SurfaceSplat: Connecting Surface Reconstruction and Gaussian Splatting

Zihui Gao[1,3]*, Jia-Wang Bian[2]*, Guosheng Lin[3], Hao Chen[1]†, Chunhua Shen[1]

[1]Zhejiang University, China [2]ByteDance Seed [3]Nanyang Technological University, Singapore

*Equal contribution †Corresponding author

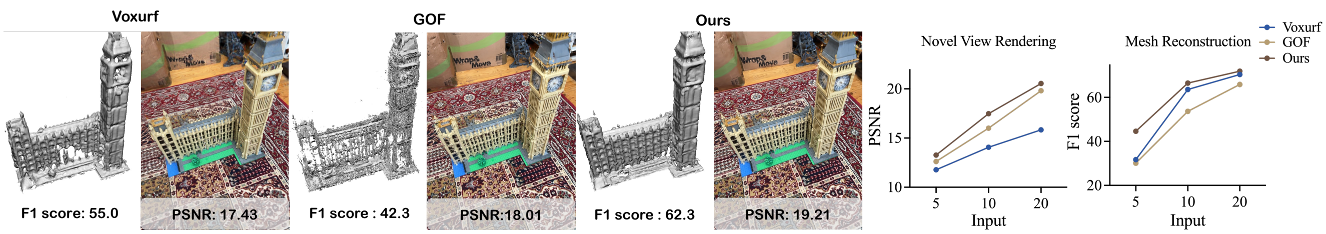


Figure 1. **Sparse view reconstruction and rendering comparison**. *Left:* Qualitative results from 10 images evenly sampled from a casually captured 360-degree video. *Right:* Quantitative analysis of 5, 10, and 20 input views, averaged across the selected 9 MobileBrick test scenes. 3DGS-based methods (e.g., GOF) achieve superior novel view rendering than SDF-based methods (e.g., Voxurf) due to their sparse representations, which capture fine details. However, SDF-based methods outperform the former in mesh reconstruction, as their dense representations better preserve global geometry. Our approach combines the strengths of both, achieving optimal performance.

Abstract

Surface reconstruction and novel view rendering from sparse-view images are challenging. Signed Distance Function (SDF)-based methods struggle with fine details, while 3D Gaussian Splatting (3DGS)-based approaches lack global geometry coherence. We propose a novel hybrid method that combines the strengths of both approaches: SDF captures coarse geometry to enhance 3DGS-based rendering, while newly rendered images from 3DGS refine the details of SDF for accurate surface reconstruction. As a result, our method surpasses state-of-the-art approaches in surface reconstruction and novel view synthesis on the DTU and MobileBrick datasets. Code will be released at: https://github.com/aim-uofa/SurfaceSplat.

1. Introduction

3D reconstruction from multi-view images is a core problem in computer vision with applications in virtual reality, robotics, and autonomous driving. Recent advances in Neural Radiance Fields (NeRF) [25] and 3D Gaussian Splatting (3DGS) [17] have significantly advanced the field. However, their performance degrades under sparse-view conditions, a common real-world challenge. This paper tackles sparse-view reconstruction to bridge this gap. Unlike approaches that leverage generative models [10, 39, 40, 43] or learn geometry priors through large-scale pretraining [6, 21, 29, 49], we focus on identifying the optimal 3D representations for surface reconstruction and novel view synthesis.

Surface reconstruction methods primarily use the Signed Distance Function (SDF) or 3DGS-based representations. Here, SDF-based approaches, such as NeuS [36] and Voxurf [41], model scene geometry continuously with dense representations and optimize them via differentiable volume rendering [25]. In contrast, 3DGS-based methods like GOF [53] and 2DGS [15] leverage a pre-computed sparse point cloud for image rendering and progressively densify and refine it through differentiable rasterization. Due to their dense representations, SDF-based methods capture global structures well but lack fine details, while the sparse nature of 3DGS-based methods enables high-frequency detail preservation but compromises global coherence. As a result, both approaches struggle with poor reconstruction quality under sparse-view conditions. Typically, SDF-based methods outperform 3DGS in surface reconstruction, while 3DGS excels in image rendering, as illustrated in Fig. 1.

Recognizing the complementary strengths of SDF-based (dense) and 3DGS-based (sparse) representations, we propose a novel hybrid approach, SurfaceSplat, as illustrated in Fig. 2. Our method is built on two key ideas: (i) **SDF for Improved 3DGS**: To address the limitation of 3DGS in learning global geometry, we first fit the global structure using an SDF-based representation, rapidly generating a smooth yet coarse mesh. We then initialize 3DGS by sampling point clouds from the mesh surface, ensuring global consistency while allowing 3DGS to refine fine details during training. (ii) **3DGS for Enhanced SDF**: To compensate for the inability of SDF-based methods to capture fine details under sparse-view settings, we leverage the improved 3DGS from the first step to render additional novel viewpoint images, expanding the dataset. This enriched supervision helps the SDF-based method learn finer structural details, leading to improved reconstruction quality.

We conduct experiments on two real-world datasets, DTU [16] and MobileBrick [19]. Our method, SurfaceSplat, achieves state-of-the-art performance in sparse-view novel view rendering and 3D mesh reconstruction. In summary, we make the following contributions:

- We propose SurfaceSplat, which synergistically combines the strengths of SDF-based and 3DGS-based representations to achieve optimal global geometry preservation while capturing fine local details.
- We conducted a comprehensive evaluation and ablations on DTU and MobileBrick datasets. SurfaceSplat achieves state-of-the-art performance in novel view synthesis and mesh reconstruction under sparse-view conditions.

2. Related work

2.1. Novel View Synthesis from Sparse Inputs

Neural Radiance Fields (NeRFs)-based methods [2, 3, 5, 7, 12, 25, 26, 33, 35, 45, 55] have revolutionized novel view synthesis with implicit neural representations, and 3DGS-based methods [17, 23, 32, 34, 44, 47, 52] enable efficient training and real-time rendering through explicit 3D point clouds. However, both approaches suffer from performance degradation in sparse-view settings. To address this issue, recent methods have explored generative models [10, 39, 40, 43] or leveraged large-scale training to learn geometric priors [6, 21, 29, 49]. Unlike these approaches, we argue that the key challenge lies in the lack of effective geometric initialization for 3DGS. To overcome this, we investigate how neural surface reconstruction methods can enhance its performance.

