

# Seam360GS: Seamless 360° Gaussian Splatting from Real-World Omnidirectional Images

Changha Shin    Woong Oh Cho    Seon Joo Kim  
 Yonsei University  
 {changhashin, wocho, seonjookim}@yonsei.ac.kr

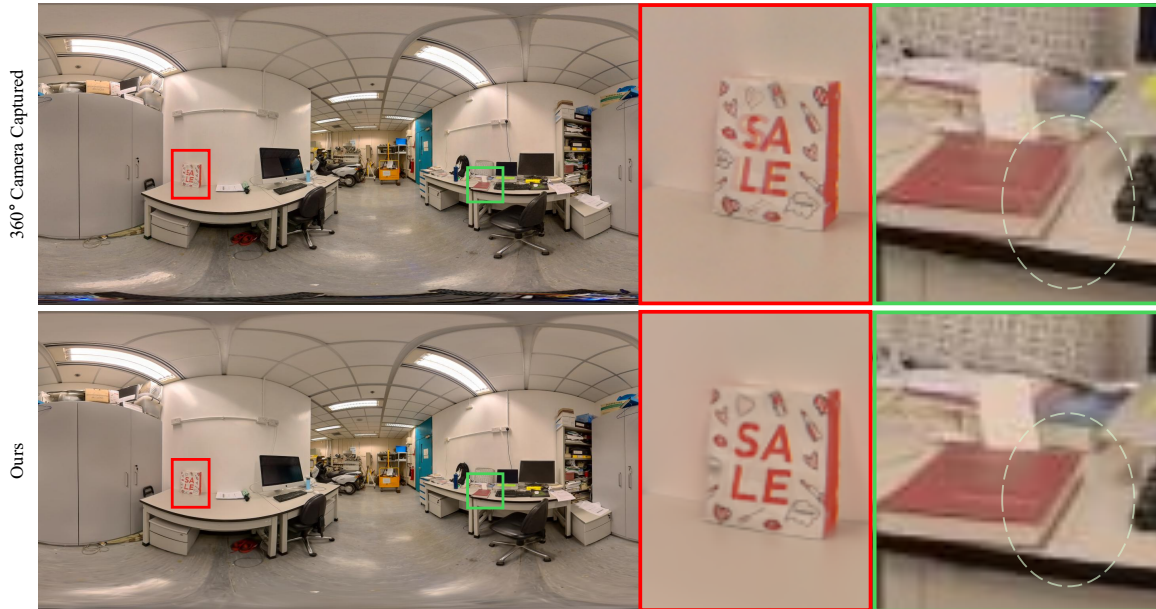


Figure 1. (Top) An indoor scene captured by a 360° camera, with red and green boxes indicating regions affected by stitching artifacts. (Bottom) Our model’s novel view synthesis result, demonstrating seamless rendering of scene details—including text and objects—free from stitching artifacts, with overall improvements in rendering quality and consistency.

## Abstract

360° visual content is widely shared on platforms such as YouTube and plays a central role in virtual reality, robotics, and autonomous navigation. However, consumer-grade dual-fisheye systems consistently yield imperfect panoramas due to inherent lens separation and angular distortions. In this work, we introduce a novel calibration framework that incorporates a dual-fisheye camera model into the 3D Gaussian splatting pipeline. Our approach not only simulates the realistic visual artifacts produced by dual-fisheye cameras but also enables the synthesis of seamlessly rendered 360° images. By jointly optimizing 3D Gaussian parameters alongside calibration variables that emulate lens gaps and angular distortions, our framework transforms imperfect omnidirectional inputs into flawless novel view synthesis. Extensive evaluations on real-world datasets confirm that our method produces seamless render-

ings—even from imperfect images—and outperforms existing 360° rendering models.

## 1. Introduction

Omnidirectional cameras have become integral to applications such as virtual reality, robotics, and autonomous navigation due to their ability to capture the entire surrounding environment in a single shot. These devices offer an expansive field of view and rich spatial context, both of which are crucial for robust scene understanding and high-level processing. However, most consumer-grade systems (e.g., Insta360, Ricoh360 [1, 2]) rely on dual fisheye lens configurations, which inherently introduce challenges. The physical separation between lenses and the angular distortions from the image stitching process result in panoramas that deviate from ideal geometric calibration, as illustrated in the top panel of Fig. 1. These calibration inaccuracies

can impair downstream tasks that depend on precise spatial reconstructions.

Recent advances in 3D reconstruction exemplified by Neural Radiance Fields (NeRF) [25] and 3D Gaussian splatting (3DGS) [16] have enabled photorealistic novel view synthesis. NeRF employs deep networks to learn a continuous radiance field from multi-view images, while 3DGS represents a scene using a sparse set of anisotropic Gaussians, allowing for efficient, real-time rendering. Since these methods were primarily designed for perspective images, they often fail to produce accurate renderings for omnidirectional data. To address these challenges, several works [4, 21, 22, 29] extend 3DGS to reconstruct an omnidirectional radiance field by projecting 3D Gaussians onto the tangent planes of a unit sphere. However, these approaches generally presume that the panoramic images have been accurately calibrated through spherical mapping.

Numerous studies [12, 23, 27, 28, 30, 33] have sought to improve novel view synthesis through camera calibration; however, most of these methods struggle with omnidirectional images having extremely wide fields of view. A recent 3DGS-based approach [15] calibrates lens distortions in 360° images while refining camera poses. However, it addresses only camera distortions and neglects the inter-camera gap in dual fisheye systems—an omission that significantly constrains seamless image synthesis.

