

GSSA: Gaussian Surfels with Spatial Awareness for Surface Reconstruction

PREPRINT

Haojun Tang¹, Siyuan Zou^{1,2,*}, Hongbo Pan¹, Yixin Lu³, and Shun Zhou¹

¹School of Geoscience and Info-Physics, Central South University

²Key Laboratory of China-ASEAN Satellite Remote Sensing Applications, Ministry of Natural Resources

³Guangxi Beibu Gulf Investment Group Co., Ltd.

ABSTRACT

3D Gaussian Splatting is effective in multi-view surface reconstruction tasks, fundamental to photogrammetry applications. Mainstream pipelines focus on optimizing for explicit Gaussians to enhance volumetric rendering quality, typically relying on rendering-based Truncated Signed Distance Function (TSDF) methods for reconstruction, while the training and meshing often suffer from missing details and geometric distortions under occlusions or restricted viewpoints. In this paper, we focus on a new problem definition: directly formulating implicit surface representation exploiting geometric features of trained Gaussian Surfels. For this problem, we introduce **Gaussian Surfels with Spatial Awareness (GSSA)** for reconstruction, a spatially aware reconstruction framework that optimizes for Surfel distributions from geometric primitives via a 3D refiner, and constructs SDF by integrating signed distances from voxel vertices to nearby Surfels. Compared with state-of-the-art GS-based, especially Surfel-based, reconstruction methods, GSSA achieves both competitive geometric accuracy and efficiency. On terrestrial (indoor object) and airborne (photogrammetry) reconstruction benchmarks, GSSA maintains a competitive balance between reconstruction time and accuracy. Further migration experiments demonstrate that GSSA can also be integrated as a plug-and-play module into general Surfel-based pipelines. The source code is publicly available at <https://github.com/DonaldTrump-coder/GSSA>.

Keywords Photogrammetry · Surface Reconstruction · Gaussian Surfels · Gaussian Splatting · Signed Distance Function

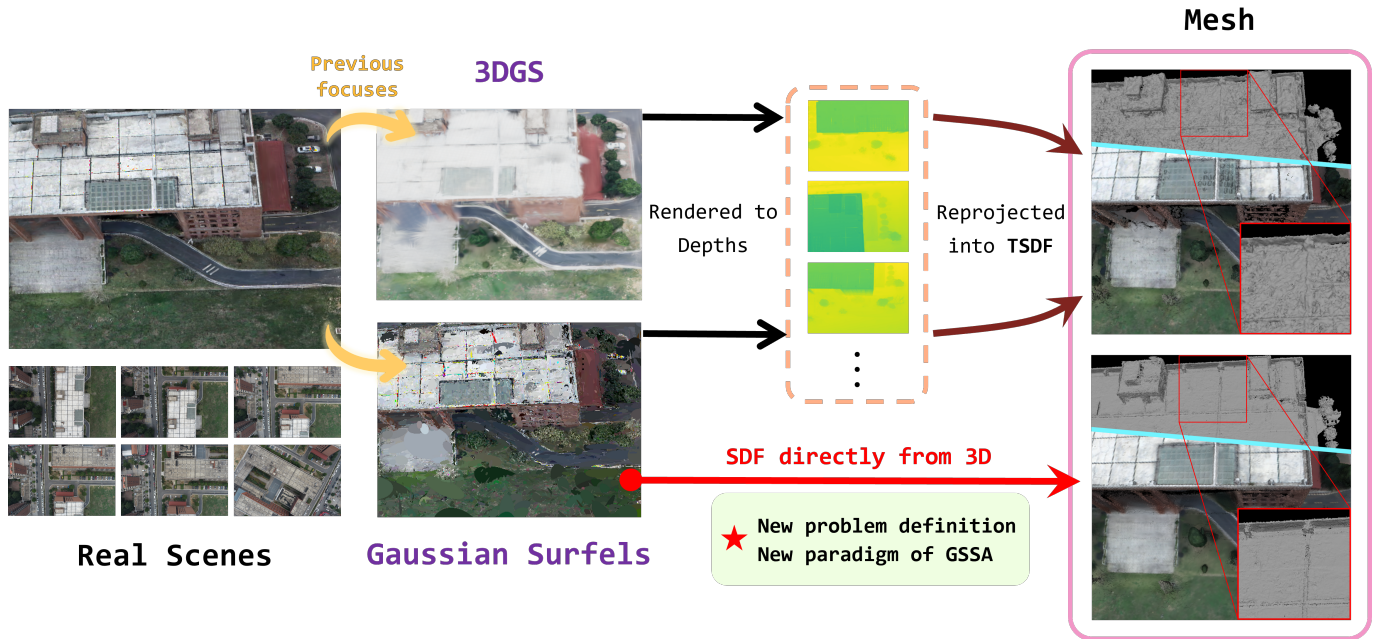


Figure 1: Comparison between previous rendering-based reconstruction paradigm and GSSA’s paradigm.

1 INTRODUCTION

Surface reconstruction from scene representations is a fundamental task in photogrammetry and computer vision [1, 2], supporting a wide range of applications such as urban modeling [3, 4] terrain reconstruction [5, 6], infrastructure mapping [7, 8], her-

itage archiving [9], and autonomous driving [10]. Various studies in computer graphics [11], survey [12, 13], and SLAM [14] employ Signed Distance Function (SDF) derived from depth maps to extract continuous surfaces from discrete representations. SDF provides an implicit representation of surfaces, from which triangular meshes can be efficiently extracted using iso-

Table 1. Comparison of strategy characteristics across mainstream SDF-based methods.

Method	Gaussian Primitives	Training Pipeline Modification	Directly from Rendering	Source of SDF Values
PGSR	Surfels	✓	✓	2D
2DGS	Surfels	✓	✓	2D
GOF	Ellipsoids	✓	✓	2D
NeuS	None	✓	✓	2D
Ours	Surfels	✗	✗	3D

surface extraction algorithms such as Marching Cubes [15] and Marching Tetrahedra [16, 17]. In conventional photogrammetric workflows, dense multi-view matching is used to generate depth maps, which can then be integrated into SDF or point clouds for surface reconstruction [18, 19]. Recently, explicit 3D Gaussian Splatting [20] has emerged as an efficient representation for rendering and transferred from Novel View Synthesis to reconstruction tasks [21, 22], with previous work providing insights by projecting Gaussians into depths for SDF [23]. Therefore, constructing SDFs from explicit Gaussian representations forms a significant bridge between continuous surfaces and discrete Gaussian primitives [24], showing great promise for reconstruction from remote sensing and photogrammetry data. Recent Surfel-based Splatting [25] enables more precise surface reconstruction. However, existing approaches combining Gaussian Surfels with SDF [26, 27] primarily focus on optimizing Gaussians through volumetric rendering, typically generating iso-surfaces by rendering depth from multiple viewpoints and fusing them into TSDF volumes. Projecting Gaussians into images and back-projecting to compute SDF values cannot fully leverage Surfels’ spatial geometries. In this redundant 2D–3D switching paradigm, loss of fine-grained details is brought out, irregularities in Gaussian primitives can distort surfaces, and occlusions can result in missing structures [12, 28], especially in the reconstruction of real-world objects and vertical photogrammetry tasks with limited viewpoints and complex lighting conditions.

As discussed above, multi-view SDF fusion is prone to cross-view depth inconsistencies, viewpoint occlusions, and unstable viewpoint coverage. A direct global reconstruction strategy would circumvent these limitations by operating on the Gaussian primitives themselves rather than on per-view rendered samples. Leveraging the observation that Gaussian Surfels trained by multi-view optimization effectively conform to the actual scene distribution [26], we shift the focus away from optimizing Gaussian representations from multi-view images, and instead propose a novel problem: constructing implicit function from Gaussian Surfels trained by volumetric rendering, particularly via spatial features of Gaussian Surfels. Notably, this formulation decouples surface reconstruction from the multi-view training pipeline via NVS: optimization of Gaussians and extraction of surfaces. While a substantial body of work has advanced the optimization strategies for producing high-quality Gaussians, little attention has been devoted to innovating the implicit representation construction in the extraction stage. Under this decoupling, high-quality Gaussians produced by existing open-source methods can be readily repurposed for surface extraction, without repeating costly training procedures. Furthermore, this focus inherently demands stronger generalization across diverse training strategies, aligning naturally with a modular engineering philosophy.

We introduce a novel reconstruction framework, GSSA, with a spatial-aware paradigm for this novel focus mentioned above. As shown in **Figure 1**, upper pipelines illustrate previous Gaussian-based reconstruction, where Gaussians are rendered into multi-view depth maps and fused into TSDF for surface extraction. In contrast, the proposed method focuses on directly obtaining SDF from geometric features of Surfels without rendering. GSSA integrates for the voxel-based SDF via signed distances from vertices to nearby Surfels, which are previously denoised and filled in a 3D refiner. The pipeline seeks for surface representations highly consistent with the distribution of Gaussian Surfels, enabling surface reconstruction with more accuracy and fidelity.

In addition, a tabular comparison of methodological characteristics against mainstream SDF-based reconstruction methods (state-of-the-art NVS-based methods such as PGSR [25], 2DGS [26], GOF [29], etc.) is presented in **Table 1**, highlighting the proposed GSSA as a unique, plug-and-play reconstruction paradigm with spatial awareness to Gaussian Surfels. Since our proposed GSSA does not introduce a new training strategy, it is able to serve as a plug-and-play module applicable to Gaussian Surfels pre-trained by other methods such as 2DGS.

The key contributions of this study are outlined as follows:

1. A new problem defined for reconstruction: implicitly representing surfaces from trained Gaussian Surfels by explicitly leveraging 3D geometric features, providing an alternative to previous rendering-based paradigms.
2. A reconstruction framework containing a 3D-aware refiner that denoises and fills the Gaussian Surfels, followed by a voxel-based reconstruction method that integrates signed distances from voxel vertices to the nearby refined Gaussian Surfels, enabling robust and accurate SDF construction.
3. Comprehensive experiments demonstrate that the framework allows for more accurate reconstruction compared to baselines in handling reconstruction tasks such as terrestrial object reconstruction and airborne photogrammetry reconstruction.

2 RELATED WORK

2.1 Mesh Reconstruction

Point-based approaches. Surface reconstruction based on point cloud is a classic perspective in 3D reconstruction, still widely used in reconstruction tasks from photogrammetry [7, 30]. Delaunay triangulation [31] constructs tetrahedra from point clouds to establish topological relationships, with geometric details preserved through algorithms such as α -Shapes [32] or s-t cuts optimization [33]. However, the Delaunay method is prone

to noise interference, generally requiring dense point clouds with low spatial error to produce reliable triangulations, which are often generated by Multi-View Stereo (MVS) [34] methods or Airborne Laser Scanning (ALS) [35] methods, adding significant extra processing time to the pipeline. The work [36] constructs an initial mesh with Delaunay initialization and s-t cuts optimization, followed by refinement with multi-view consistency. However, this optimization method scales poorly with point cloud density, leading to substantial time and CPU computational overhead. Recent learning-based approaches such as BSP-Net [37] and PointTriNet [38] leverage deep networks for triangulation topologies construction directly from explicit point cloud inputs. This learning-based paradigm leverages the strong representational capacity of Neural Networks to reason about the underlying geometric structures of point clouds. However, such data-driven approaches need ground truth surface for supervision and the generalization heavily depends on the type, scale and other characteristics of the training data. Consequently, constrained by the scale and the associated training cost, these learning methods have mostly been applied to reconstruction on small-scale data (chairs, etc.), and have rarely been deployed at scenes of photogrammetry scale. Overall, the point-based methods mentioned above either rely on dense point clouds, optimize for the whole mesh directly, or require large Neural Networks with large-scale discrete inputs, yielding a certain robustness at the expense of lower optimization efficiency.

Iso-surface-based approaches. In contrast to pipelines directly using point clouds for topologies, iso-surface-based strategies extract surfaces indirectly via learned continuous implicit fields and iso-surface information, which are more robust to noise [39]. The surfaces can then be extracted from iso-surfaces via such algorithms as Marching Cubes [15], Marching Tetrahedra [17], etc., depending on the discretization of the underlying implicit fields. Poisson surface reconstruction [40, 41] is among the widely used, obtaining indicator function fields from oriented point clouds with a Poisson Equation, yet tending to result in overly smooth surfaces from inaccurate normal estimations. Global SDF-based methods, especially TSDF [42], are a volumetric, extensively used approach that computes iso-surface from depths. Further optimized TSDF works, such as Voxel Hashing [43], InfiniTAM [44, 45], and octree-based TSDF [46], achieve significant reductions in memory requirements through the use of efficient CPU storage structures. Due to its features of integration from depth ranges, TSDF is also employed in real-time reconstruction systems [14]. However, accurate TSDF reconstruction relies on multi-view depth rendering consistency, and tends to accumulate inaccuracies under occlusions or limited viewpoint coverages. Accordingly, in high-precision photogrammetric reconstruction tasks, both Poisson reconstruction and TSDF-based methods typically need dense MVS-derived point clouds as input, rather than sparse points [47, 48].

Neural volumetric methods [49] are recently utilized to construct iso-surface based on complex point-wise or image features, and surfaces can be extracted in the form of occupancy fields [50], unsigned distance fields [51] and signed distance fields [52, 53]. Differentiable Poisson Solver [54] also provides a learning-based paradigm for implicit Poisson reconstruction methods to process unoriented, noisy input point clouds. However, neural methods may suffer from lack of geometric interpretability, limitation of training data and consumption of prolonged training time.

In addition, for most practical cases, such deep learning-based approaches are even outperformed by standard Poisson reconstruction, mainly due to the limited generalization capacity of the training data [55].

2.2 Novel View Synthesis

The NVS problem remains a challenging topic in computer graphics [56] as well as aerial photogrammetry [57]. By pursuing multi-view consistency on 3D scene representations, it is significant for both novel view generation as well as reality-based 3D reconstruction. Thus, the multi-view consistency optimization is also regarded as a potential approach for surface reconstruction. NeRF [58] and subsequent implicit representation methods [59] have achieved impressive synthesis in NVS via neural volume rendering. Concurrently, numerous NeRF-based methods [60, 61] reconstruct surfaces volumetrically driven by implicit rendering and NVS objectives, with a key insight of using SDF as the density field for volumetric rendering. Despite the strong rendering fidelity, training and rendering efficiency remain a persistent bottleneck for such implicit volumetric methods.