2.2. Neural Surface Reconstruction

SDF-based methods, such as NeuS [36], VolSDF [46], Neuralangelo [20], and PoRF [4] use dense neural representations and differentiable volume rendering to achieve high-quality reconstructions with 3D supervision. However, they suffer from long optimization times and require dense viewpoint images. Recent methods, such as 2DGS [15] and GOF [53], extend 3DGS [17] by leveraging modified Gaussians and depth correction to accelerate geometry extraction. While 3DGS-based methods [1, 8, 13, 15, 18, 37, 53, 54] excel at capturing fine local details, their sparse representations struggle to maintain global geometry, leading to incomplete and fragmented reconstructions. This paper focuses on integrating the strengths of both representations to achieve optimal neural surface reconstruction.

2.3. Combing 3DGS and SDF

Several recent approaches have integrated SDF-based [27, 28] and 3DGS-based representations to improve surface reconstruction. NeuSG [9] and GSDF [50] jointly optimize SDF and 3DGS, enforcing geometric consistency (e.g., depths and normals) to improve surface detail [14]. Similarly, 3DGSR [24] combines SDF values with Gaussian opacity in a joint optimization framework for better geometry. While effective in dense-view settings, these methods struggle to reconstruct high-quality structures under sparse-view conditions, as shown in our experiments in Sec. 4. Our approach specifically targets sparse-view scenarios by leveraging a complementary structure to enhance both rendering and reconstruction quality.

3. Method

Our method takes sparse viewpoint images with camera poses as input, aiming to reconstruct 3D geometry and color for novel view synthesis and mesh extraction. Fig. 2 provides an overview of SurfaceSplat. In the following sections, we first introduce the preliminaries in Sec. 3.1, then explain how SDF-based mesh reconstruction improves 3DGS for novel view synthesis in Sec. 3.2, and finally describe how 3DGS-based rendering enhances SDF-based surface reconstruction quality in Sec. 3.3.

3.1. Preliminaries

SDF-based representation. NeuS [36] proposes to model scene coordinates as signed distance function (SDF) values and optimize using differentiable volume rendering, similar to NeRF [25]. After optimization, object surfaces are extracted using the marching cubes algorithm [22]. To render a pixel, a ray is cast from the camera center o through the pixel along the viewing direction v as $\{p(t) = o+tv|t \geq 0\}$, and the pixel color is computed by integrating N sampled points along the ray $\{p_i = o + t_i v|i = 1, ..., N, t_i < t_{i+1}\}$ using volume rendering:

$$\hat{C}(r) = \sum_{i=1}^{N} T_i \alpha_i c_i, \;\; T_i = \prod_{j=1}^{i-1}(1 - \alpha_j), \tag{1}$$

where α_i represents opacity and T_i is the accumulated transmittance. It is computed as:

$$\alpha_i = \max\left(\frac{\Phi_s(f(p(t_i))) - \Phi_s(f(p(t_{i+1})))}{\Phi_s(f(p(t_i)))}, 0\right), \tag{2}$$

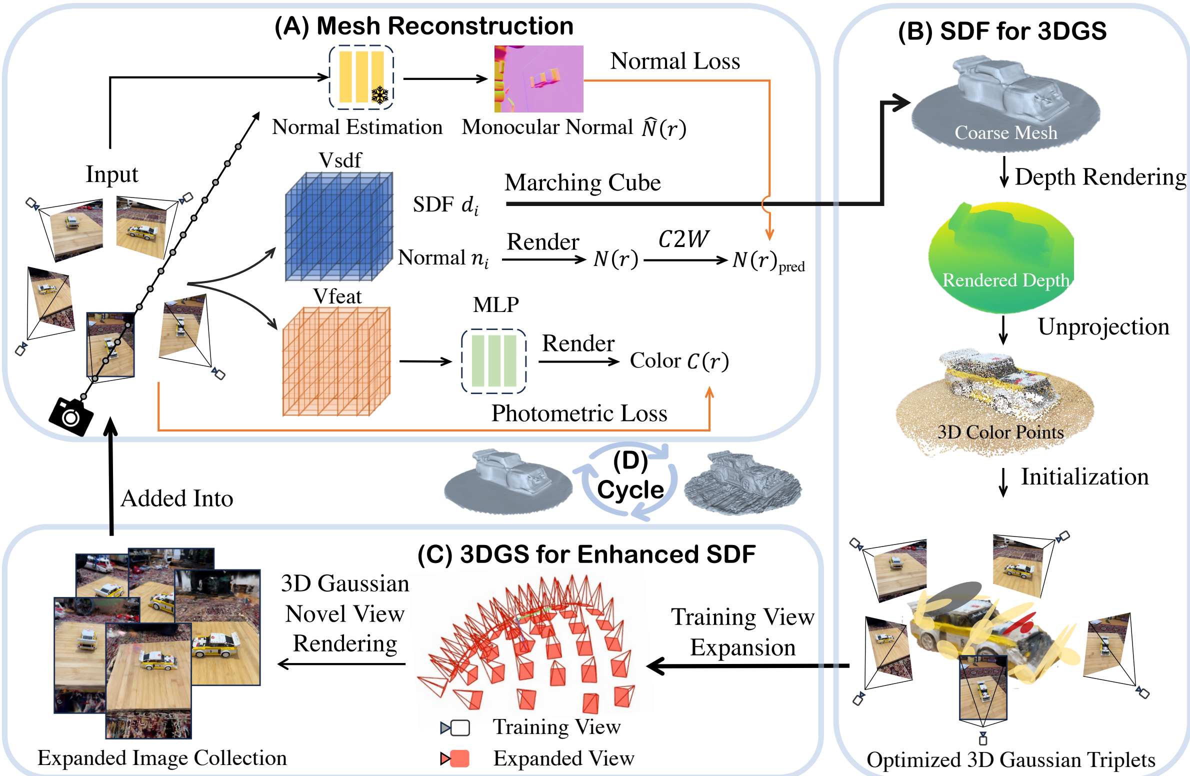


Figure 2. Overview of the proposed SurfaceSplat. (A) We reconstruct a coarse mesh using an SDF-based representation. (B) Point clouds are sampled from the mesh surface to initialize 3DGS. (C) 3DGS renders new viewpoint images to expand the training set, refining the mesh. (D) Steps B and C can be repeated for iterative optimization, progressively improving performance.