In this paper, we propose an integrated calibration framework that jointly optimizes both the 3D Gaussian parameters and additional calibration variables to compensate for lens gap separation and angular distortions. Specifically, we redefine Gaussian splat positions by translating them to correspond with the optical centers of the fisheye cameras and apply 3D transformations to simulate rotational distortions introduced by the camera rays. This strategy refines the reconstructed omnidirectional radiance field and facilitates seamless 360° image rendering, as shown in the bottom panel of Fig. 1, all while maintaining the efficiency of the Gaussian Splatting paradigm. In summary, our contributions are:

- We propose a novel framework that integrates omnidirectional dual-fisheye distortion modeling with the 3D Gaussian splatting paradigm. Our algorithm optimizes the ideal 3D positions of Gaussian splats by jointly calibrating for both lens distortions and the inter-camera gap.
- Our approach enables seamless omnidirectional novel view synthesis without incurring additional computational overhead. During training, we generate realistic, imperfect 360° images that simulate dual fisheye artifacts and compare them to actual captures to learn robust calibration parameters. At inference, the calibration module is deactivated, ensuring that the system operates efficiently while still delivering flawlessly rendered 360° views.

- Extensive experiments on real-world 360° benchmark datasets [9, 14] demonstrate that our method consistently outperforms existing state-of-the-art algorithms for omnidirectional novel view synthesis.

## 2. Related Works

### 2.1. Classical Image Stitching Approaches

Classical omnidirectional image stitching techniques produce seamless panoramic images from overlapping views captured by fisheye or dual-fisheye cameras. Early methods followed a sequential pipeline starting with rigorous calibration to correct the pronounced radial distortions inherent in wide-angle lenses. By accurately modeling these distortions, the input images were normalized to establish a robust foundation for subsequent feature extraction and alignment [7, 24]. In this pipeline, robust keypoints were detected using descriptors such as SIFT [24] or SURF [5] and then matched across overlapping regions, with RANSAC [11] used to compute global homographies under the assumption of planar scenes—a premise that often fails in environments with significant parallax and depth variations.

To address these challenges, later research introduced multi-homography models and spatially varying warping strategies that adapt transformations locally to better accommodate non-planar geometries. Advanced blending techniques were concurrently developed to smooth intensity variations and eliminate visible seams, ensuring consistent brightness and color continuity. However, these traditional stitching pipelines generally assume near-ideal capture conditions and sufficient overlap, which can result in artifacts when applied to consumer-grade dual-fisheye systems.

More recently, deep learning approaches have emerged to overcome these limitations by learning robust, high-level representations that directly tackle issues such as parallax, non-planarity, and exposure mismatches in dual-fisheye and 360° imaging systems. For instance, the OmniStitch [18] integrates depth cues via a depth-aware network, enabling more reliable alignment across varying distances and complex geometries while reducing ghosting and misregistration artifacts. Similarly, attentive deep stitching methods [17] employ attention mechanisms to adaptively fuse overlapping regions, effectively managing local intensity differences and subtle misalignments, while additional depth-aware strategies [32] further enhance robustness under large parallax conditions. Comprehensive surveys on deep learning for omnidirectional vision [3] highlight that these advances relax many of the restrictive assumptions of classical pipelines, achieving superior visual consistency and artifact suppression. Nonetheless, despite producing perceptually appealing panoramas, deep learning-based methods remain constrained by the inherent physical limitations of the imaging process, often resulting in suboptimal geometric fidelity for applications demanding rigorous spatial accuracy.

## 2.2. Novel View Synthesis for 360° Imaging

Neural Radiance Fields (NeRF) [25] have revolutionized novel view synthesis by representing scenes as continuous volumetric radiance fields parameterized by neural networks. However, NeRF typically requires extensive training time and significant computational resources, prompting research into more efficient representations. In this context, 3D Gaussian splatting has emerged as a compelling alternative. By modeling a scene as a collection of anisotropic Gaussian primitives, this approach offers an explicit, point-based representation that can be rendered in real time while maintaining competitive visual quality. Unlike the implicit volumetric representations in NeRF, 3D Gaussian splatting leverages well-localized Gaussian functions to approximate scene geometry and appearance, thereby enabling rapid synthesis of novel views with significantly reduced rendering costs.

**NeRF-Based Approaches.** While the foundational NeRF framework has already been described in earlier works, its extension to omnidirectional data has spawned a number of specialized approaches tailored for 360° imagery. Methods such as [8, 9, 14, 20] adapt the standard NeRF pipeline to accommodate the challenges posed by panoramic inputs. These approaches modify the conventional sampling and representation strategies to work in spherical coordinates, thereby effectively handling the wide field-of-view inherent to 360° cameras. For example, 360Roam [14] employs geometry-aware sampling to capture indoor environments robustly, while 360FusionNeRF [20] and PanoGRF [8] extend the NeRF paradigm to support wide-baseline panoramas by integrating specialized feature grids and joint guidance strategies. Notably, EgoNeRF [9] focuses on egocentric scenes by utilizing a quasi-uniform spherical grid that adapts to the unbounded nature of panoramic views. Notably, while these approaches generally deliver impressive reconstruction quality, 360Roam [14] points out a potential limitation: manufacturing imperfections and suboptimal factory calibration can introduce noticeable stitching artifacts in the source images, which may adversely affect the final rendering quality.

**3D Gaussian Splatting Approaches.** Recent advances in omnidirectional radiance field reconstruction—exemplified by works such as 360GS [4], ODGS [21], OP43DGS [29], and OmniGS [22] extend the 3D Gaussian Splatting framework to 360° imagery by leveraging specialized rasterizers that project 3D Gaussians onto tangent planes of a unit sphere. 360GS adapts the standard splatting technique for optimal panoramic rendering, ODGS enhances fine detail recovery through directional augmentation, OP43DGS employs an optimized splatting approach to accurately reconstruct depth and complex scene structures, and OmniGS leverages the optical properties of omnidirectional images

to achieve fast, precise radiance field reconstruction.

While most NeRF and 3DGS methods assume that mapping omnidirectional images to a spherical domain ensures inherently accurate internal geometry, our work challenges this notion by showing that dual fisheye camera systems introduce intrinsic calibration issues. Specifically, a gap between the two fisheye lenses and the resulting angular distortions in the camera rays deviate from the idealized geometric model.

