

Optimizing 3D Gaussian Splatting for Sparse Viewpoint Scene Reconstruction

Shen Chen¹, Jiale Zhou^{1,*} and Lei Li²

Abstract—3D Gaussian Splatting (3DGS) has emerged as a promising approach for 3D scene representation, offering a reduction in computational overhead compared to Neural Radiance Fields (NeRF). However, 3DGS is susceptible to high-frequency artifacts and demonstrates suboptimal performance under sparse viewpoint conditions, thereby limiting its applicability in robotics and computer vision. To address these limitations, we introduce SVS-GS, a novel framework for Sparse Viewpoint Scene reconstruction that integrates a 3D Gaussian smoothing filter to suppress artifacts. Furthermore, our approach incorporates a Depth Gradient Profile Prior (DGPP) loss with a dynamic depth mask to sharpen edges and 2D diffusion with Score Distillation Sampling (SDS) loss to enhance geometric consistency in novel view synthesis. Experimental evaluations on the MipNeRF-360 and SeaThru-NeRF datasets demonstrate that SVS-GS markedly improves 3D reconstruction from sparse viewpoints, offering a robust and efficient solution for scene understanding in robotics and computer vision applications.

I. INTRODUCTION

The use of RGB cameras in robotic vision systems for 3D scene reconstruction is essential for acquiring multiple viewpoints, a fundamental requirement for high-quality novel view synthesis (NVS). However, in practical scenarios, obtaining dense multi-view data is often impractical, especially in resource-constrained or complex environments. This limitation necessitates developing methods that can achieve effective scene reconstruction from sparse viewpoints. Traditional Neural Radiance Fields (NeRF) [1]–[3] have shown strong performance in NVS, but their pixel-level ray rendering is computationally intensive and not well-suited for scenarios with sparse input data, requiring substantial resources and processing time.

In contrast, 3D Gaussian Splatting (3DGS) [4] employs an explicit representation that significantly reduces both training and rendering times while maintaining high-quality outputs. This method initializes a set of 3D Gaussians from point clouds generated by Structure from Motion (SfM) [5] or via random initialization. It uses adaptive density control to clone and prune these Gaussians, enhancing scene detail representation. Leveraging the smooth, differentiable properties of Gaussian distributions, 3DGS enables rapid rasterization by projecting 3D Gaussians onto 2D image planes, supporting efficient rendering and interpolation [4], [6], [7].

3D Gaussian distributions effectively capture details across multiple scales, and their projection onto a 2D plane simplifies

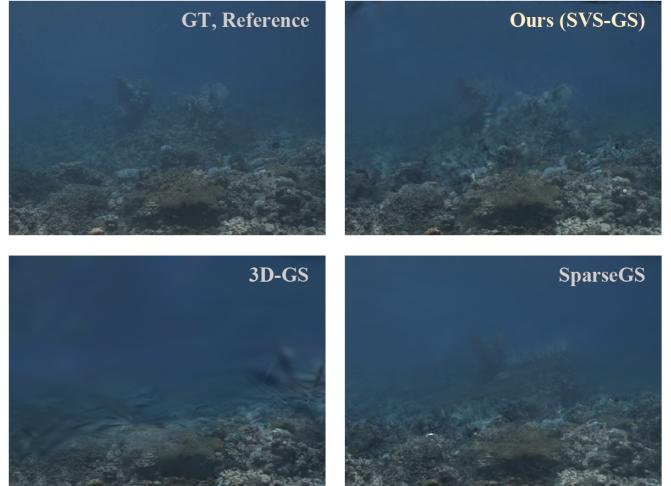


Fig. 1. We propose a sparse Viewpoint scene Reconstruction framework. Comparison of 3DGS [4] and SparseGS [8] with our SVS-GS trained on 8 views shows that SVS-GS outperforms the other methods in synthesizing close-up scenes.

the rasterization process. While this method is capable of efficiently representing complex, large-scale scenes or objects, the absence of size constraints for each 3D Gaussian primitive leads to a loss of detail when reconstructing fine objects, especially upon zooming in. This limitation is particularly evident when dealing with extremely thin lines, where it can result in inaccuracies that hinder the precise capture and reproduction of slender structures and small features, thereby compromising the overall visual realism and detail fidelity of the scene [9], [10]. Moreover, in practical applications, 3D Gaussian Splatting (3DGS) requires densely sampled multi-view scenes to achieve optimal results [11]–[13]. However, obtaining such extensive viewpoint data is often impractical in resource-constrained or complex environments. The unconstrained size of primitives in 3DGS and the reliance on dense multi-view image data present significant challenges for practical applications, such as autonomous vehicle navigation.

3DGS methods are heavily dependent on the density and quality of initial point clouds derived from dense multi-view inputs, which limits their effectiveness in sparse-viewpoint scenarios. To address the inherent limitations of 3DGS, we propose a sparse-view 3DGS framework, termed SVS-GS. To impose size constraints on the 3D Gaussian primitives, we introduce a 3D smoothing filter [10]. This filter regulates the diffusion range of Gaussian primitives in both 3D space and their 2D projections, ensuring the preservation of

¹East China University of Science and Technology, zhou.jiale@ecust.edu.cn

²University of Washington, lilei@di.ku.dk

*Corresponding author.

more details during reconstruction, particularly for small and thin structures. In standard 3DGS, the initial 3D Gaussian primitives are derived from point cloud data generated by COLMAP [5], [14]. However, sparse views yield a limited number of initial points, resulting in low point cloud density, which adversely affects the distribution and quality of Gaussian primitives. To enhance the density of these initial 3D Gaussian primitives, we introduce a local adaptive density scaling module. This module dynamically increases the density of Gaussian primitives based on the sparse point clouds, producing a denser set of 3D Gaussian primitives.

