Cross-Spectral Gaussian Splatting with Spatial Occupancy Consistency

Haipeng Guo, Huanyu Liu*, Jiazheng Wen, Junbao Li,

Faculty of Computing, Harbin Institute of Technology, Harbin, China
haipengguo.hit@gmail.com, liuhuanyu@hit.edu.cn, 22b903087@stu.hit.edu.cn, lijunbao@hit.edu.cn

Abstract

Using images captured by cameras with different light spectrum sensitivities, training a unified model for cross-spectral scene representation is challenging. Recent advances have shown the possibility of jointly optimizing cross-spectral relative poses and neural radiance fields using normalized cross-device coordinates. However, such method suffers from cross-spectral misalignment when collecting data asynchronously from devices and lacks the capability to render in real-time or handle large scenes. We address these issues by proposing cross-spectral Gaussian Splatting with spatial occupancy consistency, strictly aligns cross-spectral scene representation by sharing explicit Gaussian surfaces across spectra and separately optimizing each view's extrinsic using a matching-optimizing pose estimation method. Additionally, to address field-of-view differences in cross-spectral cameras, we improve the adaptive densify controller to fill non-overlapping areas. Comprehensive experiments demonstrate that SOC-GS achieves superior performance in novel view synthesis and real-time cross-spectral rendering.

Code — https://github.com/GuoHP-HIT/SOC-GS.

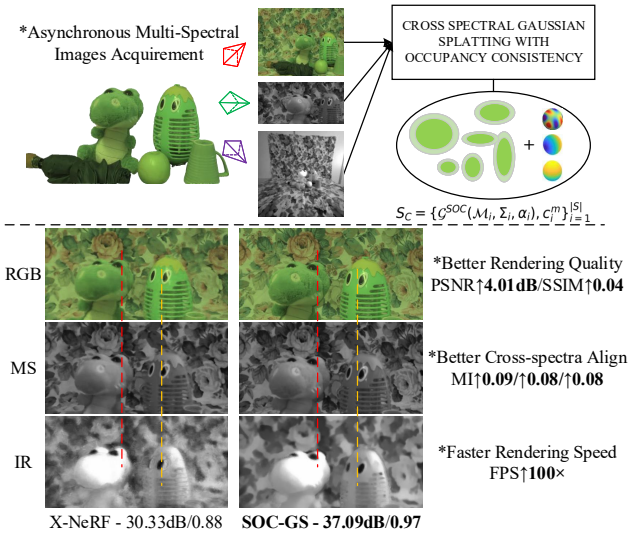


Figure 1: **Cross-spectral rendering comparison.** We propose SOC-GS for cross-spectral scene representation and pose estimation. Our method achieves more strictly alignment and better rendering quality in cross-spectral NVS.

Introduction

Novel View Synthesis (NVS) defines a task that represents the scene by the given sparse views, and generates novel views invisible during sampling, play an important role in fields such as 3D reconstruction (Remondino et al. 2023), AR/VR (Li et al. 2022), and autonomous driving (He et al. 2024) et al. With the proposal of NeRF (Mildenhall et al. 2021), use neural radiance fields to represent scenes as a 5D vector-valued function $F(\mathbf{x}, \theta, \phi)$, and map the representation to color $\mathbf{c}$ and volume density σ by MLP gradually become the mainstream choice for NVS. Further, 3DGS (Kerbl et al. 2023) represents the scene surface as a set of deterministic Gaussian functions, realize more adaptive and flexible 3D object representation which overcomes the limitations of volume rendering methods. Meanwhile, the differentiable rasterisation makes Gaussian splatting optimizing rapidly.

However, in real-world scene representation, visible spectral images alone may not provide complete scene views. Additional sensor data can be introduced to enhance the overall comprehensiveness of scene representation. For instance, IMU and LIDAR pointcloud data can serve as Gaussian Splatting surfaces initialization (Hong et al. 2024), while depth-resolved imaging sonar (Qadri et al. 2024), thermal images (Ye et al. 2024) and multi-spectral images (Li et al. 2024) can provide supplementary imaging data beyond RGB cameras, enhancing scene representation. Generally, the introduction of supplementary modalities in these systems is typically achieved by developing an imaging system that maintains fixed relative positions. This limitation arises from the challenge of cross-modality information alignment. When processing single spectral image inputs, camera poses can be easily obtained using the Structure-from-Motion (SfM) method. However, this approach proves ineffective when used with cross-spectral images. In cross-spectral scene representation, only the camera poses of one spectrum can be obtained by SfM; additional spectral camera poses obtained independently through SfM do not main-

*Corresponding authors

tain spatial consistency.

X-NeRF (Poggi et al. 2022) first proposed normalized cross device coordinates (NXDC) to align ray sampling between cameras and used a joint neural radiance field to represent cross-spectral scenes. The pose of each camera is determined by learning a fixed relationship with a specified standard spectra. Evidently, this approach results in cross-spectral representations that are dependent on specific imaging systems, thereby limiting the system ability. Additionally, the implicit representation complicates real-time rendering and the cross-spectral representation of large scenes.

In this paper, we propose SOC-GS to address limitations in current methods. Specifically, by leveraging the spatial occupancy consistency of the cross-spectral scene surface, we describe the consistency surface using 3D Gaussians. The fundamental attributes of these Gaussians are shared across the cross-spectral scene, with the colors of each spectrum represented by independent spherical harmonic coefficients. Given these settings, we individually optimize the poses of each view to reduce reliance on specific imaging systems. Due to the challenges in aligning cross-spectral cameras using a single matching or optimization method, we propose a two-stage Gaussian joint optimization pipeline for determining cross-spectral cameras poses. Initially, we use the attributes of Gaussians with spatial occupancy consistency and the LoFTR (Sun et al. 2021) matcher to roughly estimate the poses of cross-spectral cameras. Subsequently, the preliminary poses determined in the initial phase serve as initialization for the joint optimization of poses and Gaussian Splatting, aiming to achieve a cross-spectral Gaussian scene representation. Moreover, to enhance rendering quality, we improve a cross-spectral adaptive densify controller that is guided by the primary spectrum. This controller aims to enhance the reconstruction quality of non-overlapping areas within the FOV in cross-spectral view inputs.

