3D Reconstruction via Depth and Normal Priors Guided 3D Gaussian Splatting

Jiaxun Jiang, Qingdong Wang*, Li Zhang*

State Key Laboratory of Spatial Datum, Chinese Academy of Surveying & Mapping, Beijing, 100036, China
xunnn123@163.com, wangqd@casm.ac.cn, zhangl@casm.ac.cn
* Denotes corresponding author

Keywords: 3D reconstruction, 3D Gaussian Splatting, Novel View Synthesis, Large-Scale Scene Reconstruction.

Abstract

This paper introduces a novel 3D reconstruction method that leverages depth and normal priors within a 3D Gaussian splatting framework. The approach aims to address the limitations of traditional 3D reconstruction methods, which often involve complex pipelines, high computational and storage demands, and detail loss. Our method begins by constructing a low-precision global Gaussian radiance field, followed by adaptive scene and data partitioning to enhance optimization efficiency while maintaining load balance. We integrate AI-powered depth and normal estimation techniques to establish geometric priors, which effectively reduce artifacts, accelerate convergence, and improve synthesis quality under sparse views. Furthermore, we propose a constraint mechanism based on the shape and opacity of Gaussians to suppress floating artifacts and enhance model robustness. Experimental results demonstrate that our method achieves better reconstruction quality, and strong generalization capabilities for large-scale 3D reconstruction.

1. Introduction

Existing 3D reconstruction methods pose several challenges in application scenarios (Schonberger et al., 2016). First, the complex pipeline, including dense matching, 3D mesh reconstruction, automated texture mapping, and so on, requires substantial computational power and is time-consuming. Second, the huge storage space needed for enormous geometry and texture data consumes significant storage capacity. Third, there is a significant loss of detail, especially in weak texture areas. Fourth, for real-time rendering, a series of cumbersome post-processing procedures, such as LOD generation, 3D mesh refinement and texture refinement, are inevitable, which are also time-consuming. All these challenges make existing 3D reconstruction pipeline hard to apply to some scenarios, such as emergency management of large-scale scene.

In recent years, novel view synthesis (NVS) has gained increasing attention. NVS has been dominated by neural neural radiance fields (NeRF) based methods in the past few years. Block-NeRF (Tancik et al., 2022) and Mega-NeRF (Turki et al., 2022) adopted the divide-and-conquer strategy in large-scene radiance field reconstruction. Due to the long training time and slow rendering speed of NeRF-based methods (Mildenhall et al., 2021; Barron et al., 2022; Niemeyer et al., 2022), 3D Gaussian splatting (3DGS) is proposed (Kerbl et al., 2023), which achieves significant improvements in training time and rendering speed. However, most existing 3DGS-based methods are designed for small scenes. VastGaussian is the first work for exploring the application of 3DGS under large-scale scenes (Lin et al., 2024). CityGaussian (Liu et al., 2024) offers an improved framework that integrates parallel training, compression, and fast rendering based on Level of Detail. However, in large-scale scene reconstruction, 3DGS-based methods still encounter issues such as high memory consumption, slow training speed, and poor reconstruction quality under sparse views.

Aiming at the problems mentioned above, this paper proposes a depth and normal priors guided 3D Gaussian Splatting to achieve efficient 3D reconstruction. The method first constructs a low-precision global 3D Gaussian radiance field as the initial representation. Subsequently, based on 3D Gaussian distribution perception technology, adaptive scene partitioning and view partitioning are realized to improve optimization efficiency while maintaining load balance. To accelerate training, real-scene 3D data and AI-powered depth and normal estimation techniques (Cao et al. 2022) are introduced to establish geometric priors based on depth and normal consistency, which effectively reduce artifacts, speed up convergence, and improve synthesis quality under sparse views. In addition, a constraint mechanism based on the shape and opacity of Gaussians is proposed to suppress floating artifacts and enhance model robustness.

2. Method

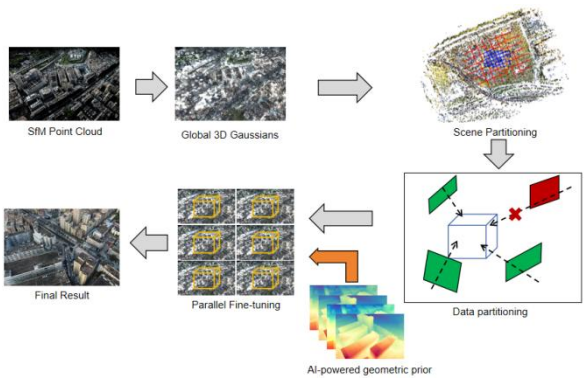

Figure 1. Overview of the proposed method.