For the optimization of the 3D Gaussian primitives, we employ score distillation sampling (SDS) loss [15] to integrate 3DGS with 2D diffusion, incorporating depth prior information to constrain the positions and sizes of the 3D Gaussian primitives. Additionally, we introduce a dynamic depth mask and Gradient Profile Prior (GPP) loss [16] to enhance the sharpness of edges in the depth maps. SVS-GS effectively addresses gaps in the sparse point cloud data while simultaneously improving the uniformity and spatial coverage of the initial Gaussian primitives, thereby enhancing precision and detail fidelity in 3D scene reconstruction.

Our main contributions are as follows:

- **Novel Sparse-View Framework:** SVS-GS reduces dependency on dense multi-view data by optimizing Gaussian primitive distributions, improving practicality and efficiency.
- **Adaptive Density Scaling:** A local adaptive density scaling module generates denser initial 3D Gaussian primitives, addressing the problem of sparse point clouds.
- **Enhanced Optimization Techniques:** Integration of SDS loss with 2D diffusion, dynamic depth masks, and depth priors ensures precise control over Gaussian primitives, improving detail reconstruction.

A. Novel View Synthesis

Implicit representations for novel view synthesis (NVS), particularly Neural Radiance Field (NeRF)-based methods, have gained substantial attention in recent years [1], [15], [18]–[20]. NeRF [1], [18] utilizes a multi-layer perceptron (MLP) [21], [22] to predict radiance and density at 3D locations and viewing directions, leveraging classical volume rendering techniques [23] to generate high-quality novel views. Despite their strengths, these methods can produce artifacts when handling high-frequency details. To address this, Mip-NeRF [2] introduces multi-scale features and anti-aliased conical frustums to minimize blurring. While NeRF-based approaches are effective for objects and small-scale scenes, inaccuracies in camera parameters can accumulate errors in large-scale, unbounded environments, affecting reconstruction quality. Mip-NeRF 360 [24] alleviates these issues with non-linear scene parameterization and online distillation techniques to reduce artifacts in large-scale scenes.

In scenarios with sparse input views, NeRF models are prone to overfitting, which limits their ability to generalize to novel perspectives [25], [26]. Several methods have been

proposed to enhance reconstruction accuracy in such settings. Depth-Supervised NeRF (DSNeRF) [20] combines color and depth supervision to produce more detailed scenes, while SPARF [27] uses pixel matching and depth consistency loss to achieve high-precision 3D scene generation from sparse inputs.

B. Primitive-Based Rendering

Primitive-based rendering techniques, which rasterize geometric primitives onto a 2D plane, have gained widespread adoption due to their high efficiency [28]–[30]. Differentiable point-based rendering methods [31], [32] are particularly effective for novel view synthesis (NVS) because they offer optimization-friendly representations of complex scene structures. Recently, the introduction of 3D Gaussian Splatting (3DGS) [4] has renewed interest in explicit representation methods. Unlike implicit representations, explicit representations directly encode the geometry and lighting information of a scene, reducing computational complexity. However, 3DGS adapts Gaussian primitives to each training image independently, often neglecting the global structural coherence of the scene [33]. Additionally, the lack of size constraints during training can lead to artifacts in rendered novel views. To address these issues, Structured 3D Gaussians (Scaffold-GS) [33] introduces anchor points to guide the distribution of 3D Gaussian primitives, enhancing the structural integrity of the scene. Mip-Splatting [10] further improves 3DGS by incorporating a 3D smoothing filter and a 2D mipmap filter to constrain the size of Gaussian primitives, thereby capturing finer scene details.

Most 3DGS-based methods initialize using point clouds generated from Structure-from-Motion (SfM) techniques, such as COLMAP. These methods rely on dense input images to maintain sufficient point cloud density, which is crucial for high-quality scene reconstruction. When the input images are sparse, the resulting point clouds also become sparse, limiting the capacity of 3D Gaussian primitives to capture intricate geometric details during generation and optimization [12], [34]. This sparsity can cause the models to overfit to the limited training views, thereby hindering generalization to novel viewpoints and reducing the effectiveness of scene reconstruction. SparseGS [8] attempts to mitigate the dependency on dense input by incorporating 2D diffusion and depth information.

