HO-Gaussian: Hybrid Optimization of 3D Gaussian Splatting for Urban Scenes

Zhuopeng Li[1] , Yilin Zhang[1], Chenming Wu[2], Jianke Zhu[1⋆], Liangjun Zhang[2]

[1] Zhejiang University {lizhuopeng,jkzhu}@zju.edu.cn
[2] Baidu Research

Fig. 1: Illustration of 3D Gaussian Splatting (3DGS) [22] and HO-Gaussian (Ours). Compared with 3DGS initialized by SfM points, our method has richer Gaussian geometric information in low-texture, sky and distant areas, and shows significant improvement in the task of synthesizing novel views.

Abstract. The rapid growth of 3D Gaussian Splatting (3DGS) has revolutionized neural rendering, enabling real-time production of high-quality renderings. However, the previous 3DGS-based methods have limitations in urban scenes due to reliance on initial Structure-from-Motion (SfM) points and difficulties in rendering distant, sky and low-texture areas. To overcome these challenges, we propose a hybrid optimization method named *HO-Gaussian*, which combines a grid-based volume with the 3DGS pipeline. HO-Gaussian eliminates the dependency on SfM point initialization, allowing for rendering of urban scenes, and incorporates the Point densification to enhance rendering quality in problematic regions during training. Furthermore, we introduce Gaussian Direction Encoding as an alternative for spherical harmonics in the rendering pipeline, which enables view-dependent color representation. To account

⋆ corresponding authors

for multi-camera systems, we introduce neural warping to enhance object consistency across different cameras. Experimental results on widely used autonomous driving datasets demonstrate that HO-Gaussian achieves photo-realistic rendering in real-time on multi-camera urban datasets.

Keywords: Novel View Synthesis · Urban Scenes · Gaussian Splatting

1 Introduction

Urban scene simulation is of great significance in autonomous driving, which usually generates diverse and large-scale data for training and evaluating various models in autonomous driving, such as occupancy prediction [52, 57], segmentation [5, 13, 14], and object detection [6, 24, 44, 57].

Despite the conventional methods using image-processing techniques to render images in urban scene simulation [2,25], recent neural radiance fields (NeRF)-based approaches [30] have demonstrated remarkable capabilities in facilitating realistic reconstructions and synthesizing novel views from multi-view images. For instance, Block-NeRF [42] decomposes a large scene into blocks and trains NeRFs for each block. DNMP [27] proposes a deformable neural mesh primitives to represent urban scenes by combining mesh-based rendering and neural representations, where each voxel is accurately initialized as a deformable neural mesh primitive. To address the challenge of capturing precise geometry in urban-level scenes, most methods [34,36,46,52] promote the learning of geometry by either incorporating LiDAR observations to supervise NeRF or using LiDAR point clouds as the initialization of the scene [34, 51, 56]. Nevertheless, synthetic views generated by these methods often exhibit artifacts and lack fine details in texture. Furthermore, the real-time rendering capability of these methods is constrained by the NeRF representation, where volume rendering requires a large number of ray samples per pixel to inference the MLP network.

In contrast to NeRF, a more recent technique called 3D Gaussian Splatting (3DGS) [22] offers a notable alternative. 3DGS utilizes explicit 3D Gaussians to represent the scene and has made significant advancements in efficiently rendering novel views. Moreover, 3DGS has been successfully integrated into various platforms and traditional rendering pipelines, including lightweight web rendering and game engines, such as Unity [16] and Unreal [21]. However, the effectiveness of 3DGS heavily relies on well-initialized point clouds acquired from either SfM (Structure-from-Motion) [39, 47] or SLAM (Simultaneous localization and mapping) [9, 11]. While random initialization proves to be effective for object-centric scenes in 3DGS, it becomes inadequate in dealing with unbounded urban environments with limited views. SfM and SLAM systems often treat regions with uniform texture as outliers, resulting in incomplete 3D reconstructions. Additionally, regular LiDAR devices have limited range in providing point cloud data. These factors make it difficult for 3DGS to be optimized effectively, resulting in subpar rendering effects. Moreover, the explicit 3D Gaussian representation occupies large amount of disk space.

To tackle the above challenges in synthesizing urban scenes by 3DGS in real-time, this paper presents HO-Gaussian, a point-free representation method for multi-camera urban scenes. To facilitate end-to-end optimization, we suggest a hybrid scheme to optimize both the volume and Gaussian. Our approach presents point densification to circumvent the reliance on initialization points inherent to 3DGS-based methods. Specifically, we improve rendering quality by introducing a volume to optimize the position of Gaussians, facilitating the learning of geometric information in sky, distant and low-texture areas. To enhance the capabilities of scene representation in the rendering pipeline, we introduce Gaussian positional and directional encoding technique, which effectively models spherical harmonics and reduces the disk space requirement of Gaussian splatting method. Notably, storing view-dependent spherical harmonic functions even within a range of just a few hundred meters requires gigabytes of disk space in large-scale urban scenes. To render urban scenes with multi-camera, we introduce neural warping scheme that ensures the consistent rendering results across multiple cameras through mutual learning between different camera perspectives. This approach reduces the risk of overfitting to the specific viewpoints, enhancing the generalization capability of the rendering pipeline. The contributions of this paper can be summarized as follows.

- We propose a novel pipeline to learn the positions of Gaussians from a grid-based volume, which allows for better optimization of geometric information and rendering results in urban scenes.
- We present the Gaussian positional and directional encoding that improve the scene representation of rendering pipeline, addressing the excessive disk space usage associated with spherical harmonic functions in 3DGS-based methods. Additionally, the presented neural warping scheme enables our approach to efficiently synthesize novel views across various cameras.
- Extensive experiments on widely-used autonomous driving datasets demonstrate the effectiveness of our proposed method compared to either the previous NeRF-based methods or 3DGS-based approaches.

