3D Gaussian Splatting for Large-Scale 3D Reconstruction: An Evaluation and Quality Analysis

Jiangxue Yu [1], Yueling Liao [1], San Jiang [2,3,*], Xing Zhang [2,3], Zhijun Wang [4], Qingquan Li [2,3]

[1] School of Computer Science, China University of Geosciences, Wuhan 430074, China
[2] Guangdong Key Laboratory of Urban Informatics, Shenzhen University, Guangdong Shenzhen, 518060, China
[3] MNR Key Laboratory for Geo-Environmental Monitoring of Great Bay Area, Shenzhen University, Guangdong Shenzhen, 518060, China
[4] Guangdong Laboratory of Artificial Intelligence and Digital Economy (Shenzhen), Guangdong Shenzhen, 518060, China

KEY WORDS: Photogrammetry; 3D reconstruction; 3D Gaussian Splatting; Structure from Motion

ABSTRACT:

Large-scale 3D reconstruction has emerged as a key research in the fields of photogrammetry and computer vision. 3D Gaussian Splatting (3DGS) has become a mainstream approach due to its efficient rendering, but it confronts critical challenges in large-scale scenarios: excessive memory overhead and inadequate geometric accuracy. Meanwhile, the traditional Structure from Motion and Multi-view Stereo (SfM-MVS) framework, despite its cumbersome process, continues to exhibit robust performance. Notably, a systematic evaluation comparing these two paradigms in large-scale scenes remains absent. To address this, we develop a unified verification framework to evaluate the texture rendering quality and geometric reconstruction precision of several recent methods using real-world datasets. The results indicate that SfM-MVS methods still maintain an advantage in the completeness and accuracy of geometric reconstruction. In contrast, 3DGS methods have achieved breakthroughs in local accuracy or rendering-geometry synergy, yet their global consistency requires further improvement.

1. INTRODUCTION

In recent years, the application of 3D reconstruction technology has become increasingly widespread in fields such as urban planning, virtual reality, and autonomous driving, making large-scale scene reconstruction a major research hotspot. The traditional framework, which combines Structure from Motion (SfM) (Jiang et al., 2020, Jiang et al., 2022, Jiang et al., 2024, Chen et al., 2024) and Multi-view Stereo (MVS) (Furukawa et al., 2015, Sadeq, 2025), reconstructs realistic scenes through a sequence of steps including dense matching, point cloud meshing, and texture mapping. However, this framework suffers from a lengthy and computationally intensive workflow, leading to issues such as long processing times and suboptimal reconstruction quality. Recently, Neural Radiance Fields (NeRF) (Mildenhall et al., 2021) have demonstrated significant potential in novel view synthesis. By training a deep neural network, NeRF can directly map spatial coordinates to color and density, bypassing the cumbersome steps of the SfM-MVS framework. Nevertheless, NeRF-based methods (Yu et al., 2021, Barron et al., 2021, Barron et al., 2022) are computationally demanding, requiring extensive training time and resources. Their implicit representation also makes scene manipulation and editing difficult. As an efficient alternative, 3D Gaussian Splatting (3DGS) (Kerbl et al., 2023) employs an explicit representation and a highly parallelized workflow. Compared to neural implicit representations like NeRF, 3DGS significantly accelerates the rendering speed for novel view synthesis and has gradually become a mainstream algorithm in the field due to its advantages in training and rendering efficiency.

However, when applied to large-scale complex environments (e.g., urban scenes, intricate terrains), 3DGS still faces three critical challenges: excessive memory overhead, prolonged training/compression times, and insufficient geometric accuracy. For computation and storage, representing a scene with millions of Gaussian primitives requires over 15GB of memory, with dynamic and extremely large scenes potentially exceeding terabytes. Backpropagation needs to iterate through thousands of images, resulting in long iteration times and significant memory pressure. In complex scenes, 3DGS struggles to model multi-scale structures; dynamic elements cause Gaussian primitive redundancy or degradation; and special materials/lighting conditions are difficult to simulate. Geometrically, discrete Gaussian ellipsoids fail to align with real-world surfaces, resulting in detail loss, artifacts, noise, and poor global consistency—making their precision inferior to traditional SfM-MVS methods.

In the evolution of 3DGS technology, researchers have proposed optimization methods for large-scale scene reconstruction, which can be broadly categorized into two types. The first category focuses on rendering optimization, including methods like VastGaussian (Lin et al., 2024), CityGaussian (Liu et al., 2024), Hier-GS (Kerbl et al., 2024), DoGaussian (Chen and Lee, 2024), and BlockGaussian (Wu et al., 2025), which improve rendering quality through partitioning strategies and parallel processing. The second category incorporates geometric optimization, with methods such as GigaGS (Chen et al., 2025), CityGaussianV2 (Liu et al., 2025), CoSurfGS (Gao et al., 2024), and CityGS-X (Gao et al., 2025) aiming to enhance geometric accuracy and consistency. Currently, there is a lack of a systematic and comprehensive performance evaluation comparing the traditional SfM-MVS framework with 3DGS and its derivatives for large-scale scene reconstruction. The specific differences in key metrics between these methods have not been clearly defined, nor has their practical effectiveness for large-scale reconstruction been demonstrated. Therefore, this paper establishes a unified verification framework to conduct a holistic performance evaluation of various methods in terms of ren-

*Corresponding author

dering and geometry, using real-world datasets to analyze their advantages and limitations in large-scale 3D reconstruction.

