PaD-GS: Leveraging Distortion Map for Panoramic Gaussian Splatting

Yihang Xu[1,2] and Qiulei Dong⋆[1,2]

[1] School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, 100049, China
[2] State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China
xuyihang2024@ia.ac.cn, qldong@nlpr.ia.ac.cn

Abstract. Recently, 3D Gaussian Splatting (3DGS), which succeeded in perspective image rendering, has been extended to handle the panoramic image rendering task. However, the performance of the existing methods in literature is generally limited due to the severe distortions involved in panoramic images. To address this problem, we construct a distortion map under the panoramic imaging model, where the value of each pixel reflects its corresponding distortion degree, and accordingly, we propose a novel ***Pa***noramic ***G***aussian ***S***platting method by utilizing this ***D***istortion map, called PaD-GS. The proposed PaD-GS, consisting of a distortion-aware Gaussian decoding module and a distortion-aware opacity modulation module, employs a whole-to-partial strategy to impose the distortion map on Gaussian representation learning. Specifically, the distortion-aware Gaussian decoding module is designed to decode the ensemble of learnable scene features and the distortion map into a set of Gaussian representations, so that the intrinsic distortion information of the distortion map could be injected into the whole Gaussian representations. The distortion-aware opacity modulation module is designed to further impose the distortion map by adaptively modulating the partial attribute (opacity) of each Gaussian representation. Thanks to the introduced distortion map, the proposed PaD-GS could alleviate the negative influence of severe distortion involved in panoramic images. Extensive experimental results on two panoramic datasets demonstrate that PaD-GS significantly outperforms several state-of-the-art methods for panoramic image rendering in most cases. The code is available at https://github.com/CosyXu/PaD-GS.

Keywords: 3D Gaussian Splatting · Panoramic Image Rendering · Distortion Map

1 Introduction

Novel view rendering for panoramic images is a challenging technique in computer vision, which could be widely applied to various scenarios, such as vir-

⋆ Corresponding author

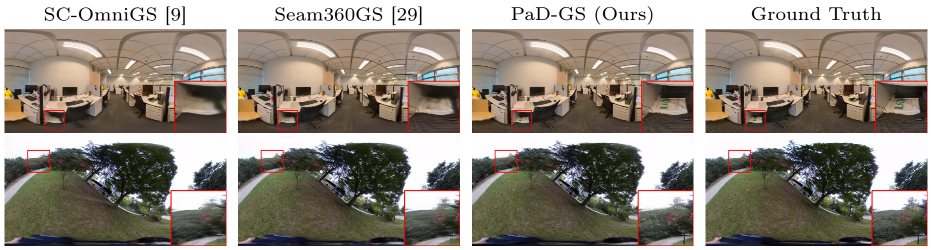

Fig. 1: Visualization results of SC-OmniGS [9], Seam360GS [29], and the proposed PaD-GS on the 360Roam [10] and Ricoh360 [4] datasets. For each visualization result of the methods, the sub-figure at the bottom right corner is the enlarged region corresponding to the highlighted red rectangle of the rendered image.

tual reality [31, 37, 38], autonomous navigation [17, 32, 35], and 3D reconstruction [12,27,34]. The existing methods in literature could be roughly divided into two categories: NeRF (Neural Radiance Fields [22])-based methods and 3DGS (3D Gaussian Splatting [14])-based methods. Since the NeRF-based methods for panoramic image rendering generally require time-consuming training and inference as indicated in [16, 19], the 3DGS-based methods for panoramic image rendering have attracted much attention in recent years.

In fact, 3DGS was originally designed for handling the perspective image rendering task [2,3,8,14,21,39]. In theory, it could be extended to handle panoramic images by integrating an equirectangular projection function into the 3DGS rasterization pipeline [16, 19]. However, the performance of such a straightforward extension is generally limited, mainly due to the severe distortions introduced by the equirectangular projection as verified in [13, 28]. To alleviate this problem, several recent works [9, 29] attempted to handle panoramic distortions by jointly optimizing camera parameters and Gaussian representations. However, the distortion problem still haunts these methods, as illustrated in Figure 1.

To address the distortion problem, we construct a distortion map under the panoramic imaging model, where each pixel encodes its local distortion degree. Based on this ***D***istortion map, we propose a novel ***Pa***noramic ***G***aussian ***S***platting method, called PaD-GS, which incorporates distortion information into the Gaussian representation learning in a whole-to-partial manner. The proposed PaD-GS consists of two modules: a distortion-aware Gaussian decoding module and a distortion-aware opacity modulation module. The distortion-aware Gaussian decoding module is designed to leverage the distortion map together with the learnable scene features to generate Gaussian representations at a whole level. Then, considering that panoramic distortions vary significantly across the image, which leads to non-uniform influence on the Gaussian representation learning, the Gaussians located in highly distorted regions may introduce unreliable contributions during opacity accumulation for rendering. To alleviate this problem, the distortion-aware opacity modulation module is designed to further incorporate the distortion map at a partial level, adaptively adjusting each

Gaussian's opacity according to its associated distortion degree. By integrating the distortion map for the whole Gaussian generation and the partial Gaussian refinement, PaD-GS could effectively endow the Gaussian representations with the anti-distortion ability for improved panoramic image rendering.

In sum, our main contributions include:

- We introduce the distortion map that is constructed under the panoramic imaging model. This distortion map could reflect the distortion degrees of different pixels of a panoramic image effectively. To our best knowledge, this work is the first attempt to bring the distortion map into Gaussian representation learning, which might give some insights into the 3DGS technique.
- We design two modules to impose the distortion map on Gaussian representation learning, including the distortion-aware Gaussian decoding module for learning a whole Gaussian representation and the distortion-aware opacity modulation module for refining the partial attribute (opacity). Accordingly, the Gaussian representations are endowed with the anti-distortion ability.
- We propose PaD-GS for panoramic image rendering by integrating the above two modules under a whole-to-partial strategy. Its effectiveness has been verified by the results in Section 4.

2 Related Work

This section provides reviews on the existing NeRF-based panoramic image rendering methods and 3DGS-based panoramic image rendering methods respectively.