## 2.3. Neural Rendering with Camera Calibration

Recent research in neural rendering has increasingly focused on jointly optimizing scene representations and camera parameters to overcome the limitations imposed by imperfect calibration. For instance, NeRF-- [33] was one of the earliest works to jointly optimize scene representations and camera parameters, thereby eliminating the need for pre-calibrated inputs. Building on this foundation, methods such as SCNeRF [30] integrate both intrinsic and extrinsic calibration into the NeRF optimization process by simultaneously refining camera intrinsics (e.g., focal length and distortion coefficients) and extrinsics (e.g., camera poses) to compensate for lens imperfections and misalignments common in consumer-grade imaging systems. Other approaches, including BARF [23], leverage bundle adjustment to iteratively refine camera poses, while additional works like Nope-NeRF [6], CAMP [27], and SPARF [28] incorporate advanced pose estimation and sparse representation techniques for efficient view synthesis. Collectively, these methods demonstrate a powerful trend toward reducing reliance on pre-calibrated inputs by addressing calibration errors during training, thereby enhancing the overall quality of novel view synthesis.

Similar calibration strategies have been extended to alternative neural rendering approaches such as 3D Gaussian Splatting. In particular, COLMAP-Free 3D Gaussian Splatting [12] has emerged that eliminates the dependency on external Structure-from-Motion pipelines. These approaches embed calibration corrections within the Gaussian splatting framework itself, jointly optimizing both the scene representation and the camera parameters. This integration not only streamlines the processing pipeline but also enhances robustness against residual calibration errors, whether they originate from intrinsic distortions or extrinsic misalignments. Collectively, these calibrated neural rendering methods demonstrate a powerful trend towards reducing reliance on pre-calibrated inputs. By addressing both intrinsic and extrinsic errors during training, these methods achieve improved alignment and rendering quality, a critical advancement for challenging scenarios such as 360° novel view synthesis from dual-fisheye cameras.

Recent works have also focused on calibrating 360° images for neural rendering. For example, Omni-NeRF

[13] leverages raw dual fisheye images to reconstruct neural radiance fields, primarily mitigating wide field-of-view lens distortions, yet it does not address the stitching artifacts in omnidirectional images commonly observed in consumer-grade systems. Additionally, most methods rely on equirectangular images that have been pre-corrected, which makes obtaining raw fisheye images challenging in many cases. SC-OmniGS [15] jointly optimizes both lens distortions and camera poses to enhance the quality of the reconstructed scene. It effectively addresses lens distortions using a UV mapping approach, correcting them directly within the 2D image domain. However, this method primarily focuses on lens distortions and does not account for inter-camera gaps. In contrast, our algorithm explicitly adjusts the positions of Gaussian splats in the 3D scene space by jointly calibrating for both lens distortions and inter-camera gaps, leading to more accurate geometric reconstruction and seamless omnidirectional rendering.

### 3. Methodology

In this section, we present the preliminaries on 3D Gaussian splatting and its extension to 360° imaging, and then describe our integrated calibration framework.

#### 3.1. Preliminaries: 3D Gaussian splatting

3D Gaussian splatting represents a scene as a set of anisotropic Gaussian primitives, each of which encodes local geometric and appearance information. A single Gaussian primitive is defined by its center, covariance, and associated color and opacity. Formally, a Gaussian is given by:

$$G(\mathbf{x}) = \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1}(\mathbf{x} - \boldsymbol{\mu})\right), \quad (1)$$

where  $\boldsymbol{\mu} \in \mathbb{R}^3$  is the center of the Gaussian and  $\Sigma \in \mathbb{R}^{3 \times 3}$  is the covariance matrix that captures its anisotropic spread.

During rendering, the 3D Gaussians are projected onto the image plane via a camera projection function  $\pi(\cdot)$ . The projected Gaussians are then sampled and  $\alpha$ -blended to produce the final image. In our simplified formulation, the image intensity at pixel  $p$  is computed as

$$I(p) = \sum_i \alpha_i(p) c_i, \quad (2)$$

where the blending weight for the  $i$ th Gaussian is defined as

$$\alpha_i(p) = o_i \exp\left(-\frac{1}{2}\|p - \pi(m_i)\|_{W_i}^2\right). \quad (3)$$

Here,  $o_i$  is the opacity of the  $i$ th Gaussian,  $\pi(m_i)$  denotes its projected center, and  $\|p - \pi(m_i)\|_{W_i}^2$  represents the Mahalanobis distance with respect to the projected covariance  $W_i$ . This differentiable forward rendering pipeline enables

the joint optimization of all Gaussian parameters through photometric loss, even under sparse sampling conditions.

To adapt 3D Gaussian splatting for omnidirectional imaging, [4, 21, 22, 29] project the Gaussian primitives onto a unit sphere, normalizing each 3D point’s coordinates to capture a full 360° view of the scene. While extending 3D Gaussian splatting to omnidirectional imaging establishes an effective rasterizer by projecting Gaussian primitives onto a unit sphere, their approach inherently assumes that the input 360° images are accurately stitched. These imperfections degrade geometric accuracy and visual consistency in the rendered omnidirectional outputs.

#### 3.2. Dual-Fisheye Distortion Modeling and Calibration for 3D Gaussian splatting

To simulate the inherent imperfections of dual-fisheye systems, we convert the ideal Gaussian primitives into camera-specific representations via two key steps: (1) translating the Gaussians to align with the dual-fisheye centers, and (2) applying angular distortion to emulate errors arising in the omnidirectional stitching process. A schematic overview of our proposed algorithm is illustrated in Fig. 2.