2. 3D RECONSTRUCTION METHODS

For performance evaluation, six methods have been selected. In addition to the traditional SfM-MVS framework, we include several advanced methods based on 3DGS technology: Vast-Gaussian, CityGaussian, BlockGaussian, and two geometry-optimized methods, CityGaussianV2 and CityGS-X.

2.1 3DGS Framework

The classic SfM-MVS framework is a foundational 3D reconstruction technique in photogrammetry and computer vision, recovering scene 3D structure from multi-view images via a well-defined workflow. The core workflow begins with extracting key feature points from the input images, followed by establishing correspondences between these points across different images using feature matching algorithms. With these matched pairs, the SfM algorithm employs incremental or global bundle adjustment (BA) optimization to recover the intrinsic and extrinsic camera parameters and construct a sparse 3D structure of the scene. This sparse reconstruction serves as a prerequisite for both MVS and most 3DGS algorithms. MVS utilizes these camera parameters to compute the 3D coordinates of points in the scene via stereo matching, generating a dense point cloud. This point cloud is then converted into a triangle mesh model. Finally, texture mapping is applied to assign color information from the images to the mesh surface, completing the 3D model.

As shown in Figure 1, 3DGS takes the sparse point cloud generated by SfM as its initial input. It models the geometric structure and appearance features of the scene using explicit 3D Gaussian distributions, with the core being the construction of a set of 3D Gaussian primitives with clearly defined parameters.Each 3D Gaussian is composed of a position μ, a covariance matrix Σ, an opacity α, and a color c. Its probability density function is given by:

$$G(x) = \frac{1}{(2\pi)^{\frac{3}{2}}|\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right) \quad (1)$$

where $x = (x, y, z)$ represents spatial coordinates, and $|\Sigma|$ denotes the determinant of the covariance matrix, which directly determines the shape and spatial extension range of the Gaussian ellipsoid. To flexibly control the orientation and scale of the Gaussian ellipsoid, the covariance matrix Σ can be decomposed into the product of a rotation matrix R and a scaling matrix S: $\Sigma = RSS^TR^T$.

During the training process, 3DGS continuously optimizes the Gaussian parameters through the backpropagation algorithm while integrating an adaptive density control mechanism. The optimization loss function consists of L_1 loss and structural similarity loss $L_{D\text{-SSIM}}$:

$$L = (1-\lambda)L_1 + \lambda L_{D\text{-SSIM}} \quad (2)$$

The L_1 minimizes the difference between the predicted color and the real image color to ensure basic rendering accuracy, while $L_{D\text{-SSIM}}$ considers visual perception factors such as image structure, texture, and contrast, improving the adaptability of rendering results to different lighting conditions.

Based on depth sorting results, the color c_i of each Gaussian is calculated using Spherical Harmonics (SH). Meanwhile, the effective contribution of each Gaussian to the pixel (α_i') is computed according to the projection range of the Gaussian on the image plane and its opacity α_i: $\alpha_i' = \alpha_i G(x)$. Finally, the pixel color is calculated via Alpha blending:

$$C(x) = \sum_{i \in N} \alpha_i' c_i \prod_{j=1}^{i-1}(1-\alpha_j') \quad (3)$$

where N is the set of Gaussian primitives covering the pixel. By accumulating the transparency attenuation of distant Gaussians, the dominant role of nearby Gaussians in determining pixel color is achieved, restoring real visual occlusion effects and generating high-fidelity 2D rendered images.

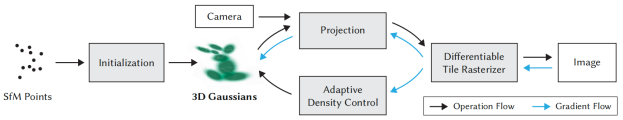

Figure 1: 3D Gaussian Splatting workflow. (Kerbl et al., 2023)

2.2 Principle of 3DGS for large-scale scenes

The basic 3DGS framework achieves efficient rendering via explicit Gaussian primitives and parallelized workflows, but direct application to large-scale scenes reveals core challenges: excessive memory overhead, insufficient global consistency, and difficulty modeling complex structures. To address these, the research community has developed various improved algorithms focusing on ”block strategy optimization” and ”geometric accuracy enhancement”, forming targeted technical paths. As shown in Figure 2, taking the sparse point cloud and camera parameters output by SfM as the unified input, the computational pressure of large-scale scenes is disassembled through scene partitioning and view allocation to avoid memory overload; then gaussian primitive optimization is carried out within the partition, and the local modeling accuracy is improved by combining constraints such as photometric loss and depth loss; finally, regional merging and consistency optimization are used to eliminate block boundary artifacts and ensure global rendering and geometric performance. In the following, around this technical path, five typical large-scale 3DGS algorithms, namely Vast-Gaussian, CityGaussian, BlockGaussian, CityGaussianV2, and CityGS-X, will be introduced in detail, and the differentiated designs of each method in block strategy, optimization objectives, and performance improvement will be analyzed.

