Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Gaussian Surface Reconstruction with Adaptive Depth Supervision for Geographic Information Systems

Xiangqian Shen
Wuhan University
Yusen Wang
DFRD
Tuo Cao
Institute of AI for Industries, Chinese Academy of Sciences
Jiongming Qin
Wuhan University
Chunxia Xiao
cxxiao@whu.edu.cn
Wuhan University

Research Article

Keywords:

Posted Date: November 16th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11609305/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Gaussian Surface Reconstruction with Adaptive Depth Supervision for Geographic Information Systems

Xiangqian Shen[1,2], Yusen Wang[3], Tuo Cao[4], Jiongming Qin[1], Chunxia Xiao[1*]

[1]School of Computer Science, Wuhan University, Wuhan, Hubei, China.
[2]Kunming Prospecting Design Institute of China Nonferrous Metals Industry Co., Ltd., Kunming, Yunnan, China.
[3]DFRD, Wuhan, Hubei, China.
[4]Institute of AI for Industries, Chinese Academy of Sciences, Nanjing, Jiangsu, China.

*Corresponding author(s). E-mail(s): cxxiao@whu.edu.cn;
Contributing authors: shenxq@whu.edu.cn;
tc-wangyusen@dfmc.com.cn; tcao@iaii.ac.cn; jiongming@whu.edu.cn;

Abstract

Surface reconstruction from overlapping images supplies geometry for three-dimensional geographic information systems (GIS), but stereo depth errors can constrain the recovered surfaces. We present HelixGau, a Gaussian surface reconstruction method that filters and updates stereo depth during optimization. Local correlation with monocular depth selects pixels for supervision, while stereo estimates supply the supervised values. Rendered planes with higher photometric correlation across views provide candidates for stereo plane updates. The updated depth maps are filtered before supervision, and surfaces are extracted after optimization. On 16 BlendedMVS scenes, mean F1 is 88.64 with a standard deviation of 0.09 over four runs, compared with the highest baseline mean of 76.85. Mean F1 is 0.500 on six Tanks and Temples scenes. On the Technical University of Denmark (DTU) benchmark, overall error is 0.505 mm, compared with the lowest baseline error of 0.504 mm. Component ablations measure the separate effects of filtering and updates on the reconstructed geometry.

Keywords: Geographic information systems, Spatial data acquisition, Surface reconstruction, Gaussian splatting, Multi-view stereo, Depth supervision

1 Introduction

Three-dimensional (3D) geographic information systems (GIS) use surface models to represent the geometry of observed environments. Acquiring these models from overlapping images is a computational step before their registration and organization with geographic information. Errors in the acquired surface remain relevant after this processing, even when rendered images appear plausible. Han et al. [8] examine the placement of 3D models in GIS, while Wang et al. [22] review image-based reconstruction from a photogrammetric perspective. Together, these studies motivate treating surface geometry as a distinct output of spatial data acquisition.

Three-dimensional Gaussian splatting [11] represents a scene with explicit primitives optimized through image rendering. Planar primitives and rendered depth connect this representation to surface reconstruction [3, 5, 9]. Image fitting, however, does not uniquely determine the surface recovered from those primitives. Different geometric configurations can produce similar images, particularly where observations provide limited constraints. Depth supervision introduces additional geometric information into optimization, but the reconstruction then depends on which depth estimates are used and how long they remain fixed.

Stereo depth is estimated from correspondences across images and their camera geometry. Its reliability varies within a scene because visibility, texture, and image coverage vary across pixels. A depth map can therefore contain both surface detail and matching errors. Applying the entire map as supervision gives these estimates the same role in the depth loss, although their agreement with the underlying surface differs. We therefore examine local selection of stereo depth without ground-truth geometry during training.

Monocular depth supplies complementary information about local scene structure [20, 25]. It can be used either as a numerical depth target or as evidence for selecting another target. These choices have different consequences for supervision. We use local correlation between monocular depth and stereo depth to select pixels, while retaining the stereo values in the depth loss. This separates the source of numerical supervision from the information used to filter it. Correlation measures agreement in local variation, and shared errors can still pass this test.

The depth maps also need a mechanism for revision during optimization. As Gaussian parameters change, their rendered planes provide alternatives to the initial stereo planes. A fixed stereo map cannot incorporate these candidates. Conversely, directly using rendered depth as its own target can preserve reconstruction errors. We therefore compare the image patches induced by rendered and stereo planes across views. Rendered planes with higher photometric correlation provide candidates for stereo plane updates, after which the resulting depth maps are filtered again. The comparison uses image evidence rather than assuming that the current rendered geometry is correct.

We present HelixGau, which combines these filtering and update operations within the rendering formulation of planar-based Gaussian splatting reconstruction (PGSR) [3]. Stereo plane estimates are maintained alongside the Gaussian representation. Selected stereo depths supervise Gaussian parameters between updates, and rendered planes supply candidates at the update intervals. The pretrained monocular

estimator and plane matching operate outside the gradient optimization of the Gaussians. Surface extraction follows optimization. The method thus changes the use of depth supervision without requiring semantic object labels or differentiating through an extracted mesh.

Filtering changes the pixel set used by the depth loss, whereas updates change the stereo planes from which its targets are computed. These decisions cannot be assessed solely by observing whether a rendered image becomes closer to a training image. An updated plane may produce higher photometric correlation while retaining a depth error, and two depth estimates can share an incorrect local structure. The reconstructed surface and the intermediate depth maps therefore provide different evidence about the same optimization process.

We examine both types of evidence. Surface distances, precision, and recall measure the geometric output after extraction. Depth-error measurements assess the stereo values used for supervision, while update-mask measurements describe the pixels selected for plane updates. The component ablations distinguish configurations with stereo depth, monocular depth filtering, and depth updates. Together, these measurements show what changes when each operation is enabled, rather than attributing all differences between complete methods to one component. The diagnostics use reference geometry for assessment; reference geometry does not enter the filtering or update rules. Runtime and memory measurements describe the computation needed to maintain and revise the stereo estimates alongside the Gaussians. The evaluation thus follows the method from depth selection through optimization to its extracted geometric output.

Evaluation uses the Technical University of Denmark (DTU) benchmark [10], a 16-scene BlendedMVS subset [26], and six Tanks and Temples scenes [12]. The experiments measure extracted surface geometry. Component ablations examine filtering and updates, while depth errors and update-mask measurements relate these operations to the targets used during optimization. Computational measurements report their cost. This links the problem of selecting and revising depth supervision to the geometry produced for spatial modeling.

The contributions are as follows.

- A Gaussian surface reconstruction procedure maintains stereo plane estimates during optimization, combines depth filtering with depth updates, and extracts surface geometry after optimization for image-based spatial data acquisition.
- A local correlation test uses monocular depth to select stereo depth targets. Monocular estimates control pixel selection, while stereo estimates supply the numerical values used for supervision.
- A photometric comparison across views uses rendered planes as candidates for stereo plane updates. The resulting depth maps are filtered before they enter subsequent depth supervision.
- Experiments on three reconstruction benchmarks measure surface geometry, component effects, stereo depth errors, update-mask behavior, and computational cost. Four runs characterize the full method on the evaluated BlendedMVS subset.

2 Reconstruction problem

2.1 Input and reconstructed geometry

Let $\mathcal{I} = \{I_v, K_v, R_v, t_v\}_{v=1}^{V}$ contain V overlapping images with camera intrinsics and poses. Let $\mathcal{P}_0$ be a sparse point cloud obtained by structure from motion (SfM). The camera estimates and sparse points are supplied to reconstruction. They determine the image projections and initialize the Gaussian primitives. The reconstruction considered here holds the cameras fixed and optimizes the Gaussian parameters.

The output is an optimized Gaussian representation $\mathcal{G}$ together with extracted surface geometry. Rendered depth and normals are intermediate outputs used to obtain this surface. In GIS data preparation, the extracted geometry can subsequently be registered and organized with geographic information. The task addressed here is to estimate that geometry from the images, rather than assign semantic classes to reconstructed objects. This input and output definition also distinguishes the optimized scene representation from the surface measured in the experiments.

2.2 Roles of the depth estimates

For each reference view, a monocular estimator produces D_{mono}. A stereo estimator maintains a plane map $F = \{f_p\}$, where each f_p describes a plane at reference pixel p. Its intersection with the corresponding camera ray gives the stereo depth $D_{\text{multi}}(p)$. The current Gaussian representation produces rendered depth D_{render} and normals N_{render} in the same reference view.

These depth estimates have separate roles. Monocular depth supplies local variation for a correlation test. Stereo depth supplies the numerical targets for the depth loss. Rendered depth and normals describe the current reconstruction and provide plane candidates for updating the stereo estimates. Monocular values do not replace stereo values in the depth loss. Similarly, rendered depth is not used as an unrestricted copy of the target map. Its plane candidates must pass the photometric comparison described in Section 3.4.

2.3 Depth filtering and depth updates

The binary filter mask M_{conf} selects pixels that pass the local correlation, stereo validity, and photometric cost tests. Their stereo depths enter the depth loss. The mask therefore determines where the depth loss is evaluated. Its correlation test compares local variation in the monocular and stereo maps, without using ground-truth geometry. The mask does not represent a calibrated probability that a depth is correct.

A separate binary mask M_{update} identifies rendered plane candidates selected by comparing normalized cross-correlation (NCC) across views. This mask controls changes to the stereo plane map, rather than directly selecting pixels for the Gaussian depth loss. At an update interval, the stereo estimator evaluates the selected rendered planes and perturbed candidates. Depth is recomputed from the resulting planes and filtered before subsequent supervision. A pixel selected for an update is therefore not automatically accepted for the depth loss.

Between updates, Gaussian optimization uses the current selected stereo depths. The rendered geometry changes as the Gaussian parameters change and supplies candidates at later update intervals. Filtering and updates consequently act on different parts of this process. Filtering selects depth targets, while updates revise their values through the stereo planes. Figure 1 shows their execution within reconstruction.

2.4 Optimization and geometric assessment

Gradients optimize Gaussian positions, orientations, scales, opacity, and appearance parameters. The pretrained monocular network is used for inference. Stereo matching, binary filtering, and plane updates are auxiliary operations outside this gradient computation. Surface fusion is performed after optimization, using the dataset-specific extraction procedures described in Section 4.1.

The experiments assess the extracted surface against reference geometry. Component ablations test how filtering and updates change this output. Separate depth-error and update-mask measurements examine the stereo estimates used during optimization, while runtime and memory measurements quantify the associated computation. These assessments connect the two depth operations to surface reconstruction without treating image fit as a substitute for geometric evaluation.