Overall, our contributions can be summarized as:

- We propose SOC-GS, a pipeline that achieves cross-spectral representation from freely acquired images using shared Gaussian attributes with independent color spherical harmonics. Additionally, we enhance the reconstruction of non-overlapping areas with a cross-spectral densify controller.
- We propose a two-stage cross-spectral pose optimization pipeline, initiating poses by utilizing substantialized Gaussians and a pre-train keypoint matcher LoFTR, followed by a joint differentiable optimization of both poses and the scene.
- We additionally collect the RealSense dataset for cross-spectral scene representation. We evaluate the performance of SOC-GS using the X-NeRF and RealSense datasets, which involve bi-modality and tri-modality cross-spectral scene representation, and provide a comprehensive comparison with several advanced methods based on NeRF and 3DGS.

Related Works

Our work primarily focuses on two key components: the joint optimization of pose and scene, and the representation across different modalities. Below, we will discuss the current research in each of these components.

NVS with Camera Pose Optimization. Deep learning based novel view synthesis develop rapidly after the proposal of NeRF (Mildenhall et al. 2021), several researches focused on enhancing the rendering quality of NeRF (Zhang et al. 2020; Xu et al. 2022; Barron et al. 2021, 2022; Müller et al. 2022) and the efficiency in training or rendering process (Chen et al. 2022; Sun, Sun, and Chen 2022; Fridovich-Keil et al. 2022; Yu et al. 2021; Wang et al. 2022). Likewise, some researches attempt to reconstruct scenes from sparse views with unknown camera poses. NeRF−− (Wang et al. 2021a) first attempt to additionally solve the camera poses during NeRF optimization, BARF (Lin et al. 2021) and GARF (Chng et al. 2022) further proposed coarse-to-fine positional encoding strategy and Gaussian-MLPs to achieve more precise joint optimization of pose and scene. Nope-NeRF (Bian et al. 2023) achieved the best performance of NeRF-based methods by incorporating inter-frame monocular depth loss. Following the significant improvement of NVS quality and real-time rendering ability by 3DGS (Kerbl et al. 2023), CF-3DGS (Fu et al. 2023) and InstantSplat (Fan et al. 2024) jointly optimize pose with Gaussians by progressively grow Gaussians and integrating point-based representation with end-to-end dense stereo model separately. Our SOC-GS also built on 3DGS pipeline to represent scene from cross-spectral images and jointly optimize cross-spectral Gaussians and pose.

Cross Modality Scene Representation. Cross modality scene representation plays an important role in world simulation and SLAM. Several simulators for real-world scenes incorporate different sensor into their scene representation. For instance, AADS (Li et al. 2019) and UniSim (Yang et al. 2023) integrate LIDAR point cloud into autonomous driving simulators, while DrivingGaussian (Zhou et al. 2023) utilizes LIDAR as a prior to initialize Gaussian Splatting in driving simulator. Similarly in SLAM, previous works (Hong et al. 2024; Wu et al. 2024; Jeong, Yoon, and Park 2018) utilized LIDAR priors to improve the performance of SLAM systems, the follow-up researches integrate more sensors such as IMU et al. into SLAM systems (Lang et al. 2024; Sun et al. 2024). Additionally, some researches introduce other spectral images lacking depth reference information into scene representation. SpectralNeRF (Li et al. 2024) represent multi-spectral scenes by multiple networks, X-NeRF (Poggi et al. 2022) integrate cross-spectral representation into NeRF by normalized cross-device coordinates, addressing the representation of inputs from multiple cameras. Our SOC-GS continue the mode of X-NeRF, reach the real-time rendering of cross-spectral scene representation, and eliminate the reliance on stationary imaging devices.

Method

We propose a 3DGS-based cross-spectral scene representation pipeline. As shown in Fig.2, the pipeline consists of three components: single spectral Gaussian pre-training, cross-spectral camera poses initialization and jointly Gaussian Splatting training. In the initial step, we pre-train Gaussian attributes by RGB images and corresponding SfM

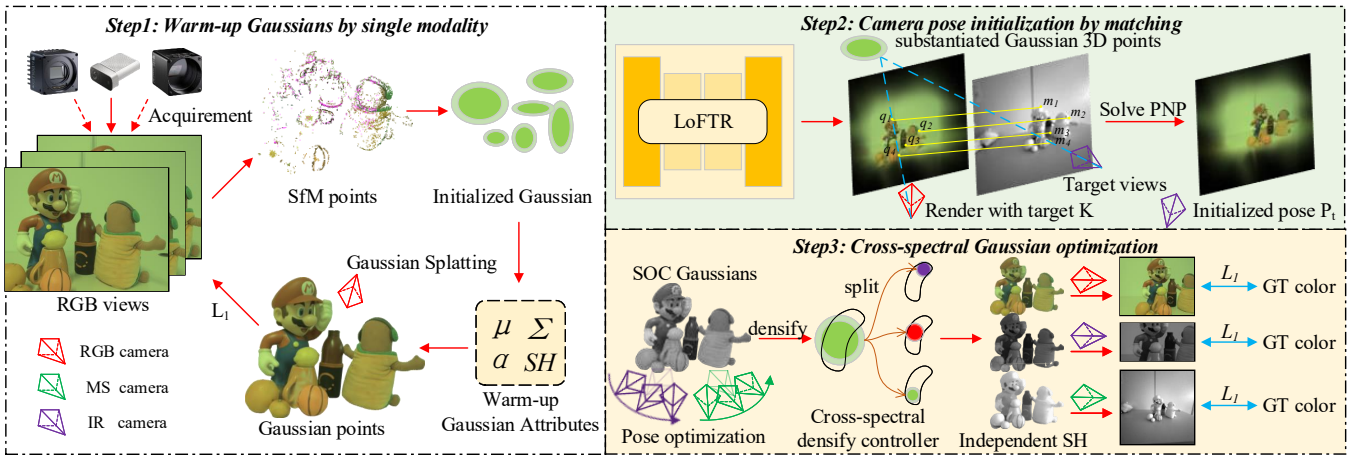


Figure 2: **Training pipeline of SOC-GS.** Given images acquired by cross-spectral devices, we first perform Gaussian warm-up on the RGB spectral, then use LoFTR keypoint matcher to obtain 2D matching points between rendered image(with RGB camera extrinsic and target intrinsic) and target spectral ground truth, the Gaussians are substantiated to solve PNP in the process. Finally, the spatial occupancy consistency Gaussians differentiably jointly optimized with target poses, and densified by a cross-spectral controller.

pointcloud. Subsequently, we substantialize Gaussians and initialize cross-spectral poses based on matching. Finally, the poses and Gaussians are jointly differentiably optimized to achieve cross-spectral scene representation.