In subsequent work, 3D Gaussian Splatting [20], featuring explicit, differentiable, and parallel rendering, has significantly improved the efficiency of training for NVS while retaining competitive rendering quality. Since this technique explicitly represents the scene via anisotropic 3D Gaussian ellipsoids, later explorations for 3DGS not only focus on NVS problems [62, 63] but also extend to 3D point cloud [64, 65] and surface [21, 22] reconstruction. Such GPU-based multi-view consistency optimization achieves scene densification substantially faster than widely used MVS pipelines. Similarly, we leverage this fast scene optimization to extract mesh, providing more insights for the photogrammetry and reconstruction community.

2.3 Gaussian Representation for Surface Reconstruction

As mentioned above, 3D Gaussian Splatting inspires on efficient meshing. GaMeS [22] and SuGAR [21] directly bind Gaussians to the surface, enabling rapid production of optimized mesh via the features of fast rendering. Those approaches utilize the geometry of Gaussians explicitly for the surface, but they introduce limitations such as insufficient topology for optimized mesh and poor densification for Gaussians. GauS [23] constructs iso-surfaces based on rendered depth from Gaussian ellipsoids via TSDF integration. GOF [29] establishes an opacity field through rendering-based training in place of the depth-based geometry used in previous methods, from which the surface level-set is directly extracted, however still remaining a rendering paradigm for SDF construction grounded in multi-view imagery. These methods use Gaussian ellipsoids as representation, resulting in the lack of multi-view consistency in rendered normal and depths. In addition, GOF employs Marching Tetrahedra [17] over dense tetrahedral grids, requiring higher memory consumption than Marching Cubes [15] on voxel-based SDF methods.

Several previous works including 2DGS [26] and PGSR [25] use Gaussian Surfels to represent the scene, enabling both rapid training and close conformation to surfaces. This demonstrates that fitting scene surfaces with planar elliptical Gaussians contributes

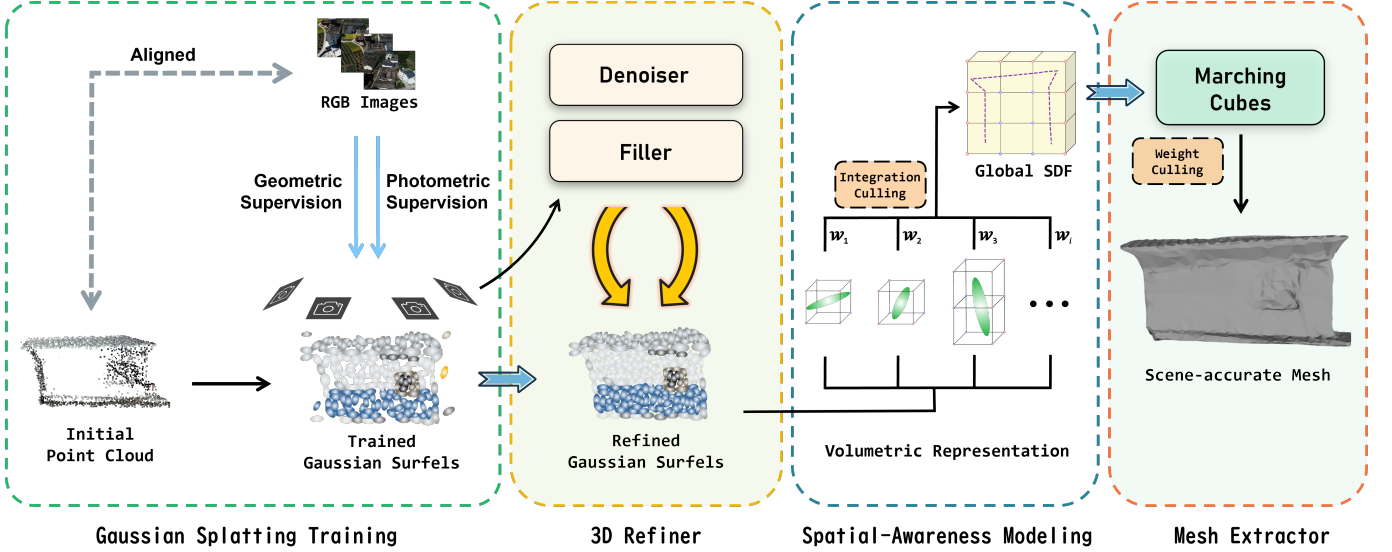


Figure 2. The overview of GSSA.

to a more stable optimization process. Various rendering-based surface extraction methods [66, 67] are proposed for this representation, leading to more geometrically accurate meshing. However, those previous rendering-based paradigms still rely on rendered depths for SDF surface representation, suffering from rendered distortions and occlusions due to the previously mentioned viewpoint limitations.

3 PRELIMINARIES

Before introducing our method, we first present our novel problem focus and its definition in reconstruction tasks in 3.1. Then we display the Gaussian Surfel primitives used to construct the SDF representation in 3.2, as the background and foundation for our approach.

3.1 Problem Description

Surface reconstruction task takes a set of calibrated multi-view images as input and produces an opaque 3D mesh consisting of vertices and faces. Previous works for surface reconstruction from Gaussian-based explicit representations mainly seek for more optimized Gaussians, such as using Surfel representations, adding geometric supervisions, etc. Yielding more refined rendered results, these works primarily rely on rendering-based TSDF methods for surface representation:

$$\text{TSDF}(\mathbf{p}) = \begin{cases} D(\mathbf{p}) - \text{depth}, & \text{if } |D(\mathbf{p}) - \text{depth}| \leq \delta \\ \delta, & \text{if } (D(\mathbf{p}) - \text{depth}) > \delta \\ -\delta, & \text{if } (D(\mathbf{p}) - \text{depth}) < -\delta \end{cases} \quad (1)$$

where $\mathbf{p}$ denotes a vertex in voxel space, $D(\mathbf{p})$ denotes depth of the vertex projected into a camera, and depth represents the corresponding rendered depth on camera pixel.

Unlike prior works, we focus on a new problem defined as constructing implicit iso-surfaces from trained Gaussian Surfels, since we believe Surfel structures, trainable with rendering differentiability, can follow the surface geometry and possess the structural consistency well enough.

Gaussian Surfels are Gaussian ellipsoidal patches, and we assume that we can obtain Gaussian Surfels distributed consistently with real scenes in general via volumetric-rendering based trainings. Then we propose our method, aiming at the problem we present above and thus forming a new paradigm in reconstruction via Gaussian representations.

3.2 Primitives Definition

Means. The spatial position of a Gaussian Surfel $\mathbf{x} \in \mathbb{R}^3$; the coordinates of each ellipsoidal Surfel center.

Rotations. Rotation matrix $\mathbf{R} \in SO(3)$, defining the orientation of the Surfel. We follow the similar implementation to the work [26], $\mathbf{R} = [\mathbf{u} \ \mathbf{v} \ \text{normal}]$. $\mathbf{u}$, $\mathbf{v}$ denote the principal axes (major and minor), while normal specifies the Surfel’s normal. It should be noted that, all vectors associated with the Surfel’s parameters $\mathbf{u}$, $\mathbf{v}$, and normal in this paper are unit vectors, used solely to denote direction.

Scales. The lengths of the two principal axes of the Surfel, $s_u, s_v \in \mathbb{R}$, along the directions of $\mathbf{u}, \mathbf{v}$, respectively.

Opacities. $\alpha \in \mathbb{R}$ denotes the weight of each Gaussian Surfel in the radiance field rendering. In 3D space, it can also be regarded as a confidence reflecting how closely the Surfel fits the scene surface.

SHs. The spherical harmonic (SH) features of Gaussian Surfels are used to represent view-dependent colors, denoted as $\mathbf{f}_n \in \mathbb{R}^{3 \times (k+1)^2}$, where k is the order of SH.

4 MATERIALS AND METHODS

We propose GSSA, a novel paradigm-level framework that directly extracts mesh from Gaussian Surfels, with the overview in Figure 2. During training, Gaussian Surfels are optimized under multi-view supervisions. Subsequently, in our designed 3D refiner, Surfels are denoised based on spatial-feature thresholds and filled on planar surfaces. Then a weighted SDF field

is integrated from geometric primitives of Surfels for the eventual surface extraction via Marching Cubes. In the following sections, we describe our spatially aware 3D refiner for Surfels in 4.1 and SDF construction intuitive insights in 4.2.

4.1 Gaussian Surfels 3D Refiner

Owing to the lack of context-aware constraints, limited training viewpoints and constrained densification strategies during the 2D multi-view consistency optimization, the resulting Gaussian Surfels exhibit geometric inconsistencies relative to the real scene, such as floating outliers, oversized Surfels and holes in relatively large planar regions. Since our proposed paradigm reconstructs a consistent scene surface directly from the geometric distribution of Surfels, the trained Surfels need to be refined under the guidance of 3D spatial coherence. Thus, we introduce a 3D refiner module before establishing the SDF representation. This module leverages the spatial awareness of Gaussian Surfels and directly optimizes them through geometric cues. The refiner includes a subtractive denoiser and an additive planar filler.

Denoiser. The denoiser directly identifies the obviously noisy Gaussian Surfels irrelevant to the real structures according to their minimum scales and distances to nearest neighbors:

$$\min(s_u, s_v) > \epsilon_s \quad (2)$$

$$\|\mathbf{x} - \mathbf{x}_{nn}\| > \epsilon_{dn} \quad (3)$$

ϵ_s is a threshold for scales of Surfels, and ϵ_{dn} is a distance threshold. $\mathbf{x}_{nn}$ is the mean of a Surfel’s nearest neighbor. In this way, we can identify and pre-remove the noisy Gaussian Surfels with excessively large sizes and spatial outlier tendencies.

Filler. In planar regions, a filler is used to perform the filling of Surfels directly in 3D space. Denser Surfel coverage also improves the robustness of SDF integration, suppressing individual noise through weighted averaging.

To identify planar regions, we consider, for each Surfel, its k -nearest Surfel neighbors based on the means and perform a joint judgment using both distance and normal consistency:

$$s \in \text{Plane} \quad \text{if} \quad \forall s_j \in \mathcal{N}_k(s), \begin{cases} \|\mathbf{x} - \mathbf{x}_j\| < \epsilon_d, \\ |(\mathbf{x} - \mathbf{x}_j) \cdot \mathbf{n}| < \epsilon_p, \\ |\mathbf{n} \cdot \mathbf{n}_j| > \epsilon_n \end{cases} \quad (4)$$

where $\mathbf{x}$ denotes the means of Gaussian Surfels, $\mathbf{n}$ denoting the normal of Surfels (the abbreviation of **normal**). s or s_j refers to Surfel elements, which is different from the scale vectors s_u , s_v or s . The filling needs to be conducted between two Surfels. Then we set its nearest k_c Surfel neighbors in those k neighbors to be the candidates along with the Surfel.

For a pair of candidate Gaussian Surfels s_i, s_j to be filled, we first determine the principal axis of each Surfel along which filling may be needed:

$$\mathbf{m}_i = \begin{cases} \mathbf{u}_i, & |(\mathbf{x}_i - \mathbf{x}_j) \cdot \mathbf{u}_i| \geq |(\mathbf{x}_i - \mathbf{x}_j) \cdot \mathbf{v}_i| \\ \mathbf{v}_i, & \text{otherwise} \end{cases} \quad (5)$$

$\mathbf{m}_j$ is determined in the same manner. When the sum of the lengths of the two principal axes is small enough (smaller than a certain fraction of the distance between the Surfels $n\|\mathbf{x}_i - \mathbf{x}_j\|$),

filling is required along the principal axes of Gaussian Surfels. In this case, a third Gaussian Surfel is inserted. Its mean $\mathbf{x}$ is defined as the point closest to the two principal axes, as illustrated in Figure 3. This constitutes the core idea of our pipeline, although the associated expressions can become rather involved.

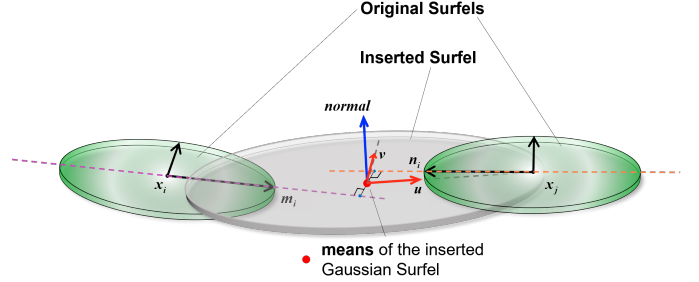


Figure 3. The insertion of New Gaussian Surfel. The mean is at the point closest to the two principal axes.

Here we show how to obtain the closest point and corresponding Gaussian primitives:

The two points on the principal axes corresponding to the original two Gaussian Surfels can be described by the parameters s, t as

$$\begin{cases} \mathbf{p}_i = \mathbf{x}_i + s \mathbf{m}_i \\ \mathbf{p}_j = \mathbf{x}_j + t \mathbf{m}_j \end{cases} \quad (6)$$

We minimize the distance $d(s, t)$ between the two points to solve for s, t . Thus we set both $\frac{\partial}{\partial s} d^2$ and $\frac{\partial}{\partial t} d^2$ to 0, obtaining the corresponding parameters

$$s = \frac{BE - D}{1 - B^2}, \quad t = \frac{E - BD}{1 - B^2} \quad (7)$$

where

$$\begin{cases} D = \mathbf{m}_i \cdot (\mathbf{x}_i - \mathbf{x}_j) \\ E = \mathbf{m}_j \cdot (\mathbf{x}_i - \mathbf{x}_j) \\ B = \mathbf{m}_i \cdot \mathbf{m}_j \end{cases} \quad (8)$$

and thus the Gaussian Surfel is inserted with

$$\mathbf{x} = \frac{(\mathbf{x}_i + s \mathbf{m}_i) + (\mathbf{x}_j + t \mathbf{m}_j)}{2} \quad (9)$$

The vector $\mathbf{u}$ is defined as the direction from its mean $\mathbf{x}$ to the mean of one of the two original Gaussian Surfels. We also determine **normal** with the directions of two Surfels’ principal axes, and then compute the direction of $\mathbf{v}$ as the cross product of **normal** and $\mathbf{u}$. The values of s_u and s_v are determined based on distances from mean $\mathbf{x}$ of the inserted Surfel to the means of the other two, and we use a factor of 1.1 for these distances in our implementation. The remaining primitives (opacities and colors) of new Surfel are set as the average of those from two original Surfels.

4.2 Constructing SDF with Geometric Features

For Gaussian Surfels, optimized under photometric and geometric supervision and refined (denoised and filled with spatial features discussed in 4.1), we consider the Surfels to be dense

and accurately aligned with scene surfaces. Our method is inspired to directly construct an implicit function from explicit Surfels that closely adhere to the actual geometric structures.

