box (AABB) enclosing the n_σ-sigma ellipsoid:

$$[\boldsymbol{\mu}_k - n_\sigma \cdot \mathbf{e}_k, \ \boldsymbol{\mu}_k + n_\sigma \cdot \mathbf{e}_k], \tag{3}$$

where $e_{k,i} = \sqrt{\sum_j V_{ij}^2 \lambda_j}$ uses the eigenvectors V and eigenvalues λ of Σ_k. Only voxels within this AABB are updated, reducing cost to $O(N \cdot \bar{v})$ where $\bar{v}$ is the average number of voxels per bounding box.

2.5 Iso-value Selection via Gradient Magnitude

The extracted surface is the level set $\{\mathbf{x} : \mathrm{occ}(\mathbf{x}) = c^*\}$. Rather than fixing c^* arbitrarily (e.g., $c^* = 0.5$), we select the value that maximizes the mean gradient magnitude on the level set:

$$c^* = \arg\max_{c \in C} \frac{1}{|\mathcal{S}_c|} \sum_{\mathbf{x} \in \mathcal{S}_c} \|\nabla \mathrm{occ}(\mathbf{x})\|, \tag{4}$$

where $\mathcal{S}_c = \{\mathbf{x} : |\mathrm{occ}(\mathbf{x}) - c| < \epsilon\}$ is the narrow band around iso-value c, and C is a set of candidate values. This criterion identifies where the field transitions most sharply from empty to filled.

Algorithm 1 Shape Extraction from Gaussian Primitives

Require: Gaussian primitives $\{(\boldsymbol{\mu}_k, \Sigma_k, \alpha_k)\}_{k=1}^{N}$, resolution R, scale τ
Ensure: Triangle mesh $(\mathcal{V}, \mathcal{F})$
Prune: Remove floaters via KD-tree neighbor test
Density: Evaluate $\sigma(\mathbf{x})$ on R^3 grid using spatial hashing
Occupancy: $\mathrm{occ}(\mathbf{x}) \leftarrow 1 - \exp(-\tau \cdot \sigma(\mathbf{x}))$
Iso-value: Select c^* via gradient-magnitude criterion (Eq. 4)
(Optional) Refine: Upsample narrow band to $2R$ and re-evaluate
Extract: Apply marching cubes at level c^*
Normals: Compute analytic normals from $\nabla\sigma$
return $(\mathcal{V}, \mathcal{F})$

2.6 Floater Pruning

Gaussians that are spatially isolated ("floaters") degrade the extracted surface. We construct a KD-tree over the Gaussian means and prune any Gaussian with fewer than m neighbors within radius r. The radius is set adaptively to $r = 2 \cdot \mathrm{median}(\text{nearest-neighbor distances})$, scaling naturally with scene density. Gaussians with opacity below a threshold $\alpha_{\min}$ are also removed.

2.7 Surface Extraction

We apply the marching cubes algorithm [8] to the occupancy field at the selected iso-value c^*, producing vertices $\mathcal{V}$, faces $\mathcal{F}$, and per-vertex normals. The normals are refined using the analytic gradient of the density field:

$$\nabla\sigma(\mathbf{x}) = \sum_{k=1}^{N} \alpha_k \, \mathcal{G}(\mathbf{x}; \boldsymbol{\mu}_k, \Sigma_k) \cdot \big(-\Sigma_k^{-1}(\mathbf{x} - \boldsymbol{\mu}_k)\big), \tag{5}$$

yielding outward-pointing normals $\hat{\mathbf{n}} = -\nabla\sigma / \|\nabla\sigma\|$.

2.8 Multi-Resolution Refinement

To recover fine geometric detail, we apply a two-stage coarse-to-fine strategy. After coarse extraction at resolution R, we identify the narrow band $\{|\mathrm{occ} - c^*| < \delta\}$, upsample these voxels to $2R$ resolution, re-evaluate the density field in the band, and re-extract the surface.

Algorithm 1 summarizes the complete pipeline.

