

City-Level 3D Surface Reconstruction with Viewpoint Orientation Partitioning and Scene Completion

Liang Han¹, Wenyuan Zhang¹, Junsheng Zhou^{1✉}, Yu-Shen Liu^{1✉}, and Zhizhong Han²

¹ School of Software, Tsinghua University, Beijing, China
{hanl23, zhangwen21, zhou-js24}@mails.tsinghua.edu.cn,
liyushen@tsinghua.edu.cn

² Department of Computer Science, Wayne State University, Detroit, USA
h312h@wayne.edu

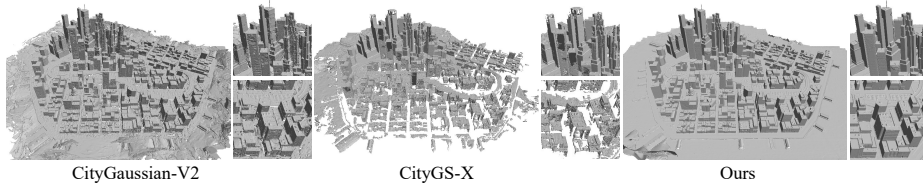


Fig. 1: Comparison of reconstruction on the MatrixCity dataset. Compared with state-of-the-art large-scale reconstruction methods CityGaussian-V2 [31] and CityGS-X [10], Our method achieves more faithful surface reconstruction.

Abstract. Multi-view 3D surface reconstruction is a longstanding challenge in computer vision. Although recent large-scale reconstruction methods based on 3D Gaussian Splatting (3DGS) achieve impressive novel-view synthesis, producing high-quality surfaces over large scenes remains difficult, due to complex geometry, long optimization, and limited memory. In this paper, we propose a novel yet simple partitioning method to efficiently and faithfully reconstruct large-scale scene surfaces. Our key insight lies in a scene partitioning method based on viewpoint orientation. This partitioning approach ensures that views with similar orientations are jointly involved for more accurate depth estimations, leading to precise surface reconstructions and balanced computation on multiple GPUs in parallel. In addition, we propose a strategy to detect and repair missing regions in the initial point cloud caused by sparse viewpoints or insufficient textures, thereby further improving the geometric quality. Extensive experiments on the GauU-Scene, MatrixCity, and UrbanScene3D datasets demonstrate that our method outperforms the state-of-the-art approaches in surface reconstruction for large-scale scenes. Project page: <https://hanl2010.github.io/VOP-GS>.

Keywords: Surface Reconstruction · Large-scale Scene · 3D Gaussian Splatting

1 Introduction

3D scene reconstruction is a fundamental task in the field of computer vision. Recently, neural implicit representation methods have made significant progress in this area. In particular, approaches based on Neural Radiance Fields (NeRF) [33] have demonstrated strong capabilities in novel view synthesis [1, 34] and geometry recovery [6, 45, 57, 63–65] using multilayer perceptrons (MLPs) to represent volumetric radiance fields. Meanwhile, 3D Gaussian Splatting (3DGS) [18], an emerging technique, achieves faster rendering speed than traditional NeRF methods. 3DGS learns a set of Gaussian primitives through differentiable rasterization, thereby improving the efficiency of geometry recovery.

Although 3DGS accelerates radiance field learning through discrete and explicit Gaussian representations, it remains limited by high memory usage and long training time [4, 26] in large-scale scene reconstruction. Currently, most large-scale reconstruction methods [26, 30, 49] adopt a divide-and-conquer strategy with 3DGS, inspired by NeRF-based approaches [43, 46] that partition the entire scene into spatially adjacent blocks. Each block is represented by a relatively small number of Gaussian primitives and trained with a subset of views, enabling efficient and parallel training on standard multi-GPU setups, which significantly improve the reconstruction quality and speed.

However, spatial partitioning strategies for scene decomposition face challenges such as occlusion, imbalance of complexity, and multiple block fusion. The visibility issue arises when a view intersects the boundary of a block. Since Gaussians outside the block are absent during training, regions near the boundary suffer from incomplete visibility, leading to artifacts in reconstruction. Although existing methods [26, 30, 49] have addressed some of these issues to varying degrees, most of them are primarily designed for novel view synthesis rather than accurate surface reconstruction. Due to inconsistencies and discontinuities at the boundaries between scene partitions, geometric estimation in these regions tends to be error-prone and unstable.

In order to reconstruct more accurate large-scale geometric surfaces, we aim to design an effective scene partitioning method that primarily relies on camera orientations, and also camera positions serving as a secondary cue. Our key insight is that, under the same optimization constraints, the accuracy of geometric reconstruction is largely affected by the overlap of viewpoints. In general, views with a large overlap are more beneficial for accurate geometry reconstruction, whereas views with little overlap or large angular differences are detrimental. This observation has been justified by many surface reconstruction methods [13, 16, 17, 35, 56], which demonstrate that selecting source views with small viewpoint differences and high overlap leads to higher-quality geometry.

Based on this observation, we propose a novel yet simple scene partitioning algorithm that performs lightweight clustering of all input views according

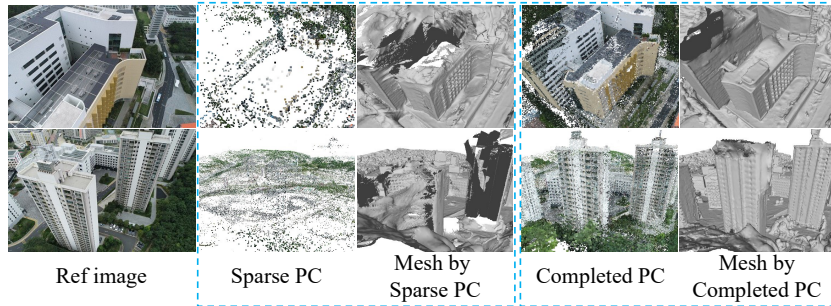


Fig. 2: Visual comparison of the point clouds (PC) and corresponding meshes before and after completion. The left column shows the reference image, the middle part presents the COLMAP point cloud and its corresponding mesh reconstruction, and the right part presents the completed point cloud and the corresponding reconstructed mesh.

to their camera orientations and positions. Our method produces an optimal partition of the scene into viewpoint-consistent groups, each of which exhibits relatively small geometry changes and thereby significantly improves our robustness in surface reconstruction. To further enhance local coherence, we introduce both geometric and photometric constraints among views within each camera group, which effectively preserves scene consistency and recovers fine-grained details at both local and global scales. Each group is then trained separately on multiple GPUs, enabling efficient and distributed learning on large scenes.

