SeG-Gaussian:Segmentation-Guided 3D Gaussian Optimization for Novel View Synthesis

Ling-Xiao Zhang, Chenbo Jiang, Yu-Kun Lai *Senior Member, IEEE* and Lin Gao *Member, IEEE*

***Abstract*—Radiance field based methods have recently revolutionized novel view synthesis of scenes captured with multi-view photos. A significant recent advance is 3D Gaussian Splatting (3DGS), which utilizes a set of 3D Gaussians to represent a radiance field, yielding high-fidelity results in real-time rendering. However, we have observed that 3DGS struggles to capture the necessary details in sparsely observed regions, where there is not enough gradient for effective split and clone operations. In this paper, we present a novel solution to address this limitation. Our key idea is to leverage segmentation information to identify poorly optimized regions within the 3D Gaussian representation. By applying split or clone operations on the corresponding 3D Gaussians in these regions, we aim to refine the spatial distribution of Gaussians and enhance the overall quality of high-fidelity 3D scene reconstruction. To further optimize the reconstruction process, we introduce two spatial regularization terms: repulsion loss and smoothness loss. These terms effectively minimize overlap and redundancy among Gaussians, reducing outliers in the synthesized geometry. By incorporating these regularization techniques, our approach achieves state-of-the-art performance in real-time novel view synthesis and significantly improves visibility in less observed regions, leading to a more compact and accurate 3D scene representation.**

***Index Terms*—Gaussian Splatting, Radiance Fields, Semantic Guidance, Regularization.**

I. Introduction

HIGH-fidelity novel view synthesis plays a crucial role in various vision and graphics applications such as virtual reality, robotics, video games, and film. Radiance field based methods have emerged as powerful techniques for achieving remarkable results in these fields. Among recent advancements, 3D Gaussian Splatting (3DGS) [1] has emerged as a powerful approach, leveraging anisotropic 3D Gaussians to represent complex scenes efficiently. Compared to traditional mesh-based methods and volumetric representations like Neural Radiance Field (NeRF) [3], 3DGS provides both high-quality reconstruction and real-time rendering capabilities, making it an attractive choice for low-cost 3D content creation. 3D Gaussian Splatting (3DGS) [1] adopts explicit anisotropic 3D Gaussians as primitives and assigns view-dependent radiance with spherical harmonics and opacity to each Gaussian. 3DGS also incorporates an adaptive control strategy, involving split and clone operations, to efficiently optimize regions of interest. This strategy is vital for enhancing the representation power of 3DGS since it relies on placing the appropriate Gaussians in the correct positions. However, 3DGS may struggle to represent details in inconspicuous regions. Fig. 1 illustrates this limitation in the representation power of 3DGS.

Recent works have sought to address these limitations. AbsGS [4] and GOF [5] refine the densification process by considering absolute gradient values instead of directional gradients, allowing more robust capture of fine details in regions with subtle geometric variations. Concurrently, 3DGS-MCMC [6] reformulates Gaussian placement as a probabilistic sampling problem, interpreting Gaussians as samples drawn via Stochastic Gradient Langevin Dynamics (SGLD). By introducing noise into Gaussian updates and redefining the clone operation as a relocalization scheme, their approach reduces dependence on initialization and improves Gaussian efficiency. While 3DGS-MCMC provides a powerful global optimization framework, it does not explicitly consider the semantic structure of the scene when redistributing Gaussians. The key limitation of AbsGS and GOF lies in their reliance on view-space positional gradients, and they still densify Gaussians when the average gradient magnitude surpasses a threshold, even though they take absolute values before averaging.

Our method aims to address the spatial distribution issue in 3DGS by placing Gaussians in appropriate positions. In other words, our approach focuses on optimizing the spatial distribution of Gaussians. Firstly, we observe that 3D Gaussians that are not well optimized often share some common semantic region. This limitation arises from the reliance of the original 3DGS approach on accumulated color gradients for the split and clone operations. As a result, regions that are only sparsely observed by a few views may not accumulate sufficient gradients to trigger the split and clone operations. Consequently, these regions are not adequately optimized and can exhibit suboptimal representations in the 3D Gaussian splatting process. For example, in the case of the bicycle scene shown in Fig. 1, the shrub grass region is not well optimized, while the bench is better optimized. Based on this observation, we believe that identifying which regions have poorly optimized Gaussians and applying splitting and cloning operations specifically to those regions can lead to better results. This idea is inspired by the work of Häne et al. [7], [8],

This work was supported by the National Natural Science Foundation of China(No. 62322210), Beijing Municipal Science and Technology Commission (No. Z231100005923031), and Innovation Funding of ICT, CAS (No. E461020). (Corresponding author: Lin Gao.)

Ling-Xiao Zhang and Lin Gao are with the Beijing Key Laboratory of Mobile Computing and Pervasive Device, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China. Ling-Xiao Zhang and Lin Gao are also with the University of Chinese Academy of Sciences, Beijing 101408, China (e-mail: zhanglingxiao@ict.ac.cn;gaolin@ict.ac.cn).

Chenbo Jiang is with McGill University, 845 Rue Sherbrooke O, Montréal, QC H3A 0G4, Canada (e-mail: cbjiang@foxmail.com).

Yu-Kun Lai is with the School of Computer Science and Informatics, Cardiff University, CF24 4AG, Cardiff, U.K. (e-mail: laiy4@cardiff.ac.uk).

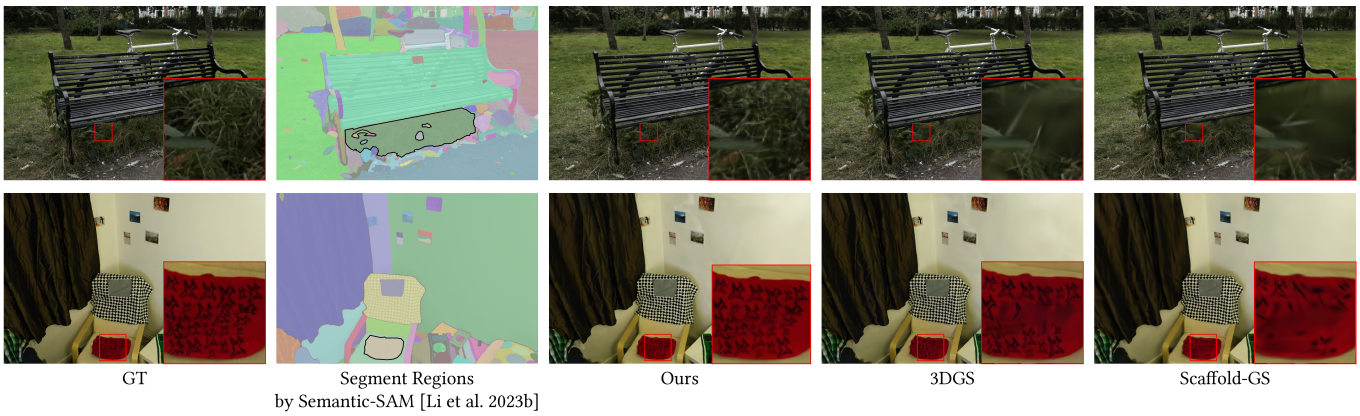


Fig. 1. 3D Gaussian Splatting (3DGS) [1] represents radiance fields through a collection of 3D primitives known as 3D Gaussians. These primitives possess radiance and opacity attributes. Despite its effectiveness, optimizing the spatial distribution of these primitives poses a challenge. In situations such as the BICYCLE scene, where the shrub grass region is not well optimized, 3DGS and other methods based on it, such as Scaffold-GS [2], struggle to accumulate sufficient gradients for split and clone operations. This limitation prohibits the achievement of detailed region reconstruction. Our proposed method utilizes segmentation information to guide spatial optimization, enabling the reconstruction of intricate regions with infrequent occurrence.

who suggest that semantic segmentation and reconstruction tasks in images can mutually benefit each other, as they share many connections. As the change of visibility is usually due to occlusion, 3D Gaussians not well optimized tend to be clustered into regions, with boundaries often aligned with object boundaries, so segmentation can be useful to identify such regions. On the other hand, it is also desirable to have optimized 3DGS with more regularly placed Gaussians and fewer outliers. We observe that the original 3DGS uses plenty of Gaussians to represent a scene, and this causes redundancy and overlap. So we introduce regularization terms to constrain the spatial distribution, leading to a more regular and efficient representation.