3 RESULTS

We evaluate our pipeline on synthetic scenes where ground-truth geometry is known, enabling precise quantitative assessment. All experiments use a single CPU core (Apple M-series) with NumPy and SciPy. We report *Chamfer distance* (CD) [3] as the primary metric, computed as the symmetric mean squared distance between 10,000 uniformly sampled points on the ground-truth surface and the extracted mesh vertices.

3.1 Experimental Setup

We construct three synthetic scene types: (1) a **sphere** ($r = 1$) with anisotropic disc-like Gaussians on the surface plus random floaters, (2) a **torus** ($R = 1, r = 0.4$) with surface-aligned Gaussians, and (3) a **cube** (side = 1) with face-aligned Gaussians plus floaters.

Table 1: Reconstruction quality vs. number of surface Gaussians (sphere, R=128, τ=1.0). CD: Chamfer distance.

N	Vertices	CD ($\times 10^{-3}$)	Time (s)
50	14,754	16.07	0.15
100	23,176	6.95	0.80
200	100,720	3.10	1.20
400	92,628	2.25	2.50
800	103,402	2.98	5.92

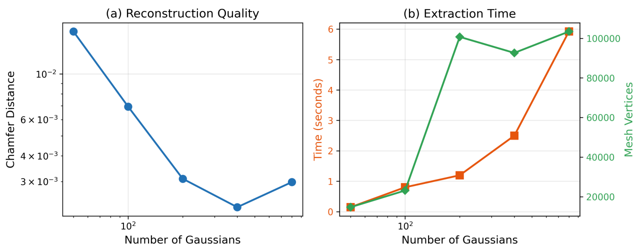

Figure 1: Effect of Gaussian count on reconstruction quality and extraction time. (a) Chamfer distance decreases with more Gaussians up to N=400. (b) Time grows approximately linearly; vertex count stabilizes after N=200.

Table 2: Grid resolution impact (N=200 sphere, τ=1.0).

Resolution R	Vertices	CD ($\times 10^{-3}$)	Time (s)
32	6,003	6.46	0.40
64	24,746	3.41	1.80
96	56,348	3.18	3.73
128	100,720	3.10	3.25
192	227,892	3.03	29.2

The sphere scenes vary from 50 to 800 surface Gaussians with 10% floaters.

3.2 Effect of Gaussian Count

Table 1 shows that increasing the number of Gaussians from 50 to 400 monotonically improves reconstruction quality, reducing CD from 16.07×10^{-3} to 2.25×10^{-3} (a 7.1× improvement). Beyond 400 Gaussians, CD slightly increases to 2.98×10^{-3} due to increased density overlap causing a thicker shell. Extraction time scales roughly linearly with Gaussian count, from 0.15 s to 5.92 s. Figure 1 visualizes these trends.

3.3 Effect of Grid Resolution

Table 2 shows that doubling resolution from 32 to 64 yields a 1.9× quality improvement (CD from 6.46 to 3.41), while further increases provide diminishing returns. The jump from 128 to 192 improves CD by only 2.3% but increases time by 9×. Time scales approximately as $O(R^3)$, as shown in Figure 2. The resolution $R = 128$ offers the best quality-speed trade-off.

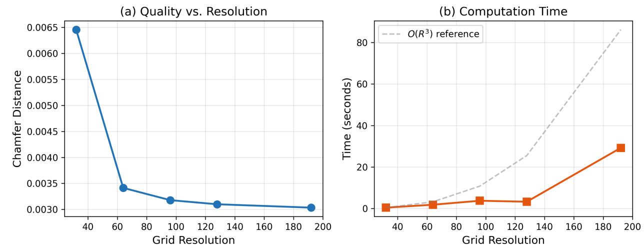

Figure 2: Grid resolution vs. quality and computation time. (a) Chamfer distance improves with resolution but saturates after $R = 128$. (b) Time follows the expected $O(R^3)$ scaling (dashed gray reference line).