Nevertheless, it is often difficult to ensure sufficient image coverage for every region during large-scale data capture. Moreover, regions with limited textures make it difficult for Structure-from-Motion (SfM) [38] to reconstruct reliable geometry, leading to extremely sparse or even missing 3D points. Consequently, 3DGS-based reconstruction methods often suffer from poor reconstruction quality or incomplete surfaces in these regions. As shown in Fig. 2, the sparse point cloud predicted by COLMAP [39] exhibits noticeable holes with insufficient image coverage, which further causes defects on surfaces. To address this issue, we propose a scene completion strategy to detect and complete the holes in the initial point cloud. Specifically, we first identify the images corresponding to the regions with holes by analyzing the uniformity of the projected points on each image. Then, for each candidate image, we locate its neighboring views and construct image pairs, which are fed into a pretrained image feature matching network [36] to predict dense point clouds. Finally, these dense point clouds are used to refine and complete the sparse point cloud, providing a more reliable geometric initialization for large-scale surface reconstruction.

In summary, our main contributions are as follows:

- We propose a novel, simple, and effective scene-partitioning strategy that clusters views based on their camera orientations and positions. By grouping highly overlapping views with minimal geometric variation, our method

substantially improves the robustness and quality of large-scale surface reconstruction.

- We introduce a scene completion method for detecting and completing missing regions in the initial scene point cloud, which further enhances surface reconstruction quality in regions with insufficient view coverage.
- Extensive experiments on widely used large-scale scene datasets demonstrate that our method achieves state-of-the-art performance compared with existing large-scale surface reconstruction approaches.

2 Related Work

2.1 Object-Centric and Small-Scale Modeling

Neural implicit representations can be constructed from both images [1, 33, 34, 45, 57] and point clouds [20, 21, 23, 40, 48, 50, 52, 55, 62], providing a unified paradigm for 3D scene modeling. Neural Radiance Fields (NeRF) [33] have revolutionized novel view synthesis and implicit 3D reconstruction by learning a continuous volumetric radiance field from multi-view images. Many subsequent methods [1, 34, 45, 57] have improved NeRF’s efficiency, quality, or geometry extraction. Recent advances in 3D Gaussian Splatting (3DGS) [18] enable efficient rendering with explicit, differentiable Gaussian primitives, achieving high-fidelity results in compact scenes. Several extensions [2, 5, 12, 15, 22, 24, 32, 59] enhance surface fidelity by incorporating signed distance functions, normal priors, or geometry-aware regularization, yet remain constrained to relatively small scenes. However, most of these methods focus on object-centric or small-scale scenes, where dense multi-view coverage and limited scene complexity make optimization tractable.

2.2 Large-Scale Novel View Synthesis

To scale the neural rendering to large outdoor or urban environments, a series of NeRF-based methods have been proposed. Block-NeRF [43] pioneered splitting urban street scenes into multiple fixed blocks to enable scalable large scene rendering. However, since each sub-model is trained independently, inconsistencies may arise in geometry and appearance across block boundaries. MegaNeRF [44] instead leverages visibility-based geometric clustering to assign pixels to region specific sub-NeRF modules. GF-NeRF [41] further trains a global coarse model and then fine-tunes local encoders under global guidance to improve boundary consistency and geometry details. BirdNeRF [61] focuses on aerial imagery by partitioning scenes based on camera positions and applying a projection-guided fusion strategy on top of the sub-model training. Meanwhile, 3DGS-based methods such as PyGS [47] and Octree-GS [37] adopt level-of-detail (LoD) hierarchies to reduce redundant Gaussians. Both VastGaussian [26] and HUG [42] introduce visibility-aware block partitioning strategy. BlockGaussian [49] further refines block division based on structural complexity to balance the computational cost of each block. OccluGaussian [28] proposes an occlusion-aware scene division

strategy for large-scale indoor scene reconstruction to improve efficiency and resource allocation. Other methods [9, 29] improve large-scale scene rendering efficiency by optimizing the Gaussian representation or enhancing the rasterization pipeline. DoGaussian [4] performs distributed training on each scene block after partitioning. Momentum-GS [8] introduces scene momentum self-distillation to address the scalability issue of Gaussian decoders in large-scale scene reconstruction. REUrbanGS [60] designs a controllable LOD generation strategy that enables dynamic LOD selection and real-time rendering under limited computational resources. Although these methods have achieved high-quality view synthesis in city-scale scenes, they generally focus on appearance modeling rather than precise geometry reconstruction.

2.3 Large-Scale Surface Reconstruction

Compared with novel view synthesis, large-scale surface reconstruction has received considerably less attention in recent years. Some methods [46, 54] explicitly target city-scale scene geometric recovery using divide-and-conquer representations based on Signed Distance Fields (SDF), but these approaches typically come with substantial computational overhead. Within the 3DGS family, GigaGS [3] partitions the scene into multiple blocks and imposes multi-view photometric and geometric consistency constraints under an LoD framework. CityGaussian-V2 [31] extends CityGaussian [30] by introducing decomposed-gradient densification and depth regression to mitigate blurring artifacts. CityGS-X [10] further pushes efficiency by enabling voxel-level parallelism and eliminates the need for merge-and-partition operations. Despite these advances, most existing methods rely on spatial partitioning strategies and require monocular depth priors for regularization, while insufficiently exploiting cross-view consistency. As a result, achieving robust and efficient large-scale surface reconstruction remains a challenging problem.

3 Method

Overview. The overview of our method is shown in Fig. 3. We first cluster all input views based on their camera orientations and positions to form multiple training view groups, producing an optimal partition of the scene which exhibits minimal geometric inconsistency. We then analyze the density and uniformity of the projected points from the initial point cloud on each image to identify images covering holes. For these detected images, we employ a pretrained image feature matching network to predict geometric-complete point clouds, which are then used to inpaint the missing areas in the initial scene point cloud. To further improve the geometric accuracy, for each view we identify a set of source views within the same view group and impose multi-view consistency constraints to enforce local coherence.

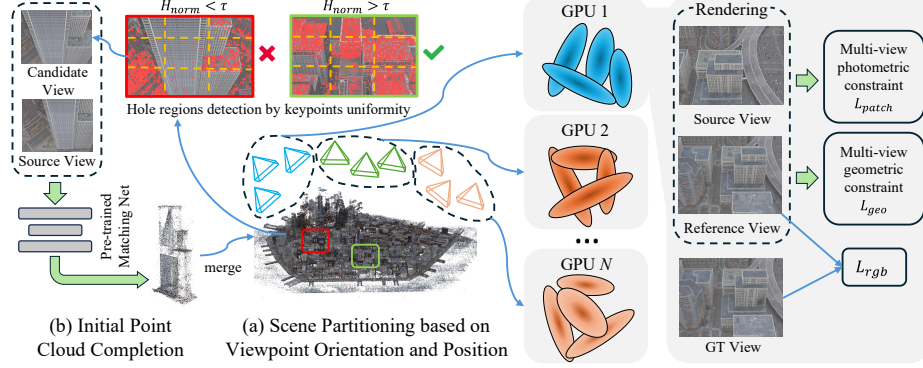


Fig. 3: The overview of our method. (a) We group all input views based on their camera orientations and positions, and we train 3DGS with multi-view consistency constraints in each group. (b) We detect regions with holes in the point cloud by analyzing the uniformity of valid keypoints, where the red points in the image indicate the valid keypoints. The corresponding images are then fed into a pretrained matching network to generate dense point clouds, which are used to refine and complete the initialization point cloud of the scene.