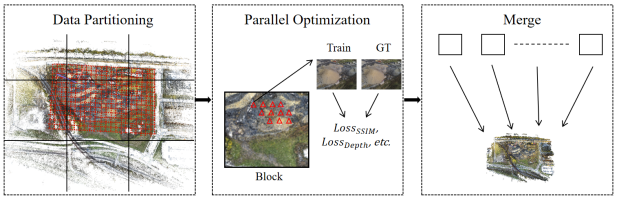

Figure 2: General technical process of the large-scale 3DGS algorithm.

2.3 VastGaussian

VastGaussian (Lin et al., 2024) proposes a method for large-scale scene reconstruction and real-time rendering based on a 3D Gaussian point cloud. Its core workflow consists of four key steps. First, a progressive partitioning strategy divides the large scene into multiple sub-regions, with training data and point clouds allocated based on spatially-aware visibility criteria. Each sub-block boundary is extended by 20% to create overlapping areas, providing a buffer for subsequent consistency optimization and effectively reducing boundary artifacts. Second, each sub-region is optimized independently, combining the 3D Gaussian point cloud representation with a decoupled appearance modeling technique to adapt to appearance variations from different viewpoints while maintaining geometric consistency. Third, after optimization, redundant point clouds at the boundaries are removed, and the sub-regions are merged, leveraging shared views to ensure a seamless connection of the scene. Finally, the appearance modeling module is discarded, and the merged 3D Gaussian point cloud is used directly for high-quality real-time rendering. Through this innovative ”partition-optimize-decouple-merge” workflow, the method effectively addresses issues of memory constraints, low training efficiency, and appearance inconsistencies in large-scale scene reconstruction.

2.4 CityGaussian

CityGaussian (Liu et al., 2024) adopts a ”divide-and-conquer” strategy, partitioning the entire scene into multiple spatially adjacent blocks. Each block is represented by a relatively small number of Gaussian primitives and is trained independently with a smaller subset of data. During the block partitioning process, a global coarse Gaussian model is first trained to serve as an initial prior. This prior is then used to guide scene partitioning and view allocation. By introducing this global prior, the method effectively mitigates the issue of drift in boundary regions and improves the consistency of the merged result. Furthermore, to handle the non-uniform distribution of elements in the scene, CityGaussian employs a partitioning method based on a contracted space. It normalizes the point cloud into a bounded cube through a non-linear coordinate transformation before performing uniform partitioning. This ensures a more even distribution of points within each block, thereby improving load balancing. In the rendering stage, CityGaussian introduces a Level of Detail (LoD) strategy. As objects move farther from the camera, they occupy a smaller area on the screen and contribute less high-frequency information. Consequently, distant, low-detail areas can be well-represented by lower-fidelity models, while nearby areas use high-fidelity models. This strategy significantly reduces the memory and computational requirements for the Gaussian models with minimal performance degradation, achieving a high compression ratio while preserving rendering fidelity.

2.5 BlockGaussian

BlockGaussian (Wu et al., 2025) introduces a novel framework that combines a content-aware scene partitioning strategy with visibility-aware block optimization to achieve efficient and high-quality large-scale scene reconstruction. This approach improves upon the scene partitioning, optimization, and merging challenges faced by traditional 3DGS methods in large-scale scenarios. The method first partitions the scene recursively based on the density of the sparse point cloud, balancing the content complexity of the blocks with the computational load. It then assigns supervisory views to each block based on the proportion of visible keypoints. During the block optimization stage, auxiliary point clouds are introduced to address supervision mismatches. A joint photometric and depth loss is used to optimize both the in-block Gaussians and the auxiliary Gaussians. This is combined with mini-batch optimization to enhance stability, with densification applied only to the in-block Gaussians. Additionally, a pseudo-view geometric constraint is proposed. By perturbing the camera poses of the training views to generate pseudo-views and warping the pseudo-view images into the original view space, a loss is calculated between the warped and ground-truth images to suppress floating artifacts. Compared to VastGaussian and CityGaussian, BlockGaussian’s partitioning strategy is more flexible and better adapts to variations in scene content distribution.