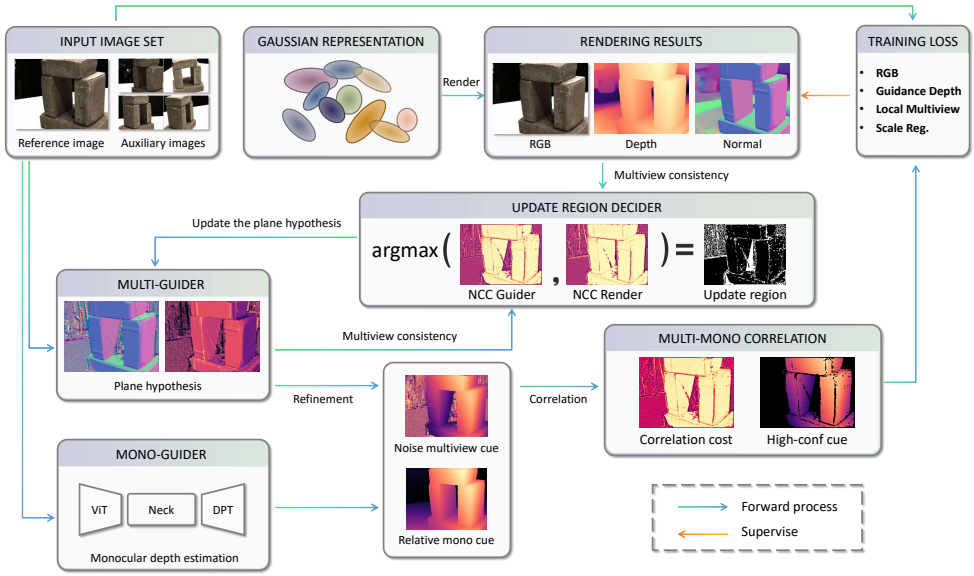


Fig. 1 HelixGau surface reconstruction. Mono-Guider and Multi-Guider denote the monocular and stereo depth estimators. Multi-Mono Correlation denotes depth filtering using local correlation, and Update Region Decider denotes the photometric comparison used for depth updates. The updated stereo depth is filtered before supervision

3 Gaussian surface reconstruction with depth filtering and updates

3.1 Planar Gaussian representation

We use the planar Gaussian rendering formulation of PGSR [3]. A primitive with mean μ_i, rotation R_i, and scale matrix S_i has covariance $\Sigma_i = R_i S_i S_i^\top R_i^\top$ and density

$$G_i(x) = \exp\left[-\tfrac{1}{2}(x-\mu_i)^\top \Sigma_i^{-1}(x-\mu_i)\right]. \tag{1}$$

For camera rotation R_v and translation t_v, its image center is $\mu_i' = \pi(K_v(R_v\mu_i + t_v))$, where π denotes homogeneous division. Its image covariance is $\Sigma_i' = J_v R_v \Sigma_i R_v^\top J_v^\top$, with J_v the Jacobian of perspective projection at the camera-space mean. Front-to-back blending gives

$$C(p) = \sum_{i\in\mathcal{N}(p)} T_i(p)\alpha_i(p)c_i, \qquad T_i(p) = \prod_{j<i}(1-\alpha_j(p)). \tag{2}$$

Here $\mathcal{N}(p)$ contains the primitives projected onto pixel p, $\alpha_i(p)$ is the opacity-weighted Gaussian contribution, and $T_i(p)$ is transmittance before primitive i. The view-dependent color c_i is represented by spherical harmonics. A scale penalty $\mathcal{L}_s$ minimizes the smallest principal scale of each Gaussian, promoting planar primitives.

Let $n_i^{(v)}$ and $d_i^{(v)}$ denote a primitive's normal and plane distance in reference-camera coordinates. The rendered normal and plane-distance maps are

$$N_{\text{render}}(p) = \sum_i T_i(p)\alpha_i(p)n_i^{(v)}, \tag{3}$$

$$\text{Dist}(p) = \sum_i T_i(p)\alpha_i(p)d_i^{(v)}. \tag{4}$$

Depth is obtained by intersecting the viewing ray with the blended plane

$$D_{\text{render}}(p) = \frac{\text{Dist}(p)}{N_{\text{render}}(p)^\top K_v^{-1}\tilde{p}}. \tag{5}$$

Here $\tilde{p}$ is the homogeneous pixel coordinate. Following Mini-Splatting [6], we rank primitives by their maximum alpha contribution across views and periodically remove the lowest-weighted 1%. Depth filtering and updates operate during optimization of this Gaussian representation.

3.2 Monocular depth and stereo depth

Depth Anything V2 [25] produces the monocular depth map D_{mono} used for filtering. A custom implementation on the NVIDIA CUDA platform for graphics processing unit (GPU) computing estimates stereo depth using PatchMatch [2] and the stereo

method of Xu et al. [24]. It maintains a plane $f_p = (n_p, d_p)$ at each reference pixel. In reference-camera coordinates the plane satisfies $n_p^\top X = d_p$.

A hypothesis warps a reference patch into a source view through

$$\tilde{q}' \sim H(f_p)\tilde{q}, \qquad H(f_p) = K_{\mathrm{src}} \left(R + \frac{T n_p^\top}{d_p} \right) K_{\mathrm{ref}}^{-1}, \tag{6}$$

where R, T transform reference-camera coordinates to source-camera coordinates. For image patch W_p and patch means $\bar{I}_{\mathrm{ref}}$ and $\bar{I}_{\mathrm{src}}$, its photometric score is

$$\mathrm{NCC}(p, f_p) = \frac{\sum_{q \in W_p} (I_{\mathrm{ref}}(q) - \bar{I}_{\mathrm{ref}})(I_{\mathrm{src}}(q') - \bar{I}_{\mathrm{src}})}{\sqrt{\sum_{q \in W_p} (I_{\mathrm{ref}}(q) - \bar{I}_{\mathrm{ref}})^2 \sum_{q \in W_p} (I_{\mathrm{src}}(q') - \bar{I}_{\mathrm{src}})^2}}. \tag{7}$$

Higher NCC means better photometric agreement. The equivalent matching cost is $c = 1 - \mathrm{NCC}$, for which lower is better. This convention applies to stereo estimation and depth updates.

PatchMatch propagates neighboring hypotheses and introduces random perturbations $\mathcal{R}_p$ according to

$$f_p \leftarrow \arg \max_{f \in \{f_p, f_{x-1,y}, f_{x,y-1}\} \cup \mathcal{R}_p} \mathrm{NCC}(p, f). \tag{8}$$

The plane induces $D_{\mathrm{multi}}(p) = d_p / (n_p^\top K_{\mathrm{ref}}^{-1} \tilde{p})$. The stereo depth values provide the targets for depth supervision. Depth filtering determines which of these targets enter the depth loss.

3.3 Depth filtering

Depth filtering compares local variation in monocular depth and stereo depth. Each map is divided by its maximum before the comparison

$$D'_k = D_k / \max(D_k), \qquad k \in \{\mathrm{mono}, \mathrm{multi}\}. \tag{9}$$

This normalization changes the numerical range without estimating an alignment between the maps. Let M be the binary validity mask and $\Omega(p)$ an $s \times s$ neighborhood. For windows with $N_p = \sum_{q \in \Omega(p)} M(q) > 0$, define

$$\mu_k(p) = \frac{1}{N_p} \sum_{q \in \Omega(p)} M(q) D'_k(q), \tag{10}$$

$$\sigma_k(p) = \left[\frac{1}{N_p} \sum_{q \in \Omega(p)} M(q) (D'_k(q) - \mu_k(p))^2 + \epsilon \right]^{1/2}, \tag{11}$$

$$\rho(p) = \frac{1}{N_p} \sum_{q \in \Omega(p)} \frac{M(q)(D'_{\mathrm{mono}}(q) - \mu_{\mathrm{mono}}(p))(D'_{\mathrm{multi}}(q) - \mu_{\mathrm{multi}}(p))}{\sigma_{\mathrm{mono}}(p)\sigma_{\mathrm{multi}}(p)}. \tag{12}$$

We use $\epsilon = 10^{-6}$. Ignoring this stabilizer, centered normalized correlation is invariant to positive affine transformations of either map. The invariance follows from centering and normalization by standard deviation. The property applies to affine changes within the compared depth representation, not arbitrary nonlinear transformations.

The filter mask selects pixels with $\rho(p) > \tau_c$ that also pass the stereo validity and photometric cost tests. Let M_{valid} denote their combined binary mask. Then

$$M_{\text{conf}}(p) = M_{\text{valid}}(p)\,\mathbf{1}[\rho(p) > \tau_c]. \tag{13}$$

The default correlation threshold is $\tau_c = 0.3$, and the NCC cost parameter is 0.4. The correlation score measures agreement in local depth variation. It is not a calibrated probability of correctness. Shared depth errors and patches with little depth variation can limit the filtering.

3.4 Depth updates

At fixed optimization intervals, rendered planes f_p^{render} are compared with the stereo planes f_p^{stereo} using Eq. 7. A rendered plane is selected for an update when

$$M_{\text{update}}(p) = \mathbf{1}\big[\text{NCC}(p, f_p^{\text{render}}) > \text{NCC}(p, f_p^{\text{stereo}})\big]\,. \tag{14}$$

At selected pixels, the stereo estimator evaluates the rendered planes together with randomly perturbed candidates and keeps the plane with the highest photometric correlation. It then computes stereo depth from the updated planes. Depth filtering and the stereo validity tests determine which updated depths enter the depth loss.

The update rule revises the stereo planes, while depth filtering controls where stereo depth supervises the Gaussians. Higher NCC can still accompany larger geometric error under ambiguous texture, occlusion, or view-dependent reflectance. Section 4.5 measures the updated depth against ground truth. Figure 2 shows stereo and rendered depth during optimization.

3.5 Optimization and surface extraction

The Gaussian parameters minimize

$$\mathcal{L} = \mathcal{L}_{\text{rgb}} + \lambda_s \mathcal{L}_s + \lambda_{\text{mvc}} \mathcal{L}_{\text{mvc}} + \lambda_{\text{depth}} \mathcal{L}_{\text{depth}}. \tag{15}$$

The appearance loss combines absolute image error with the structural similarity index (SSIM)

$$\mathcal{L}_{\text{rgb}} = (1 - \lambda_{\text{ssim}})\|C - \hat{C}\|_1 + \lambda_{\text{ssim}}(1 - \text{SSIM}(C, \hat{C})). \tag{16}$$

Here $\hat{C}$ is the reference image. The planarization term $\mathcal{L}_s$ penalizes the minimum Gaussian scale. The local multi-view term $\mathcal{L}_{\text{mvc}}$ follows PGSR's rendered-plane photometric consistency. For a nonempty set of selected pixels, depth supervision is

$$\mathcal{L}_{\text{depth}} = \frac{\sum_p M_{\text{conf}}(p)\|D_{\text{render}}(p) - D_{\text{multi}}(p)\|_1}{\sum_p M_{\text{conf}}(p)}. \tag{17}$$

The target in Eq. 17 is stereo depth. Monocular depth affects its selection through Eq. 13. The monocular network is used for inference, and PatchMatch estimates the stereo planes. Gradients optimize the Gaussian parameters under the selected stereo depths. The binary filtering and depth updates are performed outside this gradient computation.