Specifically, we first utilize Semantic-SAM [9] to extract several segmentation masks for each image. Semantic-SAM is capable of generating more detailed and multi-level masks and achieves higher mask quality compared to SAM [10]. We propose a different strategy for split and clone operations compared to the original 3DGS. In 3DGS, the decision to perform splitting and cloning on a particular Gaussian is determined by the accumulated color gradient of each Gaussian. However, it is challenging to obtain effective gradient accumulation for regions that appear less frequently in the input images. In contrast, our strategy involves testing whether each mask region has been adequately optimized. If not, we perform a re-projection of the pixels in that region onto the Gaussians to determine which Gaussians correspond to the mask. Subsequently, we apply split or clone operations on those Gaussians to improve the optimization for the corresponding mask region. Please note that 3D segmentation and multi-view consistency are not necessary for our objectives. Our main focus is novel view synthesis results, which naturally produce 2D images. Hence, it is sufficient to use these 2D images directly to identify the regions that need improvement. Upon identifying these regions, we can then re-project them onto 3D Gaussians for our optimization. Such re-projections from different training images are aggregated, and a 3D Gaussian corresponds to at least one such region is considered as needing improvements. During the optimization process, we incorporate two regularization terms for spatial regularization: repulsion loss and smoothness loss. The repulsion loss aims to minimize overlap and redundancy among the Gaussians. By penalizing Gaussians that are too close or overlapping, we encourage a more even distribution of Gaussians in space. On the other hand, the smoothness loss is employed to reduce outliers. It forces outlier Gaussians to move to the local surface that is fitted by neighboring Gaussians. By incorporating these regularization terms, we can improve the overall spatial distribution and quality of the synthesized geometry.

Experimental results demonstrate that our method achieves state-of-the-art in several metrics for real-time rendering in novel view synthesis and better visual quality than others. Our method exhibits noticeable benefits in regions that are less frequently observed, such as the bicycle scene, as shown in Fig. 1. We also evaluate reconstruction results in different levels of visibility, as shown in Sec. IV-D. Besides, our method can be integrated as a plug-and-play module and combined with other 3D-Gaussian-based methods, which have the same densification strategy as the original 3DGS. We show the results of combining Scaffold-GS [2] with our method.

Overall, our contributions can be summarized as follows:

1) We introduce SeG-Gaussian, a novel approach that leverages segmentation guidance and adaptive density control to improve the distribution of 3D Gaussians, which can be integrated as a plug-and-play module. This leads to better results in synthesizing novel views and enhances the overall quality of the rendered images.

2) We propose the integration of spatial regularization terms, including repulsion loss and smoothness loss, to refine the spatial arrangement of the Gaussians. This further improves the rendering quality by reducing overlap, redundancy, and outliers in the synthesized geometry.

3) Experiments demonstrate that our method achieves state-of-the-art performance for real-time rendering in novel view synthesis. Additionally, our approach produces visually superior results with enhanced details in regions that are less

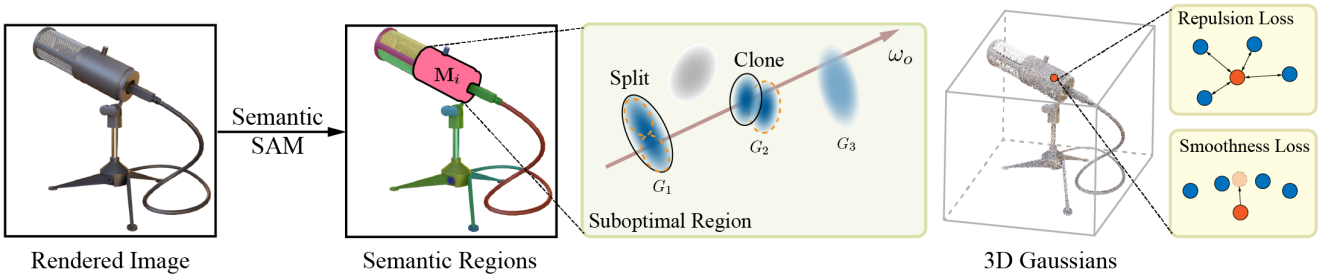


Fig. 2. Overview of SeG-Gaussian. For each training view, we extract K masks through Semantic-SAM. For each region $\mathbf{M}_i$, we calculate the mean L_1 error $\mathcal{L}_{M_i} = \frac{1}{\|\mathbf{M}_i\|_0}\|(\mathbf{I}-\hat{\mathbf{I}})\mathbf{M}_i\|_1$ between the region $\mathbf{M}_i$ of ground truth image $\mathbf{I}$ and rendered image $\hat{\mathbf{I}}$. If $\mathcal{L}_{M_i}$ is greater than the mean L_1 error for the whole image $\mathcal{L}_1 = \frac{1}{\|\mathbf{I}_i\|_0}\|(\mathbf{I}-\hat{\mathbf{I}})\|_1$, Gaussians that lie in that suboptimal region should be densified. To prevent excessive densification of Gaussians, we involve Gaussians G_j with weights $\alpha_j T_j > 0.5$ that contribute significantly to the final color in the densification process. We perform densification for every pass through the training dataset. To further improve the 3D Gaussian distribution, we introduce two spatial regularization terms on 3D Gaussians, repulsion loss L_r, which separates Gaussians that are too close, enabling a more even and regular Gaussian representation of the scene, and smoothness loss L_s that forces outlier Gaussians to move onto the local surface fitted by neighbors.

frequently observed.

II. Related Work

A. Novel View Synthesis

Novel View Synthesis (NVS) involves generating images from new viewpoints based on a set of posed images. NeRF [3], a representative method in this field, utilizes volume rendering [11], [12] to accomplish this task. NeRF represents the scene as implicit MLPs [13]–[15], which take the position and direction as input and output color and density. With the success of NeRF, several follow-up methods have aimed to enhance the quality and speed of neural radiance fields. Mip-NeRF [16], Mip-NeRF360 [17], and Zip-NeRF [18] achieve impressive visual results in neural radiance fields, but they require considerable training time and cannot render in real-time. To speed up neural radiance fields, some works have proposed different representations [19]–[28] to accelerate radiance fields. However, these methods struggle to find a balance between rendering quality and speed.

Recently, there have been some advancements in radiance fields that gradually shift away from using neural networks and instead employ explicit methods to represent scenes, resulting in state-of-the-art results in real-time rendering. MobileNeRF [29] uses an explicit mesh to represent geometry and assigns neural features to the geometry. This allows for the use of a traditional rendering pipeline to render feature images, which are then used to obtain RGB values using a lightweight network. Point-NeRF [30] uses an explicit point cloud to represent geometry and assigns features to the points. During volume rendering, point features are converted to radiance to render images. BakedSDF [31] optimizes a hybrid neural volume-surface representation and bakes them onto triangle meshes. These methods do not completely abandon neural networks but instead utilize explicit geometry as a prior. Plenoxels [32] represents a scene as a voxel grid and use spherical harmonics (SH) features to represent view-dependent radiance. To render the results, Plenoxels needs to interpolate continuous radiance fields and conduct volume rendering. 3D Gaussian Splatting (3DGS) [1] departs from the rendering approach of NeRF and instead utilizes splatting [33], [34] for rendering. 3DGS represents the scene as a collection of 3D Gaussians, with opacity and radiance defined on each Gaussian. Additionally, it uses spherical harmonics (SH) to represent radiance from different viewpoints. Through this method, it achieves high-quality and real-time rendering simultaneously. Following that, many orthogonal methods aimed at improving rendering quality and compressing 3D Gaussian representations have emerged [2], [35]–[42]. Scaffold-GS [2] uses anchor points to distribute local 3D Gaussians, and predicts their attributes on-the-fly based on the viewing direction. FreGS [35] designs a progressive frequency regularization to tackle the over-reconstruction issue within the frequency space. Mip-Splatting [43] introduces a 3D smoothing filter and a 2D Mip filter, eliminating multiple artifacts and achieving alias-free renderings. These methods improve the quality of novel view synthesis and are orthogonal to our method, which can be combined together to produce better results.

B. Primitive-based Differentiable Rendering

Finding a suitable primitive to represent the scene is indeed a highly effective approach, especially in the field of differentiable rendering. A good primitive can serve as an efficient inductive bias to the scene. NeurMiPs [44] uses a mixture of planes to represent scenes. DBW [45] represents a scene with deformable primitives, including a background icosphere, a ground plane, and object primitives. Recently, there has been a significant focus on point-based differentiable rendering [30], [46]–[50]. Among them, the splatting-based method should be highlighted, which requires less training time and less storage, and achieves real-time rendering. Differentiable Surface Splatting [48] uses surface splatting to represent geometry. 3D Gaussian Splatting (3DGS) [1] uses anisotropic Gaussian splatting to represent radiance fields and achieves impressive results while rendering in real-time. For 3DGS, adaptive control, including splitting and cloning for Gaussians, is a key factor in its representation power. However, this strategy may limit the optimization of 3DGS in capturing fine details, as there might not be sufficient gradients available to guide the splitting or cloning operations. We propose a

segmentation-based strategy to identify areas that are not well optimized and subsequently optimize them.