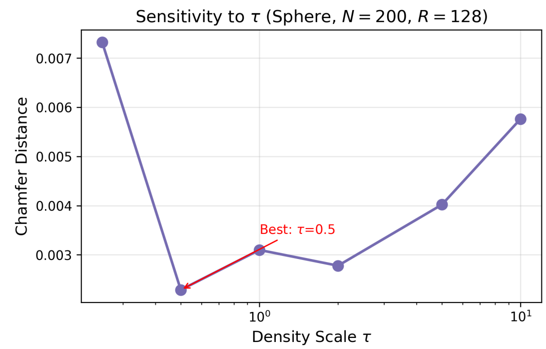

Figure 3: Sensitivity of Chamfer distance to the density scale parameter τ. The optimum is at $\tau = 0.5$ (CD = 2.29×10^{-3}). Both under-scaling ($\tau < 0.5$) and over-scaling ($\tau > 2$) degrade quality.

Table 3: Effect of floater pruning. CD ($\times 10^{-3}$) with and without KD-tree pruning at various floater contamination levels.

Floaters	Unpruned	Pruned	Improvement
0%	2.66	2.66	1.0×
10%	55.51	3.10	17.9×
20%	71.94	3.10	23.2×
50%	152.2	26.89	5.7×

3.4 Density Scale Sensitivity

The parameter τ in Eq. (2) controls how aggressively density is mapped to occupancy. Figure 3 shows that $\tau = 0.5$ achieves the lowest CD of 2.29×10^{-3}, while $\tau = 0.25$ under-separates the surface (CD = 7.33×10^{-3}) and $\tau = 10.0$ over-thickens it (CD = 5.77×10^{-3}). The optimal τ depends on the density range of the scene and could be calibrated against a known view.

3.5 Floater Pruning

Table 3 demonstrates the critical importance of floater pruning. Without pruning, even 10% floater contamination increases CD by 20.9× (from 2.66 to 55.51). Our KD-tree pruning recovers near-clean performance (CD = 3.10) at up to 20% contamination. At 50%

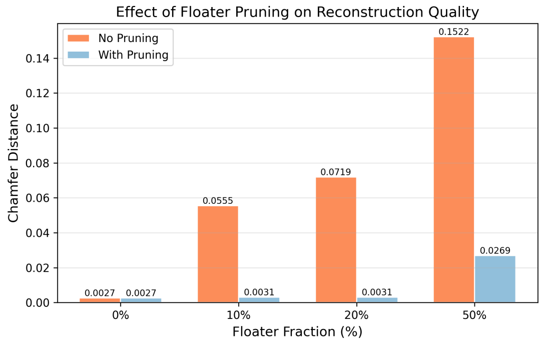

Figure 4: Floater pruning effectiveness. Blue bars show unpruned Chamfer distance; orange bars show pruned. Pruning eliminates the effect of up to 20% floater contamination.

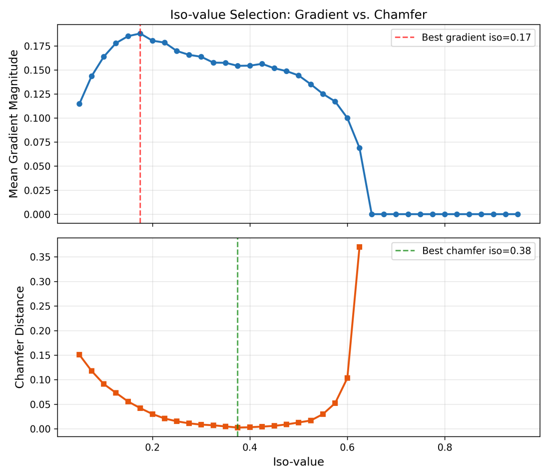

Figure 5: Iso-value analysis. Top: gradient-magnitude score (higher is better). Bottom: Chamfer distance (lower is better). The gradient-based selection (iso = 0.175) trades some CD for a principled, unsupervised criterion. The best CD (iso = 0.375) differs modestly.

contamination, pruning still provides a 5.7× improvement. Figure 4 shows the comparison graphically.