We first generate a low-precision global 3D Gaussian radiance field based on the initial SfM points. Then a adaptive scene partitioning is performed based on the global 3D Gaussians. For each block, the data is allocated by the position and visibility. Based on the data partitioning, all the blocks are fine-tuned in parallell. During the fine-tune process, the AI-powered depth

and normal priors are introduced to enhance the quality of noval view synthesis for each block. Finally, the whole scene is achieved by fusing all the Gaussians in each block.

2.1 3D Gaussian Splatting Preliminary

The 3D Gaussian Splatting models a scene using a set of discrete 3D Gaussian primitives. Each Gaussian primitive is parameterized by 3D position p_k, opacity $o_k \in [0,1]$, some geometric properties, such as scales s_k and rotation R_k for constructing Gaussian covariance, and spherical harmonics (SH) coefficients $f_k \in R^{3\times16}$ that determine the view-dependent color c_k. During rendering, the 3D Gaussians is projected into camera space as 2D Gaussians. These 2D Gaussians are sorted by depth and then rendered by alpha-blending to generate pixel colors C .

$$C = \sum_{i=1}^{n} c_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j), \quad (1)$$

where C is the color value of a pixel. The RGB color of each Gaussian primitive, denoted as c_i, is calculated based on spherical harmonics (SH) coefficients f_i. The transparency weight α_i is determined by the projected 2D Gaussian distribution and the learned opacity of the Gaussian.

For depth map, it also can be rendered following the alpha-blending to generate pixel depth D.

$$D = \sum_{i=1}^{n} d_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j), \quad (2)$$

where the d_i is the depth value of the center point of Gaussian primitive in camera space.

2.2 Adaptive Scene and Data Partitioning

In large-scale 3D reconstruction, adaptive scene partitioning is vital for efficient computational resource management and high-quality results. The conventional divide-and-conquer approach involves splitting the scene into smaller sub-regions, independently optimizing each, and merging the results. However, this method faces challenges like uneven point cloud distribution, occlusions, and computational load imbalance.

Partitioning based on SFM sparse point clouds is a common approach but has limitations. While it provides a basic scene structure, it fails to fully capture the scene's radiance field Gaussian distribution, which is crucial for accurate reconstruction. In this paper, we employ a global Gaussian distribution based scene partitioning. This approach offers a more accurate basis for dividing the scene into manageable parts.

2.2.1 Low-Precision Global 3D Gaussian Radiance Field: To overcome hardware limitations that can cause failures in global Gaussian construction, a random view sampling policy is employed to construct a low-precision global Gaussian radiance field progressively.

The Gaussians is initializd with the SfM sparse point cloud. Due to the GPU memory constraints, directly building a high-precision global Gaussian radiance field is impractical. Instead, we use a progressive approach with random view sampling. In each iteration, we randomly select 500 images from the dataset to optimize the Gaussians, gradually refining the parameters of Gaussians over time. This method efficiently utilizes memory and computational resources, as it avoids loading all images into memory at once.

We conduct approximately 8,000 iterations to balance model accuracy and resource usage. Each iteration updates the Gaussian parameters based on the sampled images, incrementally enhancing the radiance field. The iterative process allows the gradual refinement of the Gaussian radiance field, improving the quality of the reconstruction step by step.

2.2.2 Adaptive Scene Partitioning in Large-Scale 3D Reconstruction: The process of splitting and pruning 3D Gaussians, often used to refine the representation of the scene, inherently changes their distribution compared with the initial SfM sparse points. This dynamic distribution can be leveraged to partition the scene adaptively, ensuring that each partition contains a balanced and manageable number of Gaussians.

The 3D space is divided into a voxel grid, with each Gaussian assigned to a voxel based on its center coordinates. The density of Gaussians within each voxel is calculated to determine the complexity of the scene in that region.

The scene is recursively partitioned into sub-blocks using a binary tree structure based on the density. Blocks with higher Gaussian density are divided into smaller sub-blocks. The partitioning process continues until the number of Gaussians within each block falls below a specified threshold. This ensures that each block has a balanced computational load while adapting to the varying density of the Gaussians.

2.2.3 Data Partitioning: For each block, we hope there are sufficient supervision during optimization. However, different views contribute variably to a block or partition, simply using the views that fall within a block as supervision is often insufficient, which can lead to several issues: 1) Incomplete scene representation, The views within a block may not capture the full context of the scene, leading to incomplete supervision and potential reconstruction errors. 2) Occlusions and artifacts: Intra-block views may not account for occlusions or complex geometry, resulting in artifacts and inconsistencies in the final reconstruction. 3) Limited detail: Restricting supervision to intra-block views can limit the level of detail captured, especially in areas where the block boundaries intersect with significant scene features.