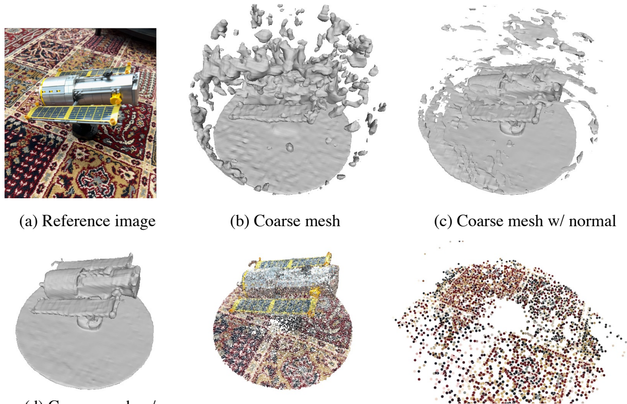
where $f(x)$ is the SDF function and $\Phi_s(x) = (1+e^{-sx})^{-1}$ is the Sigmoid function, with s learned during training. Based on this, Voxurf [41] proposes a hybrid representation that combines a voxel grid with a shallow MLP to reconstruct the implicit SDF field. In the coarse stage, Voxurf [41] optimizes for a better overall shape by using 3D convolution and interpolation to estimate SDF values. In the fine stage, it increases the voxel grid resolution and employs a dual-color MLP architecture, consisting of two networks: g_{geo}, which takes hierarchical geometry features as input, and g_{feat}, which receives local features from $\boldsymbol{V}^{(\text{feat})}$ along with surface normals. We incorporate Voxurf in this work due to its effective balance between accuracy and efficiency.

3DGS-based representation. 3DGS [17] models a set of 3D Gaussians to represent the scene, which is similar to point clouds. Each Gaussian ellipse has a color and an opacity and is defined by its centered position x (mean), and a full covariance matrix Σ: $G(x) = e^{-\frac{1}{2}x^T\Sigma^{-1}x}$. When projecting 3D Gaussians to 2D for rendering, the splatting method is used to position the Gaussians on 2D planes, which involves a new covariance matrix Σ' in camera coordinates defined as: $\Sigma' = JW\Sigma W^T J^T$, where W denotes a given viewing transformation matrix and J is the Jacobian of the affine approximation of the projective transformation. To enable differentiable optimization, Σ is further decomposed into a scaling matrix S and a rotation matrix R: $\Sigma = RSS^T R^T$.

3.2. SDF for Improved 3DGS

3DGS [17] typically initializes with sparse point clouds estimated by COLMAP [31], which are often inaccurate or missing in low-texture or little over-lapping regions. To address this, we propose initializing 3DGS by uniformly sampling points from a mesh surface derived from a SDF representation, ensuring high-quality novel view rendering while preserving global geometry. Below, we detail our proposed method for mesh reconstruction, mesh cleaning, and point cloud sampling. A visual example of the reconstructed meshes and sampled points is shown in Fig. 3.

Coarse mesh reconstruction. Given M sparse images $\{\mathcal{I}\}$ and their camera poses $\{\boldsymbol{\pi}\}$, our objective is to reconstruct a 3D surface for sampling points. As our focus is on robust global geometry rather than highly accurate surfaces, and to ensure efficient mesh reconstruction,



(a) Reference image (b) Coarse mesh (c) Coarse mesh w/ normal

(d) Coarse mesh w/ normal and clean (e) Color points sampling (f) COLMAP sparse points

Figure 3. Visualization of our mesh reconstruction, cleaning, and point sampling. (b) Naïve coarse mesh reconstruction following Voxurf [41]. (c) Coarse mesh reconstructed with our proposed normal loss, reducing floaters. (d) Post-processed mesh with both normal loss and our cleaning methods. (e) Our sampled point clouds used for initializing 3DGS. (f) COLMAP-estimated point clouds, typically used for 3DGS initialization.

we adopt the coarse-stage surface reconstruction from Voxurf [41]. Specifically, we use a grid-based SDF representation $\boldsymbol{V}^{(\text{sdf})}$ for efficient mesh reconstruction. For each sampled 3D point $\mathbf{x} \in \mathbb{R}^3$, the grid outputs the corresponding SDF value: $\boldsymbol{V}^{(\text{sdf})} : \mathbb{R}^3 \rightarrow \mathbb{R}$. We use differentiable volume rendering to render image pixels $\hat{C}(r)$ and employs image reconstruction loss to supervise. The loss function $\mathcal{L}$ is formulated as:

$$\mathcal{L} = \mathcal{L}_{\text{recon}} + \mathcal{L}_{TV}\left(V^{(\text{sdf})}\right) + \mathcal{L}_{\text{smooth}}\left(\nabla V^{(\text{sdf})}\right), \quad (3)$$

where the reconstruction loss $\mathcal{L}_{\text{recon}}$ calculates photometric image rendering loss, originating from both the g_{geo} and g_{feat} branches. The $\mathcal{L}_{TV}$ encourages a continuous and compact geometry, while the smoothness regularization $\mathcal{L}_{\text{smooth}}$ promotes local smoothness of the geometric surface. We refer to Voxurf [41] for the detailed implementation of the loss functions. The coarse reconstruction typically completes in 15 minutes in our experiments.

Due to the limited number of training views, the learned grid often exhibits floating artifacts, as shown in Fig. 3 (b), which leads to incorrect point sampling. To mitigate this, we introduce a normal consistency loss to improve training stability, effectively reducing floaters and smoothing the geometric surface. Our approach leverages the predicted monocular surface normal $\hat{N}(\mathbf{r})$ from the Metric3D model [48] to supervise the volume-rendered normal $\bar{N}(\mathbf{r})$ in the same coordinate system. The formulation is:

$$\mathcal{L}_{\text{normal}} = \sum \left(\|\hat{N}(\mathbf{r}) - \bar{N}(\mathbf{r})\|_1\right). \quad (4)$$

We integrate this loss with Eqn. (3) during training to effectively remove floaters. Fig. 3 (c) shows a coarse mesh reconstructed with the normal loss, demonstrating improved surface smoothness and reduced artifacts.

Mesh cleaning. Even though the proposed normal loss significantly reduces floaters, some still persist, adding noise to the subsequent 3DGS initialization. To mitigate this, we apply a mesh cleaning step that refines the coarse mesh by removing non-main components. Specifically, we first use Marching Cube algorithm [38] to extract triangle mesh $\mathcal{M} = (\mathcal{V}, \mathcal{F})$ from SDF grid $\boldsymbol{V}^{(\text{sdf})}$. Then we cluster the connected mesh triangles to $\{\mathcal{F}_i\}$, identify the largest cluster index: $|\mathcal{F}_{i_{\max}}| = \max(|\mathcal{F}_i|)$ and get remove parts

$$\mathcal{F}_{\text{remove}} = \{f \in \mathcal{F} \mid f \notin \mathcal{F}_{i_{\max}}\}. \quad (5)$$

Finally, we filter the floaters $\mathcal{F}_{\text{remove}}$ from $\mathcal{M}$, resulting in $\mathcal{M}_1 = \mathcal{M} \setminus \mathcal{F}_{\text{remove}}$. Fig. 3 (d) illustrates the refined mesh after applying our cleaning method.