2.1 NeRF-Based Panoramic Image Rendering

In recent years, Neural Radiance Fields (NeRF) [22] have emerged as a popular technique for novel view rendering, and a number of works have extended this technique to panoramic images. Huang et al. [10] proposed a geometry-aware omnidirectional radiance field framework with adaptively assigned local radiance fields for rendering indoor panoramic roaming scenes. Choi et al. [4] proposed a spherical-grid-based NeRF framework for large-scale egocentric panoramic video reconstruction, which adopted a quasi-uniform spherical parameterization with balanced angular grids to efficiently model unbounded outdoor scenes. Kulkarni et al. [15] proposed a semi-supervised NeRF framework for novel view synthesis from a single panoramic image, which incorporated geometric supervision via depth reprojection and semantic consistency constraints based on features extracted from CLIP [24] to progressively guide training. Wang et al. [30] further proposed PERF, a panoramic NeRF framework trained from a single panorama, which was guided by Stable Diffusion [25] and a monocular depth estimator to maintain appearance and geometric consistency.

It is noted that although the above NeRF-based methods could handle the panoramic image rendering task effectively, both their training and rendering costs are generally high due to the intrinsic architecture of NeRF.

2.2 3DGS-Based Panoramic Image Rendering

In comparison to NeRF, 3D Gaussian Splatting (3DGS) [14] has shown its competitive accuracy and much faster speed for panoramic image rendering. For example, Huang et al. [11] proposed a projection-optimized 3D Gaussian Splatting framework, which introduced an optimal Gaussian projection strategy, enabling improved rendering performance under camera models with large fields of view (e.g., omnidirectional cameras). Bai et al. [1] proposed a panoramic 3D Gaussian Splatting framework for panoramic images, which adopted a two-stage Gaussian projection strategy to model panoramic geometry and further incorporated indoor layout priors to enhance performance in texture-less regions of indoor scenes. Lee et al. [16] further proposed a geometrically grounded omnidirectional Gaussian splatting rasterization pipeline to achieve more effective and efficient panoramic image rendering. Li et al. [19] proposed an improved omnidirectional Gaussian splatting framework with a GPU-accelerated rasterizer that splatted 3D Gaussians directly onto the equirectangular image plane, enabling more straightforward differentiable optimization for panoramic image rendering. More recently, Li et al. [18] proposed SPaGS, which combined explicit 3D Gaussian representations with ray-casting-based rendering and introduced axis-aligned bounding box computation for panoramic images to facilitate spherical-geometry-aligned panoramic image rendering. In addition, Huang et al. [9] and Shin et al. [29] introduced camera-model-aware 3D Gaussian splatting frameworks that jointly optimized 3D Gaussians and omnidirectional camera parameters to handle distortions in panoramic images.

Unlike the above methods that generally focus on optimizing panoramic-based Gaussian projections or adjusting camera parameters, the proposed method leverages a constructed distortion map to guide 3D Gaussian learning in a whole-to-partial manner.

3 Method

In this section, the proposed PaD-GS is introduced in detail. First, the preliminary of the panoramic 3DGS is given. Then, the architecture of PaD-GS is described. Next, the construction process of the distortion map is explained, and the two modules in PaD-GS are introduced. Finally, the loss function is given.

3.1 Preliminary: Panoramic 3DGS

3DGS [14] explicitly represents scenes using a set of 3D Gaussian ellipsoids, which are initialized by the sparse point cloud obtained from existing SfM (Structure-from-Motion) methods [5–7, 20, 23, 26, 36]. Each 3D Gaussian, G_i, is parameterized by the 3D center position $\mu_i \in \mathbb{R}^3$ and the 3D covariance matrix $\Sigma_i \in \mathbb{R}^{3\times3}$ in the world coordinate system as follows:

$$G_i(x) = e^{-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i)}, \tag{1}$$

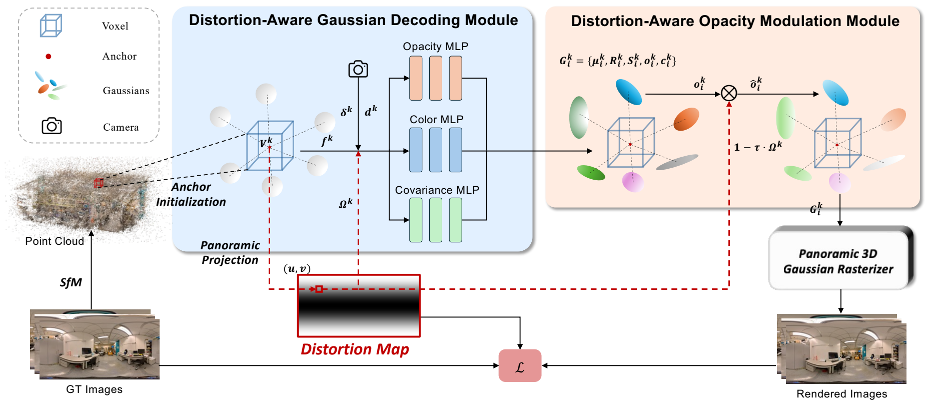

Fig. 2: Architecture of PaD-GS. PaD-GS consists of a distortion-aware Gaussian decoding module and a distortion-aware opacity modulation module. During training, anchors are first initialized from the point cloud obtained via the SfM method [23], which are then decoded with the distortion map in the distortion-aware Gaussian decoding module to generate whole 3D Gaussian representations. These Gaussians are then passed into the distortion-aware opacity modulation module for extended distortion-aware Gaussian representations with refined opacities. Finally, the resulting Gaussians are fed into the panoramic 3D Gaussian rasterizer for image rendering.

where Σ_i is typically decomposed as follows:

$$\Sigma_i = R_i S_i S_i^T R_i^T, \tag{2}$$

where $S_i \in \mathbb{R}^3$ denotes a scaling matrix and $R_i \in SO(3)$ denotes a rotation matrix. In addition, each 3D Gaussian also includes the opacity $o_i \in (0,1)$ and the Spherical Harmonic (SH) coefficients $sh_i \in \mathbb{R}^L$ (L denotes the SH degree) to model its transparency and view-dependent color.