Visibility-based allocation addresses this by ensuring that each block receives image data that is directly relevant to the content within that block. For each block, determine which cameras have a clear line of sight to the Gaussians within that block. This involves projecting the block's Gaussians into the camera views and checking for visibility. Based on visibility calculations, select the most relevant views for each block. These views should provide the best supervision for the Gaussians within the block. Consider the number of visible Gaussians and the quality of the view. Assign different weights to views based on their relevance and contribution to the reconstruction. Views with more visible Gaussians tend to contribute more significantly to the block, which can be given higher weights.

In addition to evaluating the contribution of different views within a block, we also select views outside the block in a similar way. By carefully choosing these external views, we can gather more comprehensive information, which helps to improve the accuracy and completeness of the scene reconstruction. This approach ensures that both internal and external views contribute effectively to the final result, leading to a more detailed and precise 3D model.

After the data partitioning is completed, each partition (block) can be fine-tuned based on the global Gaussian radiance field.

2.3 AI-powered Geometric Regularization

Introducing depth and normal regularization is crucial for the fine-tuning of each block. These geometric priors help mitigate artifacts, improve geometric accuracy, and enhance quality of novel view synthesis under insufficient supervision conditions, especially in areas with sparse input views.

2.3.1 Depth Regularization: Depth maps provide precise depth information, which helps in accurately placing the Gaussians where the objects are. We introduce depth regularization by Monocular depth estimation. In our work, a pre-trained depth estimator, depth anything v2, is selected. Different from prior work (Zhu et al., 2024), which align scale between estimated depths and the scene by comparing the estimated depths with the SfM sparse points. We employ Pearson correlation similarity for soft depth supervision. The Pearson correlation coefficient (Cohen et al., 2009) is both scale-invariant and shift-invariant, which makes it well-suited for evaluating the similarity of depth maps with different scales.

$$\mathcal{L}_{depth} = \mathrm{Corr}(D_r, D_e) = \frac{\mathrm{Cov}(D_\mathrm{r}, D_\mathrm{e})}{\sqrt{\mathrm{Var}(D_\mathrm{r}) \cdot \mathrm{Var}(D_\mathrm{e})}}, \quad (3)$$

where D_e donates the monocular depth estimated by the prior estimator, D_r is the rendered depth.

2.3.2 Normal Regularization: Normal maps offer detailed information about the orientation of surfaces. The orientation information helps in refining the shape and appearance of Gaussians. In our work, we estimate the normal map from estimated depth map D_e as a supervisory signal for the scene by calculating its gradient. Compared to using pre-trained models for normal estimation, which involves significant complexities and resource demands, the gradient of depth map offers a more efficient alternative for normal map acquisition. These normals guide the shape of Gaussian ellipsoids to better conform to realistic surface geometries, thereby enhancing both scene smoothness and geometric details. For a pixel (u,v) in a depth map D, the normal vector can be computed as:

$$N(u,v) = \frac{\nabla D(u,v)}{\|\nabla D(u,v)\|}, \quad (4)$$

where $\nabla D(\cdot)$ is the gradient of depth, we use an L1 loss directly during training to enforce this supervision effectively.

$$\mathcal{L}_{normal} = \mathcal{L}_1(N, N_e), \quad (5)$$

where N represents the normal from the gradient of Gaussian-rendered depth map, N_e represents the normal from the gradient of the depth map of depth estimator.

Overall, our geometric regularization framework incorporates two key components: monocular depth regularization, monocular normal regularization. The overall composite loss function is formally defined as follows:

$$\mathcal{L}_{geo} = \lambda_1 \mathcal{L}_{depth} + \lambda_2 \mathcal{L}_{normal}, \quad (6)$$

where λ_1, λ_2 represent the weights of each loss respectively.

2.4 Gaussian Primitive Constraints

In large-scale scenes, some areas inevitably have sparse image data, and scene partitioning and data partitioning may increase the probability of this situation. In this section, we first analyze the behavior of Gaussians in such cases. Then, we introduce constraint strategies for scale and opacity to further address the challenges of Gaussian optimization when supervision is limited.

2.4.1 Shape Constraints: In the standard Gaussian framework, the Gaussians with too large or small scale are eliminated or split during densification iterations. However, this approach proves to be unreliable in some low image overlap scenarios.

In areas with low image overlap, the optimization process is prone to generate the overly elongated and oversized Gaussian ellipsoids to cover larger areas. Moreover, due to the lack of sufficient supervision, these elongated and oversized Gaussians can not be effectively split into smaller one during the adaptive densification process. These Gaussians severely degrade the quality of novel views synthesis. To detect and remove these abnormal Gaussians, we introduce a abnormal Gaussian pruning strategy that detect oversized Gaussians by their longest axis and detect the overly elongated Gaussians by the ratio between the longest and second-longest axes. The criterion is defined as follows:

$$G_{ir} = \begin{cases} \frac{s1}{s2} > r \\ s1 > t \end{cases}, \quad (7)$$

here, s1 and s2 represent the lengths of the longest and second-longest axes of the Gaussian ellipsoid, respectively. r is the threshold for alongated Gaussians detection, and t is the threshold for oversized Gaussians detection.