2 Related Work

2.1 Scene Representation

3D data can be represented in various forms, such as point clouds, meshes, and voxels. These representations are typically obtained through techniques like Structure from Motion (SfM) [10,12,39,41,47], Multi-View Stereo (MVS) [17,20, 40,55], or scanning with LiDAR devices (SLAM systems) [4,9,11,32,33]. SfM and SLAM systems often treat regions with uniform textures as outliers, resulting in incomplete 3D reconstructions. MVS methods, on the other hand, strive to estimate dense 3D geometry while focusing on static scenes and highly overlapping image sets with known poses. However, current neural rendering technologies such as NeRF [28,29,42,46,50,53], Point-based Rendering [1,18,23,26,37,38,58]

and 3DGS [7, 8, 22, 43, 58] rely on accurate geometric representations to provide clues. To enhance the quality of novel view synthesis in urban scenes, we propose to learn from a grid-based volume and optimize the geometric information through a Gaussian pipeline. This enables us to supplement the geometry information of distant and low-texture areas.

2.2 NeRF for Urban Scenes

The implicit neural representation proposed by NeRF has achieved promising results in various scenes, and many researchers have applied it to large-scale outdoor scenes. Block-NeRF [42] first proposed a NeRF composed of multiple regional blocks, using appearance, and exposure embedding to model data in different time periods. SUDS [46] decomposes the scene into three separate hash table data structures to efficiently encode static, dynamic and far-field radiance fields. LocalRF [28] presents an innovative approach by implementing a progressive strategy to dynamically assign local radiance fields. Furthermore, some methods [15, 27, 48, 52, 54] have achieved promising results by applying NeRF to driving scenarios. However, NeRF-based methods often suffer from artifacts and lack of realistic textures in outdoor scenes. Due to the limited representation capabilities and inefficient rendering pipeline, it is difficult for them to be widely used in the real-world large scale applications.

2.3 Point-based Rendering and 3D Gaussian Splatting

Point-based rendering methods are widely used to efficiently render unstructured geometric samples. NPBG [1] utilizes neural textures to encode local geometric shapes and appearances, enabling high-quality synthesis of novel views from point clouds. Building upon this, ADOP [37] proposes a point-based differentiable neural rendering pipeline that leverages single-pixel rasterization to refine all input parameters. Recently, 3D Gaussian Splatting [22] combines the concept of point-based rendering [1] and splatting [58] techniques for rendering, which employs explicit 3D Gaussian as the representation of the scene. It achieves real-time rendering while maintaining sufficient quality. The explicit representation without a neural network brings fast rendering speed, however, it makes the Gaussian method require a lot of memory and storage resources due to storing Gaussian-related properties, such as covariance matrices and higher-order spherical harmonics. In our work, we employ Gaussian directional encoding as an alternative of spherical harmonics that consume huge disk space.

3 Method

Our proposed HO-Gaussian method tackles the challenges of novel view rendering for urban scenes captured by multiple cameras. A key contribution is an end-to-end rendering pipeline grounded in Gaussian splatting, which mitigates the dependency on initial SfM points by employing a grid-based volume. The

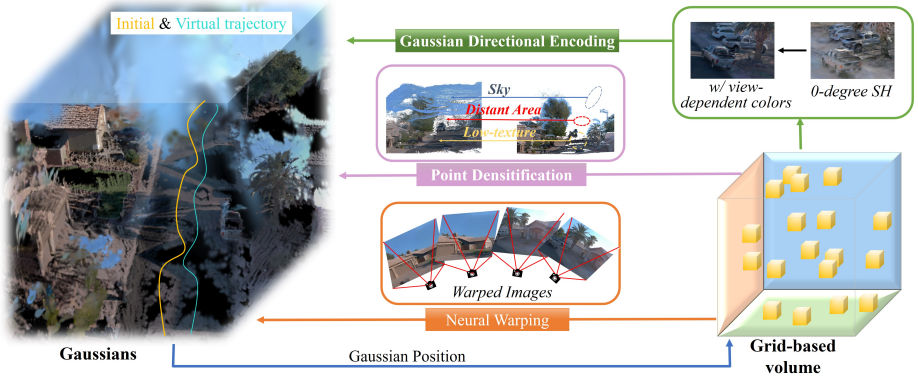


Fig. 2: Pipeline. The hybrid optimization starts from a grid-based volume, creating a set of Gaussian points, with the grid-based volume and Gaussian pipeline iteratively optimized. Subsequently, at regular intervals, point densification provides new positions to the Gaussian pipeline to populate problematic regions. Here, view-dependent color is encoded by the Gaussian Directional Encoding, replacing spherical harmonics. Finally, we supply virtual viewpoints to the Gaussian pipeline through neural warping, enhancing consistent appearance and geometry for multi-camera scenes.

rendering pipeline facilitates hybrid optimization, bolstering the representation capacity of scene while enriching its geometric information. Crucially, it circumvents the drawbacks of Gaussian splatting in large-scale urban scenarios, such as redundant disk usage, through the design of a grid-based volume representation and Gaussian directional encoding. To enhance the adaptability of rendering pipeline to multi-camera urban scenes and mitigate the risk of overfitting to specific viewpoints, we introduce a novel neural warping module. Moreover, HO-Gaussian addresses the challenges inherent to multi-camera urban scenes. It achieves real-time rendering performance while preserving photo-realistic texture details.

3.1 Preliminary

The 3D Gaussian Splatting method [22] introduces an explicit representation for 3D scenes that leverages the diverse properties of Gaussians to capture the scene geometry. The approach commences with a set of SfM points and subsequently employs a collection of anisotropic 3D Gaussian models to represent the scene. These Gaussian models inherit various properties inherent to volumetric representations while achieving efficient rendering through a tile-based rasterization algorithm. The 3D Gaussian can be mathematically formulated as:

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

where x denotes an arbitrary position within the 3D scene. Each 3D Gaussian splat is assigned a position (mean) μ, and $\mathbf{\Sigma}$ represents the covariance matrix of

the 3D Gaussian. To ensure the positive semi-definiteness of $\mathbf{\Sigma}$, it is represented as the product of a scaling matrix $\mathbf{S}$ and a rotation matrix $\mathbf{R}$:

$$\mathbf{\Sigma} = \mathbf{RSS^T R^T}. \tag{2}$$

The Gaussian splatting method leverages splatting techniques [58] to project the 3D Gaussians onto 2D image planes for rendering purposes. Given the viewing transformation $\mathbf{W}$ and the Jacobian of the affine approximation for the projective transformation $\mathbf{J}$ [59], the covariance matrix $\mathbf{\Sigma'}$ in camera coordinates can be computed as $\mathbf{\Sigma'} = \mathbf{JW\Sigma W^T J^T}$.