For the rendering of panoramic images, 3D Gaussians are projected onto the 2D equirectangular image plane under the panoramic imaging model. Given the viewing transformation matrix W and the Jacobian matrix J, the 2D covariance matrix Σ'_i and the 2D center position μ'_i are formed as:

$$\Sigma'_i = JW\Sigma_i W^T J^T, \qquad \mu'_i = JW\mu_i, \tag{3}$$

where the Jacobian matrix J is the affine approximation of the panoramic projection function that maps 3D points to their longitude–latitude coordinates on the unit sphere, followed by the equirectangular mapping. These projected 2D Gaussians are then α-blended to compute the final color $C(p)$ for each pixel as follows:

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

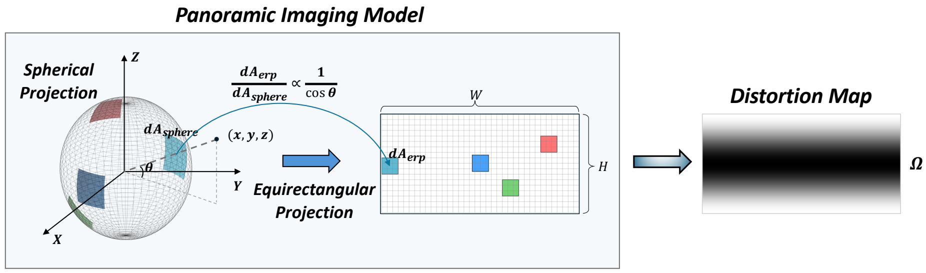

Fig. 3: Illustration of the distortion map construction. The distortion map is derived from the panoramic imaging model, combining spherical projection and equirectangular projection, where each pixel's value encodes the distortion degree according to its corresponding polar angle θ.

where $\mathcal{N}$ is the set of Gaussians overlapping the pixel, c_i is the view-dependent color, and the screen-space opacity α_i is defined as:

$$\alpha_i = o_i e^{-\frac{1}{2}(y-\mu_i')^T \Sigma_i'^{-1}(y-\mu_i')}, \tag{5}$$

where y represents the 2D equirectangular coordinates of the Gaussian.

3.2 Architecture

The architecture of the proposed PaD-GS is shown in Figure 2. PaD-GS takes panoramic images captured in a same scene as input, consisting of a distortion-aware Gaussian decoding module and a distortion-aware opacity modulation module. The two modules are designed to generate and refine Gaussian representations respectively by leveraging the constructed distortion map in a whole-to-partial manner.

At the training stage, the scene is first voxelized from a reconstructed sparse point cloud obtained by the SfM method [23]. The center of each voxel is initialized as an anchor, which is equipped with learnable parameters to spawn 3D Gaussians. Then, the initialized anchors together with the distortion map and relative viewpoint are jointly fed into the distortion-aware Gaussian decoding module, which outputs the corresponding Gaussian representations of each anchor via three MLPs. Next, the distortion-aware opacity modulation module leverages the distortion map to further refine these Gaussian representations and adaptively modulates their opacities according to their distortion degrees, outputting extended distortion-aware Gaussian representations. Finally, the extended Gaussian representations are passed into the panoramic 3D Gaussian rasterizer [19] for image rendering, where the panoramic projection function is employed for splatting Gaussians onto the panoramic image.

In the following subsections, we detail the construction of the distortion map and the two modules of PaD-GS.

3.3 Distortion Map Construction

To explore an effective encoding method for distortions in panoramic images that can be incorporated into the Gaussian learning process, we construct a distortion map directly from the panoramic imaging geometry, as illustrated in Figure 3.

In the panoramic imaging model, a 3D point $(x, y, z) \in \mathbb{R}^3$ defined in the camera coordinate system is first projected onto a unit sphere centered at the camera. The corresponding azimuth angle $\phi \in [-\pi, \pi]$ and polar angle $\theta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ are formulated as:

$$\phi = \arctan(y, -x), \qquad \theta = \arctan\left(z, \sqrt{x^2 + y^2}\right), \tag{6}$$

where $\arctan(\cdot, \cdot)$ denotes the inverse tangent function. Then, the spherical coordinates (ϕ, θ) are mapped onto the panoramic image plane via the equirectangular projection, and the corresponding pixel coordinates (u, v) are formulated as:

$$u = \frac{\phi + \pi}{2\pi} W, \qquad v = \frac{\theta + \frac{\pi}{2}}{\pi} H, \tag{7}$$

where W and H are the image width and height respectively.

Under the spherical parameterization of the unit sphere defined in Equation (6), points with a fixed polar angle θ lie on a latitude circle of radius $\cos \theta$. A small angular variation $(d\phi, d\theta)$ thus corresponds to a spherical patch whose side lengths are $\cos \theta \, d\phi$ (along longitude) and $d\theta$ (along latitude). Therefore, the spherical surface element dA_{sphere} is formulated as:

$$dA_{\text{sphere}} = \cos \theta \, d\phi \, d\theta. \tag{8}$$

Then, the corresponding side lengths du and dv on the panoramic image plane could be obtained from Equation (7):

$$du = \frac{W}{2\pi} d\phi, \qquad dv = \frac{H}{\pi} d\theta. \tag{9}$$

Accordingly, the resulting image area element dA_{erp} is formulated as:

$$dA_{\text{erp}} = du \, dv = \frac{WH}{2\pi^2} \, d\phi \, d\theta. \tag{10}$$

Finally, the ratio between the two area elements is computed as:

$$\frac{dA_{\text{erp}}}{dA_{\text{sphere}}} = \frac{WH}{2\pi^2} \cdot \frac{1}{\cos \theta}. \tag{11}$$

Thus, the local area distortion introduced by equirectangular projection is proportional to $\frac{1}{\cos \theta}$, indicating that larger polar angles correspond to greater distortions. Based on this geometric property, we normalize the polar angle θ to the range $[0, 1]$, where 0 corresponds to $\theta = 0$ (no distortion) and 1 corresponds to $\theta = \pm\frac{\pi}{2}$ (severe distortion), and construct the distortion map $\Omega \in \mathbb{R}^{H\times W}$ according to Equation (7):

$$\Omega(u,v) = 1 - \cos^2\theta = 1 - \cos^2\left(\frac{\pi}{H}v - \frac{\pi}{2}\right). \tag{12}$$

It could be noted from Equation (12) that the gray value of each pixel reflects its corresponding distortion degree.

3.4 Distortion-Aware Gaussian Decoding Module

After obtaining the distortion map, the distortion-aware Gaussian decoding module is designed to incorporate distortion information into the generation of each anchor's whole Gaussian representations, enabling distortion-aware modeling of the learned Gaussians. It takes each anchor as input and outputs a set of 3D Gaussian representations corresponding to the anchor via the MLPs.