After optimization, rendered geometry is fused into a surface representation. DTU uses truncated signed distance function (TSDF) fusion [4]; BlendedMVS uses depth-consistency filtering and point-cloud fusion. This final extraction is outside the differentiable optimization objective. Geometric evaluation measures these extracted outputs.

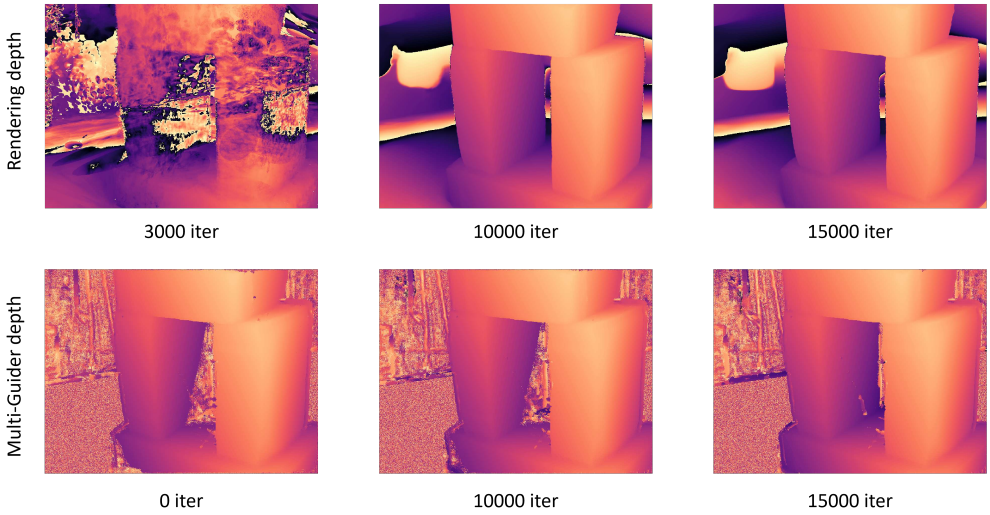


Fig. 2 Stereo depth and rendered depth during optimization. The Multi-Guider label identifies the stereo depth estimator. Rendered planes selected by photometric correlation are used to update the stereo planes

4 Experimental evaluation

4.1 Datasets, selection, and evaluation protocol

We evaluate on three benchmarks. DTU [10] comprises the 15 object-centric scans listed in Appendix A. BlendedMVS [26] comprises the 16 evaluated scenes whose complete identifiers appear in Appendix A. Tanks and Temples [12] comprises Barn, Caterpillar, Courthouse, Ignatius, Meetingroom, and Truck.

BlendedMVS subset provenance. We randomly sampled 20 scenes from the dataset's single-object and large-scene categories. Four were subsequently excluded after raw-data problems and unsuccessful reconstruction or evaluation by some compared methods. The quantitative tables therefore cover the 16 retained scenes, not the full initial sample. The final 16-scene subset is conditional on this exclusion process.

The four excluded scene identifiers and method-specific failure logs are not reported here; the results should therefore be read as comparisons on the retained scenes.

Surface evaluation. DTU depth maps are fused by TSDF and evaluated against official ground-truth point clouds. The mean reconstruction-to-reference distance (d2s) measures accuracy; reference-to-reconstruction distance (s2d) measures completeness. Overall is their arithmetic mean. DTU distances are in millimeters. On BlendedMVS, rendered depth and normal maps undergo multi-view consistency filtering and unprojection before point-cloud fusion. Precision and recall are nearest-surface match proportions, and F1 is their harmonic mean. For scene i, let

$$L_i = \max_{a \in \{x,y,z\}} (b_{i,a}^{\max} - b_{i,a}^{\min}), \qquad \delta_i = L_i/2048, \tag{18}$$

where the bounds are those of the ground-truth scene. Equivalently, a match requires normalized distance $d/L_i \leq 1/2048$. The same rule is applied to all compared methods. We also report stricter tolerances $L_i/3072$ and $L_i/4096$. These normalized-distance thresholds are distinct from the correlation threshold τ_c. We apply this BlendedMVS evaluation protocol to every compared method.

Tanks and Temples results use the benchmark evaluation procedure with the reported rigid alignment. F1 is shown on a 0 to 1 scale there and on a 0 to 100 scale for BlendedMVS. Means are scene-wise averages; the mean of scene-wise F1 values need not equal the harmonic mean of the averaged precision and recall.

4.2 Implementation and comparison conditions

All optimization-based comparisons use one NVIDIA RTX 3090 GPU. Gaussian primitives are initialized from COLMAP SfM points [18]. Depth Anything V2 produces monocular depth, and the stereo estimator runs at half the input image resolution. We use $s = 7$, $\tau_c = 0.3$, NCC cost parameter 0.4, $\lambda_s = 100$, $\lambda_{\text{mvc}} = 0.15$ on 7×7 patches, and $\lambda_{\text{depth}} = 1$. Depth updates occur every 5,000 iterations within a 30,000-iteration optimization. These settings are shared across scenes.

The Gaussian baselines include two-dimensional Gaussian splatting (2DGS) [9], Gaussian opacity fields (GOF) [27], and PGSR+ [3]. We also compare rasterizing depth in Gaussian splatting (RaDe-GS) [28], sorted opacity fields (SOF) [15], mesh-in-the-loop Gaussian splatting (MILo) [7], and triangulation-guided Gaussian splatting (TriaGS) [19]. PGSR+ denotes the officially updated implementation used in the experiments. For DTU and BlendedMVS, the compared optimization methods are evaluated with common input images, camera poses, image resolution, sparse initialization, and evaluation code. All methods use TSDF extraction on DTU and depth-consistency fusion on BlendedMVS. Each baseline uses its repository-recommended configuration without per-scene retuning. The cost comparison uses 30,000 iterations per method. The Tanks and Temples runs follow the 2DGS-compatible input, extraction, and evaluation setup; their comparison includes SOF, MILo, and TriaGS.

We report means and standard deviations from four independent HelixGau runs on BlendedMVS. The compared baselines, other ablation variants, DTU, and Tanks and Temples are reported from single runs. Consequently, the reported standard deviation

describes HelixGau's variability on BlendedMVS and does not establish paired statistical significance against the baselines. In particular, the 0.001 mm DTU difference is interpreted only as numerical proximity.

4.3 Geometric reconstruction across scene configurations

On DTU, HelixGau has average d2s/s2d/overall values of 0.499/0.511/0.505 mm. PGSR+ yields 0.496/0.512/0.504 mm. HelixGau has a lower completeness error by 0.001 mm, while PGSR+ has lower accuracy and overall errors by 0.003 and 0.001 mm, respectively. Appendix B reports all per-scene distances.

On the retained BlendedMVS subset, HelixGau's mean precision, recall, and F1 are 90.27±0.28, 90.00±0.09, and 88.64±0.09, respectively. TriaGS has the highest baseline mean F1 at 76.85. The difference from HelixGau is 11.79 F1 points. Table 1 gives scene-wise F1; Appendix B provides precision and recall. HelixGau has the highest reported F1 in 14 of 16 scenes. It does not lead on S05 or S16; S16 also illustrates a remaining completeness limitation, with recall 32.96 compared with TriaGS at 35.66.

Table 1 BlendedMVS F1 at $\delta_i = L_i/2048$ (0 to 100; higher is better). HelixGau reports mean ± standard deviation over four runs; baselines use one run. Complete scene identifiers are in Appendix A. Bold and underlined values are best; underlined values are second best

Scene	2DGS	GOF	PGSR+	RaDe-GS	SOF	MILo	TriaGS	HelixGau
S01	76.53	90.93	87.10	88.22	88.35	90.22	90.93	**93.41±0.17**
S02	32.83	55.90	77.36	46.23	58.85	49.66	53.47	**93.96±0.17**
S03	13.68	47.95	43.47	52.03	54.47	48.19	66.44	**69.88±0.16**
S04	26.71	54.26	93.04	58.67	64.27	59.20	87.17	**93.71±0.03**
S05	76.68	72.12	**98.41**	75.16	51.73	82.50	91.80	98.39±0.21
S06	29.84	32.93	54.79	33.39	29.51	39.62	48.23	**65.00±0.10**
S07	50.54	95.35	87.07	93.80	93.48	94.72	96.11	**98.42±0.07**
S08	73.04	81.05	78.28	81.44	80.86	81.40	81.51	**95.28±0.12**
S09	60.38	73.88	67.70	72.10	67.34	69.61	76.27	**94.33±0.13**
S10	58.51	70.97	70.65	69.44	56.05	67.72	72.65	**92.52±0.04**
S11	82.18	90.15	86.19	89.43	90.85	89.26	88.43	**93.91±0.19**
S12	82.06	89.35	85.19	89.48	91.28	89.07	88.96	**94.10±0.11**
S13	46.20	50.01	81.96	53.06	38.44	62.72	63.07	**94.88±0.12**
S14	81.52	92.99	88.68	92.08	92.28	91.70	92.93	**94.40±0.24**
S15	82.62	76.06	78.50	73.80	71.92	77.32	82.03	**97.61±0.06**
S16	13.44	26.07	44.71	26.31	26.88	31.86	**49.54**	48.48±0.37
Average	55.42	68.75	76.44	68.42	66.04	70.30	76.85	**88.64±0.09**

Table 2 examines stricter matching tolerances. The mean F1 differences from TriaGS are 11.79, 15.26, and 17.60 points at $L/2048$, $L/3072$, and $L/4096$, respectively. The F1 difference increases as the geometric match tolerance decreases on this subset.

On six Tanks and Temples scenes, Table 3 reports a mean F1 of 0.500 for HelixGau, compared with 0.468 for MILo, 0.462 for SOF, and 0.409 for TriaGS. HelixGau leads

on four scenes, whereas SOF leads on Barn and MILo on Courthouse. The scene-wise values show that the ordering varies across scenes.