Reconstruct a Gaussian Surfel in voxel space. First, we consider a simple idea of constructing facet representation consistent with a single Gaussian Surfel within a voxel. Signed distance d_i from vertex i to the Surfel, along the Surfel normal’s direction, is used for the vertex’s function value:

$$SDF(i) = d_i = (\mathbf{X}_i - \mathbf{x}) \cdot \mathbf{normal} \quad (10)$$

where $\mathbf{X}_i$ denotes the coordinates of the voxel vertex, while $\mathbf{x}$, $\mathbf{normal}$ are the corresponding primitives of Gaussian Surfel. By constructing SDF within the voxel, vertex positions can then be linearly interpolated along the edges, yielding mesh facets, as shown in Figure 4.

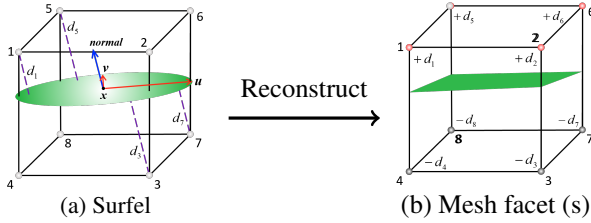


Figure 4. Simplified illustration of meta-reconstruction, showing signed distance computation from a Gaussian Surfel within a voxel.

It seems that the definition of SDF gives the signed distance to the infinite supporting plane of that Surfel, which can deviate from the true Euclidean distance to the bounded Surfel disk near its edges. Since Gaussian Surfels are densely distributed throughout the scene, these edge-proximity artifacts can subsequently be mitigated by fusing the SDFs from multiple Gaussians, thereby yielding a reliable approximation of the Euclidean distance.

Integration for a global SDF representation. We then consider the integration of all Surfels and extend the method to global SDF. For each Gaussian Surfel, we consider nearby vertices when constructing SDF. As it may span multiple voxels, we employ the following strategy to search for vertices to be integrated:

$$(i, j, k) \in \mathcal{N}((i_m, j_m, k_m), \epsilon), \quad \epsilon = \left\lceil \frac{\max(s_u, s_v)}{\text{VoxelSize}} \right\rceil \quad (11)$$

where i_m, j_m, k_m denote the indices of the vertex closest to the mean $\mathbf{x}$ of Gaussian Surfel. For vertices with indices i, j, k , the corresponding implicit function value is computed using Equation 10.

To avoid noise around complex corners, a weighted integration is adopted. Considering the confidence variance of different Surfels and encouraging the reconstruction of Surfels with higher confidence, for each Surfel, the SDF construction for vertex i also includes an associated weight conditioned on Surfel’s opacity:

$$W(i) = e^\alpha - 1 \quad (12)$$

Compared to a simple linear function with α , we believe that this exponential formulation assigns higher integration confidence to

high-opacity Surfels in accumulation. The integration process for each Surfel is the same as that used in TSDF:

$$SDF^{(t+1)}(i) = \frac{W^{(t)}(i) SDF^{(t)}(i) + W(i) SDF(i)}{W^{(t)}(i) + W(i)} \quad (13)$$

$$W^{(t+1)}(i) = W^{(t)}(i) + W(i) \quad (14)$$

Surfels with higher opacity convey greater geometric confidence, making surrounding SDF more likely to conform to their shapes. This is also consistent with the rendering process of 2DGS, where Surfels with higher opacity already carry greater weight during rendering-based optimization. The effectiveness of this approach is further validated by subsequent ablation studies and statistical results. In addition, by setting appropriate thresholds, vertices in noisy regions with few integration contributions can be easily identified and removed, which we will discuss in the rest of this section.

For vertex colors representing the textural appearance, we adopt the same integration like the SDF. Vertex colors are integrated from the SH-mapped values of Gaussian Surfels.

Integration culling of vertices. To avoid non-existent local structures, we perform culling of integrated vertices both along the plane of the Surfel (Projection Culling) and along its normal (Distance Culling). As illustrated in Figure 5, the integration culling is highly coupled with the SDF integration process.

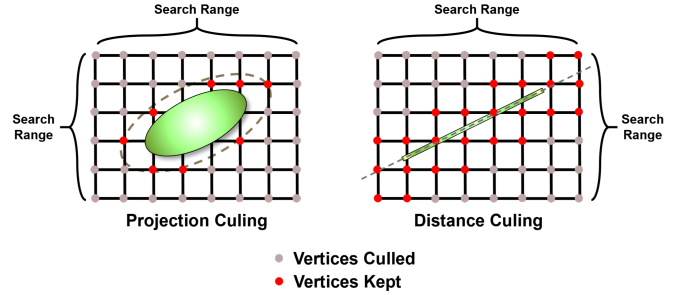


Figure 5. The culling of vertices during Gaussian Surfels integration. **Left:** Vertices in Projection Culling. **Right:** Vertices in Distance Culling.

On the Surfel, Projection Culling is performed based on the Mahalanobis distance ρ of each vertex projected onto Surfel plane:

$$\rho^2 = \left(\frac{(\mathbf{X}_i - \mathbf{x}) \cdot \mathbf{u}}{s_u} \right)^2 + \left(\frac{(\mathbf{X}_i - \mathbf{x}) \cdot \mathbf{v}}{s_v} \right)^2 \quad (15)$$

Specifically, only vertices with bounded Mahalanobis distances are retained, preventing Surfels from being reconstructed excessively large. The corresponding threshold of the Projection Culling ϵ_p will be presented in Section 5.

Along the normal direction, only vertices of the voxel containing the Surfel plane are kept, only constructing the SDF field exclusively in the Surfel-overlapping regions, based on their distance to the Gaussian surface.

Weight culling of vertices. To minimize the influence of noisy Surfels, we perform vertex culling based on a weight threshold ϵ_w coupled with the iso-surface extraction, as mentioned in SDF weight integration. Vertices with weights below the threshold

are removed during Marching Cubes, and the voxels containing those removed vertices are excluded from the subsequent extraction.

5 RESULTS

The implementation details and experiment results are presented below, demonstrating that our extraction method surpasses mainstream GS-based methods in both visual quality and quantitative metrics, on our presented novel problem aimed at reconstruction from Gaussian Surfels. The experiments are conducted across several reconstruction datasets, particularly including terrestrial object reconstruction datasets and photogrammetry datasets.

5.1 Experimental Setups

Implementations. In our implementation of Gaussian Surfels Splatting training, to obtain the rendering-trained Gaussian Surfels as we assume, we follow the strategy of CityGaussianV2-Coarse [27], including photometric supervision $\mathcal{L}_{\text{RGB}}$, monocular depth supervision $\mathcal{L}_{\text{Depth}}$ from Depth Anything V2 [68], regularization terms for Depth Distortion $\mathcal{L}_d$ and Depth-Normal Consistency $\mathcal{L}_n$. We also follow its Elongation Filter and DGD Strategies. The trainings are conducted on a single NVIDIA RTX 4090 GPU. Considering the over-densification and the associated risk of out of CUDA memory (OOM) in the late stage of training, we train for 20,000 steps in each scene to save for time and CUDA memory, concurrently obtaining Gaussian Surfels with insufficient geometric supervisions.

For a 3D refiner, we first perform geometric threshold-based denoising via denoiser with the thresholds $\epsilon_s=0.3$, $\epsilon_{dn}=0.5$. The selection of these two parameters is guided by scene-scale heuristics and was refined through a series of tuning trials. We observed that Gaussian Surfels optimized via multi-view consistency generally cover scene details with sufficient density. Under both the real-world scale of photogrammetric data and the relative scale estimated by COLMAP [69], individual Surfels whose scale exceeds the preset threshold are typically too coarse to represent fine details and can therefore be treated as noise.

When subsequently processed via filler for identifying planar regions, we consider each Surfel’s $k=15$ nearest neighbors balancing both efficiency and accuracy, with the geometric thresholds set as $\epsilon_d=0.7$, $\epsilon_p=0.05$, $\epsilon_n=0.95$. The factor used to determine whether filling is needed is set to $n=1.5$. For the selection of the thresholds ϵ_p and ϵ_n , following the smoothness-constrained region growing framework [70], we jointly enforce normal consistency (ϵ_n) and coplanarity (ϵ_p) as dual criteria for identifying planar Surfel regions, where the former uses the angular deviation between normals and the latter measures the perpendicular deviation from the seed plane. The corresponding parameter usage and selections are both expressed in a dot product form as illustrated in prior work [71]. The threshold ϵ_d and factor n are also determined empirically. However, factor n is not associated with scene scales and carries a clear geometric intuition. It controls the permissible gap relative to the combined extent of two adjacent Surfels. Setting n too low (such as 1.0) would require exact adjacency, risking incomplete filling of small holes when Surfels are obliquely oriented relative to one another, while setting a too large n (such as 2.0) risks bridging across genuine surface boundaries or incurring unnecessary computation. n

$= 1.5$ provides a practical trade-off between these two failure modes.

For SDF modeling, when performing Projection Culling, the Mahalanobis distance threshold is set to $\epsilon_p=3$, and the weight culling threshold is set to $\epsilon_w=1.05$. The threshold ϵ_p comes from the rendering principle of Gaussian Surfels [26, 27], where a Surfel is evaluated up to 3σ from its center during rasterization, beyond which its contribution becomes negligible. We therefore adopt $\epsilon_p=3$ times the Surfel’s scale as its effective spatial extent for the purpose of Projection Culling. ϵ_w is empirically chosen and cross-validated with per-scene average weight statistics, as discussed in Section 5.6.

For the empirical hyperparameters mentioned above, the sensitivity analysis and the statistically grounded ablation study are described in Appendix C and Appendix D, respectively.

During the search for nearby vertices of Gaussian Surfels in our method, we ensure the efficiency not only by controlling the search ranges and voxel size, but also by employing loop-level parallelism via compiler directives on multi-core CPU architectures. Compared with per-element sequential search, this strategy achieves faster near-linear scaling with respect to the number of vertices.

In addition, during the voxel SDF construction, we adopt a sparse voxel strategy wherein the voxels of the entire scene are partitioned into multiple blocks. Memory for a given block is allocated only when a voxel vertex is retrieved by a Gaussian search, thereby reducing the memory footprint of scene reconstruction and accelerating the subsequent iso-surface extraction.

During mesh accuracy evaluation, to reduce the impact of noisy initial points from unoptimized regions, we obtain the final mesh via a visibility-based visual projection. Specifically, the mesh is projected into viewpoints used for rendering-based training, and only the visible parts are evaluated as our results.

Baselines. Our work focuses on using Gaussian Surfels as input to obtain scene representation and surface reconstruction. Thus we compare our method with mainstream pipelines addressing the same problem as baselines: Gaussian Surfels + TSDF (the perspective used by state-of-the-art reconstruction methods such as ULSR-GS [72], 2DGS, GauS [23], etc.), Gaussian Surfels + Poisson reconstruction (the perspective mentioned in [66]), and a Delaunay-based α -Shape method directly utilizing Surfel means. Baselines also include another Gaussian ellipsoid-based SOTA methods, GOF [29] and PGSR [25], to assess the effectiveness of Gaussian Surfel representations we choose. A neural volumetric method, NeuS [60], is also chosen as an NVS-based baseline. We use the same voxel size in same scenes both for TSDF and our method, but different in different scenes due to memory limitations. For the majority of our general experiments, considering the balance between computational cost and granularity, we adopted a voxel size of 0.2 (corresponding to 0.2 m in real-world scale) for absolute-scale scenes in the UAV dataset, and a voxel size of 0.03 for scenes reconstructed under relative scale via COLMAP [69] in other datasets. We directly use the means and normals of Surfels for Poisson method.

TSDF-based method is considered as our primary baseline, since it shares a conceptually opposite paradigm to ours: TSDF constructs an implicit function via rendering from Gaussian Surfels,

while ours constructs the SDF directly from Surfels’ spatial features geometrically.

Datasets. The real-scene reconstruction datasets we use include four open-source datasets and our own UAV photogrammetry dataset. The DTU dataset [73] covers a wide range of indoor objects under seven different lighting conditions with 124 scenes in total, with the MVS reconstruction results as the ground truth of each scene. Tanks & Temples [74] is a multi-scene reconstruction dataset, with its surrounding-view features introducing occlusions and challenging outdoor illumination, providing LiDAR ground truth geometries in seven scenes. Deep Blending [75] is a multi-scene dataset for multi-view rendering, also used as a reconstruction benchmark, and MVS reconstruction results are provided as surface ground truth. We choose its indoor scenes (5 in total) to assess our method in complex indoor lighting features. BlendedMVS [76] is a Multi-View Stereo dataset spanning diverse application scenarios, including scenes derived from photogrammetry, providing MVS reconstruction results as ground truth meshes. We select three of its photogrammetry scenes for evaluating our methods under photogrammetry tasks. Furthermore, a vertical photogrammetry dataset from UAV is used for evaluation to assess our method in photogrammetry tasks. The dataset is constructed with images from vertical photogrammetry as inputs and Poisson-reconstructed LiDAR meshes as ground truth.

In the following, we highlight how our pipeline leads to improved reconstruction results.

5.2 Details Evaluation Between TSDF and Our Method

To prove our more robust perspective with spatial awareness of Gaussian Surfels and display qualitative results in detail, we first compare our method with the primary baseline on Drjohnson Scene from Deep Blending, as well as a building scene from our UAV photogrammetry dataset, which we refer it as UAV dataset for short. As shown in Figures 6 and 7, our method produces mesh with clearer texture (the carpet of Drjohnson), more realistic geometry (the chair and floor in Drjohnson), more accurate corner structures (occlusions and over-smoothing happen when using TSDF), and more complete scene components (the tree in UAV dataset).

5.3 Evaluation on Terrestrial Reconstruction Dataset

Our results on an open-source indoor object reconstruction dataset, DTU Dataset [73] and Tanks & Temples dataset [74], are displayed both qualitatively and quantitatively compared with all our baselines. Figure 8 presents a comparison of reconstruction quality. We not only display our geometric mesh and rendered mesh, along with results from our baselines, but also additionally include the corresponding real images of mesh regions in the first column, for the reconstructed ground truth may not stand for the most accurate representation.

Our method achieves significantly higher geometric quality and less surface distortions on DTU and Tanks & Temples dataset than other baselines. Compared to rendering-based methods, TSDF (depth rendering-based) and GOF (opacity rendering-based), our method avoids inaccuracies from unstable depth and blurred iso-surfaces caused by multi-view inconsistencies. Compared with the Poisson and α -Shape method, ours is more robust



Figure 6. Examples of comparisons between TSDF and GSSA in Drjohnson Scene of Deep Blending. Red boxes (or orange boxes, along with arrows of the corresponding color) indicate detail regions of the reconstructed surface.