II. PRELIMINARIES

3DGS employs anisotropic Gaussians to effectively capture the varying scales and orientations present within a scene. Each 3D Gaussian primitive, denoted as $\{\mathcal{G}_n \mid n = 1, \dots, N\}$, is characterized by several parameters: a center position $\mu_n \in \mathbb{R}^{3 \times 1}$, a covariance $\Sigma_n \in \mathbb{R}^7$, a color $c_n \in \mathbb{R}^3$, and an opacity $\alpha_n \in \mathbb{R}^1$. The Gaussian function is defined as:

$$\mathcal{G}_n(x) = e^{-\frac{1}{2}(x-\mu_n)^T \Sigma_n^{-1} (x-\mu_n)}, \quad (1)$$

where x denotes points queried around the center position μ_n . The size and orientation of each 3D Gaussian primitive are determined by the semi-definite parameters $\Sigma_n =$

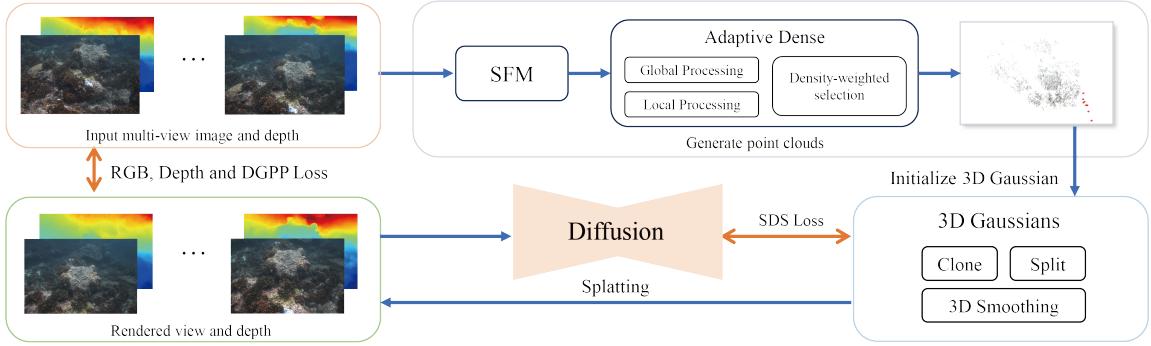


Fig. 2. Overall framework. Starting with multi-view images and corresponding depth maps (obtained from Monocular Depth Estimation Models [17]), point clouds are generated by SfM and undergo adaptive density processing to optimize the density distribution of the point clouds. The point clouds are initialized as 3D Gaussian distributions and further refined through operations such as RGB, depth, and DGPP Loss. The SDS loss function is integrated to ensure geometric consistency and reduce noise.

$R_n S_n (R_n S_n)^T$, where $R_n \in \mathbb{R}^4$ represents a rotation matrix and $S_n \in \mathbb{R}^3$ is a scaling matrix.

To render images from different viewpoints, differential splatting is applied to project the 3D Gaussians onto camera planes. This process involves the viewing transformation W_n and the Jacobian matrix J_n , resulting in a transformed covariance:

$$R'_n = J_n W_n \Sigma_n (J_n W_n)^T. \quad (2)$$

For color construction, 3DGS utilizes spherical harmonics to model the color c_n of each Gaussian, incorporating its opacity α_n . When rendering from a novel viewpoint, the 3D Gaussians are projected onto 2D planes, and the resulting color $C_r(x)$ for a given ray r is computed as:

$$C_r(x) = \sum_{i \in M} c_i \sigma_i \prod_{j=1}^{i-1} (1 - \sigma_j), \quad \sigma_i = \alpha_i G_i^{2D}(x), \quad (3)$$

where c_i and α_i represent the color and opacity of the i -th Gaussian, respectively. Here, the ray r originates from the camera center corresponding to the observation viewpoint. Finally, an adaptive density control mechanism is implemented to dynamically clone and prune the 3D Gaussians, maintaining a balance between computational efficiency and scene detail.

III. METHODS

A. Problem Formulation

In the context of scene reconstruction, optimizing the initialized 3D Gaussian primitives necessitates a set of multi-view images $I = \{I_1, I_2, \dots, I_k\}$ and the corresponding point clouds $P = \{p_1, p_2, \dots, p_M\}$. The multi-view images I are first utilized to generate an initial point cloud P through Structure-from-Motion (SfM) techniques. Subsequently, these images guide the optimization of 3D Gaussian Splatting (3DGS) by comparing them with the rendered images, thereby refining the 3D Gaussian primitives to improve scene representation.

The quality of novel view synthesis (NVS) in 3DGS is heavily influenced by the density and distribution of point clouds P and the quality of the input multi-view images

I. When robotic vision systems rely exclusively on RGB cameras with limited data, the resulting sparse point clouds and input images can significantly impair the completeness and level of detail in the geometric representation, limiting the capacity of 3DGS to accurately capture scene complexity. This limitation becomes particularly critical in complex or unbounded environments, where inadequate data hampers the ability to represent intricate geometric structures and variations in lighting, thereby reducing the effectiveness of scene reconstruction.

B. Initialize Adaptive Dense

In 3D scene reconstruction, a combined strategy of global and local processing is employed to balance the accuracy of the overall structure with the refinement of local details. Global processing is responsible for capturing the broad geometric structure of the entire scene, while local processing focuses on enhancing the detail representation within specific regions.

1) *Global Processing*: The primary objective of global processing is to ensure the geometric consistency of the entire scene. Using the point clouds $P_{\text{init}} = \{p_i \mid i = 1, \dots, k\}$ generated by SfM, we first address the overall structure to obtain a comprehensive spatial framework and point cloud density distribution. The global processing optimizes P_{init} to derive a global density function $\rho(p)$:

$$\rho_{\text{global}}(p) = \int_P \exp\left(-\frac{\|p - q\|^2}{2\sigma_p^2}\right) f(q) dq, \quad (4)$$

where each point $p_i \in P$ has coordinates (x_i, y_i, z_i) , q represents the potential nearest neighbors of the point p . $f(q)$ is the density function, representing the weight or density at point q . This density function is utilized to assess the distribution of points across the point clouds, ensuring that the essential geometric structures are retained at the global level.