The strategy is applied after a certain number of iterations during the training process.

2.4.2 Opacity Constraints: In the standard Gaussian framework, to prevent the optimization process from getting stuck in local optima, the opacities of all Gaussians are periodically reset during training. And a minimum opacity threshold is applied throughout the training to remove Gaussians that have opacities below this threshold, which helps in eliminating artifacts and unnecessary Gaussians, improving the overall quality and efficiency of the model.

However, this strategy encounters challenges in partitioned scenes. The limited training images and minimal angle variation within each block lead to insufficient supervision. This causes some Gaussian opacities to get stuck in local optima, staying above predefined threshold. These translucent Gaussians persist near true surfaces, introducing floating artifacts and generating blurred depth maps that weakening geometric regularizations and lowering rendering quality.

To address this issue, we dynamically adjust the opacity threshold.

$$O_{T+1} = O_T + \Delta O, \quad (8)$$

here, O_{T+1}, O_T represents the current opacity threshold and previous opacity threshold, while ΔO denotes the increment in opacity value applied after each interval of iterations.

3. Experiments

3.1 Experimental Setup

Datasets. Our method has been rigorously evaluated across three real-world datasets: Mill19 (Turki et al., 2022), LLFF (Mildenhall et al., 2019), DTU (Jensen et al., 2014) . The Mill19 dataset consists of aerial images captured by real-world drones, with each scene containing thousands of high-resolution images. In both the training and testing phases, we maintained the same dataset partitioning as Mega-NeRF (Turki et al., 2022). To ensure a fair comparison across all experiments, we uniformly applied a $4\times$ downsampling to each image, following the approach of previous studies. The LLFF dataset features 8 intricate forward-facing scenes. Following previous research (Niemeyer et al., 2022), every eighth image is chosen for testing, while the remaining images are used to uniformly sample sparse training views. The DTU dataset comprises numerous object-centric scenes, and we have selected 15 of them, adhering to the same dataset split protocol. During evaluation, background regions are masked. The image resolutions for the LLFF, DTU are 1/8, 1/4 respectively.

We use PSNR, SSIM, and LPIPS metrics to quantitatively evaluate the quality of reconstruction. Higher values of PSNR and SSIM signify superior reconstruction quality, while lower LPIPS scores reflect higher perceptual fidelity.

BaseLines. We choose some state-of-the-art synthesis methods for comparison, including RegNeRF (Niemeyer et al., 2022) , FreeNeRF (Yang et al., 2023) and SparseNeRF (Wang et al., 2023), 3DGS (Kerbl et al., 2023), FSGS (Zhu et al., 2024), CityGaussian (Liu et al., 2024), VastGaussian (Lin et al., 2024). For most baselines, we directly cite their best quantitative result in their respective papers. For the 3DGS, we employ the results from our own implementation.

Implementation Details. Our model is constructed based on the official PyTorch implementation of 3D Gaussian Splatting. Throughout all datasets, we conduct a total of 10,000 training iterations. For the LLFF and DTU scenes, we set the adaptive sampling rate within the range of 0.05 to 0.25. In terms of geometric regularization, we utilize the Depth Anything V2 model, which is capable of predicting monocular depth for both small-scale and large-scale scenes from images. This regularization weights λ_1, λ_2 are set to 0.05. The Gaussian shape constraints are activated at the 8,000th iteration, with the parameters r and t set to 4 and 1.1. Regarding the opacity constraint, the ΔO is set to 0.1. All experiments were carried out using NVIDIA 4090 GPUs.

3.2 Comparsion with Other Methods

Mill19 Datasets. Compared to existing methods, our method achieves comparable performance with CityGS, and outperforms VastGS, 3DGS and Mega-NeRF in all scenes. Compare with standard 3DGS, our method can capture more high-frequency details. In contrast to CityGaussian (CityGS), our method results in fewer artifacts, the quality of image synthesis has been further enhanced, which primarily due to the depth and normal regularization employed in this paper. Although our method achieves the best PSNR in both scene, the SSIM is a little bit lower than CityGS in residence sence and LPIPS is higher than CityGS in Rubble. We attribute this to the AI-based depth estimation. While it effectively reduces artifacts and boosts visual quality, there is still estimation errors in certain specific scenes, which might slightly affect the reconstruction of some fine texture. Overall, the improvement from the geometric regularization demonstrates that current depth estimation pre-trained models (depth Anything) have already exceeded our expectations.

