Distance Field Rasterization for End-to-End Mesh Reconstruction

JINKAI CUI, University of Science and Technology of China, China
KAIWEN SONG, University of Science and Technology of China, China
CHUMENG NIU, University of Science and Technology of China, China
JUYONG ZHANG*, University of Science and Technology of China, China

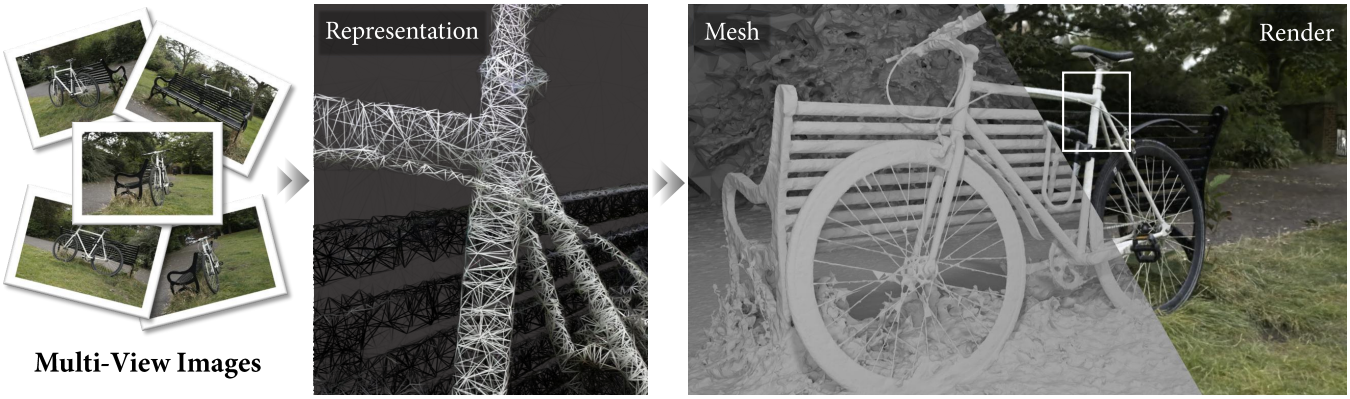

Fig. 1. Our method introduces a rasterizable SDF representation for end-to-end mesh reconstruction from multi-view images. With this representation, our method achieves high-quality mesh reconstruction (right). The middle column shows the optimized tetrahedral mesh which supports the SDF.

Rasterization based methods have recently enabled high-quality novel view synthesis at real-time rates, but their underlying volumetric primitives do not expose a direct, globally consistent surface representation, leaving surface extraction to heuristic post-processing. In contrast, implicit signed distance field (SDF) methods provide well-defined surfaces but are typically optimized with computationally expensive ray marching. We propose **SDFRaster**, a rasterizable SDF representation that bridges this gap by combining the efficiency of rasterization with signed distance field for end-to-end mesh reconstruction. Starting from a Delaunay tetrahedralization, we optimize a continuous SDF over a tetrahedral grid and render it efficiently by rasterizing tetrahedra and alpha-compositing their contributions. We further integrate differentiable Marching Tetrahedra into the optimization loop, enabling end-to-end mesh reconstruction without post-processing mesh extraction. Experiments on DTU and Tanks and Temples demonstrate that SDFRaster achieves higher-quality and more complete surface reconstructions with lower storage cost than state-of-the-art approaches. Project page: ustc3dv/SDFRaster

*Corresponding author (juyong@ustc.edu.cn).

Authors' Contact Information: Jinkai Cui, cuijk@mail.ustc.edu.cn, University of Science and Technology of China, China; Kaiwen Song, SA21001046@mail.ustc.edu.cn, University of Science and Technology of China, China; Chumeng Niu, niuchumeng@mail.ustc.edu.cn, University of Science and Technology of China, China; Juyong Zhang, juyong@ustc.edu.cn, University of Science and Technology of China, China.

ACM 1557-7368/2026/4-ART
https://doi.org/XXXXXXX.XXXXXXX

CCS Concepts: • **Computing methodologies** → **Reconstruction**; **Rasterization**; **Mesh models**.

Additional Key Words and Phrases: Differentiable Rendering, Surface Reconstruction, Multi-view-to-3D

ACM Reference Format:
Jinkai Cui, Kaiwen Song, Chumeng Niu, and Juyong Zhang. 2026. Distance Field Rasterization for End-to-End Mesh Reconstruction. *ACM Trans. Graph.* 1, 1 (April 2026), 11 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Mesh reconstruction is a fundamental problem in computer graphics and computer vision, traditionally addressed by multi-view stereo (MVS) methods [Kazhdan and Hoppe 2013; Schönberger and Frahm 2016; Schönberger et al. 2016; Yao et al. 2018; Yu and Gao 2020] that recover explicit geometry from calibrated images, yet often suffer from noise sensitivity and limited robustness in complex scenes. Recently, neural implicit representations model surfaces [Cai et al. 2022; Jiang et al. 2022; Li et al. 2023; Wang et al. 2021b,a, 2023; Yariv et al. 2021, 2020] as continuous functions and improve completeness and smoothness, but they typically require dense sampling along rays and frequent network evaluations during training, making optimization computationally expensive. Building upon these advances, rasterization based methods [Kerbl et al. 2023; Mai et al. 2025; Sun et al. 2025] have emerged as an efficient alternative, where fast GPU rasterization enables rapid rendering and optimization. Leveraging this computational advantage, many recent methods [Chen et al. 2025; Huang et al. 2024; Oh et al. 2025] reconstruct surfaces efficiently, making rasterization based representations a compelling choice when reconstruction efficiency and scalability are primary concerns.

Although rendering quality and efficiency are high, mainstream rasterization based methods typically rely on volumetric primitives, which do not expose a well-defined surface representation such as signed distance fields. As a result, mesh reconstruction is often delegated to post-processing such as depth fusion, which can introduce noise, incompleteness, and accuracy degradation. On the other hand, the lack of a well-defined surface representation leads to weak inherent multi-view geometric consistency. Consequently, existing 3DGS-based surface reconstruction methods typically promote consistency only indirectly, for example by adding multi-view regularizers [C hen et al. 2025; Chen et al. 2024b], introducing architectural inductive biases [C hen et al. 2025; Dai et al. 2024b; Huang et al. 2024], or adopting dual-branch designs that co-optimize 3DGS with a surface branch via mutual supervision [Dai et al. 2024a; Guédon et al. 2025]. However, these constraints typically act on intermediate cues, such as depths and normals rendered by 3DGS, rather than directly supervising the underlying geometry, thereby weakening the coupling between the optimization objective and the final surface quality.

These limitations call for a representation that combines the well-defined surface geometry of SDF with the efficiency of rasterization. To this end, we propose **SDFRaster**, a rasterizable SDF representation for end-to-end mesh reconstruction. Building on the representation of Radiance Meshes [Mai et al. 2025], we discretize the 3D scene using a Delaunay tetrahedralization and parameterize the SDF at the tetrahedral vertices. Within each tetrahedron, the SDF is obtained by linear interpolation of the vertex values, resulting in a continuous, piecewise-linear signed distance field over the scene. Thanks to this multi-view-consistent geometry, whose zero level set directly defines the surface, SDFRaster avoids the post-processing used in volumetric primitive methods and enables direct mesh extraction from the optimized geometric field. To render and optimize the signed distance field efficiently, we map the SDF to volumetric opacity and employ a rasterization-based differentiable renderer that rasterizes tetrahedra, computes intersections between rays and tetrahedra, and alpha-composites their contributions. This removes the need for per-ray dense sampling in ray marching renderers and enables fast, scalable optimization on modern GPU architectures.

Moreover, to better align field optimization with mesh quality, we extract meshes during optimization and couple two complementary geometry representations. Specifically, we integrate differentiable Marching Tetrahedra [Treece et al. 1999] into the optimization process, enabling surface-based regularization and geometry consistency losses directly on the extracted mesh. In addition, we render depth and normals from the SDF field and enforce their consistency with those rendered from the extracted mesh, tightening the link between appearance fitting and surface fidelity. Finally, we propose a surface-centric adaptive resolution strategy that concentrates representation capacity near the surface, improving local geometric detail while keeping the extracted meshes compact. Extensive experiments show that SDFRaster is over $6\times$ faster than implicit SDF baselines, achieves the lowest Chamfer distance among explicit baselines on DTU, and maintains competitive F1 on TnT while producing compact meshes (about $3\times$ smaller in storage than meshes extracted via TSDF fusion).

In summary, the main contributions of this paper include:

- We propose a Delaunay tetrahedral SDF representation that supports efficient rasterization based differentiable rendering and provides well-defined geometry.
- We integrate differentiable Marching Tetrahedra into optimization to extract meshes on the fly, enabling end-to-end mesh reconstruction and avoiding reliance on post-processing.
- We adapt representation capacity near the evolving zero-level set to recover fine details while keeping the extracted meshes compact and scalable to large scenes.

2 Related Work

2.1 Novel View Synthesis

Neural radiance fields [Mildenhall et al. 2020] model scenes as continuous volumetric functions optimized with differentiable volume rendering. Subsequent work improves quality and efficiency through compact parameterizations such as feature grids and tensor factorization, and through antialiasing or unbounded scene modeling [Barron et al. 2022, 2023; Chen et al. 2022b; Fridovich-Keil et al. 2022; Gao et al. 2022; Garbin et al. 2021; Müller et al. 2022; Reiser et al. 2024; Song et al. 2024; Sun et al. 2022]. These methods achieve strong novel view synthesis, but often require extensive training time.