3.1 Scene Partitioning

Views to Clusters. To effectively organize a large amount of camera poses in a large-scale scene, we adopt a two-step clustering strategy. The first step clusters cameras based on their orientations, so that cameras with similar viewing directions are grouped together. The cosine distance is used to measure angular similarity,

$$\text{dist}(\mathbf{d}_i, \mathbf{d}_j) = 1 - \mathbf{d}_i^\top \mathbf{d}_j, \quad (1)$$

where $\mathbf{d}_i$ and $\mathbf{d}_j$ denote the normalized viewing direction vectors of the i -th and j -th cameras. DBSCAN [7] is applied with a cosine distance threshold ϵ_{dir} to obtain clusters of cameras with similar orientations.

Cluster Splitting into Sub-clusters. After the initial clustering, some groups may contain much more cameras than other groups. To balance the number of cameras in each group for keeping each cluster at a manageable size, we further subdivide any cluster that contains more than M_{max} cameras. For each one of such clusters, we estimate the required number of sub-clusters as $K = \lceil \frac{n}{M_{max}} \rceil$, where n donates the number of cameras in this cluster. We then apply K-means clustering to split the cluster into K sub-cluster according to 3D positions $\{\mathbf{p}_i\}_{i=1}^n$. This ensures that each sub-cluster contains no more than M_{max} cameras.

This two-step clustering approach produces robust, viewpoint-consistent camera groups with balanced sizes, enabling efficient parallel processing and local reconstruction in large-scale scenes.

Sub-cluster Refinement. After splitting each cluster into subsets $\{G_1, G_2, \dots, G_K\}$ based on camera orientations and positions, we further refine the cluster-

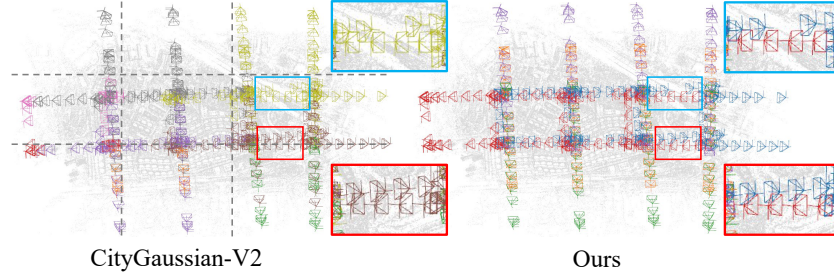


Fig. 4: Visual comparison of scene partitioning in our method and CityGaussian-V2. The dashed lines in the left figure indicate the nine spatial partitions used in CityGaussian-V2. Cameras belonging to the same group are shown in the same color. The difference is highlighted.

ing to ensure that all views within each group have sufficient overlap for reliable geometry optimization, especially for views located near the boundaries of the group. Specifically, for each view $v \in G_k$, we identify a set of source views $S(v) \subseteq \mathcal{V}$ that are favorable for geometry reconstruction of v , where $\mathcal{V}$ denotes the set of all views. These source views are then merged into the corresponding cluster, forming an expanded group,

$$\tilde{G}_k = G_k \cup \bigcup_{v \in G_k} S(v). \quad (2)$$

This refinement ensures that the boundary views within each group are supported by sufficient neighboring views for applying local multi-view geometric constraints, while preserving the main structure of the initial clustering.

Fig. 4 shows a visual comparison of scene partitioning between our method and CityGaussian-V2 [31]. CityGaussian-V2 merely performs spatial partitioning based on the 3D positions of the scene, which may group cameras with completely opposite viewing directions into the same block. This design can significantly hinder accurate geometry inference in local regions. In contrast, our method clusters views based on camera orientation, ensuring that cameras beneficial to each other’s reconstruction are grouped together.

3.2 Initial Point Cloud Completion

Motivation. Incomplete initial point clouds harm the accuracy and completeness of surface reconstruction. While the splitting and cloning strategies in 3DGS can generate new Gaussians to under-reconstructed regions, these mechanisms are largely ineffective when the initial point cloud is extremely sparse or missing entirely. Previous studies [11, 14, 53] have demonstrated that a dense initialization significantly benefits 3DGS-based reconstruction, typically using dense point clouds generated by multi-view stereo (MVS) methods [39, 56] or pre-trained models such as MAST3R [19]. However, generating dense point clouds

with MVS for large-scale image sets is highly time-consuming, and the pretrained reconstruction models are generally limited to small-scale scenes.

We observe that, in large-scale reconstructions, missing regions in the initial point cloud are typically confined to areas with insufficient viewpoint coverage or poor texture. This insight motivates a targeted approach. Rather than generating a globally dense initialization, we instead focus on detecting missing regions in the initial point cloud and selectively filling them. In this way, we efficiently improve the initialization for 3DGS reconstruction while avoiding the substantial computational cost of full dense point cloud generation.

Hole Detection. To detect potential missing regions in the initial sparse point cloud, we evaluate the spatial distribution uniformity of valid 3D points projected onto each input image. Intuitively, when the reconstructed 3D points corresponding to a view are densely and uniformly distributed, the regions of the scene covered by that view are likely to be well reconstructed. In contrast, if the projected points are highly clustered or entirely absent in certain regions, the corresponding regions of the scene are likely to contain holes due to insufficient geometric initialization.

In practice, we directly utilize the 2D valid keypoints recorded by COLMAP to detect holes in the point cloud, as these 2D keypoints correspond to the reconstructed 3D points. Specifically, we divide the image plane into a fixed $N \times N$ grid, and count the number of valid keypoints falling into each cell. The distribution uniformity of valid points is then quantified by the the Shannon entropy,

$$E = - \sum_{j=1}^{N^2} p_j \log(p_j + \epsilon), \quad (3)$$

$$E_{norm} = \frac{H}{\log(N^2)}, \quad (4)$$

where p_j denotes the proportion of points in the j -th grid cell, ϵ is a small constant for numerical stability, and E_{norm} is the normalized entropy. A lower entropy E_{norm} indicates a more uneven distribution of valid points, suggesting that the corresponding image likely observes a region with insufficient geometric coverage. We therefore regard images with low entropy ($E_{norm} < \tau$) as candidates likely to correspond to missing or sparse regions in the initial point cloud, and select them for subsequent refinement.