Figure.1 shows a qualitative results. The standard 3DGS lost many details in the scene. In contrast to the CityGS, although it captures much richer details, it also exhibits some obvious artifacts, especially in the regions with dense buildings, where exists severe occlusions. Relatively, the rendering quality of our method are closest to the ground truth.

Method	Residence			Rubble		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Mega-NeRF	22.08	0.628	0.401	24.06	0.553	0.508
3DGS	20.82	0.769	0.254	24.73	0.772	0.284
VastGS	21.01	0.699	0.261	25.20	0.742	0.264
CityGS	22.00	**0.813**	0.211	25.77	0.813	**0.228**
Ours	**22.61**	0.810	**0.207**	**25.88**	**0.815**	0.229

Table1. Quantitative Comparison on large-scale scene dataset Mill19.

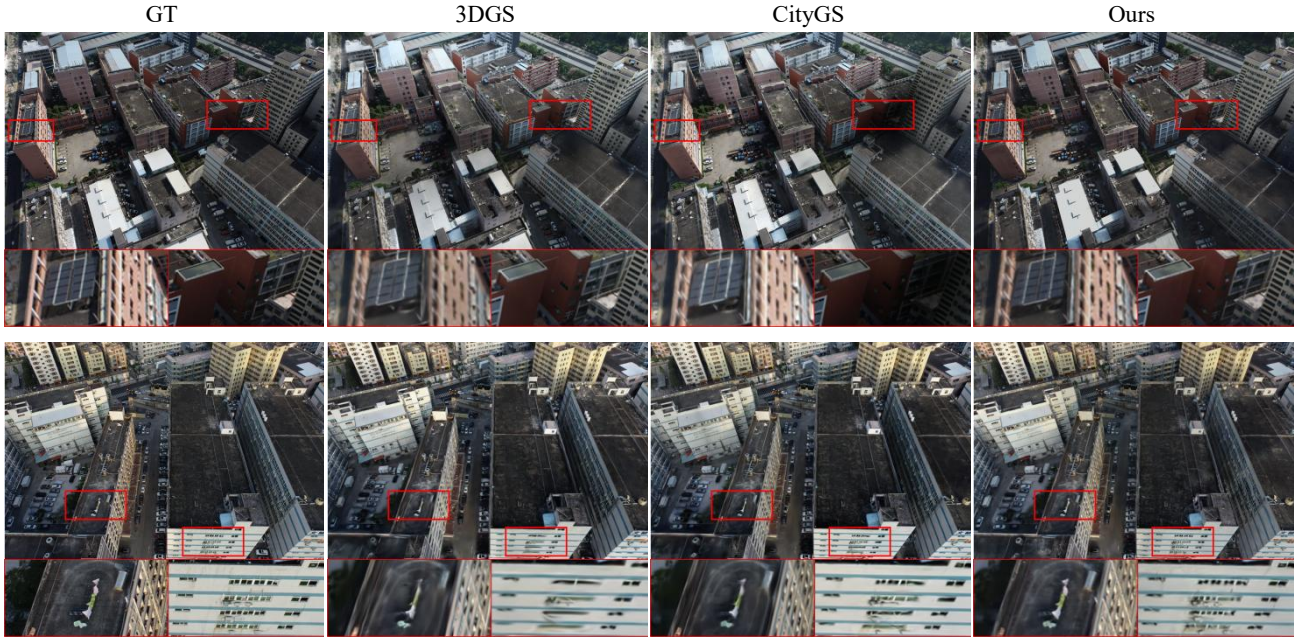

Figure 1. Qualitative comparison with SOTA methods on Mill19 dataset

LLFF Datasets. The quantitative result for the LLFF dataset are presented in Table 2, demonstrating that our method achieves the best performance in PSNR, SSIM, and LPIPS metrics. Notably, our method substantially surpasses NeRF-based method for sparse novel view synthesis under same sparse input conditions. Compared to FSGS, our method achieves an improvement of 0.96 dB in PSNR.

Method	LLFF		
	PSNR ↑	SSIM ↑	LPIPS ↓
RegNeRF	19.08	0.587	0.336
FreeNeRF	19.63	0.612	0.308
SparseNeRF	19.86	0.624	0.328
3DGS	14.27	0.398	0.420
FSGS	20.31	0.652	0.288
Ours	**21.27**	**0.750**	**0.166**

Table 2. Quantitative comparison on LLFF datasets.

DTU Datasets. The quantitative result for the DTU dataset are also presented in Table.3, affected by our geometric regularization, our method outperforms the baselines across all evaluation metrics. Specifically, when compared to the 3DGS method, we observe a substantial enhancement in our PSNR score by 7.71 dB, and our SSIM score shows an improvement of 0.162.