Explicit rendering pipelines have shifted NVS toward real-time synthesis by rasterizing primitives. 3DGS [Kerbl et al. 2023] represents scenes with explicit Gaussian primitives and achieves high-quality views at interactive rates, and variants [Huang et al. 2024; Lu et al. 2024; Radl et al. 2024] improve efficiency and stability while preserving the splatting pipeline. Recently, explicit representations not based on Gaussian splats have also appeared, such as sparse voxel rasterization [Sun et al. 2025] and cell-complex renderers [Govindarajan et al. 2025; Kulhanek and Sattler 2023; Mai et al. 2025; Song et al. 2025] that organize space into voxels or polyhedral cells for rasterization or ray tracing. These methods offer efficient NVS, but their primary goal remains view synthesis rather than accurate surface reconstruction.

2.2 Neural Surface Reconstruction

Classical multi-view reconstruction pipelines estimate depth or point clouds and then fuse them into a surface using volumetric fusion [Schönberger et al. 2016; Yao et al. 2018; Yu and Gao 2020] or implicit fitting [Kazhdan et al. 2006; Kazhdan and Hoppe 2013]. While effective in controlled settings, these pipelines are sensitive to noisy depth and view-dependent artifacts, and the final mesh quality is tied to the fusion resolution and coverage.

Neural surface reconstruction instead learns implicit fields and extracts their zero level set, while some methods based on discrete structures explore more direct mesh extraction and optimization [B inninger et al. 2025; Chen et al. 2022a; Shen et al. 2021, 2023]. Early neural implicit methods such as IDR [Yariv et al. 2020] build on differentiable rendering to model surfaces implicitly. VolSDF [Yariv et al. 2021] and NeuS [Wang et al. 2021b] convert SDF to opacity and render via per-ray volumetric integration with dense point sampling. This dense sampling makes training costly and leaves efficient large-scale reconstruction challenging [Huang et al. 2024; Li

et al. 2023]. Other variants improve robustness or reconstruction quality with stronger regularization, geometric priors, or training strategies [Fu et al. 2022; Wang et al. 2023; Yu et al. 2022a,b]. Neuralangelo leverages multi-resolution hash encodings with coarse-to-fine training and achieves high-quality surfaces [Li et al. 2023]. Recent work reduces the cost of volume rendering by adopting hybrid structures such as regular grids and factorized grids [Chen et al. 2022b; Fridovich-Keil et al. 2022; Sun et al. 2022; Wu et al. 2023]. These grid-based methods improve efficiency while preserving the implicit formulation, but they still rely on ray marching and remain expensive.

2.3 Primitive-based Surface Reconstruction

Gaussian splatting enables fast rendering and has motivated surface reconstruction on top of 3DGS [Kerbl et al. 2023]. One line of work [C hen et al. 2025; Guédon and Lepetit 2024; Huang et al. 2024; Turkulainen et al. 2025] aligns splats with surfaces using planar or normal-aware regularization, then extracts meshes from rendered depth cues via TSDF fusion [Curless and Levoy 1996; Newcombe et al. 2011]. Another line strengthens geometric regularization within the splatting pipeline with depth/normal consistency, planar constraints, or higher-order primitives [Chen et al. 2024a; Li et al. 2024, 2025; Zhang et al. 2026, 2025]. In both cases, geometry is recovered after training and still relies on depth-fusion post-processing or overlapping-primitive cues, so surface quality is limited by multi-view depth consistency and depth accuracy, often yielding incomplete or overly dense geometry.

Hybrid methods jointly optimize Gaussian primitives with implicit distance fields. GSDF [Dai et al. 2024a], GSurf [Xu et al. 2024], GS-Pull [Han et al. 2024], and 3DGSR [Lyu et al. 2024] introduce SDF supervision or gradient pulling toward the zero level set, and then extract meshes from the learned field, which improves geometric consistency but introduces a dual representation and additional optimization complexity.

More recent approaches reduce the gap between rendering and geometry with tighter extraction pipelines or mesh-based reconstruction [Held et al. 2025]. GOF constructs Gaussian opacity fields and extracts meshes on a tetrahedral grid induced by Gaussians, enabling adaptive and compact meshes without TSDF fusion [Yu et al. 2024]. MILo extracts mesh during training to reduce discrepancies between the rendered field and the extracted surface [Guédon et al. 2025]. These approaches couple surface extraction more directly to the optimized representation.

3 Method

We propose **SDFRaster**, a rasterizable SDF framework that reconstructs meshes from calibrated multi-view images (see Fig. 1). Starting from a scene bounding volume, a Delaunay tetrahedralization is built as the geometric grid. A continuous SDF is learned over the tetrahedral complex and rendered by rasterizing tetrahedra with opacity-based alpha compositing. Meshes are extracted during training via differentiable Marching Tetrahedra, enabling geometry consistency losses without fixed-resolution fusion. See Fig. 2 for a visual overview of our method.

3.1 Representation

We maintain a set of vertices $\mathcal{V} = \{\mathbf{v}_i \in \mathbb{R}^3\}$ and compute a Delaunay tetrahedralization $\mathcal{T}$ over $\mathcal{V}$, forming a tetrahedral cell complex that partitions the scene into non-overlapping tetrahedra. This grid could provide an injective parameterization of the 3D scene, which is necessary to represent a signed distance function $f : \mathbb{R}^3 \to \mathbb{R}$. To avoid discontinuities from directly optimizing discrete values, we parameterize the SDF with a compact network using multi-resolution hash encoding E_γ [Li et al. 2023; Müller et al. 2022] queried on the tetrahedral grid. The SDF and appearance branches share the same $E_\gamma(\cdot)$ but use separate heads.

Geometry. We encode geometry as a signed distance field over the tetrahedral complex. Concretely, we predict a SDF value at each grid vertex and extend it to the interior of every tetrahedron via barycentric interpolation. The SDF value at vertex $\mathbf{v}_i$ is obtained by a dedicated SDF head H_{sdf} that takes the hash-encoded feature and the raw vertex coordinates as input:

$$f_i \;=\; H_{\text{sdf}}\big(\big[E_\gamma(\mathbf{v}_i),\, \mathbf{v}_i\big]\big)\,, \tag{1}$$

where $[\cdot,\cdot]$ denotes concatenation. For any point $\mathbf{x}$ inside tetrahedron $t = (i_0, i_1, i_2, i_3)$ with barycentric coordinates $\lambda_k(\mathbf{x})$, where $k = 0, 1, 2, 3$, satisfying $\sum_{k=0}^{3} \lambda_k(\mathbf{x}) = 1$ and $\lambda_k(\mathbf{x}) \geq 0$, we define a continuous piecewise-linear SDF:

$$f(\mathbf{x}) \;=\; \sum_{k=0}^{3} \lambda_k(\mathbf{x})\, f_{i_k}. \tag{2}$$

This yields a globally defined, piecewise-linear SDF over the tetrahedral complex, which we use for differentiable rendering (Sec. 3.2) and surface extraction. Notably, because the network is evaluated only at grid vertices and interior values are obtained via interpolation, our formulation substantially reduces the number of SDF network queries relative to methods that evaluate an MLP at every sample point, such as NeuS [Wang et al. 2021b]. Moreover, once the vertex SDF values are queried, the geometry can be stored and processed as a purely explicit representation.

Appearance. For each tetrahedron t_k, we choose the centroid as the representative query point $\mathbf{O}_k$. Attaching appearance directly to vertices is unstable under Delaunay topology changes; querying a representative point per tetrahedron yields stable per-cell attributes [Mai et al. 2025]. Using a shared multi-resolution hash encoder $E_\gamma(\cdot)$ [Müller et al. 2022] and an appearance head, we predict a view-dependent base color $\mathbf{c}_k^0(\mathbf{d})$ and a linear color gradient $\nabla\mathbf{c}_k$ at $\mathbf{O}_k$, and define the color at any interior point $\mathbf{p} \in t_k$ as a first-order model around $\mathbf{O}_k$:

$$\mathbf{c}_k(\mathbf{p}, \mathbf{d}) \;=\; \mathbf{c}_k^0(\mathbf{d}) \;+\; \nabla\mathbf{c}_k \cdot (\mathbf{p} - \mathbf{O}_k)\,. \tag{3}$$

Here $\mathbf{c}_k^0(\mathbf{d})$ is view-dependent and $\nabla\mathbf{c}_k$ is a per-tetrahedron linear color gradient predicted by the appearance head. Given the SDF representation anchored on the tetrahedral complex, we next describe a differentiable rendering procedure that rasterizes the piecewise-linear signed distance field.