Figures 3 and 4 compare extracted geometry, and Figure 5 shows the Tanks and Temples reconstructions. They illustrate boundary and surface-detail differences in the displayed examples. The experiments do not isolate camera trajectory, scene size, and texture as separate factors, so the difference between datasets cannot be attributed to any one of them. Differences in extraction and ground-truth coverage across datasets also preclude directly comparing the numerical magnitude of their metrics.

Table 2 BlendedMVS F1 at three common normalized tolerances. HelixGau reports four-run mean ± standard deviation; baselines use one run

Method	$\delta = L/2048$	$\delta = L/3072$	$\delta = L/4096$
2DGS [9]	55.42	45.50	38.62
GOF [27]	68.75	57.43	49.88
PGSR+ [3]	76.44	65.66	56.48
RaDe-GS [28]	68.42	57.20	49.42
SOF [15]	66.04	55.24	47.68
MILo [7]	70.30	58.38	50.16
TriaGS [19]	76.85	65.84	56.68
HelixGau	**88.64±0.09**	**81.10±0.07**	**74.28±0.03**

Table 3 Tanks and Temples F1 after the reported benchmark alignment (0 to 1; higher is better). Results are single-run measurements

Method	Barn	Caterpillar	Courthouse	Ignatius	Meetingroom	Truck	Mean
SOF [15]	**0.613**	0.390	0.241	0.719	0.266	0.540	0.462
MILo [7]	0.563	0.381	**0.287**	0.742	0.247	0.590	0.468
TriaGS [19]	0.468	0.332	0.271	0.626	0.233	0.524	0.409
HelixGau	0.509	**0.442**	0.253	**0.800**	**0.377**	**0.618**	**0.500**

4.4 Component ablations and alternative designs

Table 4 reports the BlendedMVS ablations. **Base** uses the Gaussian representation without stereo depth supervision. **+LocalGeo** adds PGSR-style local multi-view consistency. **+Multi** instead uses stereo depth after geometric consistency checks, without the monocular correlation test. **+Mono & Multi** adds depth filtering to the stereo depth supervision. **−Decider** combines local consistency with filtered stereo depth, without depth updates. **Full** additionally enables depth updates. The rows therefore include complementary branches, not a single cumulative sequence.

At $L/2048$, stereo depth supervision gives F1 84.16. Depth filtering raises it to 85.35. Combining filtered stereo depth with local consistency gives F1 88.31 without depth updates. Enabling depth updates gives 88.64, a difference of 0.33 points. The

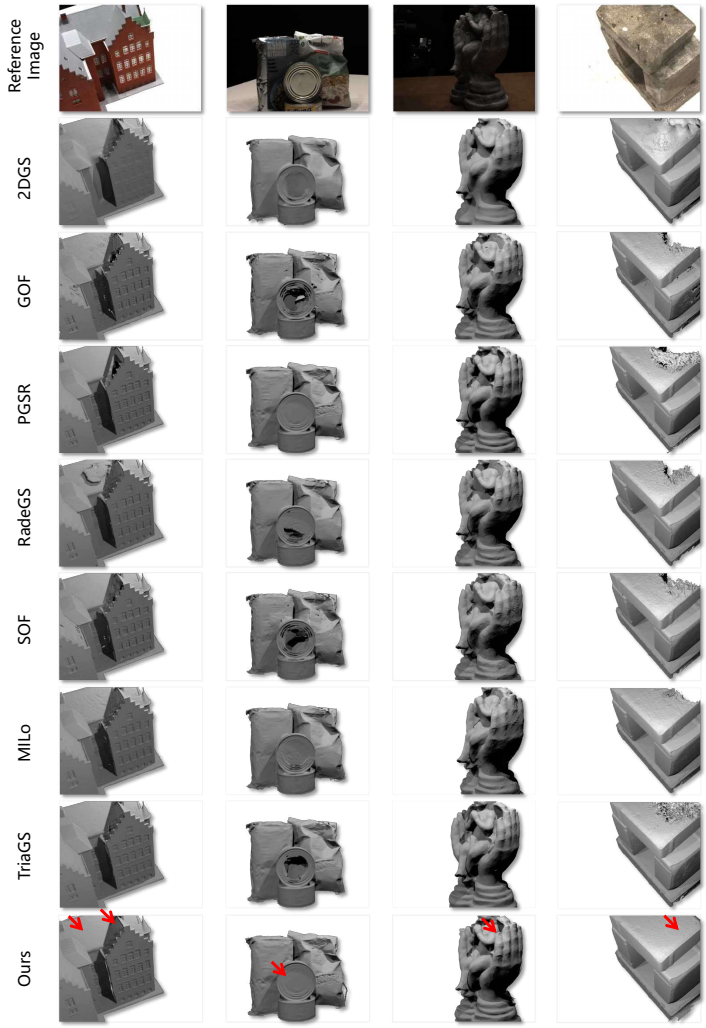


Fig. 3 Qualitative surface comparisons on DTU. Insets show local geometric differences in the displayed examples

corresponding differences at the two stricter tolerances are 0.32 and 0.85. The additional change from depth updates is smaller than the change from adding stereo depth supervision. Figures 6 and 7 show local differences between the variants.

The alternative designs in Table 5 clarify the role of monocular information. Affine-aligned monocular supervision and unmasked direct monocular supervision reduce F1 by 3.93 and 4.19 points relative to the default, respectively. Soft uncertainty fusion reduces it by 0.56, and depth-only updates by 0.17. The binary filtering configuration has higher F1 than these tested alternatives. The estimator substitution in Appendix D yields a mean change of +0.04 points with Depth Anything 3.

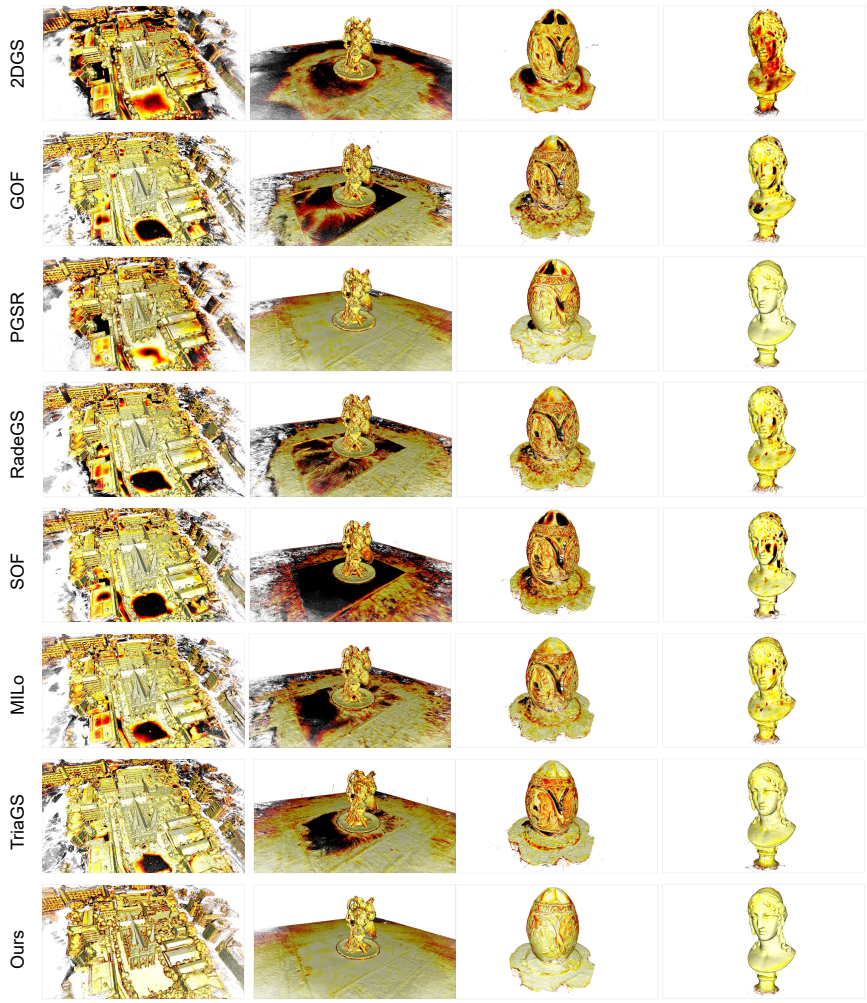


Fig. 4 Qualitative surface comparisons on BlendedMVS. Black regions indicate large reconstruction errors in the visualization

Table 4 BlendedMVS component ablations (F1, 0 to 100). Full reports four-run mean ± standard deviation; other variants use one run. The variant definitions are given in Section 4.4

Variant	$\delta = L/2048$	$\delta = L/3072$	$\delta = L/4096$
Base	62.57	52.54	44.93
+ LocalGeo	76.44	65.66	56.48
+ Multi	84.16	76.57	69.86
+ Mono & Multi	85.35	77.95	71.29
− Decider	88.31	80.78	73.43
Full (HelixGau)	**88.64±0.09**	**81.10±0.07**	**74.28±0.03**



Fig. 5 HelixGau reconstructions on Tanks and Temples, including Barn, Caterpillar, Ignatius, and Meetingroom

Table 5 Alternative depth supervision and update designs on BlendedMVS. Values are F1-point changes relative to the default configuration

Variant	Δ F1
Affine-aligned mono supervision	−3.93
Direct mono supervision (no mask)	−4.19
Soft uncertainty fusion	−0.56
Depth-only update	−0.17

4.5 Depth errors and update-mask measurements

Table 6 reports diagnostics from the training trajectories of all 16 BlendedMVS scenes, aggregated first over views and then over scenes. A depth is consistent with ground truth (GT) when $|d - d_{\mathrm{GT}}| \leq \delta_i$. The eligible set contains the pixels evaluated by the depth update comparison. Coverage is the fraction of eligible pixels updated; precision is the GT-consistent fraction of updated pixels; recall is the updated fraction of GT-consistent eligible pixels. Region precision is the GT-consistent fraction of pixels selected for depth supervision. Stereo depth errors are measured in units of δ_i.