2.6 CityGaussianV2

CityGaussianV2 (Liu et al., 2025) builds upon CityGaussian by incorporating geometric optimization. While its partitioning strategy continues the original divide-and-conquer training framework, it places greater emphasis on geometric processing during block partitioning and optimization. CityGaussianV2 uses 2D Gaussian Splatting (2DGS) (Huang et al., 2024) as its primitive. Compared to 3DGS, 2DGS replaces 3D Gaussian ellipsoids with 2D planar Gaussian disks, which provides a view-consistent geometric representation when modeling surfaces. It employs a depth distortion loss L_{Depth} and a normal consistency loss L_{Normal} as geometric constraints to effectively enhance geometric accuracy. For large-scale scene reconstruction, CityGaussianV2 introduces pseudo-depth supervision from monocular depth estimation to provide a geometric prior. It also utilizes an elongation filter to assess the elongation rate of Gaussians, excluding primitives below a certain threshold from the cloning process to prevent memory exhaustion caused by an explosion in the number of Gaussians during parallel training. Furthermore, it introduces a decomposition-based gradient optimization strategy, prioritizing the structural similarity loss L_{SSIM} and reconstructing the densification gradient. This addresses the blurry reconstruction issue that can arise with 2DGS in large-scale scenes due to the insensitivity of the L_1 loss, thereby significantly improving convergence speed and geometric precision.

2.7 CityGS-X

CityGS-X (Gao et al., 2025) proposes an innovative architecture based on a Parallelized Hybrid Hierarchical 3D (PH²-3D) representation, designed to overcome the limitations of traditional 3DGS methods in terms of computational efficiency, memory consumption, and geometric accuracy. The core idea of this architecture is to dynamically allocate a multi-level-of-detail (LoD) voxel structure across multiple GPUs, enabling efficient parallel training and rendering while eliminating the inefficient partition-merge workflow of conventional methods. The process is divided into three key parts: First, the PH²-3D representation uses a hierarchical voxel structure and a shared Gaussian decoder, representing the scene as a multi-level set of voxels that are dynamically distributed across multiple GPUs to ensure load balancing and efficient computation. Second, through a batch-level multi-task rendering technique, images are split into smaller patches and assigned to different GPUs for the parallel generation of RGB, depth, and normal maps, which significantly boosts rendering efficiency. Finally, a batch-level con-

sistency progressive training strategy is employed. This strategy sequentially performs batch-level RGB training to enhance generalization, depth-prior training to improve geometric consistency, and batch-level geometric training to refine details, all while using multi-view constraints to optimize accuracy.

3. EVALUATION FRAMEWORK AND METRICS

3.1 Evaluation Framework

We design the validation framework illustrated in Figure 3, consisting of three core steps: (1) Sparse Reconstruction. For input images, SfM (including feature extraction and matching) is performed to recover camera poses and 3D points; (2) Algorithm Processing. SfM results are fed into target algorithms to generate rendered images and extract meshes; (3) Quality Evaluation. Rendered image quality and geometric reconstruction accuracy are assessed.

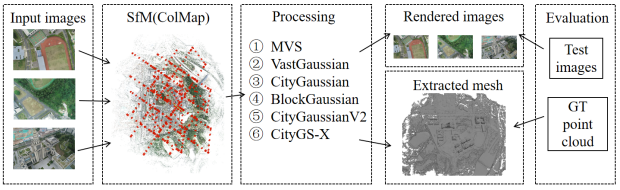

Figure 3: Validation framework.

Key implementation details of the validation framework are as follows. ColMap (Schonberger and Frahm, 2016), a popular open-source software that employs incremental SfM and integrates MVS techniques, is used to recover camera poses, generate a sparse point cloud, and perform the subsequent MVS process. The evaluation protocol for rendering quality is mature and transferable. We follow standard practice, measuring the similarity between rendered images and ground-truth images. For the evaluation of geometric reconstruction, there is currently no unified standard. Here, we follow the evaluation process of CityGaussianV2 (Liu et al., 2025). First, the ground-truth point cloud is downsampled according to the downsampling factor of the images. Next, points are uniformly sampled from the surface of the extracted mesh, registered with the ground-truth point cloud, and cropped to the same extent. Finally, the nearest neighbor distance is calculated for each point between the sampled points and the ground-truth point cloud to assess performance.

3.2 Evaluation Metrics

We select key technical metrics for 3D reconstruction to evaluate the comparative methods, as shown in Table 1. These metrics are divided into two categories to assess the performance of the algorithms in terms of rendered image quality and geometric reconstruction within the validation framework. By using these metrics, we can comprehensively evaluate the quality of the rendered images as well as the completeness and accuracy of the geometric reconstructions.

4. EXPERIMENTS AND RESULTS ANALYSIS

4.1 Experiments Setup

We conducted benchmark tests on four real-world scene datasets. Specifically, we used the Rubble and Building scenes from the

Metric	Description
SSIM	A similarity measure based on the structural information of an image.
PSNR	An objective evaluation standard based on mean squared error.
LPIPS	An image similarity metric based on human perception.
Precision	The proportion of correctly reconstructed points among all reconstructed points.
Recall	The proportion of correctly reconstructed points among all ground-truth points.
F1 Score	The harmonic mean of Precision and Recall.

Table 1: Evaluation metrics.