Hole Completion. For each candidate image, its associated neighbor source views are used to construct image pairs, which are fed into a pretrained feature matching network [36] to predict dense correspondences. The matched points are triangulated to generate dense 3D points, which are then merged with the original COLMAP sparse point cloud through simple concatenation after the geometric consistency filtering and duplicate removal, producing a refined initialization that fills the detected holes.

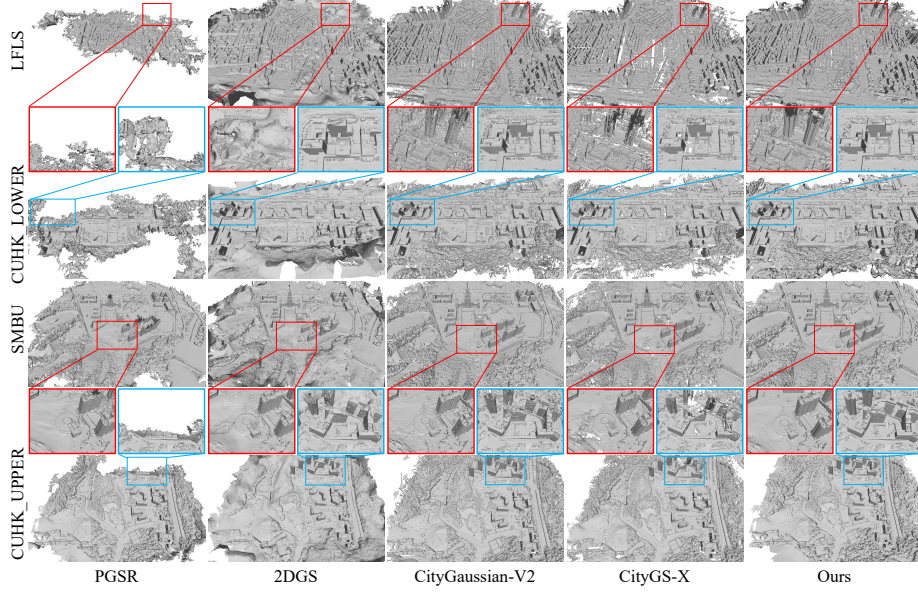


Fig. 5: Visual comparison of reconstruction on the GauU-Scene dataset.

3.3 Training Loss

Our loss function consists of the multi-view photometric consistency loss L_{patch} and multi-view geometric consistency loss L_{geo} from PGSR [2], as well as the color reconstruction loss L_{rgb} and depth-normal consistency loss L_{normal} commonly used in 3DGS-based surface reconstruction methods [2, 15, 59]. The final loss function is defined as follows,

$$L = L_{rgb} + \lambda_1 L_{patch} + \lambda_2 L_{geo} + \lambda_3 L_{normal}. \quad (5)$$

The normalized cross-correlation (NCC) [58] between patches in the reference view and the source views is used as the multi-view photometric consistency loss, defined as follows,

$$L_{patch} = \frac{1}{M} \sum (1 - NCC(P_r, P_s)), \quad (6)$$

where $P_s = H_{rs}P_r$, P_r is the patch on the reference view, and P_s is the corresponding patch on the source view, obtained by warping P_r using the homography matrix H_{rs} .

4 Experiments

4.1 Datasets

We conduct experiments on three datasets: UrbanScene3D [27], GauU-Scene [51], and a small city region from MatrixCity [25]. Specifically, we use the *Residence*

Table 1: Quantitative results of F1 score on GauU-Scene dataset. The best results are in *bold*, the second best are *underlined*.

Methods	CUHK_LOWER	CUHK_UPPER	HAV	LFLS	SMBU	SZIIT	Mean F1 $\uparrow$
2DGS [15]	0.396	0.263	0.386	0.255	0.323	0.321	0.324
PGSR [2]	0.102	0.493	0.573	0.119	<u>0.562</u>	0.151	0.333
CityGaussian-V2 [31]	0.491	0.481	<u>0.570</u>	<u>0.574</u>	0.425	0.495	0.506
CityGS-X [10]	<u>0.508</u>	<u>0.509</u>	0.521	0.567	0.556	<u>0.504</u>	<u>0.527</u>
Ours	0.536	0.537	0.554	0.594	0.597	0.545	0.560

and *Sci-Art* scenes from UrbanScene3D, six scenes from GauU-Scene, and aerial-view images from MatrixCity. Each scene in UrbanScene3D contains more than 2,500 images, while each scene in GauU-Scene includes approximately 400 to 1,200 images. MatrixCity provides over 5,000 images. The spatial extent of these scenes ranges from 0.016 km² to 2.7 km². For preprocessing, we downsample the images from UrbanScene3D and GauU-Scene by a factor of 4. For MatrixCity, we follow the CityGaussian [30] and resize the longer edge of each image to 1,600 pixels. The sparse point clouds generated by SfM for MatrixCity and UrbanScene are provided by CityGaussian-V2 [31]. In addition, GauU-Scene and MatrixCity supply ground-truth point clouds for geometric evaluation.

4.2 Implementation Details

We implement our method based on the PGSR [2] framework. Each group is trained for 30K iterations. The threshold ϵ_{dir} for camera clustering is set to 0.71, which corresponds to an angular threshold of 45 degrees, and the maximum number of cameras in each cluster M_{max} is set to 200. We select 4 source views for each reference view. The grid size N for entropy computation is set to 10, and the entropy threshold τ is set to 0.7. The hyperparameters in Eq. (5) are set as $\lambda_1 = 0.15$, $\lambda_2 = 0.03$, and $\lambda_3 = 0.05$. All experiments are conducted on 8 $\times$ RTX 3090 GPUs.

4.3 Baselines and Metrics

We compare our method with state-of-the-art large-scale surface reconstruction approaches based on 3DGS, including CityGaussian-V2 [31] and CityGS-X [10]. To compare with the PGSR [2] and 2DGS [15], we set the training iterations to 100K and reduce the spherical harmonic (SH) degree to 1 to avoid out-of-memory. To evaluate the surface quality, we report the F1 score. We additionally provide the metrics PSNR, SSIM, and LPIPS for novel view rendering in the supplementary materials.

4.4 Quality Comparisons

GauU-Scene. Tab. 1 shows the quantitative evaluation results on six scenes from the GauU-Scene [51] dataset. Our method achieves superior overall performance compared with other approaches. It can be observed that PGSR [2]

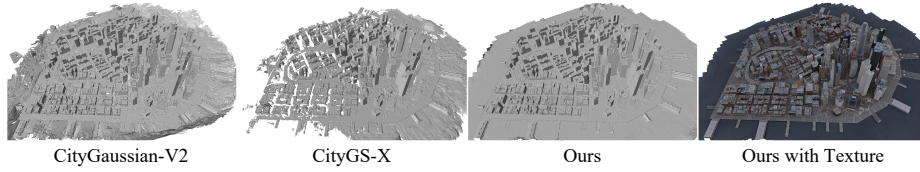


Fig. 6: Visual comparison of reconstruction on the MatrixCity dataset.