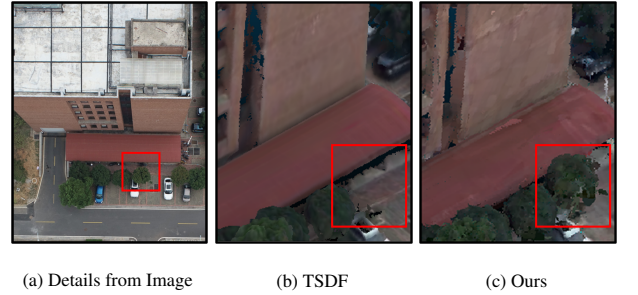


Figure 7. Examples of comparisons between TSDF and GSSA in UAV dataset. Red boxes indicate detail regions of the reconstructed surface.

to noisy, floating Gaussians. GOF, using Gaussian ellipsoids, struggles to consistently reconstruct a surface across views.

It is worth noting that GOF also performs well on the Tanks and Temples dataset, likely because the dense viewpoints in this dataset allow GOF to better capture fine details in outdoor, close-range scenes. Thus this mitigates the aforementioned surface distortion.

The reconstruction accuracy is reported in Table 2 and Table 3. We computed the average Precision and Recall across 50 different representative scenes in seven different lighting conditions of DTU Dataset, as well as seven scenes with the ground-truth scanned point cloud of Tanks & Temples dataset for each method, and then calculated the F1 Score based on these averages. We also computed the average Chamfer distance, Mean Normal Error, and Hausdorff Distance metrics for a more comprehensive evaluation. Across the overall metrics, our method outperforms all baselines in terms of geometric accuracy in these datasets, reconstructing more accurate surfaces in complex, variant lighting conditions. To illustrate the model’s performance distribution, we computed the F1 Scores from our experimental results on the DTU and Tanks & Temples datasets, visualizing them using box plots, as shown in Figure 9.

We additionally tested for the average time consumptions of different approaches under five scenes in DTU dataset, including

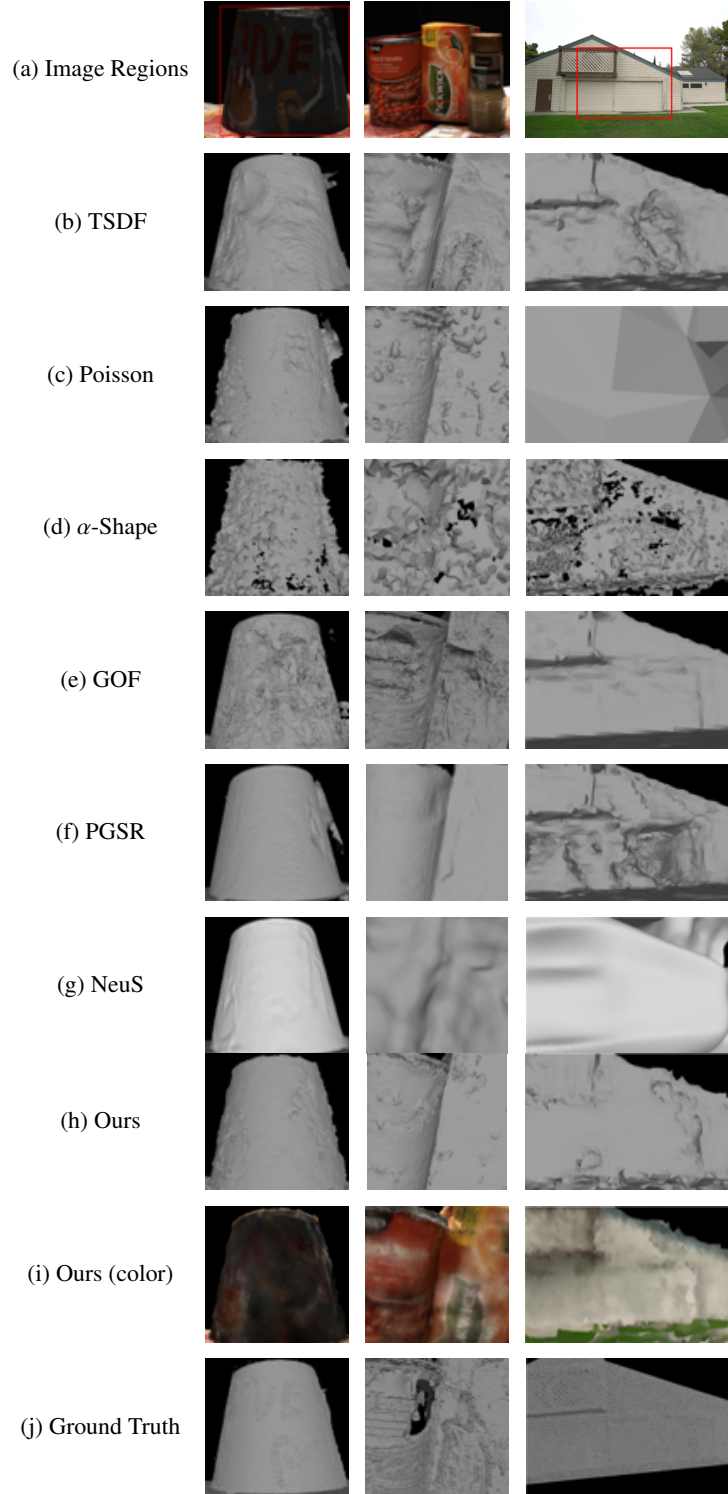


Figure 8. The qualitative results of GSSA and baselines in DTU dataset and Tanks & Temples dataset. Red boxes mark the reconstructed regions overlaid on the real image regions.

the training and surface extraction stages displayed in Table 4. During training, we trained the CityGaussianV2-Coarse framework for 20,000 iterations to obtain the trained Gaussian Surfels for TSDF, Poisson, α -Shape and our method (thus sharing the same training time), the GOF framework for 20,000 iterations,

the PGSR framework for 20,000 iterations, and the NeuS framework for 100,000 iterations.

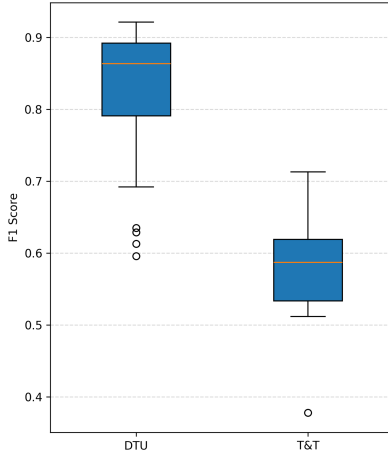
The total time consumption and accuracy are jointly displayed in Figure 10. Our approach achieves the best geometric accuracy among baseline methods with comparable computational

Table 2. Quantitative accuracy evaluation results in DTU dataset between GSSA and baselines.

Method	Precision (%) $\uparrow$	Recall (%) $\uparrow$	F1 Score (%) $\uparrow$	Chamfer Distance $\downarrow$	Mean Normal ($^\circ$) $\downarrow$	Hausdorff Distance $\downarrow$
TSDF	53.0	89.4	66.5	0.0021	77.0	0.1205
Poisson	37.9	94.0	54.1	0.0027	78.1	0.2058
α -Shape	63.2	98.1	76.8	0.0012	78.5	0.3701
GOF	69.0	95.6	80.2	0.0008	77.4	0.1288
PGSR	72.6	95.1	82.3	0.0006	76.6	0.1171
NeuS	65.2	69.7	67.4	0.0016	77.9	0.1915
Ours	74.7	97.2	84.5	0.0006	77.1	0.1099

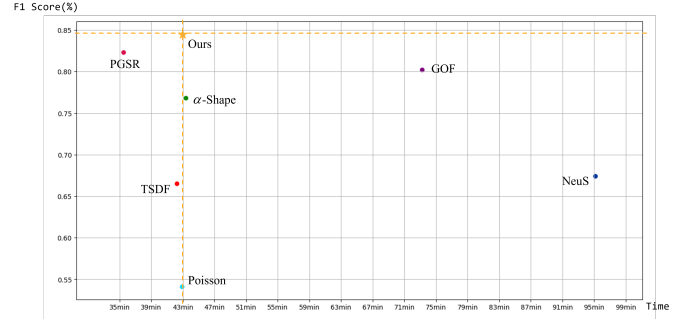
Table 3. Quantitative accuracy evaluation results in Tanks & Temples dataset between GSSA and baselines.

Method	Precision (%) $\uparrow$	Recall (%) $\uparrow$	F1 Score (%) $\uparrow$	Chamfer Distance $\downarrow$	Mean Normal ($^\circ$) $\downarrow$	Hausdorff Distance $\downarrow$
TSDF	30.3	66.9	41.7	0.0236	72.1	0.3812
Poisson	13.2	45.2	20.4	0.0420	73.6	0.6176
α -Shape	47.2	74.2	57.7	0.0227	73.0	0.6025
GOF	45.9	71.5	55.9	0.0226	69.7	0.3682
PGSR	43.5	82.1	56.9	0.0176	68.0	0.1469
NeuS	40.1	55.9	46.7	0.0124	72.3	0.1968
Ours	43.4	86.9	57.9	0.0113	69.8	0.1154

**Figure 9.** Box plots of results in DTU and Tanks & Temples datasets.**Table 4.** Average time consumption of training and surface extraction compared between different methods. The most time-consuming one is in the cell with gray background.

Method	Training	Surface Extraction	Total
TSDF	41 min 41 s	35 s	42 min 16 s
Poisson	41 min 41 s	1 min 14 s	42 min 55 s
α -Shape	41 min 41 s	1 min 43 s	43 min 24 s
GOF	69 min 54 s	3 min 22 s	73 min 16 s
PGSR	34 min 39 s	52 s	35 min 31 s
NeuS	89 min 59 s	5 min 11 s	95 min 10 s
Ours	41 min 41 s	1 min 17 s	42 min 58 s

efficiency. Among all methods, GOF and NeuS consume much more time than other pipelines both in training and surface extraction. This indicates that GOF spends lots of time in opacity-based level-set detecting for SDF and the complex Marching Tetrahedra algorithm and NeuS, following the neural volumetric

**Figure 10.** The time consumption and accuracy of different baselines and GSSA.

rendering paradigm, spends considerable time on dense voxel sampling and neural inference to achieve reasonably competitive reconstruction quality. In some cases, the Gaussian ellipsoid-based representation and rendering-based SDF construction of GOF result in low reconstruction accuracy at the same time, demonstrating the advantage of our Surfel-based, spatial-aware reconstruction method. As shown in Figure 10, our method reaches a balance between efficiency and accuracy in surface reconstruction, owing to the fast depth rendering during training and spatial awareness during SDF modeling.

5.4 Evaluation on Airborne Photogrammetry Datasets

To display the advantages of our method in photogrammetry reconstruction, we present the results in BlendedMVS dataset and our own UAV vertical photogrammetry dataset. The qualitative comparison is shown in Figure 11. When applied to photogrammetric datasets, PGSR produced severely degraded surface reconstructions, exhibiting dense artifacts and repetitive structures across the entire scene. The details can be found in Appendix A. Accordingly, it is excluded from the photogrammetric evaluation on UAV and BlendedMVS datasets, while NeuS remained.

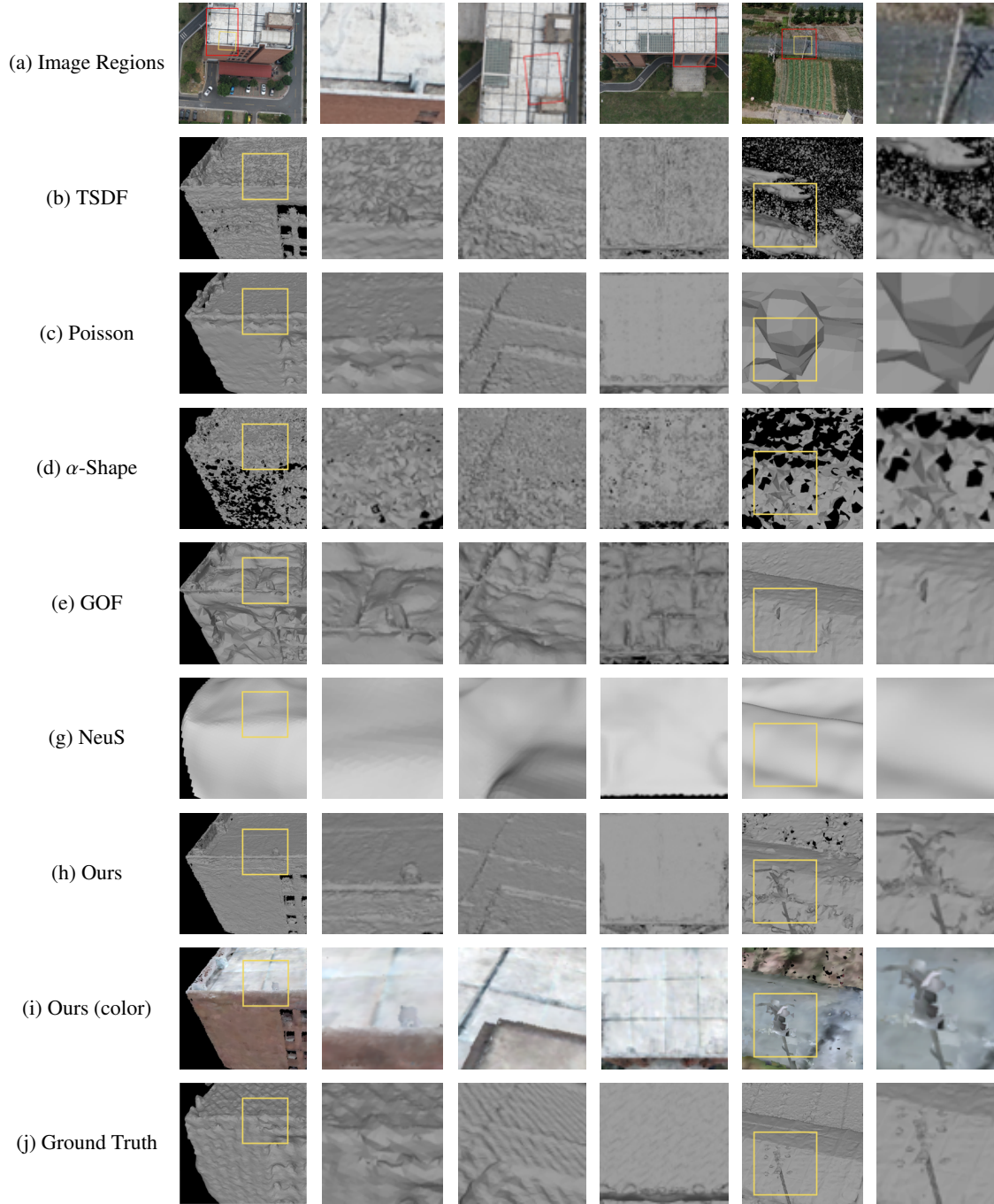


Figure 11. The qualitative results compared between GSSA and baselines in UAV dataset and BlendedMVS dataset. Yellow boxes indicate detail regions of the reconstructed surfaces.