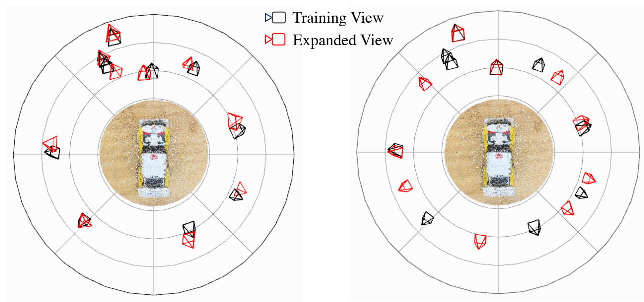
Sampling surface points for 3DGS. Since the mesh obtained from Marching Cubes includes regions that are invisible from the training views, directly sampling points from the mesh surface can introduce noise into 3DGS. To mitigate this, we propose a depth-based sampling strategy. First, we project the reconstructed mesh onto the training views using their known camera poses to generate depth maps $\{\mathcal{D}\}$. Since these depth maps originate from a 3D mesh, they maintain multi-view consistency. We then randomly sample points from valid depth regions, ensuring they correspond to visible object surfaces. The sampled pixels (u, v), along with their depth values $d(u, v)$, are back-projected to colorized 3D points $\mathbf{P} = \{(x_i, y_i, z_i) \mid i = 1, 2, \ldots, N\}$ using the following formulation:

$$\begin{bmatrix} x_i & y_i & z_i \end{bmatrix} = \boldsymbol{\pi_k} \mathbf{K}^{-1} \begin{bmatrix} d \cdot u & d \cdot v & d \end{bmatrix}^T. \quad (6)$$

This approach ensures that the sampled points are uniformly distributed on the object's surface while remaining visible in the training views, leading to a more stable and accurate 3DGS initialization. As our reconstructed mesh primarily covers foreground regions, we combine our sampled point cloud with COLMAP sparse points when rendering background regions, serving as the initialization for 3DGS. Fig. 3 (e) and (f) illustrate our sampled point clouds and COLMAP-estimated point clouds, respectively.

3.3. 3DGS for Enhanced SDF

We argue that the primary bottleneck for SDF-based mesh reconstruction is insufficient supervision due to limited training views. To address this, we generate additional novel viewpoint images using a 3DGS-based method and combine them with the original sparse views to enhance the training of SDF-based reconstruction.



(a) Camera position perturbation (b) Camera poses interpolation

Figure 4. Top-view visualization of pose expansion strategies.

Rendering novel viewpoint images. We utilize the improved 3DGS, initialized with our proposed mesh-based point sampling method, to render images. Thanks to our robust and dense point initialization, the 3D Gaussian $\mathcal{G}$ can converge after $7k$ iterations in just 5 minutes, yielding $\mathcal{G} = f(\mathbf{P}, \{I\}, \{\boldsymbol{\pi}\})$. Given new camera poses $\{\boldsymbol{\pi}_{\text{new}}\}$, the 3D Gaussian $\mathcal{G}$ can be projected to generate novel-view images as follows:

$$\{\mathcal{I}_{\text{new}}\} = \text{Splat}(\mathcal{G}, \{\boldsymbol{\pi}_{\text{new}}\}). \tag{7}$$

The newly rendered images $\{\mathcal{I}_{\text{new}}\}$ are combined with the input images $\{\mathcal{I}\}$ to train the SDF-based mesh reconstruction. The key challenge lies in selecting new camera viewpoints $\{\boldsymbol{\pi}_{\text{new}}\}$ that best enhance surface reconstruction:

$$\{\boldsymbol{\pi}_{\text{new}}\} = g\left(\{\boldsymbol{\pi}\}\right) \tag{8}$$

where g is our pose expansion strategy. To ensure new viewpoints remain consistent with the original pose distribution and avoid excessive deviation that could blur or diminish the foreground, we explore two methods for generating new camera poses. Fig. 4 shows the generated pose position.

Camera position perturbation. To generate new camera positions while preserving proximity to the original distribution, a perturbation $\Delta\mathbf{p}$ is applied to the initial camera positions $\{\boldsymbol{c}\}$. The new camera centers $\{\boldsymbol{c}'_m\}$ are computed:

$$\boldsymbol{c}'_m = \boldsymbol{c} + \Delta\mathbf{p}, \tag{9}$$

where $\Delta\mathbf{p} = (\Delta x, \Delta y, \Delta z)$ represents a controlled offset vector designed to modulate the new viewpoints.

Camera pose interpolation. Our method takes a set of camera rotation matrices $\{\mathbf{R}\}$ and camera positions $\{\boldsymbol{c}\}$ as input. To generate smooth transitions between viewpoints, we employ cubic spline interpolation [11]. This approach interpolates both camera positions and orientations, producing interpolated camera centers $\{\boldsymbol{c}'_m\}$ and rotation matrices $\{\mathbf{R}'_m\}$ that ensure visual continuity and positional coherence. By maintaining these properties, the newly generated camera poses facilitate high-quality transitions, making them well-suited for 3D mesh reconstruction. The visualizations of the images generated from new viewpoints can be found in Fig. 2 of the supplementary material.

Refining surface reconstruction. We reuse the reconstructed coarse mesh and refine it with the original and expanded novel viewpoint images. Following the fine-stage reconstruction of Voxurf [41], we increase the grid resolution and introduce a dual color network and hierarchical geometry features for detailed surface reconstruction.

3.4. Cyclic Optimization

We propose an interactive optimization process, which begins by generating an initial coarse mesh $\mathcal{M}^{(0)}$. Then, in each iteration n, the process follows two steps:

1. Rendering Step: We optimize a 3DGS model for rendering novel view images, which is initialized by sampling points from the current coarse mesh $\mathcal{M}_c^{(n)}$, represented by:

$$\mathcal{I}^{(n)} = \mathcal{R}\left(\mathcal{M}_c^{(n)}\right) \tag{10}$$

2. Meshing Step: We refine the current mesh by fine-tuning it using both the newly rendered images and the original input images:

$$\mathcal{M}_f^{(n)} = \mathcal{O}\left(\mathcal{M}_c^{(n)}, \mathcal{I}^{(n)}\right) \tag{11}$$

where $\mathcal{O}$ represents the SDF grid optimization. Then, we update the refined mesh:

$$\mathcal{M}_c^{(n+1)} = \mathcal{M}_f^{(n)}. \tag{12}$$

By iterating this process, our method allows SDF-based reconstruction and 3DGS-based rendering to complement each other, improving both reconstruction accuracy and novel view synthesis. To balance efficiency and accuracy, we typically perform only one iteration.