Spatial Occupancy Consistent Cross-Spectral 3D Gaussian Splatting

3D Gaussian Splatting decompose scene representation into a set of explicit combinations $G = \{(\mathcal{M}_i, \Sigma_i, c_i, \sigma_i)\}_{i=1}^{|G|}$, where $\mathcal{M}_i$, Σ_i, c_i and σ_i represent mean, covariance matrix, color and opacity of $i-th$ Gaussian primitive. More specifically, $\mathcal{M}_i \in R^3$ denotes the 3D coordinate of the center of anisotropic spheres. To ensure covariance matrix are positive semi-definite, the covariance matrix Σ composed of given a rotation matrix R and scaling matrix S:

$$\Sigma = \mathbf{RSS}^T\mathbf{R}^T \tag{1}$$

where the scale and rotation are denoted by 3D vector s and unit quaternion q respectively to ensure normalization. Finally, each Gaussian primitive can be expressed as:

$$G(X) = e^{-\frac{1}{2}\mathcal{M}^T\Sigma^{-1}\mathcal{M}} \tag{2}$$

Then following Zwicker et al.(Zwicker et al. 2001), project 3D Gaussians to 2D for rendering:

$$\Sigma' = JW\Sigma W^T J^T \tag{3}$$

where J is the Jacobian of the affine approximation of the projective transformation and W represents for given viewing transformation. After splatting, the image is divided into patches, the Gaussians intersected with corresponding patch are sorted by depth, then accumulate the color of each pixel by α-blending (Kopanas et al. 2022), c_i is a three channel color calculated from spherical harmonic coefficient, and α_i is the shared transparency:

$$C = \sum_{i\in\mathcal{N}} c_i\alpha_i \prod_{j=1}^{i-1}(1-\alpha_j) \tag{4}$$

Gaussian Splatting initialized by the side-effect pointcloud of SfM and optimized by minimizing the L_1 distance between rasterized view and ground truth. In this article, we propose a hypothesis that Gaussian primitives can provide suitable approximation for scene surface, and such approximation holds across different spectral. Given this premise, we can expend Gaussian Splatting to cross-spectral, let $\mathcal{G}^{SOC}$ denotes the spatial occupancy consistency Gaussian primitive with common mean $\mathcal{M}$, covariance Σ and opacity σ. Then c^m signifies the independent color representation within each spectral:

$$S_C = \{\mathcal{G}^{SOC}(\mathcal{M}_i, \Sigma_i, \alpha_i), c_i^m\}_{i=1}^{|S|} \tag{5}$$

What we need to do is optimize the unified Gaussian and spherical harmonics in each spectral. In SOC-GS, we use the views of RGB camera for preliminary 3D Gaussian pretrain, the optimization of cross-spectral poses and scene representation are both based on these pre-trained Gaussians.

Cross-spectral Pose Initialization by Matching

Our objective is to introduce cross-spectral representation based on pre-train Gaussians. Take X-NeRF for example, where the scene is captured using cameras in RGB, multi-spectral, and infrared spectral. The intrinsic of these three cameras has been previously calibrated, with only the poses of RGB camera being determined by SfM. X-NeRF combined these cameras to form an imaging system that ensure a fixed position relationship between each camera. To mitigate the influence of shutter synchronization errors on scene representation and improve system flexibility, we optimize the extrinsic of each view separately in SOC-GS, so that SOC-GS can be easily extended to sparsely constrained imaging systems. Since cameras intrinsic are known, we firstly render on pre-trained Gaussians with known RGB camera pose and target camera intrinsic that require initialized:

$$\hat{\mathbf{I}}_m^{ref} = \mathcal{G}^{pre}(\mathbf{P}_{RGB}, \mathbf{K}_m), m \in (MS, Infrared) \tag{6}$$

where $\mathcal{G}^{pre}$ represents the pre-trained Gaussian rendering function, $\mathbf{P}_{RGB}$ is the known pose of RGB camera and $\mathbf{K}_m$ represents target multi-spectral/infrared camera intrinsic. We simply register each reference image $\hat{\mathbf{I}}^{ref}$ with corresponding captured image from spectral m to form a registration pair$\{\mathbf{I}_i^m, \hat{\mathbf{I}}_m^{ref}\}_{i=1}^N$. For more general cases, just need compute the average Euclidean distance of matching keypoints between the captured and reference. The image with lowest value can be chosen to form the registration pair.

Then within predefined confidence, we use LoFTR(Sun et al. 2021) to detect matching keypoints between the rendered and query images. Different from iComMa(Sun et al. 2023), we refrain from employing matching results for differentiable optimization, as local feature matching methods exhibit instability when applied in cross-spectral views. Instead, we use the matching keypoint pairs $\{q_i, m_i\}$ to determine the shared 3D anchor points $\mathbf{p}_i$ in reverse:

$$\begin{cases} \mathbf{K}_s[R_s|t_s]\mathbf{p}_i = q_i, \\ \mathbf{K}_s[R_t|t_t]\mathbf{p}_i = m_i \end{cases} \tag{7}$$

where $\mathbf{K}_s$ denotes the camera intrinsic of source reference image. To determine the 3D anchor points, we sort the Gaussians intersected with keypoint pixels by depth, which happens to be a sub-product of vanilla Gaussian Splatting optimization. The closest Gaussian is substantialized in matrix form of an ellipsoid——mean $\mathcal{M}$ as the center of ellipsoid, quaternion q and scale s in the covariance component are normalized as the rotation and scale matrix of ellipsoid:

$$E = TRS \tag{8}$$

where T, R, S are the translation, rotation and scale matrix of the ellipsoid E. Meanwhile, representing the ray through camera optical center and keypoint of $\hat{\mathbf{I}}^{ref}$ in form of parametric equations, and perform inverse transformation in the form of ellipsoidal transformation matrix E:

$$r'_{(t)} = E^{-1}o + E^{-1}t\mathbf{d} \tag{9}$$

solving the intersection of ray and ellipsoid is equivalent to following equation:

$$\| E^{-1}o + E^{-1}t\mathbf{d} \|^2 = 1 \tag{10}$$

The detailed derivation process can be found in the appendix. By employing a series of 2D-3D registration pairs, we determine target camera pose using EPNP (Lepetit, Moreno-Noguer, and Fua 2009) and eliminate outliers by RANSAC (Fischler and Bolles 1981). We use the outcome of Perspective-n-points (PnP) method for pose initialization.