Each 3D Gaussian consists of its position, color represented by spherical harmonics (SH), opacity, rotation, and scaling. For a given pixel, it is calculated via multiplying the covariance $\mathbf{\Sigma}$ by the learned opacity α for each point, as shown in Eqn. 1. The color blending of N ordered points with overlapping pixels is determined by:

$$\mathbf{C} = \sum_{i \in N} \mathbf{c}_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j), \tag{3}$$

where $\mathbf{c}_i$ denotes the color of a point, and α_i is its opacity.

3.2 Point densification based on Hybrid Volume

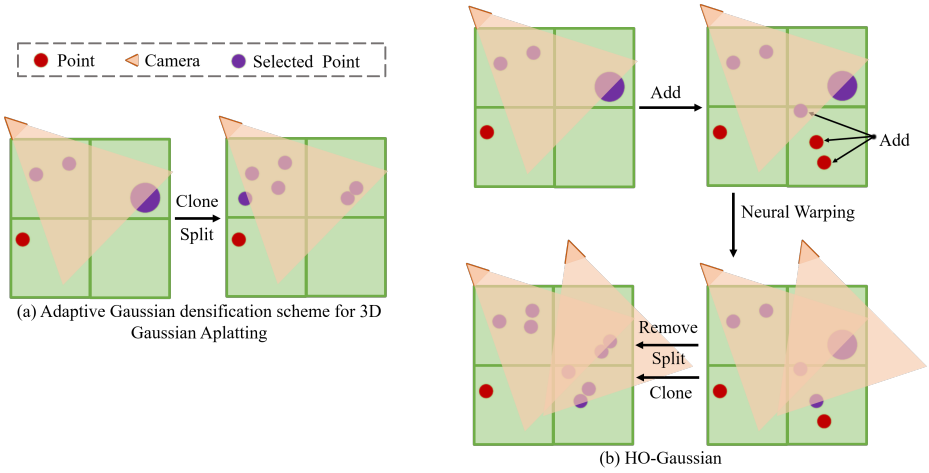
The original 3DGS method adaptively controls the number of Gaussian points through clone and split operations. It allows to convert an initial sparse set into a denser one that better represents the scene geometry by filling in empty areas and removing over-reconstructed regions. However, this approach relies on an accurate initialization of the sparse points. Moreover, the original 3DGS densification occurs only in the vicinity of the initial sparse points. For urban scenes, the initial sparse points optimized by SfM techniques often exhibit holes in regions such as the sky, distant areas, or low-texture areas, resulting in blank areas that cannot be rendered accurately. To address this limitation, we introduce a grid-based volumetric representation to provide new positions for the Gaussian pipeline, as illustrated in Fig. 3.

Point densification The grid-based volume is a continuous function f mapping location $\mathbf{x}$ and direction $\mathbf{d}$ to a volume density $\sigma \in [0, \infty)$ and color $\mathbf{c} \in [0, 1]^3$. The sigma σ_θ and color $\mathbf{c}_\theta$ indicate the density and color prediction of radiance field using MLPs f_θ, parameterized by θ:

$$\mathbf{c}_\theta, \sigma_\theta = f_\theta(\mathbf{x}, \mathbf{d}). \tag{4}$$

Given a ray $\mathbf{r}$ belonging to the ray set $\mathcal{R}$, we can infer the predicted Gaussian position from grid-based volume by $\mathbf{x} = \mathbf{o} + \hat{\mathbf{D}}_\theta(\mathbf{r})\mathbf{d_r}$, where the 3D position $\mathbf{x} \in \mathbb{R}^3$ and direction $\mathbf{d} \in \mathbb{S}^3$, $\hat{\mathbf{D}}_\theta(\mathbf{r})$ can be approximated by integrating the sampled particles along the ray direction $\mathbf{d}$. $Q(u)$ denotes the accumulated transmittance along the ray

$$\hat{\mathbf{D}}_\theta(\mathbf{r}) = \int_{u_n}^{u_f} Q(u) \sigma_\theta(\mathbf{r}(u)) u du, \tag{5}$$



(a) Adaptive Gaussian densification scheme for 3D Gaussian Aplatting

(b) HO-Gaussian

Fig. 3: Comparing densification strategies of 3DGS and our HO-Gaussian. The cloning and splitting strategies of 3DGS can effectively optimize the Gaussian distribution near the initial SfM points. However, they fail to work in low-texture or distant areas where the positions of initial points are missing. HO-Gaussian is capable of learning and optimizing Gaussian distributions beyond the initial points. First, Point densification supplies the Gaussian pipeline with missing points within viewpoints, preventing the projection of empty 2D splats. Subsequently, Neural Warping introduces virtual viewpoints, thereby covering more occluded points. Finally, clone and split operations are employed to fine-tune the positions of inaccurate splats, and Gaussian splats with opacity values α below a threshold ϵ_α are removed.

$$Q(u) = \exp\left(-\int_{u_n}^{u} \sigma_\theta(\mathbf{r}(s))ds\right), \tag{6}$$

where u_n and u_f is the predefined near and far planes for rendering, respectively. Combined with Eqn. 1, the Gaussian can be expressed as below:

$$G(x) = e^{-\frac{1}{2}(\mathbf{o}+\hat{\mathbf{D}}_\theta(\mathbf{r})\mathbf{d_r}-\mu)^T\mathbf{\Sigma}^{-1}(\mathbf{o}+\hat{\mathbf{D}}_\theta(\mathbf{r})\mathbf{d_r}-\mu)}. \tag{7}$$

For stability, we warm up the calculations in the grid-based volume. Subsequently, we select the candidate locations $\mathbf{x}$ within a bounded range and determine whether to add them to the Gaussian pipeline based on whether their density meets a predefined threshold τ. For locations satisfying with the density criterion, we introduce a new Gaussian splat into the Gaussian pipeline, along with its associated Gaussian parameters, such as the corresponding covariance matrix.