2) *Local Processing*: Following global processing, the point clouds are partitioned into several local regions N , where each region undergoes more detailed optimization. The main goal of local processing is to enhance the representation of fine details. For a local region R_i , the bounding

box is defined as:

$$p_{\min_i} = \min(p_{R_i}), \quad p_{\max_i} = \max(p_{R_i}), \quad (5)$$

where p_{R_i} denotes the points within the region R_i . The position of the newly generated points $p_r \in [p_{\min_i}, p_{\max_i}]$ is determined by uniform sampling within this bounding box. The local point cloud density function $\rho_{\text{local}}(p)$ is further refined to capture intricate geometric details:

$$\rho_{\text{local}}(p_r) = \int_{R_i} \exp\left(-\frac{\|p_r - q_r\|^2}{2\sigma_{p_r}^2}\right) f(q_r) dq_r, \quad (6)$$

where R_i represents the integration domain, which encompasses the entire range of possible values for the local region around p_r ; q_r represents the potential nearest neighbors of the point p_r .

3) *Density-weighted selection*: Upon completing the local and global density estimations, the point selection process strategically integrates these results, optimizing the balance between local precision and global coherence to enhance the overall quality of the reconstruction.

Initially, within each local region, a KD-tree [35] is constructed to identify the k nearest neighbors p_i for each point p . The distances between p and these neighbors are calculated and then converted into local density values $\rho_{\text{local}}(p)$ using a Gaussian function. Based on these density values, the probability of retaining each point $\mathbb{P}_{\text{local}}$ is determined:

$$\mathbb{P}_{\text{local}}(p_{r_j} \in p_r) \propto \rho_{\text{local}}(p_r). \quad (7)$$

Simultaneously, a similar process is conducted at the global level. The global density $\rho_{\text{global}}(p)$ is estimated by calculating the distances to the global nearest neighbors p_i , and the corresponding global retention probability $\mathbb{P}_{\text{global}}$ is computed:

$$\mathbb{P}_{\text{global}}(p_i \in p) \propto \rho_{\text{global}}(p). \quad (8)$$

The selected points from both the local P_{local} and global P_{global} density estimations are combined with the initial point cloud P_{init} using a union operation, resulting in the final point cloud P_{final} :

$$P_{\text{final}} = P_{\text{init}} \oplus P_{\text{local}} \oplus P_{\text{global}}. \quad (9)$$

This approach ensures that both the global structural integrity and local detail accuracy are maintained, thereby improving the overall quality and precision.

C. 3D Smoothing

The intrinsic and extrinsic parameters of the camera are not fixed, leading to varying degrees of artifacts when rendering novel views, especially upon magnification. In the optimization process, the coordinates $o_i = (x_{o_i}, y_{o_i}, z_{o_i})$ of any arbitrary 3D Gaussian need to be transformed from the world coordinate system to each coordinate system of camera:

$$e_i = o_i R_i + T_i = (x_{e_i}, y_{e_i}, z_{e_i}), \quad (10)$$

where R_i and T_i represent the rotation matrix and translation matrix for the i -th camera. The transformed point is then

projected onto the image plane using the intrinsic matrix of the camera:

$$x_i^s = \frac{x_{e_i}}{z_{e_i}} \cdot f_{i,x} + \frac{W_i}{2}, \quad y_i^s = \frac{y_{e_i}}{z_{e_i}} \cdot f_{i,y} + \frac{H_i}{2}, \quad (11)$$

where f_i represents the focal length of the i -th camera; H_i and W_i represent the height and width of the image, respectively. The maximum Gaussian point frequency β_k is obtained using the observed positions of the 3D Gaussians on the screen:

$$\zeta_k = \sup\left(\frac{f_i}{z_{e_i}}\right), \quad (12)$$

where $x_i^s \in [-\alpha W_i, (1+\alpha)W_i]$ and $y_i^s \in [-\alpha H_i, (1+\alpha)H_i]$. The hyperparameter α is used to extend the boundary of the image plane, ensuring that points near the image edges are considered.

After 3D smoothing filtering, the 3D Gaussian is represented as follows:

$$\mathcal{G}_k(x) = \sqrt{\frac{\Sigma_k}{\Sigma_{k_s}}} \cdot e^{-\frac{1}{2}(\mathbf{x}-\mu_k)^T \Sigma_{k_s}^{-1} (\mathbf{x}-\mu_k)}, \quad (13)$$

where $\Sigma_{k_s} = \Sigma_k + \frac{s}{\zeta_k^2} \cdot \mathbf{I}$ represents the covariance matrix after filtering.

D. Depth SDS as Optimization Guidance

Using the diffusion model to generate spatially aligned RGB images and depth maps, we can guide the 3DGS optimization process in both structure and texture. The depth map for each view is computed by accumulating the depth values of \mathcal{N} ordered Gaussian primitives along the ray, using point-based α blending:

$$D_r(x) = \sum_{i \in \mathcal{N}} d_{\mu_i} \sigma_i \prod_{j=1}^{i-1} (1 - \sigma_j), \quad (14)$$

where d_{μ_i} is the depth of the i -th Gaussian primitive center μ_i in the camera view. All depth maps from the training views are normalized for subsequent depth-based loss calculation.