In the qualitative results, our method still displays strong advantages in extracting surfaces of building geometries with large-scale structures, showing its potential for photogrammetry applications. Compared to rendering-based methods (TSDF and GOF), which produces numerous geometric distortions on the building’s top surface, our method, along with various spatial-aware methods (e.g., Poisson), demonstrates more complete and smoother reconstruction results.

As shown in the 4th row of Figure 11, our method is more robust in dealing with vertical photogrammetry data. Rendering-based TSDF fails to reconstruct the column structures beneath the building top due to rendering distortions and occlusions in training images vertically taken from UAV, and even results from Poisson and the ground truth cannot accurately represent the cavities and column structures shown in Image Regions of the first column. In contrast, our method is spatially aware of these geometries which often appear in photogrammetry tasks.

Table 5. Quantitative accuracy evaluation results in UAV dataset between GSSA and baselines.

Method	Precision (%) $\uparrow$	Recall (%) $\uparrow$	F1 Score (%) $\uparrow$	Chamfer Distance $\downarrow$	Mean Normal ($^\circ$) $\downarrow$	Hausdorff Distance $\downarrow$
TSDF	91.6	92.8	92.2	0.0189	48.1	0.3274
Poisson	80.4	91.8	85.7	0.0198	39.6	0.3269
α-Shape	87.1	83.9	85.5	0.0180	48.5	0.3251
GOF	81.0	89.8	85.1	0.0199	48.8	0.3272
NeuS	81.7	77.6	79.6	0.0202	43.0	0.3149
Ours	93.4	92.8	93.1	0.0179	34.8	0.3105

Table 6. Quantitative accuracy evaluation results in BlendedMVS dataset between GSSA and baselines.

Method	Precision (%) $\uparrow$	Recall (%) $\uparrow$	F1 Score (%) $\uparrow$	Chamfer Distance $\downarrow$	Mean Normal ($^\circ$) $\downarrow$	Hausdorff Distance $\downarrow$
TSDF	34.4	75.6	47.3	0.0057	53.4	0.4983
Poisson	49.9	82.4	62.2	0.0051	49.4	0.7649
α-Shape	93.6	68.1	78.8	0.0011	50.8	0.2435
GOF	95.1	68.6	79.7	0.0008	43.2	0.1694
NeuS	71.9	70.1	71.0	0.0018	46.0	0.2104
Ours	88.6	77.2	82.5	0.0010	47.3	0.1683

The BlendedMVS dataset provides greater reconstruction challenges with sparser aerial viewpoints and lower images resolution. As shown in the 5th row of Figure 11, Gaussian Surfels from our CityGaussianV2-coarse training backbones are sparsely distributed, leading to severe fragmentation in the TSDF and α -Shape baselines. The results of Poisson and GOF baselines indicate smoother reconstructions of both off-ground objects and terrains, however both are suffering from missing fine details such as utility poles. Although our method retains better geometric completeness for man-made off-ground objects (the greenhouse and utility pole), missing surface regions still appear on the ground due to insufficient training densification, which is an issue we will explore further in future work.

Then, we display the quantitative evaluation results in the UAV dataset, as shown in Table 5, and the results in the BlendedMVS dataset, as shown in Table 6.

In general, both our method and TSDF produces a relatively high accuracy mesh for vertical photogrammetry. Although our method obtains the highest scores in all metrics of the UAV dataset in Table 5, considering the missing data and errors caused by occlusions during LiDAR data collection, we believe the quantitative results can only serve as a partial reference. The quality of reconstruction still depends a lot on qualitative comparisons with real images.

In Table 6, although our method does not achieve the highest Precision or Recall individually, it yields the best overall F1 Score, reflecting an optimal balance between reconstruction accuracy and structural completeness in photogrammetric tasks, which is consistent with the qualitative observations reported above.

5.5 Migration Analysis

Since our method tends to reflect the distribution of Surfels, it can serve as a plug-and-play module for frameworks based on Gaussian Surfel representations and obtain surface representations. In turn, it can also be used to analyze the plausibility of Surfel distributions as well as the validity of training strategies. GSSA is migrated to perform reconstruction using Gaussian Sur-

fels optimized with the original 2DGS training strategies. Since the scene scale after different Gaussian training procedures can be regarded as consistent, we adopt the same hyperparameters for GSSA extracting surfaces from 2DGS-based Surfels as the default settings of the CityGaussianV2 framework, including voxel size, culling thresholds, and related parameters. The same applies to the TSDF method for parameters such as voxel size and SDF truncation distance. For the 2DGS framework, we train for 20,000 iterations. The results are provided in Figure 12 and Table 7. Here, CityGaussianV2 + GSSA and CityGaussianV2 + TSDF still use the Surfels trained as mentioned above, while 2DGS + GSSA and 2DGS + TSDF use the Surfels trained from the original 2DGS. We display the results of two scenes, Caterpillar from T & T and CreepyAttic from Deep Blending.

Our primary baseline, TSDF, produces meshes with much more blurred structural details (in Caterpillar) and more distortions in object geometries (in CreepyAttic). In the Caterpillar scene, due to lack of depth constraints and insufficient densification of 2DGS, some surfaces exhibit geometric deviations. In the CreepyAttic scene, GSSA produces more complete wall surfaces using 2DGS. 2DGS training strategies can be more suitable for indoor scenes with complex artificial lighting conditions. Compared to TSDF on 2DGS training framework, our method lags in rendering quality in this scenario. Specifically, under unoccluded viewpoints, rendering-based reconstruction methods can achieve superior visual quality and even accuracy, which is a gap that our method must address through further trade-off optimization. This also indicates that our method does not uniformly improve reconstruction quality over the TSDF baseline.

5.6 Ablation Studies

We perform ablation studies by adding certain modules from the base model of our method: the 3D refiner and culling strategies. To further validate the 3D refiner’s internal contributions, we decompose it into two submodules, the denoiser and the filler, and add them separately. This granular decomposition reveals both the specific role of each submodule and their interaction. The accuracy is evaluated on a subset of the scene from our UAV dataset, with results shown in Table 8 and Figure 13.

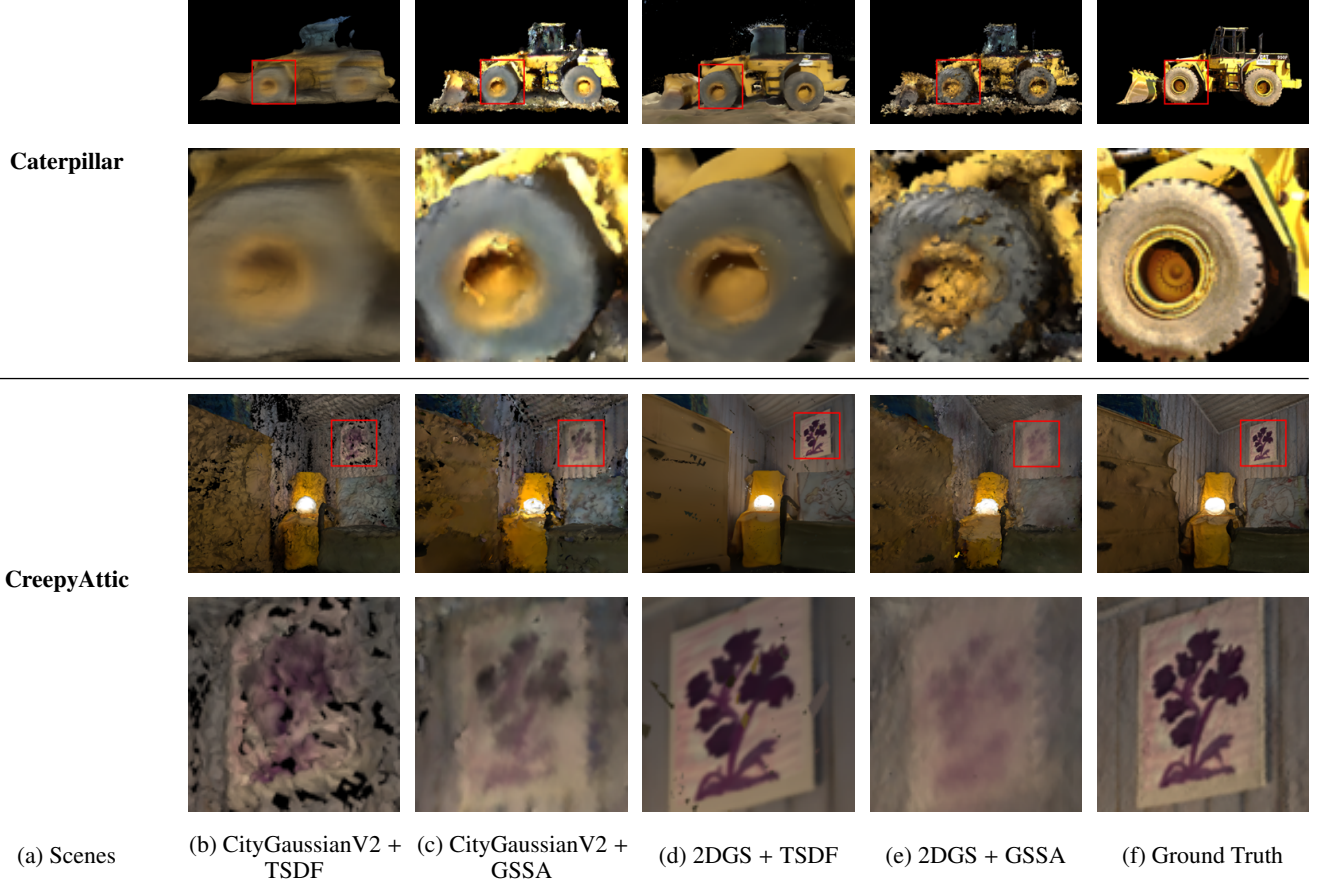


Figure 12. The rendered reconstruction results compared between TSDF and GSSA used on different Gaussian Surfels. Red boxes in the first and third columns indicate the detail regions shown in the second and fourth columns, respectively.

Table 7. The quantitative results between TSDF and GSSA used on different Gaussian Surfels. (Unit: %).

Method	Caterpillar			CreepyAttic		
	Precision ↑	Recall ↑	F1 Score ↑	Precision ↑	Recall ↑	F1 Score ↑
CityGaussianV2 + TSDF	53.8	76.5	63.2	82.1	19.5	31.5
CityGaussianV2 + GSSA	73.0	99.2	84.1	85.4	20.0	32.4
2DGS + TSDF	60.8	87.3	71.6	95.5	22.3	36.2
2DGS + GSSA	65.2	99.2	78.7	97.7	20.2	33.5

Table 8. Quantitative results of ablation comparisons. (Unit: %).

Setup	Precision ↑	Recall ↑	F1 Score ↑
Base model	73.4	93.2	82.1
w/ culling	84.9	93.1	88.8
w/ culling + w/ Denoiser	88.9	92.5	90.7
w/ culling + w/ Filler	86.3	93.1	89.6
w/ refiner	86.7	93.2	89.8
Full pipeline	89.0	93.0	91.0

In our quantitative results, although Recall does not change obviously for the general structures are still maintained, Precision demonstrates that both our refiner and culling modules enhance robustness against tiny holes and noisy points. In addition, both adding the refiner and culling modules leads to a significant

increase in metrics, demonstrating their complementary roles in ensuring high-quality reconstruction.

To demonstrate the rationality of our choice for an exponential weight function for integration, we also reconstruct meshes under other alternative functions, such as the linear function

$$W(i) = \alpha \quad (16)$$

and the constant function

$$W(i) = 1 \quad (17)$$

We demonstrate a simple illustration of these alternative weight functions in (a) of Figure 14, and display the frequency histograms of all vertex weights under different weight functions in (b), (c), and (d).

The weights in Figure 14 are counted before weight culling, reflecting all vertex weights from integration. Since the final

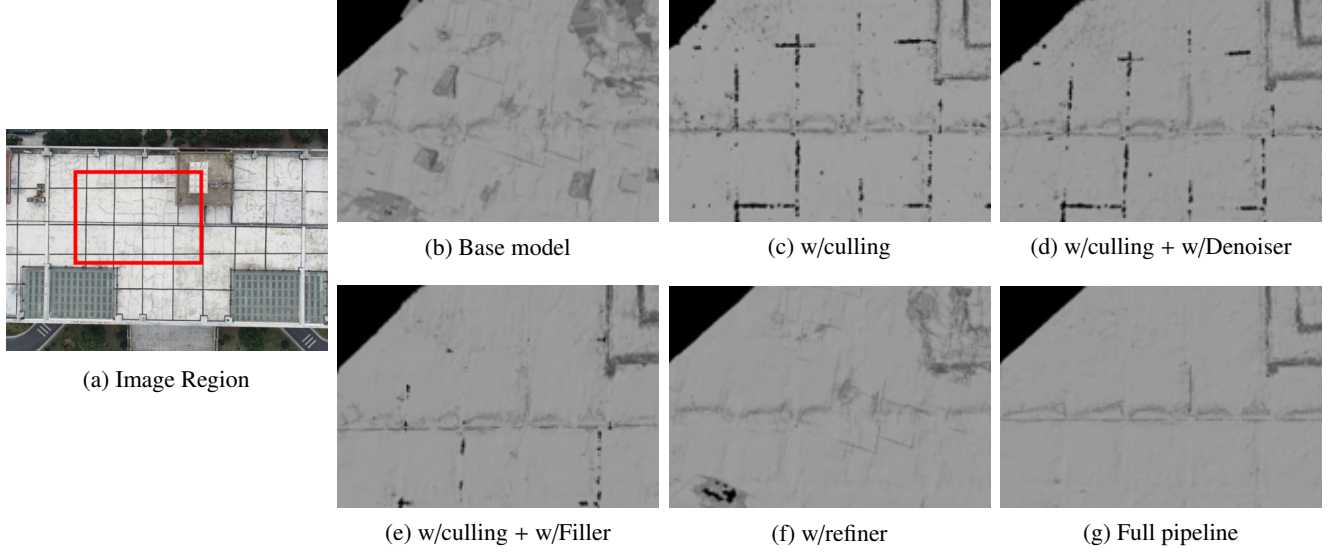


Figure 13. Qualitative results of ablation comparisons. The red box in the image region indicates the areas where the reconstructed surface is displayed.