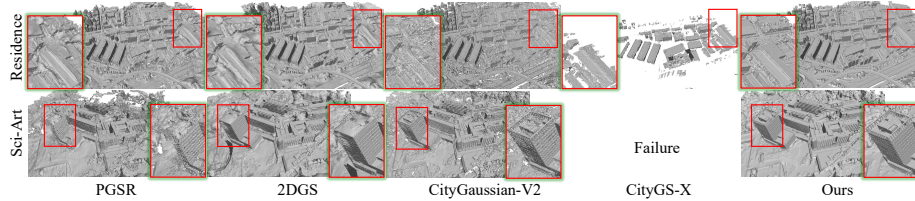


Fig. 7: Visual comparison of reconstruction on the UrbanScene3D dataset.

achieves the best performance on the *HAV* scene, which contains only 424 images and represents a relatively small-scale scene. This indicates that the non-block optimization strategy of PGSR is more advantageous for reconstructing compact scenes.

Fig. 5 provides a visual comparison of surface reconstruction results. The reconstruction results of PGSR [2] exhibit severe incompleteness, while 2DGS [15] and CityGaussian-V2 [31] produces relatively coarse surfaces, and CityGS-X [10] also presents noticeable mesh artifacts. In contrast, our method achieves smoother and more complete reconstructions. It can be observed that although CityGaussian-V2 produces more complete surfaces than PGSR on the *SMBU* and *CUHK_UPPER* scenes, its quantitative results are lower. This is because the ground-truth LiDAR point clouds provided by GauU-Scene [51] contain incomplete edge regions. Following the CityGaussian-V2, we crop out these edge regions and retain only the central regions for quantitative evaluation. Please refer to the supplementary material for more details.

MatrixCity. Tab. 2 and Fig. 6 present the quantitative results and visual comparisons on the MatrixCity [25] aerial-view dataset. Both PGSR [2] and 2DGS [15] completely fail to reconstruct the geometric surfaces of the MatrixCity due to the large-scale scene and the excessive number of images. CityGaussian-V2 [31] generates coarse reconstructions, with obvious concavities in dark regions on the buildings. The reconstruction results from CityGS-X [10] exhibit significant missing geometry, and do not match those shown in its original paper. Overall, our approach produces the highest-quality reconstructions.

UrbanScene3D. Fig. 7 illustrates the visual comparison of reconstruction results on the UrbanScene3D [27] dataset. Both PGSR and 2DGS produce distorted and defective surfaces, while CityGaussian-V2 generates coarse reconstructions. CityGS-X fails to reconstruct the *Sci-Art* scene and produces incom-

Table 2: Quantitative results of precision, recall and F1 score on MatrixCity dataset. the P and R denote precision and recall respectively.

Methods	MatrixCity		
	P	R	F1
CityGaussian-V2	0.441	0.752	0.556
CityGS-X	<u>0.444</u>	<u>0.840</u>	<u>0.581</u>
Ours	0.713	0.920	0.803

Table 3: Efficiency comparison on GauU-Scene and MatrixCity datasets.

Methods	CityGaussian-V2	CityGS-X	Ours
GauU-Scene (hours)	6.5	3	2
MatrixCity (hours)	13	5	6.5

plete surfaces in the *Residence* scene. Our method successfully reconstructs both scenes and achieves the best quality.

4.5 Efficiency Comparison

We evaluate the training efficiency of the three methods on the GauU-Scene [51] and MatrixCity [25] datasets, as shown in Tab. 3. CityGaussian-V2 [31] adopts a two-stage training strategy and divides each scene into 9 or 16 blocks. These blocks cannot be distributed to 8 GPUs all at once, and each block requires long training time, leading to lower efficiency. CityGS-X [10] avoids scene-blocking by employing voxel-level parallelism, which significantly reduces training time. Our method adopts a camera group-based parallel training strategy, where each scene of GauU-Scene contains fewer than 8 groups (up to 7), enabling full GPU utilization in one pass. with each group trained in about 2 hours. On the MatrixCity dataset, which contains over 5,000 training views, our method also demonstrates competitive training efficiency. We cluster all views into 30 groups with the same clustering parameters as other datasets and train over four rounds on 8 GPUs, with a total training time of about 6.5 hours.

4.6 Ablation Study

We conduct ablation experiments on the GauU-Scene dataset by clustering the input views using different angular thresholds, while keeping the maximum number of cameras in each cluster no more than $M_{max} = 200$. The results are shown in Tab. 4. We observe that the best reconstruction quality is achieved when the angular threshold is set to 45 degrees. A threshold that is too small or too large leads to a decline in reconstruction quality. It is worth noting that a smaller angular threshold results in a larger number of clusters, which requires more GPU resources. On the other hand, since we impose a maximum number of cameras

Table 4: Ablation studies on the GauU-Scene dataset by clustering the input views using different angular thresholds. The Groups is total number of clusters across all scenes, and the F1 score is the average over all scenes.

Angular Threshold	15	30	45	60
Groups	36	35	27	27
F1 Score	0.534	0.544	0.560	0.532

Table 5: Ablation of entropy threshold τ on the GauU-Scene and MatrixCity datasets.

τ	w/o	0.6	0.7	0.8	0.85	0.9
GauU-Scene	0.546	0.551	0.560	0.561	0.563	0.565
MatrixCity	0.772	0.797	0.803	0.805	0.806	OOM

per group in the second-stage clustering based on camera positions, increasing the angular threshold does not reduce the number of clusters. However, it introduces greater variation in camera orientations within each group, which in turn degrades the quality of the reconstruction.

To validate the effectiveness of the initialization point cloud enhancement and determine the appropriate entropy threshold τ , we conduct ablation studies on the GauU-Scene [51] and MatrixCity [25] datasets. We first compute the probability density distribution of image entropy E_{norm} for all scenes in GauU-Scene and MatrixCity, as shown in Fig. 8. It can be observed that the entropy E_{norm} of valid points in images across different scenes exhibits a similar distribution, mostly ranging between 0.6 and 1.0, indicating that a unified entropy threshold can be applied across different scenes.