First, to learn the scene representation, each anchor V^k is associated with a learnable feature $f^k \in \mathbb{R}^C$ (which is used to encode local appearance information), and a scaling factor $l^k \in \mathbb{R}^3$ along with N learnable offsets $\{P_i^k \in \mathbb{R}^3\}_{i=1}^N$ (which are used to spawn a fixed number of Gaussians around the anchor). For each anchor V^k centered at X^k, its corresponding coordinate on the panoramic image plane is computed according to the panoramic imaging model defined in Equations (6) and (7), and the corresponding distortion degree Ω^k is then obtained from the distortion map and assigned to the anchor according to Equation (12).

Subsequently, the anchor feature f^k, together with its distortion degree Ω^k, relative distance to the camera center $\delta^k \in \mathbb{R}$, and viewing direction $d^k \in \mathbb{R}^3$, are fed into three MLPs to decode the Gaussian attributes, including opacities $\{o_i^k\}_{i=1}^N$, colors $\{c_i^k\}_{i=1}^N$, and covariances $\{\Sigma_i^k\}_{i=1}^N$ (parameterized by scalings $\{S_i^k\}_{i=1}^N$ and rotations $\{R_i^k\}_{i=1}^N$). The process is formulated as:

$$\{o_i^k, c_i^k, S_i^k, R_i^k\}_{i=1}^N = \mathcal{F}_{\{\text{opa},\text{col},\text{cov}\}}\left(f^k, \Omega^k, \delta^k, d^k\right), \tag{13}$$

where $\mathcal{F}_{\{\text{opa},\text{col},\text{cov}\}}(\cdot)$ denotes the opacity, color, and covariance MLPs respectively. In addition, the centers of the Gaussian representations $\{\mu_i^k\}_{i=1}^N$ are formulated as:

$$\mu_i^k = X^k + l^k \cdot P_i^k, \tag{14}$$

where l^k denotes the scaling factor.

Through this distortion-aware decoding process, the distortion degree derived from the distortion map is incorporated into the generation of each anchor's whole Gaussian representations, endowing the learned Gaussians with the distortion-aware ability during their formation and optimization.

3.5 Distortion-Aware Opacity Modulation Module

Considering that panoramic distortions vary significantly across the image, which results in the inherently non-uniform influence on the learned Gaussians, Gaussians located in highly distorted regions may introduce unreliable contributions for rendering. To alleviate this problem, we design a distortion-aware opacity modulation module to further refine the Gaussian representations at the partial level by integrating the distortion information. Specifically, this module adaptively adjusts the opacity of each Gaussian according to its distortion degree to effectively suppress the over-contribution of Gaussians in highly distorted regions during opacity accumulation. It takes the Gaussian representations generated by the distortion-aware Gaussian decoding module as input, and outputs the extended distortion-aware Gaussian representations with refined opacities.

For the modulation of the opacities of Gaussian representations, for regions with small distortion, the decoded opacity should be preserved as much as possible to maintain the original representation learned from the whole-attribute-based Gaussian decoding, while for regions with large distortion, the opacity should be appropriately suppressed to mitigate the influence of unreliable Gaussians during rendering. Accordingly, the modulated opacity $\hat{o}_i^k$ of Gaussian G_i^k is then computed as:

$$\hat{o}_i^k = o_i^k \cdot (1 - \tau \cdot \Omega^k), \tag{15}$$

where τ is a tuning parameter that controls the strength of distortion-aware modulation. Through this partial adjustment, the 3D Gaussian representations are endowed with the anti-distortion ability, which could help suppress rendering artifacts in highly distorted regions while preserving fine details in less distorted regions.

3.6 Loss Function

The loss function $\mathcal{L}$ of the proposed PaD-GS is defined as a weighted sum of the L_1 loss and the SSIM loss [33], which measures the pixel-level errors between the rendered and ground-truth images:

$$\mathcal{L} = \lambda \| M \odot (\hat{I} - I) \|_1 + (1 - \lambda) \| M \odot (1 - \mathrm{SSIM}(\hat{I}, I)) \|_1, \tag{16}$$

where I is the ground-truth image, $\hat{I}$ is the rendered image, λ is a tuning parameter, "$\odot$" is the Hadamard product, $M \in \mathbb{R}^{H \times W}$ is the distortion-aware weight map for ensuring that each pixel contributes proportionally to its corresponding spherical surface area, which is defined as:

$$M(u, v) = \cos\theta = \cos\left(\frac{\pi}{H} v - \frac{\pi}{2}\right), \tag{17}$$

where θ denotes the polar angle in the spherical coordinate system corresponding to the pixel (u, v), which can be obtained from the inverse equirectangular projection in Equation (7).

4 Experiments

4.1 Datasets and Evaluation Metrics

Datasets As done in [16] and [19], we use the 360Roam dataset [10] and the Ricoh360 dataset [4] for evaluation. The 360Roam dataset [10] consists of 11

Table 1: Comparison on the 11 indoor scenes of the 360Roam dataset [10]. ↓ / ↑ denotes that lower / higher is better. The best results are in **bold** for each metric.