The introduced grid-based volume representation facilitates point densification, supplying the Gaussian pipeline with missing points within viewpoints, thereby preventing the projection of empty 2D splats, as illustrated in Fig. 3(b). Compared to LiDAR or SfM points, the points generated by this densification

strategy lack precision. To address this limitation, we introduce a hybrid optimization strategy, as described in Section 3.5. Through clone and split operations within the Gaussian pipeline, we fine-tune the positions of inaccurate splats and remove Gaussian splats with opacity values α below a threshold ϵ_α.

Gaussian Positional Encoding In urban scene, the rich details of large-scale scenes require a significant amount of Gaussian optimization. However, the limitations of video memory for 3D Gaussian result in an uneven distribution of a limited number of Gaussians in space, making it challenging to render the entire scene. Simply applying 3DGS to large-scale scenes can lead to low-quality reconstructions or insufficient memory. Therefore, we introduce Gaussian Positional Encoding, inspired by Kalman filter [19] and Mip-NeRF 360 [3]. By compressing the urban scene within a certain range, the Gaussian position encoding allows expressing distant areas with smaller or fewer Gaussians in the nearby regions, achieving high-quality scene reconstruction without increasing memory usage. By leveraging Gaussian positional encoding, we distort the unbounded scene domain into a finite sphere, as follows:

$$\mathbf{x}_{encoding} = \begin{cases} \mathbf{x} & \text{if } \|\mathbf{x}\|_2 \leq 1 \\ \left(2 - \frac{1}{\|\mathbf{x}\|_2}\right) \frac{\mathbf{x}}{\|\mathbf{x}\|_2} & \text{if } \|\mathbf{x}\|_2 > 1 \end{cases} \tag{8}$$

3.3 The Gaussian Directional Encoding

Urban scenes encompass a diverse array of elements, including various traffic vehicles, buildings, and other structures, coexisting under different lighting conditions. The method based on 3DGS requires a large number of Gaussians to model various objects. The rapid increase in the number of Gaussians and the use of high-order SH bring disk Space explodes. Therefore, we utilize view-dependent color c_θ instead of higher-order SH, in which only 0-degree SH is used during training. Specifically, we employ Gaussians to model 3D shapes, while a grid-based neural network generates view-dependent colors as Gaussian directional encodings. Our model adopts the Gaussian representation as: $\{(\mathcal{N}(\mathbf{x}_{encoding}(i), \mathbf{\Sigma}_i), \alpha_i, c_i)\}_{i=1}^n$, where view-dependent color c_i is obtained from the MLPs network f from Eqn. 4.

3.4 Neural Warping

Due to the limited overlapping area in multi-camera systems, it is hard to align the colors of objects at different viewpoints, which can make scenes difficult to be optimized. To this end, we introduce a neural warping strategy that simulates images from different virtual viewpoints through grid-based volume, covering various view-dependent colors and virtual positions. This approach helps the model better adapt to real-world scenarios and reduces the risk of overfitting to the specific viewpoints.

By employing a grid-based volume, we generate multiple virtual poses and positions by perturbing the existing pose. This allows us to obtain the warped

colors from the volume to assist in fitting objects in multi-cameras through the Gaussian splatting pipeline. Perturbing the pose involves rotating around the current position x_p with random angles within the range of $[-10, 10]$. The existing image point captured p_i under camera pose $[R_e, T_e]$ is distorted to image point p_v with the virtual pose $[R_v, T_v]$. We define the generation of this neural warping as follows

$$p_v = K(R_v(p_i) + T_v), \tag{9}$$

where K is the intrinsic matrix. According to Eqn. 4, we can get the color of the warped image $\mathbf{c_{v}}_\theta = f_\theta(\mathbf{x_p}, \mathbf{d})$. The warped image provides the Gaussian pipeline with more viewpoints to learn the consistent appearance and geometry of the scene.

3.5 Hybrid Optimization

In summary, we optimize the HO-Gaussian pipeline through a hybrid scheme, including Gaussian and volume optimization.

For N Gaussians and their attributes (i.e., position $\mathbf{x}_{encoding}(i)$, opacity α_i, color $c_{\theta i}$, covariance matrix $\mathbf{\Sigma}_i$), we train the entire model by sampling from ground truth and warped images. We optimize the learnable attribute parameters using a combination of L_1 loss and SSIM loss between the ground truth $c(\mathbf{r})$ and rendered images C as below

$$\mathcal{L}_{\mathrm{g}} = (1-\lambda)\mathcal{L}_1(C, c(\mathbf{r})) + \lambda\mathcal{L}_{\mathrm{SSIM}}(C, c(\mathbf{r})). \tag{10}$$

The grid-based volume consists of two MLPs with parameters, predicting density and color respectively. Optimize by minimizing the mean square error ground truth $c(\mathbf{r})$ and rendered images $\mathbf{c}_\theta(\mathbf{r})$.

$$\mathcal{L}_{\mathrm{MSE}}\left(\theta, \mathcal{R}_{\mathbf{r}}\right) = \sum_{\mathbf{r}\in\mathcal{R}_{\mathbf{r}}} \left\|\mathbf{c}_\theta(\mathbf{r}) - \mathbf{c}(\mathbf{r})\right\|^2. \tag{11}$$

To jointly optimize all parameters in our proposed Gaussian splatting pipeline, we make use of a combination of multiple losses:

$$\mathcal{L}_{\mathrm{total}} = \mathcal{L}_{\mathrm{g}} + \lambda_1\mathcal{L}_{\mathrm{MSE}}\left(\theta, \mathcal{R}_{\mathbf{r}}\right). \tag{12}$$

It is worth mentioning that the Gaussian splatting pipeline undergoes the gradient optimization for Gaussian directional encoding, we introduce the supervised learning with hyperparameters λ_1 to enhance the accuracy of Gaussian points from the grid-based volume.