**Translation of Gaussians to Dual-Fisheye Centers.** In practical dual-fisheye systems, hardware constraints lead to a misalignment between the optical centers of the front and back cameras, preventing them from coinciding with an idealized camera center. To correct for this discrepancy, we incorporate a calibration step that translates the Gaussian primitives accordingly as shown in Fig. 2(a). Specifically, we denote the front and back camera centers as  $O_F$ ,  $O_B$  respectively, relative to an ideal center  $O_{\text{ideal}}$ . This update is performed using learnable translation vectors  $\Delta T_F$  and  $\Delta T_B$ , which are optimized via backpropagation during the Gaussian Splatting training process. Specifically, the updated camera centers are given by

$$O_F = O_{\text{ideal}} + R_F \Delta T_F, \quad (4)$$

$$O_B = O_{\text{ideal}} + R_B \Delta T_B, \quad (5)$$

where  $R_F$  and  $R_B$  are the rotation matrices corresponding to the front and back cameras.

Each Gaussian, originally defined in the ideal camera coordinate system as  $G_{\text{ideal}}$ , is translated into the dual-fisheye camera coordinates. For the front and back cameras, we have

$$g_F^j = R_F^{-1}(G_{\text{ideal}} - O_F), \quad (6)$$

$$g_B^k = R_B^{-1}(G_{\text{ideal}} - O_B). \quad (7)$$

Here, the index  $j$  denotes each Gaussian element associated with the front camera, while  $k$  denotes each Gaussian element associated with the back camera.

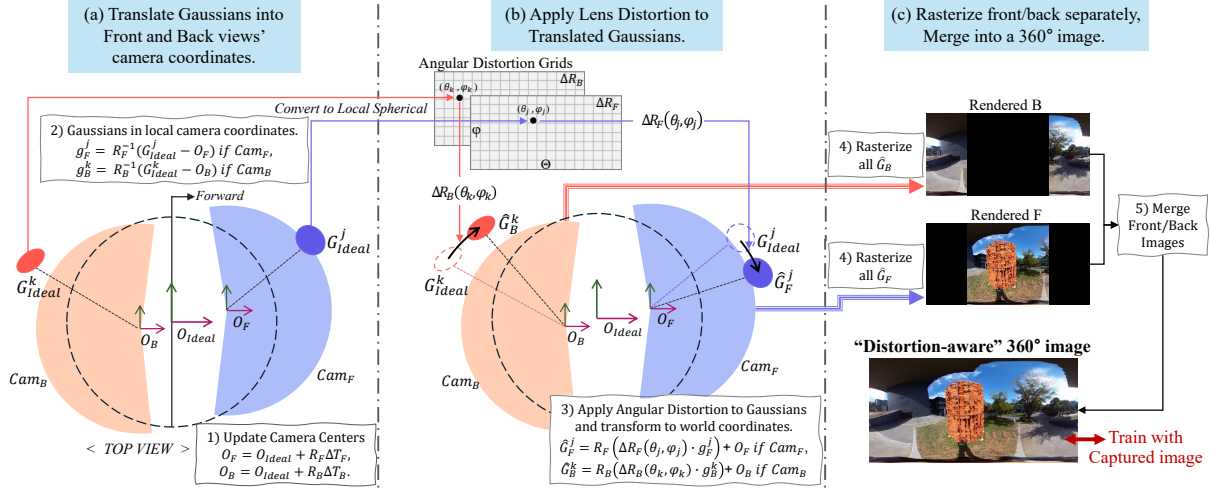


Figure 2. Overview of our seamless 360° Gaussian Splatting rendering pipeline, which integrates fisheye camera stitching and applies lens distortion to fully replicate the distortions observed in actual camera captures.

### Simulated Angular Distortion in Omnidirectional Image Stitching.

To account for the angular distortions introduced during the stitching process in dual-fisheye systems, we apply angular distortions to the Gaussian primitives. In our formulation, the angular distortion parameter  $\Delta R$  is modeled as a learnable embedding. For the front camera, this embedding is denoted as  $\Delta R_F \in \mathbb{R}^{M \times N \times 3}$ , where  $M$  and  $N$  represent the discretized resolutions along the spherical coordinates  $\theta$  and  $\phi$ , respectively, and the final dimension (of size 3) encodes the corresponding rotation adjustments. As illustrated in Fig. 2(b), given a Gaussian with spherical coordinates  $(\theta, \phi)$  in its local camera coordinate system, bilinear sampling from  $\Delta R_F$  yields a rotation offset that specifies the degree to which the Gaussian should be rotated—relative to the local camera frame—to model the stitching-induced angular distortions, satisfying  $\Delta R_F(\theta_j, \phi_j) \in SO(3)$ , which is applied in Eqs. (8)–(9) to model ray-dependent distortion. Similarly, a learnable embedding  $\Delta R_B$  is defined for the back camera. After translating each Gaussian into its local camera coordinate system, these angular distortion adjustments are applied, resulting in distorted Gaussian representations that more accurately reflect the true scene geometry.

To align the translated Gaussians with the spherical representation of each camera’s field, we apply angular distortions  $\Delta R_F(\theta, \phi)$  and  $\Delta R_B(\theta, \phi)$ . The corrected Gaussians are given by

$$G_F^j = R_F(\Delta R_F(\theta_j, \phi_j) \cdot g_F^j) + O_F, \quad (8)$$

$$G_B^k = R_B(\Delta R_B(\theta_k, \phi_k) \cdot g_B^k) + O_B. \quad (9)$$

These distortions, inherent to the dual-fisheye setup, integrate the omnidirectional image stitching process directly

into the 3D Gaussian splatting framework, enabling the realistic and seamless rendering of 360° images despite calibration imperfections.

The Transformed Gaussians are rasterized within each camera’s view using a 360° Gaussian rasterizer as depicted in Fig. 2(c). For the front and back cameras, the Gaussian primitives are independently projected to generate equirectangular images, which are then concatenated to form a complete 360° panoramic view. This procedure reproduces the characteristics of distortions observed in consumer-grade 360° captures. Consequently, the synthesized outputs accurately reflect these artifacts, making them suitable as imperfect ground truth for training and evaluating the 3D Gaussian splatting pipeline.