3.2 Rendering

Following Radiance Meshes [Mai et al. 2025], we adopt a rasterizable volume rendering framework over the tetrahedral grid. The

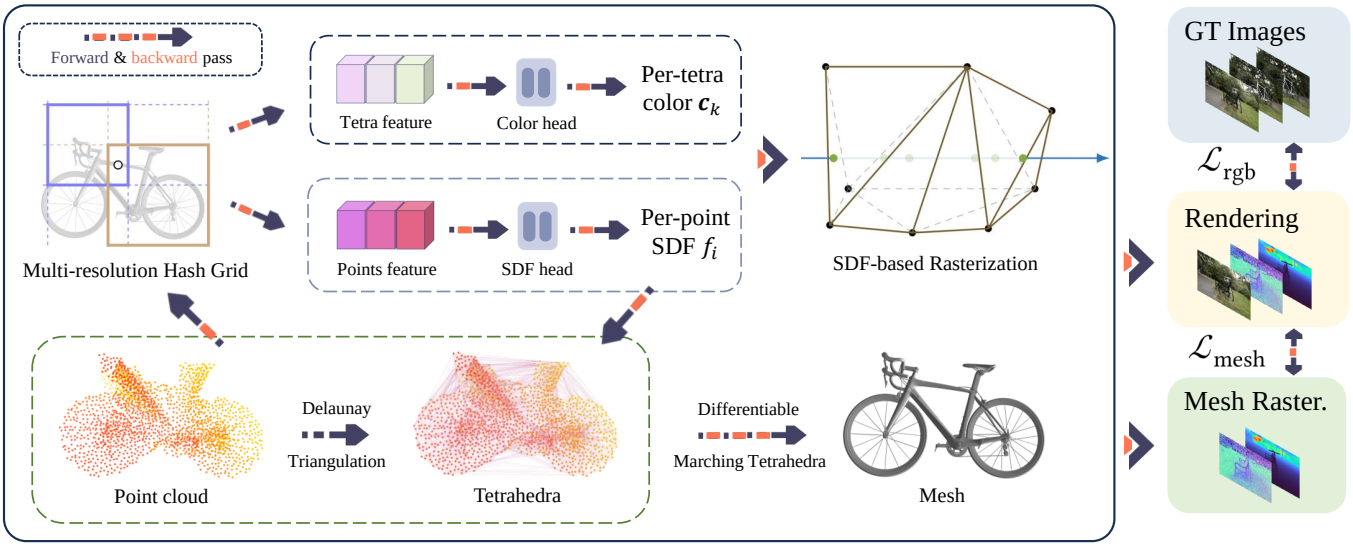

Fig. 2. **Overview of SDFRaster**. We learn a continuous SDF on a Delaunay tetrahedral grid, using a shared multi-resolution hash encoder to predict SDF values at vertices and appearance per tetrahedron. We render the images by rasterizing tetrahedra and alpha-compositing SDF-derived opacities. We apply differentiable Marching Tetrahedra on the tetrahedral grid with the learned SDF values to extract meshes in the optimization, enabling end-to-end mesh reconstruction.

signed distance field is converted to opacity using the SDF-to-opacity mapping function in NeuS [Wang et al. 2021b]. For each pixel, we consider the corresponding camera ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ and compute the exact entry and exit distances for every intersected tetrahedron t_k, denoted as $t_k^{\text{in}} < t_k^{\text{out}}$, with $\mathbf{p}_k^{\text{in}} = \mathbf{r}(t_k^{\text{in}})$ and $\mathbf{p}_k^{\text{out}} = \mathbf{r}(t_k^{\text{out}})$. Along each ray, the contributions from these tetrahedra are accumulated in a front-to-back order using alpha compositing.

SDF-based opacity. To render SDF within this framework, we map SDF values to an interval opacity as in NeuS [Wang et al. 2021b]. For each segment, we evaluate the piecewise-linear SDF at its endpoints

$$f_{\text{prev}} = f(\mathbf{p}_k^{\text{in}}), \qquad f_{\text{next}} = f(\mathbf{p}_k^{\text{out}}). \tag{4}$$

We map SDF values to an interval opacity using the logistic CDF $\Phi_s(x) = (1 + e^{-sx})^{-1}$, as in NeuS [Wang et al. 2021b], where s is the learned inverse standard deviation controlling surface sharpness.

$$\alpha_k = \max\left(\frac{\Phi_s(f_{\text{prev}}) - \Phi_s(f_{\text{next}})}{\Phi_s(f_{\text{prev}})}, 0\right). \tag{5}$$

Tetrahedron color and compositing. We then compute the tetrahedron color using the linear field from Sec. 3.1, $\mathbf{c}_k(\mathbf{p}, \mathbf{d}) = \mathbf{c}_k^0(\mathbf{d}) + \nabla \mathbf{c}_k \cdot (\mathbf{p} - \mathbf{O}_k)$. We evaluate colors at the segment endpoints and use the segment-average color

$$\bar{\mathbf{c}}_k = \tfrac{1}{2}\Big(\mathbf{c}_k(\mathbf{p}_k^{\text{in}}, \mathbf{d}) + \mathbf{c}_k(\mathbf{p}_k^{\text{out}}, \mathbf{d})\Big). \tag{6}$$

The final pixel color is accumulated by standard alpha compositing in depth order

$$\mathbf{C} = \sum_k T_k\, \alpha_k\, \bar{\mathbf{c}}_k, \qquad T_k = \prod_{l<k} (1 - \alpha_l). \tag{7}$$

Tetrahedron depth and normal. Rendering other properties mirrors color compositing by replacing the color term with the target modality; we reuse the same weights $T_k \alpha_k$ for depth and normals. For depth, we assign each intersected tetrahedron a representative depth as the segment midpoint

$$z_k = \tfrac{1}{2}(t_k^{\text{in}} + t_k^{\text{out}}), \tag{8}$$

and alpha-composite it along the ray. For normals, we assume a constant normal within each tetrahedron. Since the SDF is linearly interpolated inside t_k, its gradient is constant; for $f(\mathbf{x}) = \sum_{j=1}^{4} \lambda_j(\mathbf{x})\, f_{k,j}$,

$$\nabla f_k = \sum_{j=1}^{4} f_{k,j}\, \nabla \lambda_j(\mathbf{x}), \qquad \mathbf{n}_k = \nabla f_k / \|\nabla f_k\|_2. \tag{9}$$

The normal is obtained by alpha compositing and normalization.

3.3 Surface-Centric Adaptive Strategy

We use a surface-centric refinement strategy, together with culling and pruning for efficiency and compactness.

Densification. To concentrate resolution near the target geometry, surface-crossing tetrahedra with mixed-sign vertex SDF values are ranked by circumradius, and the top-k (5% of all tetrahedra) are split by inserting a vertex at the tetrahedron centroid before re-tetrahedralizing. This allocates capacity to the evolving zero level set where detail is needed.

We also adopt Radiance Meshes' [Mai et al. 2025] error-driven densification based on SSIM and total-variance scores aggregated per tetrahedron from multi-view residuals. Tetrahedra with the highest scores are split, and new vertices are inserted using the two most

erroneous views by intersecting their mean rays, with a barycentric fallback when the intersection is degenerate.

Culling. Culling is a non-destructive operation: tetrahedra are retained in the grid but omitted during rendering. A conservative opacity upper bound is obtained by pairing the largest and smallest vertex SDF values along a segment:

$$O_{\max} \;=\; 1 - \frac{\Phi_s(f_{\min})}{\Phi_s(f_{\max})}. \tag{10}$$

Here $f_{\min} = \min_{i\in\{1,2,3,4\}} f_i$ and $f_{\max} = \max_{i\in\{1,2,3,4\}} f_i$, where f_i are the SDF values at the tetrahedron's four vertices. A tetrahedron is skipped when $\min_{i\in t} |f_i| > \ell(s)$ and $O_{\max} < 0.1$. The band width that contains 99% of the logistic CDF mass is $\ell(s) \approx 2\ln 199/s$ [Oh et al. 2025], so it contracts around the zero level set as the learnable parameter s increases during optimization.

Pruning. Pruning is destructive and keeps the representation compact. For each tetrahedron t, we record its peak contribution as the maximum alpha-compositing weight over pixels and views,

$$c_t = \max_{i,\pi} w^{\pi}_{t,i}, \tag{11}$$

and aggregate it to vertices by $c_v = \max_{t\ni v} c_t$, where $w^{\pi}_{t,i} = T^{\pi}_{t,i}\alpha^{\pi}_{t,i}$ is the per-pixel alpha-compositing weight in Sec. 3.2. A vertex v is pruned when it is both low-contribution and far from the surface band,

$$c_v < \tau_c \quad \text{and} \quad |f(v)| > \ell(s), \tag{12}$$

where $f(v)$ is the SDF at vertex v, τ_c is a contribution threshold. This is applied only to internal vertices. We then remove associated vertices/tetrahedra and re-tetrahedralize.