4 Experimental

4.1 Implementation Details

To evaluate HO-Gaussian, we conducted experiments on an NVIDIA Tesla V100 32GB GPU. HO-Gaussian was rigorously tested on two urban scene datasets of

Fig. 4: Comparative results of novel view synthesis on Argoverse datasets. Please zoom in to view the detailed results.

12 sequences. Through both qualitative and quantitative analysis, we demonstrate convincing results, showing that our method achieves the promising performance and efficiency compared to other methods while mitigating the disk space issues caused by urban scenes. Consistent with the method described in 3DGS, our model retains all hyperparameters of 3DGS and the trained model over 30K iterations in all scenarios. The neural field for view-dependent colors uses a hash grid, followed by an MLP network with a layer number of 2 and 64 channels. The density network consists of an MLP network with a layer number of 1 and 64 channels. The values of the hyperparameters λ and λ_1 are set to 0.2 and 0.1, respectively.

4.2 Datasets

In our experiments, we evaluated our method on large-scale urban datasets, namely Waymo and Argoverse. We conducted evaluations across eight scenarios from the Waymo dataset and four scenarios from the Argoverse dataset. The Waymo dataset comprises three cameras with a resolution of 1920×1280. We tested our method on both daytime and nighttime scenes from Waymo. As for the Argoverse dataset, it features seven cameras with a resolution of 1920×1200. To thoroughly assess the representation capability of model and mitigate the potential overfitting issues arising from small scenes, we selectively perform evaluation on scenes containing more than 550 frames of image data. Following [1, 26, 28, 42, 45], we select every ten frames for testing while choosing the remaining ones for training.

4.3 Evaluation Results

To demonstrate the effectiveness of the proposed HO-Gaussian method, we compared it with NeRF-based methods for urban scenes that do not require SfM or LiDAR points, including instant NGP [31], MERF [35], Block-NeRF [42] and LocalRF [28]. These methods have shown promising results in synthesizing urban scenes. In order to intuitively reflect the effectiveness of our method, we also compared it against methods that require SfM points or LiDAR information, namely S-NeRF [49] and EmerNeRF [52], respectively. Similar to the evaluation protocols used in these methods, we employed three widely used metrics for evaluation: Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM), and Learned Perceptual Image Patch Similarity (LPIPS).

Comparing Methods without SfM (LiDAR) Point Instant NGP [31] employs hash coding for fast radiance field reconstruction. However, it is unsuitable to reconstruct the large-scale scenes. In this regard, MERF [35] proposes a novel contraction function to optimize large-scale scenes by distorting the unbounded scene domain into finite space. Significant improvements are observed in both Waymo and Argoverse datasets. Block-NeRF [42] and LocalRF [28] focus on handling outdoor large-scale scene tasks by dividing urban scenes into blocks and employing progressive rendering. Among them, Block-NeRF has poor rendering results due to the poor strategy of dividing multi-camera areas (failed on the Argoverse dataset). LocalRF achieves excellent rendering results in urban scenes, however, it is limited by the scene division strategy in the Waymo dataset. Furthermore, scene division-based NeRF methods often suffer from disk space redundancy, as shown in both Table 1 and Table 2.

Comparing Methods with SfM (LiDAR) Point In urban-level scenes, most methods rely on using LiDAR or SfM points as scene initialization to reduce unnecessary computations in the radiance field. Alternatively, supervised learning with LiDAR data is used to capture better scene geometry. For example, methods like [27, 34, 36, 46, 52] are commonly employed. In our comparison, we selected two well-performing methods, namely S-NeRF [49] and EmerNeRF [52], to assess their performance. Since these two methods require a huge amount of

Table 1: Quantitative evaluation of novel view synthesis on the Waymo and Argoverse dataset. "*" denotes half resolution.

Method	Input	Waymo	Argoverse
		PSNR↑ / SSIM↑ / LPIPS↓	PSNR↑ / SSIM↑ / LPIPS↓
NGP [31]	RGB	23.88 / 0.7369 / 0.5621	25.87 / 0.8143 / 0.4364
MERF [35]	RGB	26.28 / 0.7827 / 0.3820	27.52 / 0.8473 / 0.3204
Block-NeRF [42]	RGB	23.60 / 0.7454 / 0.5031	- / - / -
3DGS [22]	RGB	18.31 / 0.6371 / 0.6001	21.50 / 0.7979 / 0.4853
Ours	RGB	**28.03** / **0.8364** / **0.3282**	30.98 / **0.9043** / **0.2287**
Ours*	RGB	**28.62** / **0.8497** / **0.2561**	**31.52** / **0.9081** / **0.1548**
S-NeRF* [49]	RGB+SfM	24.07 / 0.6835 / 0.5215	24.36 / 0.7080 / 0.5511
EmerNeRF* [52]	RGB+LiDAR	**28.62** / 0.8053 / 0.3147	30.14 / 0.8347 / 0.3210
LocalRF [28]	RGB+Depth	23.16 / 0.8002 / 0.4201	**31.79** / 0.8837 / 0.2976
3DGS [22]	RGB+SfM	24.90 / 0.8117 / 0.3695	27.83 / 0.8795 / 0.2822

Table 2: Ablation study.