Moreover, once training is complete, our model can also render seamless 360° panoramas directly without the additional computational cost of the distortion process. Since the initial Gaussians are defined in an ideally calibrated space, omitting the distortion step during inference produces final renderings that closely approximate an ideal, distortion-free 360° image. In effect, the output resembles a 360° capture obtained even when the 3D Gaussian splatting is trained on imperfectly stitched inputs, effectively overcoming the spatial inaccuracies.

## 4. Experiments

### 4.1. Experimental Setup

**Datasets.** We evaluated camera calibration performance using a synthetic dataset derived from OmniBlender [9], designed to emulate typical distortions in dual-fisheye 360° camera setups. We generated equirectangular images by combining two virtual fisheye views (front and back), each rendered with a slightly displaced camera. The inter-camera displacement was randomly sampled between -2 and 2 cm

in all directions. Each pair of fisheye images covered a 185° field of view and was stitched into a single distorted panorama to simulate realistic lens imperfections. Each synthetic scene included 50 equirectangular images, evenly split into 25 for training and 25 for testing.

To validate whether our method achieves seamless omnidirectional synthesis across various real-world scenarios, we conducted experiments on two real-world 360° datasets, following the evaluation protocol from OmniGS [22]. Specifically, we used the Ricoh360 dataset introduced in EgoNeRF [9], consisting of scenes captured with a Ricoh360 camera, and the 360Roam dataset [14], captured using an Insta360 camera. Images from the 360Roam dataset were resized to 1520 × 760, and the lower region (below pixel row 712) was cropped to remove a mobile robot from the scene. To ensure broad applicability, our method operates directly on ERP-format panoramas, which are the standard for most publicly available 360° content.

**Metrics.** We evaluated omnidirectional novel view synthesis using Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), and LPIPS (using AlexNet) [19, 31]. Additionally, we quantified camera positional calibration accuracy by computing the Mean Absolute Error (MAE) between the predicted and actual inter-camera gaps.

**Implementations.** Implementation of Seamless 360° Gaussian Splatting was performed using the PyTorch framework [26] and the 360° rasterizer from OmniGS [22], which is built upon the original 3D Gaussian splatting framework [16]. Experiments on real-world scenes adhered to the default parameter settings of the original 3D Gaussian splatting implementation, with one modification: the camera extent for the Ricoh360 dataset from EgoNeRF [9] was scaled by a factor of 10 to accommodate the limited camera motion. To ensure a fair comparison, all baseline models were trained using their officially released code on a single NVIDIA A6000 GPU with a training time limit of 90 minutes. Due to memory limitations on the Ricoh360 dataset, OP43DGS [22] utilized the checkpoint saved immediately before training terminated as a result of an out-of-memory error. To train the learnable angular distortion grids  $\Delta R_F, R_B$  and camera gaps  $\Delta T_F, T_B$ , we initialized the learning rate at 0.001 and gradually decayed it exponentially to 0.0001. The learning rate scheduling followed the approach used in JaxNeRF [10], ensuring stable convergence while refining camera parameters throughout training. Additionally, a total variation loss was applied to  $\Delta R_F, R_B$  to enforce smoothness and prevent abrupt variations in the estimated rotations.

## 4.2. Synthetic Calibration Accuracy Assessment

We present a quantitative analysis of our model’s calibration performance using synthetic scenes. Our dataset pro-

vides ground-truth camera gap values between the dual fisheye cameras, enabling a precise evaluation of positional accuracy. For comparison, we benchmark against OmniNeRF—a self-calibrating method capable of correcting both lens distortion and inter-camera gaps. Despite training OmniNeRF for 90 minutes—more than twice the duration required by our approach—as shown in Table 1, our method achieves an average positional error of only 0.11 cm. Our algorithm not only demonstrates exceptional calibration accuracy but also ensures precise scene reconstruction, as validated by the high-fidelity renderings in Fig. 3 (top portion).

Method	Mean Absolute Error (cm)				
	<i>barbershop</i>	<i>classroom</i>	<i>restroom</i>	<i>lone_monk</i>	Average
Ours	0.1166	0.1965	0.1044	0.0423	0.1150
OmniNeRF [13]	2.761	3.602	3.454	3.377	3.299

Table 1. Inter-camera gap calibration performance comparison with OmniNeRF on the synthetic dataset. OmniNeRF utilizes dual fisheye images for self-calibration, and the calibration accuracy is reported as mean absolute error (MAE) in centimeters.

## 4.3. Quantitative and Qualitative Evaluation on Synthetic and Real-World Datasets

We conducted extensive experiments to compare our method with state-of-the-art algorithms for 360° novel view synthesis [9, 21, 22, 29] on both synthetic and real-world data. For the real-world evaluation, we analyzed 11 scenes from the EgoNeRF dataset (captured with a Ricoh360 camera) and 10 scenes from the 360Roam dataset (captured with an Insta360 camera).

As summarized in Table 2, our model outperforms competing methods across all metrics while utilizing fewer Gaussians, with particularly notable gains on indoor scenes from the 360Roam dataset. ( Full experimental results are provided in the supplementary material )

Qualitative evaluation further supports these findings. As shown in Fig. 3, our model produces the highest quality outputs, while other state-of-the-art 360° Gaussian Splatting algorithms struggle to capture fine details and accurately reproduce text. As depicted in Fig. 1, camera gap and lens

Dataset	Metric	EgoNeRF [9]	OP43DGS [29]	ODGS [21]	OmniGS [22]	Ours
Synthetic	PSNR↑	25.468	27.825	28.833	32.421	<b>34.986</b>
	SSIM↑	0.746	0.874	0.866	0.927	<b>0.947</b>
	LPIPS↓	0.232	0.116	0.116	0.053	<b>0.048</b>
	#Points	-	411K	1002K	374K	308K
Ricoh360	PSNR↑	24.639	23.778	22.556	26.064	<b>26.960</b>
	SSIM↑	0.746	0.749	0.739	0.829	<b>0.852</b>
	LPIPS↓	0.250	0.267	0.238	0.125	<b>0.105</b>
	#Points	-	329K	1276K	619K	621K
360Roam	PSNR↑	23.021	23.689	22.394	25.083	<b>26.101</b>
	SSIM↑	0.693	0.760	0.710	0.803	<b>0.829</b>
	LPIPS↓	0.368	0.234	0.264	0.145	<b>0.144</b>
	#Points	-	236K	971K	836K	447K

Table 2. Quantitative comparisons on synthetic and real-world datasets. The table summarizes the average PSNR, SSIM, LPIPS, and point count across all evaluated datasets, highlighting the performance of our novel view synthesis method.