Method	DTU		
	PSNR ↑	SSIM ↑	LPIPS ↓
RegNeRF	18.89	0.745	0.190
FreeNeRF	19.92	0.787	0.182
SparseNeRF	19.55	0.769	0.201
3DGS	15.13	0.734	0.214
Ours	**22.84**	**0.896**	**0.083**

Table 3. Quantitative comparison on DTU datasets

Figure.2 shows a qualitative results. 3DGS struggles to effectively reconstruct scenes with only sparse 3-view inputs. For FSGS, despite it achieves better reconstruction quality, it falls short in capturing fine details. In contrast, our method not only provides enhanced rendering quality but also successfully reconstruct more intricate textures and geometric details.

Furthermore, we show the results of novel view depth map synthesis on the LLFF dataset. As depicted in Figure. 3, the depth maps rendered by 3DGS are quite poor. the results of FSGS are better, but compare with our method, its depth maps have more noise, the depth map rendered by our method is smoother.

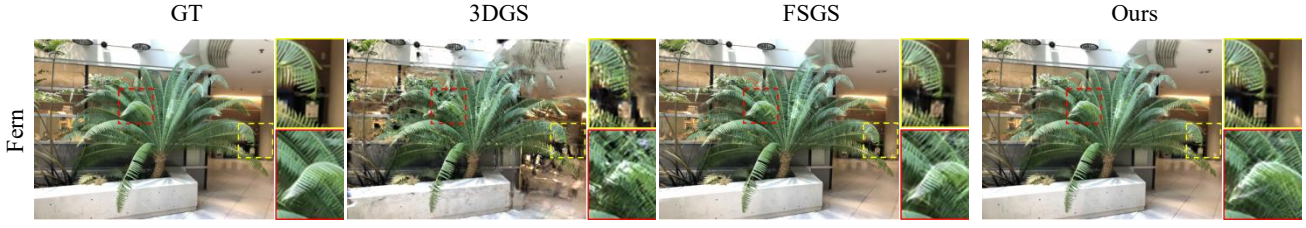

Figure 2. Qualitative comparison on LLFF dataset.