Method	PSNR↑ / SSIM↑ / LPIPS ↓
SfM points + 3DGS	27.83 / 0.8795 / 0.2822
LiDAR points + 3DGS	28.40 / 0.8754 / 0.2856
NeRF points + 3DGS	28.61 / 0.8684 / 0.2883
+ w/ Gaussian directional and position encoding	28.42 / 0.8758 / 0.3026
+ w/ Point densification	30.58 / 0.8954 / 0.2372
+ w/ Neural warping(Ours)	30.98 / 0.9043 / 0.2287

video memory for training, the resolution of the image has to be reduced to half for comparison. Among them, EmerNeRF performs scene decomposition through self-supervision and obtains promising rendering results in various scenes. However, the lack of texture details can be seen from the LPIPS metric, and the rendering speed is limited, as shown in Table 1 and Table 2.

In this paper, we advocate for the rendering of large-scale urban scenes without SfM (LiDAR) point initialization or supervision, optimizing geometric information through HO-Gaussian and obtaining high-quality rendered images. Among these methods, the Gaussian representation without point cloud initialization performs the worst, as it is challenging for 3DGS-based methods to optimize complex road conditions in large-scale urban scenes without geometric cues. By taking advantage of grid volume learning and Gaussian positions optimization, our proposed approach does not require point cloud initialization and achieves promising rendering results solely based on supervised information from images. As demonstrated in Table 1, our proposed method significantly outperforms both NeRF-based and 3DGS-based methods in almost all evaluation metrics.

Table 3: Comparison of model size, training time and rendering speed.

Method	Input	Model size	Training time	FPS
S-NeRF* [49]	RGB+SfM	103MB	15h	0.01
EmerNeRF* [52]	RGB+LiDAR	431MB	57m	0.45
LocalRF* [28]	RGB+Depth	3762MB	14h	0.11
3DGS* [22]	RGB+SfM	557MB	31m	87
Ours*	RGB	123MB	76m	71

4.4 Ablation study

In this section, we conduct both qualitative and quantitative experiments using the Argoverse dataset to evaluate each component of our method. We will use Gaussian initialized by SfM points and LiDAR points as the baseline, as shown in Table 2. To demonstrate the advantages of our proposed end-to-end hybrid optimization, we also compare using point clouds generated by NeRF as the initialization of the Gaussian method. It can be seen from the Table 2 the Gaussian method initialized by SfM points and LiDAR points performs poorly due to the lack of geometric information in low-texture areas and distance areas, resulting in blurred rendering results, as shown in Fig. 1 and Fig. 4. The point cloud generated by NeRF can make up for the shortcomings of low-texture areas. Due to the lack of accuracy, the effect is not much improved.

Compared with the original 3DGS method, we first introduced the view-dependent color based on grid volume to replace the spherical harmonic function, called w/ Gaussian directional encoding. The model size was significantly reduced from 557MB to 123MB, decreasing the disk space usage by 352.8%. Then, we introduced Gaussian positional encoding, and the results were slightly improved. Through the Point densification, we add Gaussian positions to low-texture areas and distant areas of the urban scene, and fine-tune the Gaussian positions by gradient descent of the Gaussian pipeline. As shown in Fig. 4 and Table 2, the rendering results of the Gaussian point optimization region have been significantly improved. Finally, we introduce neural warping to provide more virtual viewpoints to the Gaussian pipeline to improve rendering quality.

4.5 Complexity Analysis

Model Size and Rendering Speed Rendering speed and model size are crucial for the task of novel view synthesis of urban scenes, since it directly affects interaction efficiency and storage. We evaluate various methods that perform well in large-scale urban scenarios, as shown in Table 1 and Table 3. Based on the design of Gaussian directional encoding, our method achieves good rendering quality with smaller disk space and real-time rendering speed.

Discussion about texture quality and scene geometry To demonstrate the effectiveness of geometric optimization in our method, we present geometric

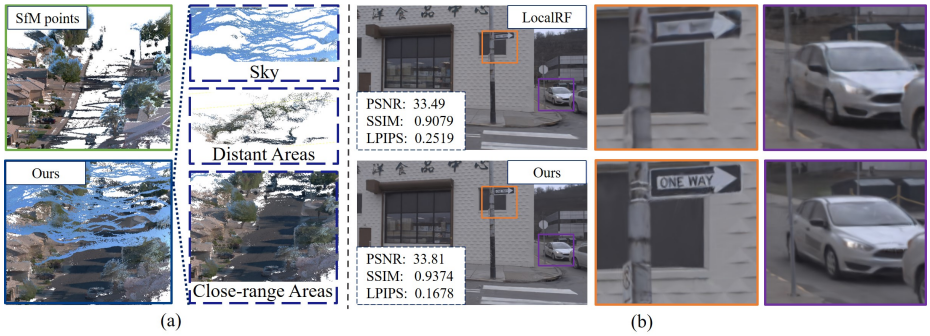


Fig. 5: Visualization of scene geometry(a) and texture quality synthesized by LocalRF and our method(b). Please zoom in to view detailed results.

predictions in a Waymo scene. Compared with SfM points, the geometry generated by our method has greater advantages in low-texture and distance areas. More results are included in the supplementary material. This section also aims to discuss the reliability of the novel view synthesis evaluation metric. The PSNR score of HO-Gaussian is slightly lower than LocalRF in Table 1. However, the error in the LPIPS metric is reduced by 30.1%. This difference occurs because PSNR focuses on pixel-level details through mean square error analysis, while the LPIPS metric measures overall image similarity. In Fig. 5, HO-Gaussian exhibits finer texture details compared to LocalRF.

5 Conclusion

For large-scale urban scenes, this paper proposes a hybrid optimization method that fuses grid-based volume with 3DGS pipeline. HO-Gaussian eliminates the dependency on SfM point initialization and enhances the rendering quality of problematic areas in urban scenes by adding Gaussian points. We introduce Gaussian positional encoding and directional encoding to improve the representation capability of the scene while reducing storage requirement. The HO-Gaussian pipeline is provided with more viewpoints by introducing neural warping to improve the consistent appearance and geometry of the scene. Extensive experiments on several autonomous driving datasets demonstrate its ability to achieve real-time, realistic rendering in multi-camera urban scenes.