We employ SDS [6], [15] to guide the optimization of 3DGS through 2D diffusion [36]. The rendered image \tilde{I} and depth map \tilde{D} from unseen viewpoints v are jointly used to optimize 3DGS through SDS:

$$\begin{aligned} \nabla_{\theta} \mathcal{L}_{\text{SDS}} &= \lambda_1 \cdot \mathbb{E}_{\epsilon_I, t} \left[w_t \left(\epsilon_{\phi}(I_t; \tilde{I}^v, t) - \epsilon_I \right) \frac{\partial I_t}{\partial \theta} \right] \\ &\quad + \lambda_2 \cdot \mathbb{E}_{\epsilon_D, t} \left[w_t \left(\epsilon_{\phi}(D_t; \tilde{D}^v, t) - \epsilon_D \right) \frac{\partial D_t}{\partial \theta} \right], \end{aligned} \quad (15)$$

where λ_1 and λ_2 are coefficients that balance the influence of image and depth; $\epsilon_{\phi}(\cdot)$ is the denoising function of 2D diffusion; $\epsilon_I, \epsilon_D \sim N(0, I)$ are independent Gaussian noises. By integrating the 2D diffusion model, 3DGS can be optimized more effectively, enabling the generated images and depth maps from new viewpoints to more accurately reflect the geometric structure and textural details of the actual scene.

E. Depth mask and Gradient Profile Prior

Since noise and irrelevant details in the distant background can negatively impact the gradient calculation process, leading to blurred edges and loss of detail in the reconstruction, we introduce a dynamic depth mask to effectively suppress high-frequency noise and artifacts from distant objects, thereby improving the geometric accuracy and visual quality of the reconstruction. To accommodate scenes with varying depth distributions, q_f for the far-distance threshold is calculated as follows:

$$q_f = q_b + \left(\frac{\beta_D}{\beta_D + \alpha_D} \right) \times \Delta q, \quad (16)$$

where α_D and β_D represent the mean and standard deviation of the depth map D , respectively. p_b is the base quantile, and Δp is the dynamic adjustment range. The generated mask M is defined as:

$$M = \mathbb{1}_{D \leq T_f} = \mathbb{1}_{D \leq \text{Quantile}(D, q_f)}, \quad (17)$$

where $\mathbb{1}_{(\cdot)}$ is an indicator function that assesses the visibility of depth map D . The mask is determined by calculating the value T_f at the quantile p_f of the depth map D . The final masked depth map ($D_m = D \odot M$) is used for gradient operations.

The Depth Gradient Profile Prior (DGPP) is introduced to enhance the sharpness and accuracy of edges in the depth map, particularly focusing on refining the texture and geometric details. The GPP loss is formulated to enforce the alignment of gradient profiles between the rendered depth map \hat{D}_m and the target depth map D_m . When the pixel positions b of \hat{D}_m and D_m correspond one-to-one, the DGPP loss function is defined as:

$$\mathcal{L}_{\text{DGPP}} = \frac{1}{b_1 - b_0} \int_{b_0}^{b_1} \|\nabla \hat{D}_m(b) - \nabla D_m(b)\|_1 db, \quad (18)$$

where $\nabla \hat{D}_m$ and ∇D_m represent the gradient fields of the rendered and target depth maps, respectively. The depth alignment ensures that the sharpness of edges is preserved and that the 3D reconstruction accurately reflects the underlying geometry.

F. Loss Function

To optimize the 3D Gaussian representation ($\{\theta_k = (\mu_k, \Sigma_k, \alpha_k, c_k)\}_k^K$), we designed a final optimization function that integrates various loss terms. Our final loss function for optimizing 3D Gaussians is defined as:

$$\begin{aligned} \mathcal{L}_{\text{final}} = & \underbrace{\mathcal{L}_{\text{RGB}}(\hat{I}(\theta), I) + \lambda_{\text{depth}} \mathcal{L}_{\text{depth}}(\hat{D}(\theta), D, M)}_{\text{loss of known view}} \\ & + \underbrace{\lambda_{\text{SDS}} \mathcal{L}_{\text{SDS}}(\tilde{I}^v(\theta), \tilde{D}^v(\theta))}_{\text{loss of novel view}}, \end{aligned} \quad (19)$$

where \hat{I} , \tilde{I}^v represent the RGB images rendered by the 3D Gaussian primitives; I represents the reference RGB image; \hat{D} , \tilde{D}^v represent the depth maps rendered by the 3D Gaussian primitives; D represents the reference depth map.

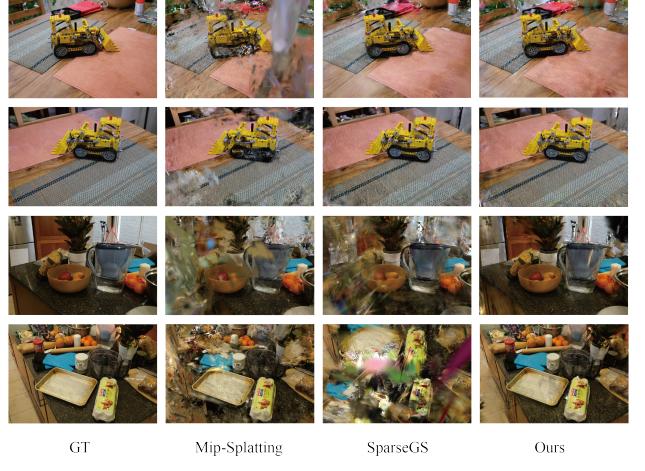


Fig. 3. Qualitative results on the Mip-NeRF 360 dataset show that our approach is perceptually similar to the ground truth.