Mill-19 (Turki et al., 2022) dataset and selected the Russian Building and Modern Building scenes from the GauU-Scene (Xiong et al., 2024) dataset. As shown in Figure 4, these four datasets contain 1678, 1940, 563, and 715 images, respectively. The image resolution for the Rubble and Building scenes is 4608×3456, while for the Russian Building and Modern Building scenes, it is 5468×3636. For the Mill-19 dataset, we reduced the image resolution by a factor of 4, following the method in Mega-NeRF (Turki et al., 2022). For the GauU-Scene dataset, we followed the practice of Kerbl et al. (Kerbl et al., 2023) and downsampled the longer side of the images to 1600 pixels. The GauU-Scene dataset also provides LiDAR data in Figure 5, which serves as the ground-truth point cloud for evaluating geometric reconstruction. Based on the availability of a ground-truth point cloud, the Mill-19 dataset was used to evaluate rendering quality, while the GauU-Scene dataset was mainly used to evaluate geometric reconstruction performance.

Figure 4: Illustration images of the four datasets.

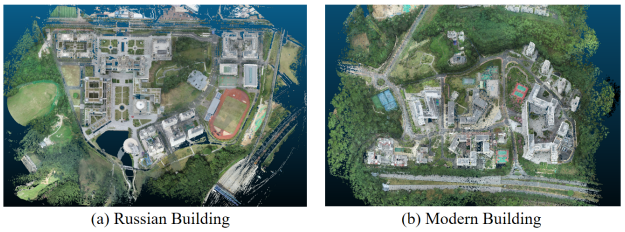

Figure 5: Visualization of the LiDAR data.

In terms of implementation details, for the partition-based algorithms, VastGaussian and CityGaussian (V2) were set to divide all datasets into 3×3 sub-blocks. BlockGaussian used adaptive partitioning, dividing Rubble into 8 blocks and Building into 7 blocks. To better focus on rendering quality evaluation for the Rubble and Building tests, we configured BlockGaussian not to

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-XI-2-2026-403-2026 | © Author(s) 2026. CC BY 4.0 License.

estimate depth maps for the training views, and CityGS-X was run without depth supervision or multi-view/single-view geometric constraints. For the geometric reconstruction evaluation, CityGS-X and CityGaussianV2 used DepthAnythingV2 (Yang et al., 2024) to estimate depth maps and included all geometric losses. Most parameters were kept at their default settings. The voxel size for mesh extraction in CityGaussianV2 and CityGS-X was uniformly set to 0.01, and the distance threshold for geometric evaluation was 0.4m. The experiments were conducted on a Linux platform equipped with an Intel Xeon Gold 6530 CPU, 64GB of memory, and an NVIDIA L40s GPU with 48GB of VRAM.

4.2 Rendering Quality Assessment

In this section, we conduct a comprehensive evaluation of the 3DGS-based algorithms in terms of rendered image quality. First, we perform a qualitative analysis through visual comparison to intuitively assess the rendering effects of each method. Figure 6(a) shows a comparison of the rendered images from four different algorithms (VastGaussian, CityGaussian, BlockGaussian, and CityGS-X) on the Rubble and Building scenes. The images rendered by BlockGaussian and CityGS-X are superior in detail preservation and realism, which is closely related to their algorithmic designs. BlockGaussian's content-aware dynamic partitioning strategy adapts well to complex structural distributions, and its pseudo-view geometric constraint reduces artifacts, resulting in sharper edges. CityGS-X's parallelized hybrid hierarchical 3D representation and batch-level multi-task rendering preserve local details while ensuring overall color consistency. Figure 6(b) compares the rendering results of the geometry-optimized methods (CityGaussianV2 and CityGS-X). It is evident that CityGS-X significantly outperforms CityGaussianV2 in rendering. For instance, it achieves superior results on the surface texture of the Russian Building and on fine details like the power lines in the lower-left of the Modern Building. This is because the 3D Gaussian primitives and shared Gaussian decoder in CityGS-X are better equipped to handle curved surfaces and fine components, whereas the 2D Gaussian primitives of CityGaussianV2 have limitations in these areas.

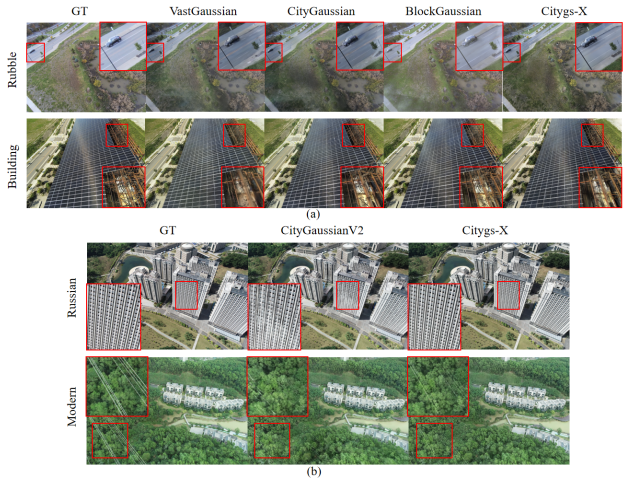

Figure 6: Visualization of the rendered images.