3.4 Optimization

We optimize the parameters of the shared hash-encoded SDF and appearance predictors together with the tetrahedral vertices, under a combined photometric and geometric objective that keeps the extracted mesh consistent with the SDF. Each iteration renders the tetrahedral field, extracts a mesh with differentiable Marching Tetrahedra, and rasterizes the mesh to obtain depth/normal maps for geometric supervision. The total loss is

$$\mathcal{L} \;=\; \mathcal{L}_{\text{rgb}} \;+\; \lambda_{\text{mesh}}\mathcal{L}_{\text{mesh}} \;+\; \lambda_{\text{field}}\mathcal{L}_{\text{field}}. \tag{13}$$

For appearance, we use an ℓ_1 reconstruction term with SSIM on rendered colors $\mathbf{C}$ and ground truth $\mathbf{C}^*$,

$$\mathcal{L}_{\text{rgb}} = \|\mathbf{C} - \mathbf{C}^*\|_1 \;+\; \lambda_{\text{ssim}}(1 - \text{SSIM}). \tag{14}$$

To link surface losses to the SDF, we extract a triangle mesh $\mathcal{M}$ as the zero level set using differentiable Marching Tetrahedra [Treece et al. 1999]. For each tetrahedron whose vertex SDFs have opposite signs, the surface intersects its edges; the resulting 3 (or 4) intersection points form 1 (or 2) triangles. We denote each intersection point as $\mathbf{v}_n$. For an edge between vertices $\mathbf{v}_i$ and $\mathbf{v}_j$ with opposite-sign SDF values f_i and f_j, the intersection point is

$$\mathbf{v}_n = \frac{f_i\,\mathbf{v}_j - f_j\,\mathbf{v}_i}{f_i - f_j}, \tag{15}$$

which is differentiable with respect to both SDF values and vertex positions, allowing gradients to flow from mesh losses to the SDF. We then enforce field and surface consistency by rendering depth and normals from both (i) the tetrahedral SDF and (ii) the mesh $\mathcal{M}$. Let $D(i)$ and $\tilde{\mathbf{N}}(i)$ denote the depth and normal rendered from the field, and $D_{\text{M}}(i)$ and $\mathbf{N}_{\text{M}}(i)$ those rendered from the mesh. The weights λ_{mesh}, λ_{field}, λ_{MD}, and λ_{MN} balance photometric, field, and mesh terms. The mesh term is

$$\mathcal{L}_{\text{mesh}} = \lambda_{\text{MD}}\mathcal{L}_{\text{MD}} \;+\; \lambda_{\text{MN}}\mathcal{L}_{\text{MN}}, \tag{16}$$

with

$$\mathcal{L}_{\text{MD}} = \sum_i \log(1 + |D(i) - D_{\text{M}}(i)|) \tag{17}$$

$$\mathcal{L}_{\text{MN}} = \sum_i \left(1 - \tilde{\mathbf{N}}(i)\cdot\mathbf{N}_{\text{M}}(i)\right) \tag{18}$$

We add a normal and depth consistency loss on the field rendering, where $\mathbf{N}_D(i)$ is estimated from the rendered depth map D via finite differences, following [Huang et al. 2024]:

$$\mathcal{L}_{\text{ND}} = \sum_i \left(1 - \tilde{\mathbf{N}}(i)\cdot\mathbf{N}_D(i)\right). \tag{19}$$

Finally, we regularize the SDF with an Eikonal loss over all tetrahedral vertices. We use numerical gradients because hash-encoded fields are only piecewise smooth and higher-order autodiff can be noisy or unstable; finite differences provide a robust estimate for Eikonal regularization [Gropp et al. 2020; Li et al. 2023]. At each vertex $\mathbf{x}$, we query the SDF and estimate its gradient via finite differences, e.g.,

$$\nabla_x f(\mathbf{x}) = \frac{f(\mathbf{x} + \epsilon_x) - f(\mathbf{x} - \epsilon_x)}{2\epsilon}, \tag{20}$$

where $\epsilon_x = [\epsilon, 0, 0]$, and $\nabla_y f(\mathbf{x})$ and $\nabla_z f(\mathbf{x})$ are computed similarly. We set ϵ to the cell size of the finest currently active hash-grid resolution.

$$\mathcal{L}_{\text{eik}} = \frac{1}{N}\sum_{i=1}^{N}(\|\nabla f(\mathbf{x}_i)\|_2 - 1)^2, \tag{21}$$

$$\mathcal{L}_{\text{curv}} = \frac{1}{N}\sum_{i=1}^{N}\left|\nabla^2 f(\mathbf{x}_i)\right|. \tag{22}$$

Here N is the number of tetrahedral vertices. The field term is

$$\mathcal{L}_{\text{field}} = \lambda_{\text{ND}}\mathcal{L}_{\text{ND}} \;+\; \lambda_{\text{eik}}\mathcal{L}_{\text{eik}} \;+\; \lambda_{\text{curv}}\mathcal{L}_{\text{curv}}. \tag{23}$$

4 Experiments

4.1 Implementation Details

Our implementation is built in PyTorch with SlangD shaders on top of the Radiance Meshes [Mai et al. 2025] codebase. We follow the training schedule in Sec. 3, use the adaptive densification and pruning strategy in Sec. 3.3, and extract meshes with differentiable Marching Tetrahedra [Treece et al. 1999] during training. Unless otherwise specified, we set the loss weights to $\lambda_{\text{eik}} = 0.01$, $\lambda_{\text{curv}} = 5\times10^{-6}$, $\lambda_{\text{ND}} = \lambda_{\text{MD}} = \lambda_{\text{MN}} = 0.05$, and $\lambda_{\text{ssim}} = 0.2$. We train on each dataset for 18,000 iterations. DTU [Aanæs et al. 2016] experiments are run on a single NVIDIA RTX 4090, while TnT [Knapitsch et al. 2017] and Mip-NeRF 360 [Barron et al. 2022] are run on a single NVIDIA L40. Average training time is about 98 minutes per DTU scene and 6 hours per TnT scene. We densify every 500 iterations from 2,000 to 16,000 and prune every 500 iterations from 4,000 to 15,000 using the criteria in Sec. 3.3. For better surface reconstruction on TnT, we increase the mesh loss weight by $5\times$. On complex

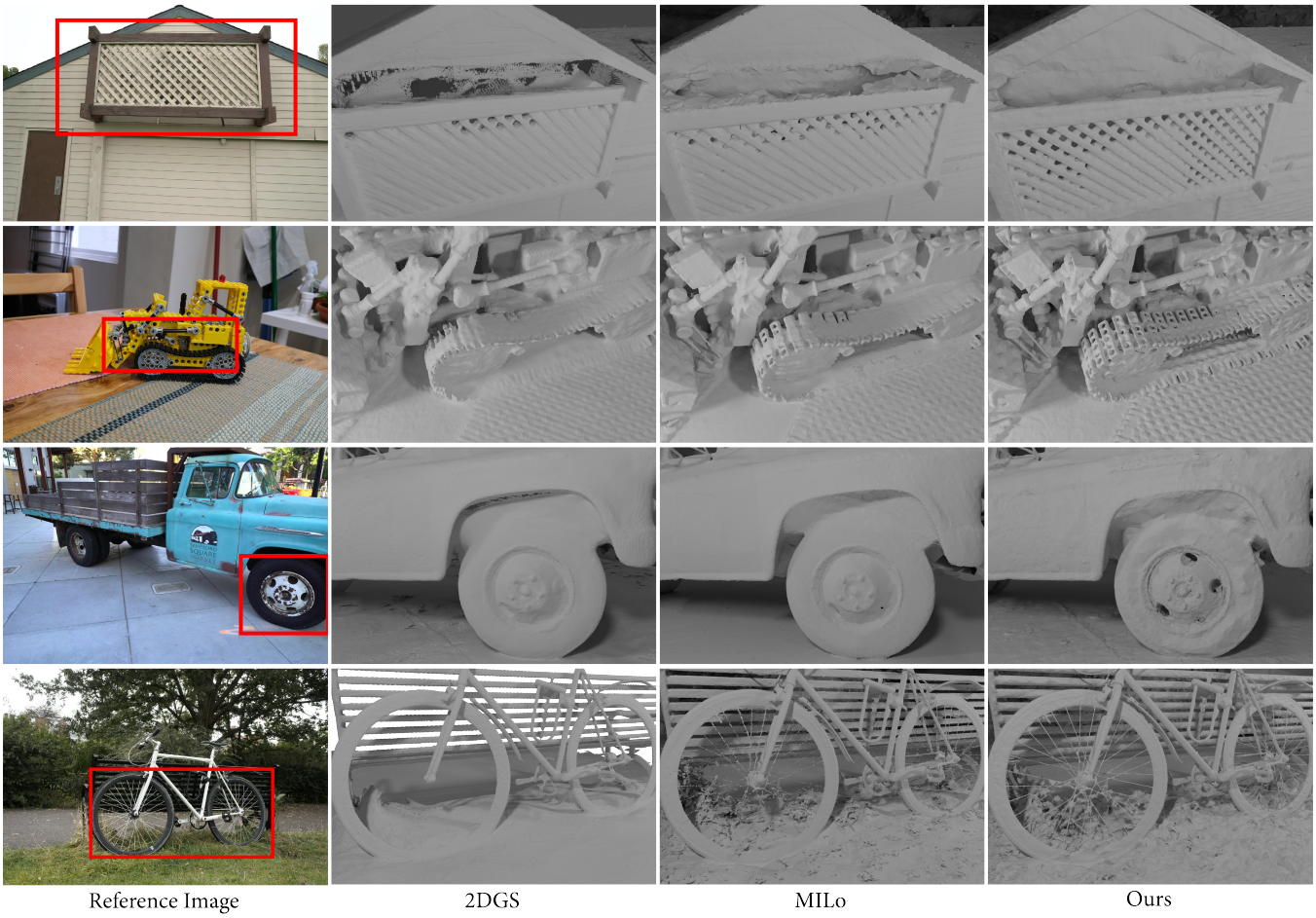

Fig. 3. **Surface Reconstruction on the Tanks and Temples Dataset [Knapitsch et al. 2017]** and Mip-NeRF 360 Dataset [Barron et al. 2022]. Qualitative comparison on four scenes (Barn, Truck, Bicycle, Kitchen) with MILo [Guédon et al. 2025], 2DGS [Huang et al. 2024], and our method. Our method converges to more accurate geometry while keeping the extracted meshes compact.