Acknowledgements

This work is supported by the National Key Research and Development Program of China (No. 2023YFF0905104) and also by National Natural Science Foundation of China under Grants (62376244).

References

1. Aliev, K.A., Sevastopolsky, A., Kolos, M., Ulyanov, D., Lempitsky, V.: Neural point-based graphics. In: Proceedings of the European conference on computer vision. pp. 696–712. Springer (2020)
2. Amini, A., Wang, T.H., Gilitschenski, I., Schwarting, W., Liu, Z., Han, S., Karaman, S., Rus, D.: Vista 2.0: An open, data-driven simulator for multimodal sensing and policy learning for autonomous vehicles. In: 2022 International Conference on Robotics and Automation (ICRA). pp. 2419–2426. IEEE (2022)
3. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5470–5479 (2022)
4. Campos, C., Elvira, R., Rodríguez, J.J.G., Montiel, J.M., Tardós, J.D.: Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. IEEE Transactions on Robotics **37**(6), 1874–1890 (2021)
5. Cen, J., Zhou, Z., Fang, J., yang, c., Shen, W., Xie, L., Jiang, D., ZHANG, X., Tian, Q.: Segment anything in 3d with nerfs. In: Oh, A., Neumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (eds.) Advances in Neural Information Processing Systems. vol. 36, pp. 25971–25990. Curran Associates, Inc. (2023)
6. Chang, M., Sharma, A., Kaess, M., Lucey, S.: Neural radiance field with lidar maps. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 17914–17923 (2023)
7. Chen, Z., Wang, F., Liu, H.: Text-to-3d using gaussian splatting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2024)
8. Cheng, K., Long, X., Yang, K., Yao, Y., Yin, W., Ma, Y., Wang, W., Chen, X.: Gaussianpro: 3d gaussian splatting with progressive propagation. arXiv preprint arXiv:2402.14650 (2024)
9. Clemente, L.A., Davison, A.J., Reid, I.D., Neira, J., Tardós, J.D.: Mapping large loops with a single hand-held camera. In: Robotics: Science and Systems. vol. 2 (2007)
10. Cui, H., Gao, X., Shen, S., Hu, Z.: Hsfm: Hybrid structure-from-motion. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1212–1221 (2017)
11. Davison: Real-time simultaneous localisation and mapping with a single camera. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1403–1410. IEEE (2003)
12. Dellaert, F., Seitz, S.M., Thorpe, C.E., Thrun, S.: Structure from motion without correspondence. In: Proceedings of the IEEE conference on computer vision and pattern recognition. vol. 2, pp. 557–564. IEEE (2000)
13. Fan, Z., Wang, P., Jiang, Y., Gong, X., Xu, D., Wang, Z.: Nerf-sos: Any-view self-supervised object segmentation on complex scenes. In: International Conference on Learning Representations (ICLR) (2023)
14. Fu, X., Zhang, S., Chen, T., Lu, Y., Zhu, L., Zhou, X., Geiger, A., Liao, Y.: Panoptic nerf: 3d-to-2d label transfer for panoptic urban scene segmentation. In: International Conference on 3D Vision. pp. 1–11. IEEE (2022)
15. Guo, J., Deng, N., Li, X., Bai, Y., Shi, B., Wang, C., Ding, C., Wang, D., Li, Y.: Streetsurf: Extending multi-view implicit surface reconstruction to street views. arXiv preprint arXiv:2306.04988 (2023)

16. Haas, J.K.: A history of the unity game engine. Diss. Worcester Polytechnic Institute **483**(2014), 484 (2014)
17. Huang, P.H., Matzen, K., Kopf, J., Ahuja, N., Huang, J.B.: Deepmvs: Learning multi-view stereopsis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2821–2830 (2018)
18. Kalaiah, A., Varshney, A.: Differential point rendering. In: Rendering Techniques 2001: Proceedings of the Eurographics Workshop in London, United Kingdom, June 25–27, 2001 12. pp. 139–150. Springer (2001)
19. Kalman, R.E.: A New Approach to Linear Filtering and Prediction Problems. Journal of Basic Engineering **82**(1), 35–45 (1960)
20. Kar, A., Häne, C., Malik, J.: Learning a multi-view stereo machine. Advances in neural information processing systems **30** (2017)
21. Karis, B., Games, E.: Real shading in unreal engine 4. Proc. Physically Based Shading Theory Practice **4**(3), 1 (2013)
22. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics **42**(4), 1–14 (2023)
23. Kopanas, G., Philip, J., Leimkühler, T., Drettakis, G.: Point-based neural rendering with per-view optimization. In: Computer Graphics Forum. vol. 40, pp. 29–43. Wiley Online Library (2021)
24. Li, L., Lian, Q., Chen, Y.C.: Adv3d: Generating 3d adversarial examples in driving scenarios with nerf. arXiv preprint arXiv:2309.01351 (2023)
25. Li, W., Pan, C., Zhang, R., Ren, J., Ma, Y., Fang, J., Yan, F., Geng, Q., Huang, X., Gong, H., et al.: Aads: Augmented autonomous driving simulation using data-driven algorithms. Science robotics **4**(28), eaaw0863 (2019)
26. Li, Z., Li, L., Zhu, J.: Read: Large-scale neural scene rendering for autonomous driving. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 1522–1529 (2023)
27. Lu, F., Xu, Y., Chen, G., Li, H., Lin, K.Y., Jiang, C.: Urban radiance field representation with deformable neural mesh primitives. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 465–476 (2023)
28. Meuleman, A., Liu, Y.L., Gao, C., Huang, J.B., Kim, C., Kim, M.H., Kopf, J.: Progressively optimized local radiance fields for robust view synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16539–16548 (2023)
29. Mihajlovic, M., Bansal, A., Zollhoefer, M., Tang, S., Saito, S.: Keypointnerf: Generalizing image-based volumetric avatars using relative spatial encoding of keypoints. In: Proceedings of the European conference on computer vision. pp. 179–197. Springer (2022)
30. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM **65**(1), 99–106 (2021)
31. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM Transactions on Graphics **41**(4), 1–15 (2022)
32. Mur-Artal, R., Montiel, J.M.M., Tardos, J.D.: Orb-slam: a versatile and accurate monocular slam system. IEEE transactions on robotics **31**(5), 1147–1163 (2015)
33. Mur-Artal, R., Tardós, J.D.: Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. IEEE transactions on robotics **33**(5), 1255–1262 (2017)