Second, we performed a quantitative analysis by calculating the SSIM, PSNR, and LPIPS metrics to evaluate the rendering quality of the different methods. Table 2 lists the specific numerical values for these metrics. The table shows that CityGaussian and CityGS-X perform well on the SSIM and PSNR metrics, indicating their advantages in structural similarity and pixel-level error. CityGaussian's global coarse Gaussian model ensures structural consistency, and its non-linear coordinate transformation reduces pixel deviation. CityGS-X's batch-level consistency training and multi-GPU parallel rendering improve structural integrity and pixel accuracy, respectively. For the LPIPS metric, CityGS-X achieves the best results because its batch-level RGB training optimizes features sensitive to human vision, and its shared Gaussian decoder adapts to lighting variations. In contrast, other methods exhibit slightly poorer perceptual consistency, potentially due to factors such as discarding the appearance module or losing high-frequency color information.

Scene	Rubble			Building		
Metrics	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓
VastGaussian	0.751	24.91	0.232	0.732	21.93	0.253
CityGaussian	0.811	**25.28**	0.231	0.773	22.00	0.258
BlockGaussian	0.794	25.17	0.237	0.755	21.60	0.241
CityGS-X	**0.817**	25.21	**0.222**	**0.797**	**22.06**	**0.208**

Table 2: Quantitative results for Rubble and Building.

4.3 Geometric Reconstruction Assessment

In this section, we conducted detailed qualitative and quantitative tests on the geometric reconstruction performance of two 3DGS methods (CityGaussianV2 and CityGS-X) and the traditional SfM-MVS method (ColMap). Figure 7 displays the visualization results of the generated geometric models. The quantitative evaluation results, including PSNR, Precision, Recall, and F1 Score, are presented in Table 3. Figure 8 shows the error comparison maps.

In the visualization results in Figure 7, the global views on the left and the locally magnified views marked by red boxes collectively demonstrate the global and detailed differences in geometric reconstruction among the three methods. From the global perspective, all three methods can generally recover the overall shape of the scene, but with different strengths. ColMap, as a traditional SfM-MVS method, produces a geometric model that captures the overall framework of the scene, although the structure appears somewhat coarse, and the transitions at building edges are not entirely natural. CityGS-X exhibits strong global structural integrity, thanks to its parallelized hybrid hierarchical 3D representation that optimizes for global consistency in large-scale scenes, resulting in a coherent framework without noticeable fractures. Although the global structure of CityGaussianV2 is slightly less coherent than that of CityGS-X, its overall accuracy is well-balanced, without significant structural distortion. The locally magnified views reveal differences in detail. For the wall textures and window frames of the Russian Building, CityGaussianV2 leverages the geometric advantages of its 2D Gaussian primitives to clearly represent the concave-convex texture of the walls and the sharp edges of the window frames with high precision. CityGS-X produces a smoother overall result with slightly blurred window frames, as its 3D Gaussian ellipsoids struggle to conform to planar structures, diluting accuracy. The wall details from ColMap are relatively coarse, limited by the feature precision of traditional stereo matching. For the roof of the Modern Building, CityGaussianV2 renders the roof lines with clear folds and continuous lines. CityGS-X shows slight distortion at the edges of roof openings, and its line continuity is somewhat weaker, stemming from the difficulty of its 3D Gaussian ellipsoids in accurately aligning with steep edges and fine components. ColMap provides a good overall roof reconstruction, but the lines are not as sharp or continuous because its mesh modeling depends on point cloud density.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-XI-2-2026-403-2026 | © Author(s) 2026. CC BY 4.0 License.

Scene	Russian Building (↑)				Modern Building (↑)			
Metrics	PSNR	Precision	Recall	F1 Score	PSNR	Precision	Recall	F1 Score
CityGaussianV2	24.22	0.557	0.522	0.539	25.68	0.650	0.391	0.488
CityGS-X	27.15	0.567	0.485	0.523	27.27	0.629	0.335	0.437
ColMap	—	**0.593**	**0.676**	**0.632**	—	**0.663**	**0.446**	**0.533**

Table 3: Quantitative results for Russian Building and Modern Building.

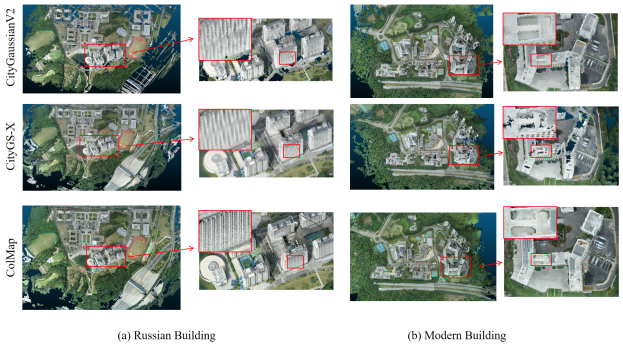

Figure 7: Visualization results of the generated geometric model.