At the measured depth updates, update coverage is 0.109 to 0.124 and precision increases from 0.400 to 0.676. The stereo depth error in the supervision region decreases from 4.14 initially to 2.70 at 25k iterations, with a plateau of 3.50 at 10k

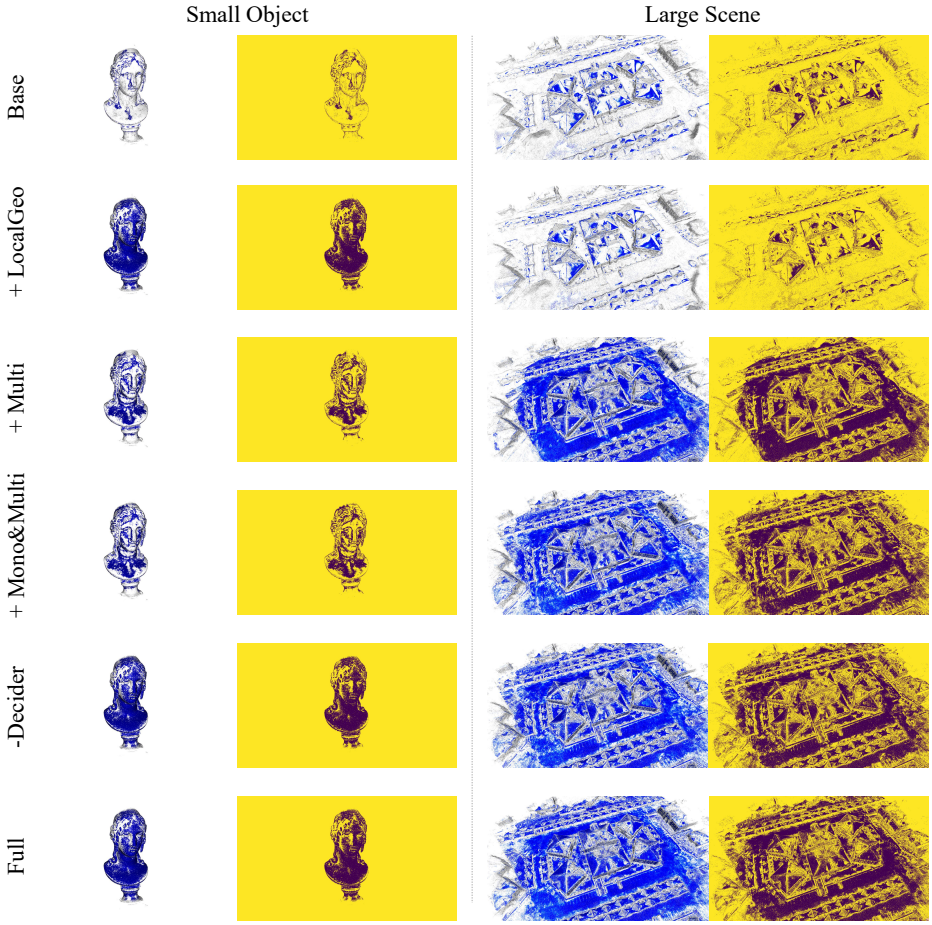


Fig. 6 Component ablation visualizations. For each scene, paired columns show point precision and the reconstruction against a contrasting background

and 15k. Pixels excluded by the filter have higher stereo depth error, ranging from 45.7 initially to 26.1 at 25k. At 25,000 iterations, the mean fraction of updated pixels outside the geometric tolerance is 32.4%. The diagnostics cover init, 5k, 10k, 15k, and 25k; they do not constitute a complete record of every scheduled depth update.

4.6 Sensitivity and appearance fit

Table 7 reports one-parameter changes around the default configuration. The tested correlation thresholds, NCC cost parameters, window sizes, and update interval change F1 by at most 0.17 points. The depth-loss weight has a larger influence. Weights 0.25, 0.5, and 2.0 change F1 by -1.26, -0.62, and $+0.32$, respectively. Thus stability over the tested gating settings should not be extended to all hyperparameters. The main experiments use depth-loss weight 1 across scenes.

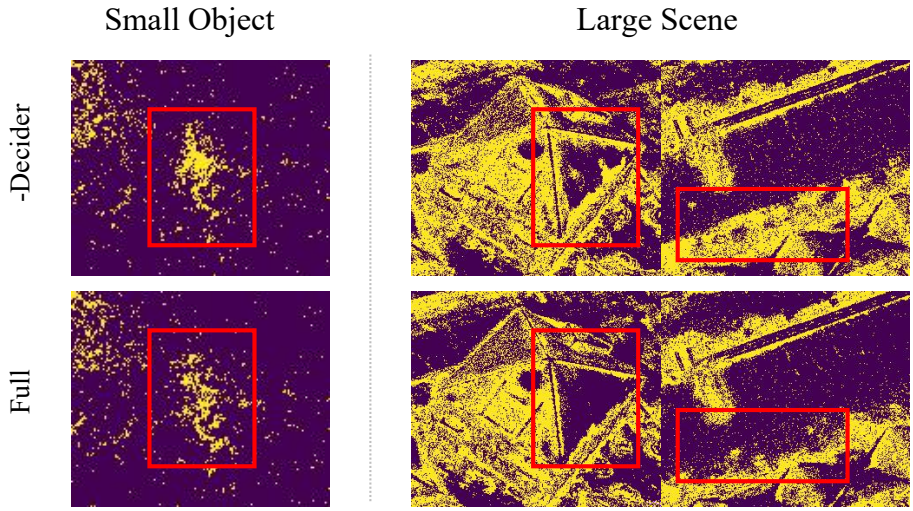


Fig. 7 Surface detail with and without depth updates. Brighter regions denote larger reconstruction errors, and boxes mark detail comparisons

Table 6 Stereo depth errors and update-mask measurements on 16 BlendedMVS scenes at the reported checkpoints. Stereo depth errors are in units of $\delta_i = L_i/2048$. Coverage, precision, recall, and region precision are defined in Section 4.5. At initialization, n/a (not applicable) indicates that no depth update has occurred

Metric	init	5k	10k	15k	25k
Coverage	n/a	0.118	0.124	0.109	0.117
Precision	n/a	0.400	0.660	0.669	0.676
Recall	n/a	0.185	0.126	0.111	0.116
Region precision	0.706	0.730	0.733	0.733	0.744
Stereo depth err. (selected)	4.14	3.64	3.50	3.50	2.70
Stereo depth err. (excluded)	45.7	34.1	30.8	30.1	26.1

Table 8 reports image fitting on the *training views*, using rendered images and bicubic-resized reference images at the same resolution. HelixGau yields peak signal-to-noise ratio (PSNR) 30.63, SSIM 0.806, and learned perceptual image patch similarity (LPIPS) 0.067. HelixGau has the second-lowest LPIPS, and TriaGS has the highest PSNR and SSIM and the lowest LPIPS among the compared methods. These measurements quantify fit to the observed views; no held-out-view rendering conclusion is drawn from this table.

4.7 Computation and memory

The cost measurements in Table 9 use a single RTX 3090 and 30,000 iterations. HelixGau takes 0.67 h per BlendedMVS scene on average, compared with 0.59 h for PGSR+ and 0.67 h for SOF. It is faster than GOF, MILo, and TriaGS in these runs, and

Table 7 BlendedMVS sensitivity results. Values are F1-point changes relative to the default configuration

Hyperparameter	Default	Swept values	Δ F1
Correlation threshold τ_c	0.3	0.15 / 0.2 / 0.4	−0.03 / −0.02 / −0.01
NCC cost parameter	0.4	0.2 / 0.3 / 0.5	−0.17 / −0.02 / 0.00
Correlation window size s	7	9 / 11 / 13 / 15	−0.08 / 0.00 / −0.07 / −0.07
Update interval	5k	every 2.5k	−0.01
Depth-loss weight λ_{depth}	1.0	0.25 / 0.5 / 2.0	−1.26 / −0.62 / +0.32

Table 8 Training-view appearance metrics on 16 BlendedMVS scenes. Reference images are bicubic-resized to the rendering resolution. These are observed-view fit metrics

Method	PSNR↑	SSIM↑	LPIPS↓
TriaGS [19]	**31.99**	**0.842**	**0.060**
MILo [7]	31.79	0.829	0.070
RaDe-GS [28]	31.42	0.821	0.078
GOF [27]	31.35	0.820	0.074
SOF [15]	30.97	0.814	0.071
PGSR+ [3]	30.46	0.788	0.080
2DGS [9]	30.44	0.775	0.104
HelixGau	30.63	0.806	0.067

slower than 2DGS and RaDe-GS. Its mean Gaussian count is 1.29M. Primitive count alone does not determine runtime because projection, blending, auxiliary matching, and memory traffic also contribute.

The reported HelixGau time includes stereo depth estimation and depth updates. One-time monocular inference, plane initialization, and initial NCC computation take 2 to 4 min per scene. Training ranges from 0.39 to 0.94 h on BlendedMVS and averages 0.73 h on DTU (range 0.67 to 0.80). Average peak allocated/reserved GPU memory is 5.3/6.2 GB on BlendedMVS, with worst case 12.5/13.9 GB for a scene with 384 views and 3.54M Gaussians. DTU averages 2.2/3.2 GB, with worst case 2.5/3.5 GB. A depth update adds a transient 0.04 to 0.46 GB. The reported rendering measurement is 46.9 frames per second at 777×581 with 1.22M Gaussians; it is a particular operating point, not a throughput guarantee for all scenes.

Extraction takes 58 to 103 s per DTU scene and about 6 min per BlendedMVS scene on average (range 0.2 to 27 min). These extraction costs are separate from Gaussian training. For fixed numbers of source views, PatchMatch iterations, candidates, and fixed patch size, the added matching and correlation work is linear in the processed pixel count. Across the full run, depth update cost also scales with the number of depth updates. This conditional complexity statement does not remove dependence on image count or resolution.

Table 9 Per-scene computational averages on 16 BlendedMVS scenes. Training uses one RTX 3090 and 30,000 iterations. Time includes stereo depth estimation and depth updates, with final surface extraction reported separately

	2DGS [9]	GOF [27]	PGSR+ [3]	RaDe-GS [28]	SOF [15]	MILo [7]	TriaGS [19]	HelixGau
#Gaussians (M)	0.84	1.40	1.45	1.28	1.31	1.20	1.61	1.29
Time (h)	0.33	1.72	0.59	0.42	0.67	2.14	0.83	0.67

5 Discussion

5.1 Depth supervision in spatial data acquisition

HelixGau filters stereo depth before it enters the Gaussian depth loss and revises the stereo planes using rendered planes. These operations address different parts of depth supervision. In the BlendedMVS ablations, depth filtering changes F1 from 84.16 to 85.35. With local consistency also enabled, depth updates change F1 from 88.31 to 88.64. The measurements distinguish the effect of filtering from the smaller additional change associated with updates.

For GIS data preparation, the extracted surface is an intermediate geometric product that can be registered and organized with geographic information. The method takes images, cameras, and sparse points as inputs and does not require semantic object labels. Its filtering and update rules are expressed through image patches and plane geometry. The experiments measure the resulting surface geometry under the evaluated camera and dataset configurations.