C. Segmentation and Reconstruction

The statement *"Image segmentation and dense 3D reconstruction contribute valuable information to each other's task"* [7], [8] points to the interconnectedness of segmentation understanding and 3D reconstruction. This insight, as highlighted in the works of Häne et al. [7], [8], showcases the potential for leveraging segmentation information to enhance the adaptive control strategy of 3D Gaussian Splatting (3DGS) [1].

In this regard, the recent method SAM [10] has achieved notable results in interactive segmentation. SAM can be employed to obtain a segmentation mask that can identify areas where the optimization of 3DGS may be lacking. There have been subsequent improvements to SAM as well. Semantic-SAM [9] introduces hierarchical segmentation with a finer grid, providing more detailed information. SAM-HQ [51] focuses on generating high-quality masks compared to the original SAM approach.

Recent methods tried to integrate 2D scene understanding methods with NeRF and 3D Gaussians to produce a semantic-embedded 3D scene representation. For example, Semantic NeRF [52] jointly encoded semantics with appearance and geometry within a NeRF for novel semantic view synthesis. LERF [53] was the first to embed CLIP [54] features into NeRF, enabling open-vocabulary 3D queries leveraging the powerful CLIP representation. LangSplat [55] and LEGaussians [56] are both methods that integrate semantic information into 3D Gaussian representations to enhance scene understanding and manipulation. However, there is currently no existing method that directly enhances reconstruction quality by incorporating semantic information. By incorporating segmentation information and techniques like SAM, Semantic-SAM, and SAM-HQ into the adaptive control strategy of 3DGS, it is possible to enhance the representation power and overall performance of 3DGS in capturing fine details and optimizing its rendering quality.

III. Method

We propose SeG-Gaussian, a segmentation-guided 3D Gaussian distribution optimization method for novel view synthesis. The pipeline is illustrated in Fig. 2. Firstly, we give the preliminary of 3D Gaussian Splatting [1] (Sec. III-A). Secondly, we introduce the segmentation-guided 3D Gaussian Distribution Optimization (Sec. III-B). We extract segmentation masks from input images to find the regions for densification. Thirdly, we introduce two spatial regularization terms to constrain the spatial distribution of 3D Gaussians (Sec. III-C).

A. Preliminary

3D Gaussian Splatting is a point-based scene representation for real-time radiance field rendering [1]. The geometry is modeled as a set of 3D Gaussians P that are defined by a full 3D covariance matrix $\mathbf{\Sigma}$ at point $\boldsymbol{\mu}$:

$$G(\mathbf{x}|\boldsymbol{\mu}, \mathbf{\Sigma}) = e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\mathbf{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})}. \tag{1}$$

Because covariance matrices need to be positive semi-definite by definition, it is modeled as $\mathbf{\Sigma} = \mathbf{RSS}^{\mathrm{T}}\mathbf{R}^{\mathrm{T}}$ with scaling matrix $\mathbf{S}$ and rotation matrix $\mathbf{R}$. To render an image for a given view with extrinsic matrix $\mathbf{W}$ and intrinsic matrix $\mathbf{K}$, the projected 2D Gaussians' mean is $\boldsymbol{\mu}' = \mathbf{KW}[\boldsymbol{\mu}, 1]^{\mathrm{T}}$, and covariance matrix is $\mathbf{\Sigma}' = \mathbf{JW\Sigma W}^{\mathrm{T}}\mathbf{J}^{\mathrm{T}}$, where $\mathbf{J}$ is the Jacobian of the affine approximation of the projective transformation. Finally, the rendered color C at pixel $\mathbf{u}$ is obtained through blending N ordered points overlapping the pixel:

$$C = \sum_{i=1}^{N} T_i \alpha_i c_i, T_i = \prod_{j=1}^{i-1}(1-\alpha_j), \tag{2}$$

where $\alpha_i = G(\mathbf{u}|\boldsymbol{\mu}_i', \mathbf{\Sigma}_i')o_i$, and c_i, o_i are the color and opacity properties of each 3D Gaussian. c_i is represented using spherical harmonics (SH).

B. Segmentation-guided 3D Gaussian Distribution Optimization

In 3D Gaussian Splatting [1], adaptive control is proposed to populate regions with missing geometric features and regions where Gaussians cover large areas in the scene. They densify Gaussians with an average magnitude of view-space positional gradients above a threshold ϵ. Their method is able to reconstruct the most detailed regions with the adaptive control of Gaussians.

However, it is hard to densify Gaussians in regions that rarely appear in all images. Because in original 3DGS [1], the Gaussian representation tends to resemble a global representation. In other words, a Gaussian is associated with color computations from various views. This makes it challenging to match the designated threshold for densification after the computation of cumulative gradients, even if significant gradients are generated in certain views. As a result, regions that are heavily occluded do not undergo sufficient optimization. As illustrated in Fig. 1, the grass under the bench and the clock behind the television occur infrequently in all images, leading to artifacts and blurring in the rendered results. More results are discussed in Sec. IV-D.

To better find the regions that need to be densified, we propose a segmentation-guided 3D Gaussian distribution optimization. First, for an input image $\mathbf{I}$, we leverage the open-set segmentation method [9] to extract K segmentation regions $\{\mathbf{M}_i, i = 1, ..., K\}$. Then, to obtain the contribution of each Gaussian G_j made in region $\mathbf{M}_i$ to the color of pixel $\mathbf{u_i}$, we keep track of the corresponding weight $w_j^{\mathbf{u_i}} = \alpha_j T_j$ for G_j at pixel $\mathbf{u_i}$ during the rendering process, and assign a weight

$$w_j^i = \max_{w_j^{\mathbf{u_i}}} w_j^{\mathbf{u_i}}, j \in \{1, ..., |P|\} \tag{3}$$

on Gaussian G_j, which represents the maximum contribution of Gaussian G_j to the region $\mathbf{M}_i$, $\|p\|$ is the number of Gaussians.

For each region $\mathbf{M}_i$, we calculate the mean (per-pixel) L_1 error $\mathcal{L}_{M_i} = \frac{1}{\|\mathbf{M}_i\|_0}\|(\mathbf{I} - \hat{\mathbf{I}})\mathbf{M}_i\|_1$ between the region $\mathbf{M}_i$ of ground truth image $\mathbf{I}$ and rendered image $\hat{\mathbf{I}}$. If $\mathcal{L}_{M_i}$ is greater than the mean L_1 error $\mathcal{L}_1 = \frac{1}{\|\mathbf{I}_i\|_0}\|(\mathbf{I} - \hat{\mathbf{I}})\|_1$ for the whole

Fig. 3. The comparisons of ours and other methods and the corresponding ground truth images. The scenes from top to bottom are: FLOWERS and ROOM from the Mip-NeRF360 dataset; DRJOHNSON and PLAYROOM from the Deep Blending dataset [57]; TRUCK and TRAIN from the Tanks&Temples dataset [58].

image, Gaussians that lie in that region should be densified, as this indicates segmentation regions not sufficiently well reconstructed. To prevent excessive densification of Gaussians, we involve Gaussians with weights $w_j^i > 0.5$ that contribute significantly to the final color in the densification process. As shown in Fig. 1, our optimization can reconstruct more details than 3DGS [1] and Scaffold-GS [2].

C. Spatial Regularization

To achieve a compact and efficient representation of the scene, we introduce two spatial regularization terms for 3D Gaussians: repulsion loss $\mathcal{L}_{rep}$ and smoothness loss $\mathcal{L}_{smooth}$. These losses jointly encourage a more uniform Gaussian distribution while preserving geometric details and reducing redundancy.

A key observation is that, in the standard 3DGS framework, the spatial distribution of Gaussians often does not align well with underlying geometric structures. Specifically, due to the absence of strong geometric constraints, Gaussians can become excessively clustered in some regions while leaving other areas under-sampled. Furthermore, outlier Gaussians may persist and contribute to unwanted artifacts in the final rendered result. To address these issues, we leverage local geometry information to construct an adaptive spatial regularization.

For each Gaussian G_i, we estimate a local geometric plane by computing the principal components of its Q nearest neighbors $B(i)$ using Principal Component Analysis (PCA) following DSS [48]. The normal of this plane is denoted as $\hat{\mathbf{n}}$, and $\hat{\mathbf{u}}_i, \hat{\mathbf{v}}_i$ are the first 2 principal components. First, we define the repulsion loss $\mathcal{L}_{rep}$, which prevents excessive clustering of Gaussians and encourages a more uniform distribution along the local surface:

$$\mathcal{L}_{rep} = \frac{1}{|P|} \sum_{i=1}^{|P|} \sum_{j \in B(i)} -o_j e^{-\frac{r_{i,j}^2}{h^2}} \tag{4}$$

where $\boldsymbol{\mu}_i$ and $\boldsymbol{\mu}_j$ denote the centers of Gaussians G_i and G_j, respectively. h is a hyper-parameter to represent finite support diameter [59], $|P|$ is the number of Gaussians. The term o_j represents the opacity of G_j, ensuring that the loss is primarily influenced by the significant Gaussians. Importantly, instead of using the full Euclidean distance, we introduce a bilateral

weighting term based on the local geometry: the term $r_{i,j} = \|\mathbf{\Pi}_i(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)\|_2$ projects the center difference vector onto the local plane, and $\mathbf{\Pi}_i = [\hat{\mathbf{u}}_i\hat{\mathbf{v}}_i][\hat{\mathbf{u}}_i\hat{\mathbf{v}}_i]^{\mathrm{T}}$. By focusing the repulsion effect on the tangent plane, we prevent unnecessary movement in the normal direction, thereby avoiding the displacement of Gaussians away from the underlying geometry.