The quantitative results in Table 3 clearly show the differences in geometric reconstruction performance. ColMap significantly outperforms the two 3DGS-derived methods in terms of F1 Score, indicating that its reconstructed model is more accurate and complete. The F1 Score is the harmonic mean of precision and recall. Specifically, while the precision of the three methods is comparable, ColMap demonstrates a significant advantage in recall, with an increase of 29.5% and 39.3% in the two scenes compared to the other methods, respectively (and 14.0% and 33.1%). Recall measures the proportion of correctly reconstructed points relative to all ground-truth points. ColMap achieves higher recall because its MVS component employs a per-pixel stereo matching strategy for systematic depth computation across all visible areas. MVS ensures comprehensive scene coverage through parallax analysis from multiple images, combined with multi-view cross-validation, guaranteeing the completeness of captured geometric elements. In contrast, the lower recall of the 3DGS methods is related to their primitive characteristics: the 2D Gaussian primitives of CityGaussianV2 are limited by their view-plane coverage and filter out low-elongation primitives, creating blind spots for stereo structures and fine components. The 3D Gaussian ellipsoids of CityGS-X struggle to accurately conform to fine components, leading to incomplete capture of ground-truth points and the lowest recall and F1 scores. As a supplement to the previous section’s tests, the PSNR of CityGS-X is noticeably higher than that of CityGaussianV2, indicating its continued significant advantage in rendering.

The geometric accuracy evaluation described above is based on the nearest neighbor distance between mesh sample points and the ground-truth point cloud. To more intuitively measure the geometric deviation between the ground-truth point cloud and the generated mesh, we used CloudCompare software to directly compute the nearest neighbor distance from the ground-truth points to the generated mesh model and create error comparison maps. As shown in Figure 8, the mesh model generated by ColMap exhibits a relatively small deviation from the ground-truth point cloud across the entire scene, with a particularly uniform error distribution on large planar surfaces of buildings. This indicates its high accuracy in overall geometric representation. In contrast, CityGaussianV2 shows smaller errors in local details, especially at the edges of buildings and in areas with complex textures, reflecting its strength in local geometric modeling. Although CityGS-X excels in rendering quality, its overall geometric deviation is slightly larger, with more noticeable errors in some fine structures and at steep edges. This is likely because its 3D Gaussian ellipsoids struggle to accurately conform to these complex geometric structures. These findings further validate the preceding analysis: traditional SfM-MVS methods still hold a significant advantage in the geometric reconstruction of large-scale scenes, particularly in terms of global consistency and completeness. While 3DGS-based methods have made remarkable progress in local accuracy and rendering efficiency, their global geometric accuracy and consistency require further improvement.

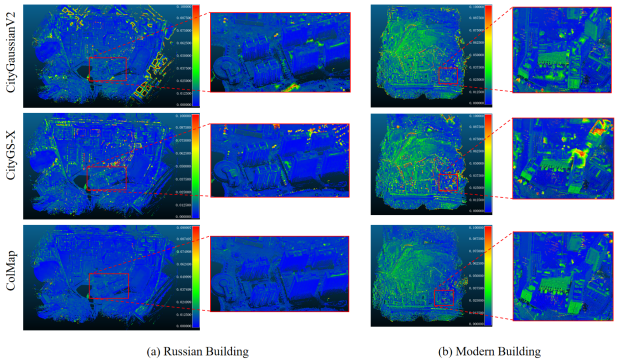

Figure 8: Error comparison maps.

5. CONCLUSION

In this paper, we addressed the performance gap between the traditional SfM-MVS framework and 3DGS-based methods in large-scale 3D reconstruction by establishing a unified validation framework. Using the real-world Mill-19 and GauU-Scene datasets, we conducted a systematic evaluation from the dual perspectives of rendering quality and geometric reconstruction. The results show that traditional methods, e.g., ColMap, still hold an advantage in the overall completeness and accuracy of geometric reconstruction. Their F1 scores and recall rates are significantly higher than those of 3DGS-derived methods, making them more suitable for applications with high demands for global consistency and reconstruction integrity. Among the 3DGS-derived methods, CityGS-X performed best on rendering quality metrics, demonstrating superior preservation of detailed textures and appearance consistency. In contrast, CityGaussianV2 showed an advantage in local geometric modeling, making it well-suited for scenarios sensitive to local geometric precision. At the same time, 3DGS-derived methods still suffer from issues such as low recall and insufficient global consistency. Future research could focus on optimizing the distribution strategy of Gaussian primitives, enhancing global geometric constraints, and exploring a deeper integration of partitioning strategies with geometric optimization. Such efforts would further balance rendering efficiency and geometric accuracy, advancing the application of 3DGS in large-scale scene reconstruction.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-XI-2-2026-403-2026 | © Author(s) 2026. CC BY 4.0 License.

ACKNOWLEDGEMENTS

This research was funded by the National Natural Science Foundation of China (Grant No. 42371442), Shenzhen Science and Technology Program (Grant No. JCYJ20250604181614019), and Shenzhen-Nanning Spatial Intelligence Joint Laboratory project (Grant No. 25220019) under the Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ).

REFERENCES

Barron, J. T., Mildenhall, B., Tancik, M., Hedman, P., Martin-Brualla, R., Srinivasan, P. P., 2021. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. *Proceedings of the IEEE/CVF international conference on computer vision*, 5855–5864.