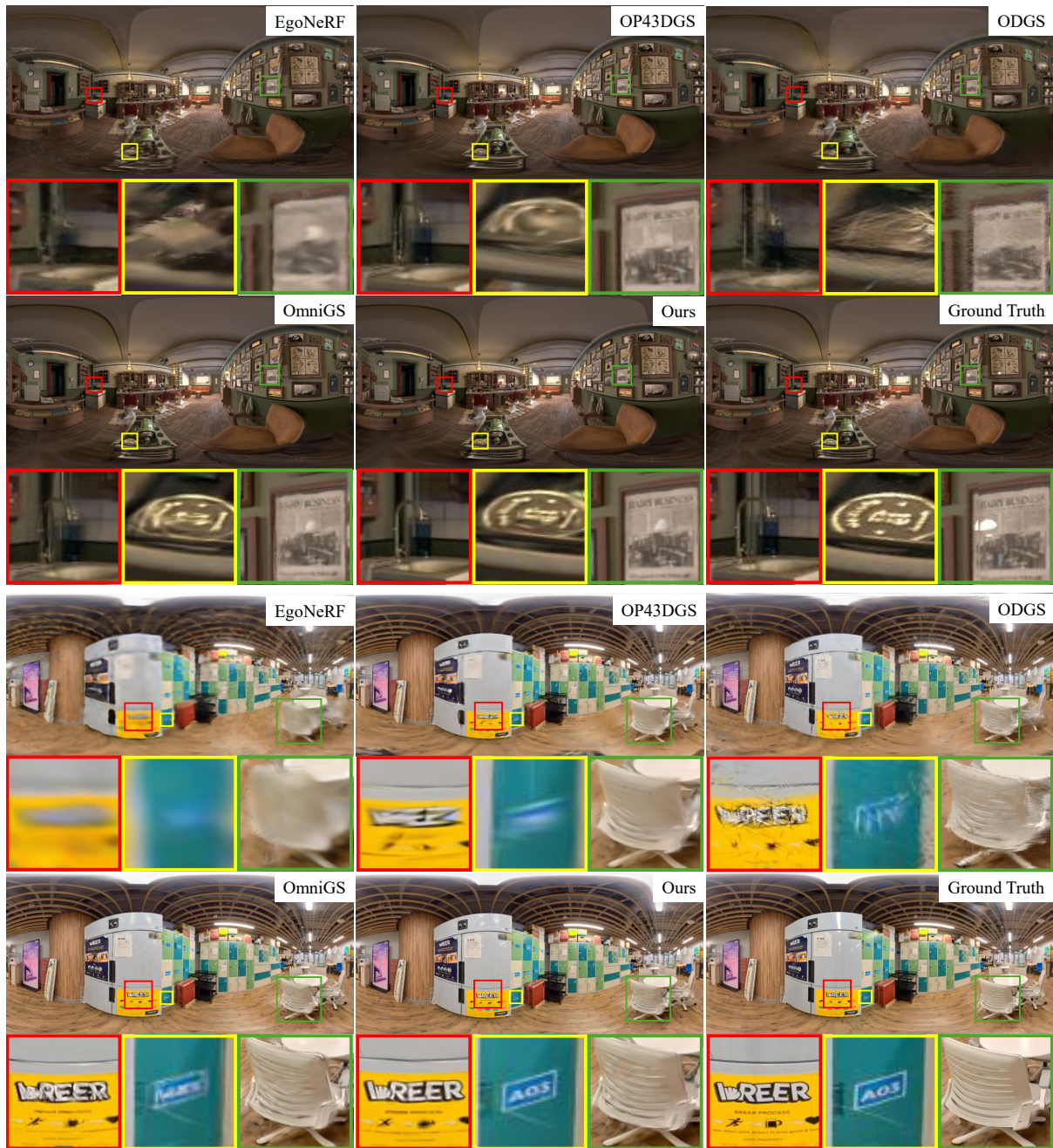


Figure 3. Qualitative comparisons of novel-view synthesis results. The top portion shows images from the synthetic dataset (*barbershop* scene), while the bottom portion displays results from the 360Roam dataset (*base* scene).

errors lead to stitching issues—resulting in artifacts such as ringing effects, object overlap, and blurriness due to imprecise ground truth.

Additionally, Fig. 4 presents a comparison between seamlessly rendered images and realistic results from real-world scenes. The ground-truth image exhibits noticeable stitching errors along the seam due to the dual fisheye configuration, whereas our seamless model reconstructs the scene with higher fidelity—sometimes even outperforming

the captured images.

We also compare training time, peak VRAM usage, and inference speed against OmniGS (see Table 4). Although our approach requires more than twice the training time and up to 2 GB additional VRAM, it achieves 30% faster rendering at test time. This speedup stems from our stitching model’s precise Gaussian fitting, which generates accurate camera rays, reducing the number of primitives needed for high-quality novel-view synthesis.