Scene	EgoNeRF [4]			OP43DGS [11]			ODGS [16]			OmniGS [19]			SPaGS [18]			PaD-GS (Ours)		
	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓
Bar	0.663	20.16	0.448	0.750	21.53	0.324	0.702	20.17	0.373	0.783	22.24	0.260	0.767	21.56	0.279	**0.808**	**23.40**	**0.230**
Base	0.652	22.30	0.481	0.766	24.17	0.296	0.716	22.98	0.359	0.797	24.79	0.223	0.783	24.67	0.260	**0.823**	**25.60**	**0.198**
Café	0.730	22.92	0.405	0.806	24.27	0.277	0.763	23.04	0.322	0.829	24.97	0.217	0.821	24.74	0.233	**0.845**	**25.64**	**0.193**
Canteen	0.683	21.22	0.488	0.726	21.77	0.388	0.691	20.71	0.417	0.747	22.20	0.335	0.718	21.55	0.349	**0.754**	**22.84**	**0.286**
Center	0.737	23.49	0.444	0.780	24.37	0.337	0.745	23.50	0.391	0.795	24.89	0.292	0.771	24.32	0.331	**0.809**	**25.69**	**0.265**
Center-1	0.763	23.53	0.446	0.808	25.00	0.335	0.773	23.79	0.391	0.825	25.71	0.287	0.801	25.19	0.332	**0.839**	**26.47**	**0.260**
Corridor	0.808	27.35	0.383	0.813	25.56	0.343	0.774	24.55	0.403	0.807	24.30	0.319	**0.833**	**27.55**	0.270	0.832	26.94	**0.247**
Innovation	0.703	24.23	0.461	0.769	25.24	0.330	0.735	23.20	0.364	0.795	25.68	0.259	0.773	25.29	0.311	**0.827**	**26.79**	**0.218**
Lab	0.816	25.73	0.355	0.869	27.00	0.241	0.839	25.76	0.290	0.888	27.99	0.188	0.872	27.67	0.228	**0.902**	**29.45**	**0.162**
Library	0.686	25.13	0.459	0.721	25.51	0.364	0.689	24.20	0.415	0.741	25.82	0.322	0.710	24.98	0.325	**0.748**	**26.04**	**0.251**
Office	0.731	24.62	0.413	0.759	24.47	0.339	0.689	21.88	0.442	0.781	24.65	0.288	0.819	26.33	0.245	**0.846**	**27.35**	**0.184**
Average	0.725	23.70	0.435	0.779	24.44	0.325	0.738	23.07	0.379	0.799	24.84	0.272	0.788	24.90	0.288	**0.821**	**26.02**	**0.227**

OmniGS [19] SPaGS [18] PaD-GS (Ours) Ground Truth

Fig. 4: Visualization results of OmniGS [19], SPaGS [18], and the proposed PaD-GS on the 360Roam dataset [10].

real-world indoor scenes captured by an omnidirectional camera mounted on a roaming robot with an original image size 6080 × 3040, and we resize them to 2048 × 1024 for both training and testing following [16]. The Ricoh360 dataset [4] consists of 12 real-world outdoor scenes captured by rotating the camera in place along a cross-shaped trajectory, with an original image size 1920 × 960 used for both training and testing following [16].

Evaluation Metrics As done in [14, 18, 19], the structural similarity index (SSIM) [33], the peak signal-to-noise ratio (PSNR), and the learned perceptual image patch similarity (LPIPS) [40] metrics are used for evaluation.

4.2 Implementation Details

Our method is implemented in PyTorch and trained on a single NVIDIA RTX 3090 Ti GPU using the Adam optimizer. At the training stage, the model is trained for 30,000 iterations for each scene on both datasets, and Gaussian densification and pruning are performed every 100 iterations between iterations 1500 and 15,000. It is noted that the bottom part of the images in the 360Roam dataset is masked during both training and testing for all methods following [19], since the base of the roaming robot is a dynamic object. The voxel size is set to 0.001 for the indoor 360Roam dataset [10] and 0.005 for the outdoor Ricoh360 dataset [4] in anchor initialization. The feature dimension is set to $C = 32$, and

Table 2: Comparison on the 12 outdoor scenes of the Ricoh360 dataset [4]. "OOM" denotes an Out-Of-Memory error.

Scene	EgoNeRF [4]			OP43DGS [11]			ODGS [16]			OmniGS [19]			SPaGS [18]			PaD-GS (Ours)		
	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓
Bricks	0.751	23.44	0.269	OOM	OOM	OOM	0.767	22.61	0.266	0.838	24.52	0.184	0.847	24.77	0.167	**0.858**	**25.25**	**0.163**
Bridge	0.737	23.48	0.295	OOM	OOM	OOM	0.723	22.06	0.318	0.785	23.82	0.236	0.806	23.98	0.203	**0.812**	**24.11**	**0.198**
Bridge-under	0.793	24.92	0.268	OOM	OOM	OOM	0.796	24.16	0.253	0.872	26.58	0.151	0.879	26.47	0.135	**0.890**	**26.98**	**0.124**
Cat-tower	0.700	24.28	0.330	0.632	16.81	0.425	0.707	22.86	0.307	0.774	24.82	0.228	0.782	25.08	0.215	**0.812**	**25.75**	**0.187**
Center	0.859	28.54	0.248	0.745	18.89	0.438	0.775	23.04	0.428	0.885	29.14	0.198	0.901	29.48	0.169	**0.905**	**29.94**	**0.167**
Farm	0.664	22.21	0.327	OOM	OOM	OOM	0.672	21.00	0.317	0.732	22.38	0.251	**0.749**	**22.48**	**0.227**	0.743	22.41	0.232
Flower	0.634	21.82	0.368	OOM	OOM	OOM	0.618	18.73	0.374	0.727	22.38	0.265	0.734	22.60	0.244	**0.739**	**22.67**	**0.240**
Gallery-chair	0.847	27.77	0.278	0.804	23.63	0.391	0.834	26.08	0.312	0.885	28.35	0.195	0.894	28.59	0.165	**0.905**	**29.26**	**0.157**
Gallery-park	0.775	25.47	0.301	0.735	22.55	0.409	0.749	21.69	0.350	0.821	26.02	0.227	0.831	26.39	0.209	**0.851**	**26.87**	**0.189**
Gallery-pillar	0.846	27.98	0.252	OOM	OOM	OOM	0.814	22.90	0.305	0.883	28.56	0.194	0.894	29.14	0.177	**0.902**	**29.19**	**0.175**
Garden	0.727	26.86	0.307	OOM	OOM	OOM	0.734	23.70	0.299	0.795	26.99	0.227	0.806	**27.55**	0.212	**0.823**	27.49	**0.202**
Poster	0.855	26.63	0.265	OOM	OOM	OOM	0.837	25.04	0.301	0.898	28.40	0.184	0.896	27.83	0.177	**0.905**	**28.58**	**0.172**
Average	0.766	25.28	0.292	–	–	–	0.752	22.82	0.319	0.825	26.00	0.212	0.835	26.20	0.192	**0.845**	**26.54**	**0.184**

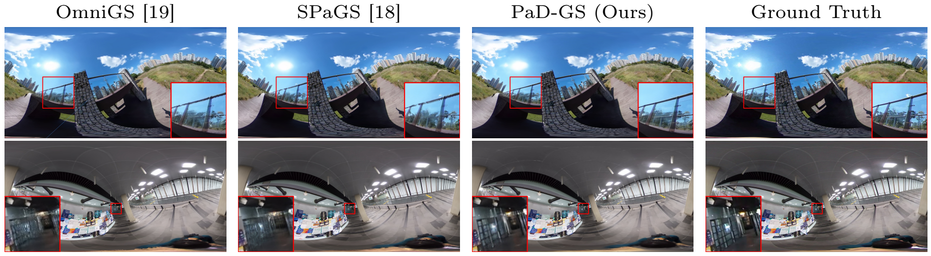

Fig. 5: Visualization results of OmniGS [19], SPaGS [18], and the proposed PaD-GS on the Ricoh360 dataset [4].

the number of Gaussians attached to each anchor is set to $N = 10$. The tuning parameters are set to $\tau = 0.9$ and $\lambda = 0.8$.