4. Experiments

4.1. Experimental Setup

Datasets. We conduct a comprehensive evaluation of the proposed method on the MobileBrick [19] and DTU [16] datasets. MobileBrick is a multi-view RGB-D dataset captured on a mobile device, providing precise 3D annotations for detailed 3D object reconstruction. Unlike the DTU dataset, which is captured in a controlled lab environment, MobileBrick represents more challenging, real-world conditions, making it more reflective of everyday scenarios. Following previous methods [19, 36, 41], we use 15 test

Table 1. Surface reconstruction and novel view synthesis results on MobileBrick. The results are averaged over all 18 test scenes with an initial input of 10 images per scene. PSNR-F is computed only on foreground regions. The best results are **bolded**.

	Mesh Reconstruction							Rendering		Time
	$\sigma = 2.5mm$			$\sigma = 5mm$			CD (mm)↓	PSNR↑	PSNR-F↑	
	Accu.(%)↑	Recall(%)↑	F1↑	Accu.(%)↑	Recall(%)↑	F1↑				
Voxurf [41]	62.89	62.54	62.42	80.93	80.61	80.38	13.3	14.34	18.34	55 mins
MonoSDF [51]	41.56	32.47	36.22	57.88	48.19	52.21	37.7	14.71	15.42	6 hrs
2DGS [15]	49.83	45.32	47.10	72.65	64.88	67.96	14.8	17.12	18.52	10 mins
GOF [53]	50.24	61.11	54.96	74.99	82.68	78.16	11.0	16.52	18.36	50 mins
3DGS [17]	\	\	\	\	\	\	\	17.19	19.12	10 mins
SparseGS [42]	\	\	\	\	\	\	\	16.93	18.74	30 mins
Ours	68.36	**69.79**	68.97	86.79	**86.82**	86.65	**9.7**	17.48	20.45	1 hr
Ours (Two cycles)	**69.61**	68.89	**69.14**	**87.79**	85.93	**86.74**	9.9	**17.58**	**20.55**	1.6 hr

Table 2. Surface reconstruction results on DTU with 5 input views. Values indicate Chamfer Distance in millimeters (mm). "-" denotes failure cases where COLMAP could not generate point clouds for 3DGS initialization. GSDF-10 is reported with 10 input images, as it fails in sparser settings. The best results are **bolded**, while the second-best are underlined.

Scan	24	37	40	55	63	65	69	83	97	105	106	110	114	118	122	Mean	Time
Voxurf [41]	2.74	4.50	3.39	1.52	2.24	2.00	2.94	**1.29**	2.49	1.28	2.45	4.69	0.93	2.74	1.29	2.43	50 mins
MonoSDF [51]	**1.30**	3.45	**1.45**	**0.61**	**1.43**	**1.17**	**1.07**	1.42	**1.49**	**0.79**	3.06	2.60	0.60	2.21	2.87	1.70	6 hrs
SparseNeuS [21]	3.57	3.73	3.11	1.50	2.36	2.89	1.91	2.10	2.89	2.01	2.08	3.44	1.21	2.19	2.11	2.43	Pretrain + 2 hrs ft
2DGS [15]	4.26	4.80	5.53	1.50	3.01	1.99	2.66	3.65	3.06	2.54	2.15	-	0.96	2.17	1.31	2.84	6 mins
GOF (TSDF) [53]	7.30	5.80	6.03	2.79	4.23	3.41	3.44	4.37	3.75	2.99	3.19	-	2.64	3.67	2.25	4.03	50 mins
GOF [53]	4.37	3.68	3.84	2.29	4.40	3.28	2.84	4.64	3.40	3.76	3.56	-	3.06	2.95	2.91	3.55	50 mins
GSDF-10 [50]	6.89	6.82	7.97	6.54	5.22	1.91	5.56	4.38	7.01	3.69	6.33	6.33	3.95	6.30	2.09	5.40	3 hrs
Ours	1.55	**2.64**	1.52	1.40	1.51	1.46	1.23	1.43	1.82	1.19	**1.49**	**1.80**	**0.54**	**1.19**	**1.04**	**1.45**	1 hr

scenes from DTU and 18 test scenes from MobileBrick for evaluation. In the MobileBrick dataset, each scene consists of 360-degree multi-view images, from which we sample 10 images with 10% overlap for sparse view reconstruction. In contrast, the DTU dataset, with higher overlap, is sampled with 5 frames per scene. We also present reconstruction results for the little-overlapping 3-view setting in the supplementary materials. For fair comparison, 3DGS-based methods are initialized using point clouds from COLMAP[30] with ground-truth poses. The selected images and poses are used for 3D reconstruction, while the remaining images serve as a test set for evaluating novel view rendering.

Baselines. We compare our proposed method with both SDF-based and 3DGS-based approaches for surface reconstruction. The SDF-based methods include MonoSDF[51], Voxurf[41], and SparseNeuS [21], which is pre-trained on large-scale data. The 3DGS-based methods include 2DGS[15] and GOF[53]. Additionally, we compare with GSDF [50], which integrates both SDF and 3DGS, similar to our approach, but is designed for dense-view settings. For novel view rendering, we evaluate all these methods along with 3DGS[17] and SparseGS[42].

Evaluation metrics. We follow the official evaluation metrics on MobileBrick, reporting Chamfer Distance, precision, recall, and F1 score at two thresholds: $2.5mm$ and $5mm$. For the DTU dataset, we use Chamfer Distance as the primary metric for surface reconstruction. To evaluate novel view rendering performance, we report PSNR for full images and PSNR-F, which is computed only over foreground regions. In each scene, we train models using sparse input images and test on all remaining views. The final result is averaged over all evaluation images.

Implementation details. We set the voxel grid resolution to 96^3 during coarse mesh training, requiring approximately 15 minutes for 10k iterations. The weight of the proposed

Table 3. Surface reconstruction results with varying numbers of input views on MobileBrick (porsche) and DTU (scan69). The Baseline represents a pure SDF-based reconstruction without the assistance from 3DGS. δ indicates the improvement.

	MobileBrick / F1 score			DTU / CD		
Input	Baseline	Ours	δ	Baseline	Ours	δ
5	33.50	**43.11**	+9.61	2.940	**1.230**	-1.710
10	59.66	**62.37**	+2.71	1.362	**1.165**	-0.197
20	63.18	**63.88**	+0.7	1.043	**0.965**	-0.078

normal loss is set to 0.05, while all other parameters follow Voxurf [41]. Next, we train 3DGS [17] for 7k iterations, which takes around 5 minutes, and render 10 new viewpoint images within 30 seconds. After expanding the training images, we increase the voxel grid resolution to 256^3 and train for 20k iterations, taking approximately 40 minutes. Thus, a complete optimization cycle takes roughly 1 hour.