However, relying solely on repulsion loss may lead to excessive separation, causing a loss of geometric detail. To mitigate this issue, we introduce the smoothness loss $\mathcal{L}_{smooth}$, which encourages Gaussians to remain close to the estimated local surface by minimizing their distance to it. Specifically, we minimize the distance between a Gaussian's center and the estimated local plane fitted by its neighboring Gaussians:

$$\mathcal{L}_{smooth} = \frac{1}{|P|}\sum_{i=1}^{|P|}\sum_{j\in B(i)} o_j d_{ij}^2, \quad d_{ij} = \hat{\mathbf{n}}\cdot(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j) \quad (5)$$

This constraint helps prevent excessive deviations from the local surface, reducing outlier Gaussians and improving the smoothness of reconstructed geometry.

By jointly optimizing these two regularization terms, our method strikes a balance between uniform Gaussian distribution and geometric detail preservation. The repulsion loss prevents excessive clustering, while the smoothness loss prevents Gaussians from deviating too far from the local surface. As a result, our full model achieves a more compact and well-structured Gaussian representation, leading to enhanced rendering quality with fewer redundant Gaussians. The ablation results are presented in Sec. IV-B.

D. Optimization Details

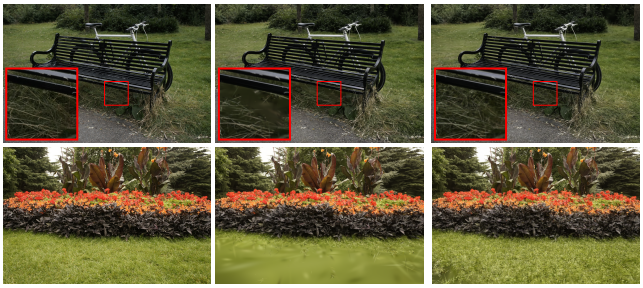
The total loss function is $\mathcal{L}_1$ combined with a D-SSIM term and two regularization terms:

$$\mathcal{L} = (1-\lambda_{ssim})\mathcal{L}_1 + \lambda_{ssim}\mathcal{L}_{D-SSIM} + \lambda_{rep}\mathcal{L}_{rep} + \lambda_{smooth}\mathcal{L}_{smooth} \quad (6)$$

We use $\lambda_{ssim} = 0.2, \lambda_{rep} = \lambda_{smooth} = 0.1$ in all our experiments. Similar to 3DGS [1], we start by optimizing the zero-order component and then introduce an additional band of the SH after every 1000 iterations until all 4 bands of SH are represented. We first run segmentation-guided optimization for 15K iterations with densification every 500 iterations and then run optimization with two geometry spatial regularization terms for another 15K iterations. In other words, these two regularization terms start to apply after 15,000 iterations. Because the computation of k-nearest neighbors (KNN) is time-consuming, we update and recalculate the KNN every 100 iterations. Due to different scene sizes, we set $h = 0.05, Q = 15$ for real-world scenes, and $h = 0.005, Q = 5$ for the synthetic *Blender* dataset.

IV. Experiment

We conduct experiments to demonstrate the effectiveness of our method. First, we compare the results of novel view synthesis, including real and synthetic data. For real data, we use the following datasets: Mip-NeRF360 [17], Deep Blending [57], and Tanks&Temples [58]. We compare against the following baselines: Plenoxels [32], InstantNGP [23], Mip-NeRF360 [17], ZIP-NeRF [18], Point-NeRF [30], 3DGS [1], AbsGS [4], GOF [5], Mip-Splatting [43] and Scaffold-GS [2]. We also show the results of the Scaffold-GS combined with our method.



(a) GT (b) Scaffold-GS (c) Scaffold-GS+Ours

Fig. 4. The comparisons of Scaffold-GS and the combination with our method and the corresponding ground truth images. The scenes are FLOWERS and ROOM from the Mip-NeRF360 dataset.

Next, we conduct some ablation experiments to explain our design choices. We demonstrate the effectiveness of our segmentation-guided approach, the two regularization terms, and the comparison with different initializations. Finally, we present two further analyses: a limited budget experiment to evaluate the efficiency of our method under fixed Gaussian capacity, and a visibility analysis to explore how semantic guidance affects performance in occluded or sparsely visible regions. The evaluation metrics we used for comparison include: PSNR (Peak Signal-to-Noise Ratio) to measure the quality of synthesized images, SSIM (Structural Similarity Index) to quantify the structural similarity between two images, and LPIPS (Learned Perceptual Image Patch Similarity) which considers human perception of image quality by calculating the perceptual distance between image patches.

A. Novel View Synthesis

In this task, our objective is to input a set of images with given camera poses and synthesize an image from a new viewpoint. We conduct experiments on both real-world data and synthetic data.

a) Real-World Scenes: First, we perform experiments in real-world scenes. Similar to Mip-Nerf360 [17], we adopt the dataset partitioning method that selects every 8th image for testing. The results are presented in Table I. The results of our method were obtained after training for 30K iterations, following the same protocol as 3DGS. The detailed numbers for each scene can be found in the supplemental material.

In a quantitative comparison involving 9 real scenes from the Mip-NeRF360 dataset [17], our method achieves state-of-the-art real-time performance in terms of SSIM and LPIPS. Although Zip-NeRF [18] achieves higher PSNR scores, it requires significantly longer training time and does not support real-time rendering. In contrast, our method provides a more efficient trade-off between quality and speed, offering fast training and real-time rendering capability.

In terms of visual quality as shown in Fig. 3, our method demonstrates significant improvements over 3DGS with fewer

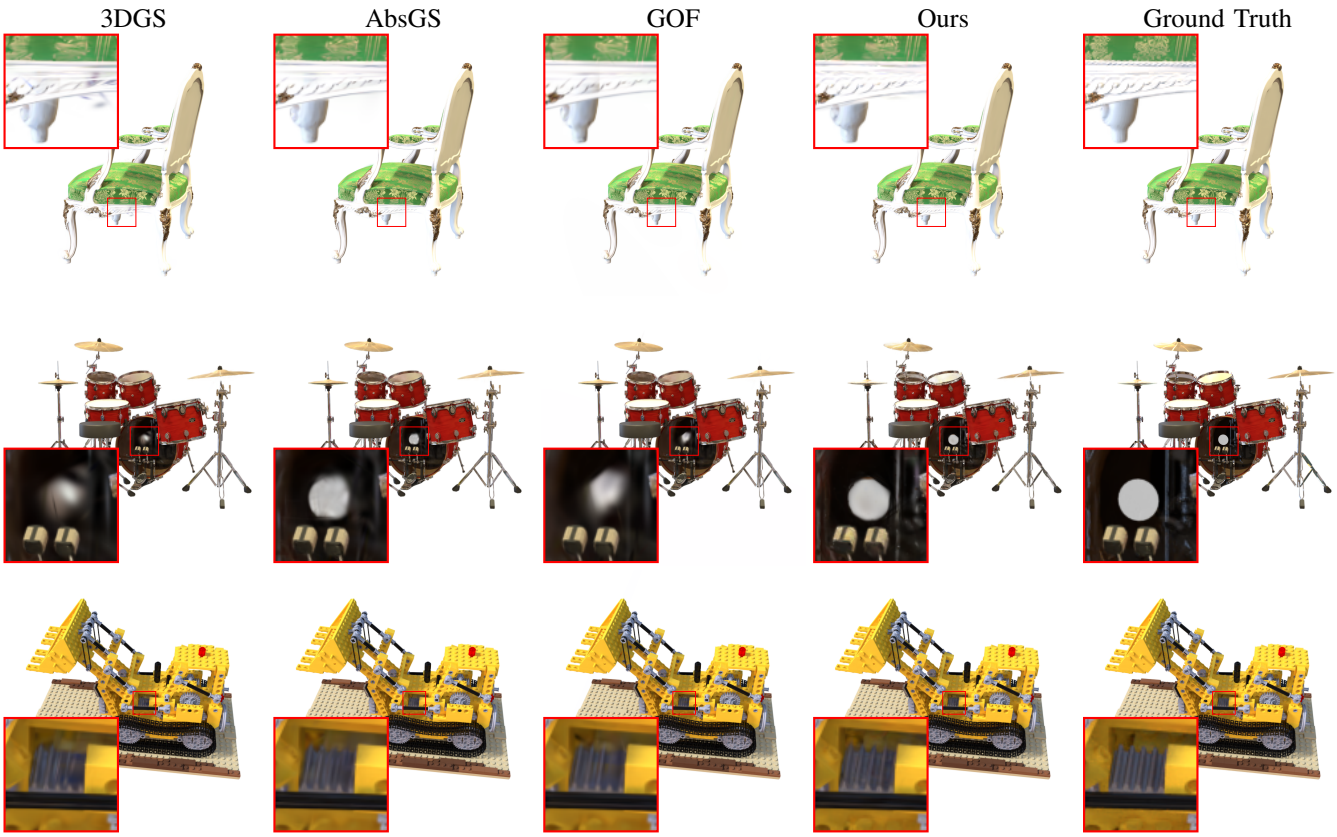


Fig. 5. The comparisons of our method with other methods and the corresponding ground truth images on synthetic *Blender* dataset. Our method can reconstruct more details than other methods, such as the gear in LEGO scene.