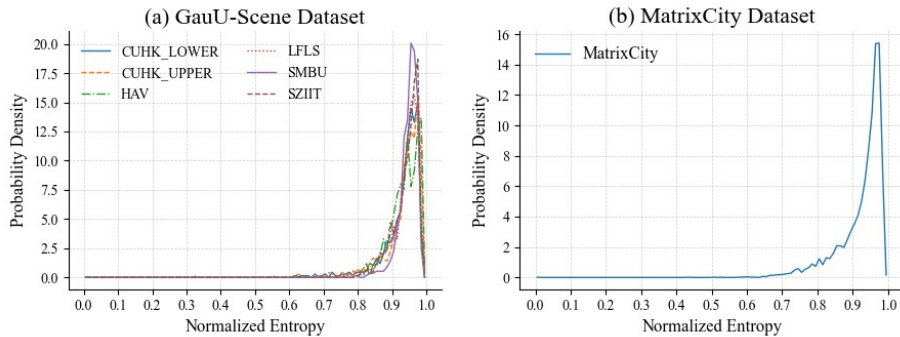


Fig. 8: Distribution of normalized entropy on the Gau-Uscene and MatrixCity Datasets.

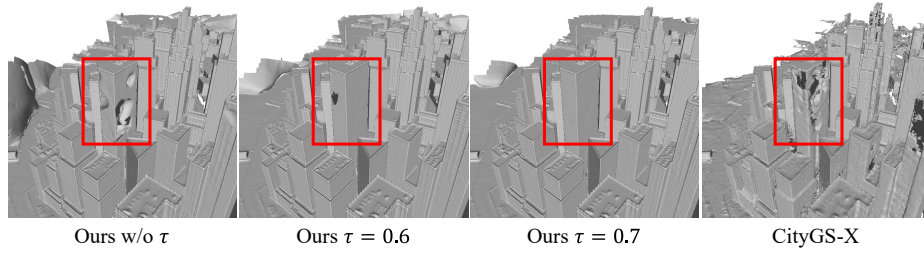


Fig. 9: Visualization of reconstruction with varying entropy threshold τ values.

We then gradually increase the entropy threshold τ starting from 0.6 and perform reconstruction and quantitative evaluation for scenes in GauU-Scene and MatrixCity, with results summarized in Tab. 5. The results show that, for both GauU-Scene and MatrixCity, reconstruction quality progressively improves as the entropy threshold τ increases, i.e., as the number of images used to enhance the initial point cloud grows. More importantly, an entropy threshold of 0.6 for the initial point cloud enhancement leads to a notable improvement in reconstruction performance, even when only a few images are used. However, as the τ increases, the performance gain becomes marginal, while the point cloud completion process becomes more time-consuming. In addition, the excessive number of points in the initial point cloud may lead to GPU memory exhaustion. Considering both the entropy distribution (as shown in Fig. 8) and computational efficiency, we set the entropy threshold to 0.7. It is worth noting that even without scene completion, our method still outperforms other baselines on MatrixCity and achieves comparable performance to CityGS-X on GauU-Scene.

Fig. 9 shows the reconstruction results under different values of τ . It can be observed that as τ increases, the defects in the mesh gradually decrease.

To further verify the effect of completing the initial point cloud on reconstruction quality of other methods, we use the completed point cloud as initialization for 2DGS [15], PGSR [2], CityGaussian-V2 [31] and CityGS-X [10] on the GauU-Scene [51] dataset and conduct quantitative evaluations of the reconstruction results, as shown in Tab. 6. The results show that incorporating the completed point clouds leads to performance gains for all methods, demonstrating that local point cloud completion effectively enhances reconstruction quality. We follow CityGaussian-V2 [31] and crop the boundaries of the provided ground-truth point clouds, retaining only the central region for evaluation. Note that the evaluation regions for each scene are therefore subsets of the full scene, while the completion of the initialization point cloud may occur outside these predefined regions, which explains why the improvement is not always noticeable in some cases.

Table 6: Quantitative results of F1 score on the GauU-Scene dataset with different initial point clouds (PC).

Methods	Init.	CUHK_LOWER	CUHK_UPPER	HAV	LFLS	SMBU	SZIIT	Mean F1 $\uparrow$
2DGS	Sparse PC	0.396	0.263	0.386	0.255	0.323	0.321	0.324
	Completed PC	0.412	0.312	0.387	0.254	0.325	0.324	0.336
PGSR	Sparse PC	0.102	0.493	0.573	0.119	0.562	0.151	0.333
	Completed PC	0.345	0.530	0.574	0.119	0.561	0.164	0.382
CityGaussian-V2	Sparse PC	0.491	0.481	0.570	0.574	0.425	0.495	0.506
	Completed PC	0.520	0.493	0.569	0.579	0.425	0.495	0.514
CityGS-X	Sparse PC	0.508	0.509	0.521	0.567	0.556	0.504	0.527
	Completed PC	0.531	0.518	0.521	0.568	0.557	0.504	0.533

5 Conclusion

We propose a novel method for large-scale scene reconstruction. We group the input views based on camera orientation and position, ensuring that the views with overlaps are clustered together to recover more accurate scene geometry. Meanwhile, we detect and complete regions with missing geometry in the initial point cloud, which further enhances surface reconstruction quality in regions with insufficient view coverage. Extensive experiments on the GauU-Scene, MatrixCity, and UrbanScene3D datasets demonstrate that our method outperforms the latest large-scale surface reconstruction methods.

Acknowledgements

The corresponding authors are Yu-Shen Liu and Junsheng Zhou. This work was partially supported by Deep Earth Probe and Mineral Resources Exploration – National Science and Technology Major Project (2024ZD1003405), and the National Natural Science Foundation of China (62272263).

References

1. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-NeRF 360: Unbounded anti-aliased neural radiance fields. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5470–5479 (2022) [2](#), [4](#)
2. Chen, D., Li, H., Ye, W., Wang, Y., Xie, W., Zhai, S., Wang, N., Liu, H., Bao, H., Zhang, G.: PGSR: Planar-based gaussian splatting for efficient and high-fidelity surface reconstruction. IEEE Transactions on Visualization and Computer Graphics **31**, 1–12 (01 2024) [4](#), [9](#), [10](#), [11](#), [14](#)
3. Chen, J., Ye, W., Wang, Y., Chen, D., Huang, D., Ouyang, W., Zhang, G., Qiao, Y., He, T.: GigaGS: 3D gaussian based planar representation for large-scene surface reconstruction. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 39, pp. 2088–2096 (2025) [5](#)