5.2 Dependence on depth estimates and image evidence

Monocular and stereo depth can contain similar errors, which can pass depth filtering. Correlation is also uninformative in patches with little depth variation. Replacing Depth Anything V2 with Depth Anything 3 changes mean F1 by +0.04 points in the tested configuration. This measurement concerns the two tested estimators and does not characterize all monocular errors.

Photometric correlation can favor incorrect planes under repetitive texture, occlusion, or reflective appearance. The update precision of 0.676 at 25,000 iterations shows that some updated pixels still exceed the geometric error tolerance. Errors can therefore enter the stereo depth maps through updates and remain after filtering. Distant regions and sky are difficult to represent with finite plane estimates. Surface topology and fusion artifacts also depend on the extraction procedure, which is outside the Gaussian optimization objective.

5.3 Evaluation coverage

BlendedMVS results describe the 16 evaluated scenes after four exclusions. Without the excluded scene identifiers and failure records, the results do not provide a failure rate for the initial 20-scene sample. Four runs characterize the full method on this subset. The baselines, other ablation variants, DTU, and Tanks and Temples have single-run results. Their variability and paired statistical differences are therefore not measured.

Computation is measured on one graphics processing unit platform and the evaluated image resolutions. The measurements do not determine runtime for unrestricted image collections. Source code, configuration files, and evaluation scripts are planned for release upon publication.

6 Related work

6.1 Surface reconstruction for spatial data acquisition

Image-based reconstruction supplies geometric observations for the preparation of 3D spatial models. Wang et al. [22] review the relation between SfM, multi-view stereo, surface reconstruction, and learned scene representations from a photogrammetric perspective. Their review distinguishes geometric accuracy from visual appearance. This distinction matters when a representation optimized through image rendering is subsequently converted into a surface. The rendered images and the extracted geometry are different outputs, and measurements of image fit alone do not determine surface accuracy.

Registration with geographic information is a further operation on the acquired model. Han et al. [8] investigate a distributed framework for placing 3D models in GIS through feature matching with remote sensing images. Their work concerns spatial reference and model placement. HelixGau addresses the estimation of surface geometry before this stage. It takes overlapping images, camera estimates, and sparse points as inputs, and produces an optimized Gaussian representation and extracted geometry. This relation locates depth supervision within spatial data acquisition without making geographic registration part of the reconstruction algorithm.

Gaussian representations have also been adapted to geographic imagery and scene organization. Aira et al. [1] use Gaussian splatting for satellite image photogrammetry, adapting the camera and rendering formulation to the observations. JumpingGS controls representation detail for aerial scene rendering [14], while Octree-GS organizes primitives at multiple levels of detail [16]. These methods concern the observation model or the organization of the representation. The question examined here concerns the stereo depth targets used to optimize a surface representation with perspective cameras.

6.2 Gaussian surface reconstruction

Three-dimensional Gaussian splatting [11] optimizes explicit primitives through differentiable image rendering. Surface reconstruction methods modify this representation or its rendering to constrain geometry. 2DGS uses oriented planar primitives [9], and Gaussian surfels flatten primitives for surface reconstruction [5]. PGSR renders plane distances and normals to compute depth and incorporates consistency across views [3]. HelixGau uses the PGSR representation, depth rendering, and local consistency term. Its depth filtering and depth updates operate alongside these components, rather than defining another Gaussian primitive.

Other methods define surfaces through fields or place extraction inside optimization. GOF identifies a surface as a level set of an opacity field and extracts a mesh

using tetrahedral grids induced by Gaussian primitives [27]. SOF introduces hierarchical resorting, a Gaussian depth formulation, and an opacity-field regularizer [15]. MILo differentiably constructs a mesh during training and couples it to the Gaussian representation [7]. These choices determine how Gaussian parameters correspond to an extracted surface. HelixGau instead retains extraction after optimization and changes the stereo depth used in the optimization objective.

The number and distribution of primitives also affect reconstruction cost. Mini-Splatting studies Gaussian densification and simplification [6]. HelixGau uses pruning based on the contribution of each primitive, as specified in Section 3. Pruning changes the Gaussian set, whereas depth filtering and depth updates determine how stereo estimates are used during optimization. Changes to the Gaussian set can also affect the rendered plane candidates. The computational measurements therefore report primitive count alongside runtime and memory, while the depth-supervision ablations compare the filtering and update operations.

6.3 Depth supervision and consistency across views

Learned depth and geometric priors provide information beyond color matching. Depth Anything V2 estimates depth from an image [25], while Depth Anything 3 predicts depth from one or more images [13]. DN-Splatter incorporates depth and normal priors into Gaussian reconstruction [20], and IndoorGS uses geometric cues including line structure [17]. Such estimates can enter a reconstruction objective as numerical targets. HelixGau assigns monocular depth a different role. Local correlation with stereo depth selects pixels, and stereo estimates supply the numerical targets at those pixels. The estimator-substitution experiment tests this use of monocular depth with two models.

Stereo estimation connects these targets to correspondences and camera geometry. PatchMatch propagates candidate correspondences and searches through random perturbations [2]. Xu et al. [24] combine multi-scale geometric consistency and planar information in multi-view stereo. The stereo estimator in HelixGau uses a custom CUDA implementation based on their method. Its plane map supplies depths for supervision and accepts rendered planes as additional candidates during optimization. The candidate evaluation remains a stereo matching operation outside gradient optimization of the Gaussian parameters.

Consistency can also be imposed on the reconstruction itself. PGSR evaluates local photometric consistency across views [3], while TriaGS constrains Gaussian geometry using triangulated points [19]. HelixGau retains the PGSR local consistency loss and separately compares patches induced by rendered and stereo planes using NCC. This comparison controls stereo plane updates. A plane with higher photometric correlation can still have geometric error, and agreement between monocular and stereo depth can retain shared errors. The depth-error and update-mask measurements therefore assess intermediate estimates as well as the final surface.

6.4 Direct prediction of visual geometry

The visual geometry grounded transformer (VGGT) predicts cameras and scene geometry directly from images [21]. Pi3 uses a permutation-equivariant model to predict

visual geometry without assigning one input image a privileged reference role [23]. These models obtain geometric estimates through inference, whereas HelixGau optimizes a Gaussian representation for each scene using supplied cameras. The distinction affects both the available inputs and the computation included in a comparison.

Appendix C reports the Pi3 results with the camera and alignment procedures stated there. Its camera alignment and subsequent geometric alignment are part of that evaluation protocol. The measurements compare the resulting point clouds under those conditions. They do not isolate the effect of depth filtering from differences in camera estimation, representation, or alignment. The component ablations provide the corresponding evidence for depth filtering and depth updates within the Gaussian reconstruction procedure.

7 Conclusion

HelixGau uses local correlation with monocular depth to filter stereo depth for Gaussian surface reconstruction. It uses rendered planes with higher photometric correlation across views to update stereo planes. The updated depth maps are filtered before supervision, and surfaces are extracted after optimization.

Mean F1 is 88.64 ± 0.09 on 16 BlendedMVS scenes and 0.500 on six Tanks and Temples scenes. DTU overall error is 0.505 mm, compared with PGSR+ at 0.504 mm. Component ablations show changes in F1 from both depth filtering and depth updates. These results measure how adaptive depth supervision affects surface geometry in image-based spatial data acquisition. Shared depth errors, photometric ambiguity, and surface extraction remain sources of geometric error.

Statements and Declarations

Funding. This research was supported by the Key Research and Development Program for Technological Innovation Project of Hubei Province (Grant No. 2025BAB020).

Competing interests. The authors declare no competing interests.

Data availability. The study uses the public DTU [10], BlendedMVS [26], and Tanks and Temples [12] datasets. The evaluated BlendedMVS scene identifiers are listed in Appendix A, and the reported numerical results are included in the article. This study introduces no new benchmark dataset.

Code availability. The authors plan to release the source code, configuration files, and evaluation scripts upon publication.

Author contributions. Xiangqian Shen wrote the manuscript and performed the main experiments and their analysis. Yusen Wang guided the implementation details, especially the CUDA modules. Tuo Cao contributed to the key experimental design and guided the writing. Jiongming Qin assisted with the experiments and result analysis. Chunxia Xiao supervised the research, provided the research direction and specific ideas, and supplied the computing resources.

Use of generative AI. OpenAI Codex was used to assist with language revision of the manuscript.

A Evaluated scene identifiers

The DTU scans are 24, 37, 40, 55, 63, 65, 69, 83, 97, 105, 106, 110, 114, 118, and 122. Table 10 maps the short BlendedMVS identifiers used in the result tables to their complete dataset identifiers. These are the 16 retained scenes evaluated in this study.

Table 10 Retained BlendedMVS scenes and short identifiers

ID	Complete dataset scene identifier
S01	`58eaf1513353456af3a1682a`
S02	`59056e6760bb961de55f3501`
S03	`59e864b2a9e91f2c5529325f`
S04	`59f363a8b45be22330016cad`
S05	`5a3ca9cb270f0e3f14d0eddb`
S06	`5a6464143d809f1d8208c43c`
S07	`5af02e904c8216544b4ab5a2`
S08	`5b08286b2775267d5b0634ba`
S09	`5ba75d79d76ffa2c86cf2f05`
S10	`5bb7a08aea1cfa39f1a947ab`
S11	`5bf26cbbd43923194854b270`
S12	`5bfc9d5aec61ca1dd69132a2`
S13	`5bfd0f32ec61ca1dd69dc77b`
S14	`5bfe5ae0fe0ea555e6a969ca`
S15	`5c1892f726173c3a09ea9aeb`
S16	`5c1af2e2bee9a723c963d019`

B Detailed geometric results

Tables 11 to 13 report the DTU per-scene d2s, s2d, and overall values. Tables 14 and 15 provide BlendedMVS precision and recall to accompany the F1 results in Table 1. All methods are evaluated under the common within-dataset protocols described in Section 4.1. Best values are bold and underlined; second-best values are underlined.