TABLE I
QUANTITATIVE COMPARISON OF OUR METHOD WITH PREVIOUS WORK, COMPUTED OVER THREE REAL DATASETS. RESULTS OF PREVIOUS NERF-BASED WORK ARE DIRECTLY QUOTED FROM THE ORIGINAL 3DGS PAPER. AND RESULTS OF OTHER METHODS ARE QUOTED FROM THEIR PAPERS WHENEVER AVAILABLE. THE 1ST, 2ND, AND 3RD-BEST PERFORMANCES ARE INDICATED BY RED, ORANGE, AND YELLOW HIGHLIGHTS RESPECTIVELY.

Dataset	Mip-NeRF360				Tanks&Temples				Deep Blending			
Metric	$SSIM^{\uparrow}$	$PSNR^{\uparrow}$	$LPIPS^{\downarrow}$	Mem	$SSIM^{\uparrow}$	$PSNR^{\uparrow}$	$LPIPS^{\downarrow}$	Mem	$SSIM^{\uparrow}$	$PSNR^{\uparrow}$	$LPIPS^{\downarrow}$	Mem
Plenoxels	0.626	23.08	0.463	2.1GB	0.719	21.08	0.379	2.3GB	0.795	23.06	0.510	2.7GB
INGP-Base	0.671	25.30	0.371	13MB	0.723	21.72	0.330	13MB	0.797	23.62	0.423	13MB
INGP-Big	0.699	25.59	0.331	48MB	0.745	21.92	0.305	48MB	0.817	24.96	0.390	48MB
Mip-NeRF 360	0.792	27.69	0.237	8.6MB	0.759	22.22	0.257	8.6MB	0.901	29.40	0.245	8.6MB
Zip-NeRF	0.828	28.54	0.189	908M	0.839	23.64	0.100	908M	0.911	30.00	0.227	908M
3DGS	0.815	27.21	0.214	734MB	0.841	23.14	0.183	411MB	0.903	29.41	0.243	676MB
AbsGS	0.820	27.49	0.191	728MB	0.853	23.73	0.162	304MB	0.902	29.67	0.236	444MB
GOF	0.825	27.42	0.233	690MB	0.827	21.01	0.131	398MB	0.875	28.14	0.275	437MB
Ours	0.822	27.70	0.203	716MB	0.856	23.95	0.160	369MB	0.905	29.81	0.243	319MB
Scaffold-GS	0.807	27.51	0.239	178MB	0.852	23.96	0.176	77.6MB	0.905	30.21	0.254	54.0MB
Scaffold-GS+Ours	0.809	27.68	0.218	159MB	0.854	24.06	0.174	71.5MB	0.907	30.30	0.251	39.2MB
Mip Splatting	0.827	27.78	0.201	800M	0.826	23.09	0.140	476M	0.903	29.40	0.239	839M
Mip Splatting + Ours	0.830	27.94	0.198	732M	0.830	23.33	0.135	447M	0.904	29.75	0.234	772M

Gaussians, particularly in detail-rich regions such as the grass in FLOWER scene and the guitar in the ROOM scene.

Compared to AbsGS [4] and GOF [5], which proposed a new metric that accumulates the norms of the individual pixel gradients, our method achieves better quality, such as the detailed representation of a chair in the DRJOHNSON scene from the Deep Blending dataset [57] and the wheels in the TRAIN scene. This is mainly because both AbsGS and GOF still densify Gaussians when the *average* magnitude of view-space positional gradients is above a threshold, even though they take the absolute value of each gradients before averaging. It struggles to capture the necessary details in sparsely observed regions, where there is not enough gradient for effective densification. In contrast, our method calculates the errors of segment regions in each training image, and if a region's average reconstruction error is greater than that of the whole image, Gaussians lying in that region should be densified, as this indicates these regions are not sufficiently well reconstructed.

In the Tanks&Temples dataset [58] including TRUCK scene and TRAIN scene, our method also demonstrates superior visual quality and can reconstruct more details than others, such as the steering wheel in scene TRUCK.

We also compare the results of the Scaffold-GS and the

TABLE II
PSNR SCORES FOR SYNTHETIC *Blender* DATASET [3], WE START WITH 100K RANDOMLY INITIALIZED POINTS THE SAME AS 3DGS.

	Mic	Chair	Ship	Materials	Lego	Drums	Ficus	Hotdog	Avg.
Plenoxels	33.26	33.98	29.62	29.14	34.10	25.35	31.83	36.81	31.76
INGP-Base	36.22	35.00	31.10	29.78	36.39	26.02	33.51	37.40	33.18
Mip-NeRF	36.51	35.14	30.41	30.71	35.70	25.48	33.29	37.48	33.09
Point-NeRF	35.95	35.40	30.97	29.61	35.04	26.06	36.13	37.30	33.30
ZIP-NeRF	35.15	34.84	31.38	31.66	34.84	25.84	33.90	37.14	33.09
3DGS	35.36	35.83	30.80	30.00	35.78	26.15	34.87	37.72	33.32
AbsGS	36.02	36.00	30.78	29.97	35.77	26.08	34.74	37.67	33.38
GOF	36.06	36.18	30.67	30.19	35.56	26.17	35.01	37.45	33.41
Ours	35.38	36.07	30.81	30.26	35.84	26.21	34.96	37.83	33.42
Scaffold-GS	36.32	35.13	30.15	30.30	34.89	26.31	34.40	37.65	33.14
Scaffold-GS+Ours	35.36	35.44	30.38	30.41	35.06	26.63	34.80	37.75	33.35
MIP-Splatting	35.55	35.70	30.78	30.12	35.45	26.14	35.12	37.78	33.33
MIP-Splatting+Ours	36.63	35.66	31.56	30.45	36.44	26.33	35.54	38.01	33.82

combination with our method, as shown in Fig. 4 and Table I. Combining Scaffold-GS with our method results in improved performance and fewer Gaussians, such as enhanced detail in the weeds under a chair in the BICYCLE scene. Scaffold-GS adopts the same splitting strategy as the original 3DGS, which densifies Gaussians based on the average magnitude of view-space positional gradients exceeding a specified threshold. This approach often fails to adequately capture essential details in regions with sparse observations. This improvement can be attributed to our method's local density control.

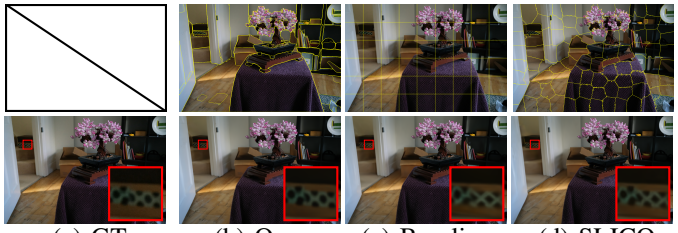
b) Synthetic Bounded Scenes: In the evaluation of the synthetic *Blender* dataset [3], we present the quantitative results in Table II. Our method achieves comparable results, starting from random initialization, which is consistent with the approach used in 3DGS. For qualitative evaluation, please refer to Fig. 5. Our method showcases superior visual quality, particularly in capturing fine details. This is because other methods rely on the average view-space gradient magnitude for Gaussian splitting. However, since each Gaussian contributes to multiple regions through volume rendering, averaging gradients over many views smooths out local variations, suppressing necessary densification in high-gradient areas, leading to reconstruction artifacts as a result. Our method overcomes this by directly comparing local and global reconstruction errors, ensuring targeted densification where needed, thus preserving finer details more effectively. For example, in the CHAIR scene, both AbsGS and GOF exhibit imperfections in the reconstruction of the seat edge, leading to penetration artifacts on the chair legs from certain viewpoints. Meanwhile, the 3DGS method introduces some floater artifacts. And our method reconstructs the vent hole of the bass drum with sharper and more complete edges in the DRUMS scene and the detailed gear in the LEGO scene compared to other methods.