4.2. Comparisons

Results on MobileBrick. Table 1 presents a quantitative comparison of our method against previous approaches. The results show that Voxurf [41], which utilizes an SDF-based representation, outperforms 2DGS [15] and GOF [53] (both 3DGS-based methods) in surface reconstruction metrics, particularly in terms of the F1 score. However, all 3DGS-based methods achieve notably better novel view rendering performance, as evidenced by their higher PSNR values compared to Voxurf. A visual comparison is illustrated in Fig. 5 and Fig. 6. BBy leveraging the strengths of both SDF and 3DGS representations, our method achieves state-of-the-art performance in surface reconstruction and novel view synthesis. To balance efficiency and performance, we adopt a single-cycle approach in practice.

Results on DTU. Table 2 presents surface reconstruction results on the DTU dataset, which is particularly challenging due to the use of only 5 uniformly sampled frames for reconstruction. SparseNeuS [21] is a pre-trained model that requires an additional 2 hours of fine-tuning. COLMAP fails to generate sparse point clouds for scene 110, preventing 3DGS initialization. GSDF [50] struggles in sparse-view settings, so we train it on 10 images. Despite these challenges, our method achieves robust reconstruction and significantly outperforms other approaches.

4.3. Ablations

Efficacy of 3DGS for Improving SDF. Table 3 compares our method with a pure SDF-based reconstruction baseline at different sparsity levels, using up to 20 images per scene. The results on MobileBrick and DTU validate the effectiveness of our 3DGS-assisted SDF approach. More results are provided in the supplementary material.

Table 4. 3DGS rendering results with different initializations, averaged across all 18 MobileBrick test scenes.

Method	Foreground PSNR
3DGS (COLMAP)	19.13
3DGS w/ mesh clean	19.88
3DGS w/ normal and mesh clean	**20.45**

Table 5. Ablation study on pose expansion strategies for in MobileBrick (aston) with 10 input images.

	F1↑	Recall(%)↑	CD (mm)↓
Baseline	55.8	49.9	8.7
Camera position perturbation	59.9	57.4	6.6
Camera poses interpolation	**60.8**	**59.1**	**6.4**

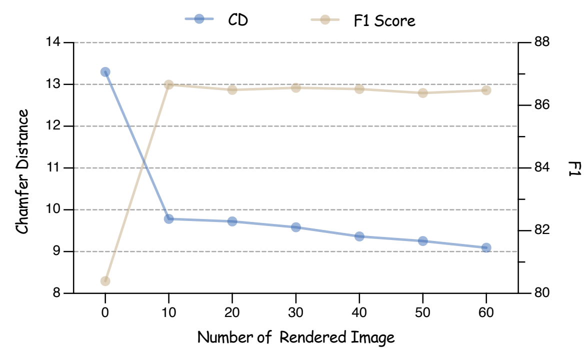


Figure 7. econstruction quality with varying numbers of 3DGS-rendered novel view images from expanded poses, averaged across all 18 MobileBrick test scenes, with an initial input of 10 images.

Efficacy of SDF for enhancing 3DGS. Table 4 compares the novel view rendering results for 3DGS using point clouds initialized with different sampling strategies. The results demonstrate that our proposed mesh cleaning and normal supervision notably improve 3DGS performance.

Number of newly rendered views. Fig. 7 illustrates the impact of the number of newly rendered images on surface reconstruction. On MobileBrick, rendering 10 novel views significantly improves Chamfer Distance (26.5%) and F1 (7.2%). As the number of novel views increases, accuracy gains gradually diminish. This suggests that while additional renderings refine reconstruction, the majority of benefits are achieved with the first 10 rendered images.

Different pose expansion strategies. Table 5 summarizes the reconstruction performance with expansion images

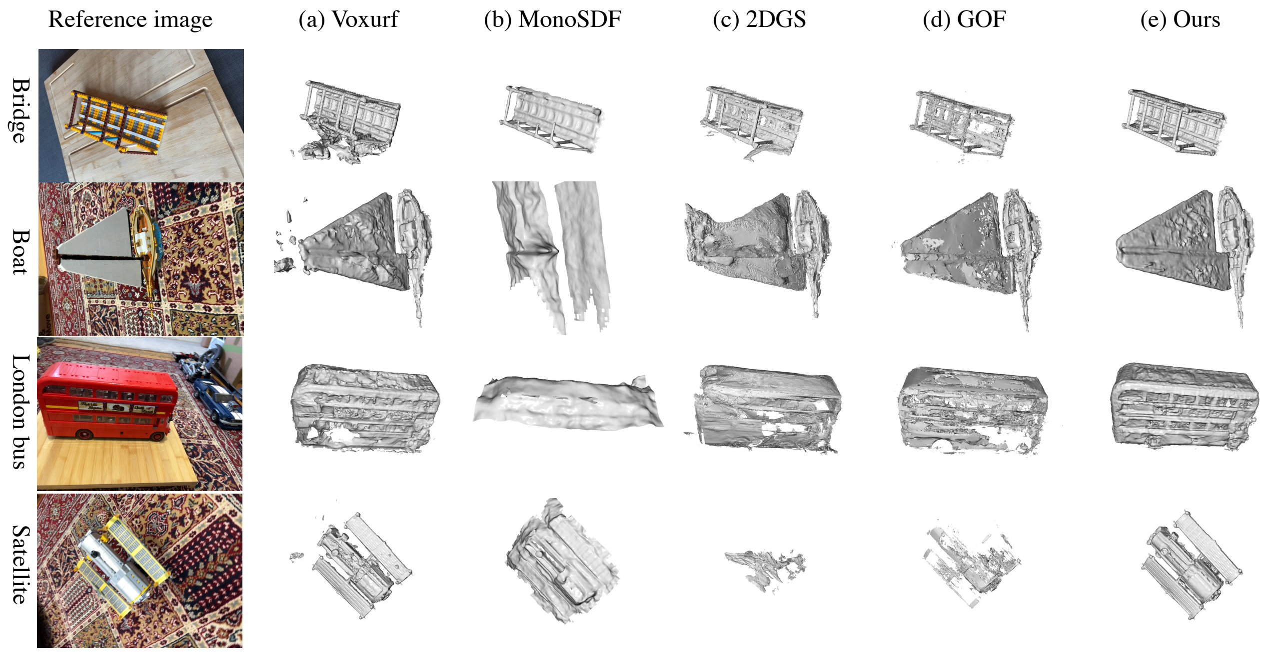


Figure 5. Qualitative mesh reconstruction comparisons on MobileBrick. See more visual results in supplementary material.