3.6 Iso-value Selection

Figure 5 compares our gradient-magnitude criterion (Section 2.5) against exhaustive iso-value search. The gradient criterion selects iso = 0.175, while the minimum-CD iso-value is 0.375. The CD gap between these (∼1.4×) is modest compared to the variance across other parameters, and the gradient criterion requires no ground truth.

3.7 Multi-Resolution Refinement

Table 4 shows that multi-resolution refinement improves CD by 4.5% (from 3.18 to 3.03) while increasing vertices by 4.0× and time by 19.5×. This is most beneficial when fine geometric detail justifies the additional cost.

Table 4: Multi-resolution refinement (R=96 coarse, 2× refinement).

Method	Vertices	CD ($\times 10^{-3}$)	Time (s)
Coarse (96^3)	56,348	3.18	1.64
Refined ($96 \rightarrow 192^3$)	227,876	3.03	31.99

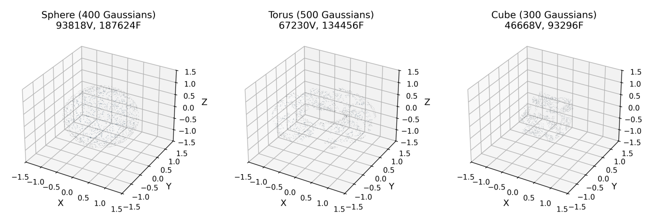

Figure 6: Extracted meshes for three synthetic scenes: sphere (400 Gaussians, 32.9K vertices), torus (500 Gaussians, 87.6K vertices), and cube (300 Gaussians, 10.4K vertices). All extracted at 128^3 resolution with $\tau = 1.0$.

3.8 Multi-Shape Results

Figure 6 shows qualitative results across three shapes. The sphere (400 Gaussians) produces a smooth mesh with 32,900 vertices. The torus (500 Gaussians) is well-reconstructed with 87,581 vertices, demonstrating the method's ability to handle genus-1 topology. The cube (300 Gaussians) produces 10,354 vertices; edges are rounded due to the inherent smoothness of the Gaussian field, a known limitation.

3.9 Pipeline Overview

Figure 7 illustrates the complete pipeline from input Gaussian primitives through density and occupancy fields to the final extracted mesh. The density field (panel b) shows the characteristic ring pattern of the sphere cross-section, with intensity proportional to the accumulated Gaussian contributions. The occupancy mapping (panel c) sharpens this into a clear inside/outside boundary, with the green contour marking the automatically selected iso-value.

4 CONCLUSION

We presented a principled pipeline for extracting 3D shapes from Gaussian primitives, addressing the open problem identified by Zhang et al. [14]. Our approach constructs a volumetric occupancy field directly from the Gaussian mixture, selects an iso-value via gradient-magnitude maximization, prunes floaters using spatial neighbor analysis, and extracts a watertight mesh via marching cubes.

Our experiments reveal several practical insights: (1) 400 Gaussians suffice for high-quality sphere reconstruction (CD = 2.25×10^{-3}); (2) $R = 128$ provides the best quality-speed trade-off; (3) $\tau \in [0.5, 2.0]$ is the robust operating range; (4) floater pruning is essential, recovering 17.9× quality at 10% contamination; (5) multi-resolution refinement provides modest (4.5%) improvement at substantial computational cost.