IV. EXPERIMENTS

A. Datasets and Evaluation Metrics

For unbounded scenes, we select six 360° coverage scenes from Mip-NeRF 360 [24] to evaluate our model. For underwater scenes, we used the SeaThru-NeRF dataset [37] to evaluate the applicability of our framework to other complex scenes. We employ tree metrics (PSNR, SSIM [38], and LPIPS [39]), to evaluate and compare our method against existing approaches.

B. Implementation Details

Our method is implemented using the PyTorch [40] framework and the open-source 3DGS [4] codebase. AdamW [41] is employed as the optimizer. For all scenes, the models were trained for 30K iterations using the same loss function, Gaussian density control strategy, and hyperparameters to optimize the 3D Gaussian primitives. Both Gaussian training and rendering tests were performed on a NVIDIA™ RTX 4090 GPU.

C. Qualitative and Quantitative Evaluation

In the qualitative analysis on MipNeRF360, as shown in the Fig.3, we compared the performance of different methods in reconstructing complex scenes. When reconstructing unbounded scenes, SVS-GS, with its integration of 3D Gaussian smoothing and depth priors, clearly outperforms traditional 3DGS and SparseGS methods by successfully capturing more intricate structures and lighting variations. These results further confirm the advantages and practical effectiveness of SVS-GS in sparse view scene reconstruction. Similarly, on the SeaThru-NeRF underwater dataset, SVS-GS again outperformed other methods, as shown in the Fig.4. Particularly in handling the challenges of complex underwater lighting conditions and sparse viewpoints, SVS-GS demonstrated greater robustness and accuracy, successfully reducing visual distortions and preserving more scene details. These quantitative results underscore the broad applicability

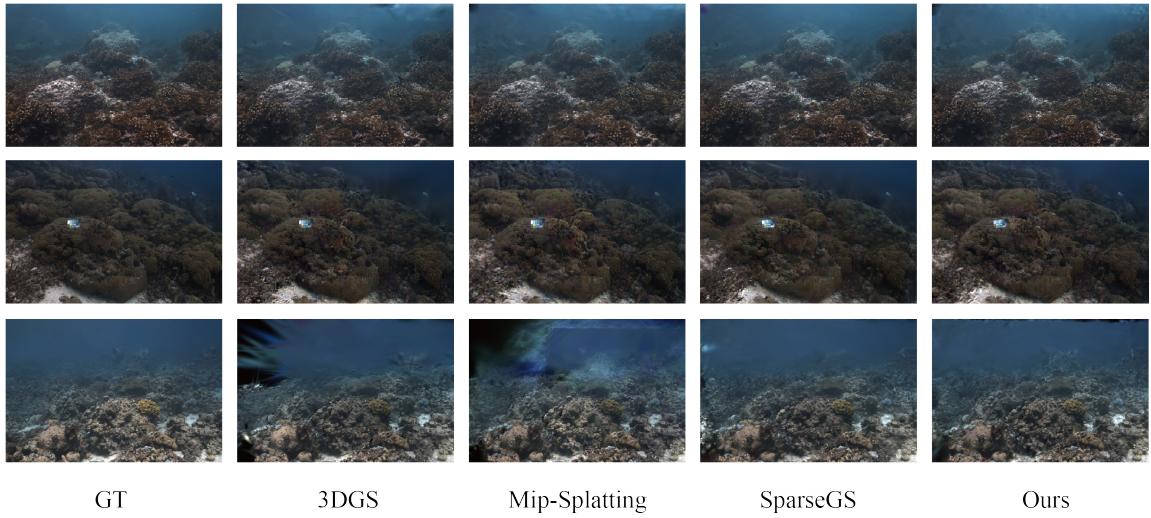


Fig. 4. Qualitative results on the SeaThru-NeRF dataset show that our method can effectively shield the influence of distant scenery.

TABLE I

QUANTITATIVE EVALUATION. COMPARISON OF METHODS ON MIP-NERF360 AND SEATHRU-NERF DATASETS. ON THE DATASETS WITH 8 INPUT VIEWS, SVS-GS OUTPERFORMS OTHER METHODS.

Dataset	Method	Mip-NeRF360			SeaThru-NeRF		
		PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Metrics	Mip-NeRF360 [24]	11.28	0.193	0.612	22.89	0.830	0.245
	3DGS [4]	10.45	0.163	0.640	22.24	0.785	0.242
	SparseGS [8]	12.33	0.225	0.593	22.99	0.769	0.234
	Mip-Splatting [10]	11.43	0.181	0.632	22.42	0.799	0.225
	Ours	12.80	0.238	0.573	23.06	0.791	0.214

of SVS-GS across different scenarios and viewpoint conditions.