Differentiable Joint Optimization

Matching-based method can rapidly provide initial pose estimation for cross-spectral cameras. However, the predicted points and confidence maps derived from cross-spectral images matching may lack precision and thoroughness, resulting in blurred Gaussian scene representation. To mitigate such blur, we further jointly optimize the poses and scene by cross-spectral views, the optimization of poses is independent of Gaussian attributes. Follow the settings of NeRF−−(Wang et al. 2021b), we optimize the poses for each input image I_i^m of modality m with trainable rotation vector ϕ_i^m and translation vector $\mathbf{t}_i^m$, then convert $\mathbf{t}_i^m$ to matrix from Rodrigues' formula to ensure the rotation matrix $\mathbf{R} \in SO(3)$. The outcome of preceding stage is set as the initial rotation and translation vector. After initialization, we learn a set of cross-spectral spatial occupancy consistency Gaussians to minimize the photometric loss between the rendered image and corresponding spectral frame $I_{v,m}$:

$$\mathcal{G}^*, \mathbf{P}_m^* = \arg\min_{\mathcal{G}, \mathbf{P}_m} \sum_{v \in N} \sum_{i=1}^{HW} \| \tilde{\mathcal{G}}^i(\mathbf{P}_{v,m}, \mathbf{K_m}) - I_{v,m}^i \| \tag{11}$$

where $\mathcal{G}$ represents Gaussian rendering function with shared attributes, $\mathbf{P}_{v,m}$ signifies the spectral dependent view poses. This formulation jointly optimize the Gaussian attributes and cross-spectral camera poses by pixel difference without introducing additional constraints. At each optimization iteration, we choose one spectral m in sequence as current optimization spectral, and utilize corresponding view to calculate the photometric loss for optimization. The photometric loss $\mathcal{L}_m$ is $\mathcal{L}_1$ combined with a D-SSIM term:

$$\mathcal{L}_m = (1 - \lambda)\mathcal{L}_1 + \lambda\mathcal{L}_{D-SSIM} \tag{12}$$

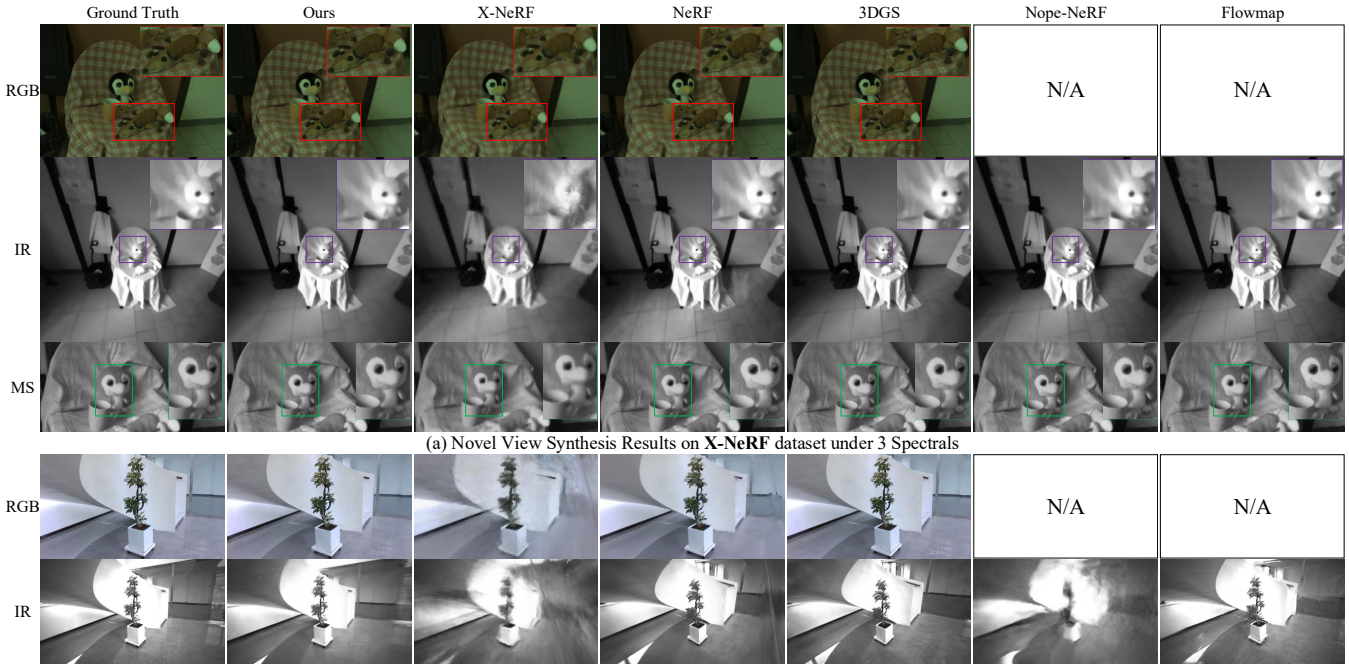
We use $\lambda = 0.2$ for all experiments, consistent with 3DGS.

Meanwhile, to mitigate the substantial expenses from automatic differentiation, we extend the derivation of poses gradient based on 3DGS. The details regarding the calculations of derivatives is presented in appendix. Note that the majority of gradients necessary for pose estimation are already utilized by Gaussian optimization, we can reuse these gradients for minimal additional computational expenses.

Cross-spectral Adaptive Densify Controller

Due to the sequential optimization conducted by SOC-GS on cross-spectral views, maintaining the Gaussian densify strategy during pre-training may result in ambiguity, and under-reconstruction of areas beyond primary FOV. Therefore, we extend the densify controller to cross-spectral. In optimization, we individually accumulate the gradient in view-space for each spectral and periodically eliminate transparent Gaussians. For possible floaters excessive occupied in view-space, we use a fixed threshold for affected pixels and remove floaters by the gradients derived from the spectral with the smallest FOV. For Gaussians with average magnitude of view-space position gradients above threshold, we also distinguish such Gaussians as under-reconstruction and over-reconstruction to apply densify. The region under-reconstruction do not involve optimization ambiguity, it simply requires cloning based on corresponding spectral.