Fig. 4. Comparison between Marching Tetrahedra (MT) [Treece et al. 1999] on the adaptive tetrahedral grid and Marching Cubes (MC) [Lorensen and Cline 1987] on a fixed-resolution grid using the same SDF optimized by SDFRaster. Despite the higher uniform resolution, Marching Cubes produces noisier surfaces, holes, and floaters. Our MT extraction yields lower Chamfer distance and a more compact mesh.

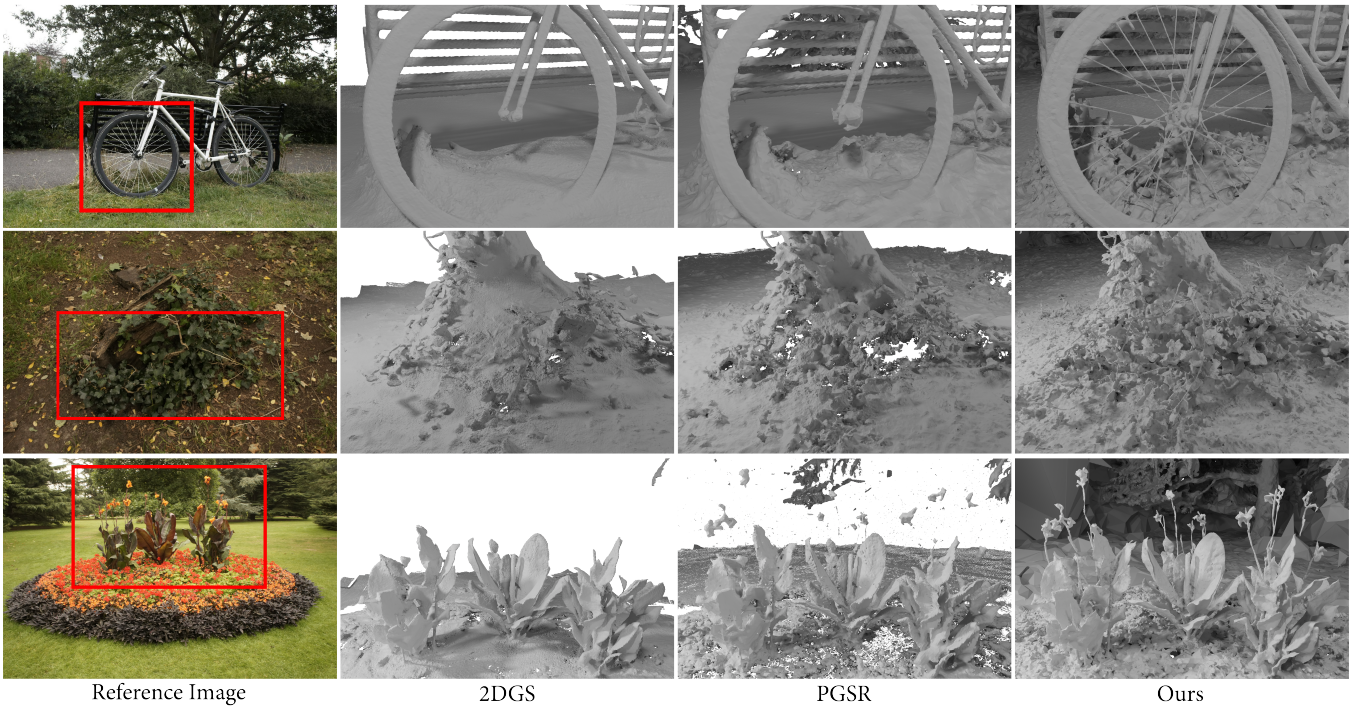

Fig. 5. Comparison between 2DGS [Huang et al. 2024] and PGSR [C hen et al. 2025], which use depth fusion for mesh extraction, and our SDF-based mesh extraction. 2DGS and PGSR render multi-view depth maps and fuse them with TSDF [Curless and Levoy 1996], a pipeline that is sensitive to view-dependent depth noise and often struggles with thin structures. In contrast, we extract the zero level set from the learned SDF on an adaptive tetrahedral grid, yielding more complete surfaces and better preserving fine details.

Table 1. **Surface reconstruction metrics on the DTU dataset**. We report the Chamfer Distance across 15 scenes (lower is better). Implicit methods are listed above and explicit methods below. Best results for implicit methods are highlighted in blue; best results for explicit methods are highlighted in red.

		24	37	40	55	63	65	69	83	97	105	106	110	114	118	122	Mean	Time
implicit	VolSDF [Yariv et al. 2021]	1.14	1.26	0.81	0.49	1.25	0.70	0.72	1.29	1.18	0.70	0.66	1.08	0.42	0.61	0.55	0.86	> 12h
	NeuS [Wang et al. 2021b]	1.00	1.37	0.93	0.43	1.10	0.65	0.57	1.48	1.09	0.83	0.52	1.20	0.35	0.49	0.54	0.84	> 12h
	Neuralangelo [Li et al. 2023]	0.37	0.72	0.35	0.35	0.87	0.54	0.53	1.29	0.97	0.73	0.47	0.74	0.32	0.41	0.43	0.61	> 12h
explicit	SuGaR [Guédon and Lepetit 2024]	1.47	1.33	1.13	0.61	2.25	1.71	1.15	1.63	1.62	1.07	0.79	2.45	0.98	0.88	0.79	1.32	52m
	SVRaster [Sun et al. 2025]	0.88	2.12	0.78	0.53	0.87	1.01	2.10	1.08	1.20	1.45	1.08	0.85	0.43	0.60	0.93	1.06	5m
	2DGS [Huang et al. 2024]	0.71	0.91	0.77	0.43	1.49	1.08	0.93	1.45	1.42	0.95	0.84	1.59	0.79	0.83	0.57	0.98	9m
	GOF [Yu et al. 2024]	0.69	0.89	0.48	0.52	1.67	1.03	0.82	1.29	1.42	0.82	0.61	1.28	0.55	0.62	0.48	0.88	55m
	MILo [Guédon et al. 2025]	0.43	0.74	0.32	0.36	0.80	0.75	0.69	1.34	1.29	0.72	0.67	0.93	0.34	0.74	0.47	0.71	61m
	Radiance Meshes [Mai et al. 2025]	3.17	5.02	1.31	0.97	4.93	4.71	2.32	4.73	2.68	2.80	2.48	2.38	1.85	2.60	1.67	2.91	79m
	Ours	0.43	0.67	0.44	0.33	0.93	0.68	0.89	1.17	1.30	0.64	0.53	1.05	0.37	0.43	0.40	0.68	98m

unbounded scenes such as TnT, we disable the Eikonal regularizer to improve convergence; an ablation is provided in Sec. 4.3.

4.2 Comparison

We evaluate on DTU [Aanæs et al. 2016] and Tanks and Temples (TnT) [Knapitsch et al. 2017] for surface reconstruction and include qualitative mesh comparisons on Mip-NeRF 360 [Barron et al. 2022]; we use the standard 15 DTU scenes and the common six-scene TnT subset (Barn, Caterpillar, Ignatius, Courthouse, Meetingroom, Truck). Mip-NeRF 360 provides unbounded scenes without ground-truth geometry, so we present qualitative mesh comparisons there.

Following standard practice, we report Chamfer distance (CD) on DTU and F1-score on TnT; for TnT, F1-score combines precision and recall at the standard distance threshold, and for DTU we follow the official evaluation script and report the average of accuracy and completeness as Chamfer distance. We cull extracted meshes with camera masks on DTU and visibility filtering on TnT before evaluation. We compare our method with explicit and implicit approaches for surface reconstruction from multi-view images. We follow each paper's mesh extraction procedure and use official implementations and recommended hyperparameters when reproducing results.