4.3 Comparative Evaluation

Here, we evaluate the proposed PaD-GS on the 360Roam [10] and Ricoh360 [4] datasets in comparison to 5 state-of-the-art methods for panoramic image rendering, including EgoNeRF [4], OP43DGS [11], ODGS [16], OmniGS [19], and SPaGS [18]. The results of all comparative methods are reproduced using the authors' publicly released codes.

Results on 360Roam We first evaluate the proposed PaD-GS on the 11 indoor scenes of the 360Roam dataset [10] in comparison to 5 state-of-the-art methods, and the corresponding results are reported in Table 1. Four points can be observed from this table: (i) The NeRF-based method EgoNeRF [4] achieves suboptimal performance on the indoor roaming scenes, mainly because it is originally designed for egocentric scene reconstruction. (ii) Methods that focus on directly optimizing panoramic-based Gaussian projections, such as OP43DGS [11] and ODGS [16], achieve limited performance on most indoor scenes, indicating that simply extending Gaussian projection to handle panoramic images is insufficient to effectively address the distortion effects. (iii) By introducing more accurate

Table 3: Ablation study of the main components of the proposed PaD-GS on the 360Roam dataset [10] (Bar) and the Ricoh360 dataset [4] (Cat-tower). "DAGD" denotes the distortion-aware Gaussian decoding module, "DAOM" denotes the distortion-aware opacity modulation module, and M denotes the distortion-aware weight map defined in Equation (17).

Method	360Roam: Bar			Ricoh360: Cat-tower		
	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓
Baseline	0.786	22.29	0.257	0.776	24.92	0.226
Baseline + DAGD	0.801	23.13	0.240	0.797	25.36	0.208
Baseline + DAGD + DAOM (w/o. M)	0.804	23.24	0.245	0.808	25.60	0.194
Baseline + DAGD + DAOM (Ours)	**0.808**	**23.40**	**0.230**	**0.812**	**25.75**	**0.187**

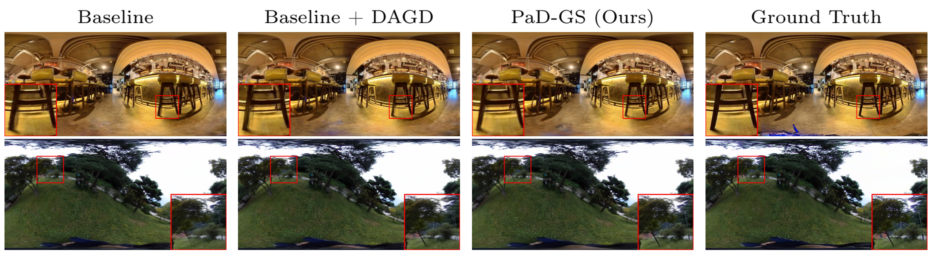

Fig. 6: Visualization results by the main components of the proposed PaD-GS on the 360Roam [10] and Ricoh360 datasets [10].

Gaussian modeling in panoramic image rendering, OmniGS [19] and SPaGS [18] achieve relatively better performance across several scenes. However, their improvements are still constrained and not consistently optimal under all metrics, mainly because they rely on local projection adaptation of Gaussian representations without incorporating the panoramic distortion into the Gaussian learning process. (iv) The proposed PaD-GS achieves the best performance among all comparative methods on most scenes as well as in the average results, demonstrating its effectiveness in handling panoramic image rendering.

In addition, Figure 4 shows two visualization results of the proposed PaD-GS as well as two comparative methods that perform relatively better than others, including OmniGS [19] and SPaGS [18]. As seen from this figure, the local regions in the highlighted red rectangles of the ground-truth images in the last column are structurally consistent and contain clear texture details, but the corresponding local regions synthesized by OmniGS and SPaGS suffer from noticeable blurring and color inconsistencies, as shown in the first and second columns. In contrast, the proposed PaD-GS not only preserves sharper boundaries but also maintains better color consistency with respect to the ground-truth image, as shown in the third column.

Results on Ricoh360 We also evaluate the proposed PaD-GS on the 12 outdoor scenes of the Ricoh360 dataset [4], and the corresponding results are reported in Table 2. Two points can also be observed from this table: (i) OP43DGS

[11] encounters Out-Of-Memory errors on most outdoor scenes. This is because outdoor panoramic scenes usually involve larger spatial scales and more complex geometry, which significantly increases memory consumption when iteratively optimizing its specially designed panoramic Gaussian projection. (ii) The proposed PaD-GS performs better than other comparative methods on most scenes as well as in the average results, consistent with the results observed on the 360Roam dataset [10]. These results further confirm that PaD-GS could effectively handle panoramic image rendering in both indoor and outdoor scenes.

Additionally, Figure 5 shows another two visualization results of the proposed PaD-GS as well as OmniGS [19] and SPaGS [18]. From the highlighted regions, it can be observed that the ground-truth images (Column 4) exhibit consistent structures. In comparison, OmniGS and SPaGS (Columns 1 and 2) tend to produce rendering results with softened edges and disrupted structural continuity, especially in highly distorted outdoor regions. In contrast, PaD-GS (Column 3) produces more faithful geometric structures consistent with the ground-truth image, even under severe panoramic distortions. Please see the supplemental material for more comparative evaluation.

4.4 Ablation Studies

To verify the effectiveness of each key element in the proposed PaD-GS, we conduct ablation studies on the 360Roam dataset [10] (Bar) and the Ricoh360 dataset [4] (Cat-tower). The proposed PaD-GS consists of two modules, including the distortion-aware Gaussian decoding (denoted as “DAGD”) module and the distortion-aware opacity modulation (denoted as “DAOM”) module. Here, the panoramic 3DGS [19] serves as the baseline, and we evaluate the proposed PaD-GS by adding the DAGD module and the DAOM module to the baseline respectively. In addition, we further evaluate PaD-GS without the distortion-aware weight map M defined in Equation (17) to verify its effectiveness. The corresponding results are reported in Table 3.