Relatively, the region over-reconstruction exist optimization ambiguity, the different accumulated gradients of each spectral result in disparate Gaussian PDF sampling during densification, repeated densify may lead to ineffective optimization. Therefore, in SOC-GS, the original Gaussian only sample according to pre-train spectral, the Gaussians added by splitting are initialized by each spectral separately. Intuitively speaking, we fill the out-of-FOV regions while maintaining pre-trained Gaussians attributes as much as possible.



(a) Novel View Synthesis Results on **X-NeRF** dataset under 3 Spectrals

(b) Novel View Synthesis Results on **RealSense** dataset under 2 Spectrals

Figure 3: **Qualitative Comparison of Novel View Synthesis on (a) X-NeRF and (b) RealSense datasets.** Each camera pose corresponds to three spectral rendering results of RGB, IR, MS from top to bottom.

Experiments

Experimental Setup

Datasets. We evaluated the performance of our method using the publicly X-NeRF dataset and our self-collected RealSense dataset. **(1) X-NeRF dataset.** X-NeRF consists of 16 indoor forward-facing scenes, each containing about 30 images from RGB, Multi-spectral and Infrared cameras, 5 images reserved for testing, and the rest for training. The three cameras are mounted on one single device to maintain a fixed relative alignment, simultaneously capturing RGB, infrared, and multi-spectral images. All scenes in the X-NeRF dataset are indoor and cover a limited range of movement. **(2) RealSense dataset.** To evaluate the performance of cross-scene representation methods in a wider variety of scenarios, we also captured three indoor scenes and two outdoor scenes using the Intel RealSense D435 depth camera. Each scene includes a variable number of RGB and binocular infrared images at a resolution of 720p. For the outdoor scenes, structured light sensors were used to obtain accurate depth maps.

Implementation Details. We implemented our method using PyTorch framework on one single RTX3090 by taichi (Hu et al. 2019, 2020, 2021) to accelerate optimization and rasterisation. We start from the point cloud output from SfM to pre-train Gaussians for 5000 iterations using only RGB images, then initialize the camera poses of the remaining spectral images. Each view that requires initialization takes approximately 2 seconds. After initialization, the Gaussian attributes are unfrozen and enhanced in combination with color spherical harmonics and cross-spectral poses, optimizing all involved spectra with each iteration. We set the confidence threshold for LoFTR matcher to 0.5.

Metrics. For novel view synthesis evaluation, we use PSNR and SSIM to compare the quality of rendered images and ground truth. To evaluate pose accuracy in cross-spectral representation, the Euclidean Metric (EM) is used instead of the ATE and RTE metrics because only the RGB camera poses are known, and the other camera poses lack ground truth. Specifically, we use LoFTR to extract matching keypoints from rendered images and the ground truth and then calculate the average EM. Only LoFTR keypoints with confidence greater than 0.5 are used for the EM computation.

Novel View Synthesis Experimental Results

We compare the NVS performance of our method with vanilla NeRF, 3D Gaussian Splatting, Nope-NeRF, Flowmap (Smith et al. 2024), and X-NeRF using both the X-NeRF and RealSense datasets. Among these methods, only X-NeRF and our method possess cross-spectral rendering capabilities, meaning the results are generated by a unified model. The results from the other methods are generated by rendering models that were independently trained on each spectra. We use NeRF and 3DGS as baseline methods to compare the rendering quality of our method. Additionally, we include Nope-NeRF and Flowmap, which are the optimal pose estimation methods based on these baselines, to evaluate the pose estimation performance. Since the poses of the RGB images are already known, we do not provide the NVS results of these two methods for the RGB images.

Set.	Tri-Modality Avg. (PSNR↑/SSIM↑/EM↓)									Bi-Modality Avg. (PSNR↑/SSIM↑/EM↓)					
Method|Spectra	RGB			Multi-spectral			Infrared			RGB			Multi-spectral		
NeRF	34.476	0.909	**1.165**	39.378	0.982	0.755	37.204	*0.980*	8.000	*34.476*	0.909	1.165	39.378	**0.982**	0.755
3DGS	*34.665*	*0.922*	1.578	*38.788*	*0.981*	0.922	**37.302**	0.981	8.043	**34.665**	0.922	*1.578*	*38.788*	*0.981*	*0.922*
Nope-NeRF	N/A	N/A	N/A	25.967	0.748	56.459	28.690	0.887	23.210	N/A	N/A	N/A	25.967	0.748	56.459
Flowmap	N/A	N/A	N/A	37.376	0.950	1.203	35.962	0.975	*7.265*	N/A	N/A	N/A	37.376	0.950	1.203
X-NeRF	31.699	0.888	2.766	35.541	0.945	0.992	33.253	0.935	15.179	32.089	0.890	2.510	35.922	0.950	1.429
SOC-GS$_{30k}$	34.809	0.922	*1.356*	36.260	0.964	*0.911*	33.801	0.970	7.201	34.472	*0.921*	1.682	38.547	0.979	0.958
SOC-GS$_{ext}$	**34.907**	**0.922**	1.350	**40.492**	**0.988**	**0.710**	*36.808*	**0.981**	**5.976**	34.483	**0.922**	**0.854**	**40.081**	0.981	**0.672**

Table 1: NVS quantitative comparisons with other methods on 16 scenes of X-NeRF dataset. The subscript $_{ext}$ denotes extending training time to match the total of all spectra's. The results from best to 3rd are represented in **bold**, underline, and *italic*.

Set.	Indoor Avg. (PSNR↑/SSIM↑/EM↓)						Outdoor Avg. (PSNR↑/SSIM↑/EM↓)					
Method|Spectra	RGB			Infrared			RGB			Infrared		
NeRF	*26.413*	*0.769*	*3.566*	*25.579*	*0.868*	*35.429*	*22.551*	*0.505*	*3.182*	18.811	0.666	48.970
3DGS	29.220	0.838	**1.321**	29.047	**0.952**	**1.609**	**25.265**	**0.728**	0.715	25.464	**0.907**	**0.787**
Nope-NeRF	N/A	N/A	N/A	19.222	0.792	89.214	N/A	N/A	N/A	13.362	0.559	126.704
Flowmap	N/A	N/A	N/A	27.732	0.883	40.513	N/A	N/A	N/A	*22.447*	*0.790*	*21.851*
X-NeRF	20.654	0.656	82.198	21.308	0.774	60.166	18.340	0.395	25.039	14.601	0.505	60.844
SOC-GS	**29.253**	**0.838**	1.507	**29.579**	0.932	9.628	25.085	0.712	**0.605**	**26.456**	0.862	4.370

Table 2: NVS quantitative comparisons in terms of PSNR, SSIM, EM on 5 scenes of Real-Sense dataset.