Scene	Ours	Ours w/o Lens Calib	Ours w/o Gap	Ours w/o Lens Calib, Gap
	PSNR $\uparrow$ / SSIM $\uparrow$ / LPIPS $\downarrow$ / #Points	PSNR $\uparrow$ / SSIM $\uparrow$ / LPIPS $\downarrow$ / #Points	PSNR $\uparrow$ / SSIM $\uparrow$ / LPIPS $\downarrow$ / #Points	PSNR $\uparrow$ / SSIM $\uparrow$ / LPIPS $\downarrow$ / #Points
<i>Bar</i>	<b>23.155 / 0.804 / 0.157</b> / 592K	22.593 / 0.785 / 0.167 / 587K	23.040 / 0.801 / 0.158 / 585K	22.497 / 0.782 / 0.168 / 587K
<i>Base</i>	<b>26.496 / 0.856 / 0.089</b> / 806K	25.487 / 0.823 / 0.101 / 804K	26.295 / 0.852 / 0.090 / 796K	24.918 / 0.805 / 0.108 / 788K
<i>Cafe</i>	<b>26.139 / 0.859 / 0.099</b> / 518K	25.472 / 0.841 / 0.108 / 556K	26.048 / 0.857 / 0.100 / 513K	25.231 / 0.835 / 0.112 / 551K
<i>Canteen</i>	<b>22.863 / 0.762 / 0.204</b> / 322K	22.652 / 0.754 / 0.208 / 326K	22.813 / 0.760 / <b>0.204</b> / 317K	22.443 / 0.744 / 0.211 / 314K
<i>Center</i>	<b>25.824 / 0.822 / 0.176</b> / 372K	25.312 / 0.808 / 0.181 / 400K	25.747 / 0.820 / 0.177 / 371K	25.135 / 0.804 / 0.184 / 395K
<i>Corridor</i>	<b>25.733 / 0.826 / 0.166</b> / 202K	25.381 / 0.818 / 0.173 / 205K	25.135 / 0.819 / 0.169 / 199K	25.238 / 0.810 / 0.174 / 206K
<i>Innovation</i>	<b>27.174 / 0.852 / 0.120</b> / 784K	26.364 / 0.821 / 0.129 / 811K	27.084 / 0.851 / <b>0.120</b> / 784K	26.053 / 0.811 / 0.133 / 801K
<i>Lab</i>	<b>30.171 / 0.917 / 0.060</b> / 329K	28.888 / 0.900 / 0.069 / 351K	29.946 / 0.914 / 0.061 / 327K	28.501 / 0.895 / 0.072 / 346K
<i>Library</i>	<b>27.236 / 0.785 / 0.191</b> / 283K	26.641 / 0.767 / 0.197 / 293K	27.187 / 0.784 / 0.192 / 281K	26.326 / 0.760 / 0.200 / 291K
<i>Office</i>	<b>26.212 / 0.804 / 0.180</b> / 265K	25.746 / 0.795 / 0.184 / 277K	25.912 / 0.801 / 0.183 / 262K	25.022 / 0.783 / 0.195 / 269K
<i>Average</i>	<b>26.101 / 0.829 / 0.144</b> / 447K	25.453 / 0.811 / 0.152 / 461K	25.921 / 0.826 / 0.145 / 444K	25.136 / 0.803 / 0.156 / 455K

Table 3. Ablation study evaluating the effectiveness of two key components in our training pipeline: translation-based adjustments for simulating camera gap and angular distortion-based transformations for modeling lens distortion. "w/o gap" indicates the removal of the camera gap component, and "w/o lens calib" denotes the exclusion of the lens distortion component. Removing these elements results in a significant performance degradation, particularly underscoring the strong impact of angular distortion correction.

Method	Scene	Train Time	Peak Mem	FPS (Test)	#Points
OmniGS	Office	20 m 31 s	5 679 MB	96.52	600K
	Library	21 m 22 s	5 140 MB	83.17	526K
Ours	Office	54 m 47 s	6 633 MB	136.12	265K
	Library	56 m 49 s	7 143 MB	112.68	283K

Table 4. Comparison of training time, VRAM usage, and test-time FPS between OmniGS [22] and our method on 360Roam scenes (Office, Library).

#### 4.4. Ablation Study on Our Seamless Rendering Components

To evaluate our calibration process, we conducted ablation experiments using the 360Roam indoor dataset (see Fig. 4), where calibration is particularly critical due to the prevalence of close-range objects. Our model adjusts Gaussian splats via two mechanisms—translation and angular distortion—thereby simulating both camera gap and lens distortion during training. We assessed the contribution of each component with experiments summarized in Tab. 3. The results indicate that both mechanisms significantly enhance performance, with angular distortion having an especially strong effect; notably, disabling both adjustments results in an average PSNR drop of 1 dB.

## 5. Conclusion

In this work, we introduced an integrated calibration framework for omnidirectional novel view synthesis that addresses the challenges inherent in dual-fisheye camera systems. By incorporating translation-based adjustments and angular distortion corrections within the 360° Gaussian Splatting paradigm, our method effectively compensates for camera gap and lens distortions, leading to highly accurate scene reconstructions. Extensive experiments on both synthetic and real-world datasets demonstrate that our approach not only outperforms state-of-the-art methods in terms of quantitative metrics such as PSNR, SSIM, and LPIPS, but also produces superior qualitative results with enhanced de-

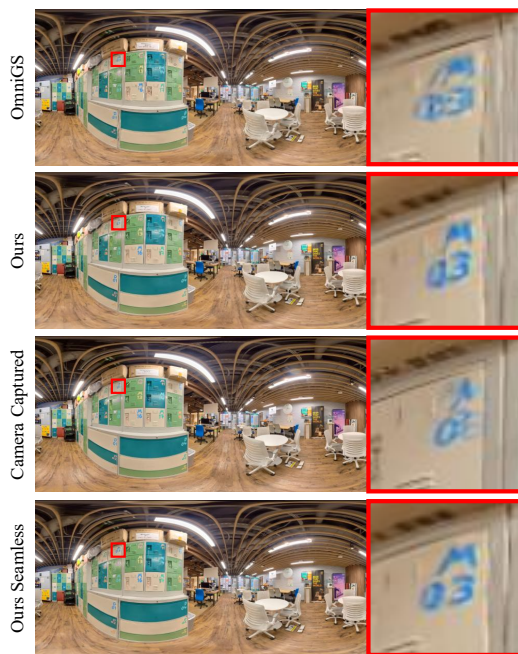


Figure 4. Qualitative comparison on the 360Roam base. The top row displays the output of the state-of-the-art model OmniGS, followed by our method’s results, the ground truth, and finally the seamless rendering output produced by our method.

tail and fewer artifacts. Our seamless rendering pipeline, validated through rigorous ablation studies, underscores the importance of precise calibration in achieving high-fidelity outputs. Future work will focus on further refining our calibration techniques, exploring real-time applications, and extending the framework to more complex imaging scenarios.