B. Ablation Studies

In this section, we conduct ablation experiments on two design choices: segmentation-guided optimization and spatial regularization. For segmentation-guided optimization, we focus on demonstrating the necessity of incorporating segmentation information in our method. We replace the segmentation mask in our approach with a patch mask to illustrate the importance of segmentation information. This highlights how segmentation information plays a crucial role in guiding the optimization process. Regarding spatial regularization terms, we conduct experiments where we remove these terms to demonstrate their necessity. By comparing the results without spatial regularization to our full method, we showcase the importance of incorporating spatial regularization to improve the synthesis quality and preserve fine details in the generated images.

TABLE III
QUANTITATIVE COMPARISONS WITH THE BASELINE THAT DIVIDES IMAGES INTO 9×6 PATCHES, SUPERPIXEL BASED SEGMENTATION METHOD SLICO [60] AND WITHOUT REPULSION AND SMOOTHNESS LOSS ON MIP-NERF360 DATASET.

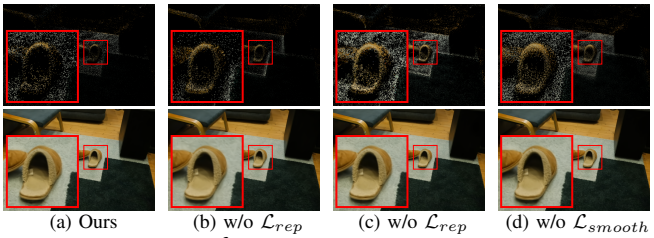
Settings	$SSIM^{\uparrow}$	$PSNR^{\uparrow}$	$LPIPS^{\downarrow}$
Baseline	0.808	27.30	0.210
SLICO	0.813	27.41	0.208
SAM	0.815	27.58	0.207
SAM-HQ	0.819	27.66	0.204
Ours	**0.821**	**27.70**	**0.202**
w/o $\mathcal{L}_{rep}$ + $\mathcal{L}_{smooth}$	0.813	27.59	0.209
w/o $\mathcal{L}_{rep}$	0.814	27.62	0.205
w/o $\mathcal{L}_{smooth}$	0.818	27.64	0.204



(a) GT (b) Ours (c) Baseline (d) SLICO

Fig. 6. Qualitative comparison of novel view synthesis with the baseline and superpixel-based method SLICO. (b) Our method can capture details, such as the stair and the box behind the door in the BONSAI scene of MipNeRF360, which are hard to capture for the (c) patch-based and (d) superpixel-based optimization.

a) Segmentation-guided Optimization: Our method incorporates segmentation masks generated by Semantic-SAM [9], which contain multi-level semantic information. To assess the significance of this information, we design masks without semantic information and utilize them for locally densifying the scene, following the same approach as our original method. We divide the images into 9×6 patches to ensure a similar number of masks as generated by Semantic-SAM [9], and use superpixel-based method SLICO [60] to produce masks. We also compare with SAM-based method, including SAM [10] and SAM-HQ [51]. The quantitative results are presented in Table III. Different SAM-based segmentation methods have slight impact on performance, but all are better than patch-based and superpixel-based segmentation. SAM-HQ produces more accurate instance segmentation results than SAM, but Semantic-SAM is able to produce more part level segmentation, which has the best performance. In Fig. 6, (b) Our method effectively captures fine details, such as the stair and the box behind the door in the BONSAI scene of Mip-NeRF360, which are challenging for the (c) patch-based and (d) superpixel-based optimization methods. This is because both baseline and SLICO methods produce segmentation with weaker semantic consistency. Typically, regions with the same semantics exhibit similar textures and materials. In this case, our method correctly segments the patterned area of the box as a distinct region, whereas the other methods merge it with



(a) Ours (b) w/o $\mathcal{L}_{rep}$ $+\mathcal{L}_{smooth}$ (c) w/o $\mathcal{L}_{rep}$ (d) w/o $\mathcal{L}_{smooth}$

Fig. 7. Qualitative comparison of Gaussians distribution without repulsion loss and smoothness loss in the ROOM scene of MipNeRF360. Our full method (a) achieves a more efficient spatial distribution of Gaussians. Our method with fewer Gaussians maintains comparable rendering quality to (b), which removes both the repulsion loss and smoothness loss. Without the repulsion loss (c), some Gaussians are located very close to each other, resulting in an uneven distribution and redundant Gaussians. Without the smoothness loss (d), the influence of the repulsion loss enforces a more uniform Gaussian distribution but at the cost of less precise geometric boundaries, also resulting in redundant Gaussians. The interplay between these two losses enables a more efficient Gaussian distribution while preserving rendering quality.

different areas, particularly the black top surface of the box. This reduces the localized reconstruction error, leading to suboptimal densification. Consequently, our method provides a more accurate and detailed representation, producing a shape closer to the ground truth's circular form, while the other methods yield a distorted, diamond-like shape.

From another perspective, the segmentation results of the baseline and SLICO methods, as shown in Table III, can be considered as cases of suboptimal or incorrect segmentation. Despite this, their PSNR and LPIPS scores are higher than those of the original 3DGS. This is mainly because of different densification strategies. This strategy helps improve the reconstruction quality even when segmentation results are not ideal.

b) Regularization: The effectiveness of the regularizations is demonstrated in Table III and Fig. 7. In Fig. 7, we intentionally set the scale to a small value to better visualize the Gaussian distribution. Our full method (a) achieves a more efficient spatial distribution of Gaussians. Our method with fewer Gaussians maintains comparable rendering quality to (b), which removes both the repulsion loss and smoothness loss. Without the repulsion loss (c), some Gaussians are located very close to each other, resulting in an uneven distribution and redundant Gaussians. Without the smoothness loss (d), the influence of the repulsion loss enforces a more uniform Gaussian distribution but at the cost of less precise geometric boundaries, also resulting in redundant Gaussians. By integrating both repulsion and smoothness regularizations, our method effectively balances Gaussian distribution uniformity and detail preservation, reducing redundant points while maintaining high-fidelity geometry and enhancing rendering accuracy compared to 3DGS.

c) Initialization from Structure from Motion (SfM): we conducted experiments comparing 3DGS and our method under random point initialization while keeping the same poses. As shown in Fig. 8, our method consistently produces better reconstructions in such cases. For instance, in the PLAYROOM scene from the Deep Blending dataset, our approach successfully recovers the wall socket even under random initialization, whereas 3DGS fails to reconstruct them. However, in large-scale outdoor scenes like TRAIN from Tanks&Temples dataset, our method struggles to reconstruct distant structures (e.g., antenna towers) under random initialization. In such scenarios, COLMAP-derived initialization proves critical for convergence and fine reconstruction, as it compensates for camera pose uncertainty and large depth ranges. These findings suggest that while our semantic-guided densification improves robustness under weak geometry, random initialization remains limited in the presence of inaccurate poses and large-scale depth variation.

TABLE IV
PSNR COMPARISON UNDER LIMITED GAUSSIAN BUDGET ON THE DEEP BLENDING DATASET. OUR METHOD ACHIEVES HIGHER RECONSTRUCTION QUALITY THAN THE ORIGINAL 3DGS UNDER THE SAME GAUSSIAN COUNT BY ALLOCATING REPRESENTATIONAL CAPACITY MORE EFFECTIVELY.

#Gaussians	30K	60K	90K	120K	150K
3DGS	26.41	27.17	27.62	27.98	28.25
Ours	**26.92**	**27.54**	**27.89**	**28.14**	**28.51**

C. Limited Budget Analysis

To evaluate the efficiency of our densification strategy under constrained capacity, we conduct experiments on the Deep Blending dataset by enforcing an upper bound on the total number of Gaussians during training. Specifically, we set a fixed maximum point count per scene and prevent further densification once this threshold is reached. Note that the pruning mechanism may remove redundant Gaussians, allowing densification to resume when the total count drops below the limit.

We compare our method with the original 3DGS under the same budget constraint, using default hyperparameters for both methods without additional tuning. As shown in Table IV, we report rendering quality (PSNR) across different Gaussian budget levels. Our method consistently achieves higher PSNR under the same point count, demonstrating that semantic-guided densification more effectively allocates representational capacity to semantically under-optimized regions, resulting in improved reconstruction quality without increasing model complexity.

In terms of training time, our method introduces a moderate overhead due to segmentation mask projection and neighborhood-based regularization. For instance, on the BICYCLE scene with an RTX 3090 GPU, the training times for 3DGS, Scaffold-GS, and our method are approximately 25, 35, and 40 minutes, respectively. This increase is comparable to other improved 3DGS variants such as Scaffold-GS, which also trade slight training overhead for better rendering quality. Moreover, we note that in practical applications, training is typically a one-time process per scene, while real-time rendering is performed many times thereafter. Hence, rendering efficiency and quality are usually more critical than minor differences in training time.