Furthermore, note that since our method is based on 3DGS, the spherical harmonic coefficients that define the color representation in SOC-GS are optimized independently for each spectral. Consequently, we extend the training iterations in the three cross-spectral rendering tasks to match the total optimization time in single-spectral representation.

Results on the X-NeRF dataset. As shown in Tab. 1, we evaluate the rendering quality across three spectra and two spectra on the X-NeRF dataset. The results demonstrate that the proposed SOC-GS outperforms the competing X-NeRF in NVS performance under both settings. Additionally, in pose optimization, SOC-GS significantly surpasses the two pose estimation methods, owing to the pre-trained 3D Gaussians as pose estimation reference. Our SOC-GS method effectively leverages the optimized consistency of Gaussian representations across cross-spectral views, leading to better performance than 3DGS in partial spectra. The qualitative experimental results in Fig.3 (a) visually support the performance gains. The rendering quality of SOC-GS is noticeably better than that of X-NeRF across all three spectra, with details and texture quality comparable to pose-known methods.

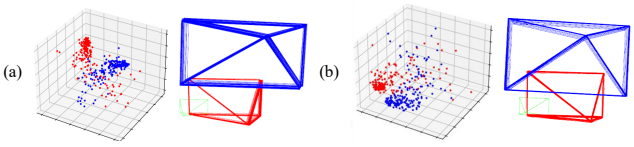


Figure 4: **Camera translations during training.** Left: we show camera translation on two scenes $(a)cvlab$ and $(b)penguin2$. Right: we show corresponding view frustums.

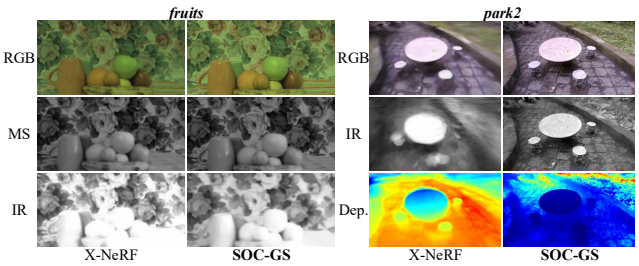


Figure 5: Quantitative results of cross-spectral rendering. We presented the results from scene $fruits$ and $park2$.

In addition to using EM metric to evaluate pose alignment from optimized Gaussians, we further plot the translation of camera centers in Fig.4 to observe the convergence speed of pose optimization. The MS and Infrared cameras are represented in blue and red respectively, with known RGB camera pose shown in green. We observe that after about 10,000 iterations, the camera positions stabilize. Concurrently, the relative positions of the three cameras become fixed, aligning with the settings used in X-NeRF collection.

Results on the RealSense dataset. Similarly, we evaluated the NVS performance on the collected RealSense dataset, incorporating the stereo infrared camera imaging as an additional spectral for scene representation. Five pairs of images were reserved for evaluation, with the results presented in Tab.2. Due to the increased span of scene acquisition, the non-strict synchronization of camera capture times leads to instability in the relative poses of the final images. In such scenes, X-NeRF's reliance on fixed relative relation-

ships results in blurred scenes, as evidenced by the qualitative results in Fig.3(b). In contrast, our SOC-GS independently optimizes the pose of each view, mitigating the impact of hardware capture limitations on scene representation.

Cross-spectral Rendering Experimental Results

We further evaluated the performance of SOC-GS and X-NeRF in cross-spectral rendering. We show the comparison results in Fig.5, the rendering intrinsic determined by minimal common FOV. The quantitative experimental results show that proposed SOC-GS perform better than X-NeRF in texture rendering (as shown in scene $fruits$) and cross-spectral alignment (as shown in scene $park2$). The rendered results of the park2 scene include depth estimation maps obtained by each method. It is evident that the depth map generated by SOC-GS aligns more closely with the rendering results, indicating that the model has learned more accurate spectral-independent geometric features of the scene. Additionally, another significant advantage of our method is the real-time rendering ability. As shown in Tab.3, we compared the cross-spectral rendering speed at different resolutions. SOC-GS can achieve about 120 fps in 1Mpx resolution rendering, much higher than X-NeRF. Additionally, we use mutual information (MI) as a metric to demonstrate the content relevance of the generated cross-spectral views. Evidently, our SOC-GS performs better.

Set.	Rendering Cost vs. Resolution (FPS↑/MI↑)				
Method|Res.	X-NeRF dataset			RealSense dataset	
	RGB_{12M}	$MS_{0.1M}$	$IR_{1.0M}$	$RGB_{0.9M}$	$IR_{0.9M}$
X-NeRF	0.01/0.78	1.82/1.07	0.16/0.99	0.15/0.61	0.15/0.46
SOC-GS	**17.3/0.87**	**235/1.15**	**119/1.07**	**95.7/1.31**	**95.7/0.53**

Table 3: Cross-spectral rendering cost across resolutions. We report FPS and MI on X-NeRF and RealSense dataset.

	Tri-modality Avg.			Bi-modality Avg.	
Set.	RGB	MS	IR	RGB	MS
p_i p_j w_i	(a) Ablation of pose optimization (PSNR↑/SSIM↑)				
√	34.00/0.93	33.25/0.94	29.19/0.94	34.18/0.93	32.60/0.93
√ √	34.36/0.93	**40.12/0.99**	31.08/0.96	33.65/0.92	**39.43/0.97**
√ √	32.01/0.92	23.41/0.82	19.31/0.83	33.46/0.92	24.20/0.80
√ √	32.80/0.92	23.12/0.91	19.34/0.83	32.13/0.92	22.21/0.75
√ √ √	**34.37/0.93**	37.56/0.98	**34.65/0.97**	**34.18/0.93**	39.10/0.98
h_p d_c	(b) Ablation of densify controller (PSNR↑/SSIM↑)				
√	**34.62/0.93**	**37.77/0.97**	32.11/0.97	**34.79/0.93**	38.97/0.98
√	33.35/0.93	36.54/0.97	33.43/0.97	34.35/0.93	38.26/0.97
√ √	34.37/0.93	37.56/0.98	**34.65/0.97**	34.18/0.93	**39.10/0.98**

Table 4: **Ablation results.** (a) Ablation of pose optimization, p_i and p_j represent for the initialization and joint optimization step. w_i represents the poses optimized independently. (b) Ablation of densify controller. d_c and h_p represent the densify controller in SOC-GS and the hold strategy.