Table 11 DTU d2s distances (mm; lower is better). All methods use single-run results

Scene	2DGS	GOF	PGSR+	RaDe-GS	SOF	MILo	TriaGS	HelixGau
scan24	0.822	0.738	**0.333**	0.640	0.602	0.524	0.400	0.338
scan37	1.143	1.287	0.625	1.143	1.026	1.188	1.037	**0.591**
scan40	1.121	0.619	**0.348**	0.448	0.489	0.439	0.444	0.352
scan55	0.523	0.456	**0.299**	0.348	0.417	0.420	0.362	0.301
scan63	1.577	1.460	**0.796**	0.912	1.270	0.928	0.915	0.809
scan65	1.028	0.875	0.566	0.781	0.926	0.793	0.727	**0.553**
scan69	0.768	1.151	**0.529**	0.791	0.878	0.816	0.674	0.529
scan83	0.923	0.852	**0.710**	0.861	1.024	0.871	0.790	0.735
scan97	1.089	1.267	**0.673**	1.038	1.366	1.050	0.939	0.678
scan105	0.656	0.730	**0.515**	0.630	0.714	0.678	0.587	0.516
scan106	0.626	0.579	**0.404**	0.550	0.724	0.648	0.510	0.430
scan110	1.206	1.993	**0.649**	1.077	1.495	1.073	0.693	0.682
scan114	0.389	0.858	0.294	0.315	0.814	0.361	0.314	**0.269**
scan118	0.862	0.581	**0.360**	0.590	0.552	0.683	0.441	0.372
scan122	0.507	0.466	0.338	0.415	0.465	0.515	0.391	**0.327**
Average	0.883	0.927	**0.496**	0.703	0.851	0.733	0.615	0.499

Table 12 DTU s2d distances (mm; lower is better). All methods use single-run results

Scene	2DGS	GOF	PGSR+	RaDe-GS	SOF	MILo	TriaGS	HelixGau
scan24	1.007	0.548	0.469	0.489	0.614	0.555	0.457	**0.425**
scan37	1.291	0.745	0.560	0.721	0.734	0.782	0.729	**0.557**
scan40	1.114	0.468	0.411	0.433	0.486	0.450	**0.407**	0.409
scan55	0.532	0.446	**0.373**	0.397	0.422	0.444	0.377	0.394
scan63	1.042	0.752	**0.378**	0.432	0.843	0.518	0.381	0.379
scan65	1.144	0.789	0.576	0.689	0.859	0.768	0.646	**0.553**
scan69	0.922	0.774	0.509	0.723	0.754	0.773	0.552	**0.508**
scan83	1.190	0.885	**0.763**	1.063	1.062	1.033	0.829	0.772
scan97	1.135	0.920	**0.676**	0.883	1.366	0.926	0.778	0.680
scan105	1.019	0.919	**0.715**	0.897	1.004	0.924	0.746	0.715
scan106	0.754	0.689	**0.508**	0.650	0.749	0.722	0.534	0.509
scan110	1.350	0.911	0.455	0.896	0.844	0.975	0.519	**0.454**
scan114	0.572	0.499	**0.431**	0.454	0.988	0.515	0.449	0.438
scan118	1.405	0.715	**0.449**	0.858	0.786	0.943	0.491	0.453
scan122	0.743	0.575	**0.407**	0.542	0.608	0.613	0.456	0.421
Average	1.015	0.709	0.512	0.675	0.808	0.729	0.557	**0.511**

Table 13 DTU overall distances (mm; lower is better). All methods use single-run results

Scene	2DGS	GOF	PGSR+	RaDe-GS	SOF	MILo	TriaGS	HelixGau
scan24	0.914	0.643	0.401	0.565	0.608	0.540	0.428	**0.382**
scan37	1.217	1.016	0.593	0.932	0.880	0.985	0.883	**0.574**
scan40	1.117	0.544	**0.379**	0.440	0.488	0.444	0.426	0.380
scan55	0.528	0.451	**0.336**	0.373	0.419	0.432	0.370	0.348
scan63	1.309	1.106	**0.587**	0.672	1.056	0.723	0.648	0.594
scan65	1.086	0.832	0.571	0.735	0.892	0.780	0.686	**0.553**
scan69	0.845	0.963	0.519	0.757	0.816	0.795	0.613	**0.519**
scan83	1.057	0.868	**0.737**	0.962	1.043	0.952	0.810	0.754
scan97	1.112	1.094	**0.675**	0.960	1.366	0.988	0.858	0.679
scan105	0.838	0.825	**0.615**	0.763	0.859	0.801	0.667	0.616
scan106	0.690	0.634	**0.456**	0.600	0.736	0.685	0.522	0.469
scan110	1.278	1.452	**0.552**	0.986	1.170	1.024	0.606	0.568
scan114	0.481	0.678	0.362	0.384	0.901	0.438	0.381	**0.353**
scan118	1.134	0.648	**0.404**	0.724	0.669	0.813	0.466	0.412
scan122	0.625	0.520	**0.372**	0.479	0.537	0.564	0.424	0.374
Average	0.949	0.818	**0.504**	0.689	0.829	0.731	0.586	0.505

Table 14 BlendedMVS precision at $\delta_i = L_i/2048$ (0 to 100; higher is better). HelixGau reports four-run mean ± standard deviation; baselines use one run

Scene	2DGS	GOF	PGSR+	RaDe-GS	SOF	MILo	TriaGS	HelixGau
S01	73.65	88.99	85.17	86.17	87.51	87.97	86.53	**90.34±0.37**
S02	28.72	48.45	69.29	41.37	49.65	42.80	45.80	**89.70±0.35**
S03	20.26	71.98	73.48	73.16	77.51	69.49	86.36	**94.15±0.66**
S04	24.11	52.78	92.19	56.11	64.22	57.11	83.84	**92.75±0.14**
S05	69.78	68.99	**97.47**	68.23	50.65	76.39	85.94	97.07±0.41
S06	21.46	24.18	43.15	24.24	21.47	29.35	35.06	**52.12±0.14**
S07	42.33	93.84	85.10	92.45	91.25	93.05	94.25	**97.28±0.15**
S08	70.14	76.46	73.71	77.55	76.29	76.15	74.46	**94.88±0.35**
S09	49.65	62.66	58.48	60.84	54.05	57.50	64.13	**91.26±0.27**
S10	56.71	68.71	67.86	66.59	52.90	64.94	65.86	**89.49±0.16**
S11	78.07	86.25	82.17	85.98	87.58	85.79	83.19	**91.15±0.40**
S12	76.72	84.44	79.47	84.87	87.27	84.37	83.30	**91.26±0.28**
S13	42.31	44.35	80.11	47.04	35.08	58.02	54.65	**93.54±0.31**
S14	79.20	**91.65**	87.33	90.53	91.26	90.30	90.36	91.47±0.48
S15	76.89	68.44	71.00	66.38	63.23	69.10	72.30	**96.25±0.15**
S16	32.68	56.24	87.61	53.93	54.11	64.30	81.09	**91.60±0.04**
Average	52.67	68.03	77.10	67.21	65.25	69.16	74.19	**90.27±0.28**

Table 15 BlendedMVS recall at $\delta_i = L_i/2048$ (0 to 100; higher is better). HelixGau reports four-run mean ± standard deviation; baselines use one run

Scene	2DGS	GOF	PGSR+	RaDe-GS	SOF	MILo	TriaGS	HelixGau
S01	79.64	92.96	89.13	90.38	89.21	92.60	95.80	**96.69±0.06**
S02	38.33	66.07	87.56	52.38	72.25	59.12	64.21	**98.64±0.07**
S03	10.32	35.95	30.86	40.38	41.99	36.89	54.00	**55.56±0.04**
S04	29.94	55.83	93.90	61.48	64.31	61.44	90.78	**94.68±0.10**
S05	85.08	75.55	99.37	83.65	52.86	89.67	98.53	**99.75±0.01**
S06	48.96	51.60	75.01	53.64	47.18	60.93	77.21	**86.33±0.37**
S07	62.69	96.91	89.13	95.20	95.83	96.46	98.05	**99.59±0.02**
S08	76.19	86.23	83.44	85.73	86.00	87.43	90.04	**95.69±0.13**
S09	77.04	90.00	80.39	88.50	89.28	88.19	94.07	**97.62±0.03**
S10	60.43	73.38	73.69	72.56	59.60	70.75	81.01	**95.77±0.11**
S11	86.73	94.42	90.63	93.17	94.37	93.02	94.37	**96.84±0.05**
S12	88.20	94.87	91.81	94.61	95.68	94.31	95.45	**97.12±0.07**
S13	50.88	57.32	83.89	60.87	42.51	68.25	74.56	**96.27±0.09**
S14	83.97	94.36	90.08	93.68	93.32	93.13	95.64	**97.53±0.05**
S15	89.26	85.58	87.77	83.09	83.39	87.78	94.78	**99.00±0.04**
S16	8.46	16.97	30.02	17.40	17.88	21.18	**35.66**	32.96±0.34
Average	61.01	73.00	79.79	72.92	70.35	75.07	83.38	**90.00±0.09**

C Comparison with feed-forward reconstruction

The Pi3 [23] experiment evaluates predicted point maps after conversion to point clouds. Pi3 estimates its own cameras, whereas the optimization-based methods use supplied cameras. Its outputs are aligned using camera-center Umeyama alignment followed by iterative closest point refinement. The optimization-based outputs do not receive that additional alignment. Consequently, this is a comparison under different camera and alignment assumptions.

Table 16 reports Pi3 DTU overall error 1.82 mm (d2s 2.61 mm and s2d 1.02 mm) and BlendedMVS F1 50.17, alongside the HelixGau values. The displayed Pi3 reconstructions in Figures 8 and 9 illustrate recovered scene layout and remaining surface noise. The values describe these models under the stated camera and alignment procedures.