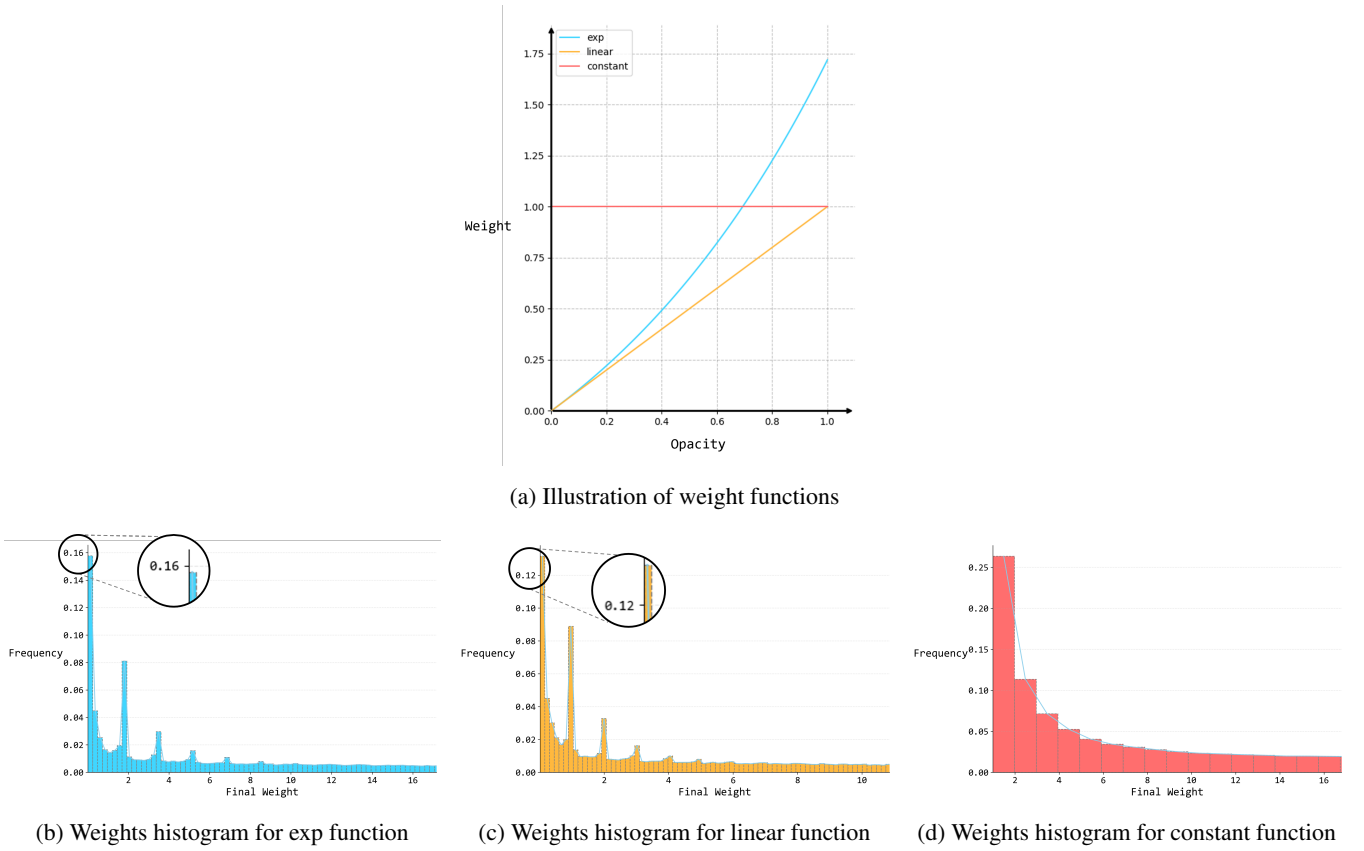


Figure 14. Different weight functions and the frequency of vertex weights under different weight functions.

weights under the constant function are accumulated to be discrete integers, we plot this histogram, in (d), at a coarser bin resolution. The weights from the exp function exhibit more pronounced differences between the first two peaks, with a larger proportion at lower values compared to the linear function. This is due to the exp function’s steepening slope, which amplifies

high-confidence weights. In our experiments, thresholding between these two peaks of the exp function effectively separates the vertex distributions on either side, yielding geometrically cleaner reconstructions with reduced noise. This can be considered as effectively separating noisy vertices from correct ones, as expected.

As we have chosen a threshold ϵ_w for weight culling in the exp function, we calculate the percentage of this threshold in weights for the exp function, and then use the corresponding percentile for other functions for a fair comparison. The results are shown in Table 9. In general, the exp weight function proves to be the best setup in these alternatives.

Table 9. Quantitative results of different weight functions. (Unit: %).

Weight Function	Precision $\uparrow$	Recall $\uparrow$	F1 Score $\uparrow$
Exp function	89.0	93.0	91.0
Linear function	86.8	93.1	89.8
Constant function	81.4	93.1	86.9

6 DISCUSSION

6.1 Potentialities

To the best of our knowledge, our work is the first to highlight the crucial role of the geometric features of Surfels for explicit utilization in reconstruction, while achieving superior geometric performance compared to mainstream reconstruction perspectives at the same time. In addition, our sparse-voxel strategies bring engineering feasibility to our method. On the UAV dataset, our method achieves a peak memory footprint of only 7.07 GB, which is even lower than that of the more mature TSDF baseline (8.48 GB). The method provides a new paradigm for obtaining surfaces more consistent with scenes, more complete and less distortions than previous rendering-based paradigm. The insight has a promising potentiality for large-scale applications in photogrammetry, SLAM, terrain modeling and other reconstruction tasks, leveraging both the fast optimization of Gaussian Splatting and high accuracy of spatial awareness.

6.2 Limitations

While our method achieves state-of-the-art reconstruction across diverse scenes, limitations still remain, which will also be part of our future work. First, since Gaussian Splatting assumes objects with full opacities, distortions may occur when dealing with semi-transparent objects. This remains an open problem and shared limitation in the field of 3DGS. A potential mitigation is to incorporate a transparency-aware rendering loss or to integrate monocular depth priors that are unaffected by surface opacity, providing a more reliable geometric signal for transparent regions. Second, our method is just built upon Surfels, while our idea of spatial awareness also provides insights for future extensions to general Gaussian ellipsoids for a more universal reconstruction framework. We regard the adaptation to 3DGS as a meaningful future work, though the Surfel representation is the natural choice for surface-centric reconstruction. Finally, compared to the mature TSDF, Poisson or other reconstruction implementations from 3D libraries, our method, bridging Surfels and SDF representations, still faces efficiency limitations. For wider application, practical acceleration strategies are needed, such as GPU-accelerated SDF integration via CUDA-based voxel hashing [43] and GPU Marching Cubes, which would reduce the SDF construction cost for large-scale scenes.

7 CONCLUSIONS

This paper defines a novel problem: constructing implicit function from rendering-based trained Gaussian Surfels with explicit geometric features. We present a method for this problem, GSSA, with spatial awareness and interpretability: spatially refining the Gaussian Surfels via a 3D refiner and directly constructing SDF from Surfels’ geometric features. This approach addresses the limitations of existing rendering-based training and surface representation pipelines. As displayed in the experiments, the method provides a new perspective on a reconstruction paradigm within Gaussian Splatting, achieving state-of-the-art geometric accuracy from trained Gaussian Surfels across various terrestrial and airborne reconstruction benchmarks, without struggling with rendered distortions brought by previous TSDF-based paradigms, displaying strong potential for object and terrain reconstruction applications in photogrammetry.

ACKNOWLEDGEMENTS

This research was funded by National Natural Science Foundation of China under Grant No. 42501565, Natural Science Foundation of Hunan Province, China under Grant No. 2026JJ60405, College Student Innovation and Entrepreneurship Training Program of Hunan Province, China under Grant No. S202510533396, Key Laboratory of China-ASEAN Satellite Remote Sensing Applications, Ministry of Natural Resources of the People’s Republic of China under Grant No. KLCARS-2025-G05, and the Open Research Fund of the Laboratory, Central South University.

REFERENCES

- [1] Xingwen Cao, Xueting Zheng, Hongwei Zheng, Xi Chen, Anming Bao, Ying Liu, Tie Liu, Haoran Zhang, Muhua Zhao, and Zichen Zhang. PesRec: A parametric estimation method for indoor semantic scene reconstruction from a single image. *International Journal of Applied Earth Observation and Geoinformation*, 133:104135, September 2024. ISSN 1569-8432. doi: 10.1016/j.jag.2024.104135.
- [2] Li Yan, Jiang Song, Hong Xie, Pengcheng Wei, Gang Li, Longze Zhu, Zhongli Fan, and Shucheng Gong. LiDGS: An efficient 3D reconstruction framework integrating lidar point clouds and multi-view images for enhanced geometric fidelity. *International Journal of Applied Earth Observation and Geoinformation*, 142:104730, August 2025. ISSN 1569-8432. doi: 10.1016/j.jag.2025.104730.
- [3] Shuhang Zhang, Chun Liu, and Norbert Haala. Guided by model quality: UAV path planning for complete and precise 3D reconstruction of complex buildings. *International Journal of Applied Earth Observation and Geoinformation*, 127:103667, March 2024. ISSN 1569-8432. doi: 10.1016/j.jag.2024.103667.
- [4] Jinghai Xu, Suyu Zhang, Haoran Jing, Craig Hancock, Peng Qiao, Nan Shen, and Karen B. Blay. Improving Real-Scene 3D Model Quality of Unmanned Aerial Vehicle Oblique-Photogrammetry with a Ground Camera. *Remote Sensing*, 16(21):3933, January 2024. ISSN 2072-4292. doi: 10.3390/rs16213933.
- [5] Chi Zhang, Yiming Yan, Chunhui Zhao, Nan Su, and Weikun Zhou. FVMD-ISRe: 3-D Reconstruction From