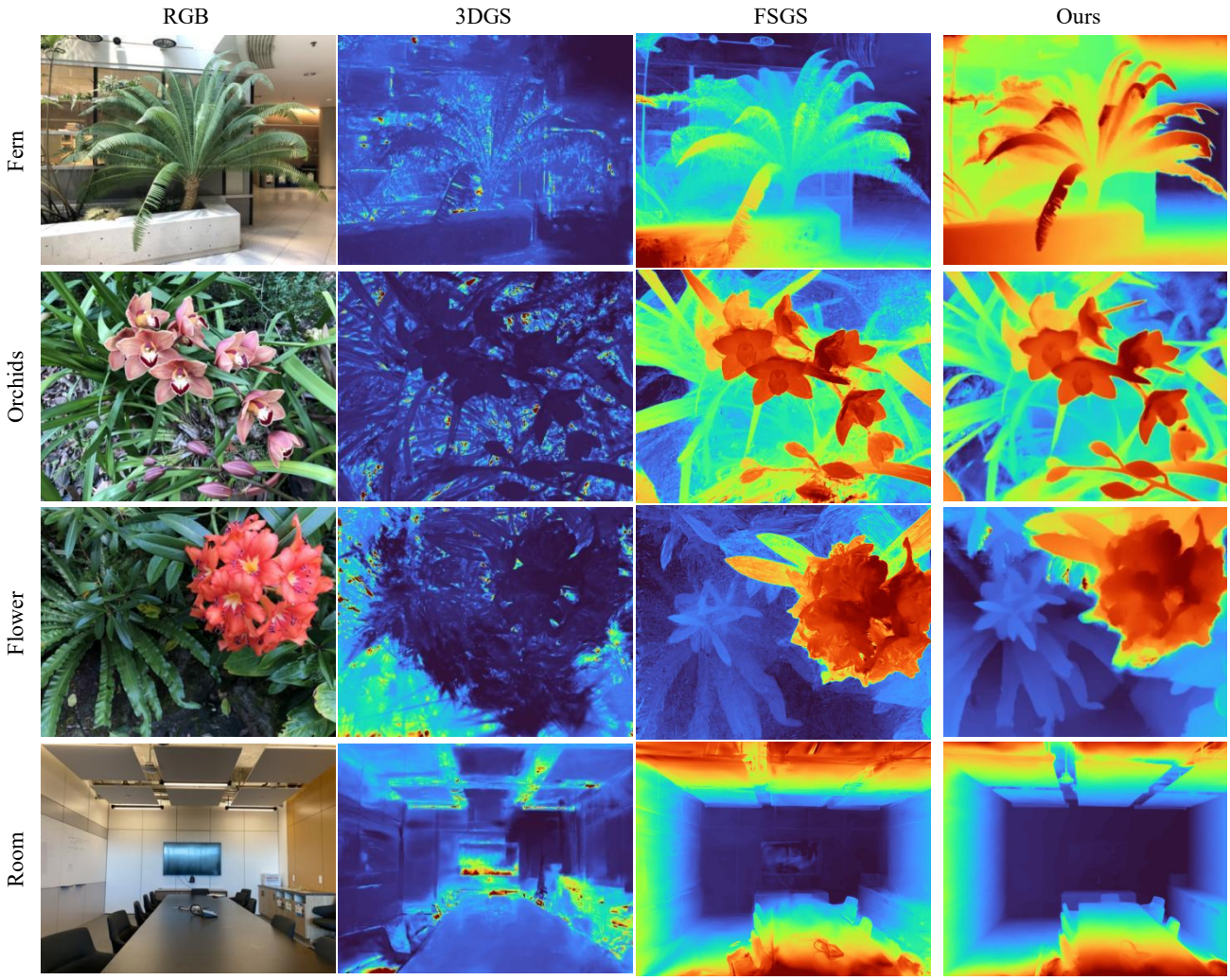

Figure 3. Depth map comparisons for Novel Views on LLFF dataset.

3.3 Ablation

We conducted our ablation experiments on both Mill19 and LLFF datasets to evaluate the geometry regularization. The results without geometric and with geometric regularization on dataset Mill19 are presents in Table 4. The second row of Table 4 highlights the enhancements achieved through geometric regularization. This component guide the positions and shapes of Gaussians to be more closely approximate the real surface structures. As depicted in Figure 4, the depth maps generated under geometric regularization (4th column) are closer to the ground truth provided by the depth estimator and are also smoother and low noisy, compared to the baseline.

Geometric regularization	PSNR ↑	SSIM ↑	LPIPS↓
w/o	22.36	0.807	0.209
w	22.61	0.810	0.207

Table. 4 Ablation experiments for large-scale scene on Mill19.

To further validate the effectiveness of our geometric regularization, we conduct an ablation experiments on LLFF dataset at 1/8 resolution with 3-view input setting, each component within the geometric regularization, as shown in Table 5.

From Table.5, we can find that the depth regularization obtains 0.19 improvement in PSNR, 0.001 improvement in SSIM, but obtains 0.003 degradation in LPIPS, which may be caused by the error from depth estimation. After turning on the normal regularization, a further improvements are achieved, 0.24 in PSNR, 0.11 in SSIM and 0.01 LPIPS, which means the normal regularization can improve the stablity of oprimization.

Depth	Normal	PSNR↑	SSIM↑	LPIPS↓
✗	✗	20.84	0.738	0.173
✓	✗	21.03	0.739	0.176
✓	✓	**21.27**	**0.750**	**0.166**

Table. 5 Ablation experiments about geometric regularization on LLFF.