Table 16 Pi3 comparison under the camera and alignment assumptions described in Appendix C. DTU overall is in mm; BlendedMVS F1 uses $\delta_i = L_i/2048$

Method	DTU overall (mm)↓	BlendedMVS F1↑
Pi3 [23]	1.82	50.17
HelixGau	**0.505**	**88.64±0.09**

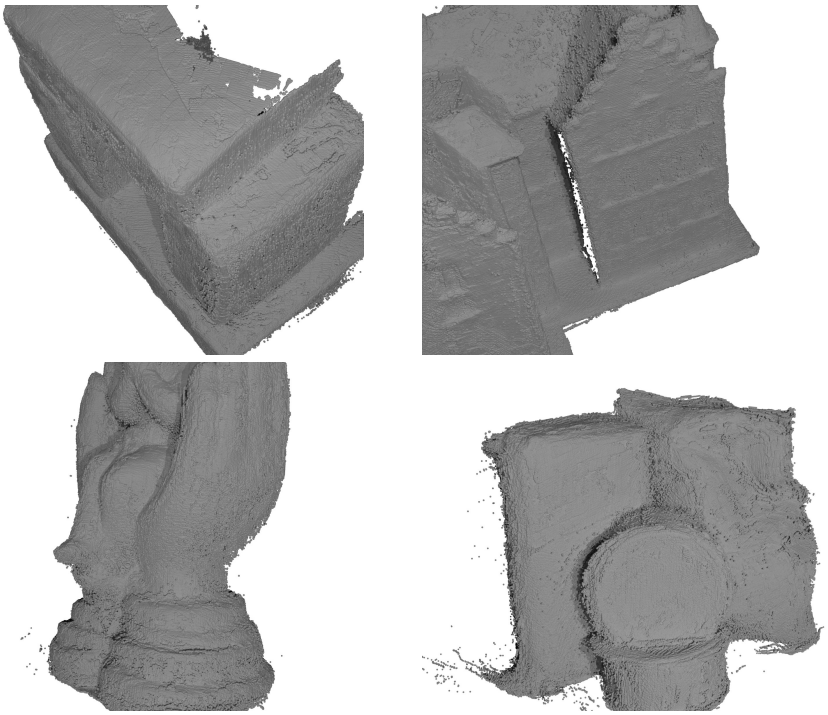


Fig. 8 Pi3 geometry visualizations on four DTU scenes

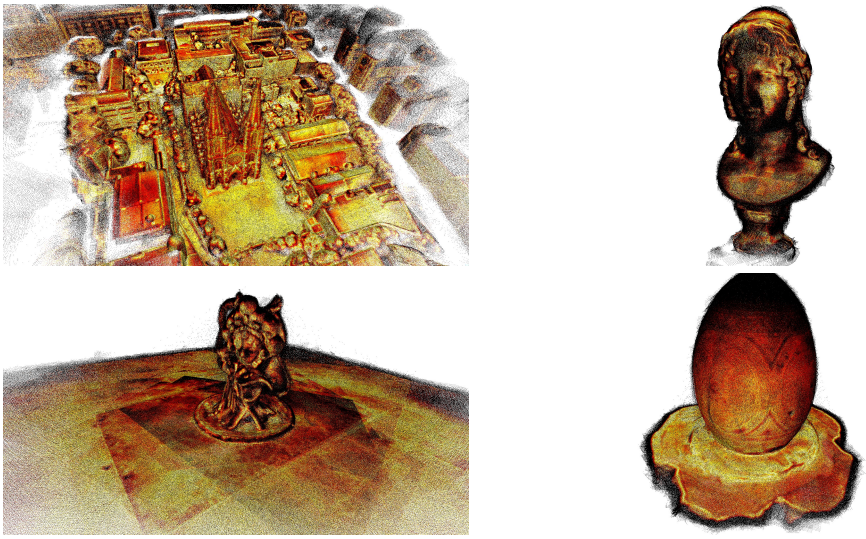


Fig. 9 Pi3 geometry visualizations on four BlendedMVS scenes

D Monocular depth estimator substitution

The Depth Anything 3 [13] substitution experiment changes only the monocular depth estimator. Table 17 reports the per-scene F1 changes on BlendedMVS. The mean change is +0.04 points, with changes ranging from −0.12 to +0.32. This experiment measures the effect of replacing the monocular estimator while keeping depth filtering and the stereo depth targets unchanged.

Table 17 F1-point changes after substituting Depth Anything 3 for the monocular depth estimator on BlendedMVS, at $\delta_i = L_i/2048$

Scene	Δ F1	Scene	Δ F1
58eaf1513353456af3a1682a	−0.01	5ba75d79d76ffa2c86cf2f05	**+0.32**
59056e6760bb961de55f3501	+0.17	5bb7a08aea1cfa39f1a947ab	+0.17
59e864b2a9e91f2c5529325f	+0.01	5bf26cbbd43923194854b270	0.00
59f363a8b45be22330016cad	+0.04	5bfc9d5aec61ca1dd69132a2	−0.03
5a3ca9cb270f0e3f14d0eddb	0.00	5bfd0f32ec61ca1dd69dc77b	+0.03
5a6464143d809f1d8208c43c	−0.12	5bfe5ae0fe0ea555e6a969ca	−0.08
5af02e904c8216544b4ab5a2	+0.02	5c1892f726173c3a09ea9aeb	+0.04
5b08286b2775267d5b0634ba	+0.10	5c1af2e2bee9a723c963d019	+0.03
Mean		**+0.04**	

References

[1] Aira LS, Facciolo G, Ehret T (2025) Gaussian Splatting for Efficient Satellite Image Photogrammetry. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 5959–5969, https://doi.org/10.1109/cvpr52734.2025.00559

[2] Barnes C, Shechtman E, Finkelstein A, et al (2009) PatchMatch: A randomized correspondence algorithm for structural image editing. ACM Transactions on Graphics 28(3):1–11. https://doi.org/10.1145/1531326.1531330

[3] Chen D, Li H, Ye W, et al (2025) PGSR: Planar-based gaussian splatting for efficient and high-fidelity surface reconstruction. IEEE Transactions on Visualization and Computer Graphics 31(9):6100–6111. https://doi.org/10.1109/tvcg.2024.3494046

[4] Curless B, Levoy M (1996) A volumetric method for building complex models from range images. In: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, pp 303–312, https://doi.org/10.1145/237170.237269

[5] Dai P, Xu J, Xie W, et al (2024) High-quality surface reconstruction using gaussian surfels. In: ACM SIGGRAPH Conference Proceedings, pp 1–11, https://doi.org/10.1145/3641519.3657441

[6] Fang G, Wang B (2024) Mini-Splatting: Representing scenes with a constrained number of gaussians. In: Proceedings of the European Conference on Computer Vision, Springer, pp 165–181, https://doi.org/10.1007/978-3-031-72980-5_10

[7] Guédon A, Gomez D, Maruani N, et al (2025) MILo: Mesh-in-the-loop gaussian splatting for detailed and efficient surface reconstruction. ACM Transactions on Graphics (TOG) 44(6):1–15. https://doi.org/10.1145/3763339

[8] Han Z, Wang Y, Zhou X, et al (2026) Towards city information modeling: a distributed framework for geo-referencing 3D models in GIS. GeoInformatica 30(2):34. https://doi.org/10.1007/s10707-026-00594-y

[9] Huang B, Yu Z, Chen A, et al (2024) 2D gaussian splatting for geometrically accurate radiance fields. In: ACM SIGGRAPH Conference Proceedings, pp 1–11, https://doi.org/10.1145/3641519.3657428

[10] Jensen R, Dahl A, Vogiatzis G, et al (2014) Large scale multi-view stereopsis evaluation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, pp 406–413, https://doi.org/10.1109/cvpr.2014.59

[11] Kerbl B, Kopanas G, Leimkühler T, et al (2023) 3D Gaussian Splatting for Real-Time Radiance Field Rendering. ACM Transactions on Graphics 42(4):1–14. https://doi.org/10.1145/3592433

[12] Knapitsch A, Park J, Zhou QY, et al (2017) Tanks and Temples: Benchmarking Large-Scale Scene Reconstruction. ACM Transactions on Graphics 36(4):1–13. https://doi.org/10.1145/3072959.3073599

[13] Lin H, Chen S, Liew J, et al (2025) Depth Anything 3: Recovering the visual space from any views. arXiv preprint https://doi.org/10.48550/arXiv.2511.10647

[14] Qin J, Zhou K, Jiang Y, et al (2025) JumpingGS: Level-jump 3D Gaussian Representation for Delicate Textures in Aerial Large-scale Scene Rendering. ACM Trans Graph 44(6):1–15. https://doi.org/10.1145/3763347

[15] Radl L, Windisch F, Deixelberger T, et al (2025) SOF: Sorted opacity fields for fast unbounded surface reconstruction. In: Proceedings of the SIGGRAPH Asia 2025 Conference Papers, pp 1–11, https://doi.org/10.1145/3757377.3763933

[16] Ren K, Jiang L, Lu T, et al (2025) Octree-GS: Towards Consistent Real-time Rendering with LOD-Structured 3D Gaussians. IEEE Transactions on Pattern Analysis and Machine Intelligence https://doi.org/10.1109/tpami.2025.3568201

[17] Ruan C, Wang Y, Guan T, et al (2025) IndoorGS: Geometric Cues Guided Gaussian Splatting for Indoor Scene Reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 844–853, https://doi.org/10.1109/cvpr52734.2025.00087

[18] Schönberger JL, Frahm JM (2016) Structure-from-motion revisited. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 4104–4113, https://doi.org/10.1109/cvpr.2016.445

[19] Tran Q, Dang T (2026) TriaGS: Differentiable Triangulation-Guided Geometric Consistency for 3D Gaussian Splatting. In: The IEEE/CVF Winter Conference on Applications of Computer Vision 2026 (WACV), https://doi.org/10.48550/arXiv.2512.06269

[20] Turkulainen M, Ren X, Melekhov I, et al (2025) DN-Splatter: Depth and Normal Priors for Gaussian Splatting and Meshing. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp 2421–2431, https://doi.org/10.1109/wacv61041.2025.00241

[21] Wang J, Chen M, Karaev N, et al (2025) VGGT: Visual geometry grounded transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 5294–5306, https://doi.org/10.1109/cvpr52734.2025.00499

[22] Wang X, Wang T, Hillemann M, et al (2026) Recent Advances in Image-Based 3D Reconstruction: a Photogrammetric Perspective on Conventional and Learning-Based Techniques. PFG https://doi.org/10.1007/s41064-026-00412-y

[23] Wang Y, Zhou J, Zhu H, et al (2025) π^3: Permutation-Equivariant Visual Geometry Learning. arXiv preprint https://doi.org/10.48550/arXiv.2507.13347

[24] Xu Q, Kong W, Tao W, et al (2023) Multi-scale geometric consistency guided and planar prior assisted multi-view stereo. IEEE Transactions on Pattern Analysis and Machine Intelligence 45(4):4945–4963. https://doi.org/10.1109/TPAMI.2022.3200074

[25] Yang L, Kang B, Huang Z, et al (2024) Depth Anything V2. Advances in Neural Information Processing Systems 37:21875–21911. URL https://depth-anything-v2.github.io/

[26] Yao Y, Luo Z, Li S, et al (2020) BlendedMVS: A large-scale dataset for generalized multi-view stereo networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 1787–1796, https://doi.org/10.1109/cvpr42600.2020.00186

[27] Yu Z, Sattler T, Geiger A (2024) Gaussian opacity fields: Efficient adaptive surface reconstruction in unbounded scenes. ACM Transactions on Graphics 43(6):1–13. https://doi.org/10.1145/3687937

[28] Zhang B, Fang C, Shrestha R, et al (2024) RaDe-GS: Rasterizing Depth in Gaussian Splatting. arXiv preprint https://doi.org/10.48550/arXiv.2406.01467