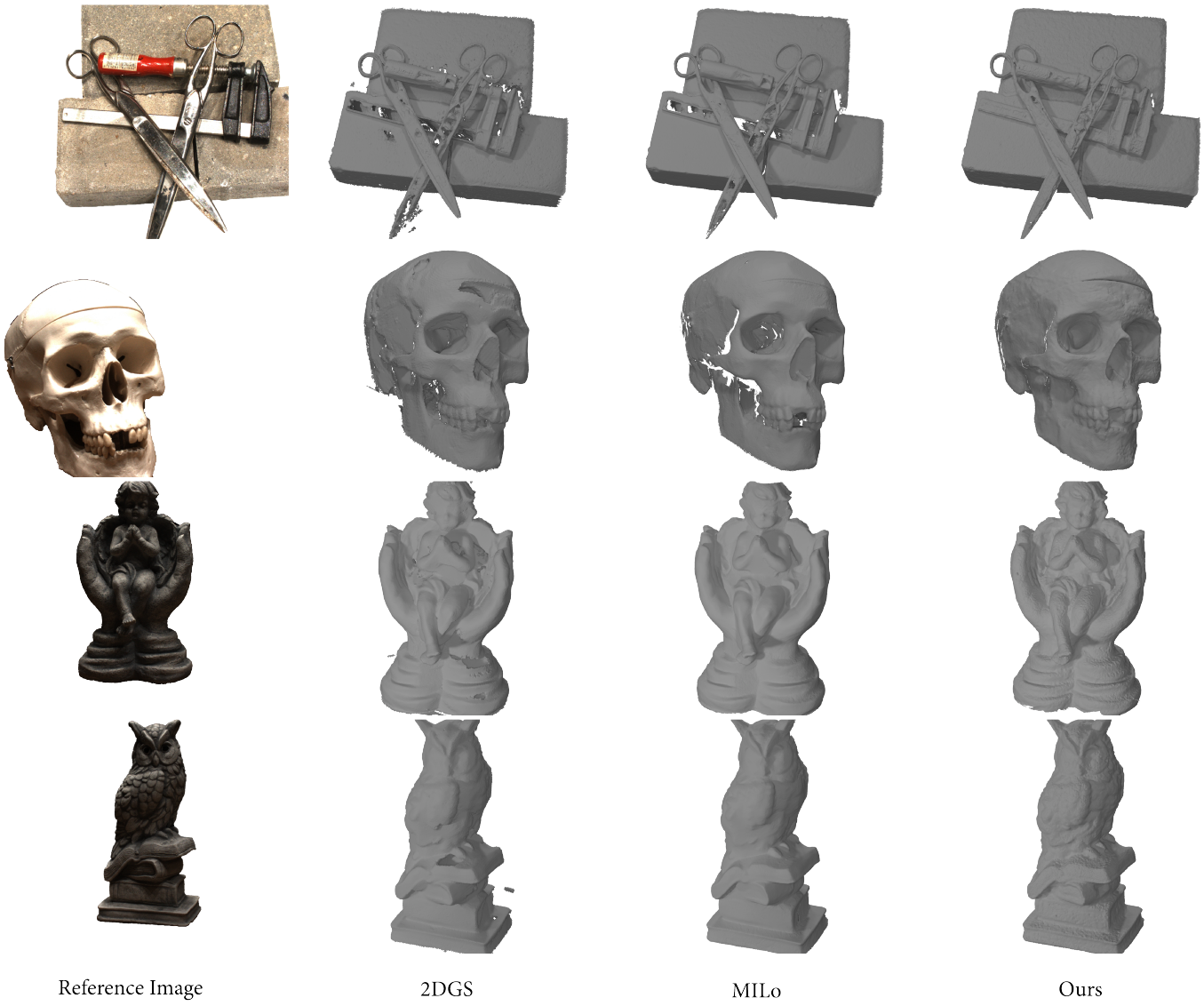

Fig. 6. **Surface Reconstruction on the DTU Dataset [Aanæs et al. 2016]**. Qualitative comparison on four scans (Scan 37, Scan 65, Scan 118, Scan 122) with MILo [Guédon et al. 2025], 2DGS [Huang et al. 2024], and our method.

Tab. 1 and Tab. 2 summarize mesh reconstruction quality on DTU and TnT. On DTU, SDFRaster improves Chamfer distance over splatting-based baselines that rely on post-processing methods to extract meshes, and Fig. 6 shows fewer holes and cleaner surfaces across four scenes. On TnT, SDFRaster achieves higher F1-scores while keeping meshes compact, indicating that the adaptive tetrahedral grid allocates resolution where it is most needed; Fig. 3 shows better thin-structure recovery and finer detail on two TnT scenes and two Mip-NeRF 360 scenes. Compared with implicit SDF baselines, SDFRaster achieves competitive accuracy while retaining the efficiency and scalability of explicit rendering. Tab. 3 further reports the novel view synthesis results on Mip-NeRF 360. Although surface reconstruction is our main goal, our method still maintains competitive rendering quality in terms of PSNR, SSIM, and LPIPS compared with previous approaches.

We also compare against 2DGS and PGSR, which render per-view depth maps and extract meshes via TSDF fusion; this depth fusion pipeline is sensitive to multi-view depth consistency and struggles with thin structures, often producing holes, missing regions, or overly dense meshes. Fig. 5 and Tab. 2 show that SDFRaster yields cleaner topology and preserves sharp edges and thin structures while using fewer mesh vertices and faces. TSDF fusion relies on a uniform voxel grid whose resolution and memory trade-off limits scalability in large scenes, while our adaptive tetrahedral grid allocates capacity where needed and enables background reconstruction without sacrificing foreground detail. By learning a

Table 2. **Quantitative comparison on the Tanks and Temples dataset**. We report the F1-score and output mesh size (MB) across six scenes (higher is better). All results are evaluated with the official evaluation scripts. Best results for implicit methods are highlighted in blue; best results for explicit methods are highlighted in red. Abbreviations: Cat. (Caterpillar), Court. (Courthouse), Ign. (Ignatius), Meet. (Meetingroom), and RadMesh (Radiance Meshes [Mai et al. 2025]).

Method	Barn	Cat.	Court.	Ign.	Meet.	Truck	Mean	Size
NeuS	0.29	0.29	0.17	0.83	0.24	0.45	0.38	-
GeoNeuS	0.33	0.26	0.12	0.72	0.20	0.45	0.35	-
Neuralangelo	0.70	0.36	0.28	0.89	0.32	0.48	0.50	-
SuGaR	0.14	0.16	0.08	0.33	0.15	0.26	0.19	-
2DGS	0.38	0.18	0.13	0.30	0.16	0.30	0.21	658.2
GOF	0.55	0.38	0.29	0.73	0.26	0.56	0.46	605.4
MILo	0.55	0.38	0.26	0.74	0.25	0.60	0.46	176.3
RadMesh	0.30	0.23	0.14	0.35	0.04	0.28	0.22	42.3
Ours	0.54	0.33	0.30	0.73	0.22	0.48	0.43	186.2

Table 3. **Quantitative results for novel view synthesis on the Mip-NeRF360 dataset**. We report PSNR, SSIM, and LPIPS for outdoor and indoor scenes separately. Abbreviations: RadMesh (Radiance Meshes [Mai et al. 2025])

	Outdoor Scenes			Indoor Scenes		
Method	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
2DGS	24.15	0.702	0.288	30.02	0.909	0.213
MILo	24.23	0.740	0.264	28.93	0.919	0.185
RadMesh	24.38	0.721	0.292	30.61	0.920	0.252
Ours	24.22	0.715	0.293	29.58	0.918	0.233

Table 4. **Ablation studies on TnT and DTU.** Left: average F1-score on Tanks and Temples. Right: average Chamfer Distance on DTU.

Tanks and Temples	
Method	F1-score ↑
Baseline	0.3358
Baseline + $\mathcal{L}_{\text{MD}}$	0.3519
Baseline + $\mathcal{L}_{\text{ND}}$	0.4137
Baseline + $\mathcal{L}_{\text{mesh}}$	0.3908
Full model (w/ all losses)	0.4023
Default setting (w/o $\mathcal{L}_{\text{eik}}$)	**0.4289**

DTU	
Method	CD ↓
w/o $\mathcal{L}_{\text{curv}}$	0.69
w/o $\mathcal{L}_{\text{eik}}$	0.72
w/o densification	0.79
w/o culling	0.68
w/o pruning	0.71
Ours (w/ all)	**0.68**

globally consistent SDF and extracting its zero level set directly, we avoid depth-fusion artifacts caused by view-dependent depth noise and obtain more complete surfaces.

4.3 Ablation Study

We conduct ablations on both TnT and DTU to evaluate the contributions of our geometry consistency losses, the Eikonal regularizer, and the adaptive refinement strategy. Tab. 4 summarizes the quantitative impact on TnT. Adding $\mathcal{L}_{\text{MD}}$ yields a modest F1 improvement, while $\mathcal{L}_{\text{ND}}$ brings a larger gain, indicating that normal and depth consistency more effectively stabilizes the field geometry. The mesh consistency term also improves F1, and the combination of these losses further improves performance, showing the mesh and field coupling and field-level normal constraints are complementary.

On bounded object-centric scenes, the Eikonal term improves reconstruction quality on DTU (Tab. 4). The adaptive refinement strategy is also important: densification and pruning improve reconstruction quality, while culling mainly benefits rendering efficiency. On complex unbounded scenes such as TnT, the Eikonal regularizer substantially hinders convergence (Tab. 4). We therefore disable $\mathcal{L}_{\text{eik}}$ on TnT. Although this removes the strict signed-distance constraint on the learned field, the zero level set remains stable for surface reconstruction under image supervision.

Since our SDF can be queried at arbitrary locations, we can also extract surfaces on a uniform grid with Marching Cubes. We compare this to Marching Tetrahedra on our adaptive tetrahedral grid in Fig. 4. Despite the higher uniform resolution, Marching Cubes yields noisier surfaces, local holes and floaters, while our extraction preserves fine details with a more compact mesh, indicating that the SDF optimization is tightly coupled to the tetrahedral grid.

5 Conclusion

We propose SDFRaster, a rasterizable signed distance field framework for end-to-end mesh reconstruction from multi-view images. The method is built on a rasterizable, multi-view consistent geometry representation that enables rapid optimization and direct mesh extraction during optimization, coupling distance field optimization with surface fidelity via differentiable Marching Tetrahedra. It further incorporates a surface-centric adaptive strategy that concentrates capacity near the zero-level set to produce compact meshes.

Experiments show that SDFRaster can produce accurate surfaces and compact meshes, and extends to unbounded scenes. We believe that rasterizable signed distance fields are a promising direction for scalable, fast, and high-fidelity surface reconstruction, providing a principled alternative to volumetric-primitive methods that lack multi-view consistent surface geometry and to implicit methods that require costly dense sampling during optimization.