- Few-View Multidate Satellite Images Based on the Implicit Surface Representation of Neural Radiance Fields. *IEEE Transactions on Geoscience and Remote Sensing*, 62: 1–14, 2024. ISSN 1558-0644. doi: 10.1109/TGRS.2024.3399786.
- [6] Junsu Leem, Seyedahmad Mehrishal, Il-Seok Kang, Dong-Ho Yoon, Yulong Shao, Jae-Joon Song, and Jinha Jung. Optimizing Camera Settings and Unmanned Aerial Vehicle Flight Methods for Imagery-Based 3D Reconstruction: Applications in Outcrop and Underground Rock Faces. *Remote Sensing*, 17(11):1877, January 2025. ISSN 2072-4292. doi: 10.3390/rs17111877.
- [7] Mubai Li, Zhenxin Zhang, Siyun Chen, Liqiang Zhang, Zhihua Xu, Xiaoxu Ren, Jinlan Liu, and Peng Sun. ColorMesh: Surface and texture reconstruction of large-scale scenes from unstructured colorful point clouds with adaptive automatic viewpoint selection. *International Journal of Applied Earth Observation and Geoinformation*, 132:104041, August 2024. ISSN 1569-8432. doi: 10.1016/j.jag.2024.104041.
- [8] Cheng Wang, Haibing Liu, and Fei Deng. Reconstructing Geometric Models from the Point Clouds of Buildings Based on a Transformer Network. *Remote Sensing*, 17(3):359, January 2025. ISSN 2072-4292. doi: 10.3390/rs17030359.
- [9] Fulong Chen, Peifeng Ma, Siliang Chen, Qingwu Hu, and Huadong Guo. Remote sensing for cultural heritage: A systematic review. *International Journal of Applied Earth Observation and Geoinformation*, 146:105039, February 2026. ISSN 1569-8432. doi: 10.1016/j.jag.2025.105039.
- [10] Chaoyin Xin, Wende Xie, Yafei Wang, Zexing Li, Bowen Wang, and Hongyang Shi. GeRAD: Geometry-Aware Neural Implicit Surface Rendering for Autonomous Driving. In *2024 IEEE 27th International Conference on Intelligent Transportation Systems (ITSC)*, pages 57–63, September 2024. doi: 10.1109/ITSC58415.2024.10920192.
- [11] Christiane Sommer, Lu Sang, David Schubert, and Daniel Cremers. Gradient-SDF: A Semi-Implicit Surface Representation for 3D Reconstruction. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6270–6279, New Orleans, LA, USA, June 2022. IEEE. ISBN 978-1-6654-6946-3. doi: 10.1109/CVPR52688.2022.00618.
- [12] Shuang Song and Rongjun Qin. Mesh Conflation of Oblique Photogrammetric Models using Virtual Cameras and Truncated Signed Distance Field. *CoRR*, abs/2308.12139, 2023. doi: 10.48550/ARXIV.2308.12139.
- [13] Yunqiang Li, Shuowen Huang, Ying Chen, Yong Ding, Pengcheng Zhao, Qingwu Hu, and Xujie Zhang. RGBTSDf: An Efficient and Simple Method for Color Truncated Signed Distance Field (TSDF) Volume Fusion Based on RGB-D Images. *Remote Sensing*, 16(17):3188, January 2024. ISSN 2072-4292. doi: 10.3390/rs16173188.
- [14] Richard A. Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J. Davison, Pushmeet Kohli, Jamie Shotton, Steve Hodges, and Andrew W. Fitzgibbon. KinectFusion: Real-time dense surface mapping and tracking. In *10th IEEE International Symposium on Mixed and Augmented Reality, ISMAR 2011, Basel, Switzerland, October 26-29, 2011*, pages 127–136, 2011. doi: 10.1109/ISMAR.2011.6092378.
- [15] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 163–169, August 1987. doi: 10.1145/37401.37422. Conference Name: SIGGRAPH '87: Computer graphics and interactive techniques ISBN: 9780897912273.
- [16] Antoine Guédon, Diego Gomez, Nissim Maruani, Bingchen Gong, George Drettakis, and Maks Ovsjanikov. MIIo: Mesh-In-the-Loop Gaussian Splatting for Detailed and Efficient Surface Reconstruction. *ACM Trans. Graph.*, 44(6):223:1–223:15, December 2025. doi: 10.1145/3763339.
- [17] Graham M. Treece, Richard W. Prager, and Andrew H. Gee. Regularised marching tetrahedra: improved iso-surface extraction. *Comput. Graph.*, 23(4):583–598, 1999. doi: 10.1016/S0097-8493(99)00076-X.
- [18] Thomas Luhmann, Stuart Robson, Stephen Kyle, and Jan Boehm. *Close-Range Photogrammetry and 3D Imaging*. De Gruyter, 2020. ISBN 978-3-11-060725-3. doi: 10.1515/9783110607253.
- [19] Jin Liu, Jian Gao, Shunping Ji, Chang Zeng, Shaoyi Zhang, and JianYa Gong. Deep learning based multi-view stereo matching and 3D scene reconstruction from oblique aerial images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 204:42–60, October 2023. ISSN 0924-2716. doi: 10.1016/j.isprsjprs.2023.08.015.
- [20] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Trans. Graph.*, 42(4):139:1–139:14, August 2023. doi: 10.1145/3592433.
- [21] Antoine Guédon and Vincent Lepetit. SuGaR: Surface-Aligned Gaussian Splatting for Efficient 3D Mesh Reconstruction and High-Quality Mesh Rendering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 5354–5363, 2024. doi: 10.1109/CVPR52733.2024.00512.
- [22] Joanna Waczynska, Piotr Borycki, Sławomir Konrad Tadeja, Jacek Tabor, and Przemysław Spurek. GaMeS: Mesh-Based Adapting and Modification of Gaussian Splatting. *CoRR*, abs/2402.01459, 2024. doi: 10.48550/ARXIV.2402.01459.
- [23] Chongjie Ye, Yinyu Nie, Jiahao Chang, Yuntao Chen, Yihao Zhi, and Xiaoguang Han. GauStudio: A Modular Framework for 3D Gaussian Splatting and Beyond, March 2024. arXiv:2403.19632 [cs].
- [24] Yanqi Bao, Tianyu Ding, Jing Huo, Yaoli Liu, Yuxin Li, Wenbin Li, Yang Gao, and Jiebo Luo. 3D Gaussian Splatting: Survey, Technologies, Challenges, and Opportunities. *IEEE Trans. Circuits Syst. Video Technol.*, 35(7):6832–6852, July 2025. doi: 10.1109/TCSVT.2025.3538684.
- [25] 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 Trans. Vis. Comput. Graph.*, 31(9):6100–6111, September 2025. doi: 10.1109/TVCG.2024.3494046.
- [26] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2D Gaussian Splatting for Geometrically Accurate Radiance Fields. *CoRR*, abs/2403.17888, 2024. doi: 10.48550/ARXIV.2403.17888.
- [27] Yang Liu, Chuanchen Luo, Zhongkai Mao, Junran Peng, and Zhaoxiang Zhang. CityGaussianV2: Efficient and Geometrically Accurate Reconstruction for Large-Scale Scenes. *CoRR*, abs/2411.00771, 2024. doi: 10.48550/ARXIV.2411.00771.
- [28] Chen Zhang and Yu Hu. CuFusion: Accurate Real-Time Camera Tracking and Volumetric Scene Reconstruction with a Cuboid. *Sensors*, 17(10):2260, 2017. doi: 10.3390/S17102260.
- [29] Zehao Yu, Torsten Sattler, and Andreas Geiger. Gaussian Opacity Fields: Efficient Adaptive Surface Reconstruction in Unbounded Scenes. *ACM Trans. Graph.*, 43(6):271:1–271:13, December 2024. doi: 10.1145/3687937.
- [30] Jiabo Xu, Zhili Zhang, Xiangyun Hu, and Tao Ke. Accelerated forest modeling from tree canopy point clouds via deep learning. *International Journal of Applied Earth Observation and Geoinformation*, 132:104074, August 2024. ISSN 1569-8432. doi: 10.1016/j.jag.2024.104074.
- [31] N. A. Golias and R. W. Dutton. Delaunay triangulation and 3D adaptive mesh generation. *Finite Elements in Analysis and Design*, 25(3):331–341, April 1997. ISSN 0168-874X. doi: 10.1016/S0168-874X(96)00054-6.
- [32] Herbert Edelsbrunner. Alpha shapes—a survey. In *Tessellations in the sciences: Virtues, techniques and applications of geometric tilings*. 2011.
- [33] P. Labatut, J.-P. Pons, and R. Keriven. Robust and Efficient Surface Reconstruction From Range Data. *Computer Graphics Forum*, 28(8):2275–2290, 2009. ISSN 1467-8659. doi: 10.1111/j.1467-8659.2009.01530.x.
- [34] Yasutaka Furukawa and Jean Ponce. Accurate, Dense, and Robust Multiview Stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(8):1362–1376, August 2010. ISSN 1939-3539. doi: 10.1109/TPAMI.2009.161.
- [35] Jiann-Yeou Rau and Bo-Cheng Lin. Automatic roof model reconstruction from ALS data and 2D ground plans based on side projection and the TMR algorithm. *ISPRS Journal of Photogrammetry and Remote Sensing*, 66(6, Supplement):S13–S27, December 2011. ISSN 0924-2716. doi: 10.1016/j.isprsjprs.2011.09.001.
- [36] Hoang-Hiep Vu, Patrick Labatut, Jean-Philippe Pons, and Renaud Keriven. High Accuracy and Visibility-Consistent Dense Multiview Stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(5):889–901, May 2012. ISSN 1939-3539. doi: 10.1109/TPAMI.2011.172.
- [37] Zhiqin Chen, Andrea Tagliasacchi, and Hao Zhang. BSP-Net: Generating Compact Meshes via Binary Space Partitioning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 42–51, 2020.
- [38] Nicholas Sharp and Maks Ovsjanikov. PointTriNet: Learned Triangulation of 3D Point Sets. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII*, pages 762–778, Berlin, Heidelberg, August 2020. Springer-Verlag. ISBN 978-3-030-58591-4. doi: 10.1007/978-3-030-58592-1_45.
- [39] Matthew Berger, Andrea Tagliasacchi, Lee M. Seversky, Pierre Alliez, Joshua A. Levine, Andrei Sharf, and Claudio T. Silva. State of the Art in Surface Reconstruction from Point Clouds. In *Eurographics 2014 - State of the Art Reports*, page 25 pages. The Eurographics Association, 2014. doi: 10.2312/EGST.20141040.
- [40] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson Surface Reconstruction. In *Symposium on Geometry Processing*, page 10 pages. The Eurographics Association, 2006. ISBN 978-3-905673-24-1. doi: 10.2312/SGP/SGP06/061-070.
- [41] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Trans. Graph.*, 32(3):29:1–29:13, July 2013. ISSN 0730-0301. doi: 10.1145/2487228.2487237.
- [42] Diana Werner, Ayoub Al-Hamadi, and Philipp Werner. Truncated Signed Distance Function: Experiments on Voxel Size. In Aurélio Campilho and Mohamed Kamel, editors, *Image Analysis and Recognition*, pages 357–364, Cham, 2014. Springer International Publishing. ISBN 978-3-319-11755-3. doi: 10.1007/978-3-319-11755-3_40.
- [43] Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Marc Stamminger. Real-time 3D reconstruction at scale using voxel hashing. *ACM Trans. Graph.*, 32(6):169:1–169:11, 2013. doi: 10.1145/2508363.2508374.
- [44] Olaf Kähler, Victor Adrian Prisacariu, Carl Yuheng Ren, Xin Sun, Philip Torr, and David Murray. Very High Frame Rate Volumetric Integration of Depth Images on Mobile Devices. *IEEE Transactions on Visualization and Computer Graphics*, 21(11):1241–1250, November 2015. ISSN 1941-0506. doi: 10.1109/TVCG.2015.2459891.
- [45] Olaf Kähler, Victor A. Prisacariu, and David W. Murray. Real-Time Large-Scale Dense 3D Reconstruction with Loop Closure. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 500–516, Cham, 2016. Springer International Publishing. ISBN 978-3-319-46484-8. doi: 10.1007/978-3-319-46484-8_30.
- [46] Frank Steinbrücker, Jürgen Sturm, and Daniel Cremers. Volumetric 3D mapping in real-time on a CPU. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2021–2028, May 2014. doi: 10.1109/ICRA.2014.6907127.
- [47] Jaesung Choe, Sunghoon Im, François Rameau, Minjun Kang, and In So Kweon. VolumeFusion: Deep Depth Fusion for 3D Scene Reconstruction. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 16066–16075, 2021. doi: 10.1109/ICCV48922.2021.01578.
- [48] M. Hermann, M. Weinmann, F. Nex, E. K. Stathopoulou, F. Remondino, B. Jutzi, and B. Ruf. Depth estimation and

- 3D reconstruction from UAV-borne imagery: Evaluation on the UseGeo dataset. *ISPRS Open Journal of Photogrammetry and Remote Sensing*, 13:100065, August 2024. ISSN 2667-3932. doi: 10.1016/j.ojphoto.2024.100065.
- [49] Wanquan Feng, Jin Li, Hongrui Cai, Xiaonan Luo, and Juyong Zhang. Neural Points: Point Cloud Representation with Neural Fields for Arbitrary Upsampling. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18612–18621, June 2022. doi: 10.1109/CVPR52688.2022.01808. Conference Name: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) ISBN: 9781665469463.
- [50] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy Networks: Learning 3D Reconstruction in Function Space. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4455–4465, June 2019. doi: 10.1109/CVPR.2019.00459.
- [51] Julian Chibane, Aymen Mir, and Gerard Pons-Moll. Neural Unsigned Distance Fields for Implicit Function Learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual.*, 2020.
- [52] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 165–174, June 2019. doi: 10.1109/CVPR.2019.00025. Conference Name: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) ISBN: 9781728132938.
- [53] Mengqi Ji, Juerger Gall, Haitian Zheng, Yebin Liu, and Lu Fang. SurfaceNet: An End-to-End 3D Neural Network for Multiview Stereopsis. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2326–2334, 2017. doi: 10.1109/ICCV.2017.253.
- [54] Songyou Peng, Chiyu Jiang, Yiyi Liao, Michael Niemeyer, Marc Pollefeys, and Andreas Geiger. Shape As Points: A Differentiable Poisson Solver. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual.*, pages 13032–13044, 2021.
- [55] ZhangJin Huang, Yuxin Wen, ZiHao Wang, Jinjuan Ren, and Kui Jia. Surface Reconstruction From Point Clouds: A Survey and a Benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):9727–9748, December 2024. ISSN 1939-3539. doi: 10.1109/TPAMI.2024.3429209.
- [56] Anurag Dalal, Daniel Hagen, Kjell G. Robbersmyr, and Kristian Muri Knausgård. Gaussian Splatting: 3D Reconstruction and Novel View Synthesis: A Review. *IEEE Access*, 12:96797–96820, 2024. doi: 10.1109/ACCESS.2024.3408318.
- [57] Yiling Yao, Bing Zhang, Wenjuan Zhang, Lianru Gao, Dailiang Peng, Bocheng Li, Yaning Wang, and Bowen Wang. ARSGaussian: 3D Gaussian Splatting with LiDAR for aerial remote sensing novel view synthesis. *ISPRS Journal of Photogrammetry and Remote Sensing*, 231:288–306, January 2026. ISSN 0924-2716. doi: 10.1016/j.isprsjsprs.2025.10.022.
- [58] 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. *Commun. ACM*, 65(1):99–106, 2022. doi: 10.1145/3503250.
- [59] 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. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5835–5844, October 2021. doi: 10.1109/ICCV48922.2021.00580. Conference Name: 2021 IEEE/CVF International Conference on Computer Vision (ICCV) ISBN: 9781665428125.
- [60] 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. *CoRR*, abs/2106.10689, 2021.
- [61] Marie-Julie Rakotosaona, Fabian Manhardt, Diego Martin Arroyo, Michael Niemeyer, Abhijit Kundu, and Federico Tombari. NeRFMeshing: Distilling Neural Radiance Fields into Geometrically-Accurate 3D Meshes. *2024 International Conference on 3D Vision (3DV)*, pages 1156–1165, March 2024. doi: 10.1109/3DV62453.2024.00093. Conference Name: 2024 International Conference on 3D Vision (3DV) ISBN: 9798350362459.
- [62] Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. MVSplat: Efficient 3D Gaussian Splatting from Sparse Multi-view Images. In *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part XXI*, pages 370–386, 2024. doi: 10.1007/978-3-031-72664-4_21.
- [63] Yuedong Chen, Chuanxia Zheng, Haofei Xu, Bohan Zhuang, Andrea Vedaldi, Tat-Jen Cham, and Jianfei Cai. MVSplat360: Feed-Forward 360 Scene Synthesis from Sparse Views. *CoRR*, abs/2411.04924, 2024. doi: 10.48550/ARXIV.2411.04924.
- [64] Ke Yan, Hui Wang, Zhuxin Li, Yuting Wang, Shuo Li, and Hongmei Yang. A Large-Scale 3D Gaussian Reconstruction Method for Optimized Adaptive Density Control in Training Resource Scheduling. *Remote Sensing*, 17(23):3868, January 2025. ISSN 2072-4292. doi: 10.3390/rs17233868.
- [65] Miriam Jäger, Markus Hillemann, and Boris Jutzi. FeatureGS: Eigenvalue-Feature Optimization in 3D Gaussian Splatting for Geometrically Accurate and Artifact-Reduced Reconstruction. *CoRR*, abs/2501.17655, January 2025. doi: 10.48550/ARXIV.2501.17655.
- [66] Pinxuan Dai, Jiamin Xu, Wenxiang Xie, Xinguo Liu, Huamin Wang, and Weiwei Xu. High-quality Surface Reconstruction using Gaussian Surfels. In *ACM SIGGRAPH*

- 2024 Conference Papers, SIGGRAPH 2024, Denver, CO, USA, 27 July 2024- 1 August 2024, page 22, 2024. doi: 10.1145/3641519.3657441.
- [67] Jichan Park, Jae-Won Suh, and Yuseok Ban. Dual-Dimensional Gaussian Splatting Integrating 2D and 3D Gaussians for Surface Reconstruction. *Applied Sciences*, 15(12):6769, June 2025. ISSN 2076-3417. doi: 10.3390/app15126769.
 - [68] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth Anything V2. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024.*, 2024.
 - [69] Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-Motion Revisited. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4104–4113, June 2016. doi: 10.1109/CVPR.2016.445.
 - [70] Tehreem Rabbani, F.A. Heuvel, and George Vosselman. Segmentation of point clouds using smoothness constraint. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36, January 2006.
 - [71] Radu Bogdan Rusu and Steve Cousins. 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation, ICRA 2011, Shanghai, China, 9-13 May 2011*, 2011. doi: 10.1109/ICRA.2011.5980567.
 - [72] Zhuoxiao Li, Shanliang Yao, Taoyu Wu, Yong Yue, Wufan Zhao, Rongjun Qin, Ángel F. García-Fernández, Andrew Levers, Jason Ralph, and Xiaohui Zhu. ULSR-GS: Urban large-scale surface reconstruction Gaussian Splatting with multi-view geometric consistency. *ISPRS Journal of Photogrammetry and Remote Sensing*, 230:861–880, December 2025. ISSN 0924-2716. doi: 10.1016/j.isprsjprs.2025.10.008.
 - [73] Henrik Aanæs, Rasmus Ramsbøl Jensen, George Vogiatzis, Engin Tola, and Anders Bjarholm Dahl. Large-Scale Data for Multiple-View Stereopsis. *Int. J. Comput. Vis.*, 120(2): 153–168, 2016. doi: 10.1007/S11263-016-0902-9.
 - [74] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: benchmarking large-scale scene reconstruction. *ACM Trans. Graph.*, 36(4):78:1–78:13, 2017. doi: 10.1145/3072959.3073599.
 - [75] Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. Deep blending for free-viewpoint image-based rendering. *ACM Trans. Graph.*, 37(6):257:1–257:15, December 2018. ISSN 0730-0301. doi: 10.1145/3272127.3275084.
 - [76] Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. BlendedMVS: A Large-Scale Dataset for Generalized Multi-View Stereo Networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 1787–1796, 2020.