(a) GT (b) 3DGS-Random (c) 3DGS-SfM (d) Ours-Random (e) Ours-SfM

Fig. 8. Comparison of reconstruction results under random and COLMAP (SfM)-based point initialization. Our method consistently outperforms 3DGS in both situations. In large-scale real-world scenes, COLMAP-based point initialization provides geometry priors that benefit the reconstruction, especially for distant structures.

(a) Upper views

(b) Lower views

Fig. 9. We constructed a new scene to assess the visibility issue of 3DGS. The scene we built incorporates natural occlusion, comprising a double board table with two decorative bottles placed on top and a waffle cake with a strawberry on the lower board. In this scene, the waffle cake is only visible from lower views (b), not from upper views (a) due to occlusion by the upper board.

(a) GT (b) Ours (c) 3DGS

Fig. 10. Comparison between our method and 3DGS for different levels of visibility. The first row is trained with 3 lower views and 100 upper views. The second row is trained with 5 lower views and 100 upper views. Our method exhibits fewer blur artifacts compared to the 3DGS.

D. Visibility

We constructed a new scene to assess the visibility issue of 3DGS, examining the performance of our method in comparison to the original 3DGS when the number of visible training views of specific objects or parts in a scene is reduced.

The scene we built incorporates natural occlusion, comprising a double board table with two decorative bottles placed on top and a waffle cake with a strawberry on the lower board. In this scene, the waffle cake is only visible from lower views, not from upper views due to occlusion by the upper board (as depicted in Fig. 9). To investigate various scenarios, we created five different training sets with varying numbers of lower views: 2, 3, 5, and 10. In the training set, we used 100 upper views along with the selected lower views, reserving 25 lower views for testing. This dataset is unbalanced, as only a few views can observe the lower waffle cake. By reducing visibility, we can evaluate the performance degradation of our method compared to the 3DGS.

Comparison results presented in Fig. 10 demonstrate that our method exhibits fewer blur artifacts compared to the 3DGS. Furthermore, Table V illustrates that our method is more robust and achieves better performance even with a reduced number of training lower views.

In Fig. 11, we present the densification process of our method and the 3DGS. The top row represents our method, while the bottom row corresponds to the 3DGS. This visualization showcases the results achieved through training on a 5-lower views dataset. The densification process is depicted from left to right. Ours is more consistent with real geometry.

V. Conclusion

In this work, we propose a novel method for refining the spatial distribution of 3D Gaussians within the framework of 3D Gaussian Splatting (3DGS) to enhance the quality of high-fidelity novel view synthesis. Our approach leverages semantic information to identify poorly optimized regions and employs splitting or cloning operations on the corresponding 3D Gaussians to improve their representation. We have introduced two spatial regularization terms, repulsion loss, and smoothness loss, to ensure a more regular and efficient distribution of Gaussians with reduced overlap and outliers. These regularization terms contribute to enhancing the overall spatial distribution of 3D Gaussians and improving the synthesized geometry.

(a) Ours

(b) 3DGS

Fig. 11. Visualization of the densification process. This visualization showcases the results achieved through training on a 5-lower views dataset. The densification process is depicted from left to right for every 6000 iterations. Ours is more consistent with real geometry.

TABLE V
VISIBILITY QUANTITATIVE COMPARISON OF OUR METHOD AND 3DGS. BECAUSE THIS IS A RELATIVELY SIMPLE SCENE, WHEN OUR TRAINING SET HAS 10 LOWER VIEWS, OUR METHOD PERFORMS ALMOST THE SAME AS THE 3DGS. BUT WHEN THE TRAINING LOWER VIEWS ARE REDUCED, THE 3DGS DECREASES FASTER THAN OUR METHOD.

Dataset	2-lower views			3-lower views			5-lower views			10-lower views		
Metric	$SSIM$	$PSNR$	$LPIPS$	$SSIM$	$PSNR$	$LPIPS$	$SSIM$	$PSNR$	$LPIPS$	$SSIM$	$PSNR$	$LPIPS$
GS-30K	0.979	26.25	0.028	0.979	26.21	0.029	0.991	31.74	0.015	**0.997**	45.72	**0.004**
Ours-30K	**0.983**	**29.20**	**0.025**	**0.989**	**30.53**	**0.017**	**0.993**	**35.41**	**0.009**	**0.997**	**46.11**	**0.004**

Experimental results on various datasets demonstrate the effectiveness of our method. Our approach achieves state-of-the-art real-time rendering performance in terms of metrics such as SSIM, LPIPS, and PSNR, while also delivering superior visual quality compared to baselines. In particular, our method excels in reconstructing details in sparsely observed regions, which significantly enhances the representation of intricate parts of scenes.

However, our method still has limitations. Our method does not model lighting and reflections explicitly, the same as 3DGS, so our method can hardly capture glossy surfaces. These can be addressed by using physically based rendering like GS-IR [37] and GaussianShader [36]. Also, using more efficient representations [40]–[42] can get a more compressed result. And our method requires longer training time due to KNN search of regularization term, which can be mitigated by using more efficient implementation methods. Furthermore, an exciting future direction is to explore semantic-level control to further enhance reconstruction quality. Our current method focuses on a general approach to adaptive Gaussian placement, but extending it to incorporate object-specific constraints—such as assigning different Gaussian attributes based on material properties—could significantly improve realism and controllability. We believe that our method opens new avenues for future research, leveraging semantic segmentation to improve not only the accuracy of 3D scene reconstruction but also its adaptability to complex materials and object structures.

REFERENCES

[1] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3D Gaussian splatting for real-time radiance field rendering," *ACM Transactions on Graphics*, vol. 42, no. 4, July 2023. [Online]. Available: https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/

[2] T. Lu, M. Yu, L. Xu, Y. Xiangli, L. Wang, D. Lin, and B. Dai, "Scaffold-GS: Structured 3D Gaussians for view-adaptive rendering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[3] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing scenes as neural radiance fields for view synthesis," in *ECCV*, 2020.

[4] Z. Ye, W. Li, S. Liu, P. Qiao, and Y. Dou, "AbsGS: Recovering fine details in 3D Gaussian splatting," in *Proceedings of the 32nd ACM International Conference on Multimedia*, 2024, pp. 1053–1061.

[5] Z. Yu, T. Sattler, and A. Geiger, "Gaussian opacity fields: Efficient adaptive surface reconstruction in unbounded scenes," *ACM Transactions on Graphics (TOG)*, vol. 43, no. 6, pp. 1–13, 2024.

[6] S. Kheradmand, D. Rebain, G. Sharma, W. Sun, Y.-C. Tseng, H. Isack, A. Kar, A. Tagliasacchi, and K. M. Yi, "3D Gaussian splatting as Markov chain Monte Carlo," *Advances in Neural Information Processing Systems*, vol. 37, pp. 80 965–80 986, 2024.

[7] C. Häne, C. Zach, A. Cohen, and M. Pollefeys, "Dense semantic 3D reconstruction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 9, pp. 1730–1743, 2016.

[8] ——, "Joint 3D scene reconstruction and class segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013.

[9] F. Li, H. Zhang, P. Sun, X. Zou, S. Liu, J. Yang, C. Li, L. Zhang, and J. Gao, "Semantic-SAM: Segment and recognize anything at any granularity," *arXiv preprint arXiv:2307.04767*, 2023.

[10] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, "Segment anything," *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

[11] R. A. Drebin, L. Carpenter, and P. Hanrahan, "Volume rendering," *ACM Siggraph Computer Graphics*, vol. 22, no. 4, pp. 65–74, 1988.

[12] M. Levoy, "Efficient ray tracing of volume data," *ACM Transactions on Graphics (TOG)*, vol. 9, no. 3, pp. 245–261, 1990.

[13] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "DeepSDF: Learning continuous signed distance functions for shape representation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[14] Z. Chen and H. Zhang, "Learning implicit fields for generative shape modeling," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[15] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3D reconstruction in function space," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[16] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. 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.

[17] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. 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.

[18] ——, "Zip-NeRF: Anti-aliased grid-based neural radiance fields," *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

[19] A. Chen, Z. Xu, A. Geiger, J. Yu, and H. Su, "TensoRF: Tensorial radiance fields," in *European Conference on Computer Vision*. Springer, 2022, pp. 333–350.

[20] S. Fridovich-Keil, A. Yu, M. Tancik, Q. Chen, B. Recht, and A. Kanazawa, "Plenoxels: Radiance fields without neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5501–5510.

[21] L. Liu, J. Gu, K. Zaw Lin, T.-S. Chua, and C. Theobalt, "Neural sparse voxel fields," *Advances in Neural Information Processing Systems*, vol. 33, pp. 15 651–15 663, 2020.