One limitation of SDFRaster is that it uses a hash-encoded MLP to parameterize the SDF, rather than explicitly storing the SDF value for each vertex. While this network parameterization provides the smoothness inductive bias inherent in MLPs, it also introduces a nontrivial evaluation overhead compared to a purely explicit splatting pipeline. Future directions worth exploring include explicitly storing the SDF on a tetrahedral mesh and replacing the implicit MLP prior with direct smoothness regularization of the field, which would eliminate the need for network queries and potentially significantly accelerate training.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (No.62272433), Anhui Provincial Natural Science Foundation (No.2508085ZD011), Key Science and Technology Project of Anhui Province (No.202523o09050004) and the Fundamental Research Funds for the Central Universities.

References

Henrik Aanæs, Rasmus Ramsbøl Jensen, George Vogiatzis, Engin Tola, and Anders Bjorholm Dahl. 2016. Large-Scale Data for Multiple-View Stereopsis. *International Journal of Computer Vision* 120, 2 (Nov. 2016), 153–168. doi:10.1007/s11263-016-0902-9

Alexandre B inninger, Ruben Wiersma, Philipp Herholz, and Olga Sorkine-Hornung. 2025. TetWeave: Isosurface Extraction using On-The-Fly Delaunay Tetrahedral Grids for Gradient-Based Mesh Optimization. *ACM Trans. Graph.* 44, 4 (8 2025). doi:10.1145/3730851 SIGGRAPH 2025 issue.

Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. 2022. Mip-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, New Orleans, LA, USA, 5460–5469. doi:10.1109/CVPR52688.2022.00539

Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. 2023. Zip-NeRF: Anti-Aliased Grid-Based Neural Radiance Fields. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, Paris, France, 19640–19648. doi:10.1109/ICCV51070.2023.01804

Danpeng C hen, Hai Li, Weicai Ye, Yifan Wang, Weijian Xie, Shangjin Zhai, Nan Wang, Haomin Liu, Hujun Bao, and Guofeng Zhang. 2025. PGSR: Planar-Based Gaussian Splatting for Efficient and High-Fidelity Surface Reconstruction. *IEEE Transactions on Visualization and Computer Graphics* 31, 9 (Sept. 2025), 6100–6111. doi:10.1109/TVCG.2024.3494046

Hongrui Cai, Wanquan Feng, Xuetao Feng, Yan Wang, and Juyong Zhang. 2022. Neural surface reconstruction of dynamic scenes with monocular rgb-d camera. *Advances in Neural Information Processing Systems (NeurIPS)* 35 (2022), 967–981.

Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. 2022b. TensoRF: Tensorial Radiance Fields. In *Computer Vision – ECCV 2022*, Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner (Eds.). Vol. 13692. Springer Nature Switzerland, Cham, 333–350. doi:10.1007/978-3-031-19824-3_20

Hanlin Chen, Tianxin Huang, Gim Hee Lee, Chen Li, Yunsong Wang, and Fangyin Wei. 2024a. VCR-GauS: View Consistent Depth-Normal Regularizer for Gaussian Surface Reconstruction. In *Advances in Neural Information Processing Systems 37*. Neural Information Processing Systems Foundation, Inc. (NeurIPS), Vancouver, BC, Canada, 139725–139750. doi:10.52202/079017-4434

Hanlin Chen, Fangyin Wei, Chen Li, Tianxin Huang, Yunsong Wang, and Gim Hee Lee. 2024b. VCR-GauS: View Consistent Depth-Normal Regularizer for Gaussian Surface Reconstruction. In *Advances in Neural Information Processing Systems*. doi:10.52202/079017-4434

Zhiqin Chen, Andrea Tagliasacchi, Thomas Funkhouser, and Hao Zhang. 2022a. Neural Dual Contouring. *ACM Transactions on Graphics (Special Issue of SIGGRAPH)* 41, 4 (2022).

Brian Curless and Marc Levoy. 1996. A Volumetric Method for Building Complex Models from Range Images. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*. ACM, 303–312. doi:10.1145/237170.237269

Bo Dai, Lihan Jiang, Tao Lu, Yuanbo Xiangli, Linning Xu, and Mulin Yu. 2024a. GSDF: 3DGS Meets SDF for Improved Neural Rendering and Reconstruction. In *Advances in Neural Information Processing Systems 37*. Neural Information Processing Systems Foundation, Inc. (NeurIPS), Vancouver, BC, Canada, 129507–129530. doi:10.52202/079017-4115

Pinxuan Dai, Jiamin Xu, Wenxiang Xie, Xinguo Liu, Huamin Wang, and Weiwei Xu. 2024b. High-quality Surface Reconstruction using Gaussian Surfels. In *ACM SIGGRAPH 2024 Conference Papers*. Association for Computing Machinery, Article 22, 11 pages. doi:10.1145/3641519.3657441

Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. 2022. Plenoxels: Radiance Fields without Neural Networks. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, New Orleans, LA, USA, 5491–5500. doi:10.1109/CVPR52688.2022.00542

Qiancheng Fu, Qingshan Xu, Yew Soon Ong, and Wenbing Tao. 2022. Geo-Neus: Geometry-Consistent Neural Implicit Surfaces Learning for Multi-view Reconstruction. In *Advances in Neural Information Processing Systems*.

Xuan Gao, Chenglai Zhong, Jun Xiang, Yang Hong, Yudong Guo, and Juyong Zhang. 2022. Reconstructing Personalized Semantic Facial NeRF Models From Monocular Video. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)* 41, 6 (2022). doi:10.1145/3550454.3555501

Stephan J. Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. 2021. FastNeRF: High-Fidelity Neural Rendering at 200FPS. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, Montreal, QC, Canada, 14326–14335. doi:10.1109/ICCV48922.2021.01408

Shrisudhan Govindarajan, Daniel Rebain, Kwang Moo Yi, and Andrea Tagliasacchi. 2025. Radiant Foam: Real-Time Differentiable Ray Tracing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 4135–4145.

Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. 2020. Implicit Geometric Regularization for Learning Shapes. In *Proceedings of the 37th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 119)*, Hal Daumé III and Aarti Singh (Eds.). PMLR, 3789–3799.

Antoine Guédon, Diego Gomez, Nissim Maruani, Bingchen Gong, George Drettakis, and Maks Ovsjanikov. 2025. MILo: Mesh-In-the-Loop Gaussian Splatting for Detailed and Efficient Surface Reconstruction. *ACM Transactions on Graphics* 44, 6 (Dec. 2025), 1–15. doi:10.1145/3763339

Antoine Guédon and Vincent Lepetit. 2024. SuGaR: Surface-Aligned Gaussian Splatting for Efficient 3D Mesh Reconstruction and High-Quality Mesh Rendering. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2024). doi:10.1109/cvpr52733.2024.00512

Zhizhong Han, Yu-Shen Liu, and Wenyuan Zhang. 2024. Neural Signed Distance Function Inference through Splatting 3D Gaussians Pulled on Zero-Level Set. In *Advances in Neural Information Processing Systems 37*. Neural Information Processing Systems Foundation, Inc. (NeurIPS), Vancouver, BC, Canada, 101856–101879. doi:10.52202/079017-3231

Jan Held, Sanghyun Son, Renaud Vandeghen, Daniel Rebain, Matheus Gadelha, Yi Zhou, Anthony Cioppa, Ming C. Lin, Marc Van Droogenbroeck, and Andrea Tagliasacchi. 2025. MeshSplatting: Differentiable Rendering with Opaque Meshes. *arXiv* (2025).

Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2024. 2D Gaussian Splatting for Geometrically Accurate Radiance Fields. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*. ACM, Denver CO USA, 1–11. doi:10.1145/3641519.3657428

Boyi Jiang, Yang Hong, Hujun Bao, and Juyong Zhang. 2022. SelfRecon: Self Reconstruction Your Digital Avatar from Monocular Video. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. 2006. Poisson Surface Reconstruction. 10 pages pages. doi:10.2312/SGP/SGP06/061-070

Michael Kazhdan and Hugues Hoppe. 2013. Screened Poisson Surface Reconstruction. *ACM Transactions on Graphics* 32, 3 (June 2013), 1–13. doi:10.1145/2487228.2487237

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuehler, and George Drettakis. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics* 42, 4 (Aug. 2023), 1–14. doi:10.1145/3592433

Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. 2017. Tanks and Temples: Benchmarking Large-Scale Scene Reconstruction. *ACM Transactions on Graphics* 36, 4 (Aug. 2017), 1–13. doi:10.1145/3072959.3073599

Jonas Kulhanek and Torsten Sattler. 2023. Tetra-NeRF: Representing Neural Radiance Fields Using Tetrahedra. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 18458–18469.

Jiahe Li, Jiawei Zhang, Xiao Bai, Jin Zheng, Xin Ning, Jun Zhou, and Lin Gu. 2024. DNGaussian: Optimizing Sparse-View 3D Gaussian Radiance Fields with Global-Local Depth Normalization. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Seattle, WA, USA, 20775–20785. doi:10.1109/CVPR52733.2024.01963

Shujuan Li, Yu-Shen Liu, and Zhizhong Han. 2025. GaussianUDF: Inferring Unsigned Distance Functions through 3D Gaussian Splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 27113–27123.