34. Ost, J., Laradji, I., Newell, A., Bahat, Y., Heide, F.: Neural point light fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18419–18429 (2022)
35. Reiser, C., Szeliski, R., Verbin, D., Srinivasan, P., Mildenhall, B., Geiger, A., Barron, J., Hedman, P.: Merf: Memory-efficient radiance fields for real-time view synthesis in unbounded scenes. ACM Transactions on Graphics **42**(4), 1–12 (2023)
36. Rematas, K., Liu, A., Srinivasan, P.P., Barron, J.T., Tagliasacchi, A., Funkhouser, T., Ferrari, V.: Urban radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12932–12942 (2022)
37. Rückert, D., Franke, L., Stamminger, M.: Adop: Approximate differentiable one-pixel point rendering. ACM Transactions on Graphics **41**(4), 1–14 (2022)
38. Sainz, M., Pajarola, R.: Point-based rendering techniques. Computers & Graphics **28**(6), 869–879 (2004)
39. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4104–4113 (2016)
40. Schönberger, J.L., Zheng, E., Frahm, J.M., Pollefeys, M.: Pixelwise view selection for unstructured multi-view stereo. In: Proceedings of the European conference on computer vision. pp. 501–518. Springer (2016)
41. Smith, M.W., Carrivick, J.L., Quincey, D.J.: Structure from motion photogrammetry in physical geography. Progress in physical geography **40**(2), 247–275 (2016)
42. Tancik, M., Casser, V., Yan, X., Pradhan, S., Mildenhall, B., Srinivasan, P.P., Barron, J.T., Kretzschmar, H.: Block-nerf: Scalable large scene neural view synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8248–8258 (2022)
43. Tang, J., Ren, J., Zhou, H., Liu, Z., Zeng, G.: Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. In: International Conference on Learning Representations (ICLR) (2024)
44. Tao, T., Gao, L., Wang, G., Chen, P., Hao, D., Liang, X., Salzmann, M., Yu, K.: Lidar-nerf: Novel lidar view synthesis via neural radiance fields. arXiv preprint arXiv:2304.10406 (2023)
45. Turki, H., Ramanan, D., Satyanarayanan, M.: Mega-nerf: Scalable construction of large-scale nerfs for virtual fly-throughs. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12922–12931 (2022)
46. Turki, H., Zhang, J.Y., Ferroni, F., Ramanan, D.: Suds: Scalable urban dynamic scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12375–12385 (2023)
47. Ullman, S.: The interpretation of structure from motion. Proceedings of the Royal Society of London. Series B. Biological Sciences **203**(1153), 405–426 (1979)
48. Wu, Z., Liu, T., Luo, L., Zhong, Z., Chen, J., Xiao, H., Hou, C., Lou, H., Chen, Y., Yang, R., et al.: Mars: An instance-aware, modular and realistic simulator for autonomous driving. In: CAAI International Conference on Artificial Intelligence. pp. 3–15. Springer (2023)
49. Xie, Z., Zhang, J., Li, W., Zhang, F., Zhang, L.: S-nerf: Neural radiance fields for street views. In: International Conference on Learning Representations (ICLR) (2023)
50. Xu, D., Jiang, Y., Wang, P., Fan, Z., Shi, H., Wang, Z.: Sinnerf: Training neural radiance fields on complex scenes from a single image. In: Proceedings of the European conference on computer vision. pp. 736–753. Springer (2022)

51. Xu, Q., Xu, Z., Philip, J., Bi, S., Shu, Z., Sunkavalli, K., Neumann, U.: Point-nerf: Point-based neural radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5438–5448 (2022)
52. Yang, J., Ivanovic, B., Litany, O., Weng, X., Kim, S.W., Li, B., Che, T., Xu, D., Fidler, S., Pavone, M., et al.: Emernerf: Emergent spatial-temporal scene decomposition via self-supervision. In: International Conference on Learning Representations (ICLR) (2024)
53. Yang, W., Chen, G., Chen, C., Chen, Z., Wong, K.Y.K.: Ps-nerf: Neural inverse rendering for multi-view photometric stereo. In: Proceedings of the European conference on computer vision. pp. 266–284. Springer (2022)
54. Yang, Z., Chen, Y., Wang, J., Manivasagam, S., Ma, W.C., Yang, A.J., Urtasun, R.: Unisim: A neural closed-loop sensor simulator. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1389–1399 (2023)
55. Yao, Y., Luo, Z., Li, S., Fang, T., Quan, L.: Mvsnet: Depth inference for unstructured multi-view stereo. In: Proceedings of the European conference on computer vision. pp. 767–783 (2018)
56. You, Z., Geiger, A., Chen, A.: Nelf-pro: Neural light field probes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2024)
57. Zhang, C., Yan, J., Wei, Y., Li, J., Liu, L., Tang, Y., Duan, Y., Lu, J.: Occnerf: Self-supervised multi-camera occupancy prediction with neural radiance fields. arXiv preprint arXiv:2312.09243 (2023)
58. Zwicker, M., Pfister, H., Van Baar, J., Gross, M.: Ewa volume splatting. In: Proceedings Visualization, 2001. VIS'01. pp. 29–538. IEEE (2001)
59. Zwicker, M., Pfister, H., Van Baar, J., Gross, M.: Surface splatting. In: Proceedings of the conference on Computer graphics and interactive techniques. pp. 371–378 (2001)