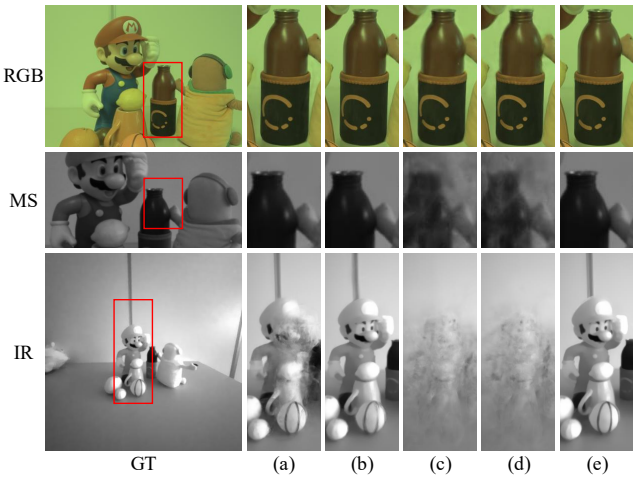


Figure 6: Quantitative results of pose optimization ablation.

Ablation Studies

We present ablation studies on 5 scenes to verify and analyze the effectiveness of each component of SOC-GS.

Fine-to-Coarse pose optimization. We present the relationship between rendering results and the two-stage pose optimization in Tab.4(a). When joint optimization is not enabled, the rendering performance for MS improves, while the infrared decreases. This is because our pose initialization method directly estimates poses well based on high spectral similarity, and fixed poses further aid in optimizing the Gaussian attributes. The absence of both steps implies the direct scene optimization using independently obtained COLMAP poses. The rendering result shown in Fig.6(d) confirm that the poses obtained independently have no consistency, which lead to rendering ghost.

Adaptive Controller. Similarly, we conducted an ablation study on the cross-spectral adaptive densify controller, as shown in Tab.4(b). When the Gaussians need to be split are not preserved, rendering performance improves under RGB and MS spectra, but decreases for the infrared spectra. Notably, the infrared camera has a much larger FOV than the others, meaning that preserving the original Gaussians is crucial for accurately reconstructing open areas.

Conclusion

We have proposed SOC-GS, a Spatial Occupancy Consistency Gaussian Splatting pipeline, to achieve real-time cross-spectral scene rendering. Our approach utilizes a two-step pose optimization strategy to initialize cross-spectral camera poses, which are then jointly optimized with the scene representation. Leveraging a cross-spectral adaptive densification controller, our method generates high-quality cross-spectral views. Comprehensive experiments have demonstrated the superiority of the proposed SOC-GS. In the future, we will further investigate the benefits of cross-spectral views on Gaussian geometry optimization to enhance the accuracy of scene geometry representation.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (NSFC) under Grants No.62271166 and No.62401177.

References

Barron, J. T.; Mildenhall, B.; Tancik, M.; Hedman, P.; Martin-Brualla, R.; and Srinivasan, P. P. 2021. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 5855–5864.

Barron, J. T.; Mildenhall, B.; Verbin, D.; Srinivasan, P. P.; and Hedman, P. 2022. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5470–5479.

Bian, W.; Wang, Z.; Li, K.; Bian, J.-W.; and Prisacariu, V. A. 2023. Nope-nerf: Optimising neural radiance field with no pose prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4160–4169.

Chen, A.; Xu, Z.; Geiger, A.; Yu, J.; and Su, H. 2022. TensoRF: Tensorial Radiance Fields. *arXiv preprint arXiv:2203.09517*.

Chng, S.-F.; Ramasinghe, S.; Sherrah, J.; and Lucey, S. 2022. GARF: gaussian activated radiance fields for high fidelity reconstruction and pose estimation. *arXiv e-prints*, arXiv–2204.

Fan, Z.; Cong, W.; Wen, K.; Wang, K.; Zhang, J.; Ding, X.; Xu, D.; Ivanovic, B.; Pavone, M.; Pavlakos, G.; et al. 2024. InstantSplat: Unbounded Sparse-view Pose-free Gaussian Splatting in 40 Seconds. *arXiv preprint arXiv:2403.20309*.

Fischler, M. A.; and Bolles, R. C. 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6): 381–395.

Fridovich-Keil, S.; Yu, A.; Tancik, M.; Chen, Q.; Recht, B.; and Kanazawa, A. 2022. Plenoxels: Radiance Fields Without Neural Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5501–5510.

Fu, Y.; Liu, S.; Kulkarni, A.; Kautz, J.; Efros, A. A.; and Wang, X. 2023. Colmap-free 3d gaussian splatting. *arXiv preprint arXiv:2312.07504*.

He, L.; Li, L.; Sun, W.; Han, Z.; Liu, Y.; Zheng, S.; Wang, J.; and Li, K. 2024. Neural Radiance Field in Autonomous Driving: A Survey. arXiv:2404.13816.

Hong, S.; He, J.; Zheng, X.; Wang, H.; Fang, H.; Liu, K.; Zheng, C.; and Shen, S. 2024. LIV-GaussMap: LiDAR-Inertial-Visual Fusion for Real-time 3D Radiance Field Map Rendering. *arXiv preprint arXiv:2401.14857*.

Hu, Y.; Anderson, L.; Li, T.-M.; Sun, Q.; Carr, N.; Ragan-Kelley, J.; and Durand, F. 2020. DiffTaichi: Differentiable Programming for Physical Simulation. *ICLR*.

Hu, Y.; Li, T.-M.; Anderson, L.; Ragan-Kelley, J.; and Durand, F. 2019. Taichi: a language for high-performance computation on spatially sparse data structures. *ACM Transactions on Graphics (TOG)*, 38(6): 201.

Hu, Y.; Liu, J.; Yang, X.; Xu, M.; Kuang, Y.; Xu, W.; Dai, Q.; Freeman, W. T.; and Durand, F. 2021. QuanTaichi: A Compiler for Quantized Simulations. *ACM Transactions on Graphics (TOG)*, 40(4).

Jeong, J.; Yoon, T. S.; and Park, J. B. 2018. Towards a meaningful 3D map using a 3D lidar and a camera. *Sensors*, 18(8): 2571.