Zhaoshuo Li, Thomas Müller, Alex Evans, Russell H. Taylor, Mathias Unberath, Ming-Yu Liu, and Chen-Hsuan Lin. 2023. Neuralangelo: High-Fidelity Neural Surface Reconstruction. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Vancouver, BC, Canada, 8456–8465. doi:10.1109/CVPR52729.2023.00817

William E. Lorensen and Harvey E. Cline. 1987. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*. ACM, 163–169. doi:10.1145/37401.37422

Tao Lu, Mulin Yu, Linning Xu, Yuanbo Xiangli, Limin Wang, Dahua Lin, and Bo Dai. 2024. Scaffold-GS: Structured 3D Gaussians for View-Adaptive Rendering. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Seattle, WA, USA, 20654–20664. doi:10.1109/CVPR52733.2024.01952

Xiaoyang Lyu, Yang-Tian Sun, Yi-Hua Huang, Xiuzhe Wu, Ziyi Yang, Yilun Chen, Jiangmiao Pang, and Xiaojuan Qi. 2024. 3DGSR: Implicit Surface Reconstruction with 3D Gaussian Splatting. *ACM Transactions on Graphics* 43, 6 (Dec. 2024), 1–12. doi:10.1145/3687952

Alexander Mai, Trevor Hedstrom, George Kopanas, Janne Kontkanen, Falko Kuester, and Jonathan T. Barron. 2025. Radiance Meshes for Volumetric Reconstruction. doi:10.48550/ARXIV.2512.04076

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 *Computer Vision – ECCV 2020*, Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (Eds.). Vol. 12346. Springer International Publishing, Cham, 405–421. doi:10.1007/978-3-030-58452-8_24

Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. 2022. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. *ACM Transactions on Graphics* 41, 4 (July 2022), 1–15. doi:10.1145/3528223.3530127

Richard A. Newcombe, Andrew Fitzgibbon, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J. Davison, Pushmeet Kohi, Jamie Shotton, and

Steve Hodges. 2011. KinectFusion: Real-time Dense Surface Mapping and Tracking. In *2011 10th IEEE International Symposium on Mixed and Augmented Reality*. IEEE, Basel, 127–136. doi:10.1109/ISMAR.2011.6092378

Seunghun Oh, Jaesung Choe, Dongjae Lee, Daeun Lee, Seunghoon Jeong, Yu-Chiang Frank Wang, and Jaesik Park. 2025. SVRecon: Sparse Voxel Rasterization for Surface Reconstruction. doi:10.48550/ARXIV.2511.17364

Lukas Radl, Michael Steiner, Mathias Parger, Alexander Weinrauch, Bernhard Kerbl, and Markus Steinberger. 2024. StopThePop: Sorted Gaussian Splatting for View-Consistent Real-time Rendering. *ACM Transactions on Graphics* 43, 4 (July 2024), 1–17. doi:10.1145/3658187

Christian Reiser, Stephan Garbin, Pratul P. Srinivasan, Dor Verbin, Richard Szeliski, Ben Mildenhall, Jonathan T. Barron, Peter Hedman, and Andreas Geiger. 2024. Binary Opacity Grids: Capturing Fine Geometric Detail for Mesh-Based View Synthesis. *SIGGRAPH* (2024).

Johannes Lutz Schönberger and Jan-Michael Frahm. 2016. Structure-from-Motion Revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Johannes L. Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. 2016. Pixelwise View Selection for Unstructured Multi-View Stereo. In *Computer Vision – ECCV 2016*, Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling (Eds.). Vol. 9907. Springer International Publishing, Cham, 501–518. doi:10.1007/978-3-319-46487-9_31

Tianchang Shen, Jun Gao, Kangxue Yin, Ming-Yu Liu, and Sanja Fidler. 2021. Deep Marching Tetrahedra: a Hybrid Representation for High-Resolution 3D Shape Synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Tianchang Shen, Jacob Munkberg, Jon Hasselgren, Kangxue Yin, Zian Wang, Wenzheng Chen, Zan Gojcic, Sanja Fidler, Nicholas Sharp, and Jun Gao. 2023. Flexible Isosurface Extraction for Gradient-Based Mesh Optimization. *ACM Transactions on Graphics* 42, 4 (Aug. 2023), 1–16. doi:10.1145/3592430

Kaiwen Song, Jinkai Cui, Zherui Qiu, and Juyong Zhang. 2025. StructuredField: Unifying Structured Geometry and Radiance Field. doi:10.48550/ARXIV.2501.18152

Kaiwen Song, Xiaoyi Zeng, Chenqu Ren, and Juyong Zhang. 2024. City-on-Web: Real-time Neural Rendering of Large-scale Scenes on the Web. In *European Conference on Computer Vision (ECCV)*.

Cheng Sun, Jaesung Choe, Charles Loop, Wei-Chiu Ma, and Yu-Chiang Frank Wang. 2025. Sparse Voxels Rasterization: Real-time High-fidelity Radiance Field Rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 16187–16196.

Cheng Sun, Min Sun, and Hwann-Tzong Chen. 2022. Direct Voxel Grid Optimization: Super-fast Convergence for Radiance Fields Reconstruction. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, New Orleans, LA, USA, 5449–5459. doi:10.1109/CVPR52688.2022.00538

G.M. Treece, R.W. Prager, and A.H. Gee. 1999. Regularised Marching Tetrahedra: Improved Iso-Surface Extraction. *Computers & Graphics* 23, 4 (Aug. 1999), 583–598. doi:10.1016/S0097-8493(99)00076-X

Matias Turkulainen, Xuqian Ren, Iaroslav Melekhov, Otto Seiskari, Esa Rahtu, and Juho Kannala. 2025. DN-Splatter: Depth and Normal Priors for Gaussian Splatting and Meshing. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. IEEE, Tucson, AZ, USA, 2421–2431. doi:10.1109/WACV61041.2025.00241

Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. 2021b. NeuS: Learning Neural Implicit Surfaces by Volume Rendering for Multi-view Reconstruction. *NeurIPS* (2021).

Xueying Wang, Yudong Guo, Zhongqi Yang, and Juyong Zhang. 2021a. Prior-Guided Multi-View 3D Head Reconstruction. *IEEE Transactions on Multimedia* 24 (2021), 4028–4040. https://api.semanticscholar.org/CorpusID:235790368

Yiming Wang, Qin Han, Marc Habermann, Kostas Daniilidis, Christian Theobalt, and Lingjie Liu. 2023. NeuS2: Fast Learning of Neural Implicit Surfaces for Multi-view Reconstruction. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, Paris, France, 3272–3283. doi:10.1109/ICCV51070.2023.00305

Tong Wu, Jiaqi Wang, Xingang Pan, Xudong Xu, Christian Theobalt, Ziwei Liu, and Dahua Lin. 2023. Voxurf: Voxel-based Efficient and Accurate Neural Surface Reconstruction. In *The Eleventh International Conference on Learning Representations*.

Baixin Xu, Jiangbei Hu, Jiaze Li, and Ying He. 2024. GSurf: 3D Reconstruction via Signed Distance Fields with Direct Gaussian Supervision. doi:10.48550/ARXIV.2411.15723

Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. 2018. MVSNet: Depth Inference for Unstructured Multi-view Stereo. In *Computer Vision – ECCV 2018*, Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss (Eds.). Vol. 11212. Springer International Publishing, Cham, 785–801. doi:10.1007/978-3-030-01237-3_47

Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. 2021. Volume Rendering of Neural Implicit Surfaces. In *Thirty-Fifth Conference on Neural Information Processing Systems*.

Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Ronen Basri, and Yaron Lipman. 2020. Multiview Neural Surface Reconstruction by Disentangling Geometry and Appearance. *Advances in Neural Information Processing Systems* 33 (2020).

Zehao Yu, Anpei Chen, Bozidar Antic, Songyou Peng, Apratim Bhattacharyya, Michael Niemeyer, Siyu Tang, Torsten Sattler, and Andreas Geiger. 2022a. SDFStudio: A Unified Framework for Surface Reconstruction.

Zehao Yu and Shenghua Gao. 2020. Fast-MVSNet: Sparse-to-Dense Multi-View Stereo With Learned Propagation and Gauss-Newton Refinement. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Seattle, WA, USA, 1946–1955. doi:10.1109/CVPR42600.2020.00202

Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. 2022b. MonoSDF: Exploring Monocular Geometric Cues for Neural Implicit Surface Reconstruction. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Zehao Yu, Torsten Sattler, and Andreas Geiger. 2024. Gaussian Opacity Fields: Efficient Adaptive Surface Reconstruction in Unbounded Scenes. *ACM Transactions on Graphics* 43, 6 (Dec. 2024), 1–13. doi:10.1145/3687937

Baowen Zhang, Chuan Fang, Rakesh Shrestha, Yixun Liang, Xiaoxiao Long, and Ping Tan. 2026. RaDe-GS: Rasterizing Depth in Gaussian Splatting. *ACM Transactions on Graphics* 45, 2, Article 19 (2026). doi:10.1145/3789201

Ziyu Zhang, Binbin Huang, Hanqing Jiang, Liyang Zhou, Xiaojun Xiang, and Shuhan Shen. 2025. Quadratic Gaussian Splatting: High Quality Surface Reconstruction with Second-order Geometric Primitives. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 28260–28270.