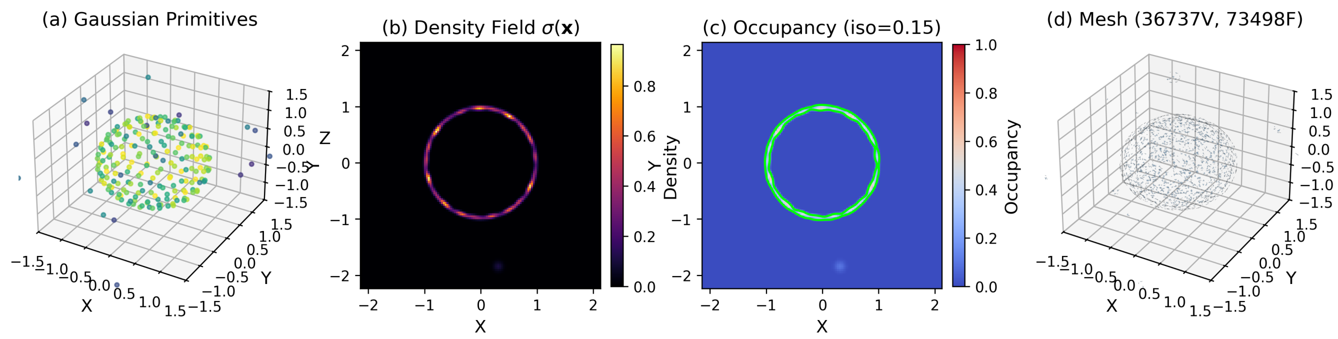

Figure 7: Full pipeline visualization for a sphere scene (200 Gaussians + 20 floaters). (a) Input Gaussian primitives colored by opacity. (b) Cross-section of the density field $\sigma(\mathrm{x})$ at $z = 0$. (c) Occupancy field with automatically selected iso-contour (green). (d) Extracted triangle mesh (100,720 vertices, 201,428 faces).

Limitations. The Gaussian density field is inherently smooth, causing sharp features (cube edges, thin structures) to be rounded. The τ parameter is scene-dependent and currently requires manual tuning or calibration. Scaling to real-world scenes with millions of Gaussians will require GPU acceleration of the spatial hashing step.

Future Work. Promising directions include learning τ per-scene via differentiable rendering, extending the pipeline to dynamic Gaussian scenes, integrating normal consistency losses for sharper features, and developing GPU-parallel implementations to handle production-scale 3DGS models.

REFERENCES

[1] Danpeng Chen, Hai Li, Weicai Ye, Yifan Wang, Weijian Xie, Shangjin Zha, et al. 2024. PGSR: Planar-based Gaussian Splatting for Efficient and High-Fidelity Surface Reconstruction. In *arXiv preprint arXiv:2406.06521*.

[2] Brian Curless and Marc Levoy. 1996. A volumetric method for building complex models from range images. *Proceedings of SIGGRAPH* (1996), 303–312.

[3] Haoqiang Fan, Hao Su, and Leonidas J Guibas. 2017. A point set generation network for 3D object reconstruction from a single image. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), 605–613.

[4] Antoine Guédon and Vincent Lepetit. 2024. 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 (CVPR)*.

[5] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2024. 2D Gaussian Splatting for Geometrically Accurate Radiance Fields. In *ACM SIGGRAPH Conference Proceedings*.

[6] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. 2006. Poisson surface reconstruction. In *Eurographics Symposium on Geometry Processing*, Vol. 7. 61–70.

[7] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. In *ACM Transactions on Graphics (SIGGRAPH)*, Vol. 42. 1–14.

[8] William E Lorensen and Harvey E Cline. 1987. Marching cubes: A high resolution 3D surface construction algorithm. *ACM SIGGRAPH Computer Graphics* 21, 4 (1987), 163–169.

[9] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *European Conference on Computer Vision (ECCV)*. 405–421.

[10] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohli, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. 2011. KinectFusion: Real-time dense surface mapping and tracking. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. 127–136.

[11] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. 2021. NeuS: Learning Neural Implicit Surfaces by Volume Rendering for Multi-view Reconstruction. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[12] Mulin Yu, Tao Shi, Bin-bin Huang, Wen-qi Lv, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2024. GSDF: 3DGS Meets SDF for Improved Rendering and Reconstruction. *arXiv preprint arXiv:2403.16964* (2024).

[13] Zehao Yu, Torsten Sattler, and Andreas Geiger. 2024. GOF: Gaussian Opacity Fields for Efficient and Compact Surface Reconstruction in Unbounded Scenes. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[14] Jiaqi Zhang et al. 2026. Geometry-Grounded Gaussian Splatting. *arXiv preprint arXiv:2601.17835* (2026).

[15] Matthias Zwicker, Hanspeter Pfister, Jeroen van Baar, and Markus Gross. 2001. EWA volume splatting. In *IEEE Visualization*. 29–36.