As seen from this table, four points can be revealed: (i) The results reported in Rows 1 and 2 show that introducing the DAGD module consistently improves the rendering performance on both datasets, demonstrating that incorporating the distortion map into the whole Gaussian generation helps the model better adapt to panoramic distortion. (ii) The results reported in Rows 2 and 4 show that further introducing the DAOM module brings additional improvements, indicating that using the distortion map to guide partial Gaussian attribute refinement helps produce more reliable Gaussian opacity during rendering. (iii) The results reported in Rows 3 and 4 show that incorporating the distortion-aware weight map M leads to performance improvement, suggesting that M helps guide the distortion-aware learning process by assigning different weights to regions with varying distortion degrees. (iv) The proposed PaD-GS achieves the best performance by incorporating both modules and the distortion-aware weight map, demonstrating its effectiveness.

To further verify the effectiveness of the designed modules in PaD-GS, several visualization results are given in Figure 6. As seen from this figure, the baseline

Table 4: Comparison of the average training time (in hours) and rendering speed (in FPS) of the comparative methods on the 360Roam [10] and Ricoh360 [4] datasets.

Method	360Roam		Ricoh360	
	Training time (h)	FPS	Training time (h)	FPS
EgoNeRF [4]	4.72	0.05	5.38	0.04
OP43DGS [11]	2.32	30.24	–	–
ODGS [16]	1.40	21.32	1.69	18.44
OmniGS [19]	0.68	62.73	0.82	53.21
SPaGS [18]	0.43	83.11	0.55	78.89
PaD-GS (Ours)	1.35	29.87	1.66	23.12

model produces obvious artifacts for slender structures (e.g., chair legs and tree trunks), especially in highly distorted regions. The proposed DAGD module improves the structural consistency of these objects by incorporating distortion information into the whole Gaussian decoding process. The full PaD-GS further exhibits sharper edges and clearer structural details with the introduction of the DAOM module, which refines Gaussian attributes at the partial level to endow them with the anti-distortion ability. Please see the supplemental material for more ablation studies.

4.5 Efficiency Comparison and Limitation

Here, we report the average training time in GPU hours and rendering speed in FPS (frames-per-second) of PaD-GS and 5 comparative methods on the 360Roam [10] and Ricoh360 [4] datasets in Table 4. Four points can be seen from this table: (i) EgoNeRF [4] exhibits the longest training time and slowest rendering speed, due to the inherent inefficiency of its ray-casting-based rendering pipeline. (ii) OP43DGS [11] and ODGS [16] are relatively time-consuming in both training and rendering, mainly because these methods introduce redundant Gaussian presentations to compensate for panoramic distortions. (iii) OmniGS [19] and SPaGS [18] exhibit shorter training time and faster rendering speed, mainly because they adopt more efficient Gaussian projection and image rendering strategies. (iv) Our full PaD-GS shows moderate performance in both training and rendering efficiency, exhibiting a clear gap compared to OmniGS [19] and SPaGS [18], mainly because the MLP-based Gaussian generation introduces additional computational overhead during training and inference. However, when combining the results in Tables 1 and 2 with this table, it can be observed that PaD-GS achieves an effective balance among training time, rendering speed, and accuracy. It would be one of our future works to further reduce the computational complexity while maintaining high performance.

5 Conclusion

In this paper, we propose PaD-GS, a novel panoramic Gaussian Splatting method that leverages a distortion map constructed under the panoramic imaging model to reflect each pixel's distortion degree. The proposed PaD-GS employs a whole-to-partial strategy to impose the distortion map on Gaussian representation learning. To integrate distortion information at the whole level, we design a distortion-aware Gaussian decoding module that decodes the learnable scene features together with the distortion map into a set of Gaussian representations. Furthermore, to refine the partial Gaussian attribute at the spatial level for anti-distortion, we design a distortion-aware opacity modulation module that adaptively modulates the opacity of each Gaussian according to its associated distortion degree. By incorporating distortion information into both whole and partial Gaussian learning, the proposed PaD-GS could effectively alleviate the negative influence of severe panoramic distortions. Experimental results on two panoramic datasets demonstrate the effectiveness of the proposed PaD-GS.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant Nos. 62376269, 61991423, 62472262), Taishan Scholarship Construction Engineering, the Beijing Municipal Science and Technology Project (Grant No. Z211100011021004).

References

1. Bai, J., Huang, L., Guo, J., Gong, W., Li, Y., Guo, Y.: 360-gs: Layout-guided panoramic gaussian splatting for indoor roaming. In: International Conference on 3D Vision. pp. 1042–1053 (2025)
2. Charatan, D., Li, S.L., Tagliasacchi, A., Sitzmann, V.: Pixelsplat: 3D gaussian splats from image pairs for scalable generalizable 3D reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19457–19467 (2024)
3. Chen, Y., Xu, H., Zheng, C., Zhuang, B., Pollefeys, M., Geiger, A., Cham, T.J., Cai, J.: Mvsplat: Efficient 3D gaussian splatting from sparse multi-view images. In: Proceedings of the European Conference on Computer Vision. pp. 370–386 (2024)
4. Choi, C., Kim, S.M., Kim, Y.M.: Balanced spherical grid for egocentric view synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16590–16599 (2023)
5. Cui, H., Shen, S., Gao, W., Liu, H., Wang, Z.: Efficient and robust large-scale structure-from-motion via track selection and camera prioritization. ISPRS Journal of Photogrammetry and Remote Sensing **156**, 202–214 (2019)
6. Dong, Q., Deng, S., Liu, Y.: Cayley rotation averaging: Multiple camera averaging under the cayley framework. IEEE Transactions on Image Processing **33**, 4173–4187 (2024)
7. Dong, Q., Gao, X., Cui, H., Hu, Z.: Robust camera translation estimation via rank enforcement. IEEE Transactions on Cybernetics **52**(2), 862–872 (2020)
8. Hahlbohm, F., Friederichs, F., Weyrich, T., Franke, L., Kappel, M., Castillo, S., Stamminger, M., Eisemann, M., Magnor, M.: Efficient perspective-correct 3D gaussian splatting using hybrid transparency. In: Computer Graphics Forum. vol. 44, pp. 70014–70025 (2025)