Appendix A The Failure of PGSR in Photogrammetry Dataset

When applying PGSR for photogrammetry-based surface reconstruction on the UAV dataset, we found that the resulting mesh contained numerous repetitive structures and artifacts extending across the entire scene, likely attributable to depth estimation bias in PGSR under nadir-view photogrammetry. Consequently, we excluded PGSR from the reported photogrammetry results, as we consider this method unsuitable for this type of data. The failure is visualized in Figure A1.



Figure A1. Visualization of the failure of PGSR when applied to UAV dataset.

Appendix B The Accuracy–Efficiency Trade-Off Results

The results of accuracy–efficiency trade-off for surface extraction (F1 Score and time) under diverse voxel sizes between our method and primary baseline, TSDF, are displayed in Figure A2. The results are reported on our UAV dataset at the real-world object scale. Due to the similarity in SDF iso-surface extraction via voxel-based representations, the two methods exhibit comparable behavior under certain settings.

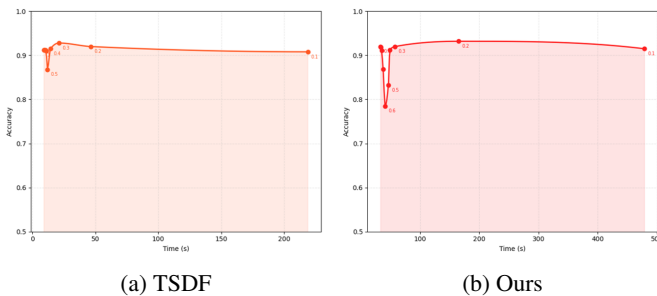


Figure A2. Accuracy–efficiency trade-off results under diverse voxel sizes.

As shown in the curves, for both voxel-based methods, the time required for surface extraction decreases as the voxel size increases. However, accuracy does not improve monotonically with decreasing voxel size. Instead, both methods exhibit local

maxima and minima in the accuracy curve. This may be because varying the voxel size also modulates the level of mesh noise. At the local maxima around the voxel size of 0.2, our method outperforms the TSDF baseline, underscoring its higher absolute accuracy. However, as the voxel size increases, our method yields lower accuracy minima, which may be attributed to sporadic noise at specific scales. Investigating and addressing this behavior remains an avenue for future work.

Appendix C Sensitivity Analysis

To evaluate the effectiveness of our empirical parameter selection and the sensitivity of result accuracy to each parameter, we conducted a sensitivity analysis of GSSA on the UAV dataset. We selected eight parameters: denoiser thresholds ϵ_s , ϵ_{dn} ; filler parameters k , ϵ_d , ϵ_p , ϵ_n ; and culling thresholds ϵ_p , ϵ_w . For each parameter, we scaled the default value upward and downward within a small range and report the resulting change in F1 Scores, as shown in Figure A3. Most of those parameters are empirically determined, while subsets such as ϵ_n and ϵ_p are adopted according to settings of prior work.

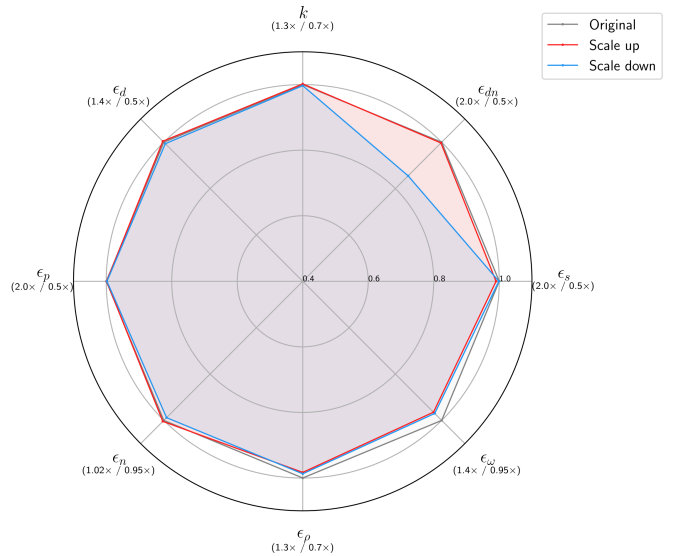


Figure A3. Sensitivity analysis of our method on 8 parameters for surface extraction.

Overall, modest perturbations of these parameters do not substantially affect the final reconstruction quality, with most results remaining within a reasonable range. The plane-identification parameters such as ϵ_d , ϵ_p exhibit low sensitivities. Varying them has negligible impact on plane filling and the final output, which underscores the stability of this module and the reasonableness of our chosen empirical ranges. Even for parameters such as ϵ_p and ϵ_w that show somewhat more noticeable variation for accuracy, our selected default values consistently yield superior performance.

In contrast, the denoising-related parameter ϵ_{dn} shows greater influence on the results when varied, indicating that it plays a critical role in the 3D refiner. Specifically, overly aggressive denoising leads to the loss of key structural details. This also confirms the effectiveness of our empirically selected value of

ϵ_{dn} . At the same time, this sensitivity within the pipeline suggests that when deploying GSSA to other scenes, the parameter ϵ_{dn} may need to be prioritized for adjustment to maintain reconstruction accuracy.

Appendix D Statistics-Based Planar Filling

In addition to the hard-constraint AND logic used in our plane filler, we compared two alternative approaches that adaptively fill Surfels for planar areas based on scene geometries: the PCA-based method and RANSAC-based method.

Specifically, the PCA-based method calculates the eigenvalues $\lambda_1, \lambda_2, \lambda_3$ of the covariance matrix formed from each Surfel and its $k = 10$ neighboring points to determine planarity with an applied threshold $\epsilon_\lambda = 0.1$:

$$s \in \text{Plane} \quad \text{if} \quad \frac{\lambda_1}{\lambda_1 + \lambda_2 + \lambda_3} < \epsilon_\lambda \quad (\text{A1})$$

The RANSAC-based method randomly samples $n=3$ points from $k=10$ neighboring points of each Surfel. A plane is fitted to these Surfel points and the number of inliers is counted. If a valid local plane can be established, the inlier Surfels is filled.

The comparison is conducted on the UAV dataset by treating different planar regions identification methods, hard-constraint AND logic method in GSSA, PCA-based method and RANSAC-based method, as interchangeable modules within the mesh extraction pipeline under the voxel size of 0.2. The accuracy and efficiency results are reported in Table A1.

Table A1. Comparison between different planar identification methods.

Method	F1 Score (%) $\uparrow$	Time (s) $\downarrow$
Hard-constraint AND logic	93.2	164.9
PCA-based	93.5	128.1
RANSAC-based	93.0	1137.1

The results indicate that statistics-based planar regions filling schemes, such as the PCA-based method, can achieve accuracy comparable to or even higher than hard-threshold methods, while offering greater efficiency due to the smaller number of neighboring points required. In contrast, the RANSAC-based method incurs substantially lower efficiency because it requires iterative plane fitting at each Surfel. In summary, PCA-based local-statistical plane identification is a promising direction for Surfel filling considering both on accuracy and efficiency. A current limitation is that it considers only the point cloud geometry without incorporating the geometric relationship with Surfel normals. Further exploration and extension of such methods, including the integration of normal information, will also be pursued in our future work.

Appendix E Visualization of Normal Maps

To illustrate the surface smoothness of the reconstruction results and reveal geometric details from additional viewpoints, we present the normal maps corresponding to selected reconstructed surfaces in Figure A4. Since PGSR is not applicable to UAV

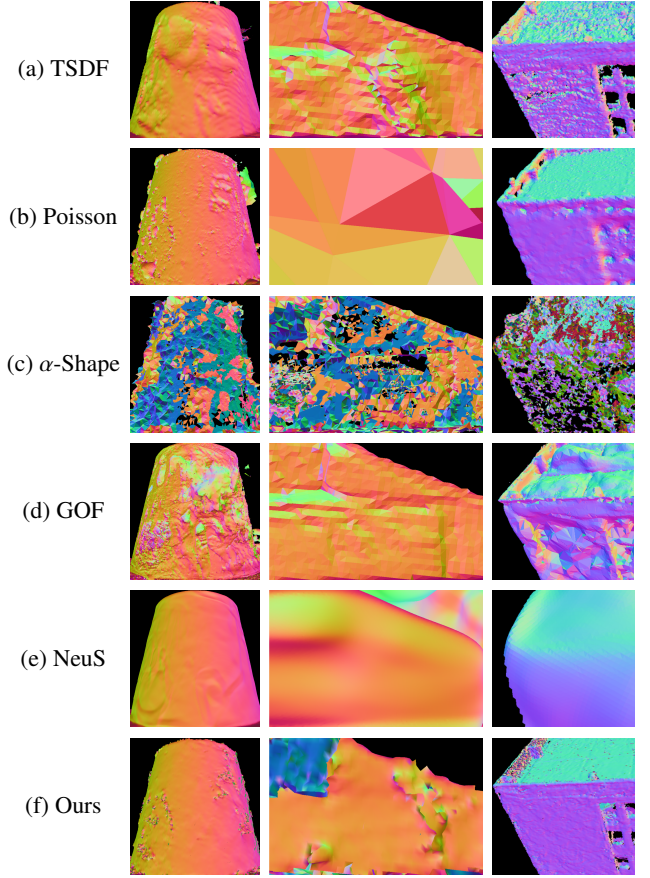


Figure A4. Normal maps of several surface reconstruction results.

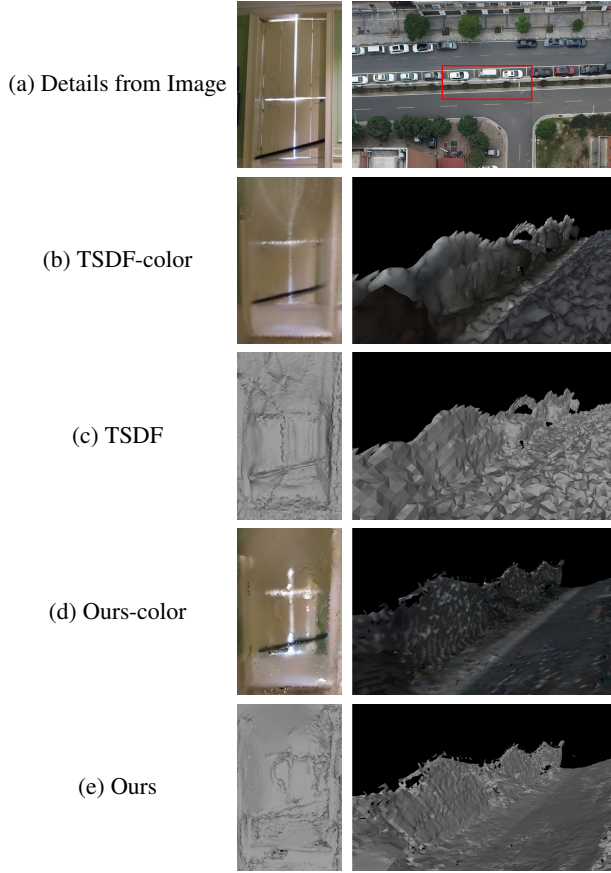
photogrammetric data, as mentioned above, the normal map results accordingly exclude this method.

The visualization results demonstrate that our method consistently preserves high-quality normal surfaces across a majority of datasets, achieving a favorable balance between positional accuracy and surface smoothness. Nevertheless, on certain datasets it remains susceptible to noise, which introduces minor, spurious surface distortions that degrade the visual quality. As noted earlier, further enhancing surface smoothness while maintaining high geometric fidelity remains a key direction for our future work.

Appendix F Additional Results

Although our method achieves superior accuracy across many scenarios, it still has certain inherent limitations. To this end, we provide additional visualizations of representative failure cases on the reconstructed surface. Figure A5 displays the failures of reconstructed surfaces compared between our method and TSDF baseline.

The failure cases here stem from two categories of causes. The first involves terrestrial data, where the sudden appearance of strong illumination leads to positional errors in surface reconstruction. This issue likewise affects rendering-based methods. In fact, positional distortions already arise during the training



As can be observed, our method achieves a moderate level of performance relative to competing approaches. This outcome should be interpreted with the following considerations in mind, including that most existing methods adopt a rendering-based reconstruction paradigm, which inherently prioritizes rendering quality. In contrast, our work is situated within the domain of surface reconstruction, where geometric accuracy constitutes the more appropriate evaluation criterion. Therefore, the visual quality metrics should serve only as a supplementary reference rather than a primary evaluation criterion.

Figure A5. Failure cases in surface reconstruction. The red box in the image region indicates the areas where the reconstructed surface is displayed.

of Gaussian Surfels. This is consistent with the limitations described above: Gaussian Splatting-based methods struggle to model transparent or strongly lit objects during training. The second failure occurs in airborne data, where boundary reconstruction of the scene fails due to insufficient image coverage and limited optimization in those regions. This is a problem that also manifests in rendering-based methods due to the same reason.

To quantitatively evaluate meshes from a perceptual quality perspective, we render reconstructed meshes against the ground-truth multi-view images on the DTU dataset and assess the results using visual quality metrics (PSNR). Since the surfaces reconstructed by NeuS lack texture information, our evaluation excludes this method. The results are shown in Table A2.

Table A2. Visual quality metrics on reconstructed surfaces.

Method	PSNR $\uparrow$
TSDF	7.19
Poisson	6.61
α-Shape	7.01
GOF	8.10
PGSR	8.51
Ours	7.79