Barron, J. T., Mildenhall, B., Verbin, D., Srinivasan, P. P., Hedman, P., 2022. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5470–5479.

Chen, J., Ye, W., Wang, Y., Chen, D., Huang, D., Ouyang, W., Zhang, G., Qiao, Y., He, T., 2025. Gigags: 3d gaussian based planar representation for large-scene surface reconstruction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39number 2, 2088–2096.

Chen, L., Wu, B., Duan, R., Chen, Z., 2024. Real-time cross-view image matching and camera pose determination for unmanned aerial vehicles. *Photogrammetric Engineering & Remote Sensing*, 90(6), 371–381.

Chen, Y., Lee, G. H., 2024. Dogs: Distributed-oriented gaussian splatting for large-scale 3d reconstruction via gaussian consensus. *Advances in Neural Information Processing Systems*, 37, 34487–34512.

Furukawa, Y., Hernández, C. et al., 2015. Multi-view stereo: A tutorial. *Foundations and trends® in Computer Graphics and Vision*, 9(1-2), 1–148.

Gao, Y., Dai, Y., Li, H., Ye, W., Chen, J., Chen, D., Zhang, D., He, T., Zhang, G., Han, J., 2024. Cosurfgs: Collaborative 3d surface gaussian splatting with distributed learning for large scene reconstruction. *arXiv preprint arXiv:2412.17612*.

Gao, Y., Li, H., Chen, J., Zou, Z., Zhong, Z., Zhang, D., Sun, X., Han, J., 2025. Citygs-x: A scalable architecture for efficient and geometrically accurate large-scale scene reconstruction. *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*.

Huang, B., Yu, Z., Chen, A., Geiger, A., Gao, S., 2024. 2d gaussian splatting for geometrically accurate radiance fields. *ACM SIGGRAPH 2024 conference papers*, 1–11.

Jiang, S., Jiang, C., Jiang, W., 2020. Efficient structure from motion for large-scale UAV images: A review and a comparison of SfM tools. *ISPRS Journal of Photogrammetry and Remote Sensing*, 167, 230–251.

Jiang, S., Li, Q., Jiang, W., Chen, W., 2022. Parallel structure from motion for UAV images via weighted connected dominating set. *IEEE Transactions on Geoscience and Remote Sensing*, 60, 1–13.

Jiang, S., Ma, Y., Jiang, W., Li, Q., 2024. Efficient structure from motion for UAV images via anchor-free parallel merging. *ISPRS Journal of Photogrammetry and Remote Sensing*, 211, 156–170.

Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G., 2023. 3D Gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4), 139–1.

Kerbl, B., Meuleman, A., Kopanas, G., Wimmer, M., Lanvin, A., Drettakis, G., 2024. A hierarchical 3d gaussian representation for real-time rendering of very large datasets. *ACM Transactions on Graphics (TOG)*, 43(4), 1–15.

Lin, J., Li, Z., Tang, X., Liu, J., Liu, S., Liu, J., Lu, Y., Wu, X., Xu, S., Yan, Y. et al., 2024. Vastgaussian: Vast 3d gaussians for large scene reconstruction. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5166–5175.

Liu, Y., Luo, C., Fan, L., Wang, N., Peng, J., Zhang, Z., 2024. Citygaussian: Real-time high-quality large-scale scene rendering with gaussians. *European Conference on Computer Vision*, Springer, 265–282.

Liu, Y., Luo, C., Mao, Z., Peng, J., Zhang, Z., 2025. Citygaussianv2: Efficient and geometrically accurate reconstruction for large-scale scenes. *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*.

Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., Ng, R., 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1), 99–106.

Sadeq, H. A., 2025. Accuracy Assessment of Dense Point Cloud Generated by Deep Learning and Semiglobal Matching. *Photogrammetric Engineering & Remote Sensing*, 91(3), 153–162.

Schonberger, J. L., Frahm, J.-M., 2016. Structure-from-motion revisited. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4104–4113.

Turki, H., Ramanan, D., Satyanarayanan, M., 2022. Mega-nerf: Scalable construction of large-scale nerfs for virtual fly-throughs. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 12922–12931.

Wu, Y., Qi, Z., Shi, Z., Zou, Z., 2025. BlockGaussian: Efficient Large-Scale Scene Novel View Synthesis via Adaptive Block-Based Gaussian Splatting. *arXiv preprint arXiv:2504.09048*.

Xiong, B., Li, Z., Li, Z., 2024. Gauu-scene: A scene reconstruction benchmark on large scale 3d reconstruction dataset using gaussian splatting. *arXiv preprint arXiv:2401.14032*.

Yang, L., Kang, B., Huang, Z., Zhao, Z., Xu, X., Feng, J., Zhao, H., 2024. Depth anything v2. *Advances in Neural Information Processing Systems*, 37, 21875–21911.

Yu, A., Ye, V., Tancik, M., Kanazawa, A., 2021. pixelnerf: Neural radiance fields from one or few images. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4578–4587.