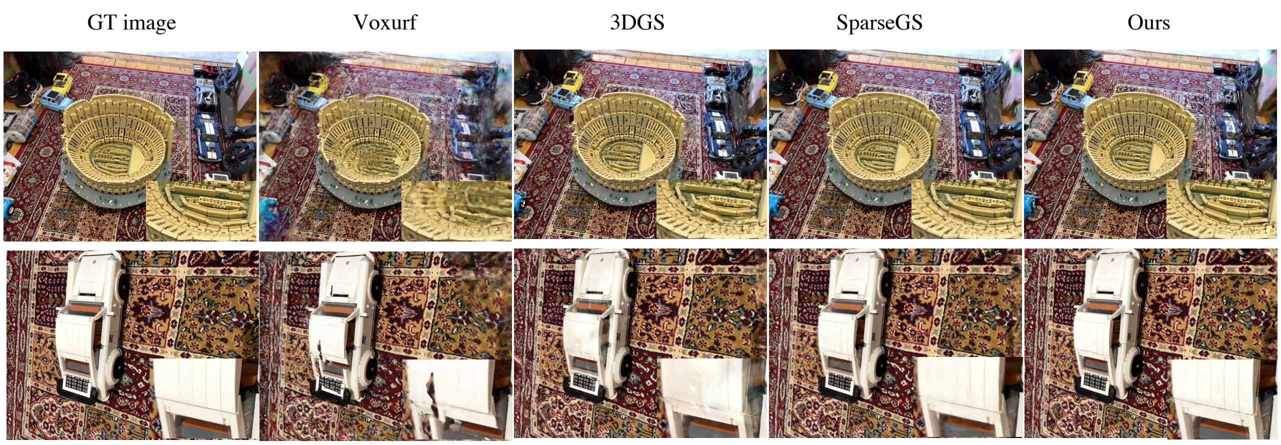


Figure 6. Qualitative novel view synthesis comparisons on MobileBrick.

from different strategies. We double the number of original input camera poses, generating new viewpoints and rendering additional images accordingly. The two strategies significantly enhance surface reconstruction quality, with camera pose interpolation yielding the greatest improvement.

5. Conclusion

This paper introduces a novel framework for sparse-view reconstruction, where SDF-based and 3DGS-based representations complement each other to enhance both surface reconstruction and novel view rendering. Specifically, our method leverages SDF for modeling global geometry and 3DGS for capturing fine details, achieving significant improvements over state-of-the-art methods on two widely used real-world datasets.

Limitation and future work. Although our method can theoretically be generalized to any SDF and novel view rendering approaches, our current implementation is built on Voxurf and 3DGS, which were selected for their efficiency-performance trade-off. As a result, our method is currently limited to object-level scenes and struggles with extremely sparse inputs, such as only two images. In the future, we aim to extend our approach to handle more diverse scenes and further improve its robustness to sparse inputs.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 62206244) .

References

[1] Jiayang Bai, Letian Huang, Jie Guo, Wen Gong, Yuanqi Li, and Yanwen Guo. 360-gs: Layout-guided panoramic gaussian splatting for indoor roaming. In *International Conference on 3D Vision 2025*. 2

[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 *Int. Conf. Comput. Vis.*, pages 5855–5864, 2021. 2

[3] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5470–5479, 2022. 2

[4] Jia-Wang Bian, Wenjing Bian, Victor Adrian Prisacariu, and Philip Torr. Porf: Pose residual field for accurate neural surface reconstruction. In *ICLR*, 2024. 2

[5] Wenjing Bian, Zirui Wang, Kejie Li, Jiawang Bian, and Victor Adrian Prisacariu. Nope-nerf: Optimising neural radiance field with no pose prior. 2023. 2

[6] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *Int. Conf. Comput. Vis.*, pages 14124–14133, 2021. 1, 2

[7] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *Eur. Conf. Comput. Vis.*, pages 333–350. Springer, 2022. 2

[8] Danpeng Chen, Hai Li, Weicai Ye, Yifan Wang, Weijian Xie, Shangjin Zhai, Nan Wang, Haomin Liu, Hujun Bao, and Guofeng Zhang. Pgsr: Planar-based gaussian splatting for efficient and high-fidelity surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 2024. 2

[9] Hanlin Chen, Chen Li, and Gim Hee Lee. Neusg: Neural implicit surface reconstruction with 3d gaussian splatting guidance. *arXiv preprint arXiv:2312.00846*, 2023. 2

[10] Yiwen Chen, Tong He, Di Huang, Weicai Ye, Sijin Chen, Jiaxiang Tang, Xin Chen, Zhongang Cai, Lei Yang, Gang Yu, et al. Meshanything: Artist-created mesh generation with autoregressive transformers. *arXiv preprint arXiv:2406.10163*, 2024. 1, 2

[11] C De Boor. A practical guide to splines. *Springer-Verlag google schola*, 2:4135–4195, 1978. 5

[12] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5501–5510, 2022. 2

[13] Antoine Guédon and Vincent Lepetit. Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5354–5363, 2024. 2

[14] Siming He, Zach Osman, and Pratik Chaudhari. From nerfs to gaussian splats, and back. *arXiv preprint arXiv:2405.09717*, 2024. 2

[15] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *SIGGRAPH 2024 Conference Papers*. Association for Computing Machinery, 2024. 1, 2, 6, 7

[16] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engil Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2014. 2, 5

[17] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 2023. 1, 2, 3, 6, 7

[18] Jiyeop Kim and Jongwoo Lim. Integrating meshes and 3d gaussians for indoor scene reconstruction with sam mask guidance. *arXiv preprint arXiv:2407.16173*, 2024. 2

[19] Kejie Li, Jia-Wang Bian, Robert Castle, Philip HS Torr, and Victor Adrian Prisacariu. Mobilebrick: Building lego for 3d reconstruction on mobile devices. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4892–4901, 2023. 2, 5

[20] Zhaoshuo Li, Thomas Müller, Alex Evans, Russell H Taylor, Mathias Unberath, Ming-Yu Liu, and Chen-Hsuan Lin. Neuralangelo: High-fidelity neural surface reconstruction. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 8456–8465, 2023. 2

[21] Xiaoxiao Long, Cheng Lin, Peng Wang, Taku Komura, and Wenping Wang. Sparseneus: Fast generalizable neural surface reconstruction from sparse views. In *Eur. Conf. Comput. Vis.*, pages 210–227. Springer, 2022. 1, 2, 6, 7