In the quantitative analysis, we systematically evaluated the performance of SVS-GS against other methods on the MipNeRF360 and SeaThru-NeRF datasets, as shown in the Table.I. Comparison of the PSNR, SSIM, and LPIPS metrics clearly demonstrates the significant advantages of SVS-GS in terms of reconstruction accuracy and image quality. On the MipNeRF360 dataset, SVS-GS achieved the highest scores in both PSNR and SSIM, indicating its superior ability to reconstruct geometric and textural details in sparse views, while also exhibiting the lowest perceptual error in the LPIPS, further validating its visual fidelity.

D. Ablations and Analysis

As shown in Table.II, we conducted ablation studies to evaluate the impact of key components in our method. The dynamic depth mask plays a crucial role in effectively reducing noise and artifacts in distant areas, confirming its importance in filtering out irrelevant depth information. DGPP sharpens edge contours, highlighting its importance in preserving details. Additionally, omitting the 3D Gaussian smoothing filter results in a noticeable increase in surface noise and artifacts, demonstrating its essential role in maintaining the smoothness and consistency of the reconstructed surfaces. The lack of SDS leads to geometric inconsistencies in the synthesized novel views, emphasizing the necessity of this component in ensuring geometric coherence and min-

TABLE II

ABLATION STUDIES ON UNDERWATER SCENES. COMPARISONS ON THE SEATHRU-NERF DATASET WITH 8 INPUT VIEWS INDICATE THAT THE MODEL WITH ALL MODULES PERFORMS BEST, WITH EACH MODULE CONTRIBUTING TO THE OVERALL PERFORMANCE.

	PSNR↑	SSIM↑	LPIPS↓
w/o dense	22.55	0.8155	0.2651
w/o depth	20.64	0.8115	0.2561
w/o 3D smoothing	22.04	0.8130	0.2731
w/o SDS	22.61	0.8189	0.2626
w/o DGPP	22.72	0.8162	0.2663
All	22.78	0.8234	0.2488

imizing visual discrepancies. Each component contributes to the effectiveness of achieving high-quality 3D scene reconstruction.

V. CONCLUSION

In this paper, we introduce SVS-GS, a novel framework for 3D scene reconstruction from sparse viewpoints, optimized for both robotic vision systems and broader computer vision tasks using only RGB cameras. Our method utilizes a dynamic depth mask to enhance geometric accuracy by selectively retaining critical depth information. Additionally, by incorporating depth priors, a 3D Gaussian smoothing filter, and Depth Gradient Profile Prior (DGPP) loss, our approach sharpens edges and preserves fine details in complex scenes. To ensure high-quality and consistent novel view synthesis, we integrate Score Distillation Sampling (SDS) loss, which reduces noise and maintains geometric coherence across different viewpoints. Experimental results demonstrate that SVS-GS outperforms existing methods in sparse viewpoint scenarios, achieving superior visual fidelity and geometric consistency. Furthermore, our framework shows robust performance across various challenging environments, making it an efficient and effective solution for 3D scene reconstruction in both robotics and computer vision applications.