9. Huang, H., Chen, Y., Li, L., Cheng, H., Braud, T., Zhao, Y., Yeung, S.K.: Sc-omnigs: Self-calibrating omnidirectional gaussian splatting. In: International Conference on Learning Representations (2025)
10. Huang, H., Chen, Y., Zhang, T., Yeung, S.K.: 360roam: Real-time indoor roaming using geometry-aware 360° radiance fields. arXiv preprint arXiv:2208.02705 (2022)
11. Huang, L., Bai, J., Guo, J., Li, Y., Guo, Y.: On the error analysis of 3D gaussian splatting and an optimal projection strategy. In: Proceedings of the European Conference on Computer Vision. pp. 247–263 (2024)
12. Huang, Z., He, J., Ye, J., Jiang, L., Li, W., Chen, Y., Han, T.: Scene4u: Hierarchical layered 3D scene reconstruction from single panoramic image for your immerse exploration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 26723–26733 (2025)
13. Ito, S., Takama, N., Ito, K., Chen, H.T., Aoki, T.: Erpgs: Equirectangular image rendering enhanced with 3D gaussian regularization. In: IEEE International Conference on Image Processing. pp. 2850–2855 (2025)
14. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3D gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics **42**(4), 139:1–139:14 (2023)
15. Kulkarni, S., Yin, P., Scherer, S.: 360fusionnerf: Panoramic neural radiance fields with joint guidance. In: 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 7202–7209 (2023)
16. Lee, S., Chung, J., Huh, J., Lee, K.M.: Odgs: 3D scene reconstruction from omnidirectional images with 3D gaussian splattings. Advances in Neural Information Processing Systems **37**, 57050–57075 (2024)
17. Li, J., Bansal, M.: Panogen: Text-conditioned panoramic environment generation for vision-and-language navigation. Advances in Neural Information Processing Systems **36**, 21878–21894 (2023)
18. Li, J., Hahlbohm, F., Scholz, T., Eisemann, M., Tauscher, J.P., Magnor, M.: Spags: Fast and accurate 3D gaussian splatting for spherical panoramas. In: Computer Graphics Forum. vol. 44, pp. 70171–70181 (2025)
19. Li, L., Huang, H., Yeung, S.K., Cheng, H.: Omnigs: Fast radiance field reconstruction using omnidirectional gaussian splatting. In: IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 2260–2268 (2025)
20. Liu, Y., Dong, Q.: Equipose: Exploiting permutation equivariance for relative camera pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1127–1137 (2025)
21. Lu, T., Yu, M., Xu, L., Xiangli, Y., Wang, L., Lin, D., Dai, B.: Scaffold-gs: Structured 3D gaussians for view-adaptive rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20654–20664 (2024)
22. 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)
23. Moulon, P., Monasse, P., Perrot, R., Marlet, R.: Openmvg: Open multiple view geometry. In: International Workshop on Reproducible Research in Pattern Recognition. pp. 60–74 (2016)
24. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning. pp. 8748–8763 (2021)

25. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10684–10695 (2022)
26. 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)
27. Sharma, S.K., Jain, K., Shukla, A.K.: 3D point cloud reconstruction using panoramic images. Applied Geomatics **16**(3), 575–592 (2024)
28. Shen, Z., Lin, C., Huang, S., Nie, L., Liao, K., Zhao, Y.: You need a transition plane: Bridging continuous panoramic 3D reconstruction with perspective gaussian splatting. arXiv preprint arXiv:2504.09062 (2025)
29. Shin, C., Cho, W.O., Kim, S.J.: Seam360gs: Seamless 360deg gaussian splatting from real-world omnidirectional images. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 28970–28979 (2025)
30. Wang, G., Wang, P., Chen, Z., Wang, W., Loy, C.C., Liu, Z.: Perf: Panoramic neural radiance field from a single panorama. IEEE Transactions on Pattern Analysis and Machine Intelligence **46**(10), 6905–6918 (2024)
31. Wang, Q., Li, W., Mou, C., Cheng, X., Zhang, J.: 360dvd: Controllable panorama video generation with 360-degree video diffusion model. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6913–6923 (2024)
32. Wang, S., Wang, Y., Fan, Z., Wang, Y., Chen, M., Wang, K., Su, Z., Li, W., Cai, X., Jin, Y., et al.: Monodream: Monocular vision-language navigation with panoramic dreaming. arXiv preprint arXiv:2508.02549 (2025)
33. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing **13**(4), 600–612 (2004)
34. Wen, Y., Qiao, Y., Lam, C.C., Brilakis, I., Lee, S., Wong, M.O.: Semantic bim enrichment for firefighting assets: Fire-art dataset and panoramic image-based 3D reconstruction. ISPRS Journal of Photogrammetry and Remote Sensing **231**, 679–703 (2026)
35. Wen, Y., Zhao, Y., Liu, Y., Jia, F., Wang, Y., Luo, C., Zhang, C., Wang, T., Sun, X., Zhang, X.: Panacea: Panoramic and controllable video generation for autonomous driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6902–6912 (2024)
36. Xu, B., Zhang, L., Liu, Y., Ai, H., Wang, B., Sun, Y., Fan, Z.: Robust hierarchical structure from motion for large-scale unstructured image sets. ISPRS Journal of Photogrammetry and Remote Sensing **181**, 367–384 (2021)
37. Yang, S., Tan, J., Zhang, M., Wu, T., Wetzstein, G., Liu, Z., Lin, D.: Layerpano3d: Layered 3D panorama for hyper-immersive scene generation. In: Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference. pp. 1–10 (2025)
38. Ye, W., Ji, C., Chen, Z., Gao, J., Huang, X., Zhang, S.H., Ouyang, W., He, T., Zhao, C., Zhang, G.: Diffpano: Scalable and consistent text to panorama generation with spherical epipolar-aware diffusion. Advances in Neural Information Processing Systems **37**, 1304–1332 (2024)
39. Yu, Z., Chen, A., Huang, B., Sattler, T., Geiger, A.: Mip-splatting: Alias-free 3D gaussian splatting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19447–19456 (2024)

40. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 586–595 (2018)