[22] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Seminal graphics: pioneering efforts that shaped the field*, pages 347–353, 1998. 2

[23] Tao Lu, Mulin Yu, Linning Xu, Yuanbo Xiangli, Limin Wang, Dahua Lin, and Bo Dai. Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 20654–20664, 2024. 2

[24] Xiaoyang Lyu, Yang-Tian Sun, Yi-Hua Huang, Xiuzhe Wu, Ziyi Yang, Yilun Chen, Jiangmiao Pang, and Xiaojuan Qi. 3dgsr: Implicit surface reconstruction with 3d gaussian splatting. *ACM Transactions on Graphics (TOG)*, 43(6):1–12, 2024. 2

[25] 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 *Eur. Conf. Comput. Vis.*, 2020. 1, 2

[26] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):1–15, 2022. 2

[27] Stanley Osher, Ronald Fedkiw, Stanley Osher, and Ronald Fedkiw. Constructing signed distance functions. *Level set methods and dynamic implicit surfaces*, pages 63–74, 2003. 2

[28] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 165–174, 2019. 2

[29] Yufan Ren, Fangjinhua Wang, Tong Zhang, Marc Pollefeys, and Sabine Süsstrunk. Volrecon: Volume rendering of signed ray distance functions for generalizable multi-view reconstruction. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 16685–16695, 2023. 1, 2

[30] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016. 6

[31] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. In *ACM Siggraph*, pages 835–846, 2006. 3

[32] Xiaowei Song, Jv Zheng, Shiran Yuan, Huan-ang Gao, Jingwei Zhao, Xiang He, Weihao Gu, and Hao Zhao. Sags: Scale-adaptive gaussian splatting for training-free antialiasing. *arXiv preprint arXiv:2403.19615*, 2024. 2

[33] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5459–5469, 2022. 2

[34] Hao Sun, Junping Qin, Lei Wang, Kai Yan, Zheng Liu, Xinglong Jia, and Xiaole Shi. 3dgs-hd: Elimination of unrealistic artifacts in 3d gaussian splatting. In *2024 6th International Conference on Data-driven Optimization of Complex Systems (DOCS)*, pages 696–702. IEEE, 2024. 2

[35] 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. 2

[36] 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. In *Adv. Neural Inform. Process. Syst.*, 2021. 1, 2, 5

[37] Ruizhe Wang, Chunliang Hua, Tomakayev Shingys, Mengyuan Niu, Qingxin Yang, Lizhong Gao, Yi Zheng, Junyan Yang, and Qiao Wang. Enhancement of 3d gaussian splatting using raw mesh for photorealistic recreation of architectures. *arXiv preprint arXiv:2407.15435*, 2024. 2

[38] LORENSEN WE. Marching cubes: A high resolution 3d surface construction algorithm. *Computer graphics*, 21(1): 7–12, 1987. 4

[39] Xinyue Wei, Kai Zhang, Sai Bi, Hao Tan, Fujun Luan, Valentin Deschaintre, Kalyan Sunkavalli, Hao Su, and Zexiang Xu. Meshlrm: Large reconstruction model for high-quality meshes. *arXiv preprint arXiv:2404.12385*, 2024. 1, 2

[40] Shuang Wu, Youtian Lin, Feihu Zhang, Yifei Zeng, Jingxi Xu, Philip Torr, Xun Cao, and Yao Yao. Direct3d: Scalable image-to-3d generation via 3d latent diffusion transformer. *Advances in Neural Information Processing Systems*, 37:121859–121881, 2024. 1, 2

[41] Tong Wu, Jiaqi Wang, Xingang Pan, Xudong Xu, Christian Theobalt, Ziwei Liu, and Dahua Lin. Voxurf: Voxel-based efficient and accurate neural surface reconstruction. In *Int. Conf. Learn. Represent.*, 2023. 1, 3, 4, 5, 6, 7

[42] Haolin Xiong, Sairisheek Muttukuru, Rishi Upadhyay, Pradyumna Chari, and Achuta Kadambi. Sparsegs: Real-time 360° sparse view synthesis using gaussian splatting. *Arxiv*, 2023. 6

[43] Jiale Xu, Weihao Cheng, Yiming Gao, Xintao Wang, Shenghua Gao, and Ying Shan. Instantmesh: Efficient 3d mesh generation from a single image with sparse-view large reconstruction models. *arXiv preprint arXiv:2404.07191*, 2024. 1, 2

[44] Runyi Yang, Zhenxin Zhu, Zhou Jiang, Baijun Ye, Xiaoxue Chen, Yifei Zhang, Yuantao Chen, Jian Zhao, and Hao Zhao. Spectrally pruned gaussian fields with neural compensation. *arXiv preprint arXiv:2405.00676*, 2024. 2

[45] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 20331–20341, 2024. 2

[46] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *Adv. Neural Inform. Process. Syst.*, 34:4805–4815, 2021. 2

[47] Vickie Ye, Ruilong Li, Justin Kerr, Matias Turkulainen, Brent Yi, Zhuoyang Pan, Otto Seiskari, Jianbo Ye, Jeffrey Hu, Matthew Tancik, et al. gsplat: An open-source library for gaussian splatting. *Journal of Machine Learning Research*, 26(34):1–17, 2025. 2

[48] Wei Yin, Chi Zhang, Hao Chen, Zhipeng Cai, Gang Yu, Kaixuan Wang, Xiaozhi Chen, and Chunhua Shen. Metric3d: Towards zero-shot metric 3d prediction from a single image. In *Int. Conf. Comput. Vis.*, pages 9043–9053, 2023. 4

[49] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4578–4587, 2021. 1, 2

[50] Mulin Yu, Tao Lu, Linning Xu, Lihan Jiang, Yuanbo Xiangli, and Bo Dai. Gsdf: 3dgs meets sdf for improved neural rendering and reconstruction. *Advances in Neural Information Processing Systems*, 37:129507–129530, 2024. 2, 6, 7

[51] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. *Adv. Neural Inform. Process. Syst.*, 2022. 6

[52] Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-splatting: Alias-free 3d gaussian splatting. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 19447–19456, 2024. 2

[53] Zehao Yu, Torsten Sattler, and Andreas Geiger. Gaussian opacity fields: Efficient adaptive surface reconstruction in unbounded scenes. *ACM Trans. Graph.*, 2024. 1, 2, 6, 7

[54] Baowen Zhang, Chuan Fang, Rakesh Shrestha, Yixun Liang, Xiaoxiao Long, and Ping Tan. Rade-gs: Rasterizing depth in gaussian splatting. *arXiv preprint arXiv:2406.01467*, 2024. 2

[55] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020. 2