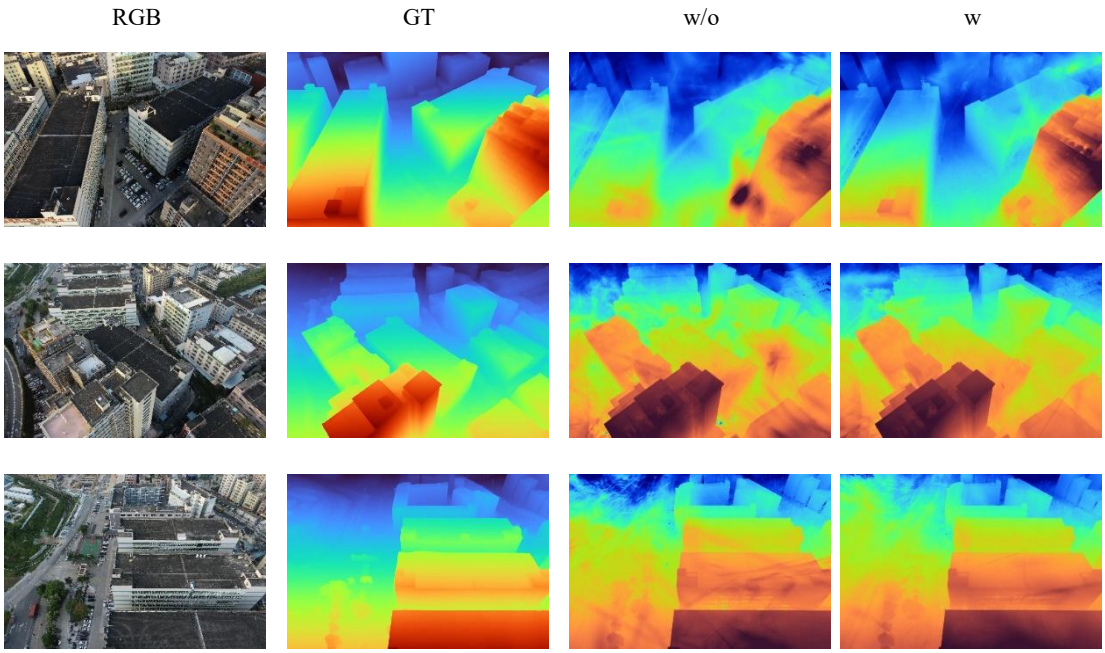

Figure 4. Depth map rendering comparisons for large-scale scene on Mill19 dataset.

4. Conclusions

We propose a depth and normal priors guided 3D Gaussian Splatting for large-scale scene reconstruction. Through the adaptive scene and data partitioning, we achieved the generation of Gaussian radiance field for large-scale scenes. To enhance the quality of view synthesis for each scene block, AI-powered depth and normal priors are incorporated. Additionally, to prevent overfitting in scenes with low image overlap, Gaussian primitive constraints are also employed to remove abnormal Gaussian primitives. Experiments demonstrate that our method achieves superior performance across multiple datasets.

Limitations and future work. As experiments demonstrate, our geometric priors are prone to be impacted by the depth estimator. In future work, we will focus on exploring a more accurate depth estimation method for large-scale scene to enhance the quality of novel view synthesis.

Acknowledgements

This research was supported by the National Key Research and Development Program of China (Grant No. 2024YFC3015600).

References

Barron, J. T., Mildenhall, B., Verbin, D., Srinivasan, P. P., & Hedman, P., 2022. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In Proceedings of the IEEE/CVF Conference.

Cao, C., Ren, X., Fu, Y., 2022. Mvsformer: Learning robust image representations via transformers and temperature-based depth for multi-view stereo. ArXiv Prepr.

Cohen, I., Y. Huang, J. Chen, and Benesty, J., 2009. Pearson Correlation Coefficient. *In Noise Reduction in Speech Processing*, 1 – 4.

Liu, Y., Guan, H., Luo, C., Fan, L., Wang, N., Peng, J. and Zhang, Z., 2024. CityGaussian: Real-Time High-Quality Large-Scale Scene Rendering with Gaussians. arXiv. https://doi.org/10.48550/arXiv.2404.01133.

Lin, J., Li, Z., Tang, X., Liu, J., Liu, S., Liu, J., Lu, Y., Wu, X., Xu, S., Yan, Y. and Yang, W., 2024. Vastgaussian: Vast 3D Gaussians for Large Scene Reconstruction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 5166-5175.

Jensen, R., Dahl, A., Vogiatzis, G., Tola, E. and Aanæs, H., 2014. Large Scale Multi-View Stereopsis Evaluation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 406 – 413.

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 – 139:14.

Mildenhall, B., Srinivasan, P. P., Ortiz-Cayon, R., Kalantari, N. K., Ramamoorthi, R., Ng, R. and Kar, A., 2019. Local Light Field Fusion: Practical View Synthesis with Prescriptive Sampling Guidelines. *ACM Transactions on Graphics (TOG)* 38: 1 – 14.

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. *Commun. ACM* 65:99 – 106.

Niemeyer, M., Barron, T. J., Mildenhall, B., Sajjadi, M. S., Geiger, A. and Radwan, N., 2022. Regnerf: Regularizing Neural Radiance Fields for View Synthesis from Sparse Inputs. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 5470 – 5480.

Schonberger, J.L., Frahm, J.-M., 2016. Structure-from-motion revisited. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 4104 – 4113.

Tancik, M., Casser, V., Yan, X., Pradhan, S., Mildenhall, B., Srinivasan, P. P., Barron, J. T. and Kretzschmar, H., 2022. Block-nerf: Scalable Large Scene Neural View Synthesis. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 8248 – 8258.

Turki, H., Ramanan, D. and Satyanarayanan, M., 2022. 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, 12,922 – 12,931.

Wang, G., Chen, Z., Loy, C. C. and Liu, Z., 2023. Sparsenerf: Distilling Depth Ranking for Few-Shot Novel View Synthesis. In Proceedings of the IEEE/CVF International Conference on Computer Vision, 9065 – 9076.

Yang, J., Pavone, M. and Wang, Y., 2023. Freenerf: Improving Few-Shot Neural Rendering with Free Frequency Regularization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 8254 – 8263.

Zhu, Z., Fan, Z., Jiang, Y. and Wang, Z., 2024. Fsgs: Real-time Few-shot View Synthesis Using Gaussian Splatting. In European Conference on Computer Vision, 145 – 163.