REFERENCES

- [1] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng, “Nerf: Representing scenes as neural radiance fields for view synthesis,” *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.
- [2] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan, “Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 5855–5864.
- [3] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman, “Zip-nerf: Anti-aliased grid-based neural radiance fields,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023, pp. 19697–19705.
- [4] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis, “3d gaussian splatting for real-time radiance field rendering,” *ACM Trans. Graph.*, vol. 42, no. 4, pp. 139–1, 2023.
- [5] Johannes L Schöberger and Jan-Michael Frahm, “Structure-from-motion revisited,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4104–4113.
- [6] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng, “Dreamgaussian: Generative gaussian splatting for efficient 3d content creation,” *arXiv preprint arXiv:2309.16653*, 2023.
- [7] Taoran Yi, Jiemin Fang, Junjie Wang, Guanjun Wu, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Qi Tian, and Xinggang Wang, “Gausiandreamer: Fast generation from text to 3d gaussians by bridging 2d and 3d diffusion models,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 6796–6807.
- [8] Haolin Xiong, Sairisheek Muttukuru, Rishi Upadhyay, Pradyumna Chari, and Achuta Kadambi, “Sparsegs: Real-time 360 {\deg} sparse view synthesis using gaussian splatting,” *arXiv preprint arXiv:2312.00206*, 2023.
- [9] Shijie Zhou, Haoran Chang, Sicheng Jiang, Zhiwen Fan, Zehao Zhu, Dejia Xu, Pradyumna Chari, Suya You, Zhangyang Wang, and Achuta Kadambi, “Feature 3dgs: Supercharging 3d gaussian splatting to enable distilled feature fields,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 21676–21685.
- [10] Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger, “Mip-splatting: Alias-free 3d gaussian splatting,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 19447–19456.
- [11] Zheng Zhang, Wenbo Hu, Yixing Lao, Tong He, and Hengshuang Zhao, “Pixel-gs: Density control with pixel-aware gradient for 3d gaussian splatting,” *arXiv preprint arXiv:2403.15530*, 2024.
- [12] Zhiwen Fan, Wenyan Cong, Kairun Wen, Kevin Wang, Jian Zhang, Xinghao Ding, Danfei Xu, Boris Ivanovic, Marco Pavone, Georgios Pavlakos, et al., “Instantsplat: Unbounded sparse-view pose-free gaussian splatting in 40 seconds,” *arXiv preprint arXiv:2403.20309*, 2024.
- [13] Jiahe Li, Jiawei Zhang, Xiao Bai, Jin Zheng, Xin Ning, Jun Zhou, and Lin Gu, “Dngaussian: Optimizing sparse-view 3d gaussian radiance fields with global-local depth normalization,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20775–20785.
- [14] Johannes Lutz Schöberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm, “Pixelwise view selection for unstructured multi-view stereo,” in *European Conference on Computer Vision (ECCV)*, 2016.
- [15] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall, “Dreamfusion: Text-to-3d using 2d diffusion,” *arXiv*, 2022.
- [16] Jian Sun, Zongben Xu, and Heung-Yeung Shum, “Gradient profile prior and its applications in image super-resolution and enhancement,” *IEEE Transactions on Image Processing*, vol. 20, no. 6, pp. 1529–1542, 2010.
- [17] S Mahdi H Miangoleh, Sebastian Dille, Long Mai, Sylvain Paris, and Yagiz Aksoy, “Boosting monocular depth estimation models to high-resolution via content-adaptive multi-resolution merging,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 9685–9694.
- [18] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth, “Nerf in the wild: Neural radiance fields for unconstrained photo collections,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 7210–7219.
- [19] Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P Srinivasan, Jonathan T Barron, and Henrik Kretzschmar, “Block-nerf: Scalable large scene neural view synthesis,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8248–8258.
- [20] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan, “Depth-supervised nerf: Fewer views and faster training for free,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12882–12891.
- [21] Zhiqin Chen and Hao Zhang, “Learning implicit fields for generative shape modeling,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 5939–5948.
- [22] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger, “Occupancy networks: Learning 3d reconstruction in function space,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4460–4470.
- [23] James T Kajiya and Brian P Von Herzen, “Ray tracing volume densities,” *ACM SIGGRAPH computer graphics*, vol. 18, no. 3, pp. 165–174, 1984.
- [24] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman, “Mip-nerf 360: Unbounded anti-aliased neural radiance fields,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 5470–5479.
- [25] Guangcong Wang, Zhaoxi Chen, Chen Change Loy, and Ziwei Liu, “Sparsenerf: Distilling depth ranking for few-shot novel view synthesis,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 9065–9076.
- [26] Jiawei Yang, Marco Pavone, and Yue Wang, “Freenerf: Improving few-shot neural rendering with free frequency regularization,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 8254–8263.
- [27] Prune Truong, Marie-Julie Rakotosaona, Fabian Manhardt, and Federico Tombari, “Sparf: Neural radiance fields from sparse and noisy poses,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 4190–4200.
- [28] Jeffrey P Grossman and William J Dally, “Point sample rendering,” in *Rendering Techniques’ 98: Proceedings of the Eurographics Workshop in Vienna, Austria, June 29—July 1, 1998*. Springer, 1998, pp. 181–192.
- [29] Hanspeter Pfister, Matthias Zwicker, Jeroen Van Baar, and Markus Gross, “Surfels: Surface elements as rendering primitives,” in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, 2000, pp. 335–342.
- [30] Yufeng Zheng, Wang Yifan, Gordon Wetzstein, Michael J Black, and Otnar Hilliges, “Pointavatar: Deformable point-based head avatars from videos,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 21057–21067.
- [31] Songyou Peng, Chiyu Jiang, Yiyi Liao, Michael Niemeyer, Marc Pollefeys, and Andreas Geiger, “Shape as points: A differentiable poisson solver,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 13032–13044, 2021.
- [32] Wang Yifan, Felice Serena, Shihao Wu, Cengiz Öztieli, and Olga Sorkine-Hornung, “Differentiable surface splatting for point-based geometry processing,” *ACM Transactions on Graphics (TOG)*, vol. 38, no. 6, pp. 1–14, 2019.
- [33] Tao Lu, Mulin Yu, Liming Xu, Yuanbo Xiangli, Limin Wang, Dahua Lin, and Bo Dai, “Scaffold-gs: Structured 3d gaussians for view-adaptive rendering,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20654–20664.
- [34] Shen Chen, Jiale Zhou, Zhongyu Jiang, Tianfang Zhang, Zongkai Wu, Jenq-Neng Hwang, and Lei Li, “Scalinggaussian: Enhancing 3d content creation with generative gaussian splatting,” *arXiv preprint arXiv:2407.19035*, 2024.
- [35] Jon Louis Bentley, “Multidimensional binary search trees used for associative searching,” *Communications of the ACM*, vol. 18, no. 9, pp. 509–517, 1975.
- [36] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer, “High-resolution image synthesis with latent diffusion models,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10684–10695.
- [37] Deborah Levy, Amit Peleg, Naama Pearl, Dan Rosenbaum, Derya Akkaynak, Simon Korman, and Tali Treibitz, “Seathru-nerf: Neural radiance fields in scattering media,” in *Proceedings of the IEEE/CVF*

Conference on Computer Vision and Pattern Recognition, 2023, pp. 56–65.

- [38] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [39] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang, “The unreasonable effectiveness of deep features as a perceptual metric,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 586–595.
- [40] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al., “Pytorch: An imperative style, high-performance deep learning library,” *Advances in neural information processing systems*, vol. 32, 2019.
- [41] I Loshchilov, “Decoupled weight decay regularization,” *arXiv preprint arXiv:1711.05101*, 2017.