Kerbl, B.; Kopanas, G.; Leimkühler, T.; and Drettakis, G. 2023. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4): 1–14.

Kopanas, G.; Leimkühler, T.; Rainer, G.; Jambon, C.; and Drettakis, G. 2022. Neural point catacaustics for novel-view synthesis of reflections. *ACM Transactions on Graphics (TOG)*, 41(6): 1–15.

Lang, X.; Li, L.; Zhang, H.; Xiong, F.; Xu, M.; Liu, Y.; Zuo, X.; and Lv, J. 2024. Gaussian-LIC: Photo-realistic LiDAR-Inertial-Camera SLAM with 3D Gaussian Splatting. *arXiv preprint arXiv:2404.06926*.

Lepetit, V.; Moreno-Noguer, F.; and Fua, P. 2009. EP n P: An accurate O (n) solution to the P n P problem. *International journal of computer vision*, 81: 155–166.

Li, C.; Li, S.; Zhao, Y.; Zhu, W.; and Lin, Y. 2022. RT-NeRF: Real-time on-device neural radiance fields towards immersive AR/VR rendering. In *Proceedings of the 41st IEEE/ACM International Conference on Computer-Aided Design*, 1–9.

Li, R.; Liu, J.; Liu, G.; Zhang, S.; Zeng, B.; and Liu, S. 2024. SpectralNeRF: Physically Based Spectral Rendering with Neural Radiance Field. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 3154–3162.

Li, W.; Pan, C.; Zhang, R.; Ren, J.; Ma, Y.; Fang, J.; Yan, F.; Geng, Q.; Huang, X.; Gong, H.; et al. 2019. AADS: Augmented autonomous driving simulation using data-driven algorithms. *Science robotics*, 4(28): eaaw0863.

Lin, C.-H.; Ma, W.-C.; Torralba, A.; and Lucey, S. 2021. BARF: Bundle-Adjusting Neural Radiance Fields. In *IEEE International Conference on Computer Vision (ICCV)*.

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

Müller, T.; Evans, A.; Schied, C.; and Keller, A. 2022. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4): 1–15.

Poggi, M.; Ramirez, P. Z.; Tosi, F.; Salti, S.; Mattoccia, S.; and Di Stefano, L. 2022. Cross-spectral neural radiance fields. In *2022 International Conference on 3D Vision (3DV)*, 606–616. IEEE.

Qadri, M.; Zhang, K.; Hinduja, A.; Kaess, M.; Pediredla, A.; and Metzler, C. A. 2024. AONeuS: A Neural Rendering Framework for Acoustic-Optical Sensor Fusion. *arXiv preprint arXiv:2402.03309*.

Remondino, F.; Karami, A.; Yan, Z.; Mazzacca, G.; Rigon, S.; and Qin, R. 2023. A critical analysis of NeRF-based 3d reconstruction. *Remote Sensing*, 15(14): 3585.

Smith, C.; Charatan, D.; Tewari, A.; and Sitzmann, V. 2024. FlowMap: High-Quality Camera Poses, Intrinsics, and Depth via Gradient Descent. *arXiv preprint arXiv:2404.15259*.

Sun, C.; Sun, M.; and Chen, H.-T. 2022. Direct Voxel Grid Optimization: Super-fast Convergence for Radiance Fields Reconstruction. *CVPR*.

Sun, J.; Shen, Z.; Wang, Y.; Bao, H.; and Zhou, X. 2021. LoFTR: Detector-free local feature matching with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 8922–8931.

Sun, L. C.; Bhatt, N. P.; Liu, J. C.; Fan, Z.; Wang, Z.; Humphreys, T. E.; and Topcu, U. 2024. MM3DGS SLAM: Multi-modal 3D Gaussian Splatting for SLAM Using Vision, Depth, and Inertial Measurements. *arXiv preprint arXiv:2404.00923*.

Sun, Y.; Wang, X.; Zhang, Y.; Zhang, J.; Jiang, C.; Guo, Y.; and Wang, F. 2023. icomma: Inverting 3d gaussians splatting for camera pose estimation via comparing and matching. *arXiv preprint arXiv:2312.09031*.

Wang, H.; Ren, J.; Huang, Z.; Olszewski, K.; Chai, M.; Fu, Y.; and Tulyakov, S. 2022. R2L: Distilling Neural Radiance Field to Neural Light Field for Efficient Novel View Synthesis. In *ECCV*.

Wang, Z.; Wu, S.; Xie, W.; Chen, M.; and Prisacariu, V. A. 2021a. NeRF–: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*.

Wang, Z.; Wu, S.; Xie, W.; Chen, M.; and Prisacariu, V. A. 2021b. NeRF–: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*.

Wu, C.; Duan, Y.; Zhang, X.; Sheng, Y.; Ji, J.; and Zhang, Y. 2024. MM-Gaussian: 3D Gaussian-based Multi-modal Fusion for Localization and Reconstruction in Unbounded Scenes. *arXiv preprint arXiv:2404.04026*.

Xu, Q.; Xu, Z.; Philip, J.; Bi, S.; Shu, Z.; Sunkavalli, K.; and Neumann, U. 2022. Point-nerf: Point-based neural radiance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5438–5448.

Yang, Z.; Chen, Y.; Wang, J.; Manivasagam, S.; Ma, W.-C.; Yang, A. J.; and Urtasun, R. 2023. Unisim: A neural closed-loop sensor simulator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1389–1399.

Ye, T.; Wu, Q.; Deng, J.; Liu, G.; Liu, L.; Xia, S.; Pang, L.; Yu, W.; and Pei, L. 2024. Thermal-NeRF: Neural Radiance Fields from an Infrared Camera. *arXiv preprint arXiv:2403.10340*.

Yu, A.; Li, R.; Tancik, M.; Li, H.; Ng, R.; and Kanazawa, A. 2021. PlenOctrees for Real-time Rendering of Neural Radiance Fields. In *arXiv*.

Zhang, K.; Riegler, G.; Snavely, N.; and Koltun, V. 2020. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*.

Zhou, X.; Lin, Z.; Shan, X.; Wang, Y.; Sun, D.; and Yang, M.-H. 2023. Drivinggaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes. *arXiv preprint arXiv:2312.07920*.

Zwicker, M.; Pfister, H.; Van Baar, J.; and Gross, M. 2001. EWA volume splatting. In *Proceedings Visualization, 2001. VIS'01.*, 29–538. IEEE.