4. Chen, Y., Lee, G.H.: DoGS: Distributed-oriented gaussian splatting for large-scale 3D reconstruction via gaussian consensus. *Advances in Neural Information Processing Systems* **37**, 34487–34512 (2024) 2, 5
5. Dai, P., Xu, J., Xie, W., Liu, X., Wang, H., Xu, W.: High-quality surface reconstruction using gaussian surfels. In: *ACM SIGGRAPH 2024 Conference Proceedings* (2024) 4
6. Darmon, F., Bascle, B., Devaux, J.C., Monasse, P., Aubry, M.: Improving neural implicit surfaces geometry with patch warping. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 6260–6269 (2022) 2
7. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: Density-based spatial clustering of applications with noise. In: *Int. Conf. knowledge discovery and data mining*. vol. 240 (1996) 6
8. Fan, J., Li, W., Han, Y., Dai, T., Tang, Y.: Momentum-GS: Momentum gaussian self-distillation for high-quality large scene reconstruction. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 25250–25260 (2025) 5
9. Feng, G., Chen, S., Fu, R., Liao, Z., Wang, Y., Liu, T., Hu, B., Xu, L., Pei, Z., Li, H., et al.: FlashGS: Efficient 3D gaussian splatting for large-scale and high-resolution rendering. In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. pp. 26652–26662 (2025) 5
10. Gao, Y., Li, H., Chen, J., Zou, Z., Zhong, Z., Zhang, D., Sun, X., Han, J.: CityGS-X: A scalable architecture for efficient and geometrically accurate large-scale scene reconstruction. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 27187–27196 (2025) 1, 5, 10, 11, 12, 14
11. Guédon, A., Ichikawa, T., Yamashita, K., Nishino, K.: MAtCha Gaussians: Atlas of charts for high-quality geometry and photorealism from sparse views. In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. pp. 6001–6011 (2025) 7
12. Guédon, A., Lepetit, V.: SuGaR: Surface-aligned gaussian splatting for efficient 3D mesh reconstruction and high-quality mesh rendering. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 5354–5363 (2024) 4
13. Han, L., Zhang, X., Song, H., Shi, K., Liu, Y.S., Han, Z.: SparseRecon: Neural implicit surface reconstruction from sparse views with feature and depth consistencies. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2025) 2
14. Han, L., Zhou, J., Liu, Y.S., Han, Z.: Binocular-guided 3D gaussian splatting with view consistency for sparse view synthesis. *Advances in Neural Information Processing Systems* **37**, 68595–68621 (2024) 7
15. Huang, B., Yu, Z., Chen, A., Geiger, A., Gao, S.: 2D gaussian splatting for geometrically accurate radiance fields. In: *ACM SIGGRAPH 2024 Conference Proceedings* (2024) 4, 9, 10, 11, 14
16. Huang, H., Wu, Y., Deng, C., Gao, G., Gu, M., Liu, Y.S.: FatesGS: Fast and accurate sparse-view surface reconstruction using gaussian splatting with depth-feature consistency. In: *Proceedings of the AAAI Conference on Artificial Intelligence* (2025) 2
17. Huang, H., Wu, Y., Zhou, J., Gao, G., Gu, M., Liu, Y.S.: NeuSurf: On-surface priors for neural surface reconstruction from sparse input views. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. vol. 38, pp. 2312–2320 (2024) 2

18. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3D gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics* **42**(4), 1–14 (2023) [2](#), [4](#)
19. Leroy, V., Cabon, Y., Revaud, J.: Grounding image matching in 3D with MAST3R. In: *European Conference on Computer Vision*. pp. 71–91. Springer (2024) [7](#)
20. Li, J., Pang, B., Wang, P.S.: Joint point cloud upsampling and cleaning with octree-based cnns. *Computational Visual Media* (2026) [4](#)
21. Li, Q., Feng, H., Gong, X., Liu, Y.S.: Learning normals of noisy points by local gradient-aware surface filtering. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 28828–28838 (2025) [4](#)
22. Li, Q., Feng, H., Gong, X., Liu, Y.S.: VA-GS: Enhancing the geometric representation of gaussian splatting via view alignment. *Advances in Neural Information Processing Systems* **38**, 173867–173886 (2026) [4](#)
23. Li, Q., Feng, H., Shi, K., Gao, Y., Fang, Y., Liu, Y.S., Han, Z.: PFF-Net: Patch feature fitting for point cloud normal estimation. *IEEE Transactions on Visualization and Computer Graphics* (2025) [4](#)
24. Li, S., Liu, Y.S., Han, Z.: GaussianUDF: Inferring unsigned distance functions through 3D gaussian splatting. In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. pp. 27113–27123 (2025) [4](#)
25. Li, Y., Jiang, L., Xu, L., Xiangli, Y., Wang, Z., Lin, D., Dai, B.: MatrixCity: A large-scale city dataset for city-scale neural rendering and beyond. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 3205–3215 (2023) [9](#), [11](#), [12](#), [13](#)
26. Lin, J., Li, Z., Tang, X., Liu, J., Liu, S., Liu, J., Lu, Y., Wu, X., Xu, S., Yan, Y., et al.: VastGaussian: Vast 3D gaussians for large scene reconstruction. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 5166–5175 (2024) [2](#), [4](#)
27. Lin, L., Liu, Y., Hu, Y., Yan, X., Xie, K., Huang, H.: Capturing, reconstructing, and simulating: the UrbanScene3D dataset. In: *European Conference on Computer Vision*. pp. 93–109. Springer (2022) [9](#), [11](#)
28. Liu, S., Tang, X., Li, Z., He, Y., Ye, C., Liu, J., Huang, B., Zhou, S., Wu, X.: OccluGaussian: Occlusion-aware gaussian splatting for large scene reconstruction and rendering. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 26643–26652 (2025) [4](#)
29. Liu, W., Guan, T., Zhu, B., Xu, L., Song, Z., Li, D., Wang, Y., Yang, W.: EfficientGS: Streamlining gaussian splatting for large-scale high-resolution scene representation. *IEEE MultiMedia* (2025) [5](#)
30. Liu, Y., Luo, C., Fan, L., Wang, N., Peng, J., Zhang, Z.: CityGaussian: Real-time high-quality large-scale scene rendering with gaussians. In: *European Conference on Computer Vision*. pp. 265–282. Springer (2024) [2](#), [5](#), [10](#)
31. Liu, Y., Luo, C., Mao, Z., Peng, J., Zhang, Z.: CityGaussianV2: Efficient and geometrically accurate reconstruction for large-scale scenes. In: *International Conference on Learning Representations*. vol. 2025, pp. 84324–84341 (2025) [1](#), [5](#), [7](#), [10](#), [11](#), [12](#), [14](#)
32. Lyu, X., Sun, Y.T., Huang, Y.H., Wu, X., Yang, Z., Chen, Y., Pang, J., Qi, X.: 3DGSR: Implicit surface reconstruction with 3D gaussian splatting. *ACM Transactions on Graphics (TOG)* **43**(6), 1–12 (2024) [4](#)
33. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: NeRF: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM* **65**(1), 99–106 (2021) [2](#), [4](#)

34. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)* **41**(4), 1–15 (2022) [2](#), [4](#)
35. Na, Y., Kim, W.J., Han, K.B., Ha, S., Yoon, S.E.: UFORecon: Generalizable sparse-view surface reconstruction from arbitrary and unfavorable sets. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 5094–5104 (2024) [2](#)
36. Ren, J., Jiang, X., Li, Z., Liang, D., Zhou, X., Bai, X.: MINIMA: Modality invariant image matching. In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. pp. 23059–23068 (2025) [3](#), [8](#)
37. Ren, K., Jiang, L., Lu, T., Yu, M., Xu, L., Ni, Z., Dai, B.: Octree-GS: Towards consistent real-time rendering with lod-structured 3D gaussians. *arXiv preprint arXiv:2403.17898* (2024) [4](#)
38. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 4104–4113 (2016) [3](#)
39. Schönberger, J.L., Zheng, E., Pollefeys, M., Frahm, J.M.: Pixelwise view selection for unstructured multi-view stereo. In: *European Conference on Computer Vision* (2016) [3](#), [7](#)
40. Shao, D., Jing, Y., Zhao, X., Mao, S., Lyu, L., Liu, X., Lu, X.: DS-MAE: Dual-siamese masked autoencoders for point cloud analysis. *Computational Visual Media* (2025) [4](#)
41. Shao, M., Xiong, F., Zhang, H., Yang, S., Xu, M., Bian, W., Wang, X.: Global-guided focal neural radiance field for large-scale scene rendering. In: *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. pp. 2558–2567. *IEEE* (2025) [4](#)
42. Su, M., Wang, Z., Au, H., Li, Y., Cao, X., Pan, C., Chen, Y., Wang, G.: HUG: Hierarchical urban gaussian splatting with block-based reconstruction for large-scale aerial scenes. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 28839–28848 (2025) [4](#)
43. Tancik, M., Casser, V., Yan, X., Pradhan, S., Mildenhall, B., Srinivasan, P.P., Barron, J.T., Kretzschmar, H.: Block-NeRF: Scalable large scene neural view synthesis. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 8248–8258 (2022) [2](#), [4](#)
44. Turki, H., Ramanan, D., Satyanarayanan, M.: Mega-NeRF: Scalable construction of large-scale nerfs for virtual fly-throughs. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 12922–12931 (2022) [4](#)
45. Wang, P., Liu, L., Liu, Y., Theobalt, C., Komura, T., Wang, W.: NeuS: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *Advances in Neural Information Processing Systems* (2021) [2](#), [4](#)
46. Wang, Y., Zhou, K., Zhang, W., Xiao, C.: MegaSurf: Scalable large scene neural surface reconstruction. In: *Proceedings of the 32nd ACM International Conference on Multimedia*. pp. 6414–6423 (2024) [2](#), [5](#)
47. Wang, Z., Xu, D.: PyGS: Large-scale scene representation with pyramidal 3D gaussian splatting. *arXiv preprint arXiv:2405.16829* (2024) [4](#)
48. Wen, X., Xiang, P., Han, Z., Cao, Y.P., Wan, P., Zheng, W., Liu, Y.S.: PMP-Net++: Point cloud completion by transformer-enhanced multi-step point moving paths. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **45**(1), 852–867 (2022) [4](#)

49. Wu, Y., Qi, Z., Shi, Z., Zou, Z.: BlockGaussian: Efficient large-scale scene novel view synthesis via adaptive block-based gaussian splatting. *arXiv preprint arXiv:2504.09048* (2025) [2](#), [4](#)
50. Xiang, P., Wen, X., Liu, Y.S., Cao, Y.P., Wan, P., Zheng, W., Han, Z.: Snowflake point deconvolution for point cloud completion and generation with skip-transformer. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **45**(5), 6320–6338 (2022) [4](#)
51. Xiong, B., Li, Z., Li, Z.: Gauu-Scene: A scene reconstruction benchmark on large scale 3D reconstruction dataset using gaussian splatting. *arXiv preprint arXiv:2401.14032* (2024) [9](#), [10](#), [11](#), [12](#), [13](#), [14](#)
52. Xu, M., Zhou, Z., Wang, Y., Qiao, Y.: Towards robustness and generalization of point cloud representation: A geometry coding method and a large-scale object-level dataset. *Computational Visual Media* **10**(1), 27–43 (2024) [4](#)
53. Xu, W., Gao, H., Shen, S., Peng, R., Jiao, J., Wang, R.: MVPGS: Excavating multi-view priors for gaussian splatting from sparse input views. In: *European Conference on Computer Vision*. pp. 203–220. Springer (2024) [7](#)
54. Yang, L., Deng, B., Zhang, J.: Scalable and high-quality neural implicit representation for 3D reconstruction. *IEEE Transactions on Visualization and Computer Graphics* (2025) [5](#)
55. Yang, Y.Q., Guo, Y.X., Liu, Y.: Swin3D++: Effective multi-source pretraining for 3D indoor scene understanding. *Computational Visual Media* (2025) [4](#)
56. Yao, Y., Luo, Z., Li, S., Fang, T., Quan, L.: MVSNet: Depth inference for unstructured multi-view stereo. In: *Proceedings of the European conference on computer vision (ECCV)*. pp. 767–783 (2018) [2](#), [7](#)
57. Yariv, L., Gu, J., Kasten, Y., Lipman, Y.: Volume rendering of neural implicit surfaces. *Advances in Neural Information Processing Systems* **34**, 4805–4815 (2021) [2](#), [4](#)
58. Yoo, J.C., Han, T.H.: Fast normalized cross-correlation. *Circuits, systems and signal processing* **28**, 819–843 (2009) [9](#)
59. Yu, Z., Sattler, T., Geiger, A.: Gaussian Opacity Fields: Efficient adaptive surface reconstruction in unbounded scenes. *ACM Transactions on Graphics* **43**(6), 1–13 (2024) [4](#), [9](#)
60. Yuan, Z., Huang, H., Xiong, Z., Wang, D., Yang, G.: Robust and efficient 3D gaussian splatting for urban scene reconstruction. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 26209–26219 (2025) [5](#)
61. Zhang, H., Xue, Y., Liao, M., Lao, Y.: BirdNeRF: fast neural reconstruction of large-scale scenes from aerial imagery. *Scientific Reports* **15**(1), 37295 (2025) [4](#)
62. Zhang, J., Zhang, J., Ding, D., Ma, Z.: ARNet: Attribute artifact reduction for G-PCC compressed point clouds. *Computational Visual Media* (2025) [4](#)
63. Zhang, W., Jia, E.Y.t., Zhou, J., Ma, B., Shi, K., Liu, Y.S., Han, Z.: NeRFPrior: Learning neural radiance field as a prior for indoor scene reconstruction. In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. pp. 11317–11327 (2025) [2](#)
64. Zhang, W., Wang, C., Shi, K., Liu, Y.S., Han, Z.: VRP-UDF: Towards unbiased learning of unsigned distance functions from multi-view images with volume rendering priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2026) [2](#)
65. Zhang, W., Yang, Y., Huang, H., Han, L., Shi, K., Liu, Y.S., Han, Z.: MonoInstance: Enhancing monocular priors via multi-view instance alignment for neural rendering and reconstruction. In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. pp. 21642–21653 (2025) [2](#)