[22] C. Sun, M. Sun, and H.-T. Chen, "Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5459–5469.

[23] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Transactions on Graphics (ToG)*, vol. 41, no. 4, pp. 1–15, 2022.

[24] W. Hu, Y. Wang, L. Ma, B. Yang, L. Gao, X. Liu, and Y. Ma, "Tri-MipRF: Tri-mip representation for efficient anti-aliasing neural radiance fields," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 19 774–19 783.

[25] J. Kulhanek and T. Sattler, "Tetra-NeRF: Representing neural radiance fields using tetrahedra," *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

[26] R. Li, H. Gao, M. Tancik, and A. Kanazawa, "NerfAcc: Efficient sampling accelerates NeRFs," *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

[27] A. Yu, R. Li, M. Tancik, H. Li, R. Ng, and A. Kanazawa, "Plenoctrees for real-time rendering of neural radiance fields," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5752–5761.

[28] E. R. Chan, C. Z. Lin, M. A. Chan, K. Nagano, B. Pan, S. De Mello, O. Gallo, L. J. Guibas, J. Tremblay, S. Khamis *et al.*, "Efficient geometry-aware 3D generative adversarial networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16 123–16 133.

[29] Z. Chen, T. Funkhouser, P. Hedman, and A. Tagliasacchi, "MobileNeRF: Exploiting the polygon rasterization pipeline for efficient neural field rendering on mobile architectures," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[30] Q. Xu, Z. Xu, J. Philip, S. Bi, Z. Shu, K. Sunkavalli, and U. Neumann, "Point-NeRF: Point-based neural radiance fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5438–5448.

[31] L. Yariv, P. Hedman, C. Reiser, D. Verbin, P. P. Srinivasan, R. Szeliski, J. T. Barron, and B. Mildenhall, "BakedSDF: Meshing neural SDFs for real-time view synthesis," *SIGGRAPH*, 2023.

[32] Sara Fridovich-Keil and Alex Yu, M. Tancik, Q. Chen, B. Recht, and A. Kanazawa, "Plenoxels: Radiance fields without neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[33] M. Zwicker, H. Pfister, J. Van Baar, and M. Gross, "EWA splatting," *IEEE Transactions on Visualization and Computer Graphics*, vol. 8, no. 3, pp. 223–238, 2002.

[34] ——, "EWA volume splatting," in *Proceedings Visualization, 2001. VIS'01.* IEEE, 2001, pp. 29–538.

[35] J. Zhang, F. Zhan, M. Xu, S. Lu, and E. Xing, "FreGS: 3D Gaussian splatting with progressive frequency regularization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[36] Y. Jiang, J. Tu, Y. Liu, X. Gao, X. Long, W. Wang, and Y. Ma, "GaussianShader: 3D Gaussian splatting with shading functions for reflective surfaces," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[37] Z. Liang, Q. Zhang, Y. Feng, Y. Shan, and K. Jia, "GS-IR: 3D Gaussian splatting for inverse rendering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[38] A. Hamdi, L. Melas-Kyriazi, G. Qian, J. Mai, R. Liu, C. Vondrick, B. Ghanem, and A. Vedaldi, "GES: Generalized exponential splatting for efficient radiance field rendering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[39] Z.-X. Zou, Z. Yu, Y.-C. Guo, Y. Li, D. Liang, Y.-P. Cao, and S.-H. Zhang, "Triplane meets Gaussian splatting: Fast and generalizable single-view 3D reconstruction with transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 10 324–10 335.

[40] S. Niedermayr, J. Stumpfegger, and R. Westermann, "Compressed 3D Gaussian splatting for accelerated novel view synthesis," *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2024.

[41] Z. Fan, K. Wang, K. Wen, Z. Zhu, D. Xu, and Z. Wang, "LightGaussian: Unbounded 3D Gaussian compression with 15x reduction and 200+ FPS," 2023.

[42] J. C. Lee, D. Rho, X. Sun, J. H. Ko, and E. Park, "Compact 3D Gaussian representation for radiance field," *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2024.

[43] Z. Yu, A. Chen, B. Huang, T. Sattler, and A. Geiger, "Mip-splatting: Alias-free 3D Gaussian splatting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[44] Z.-H. Lin, W.-C. Ma, H.-Y. Hsu, Y.-C. F. Wang, and S. Wang, "NeurMiPs: Neural mixture of planar experts for view synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[45] T. Monnier, J. Austin, A. Kanazawa, A. A. Efros, and M. Aubry, "Differentiable Blocks World: Qualitative 3D Decomposition by Rendering Primitives," in *Advances in Neural Information Processing Systems*, 2023.

[46] G. Kopanas, T. Leimkühler, G. Rainer, C. Jambon, and G. Drettakis, "Neural point catacaustics for novel-view synthesis of reflections," *ACM Transactions on Graphics (TOG)*, vol. 41, no. 6, pp. 1–15, 2022.

[47] G. Kopanas, J. Philip, T. Leimkühler, and G. Drettakis, "Point-based neural rendering with per-view optimization," in *Computer Graphics Forum*, vol. 40, no. 4. Wiley Online Library, 2021, pp. 29–43.

[48] W. Yifan, F. Serena, S. Wu, C. Öztireli, and O. Sorkine-Hornung, "Differentiable surface splatting for point-based geometry processing," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 6, pp. 1–14, 2019.

[49] Y. Zheng, W. Yifan, G. Wetzstein, M. J. Black, and O. Hilliges, "PointAvatar: Deformable point-based head avatars from videos," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2023, pp. 21 057–21 067.

[50] C. Lassner and M. Zollhofer, "Pulsar: Efficient sphere-based neural rendering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1440–1449.

[51] L. Ke, M. Ye, M. Danelljan, Y. Liu, Y.-W. Tai, C.-K. Tang, and F. Yu, "Segment anything in high quality," *Advances in Neural Information Processing Systems*, 2023.

[52] S. Zhi, T. Laidlow, S. Leutenegger, and A. J. Davison, "In-place scene labelling and understanding with implicit scene representation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 838–15 847.

[53] J. Kerr, C. M. Kim, K. Goldberg, A. Kanazawa, and M. Tancik, "LeRF: Language embedded radiance fields," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 19 729–19 739.

[54] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International Conference on Machine Learning*. PmLR, 2021, pp. 8748–8763.

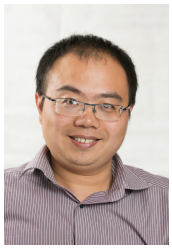
[55] M. Qin, W. Li, J. Zhou, H. Wang, and H. Pfister, "LangSplat: 3D language Gaussian splatting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20 051–20 060.
[56] J.-C. Shi, M. Wang, H.-B. Duan, and S.-H. Guan, "Language embedded 3D Gaussians for open-vocabulary scene understanding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 5333–5343.
[57] P. Hedman, J. Philip, T. Price, J.-M. Frahm, G. Drettakis, and G. Brostow, "Deep blending for free-viewpoint image-based rendering," *ACM Transactions on Graphics (ToG)*, vol. 37, no. 6, pp. 1–15, 2018.
[58] A. Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun, "Tanks and temples: Benchmarking large-scale scene reconstruction," *ACM Transactions on Graphics (ToG)*, vol. 36, no. 4, pp. 1–13, 2017.
[59] Y. Lipman, D. Cohen-Or, D. Levin, and H. Tal-Ezer, "Parameterization-free projection for geometry reconstruction," *ACM Transactions on Graphics (TOG)*, vol. 26, no. 3, pp. 22–es, 2007.
[60] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.

img/zlx.pdf

Ling-Xiao Zhang is currently an engineer with the Institute of Computing Technology, Chinese Academy of Sciences. His research interests include computer graphics and geometric processing.

img/jiangchenbo.pdf

Chenbo Jiang was a research intern mentored by Prof. Lin Gao at the Institute of Computing Technology, Chinese Academy of Sciences. He is also an incoming PhD student at McGill University and the Mila-Quebec AI Institute. His research primarily focuses on the intersection of machine learning and computer graphics.



Yu-Kun Lai received the bachelor's and PhD degrees in computer science from Tsinghua University, in 2003 and 2008, respectively. He is currently a professor with the School of Computer Science and Informatics, Cardiff University. His research interests include computer graphics, geometry processing, image processing, and computer vision. He is on the editorial boards of *IEEE Transactions on Visualization and Computer Graphics*, *Computers & Graphics* and *The Visual Computer*.

Lin Gao received the PhD degree from Tsinghua University. He is currently a professor with the Institute of Computing Technology, Chinese Academy of Sciences and the University of Chinese Academy of Sciences. He has been awarded the Newton Advanced Fellowship from the Royal Society and the Asia Graphics Association young Researcher award. His research interests include computer graphics and geometric processing.