**Limitations** A notable limitation of our approach is reduced performance in textureless regions. Uniform walls and smooth areas (e.g., the sky) lack sufficient features for reliable calibration, and simultaneous view synthesis and calibration further exacerbate errors.

## Acknowledgements

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) with a grant funded by the Ministry of Science and ICT (MSIT) of the Republic of Korea in connection with the Global AI Frontier Lab International Collaborative Research. (No. RS-2024-00469482 & RS-2024-00509279), Artificial Intelligence Graduate School Program, Yonsei University, under Grant RS-2020-11201361.

## References

- [1] Insta360 camera. <http://www.insta360.com/>. 1
- [2] Ricoh360 camera. <https://www.ricoh360.com/>. 1
- [3] Hao Ai, Zidong Cao, Jinjing Zhu, Haotian Bai, Yucheng Chen, and Ling Wang. Deep learning for omnidirectional vision: A survey and new perspectives. *arXiv preprint arXiv:2205.10468*, 2022. 2
- [4] Jiayang Bai, Letian Huang, Jie Guo, Wen Gong, Yuanqi Li, and Yanwen Guo. 360-gs: Layout-guided panoramic gaussian splatting for indoor roaming. *arXiv preprint arXiv:2402.00763*, 2024. 2, 3, 4
- [5] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). In *Computer Vision and Image Understanding*, pages 346–359. Elsevier, 2008. 2
- [6] Wenjing Bian, Zirui Wang, Kejie Li, Jiawang Bian, and Victor Adrian Prisacariu. Nope-nerf: Optimising neural radiance field with no pose prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 3
- [7] Matthew Brown and David G Lowe. Automatic panoramic image stitching using invariant features. In *International Journal of Computer Vision*, pages 59–73. Springer, 2007. 2
- [8] Zheng Chen, Yan-Pei Cao, Yuan-Chen Guo, Chen Wang, Ying Shan, and Song-Hai Zhang. Panogrf: Generalizable spherical radiance fields for wide-baseline panoramas. In *NeurIPS*, 2023. 3
- [9] Changwoon Choi, Sang Min Kim, and Young Min Kim. Balanced spherical grid for egocentric view synthesis. In *CVPR*, 2023. 2, 3, 5, 6
- [10] Boyang Deng, Jonathan T. Barron, and Pratul P. Srinivasan. JaxNeRF: an efficient JAX implementation of NeRF. 2020. 6
- [11] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 2
- [12] Yang Fu, Sifei Liu, Amey Kulkarni, Jan Kautz, Alexei A. Efros, and Xiaolong Wang. Colmap-free 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2, 3
- [13] Kai Gu, Thomas Maugey, Sebastian Knorr, and Christine Guillemot. Omni-nerf: neural radiance field from 360 image captures. In *2022 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2022. 4, 6
- [14] Huajian Huang, Yingshu Chen, Tianjia Zhang, and Sai-Kit Yeung. 360roam: Real-time indoor roaming using geometry-aware 360° radiance fields. *arXiv preprint arXiv:2208.02705*, 2022. 2, 3, 6
- [15] Huajian Huang, Yingshu Chen, Longwei Li, Hui Cheng, Tristan Braud, Yajie Zhao, and Sai-Kit Yeung. SC-omniGS: Self-calibrating omnidirectional gaussian splatting. In *The Thirteenth International Conference on Learning Representations*, 2025. 2, 4
- [16] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 2, 6
- [17] Hak Gu Kim, Heoun-Taek Lim, and Yong Man Ro. Attentive deep stitching and quality assessment for 360° omnidirectional images. *IEEE Journal of Selected Topics in Signal Processing*, 14(1):209–221, 2020. 2
- [18] Minsu Kim, Jaewon Lee, and Sungmin Park. Omnistitch: Depth-aware stitching framework for omnidirectional images. In *Proceedings of the ACM Multimedia Conference*, 2024. 2
- [19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*. Curran Associates, Inc., 2012. 6
- [20] Shreyas Kulkarni, Peng Yin, and Sebastian Scherer. 360fusionnerf: Panoramic neural radiance fields with joint guidance. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023. 3
- [21] Suyoung Lee, Jaeyoung Chung, Jaeyoo Huh, and Kyoung Mu Lee. Odgs: 3d scene reconstruction from omnidirectional images with 3d gaussian splattings. In *NeurIPS*, 2024. 2, 3, 4, 6
- [22] Longwei Li, Huajian Huang, Sai-Kit Yeung, and Hui Cheng. Omnigs: Omnidirectional gaussian splatting for fast radiance field reconstruction using omnidirectional images. *arXiv preprint arXiv:2404.03202*, 2024. 2, 3, 4, 6, 8
- [23] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 2, 3
- [24] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 2
- [25] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 2, 3
- [26] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 6
- [27] Park, Keunhong and Henzler, Philipp and Mildenhall, Ben and Barron, Jonathan T. and Martin-Brualla, Ricardo. Camp: Camera preconditioning for neural radiance fields. In *ACM Trans. Graph.*, 2023. 2, 3
- [28] Prune Truong, Marie-Julie Rakotosaona, Fabian Manhardt, and Federico Tombari. Sparf: Neural radiance fields from

- sparse and noisy poses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. [2](#), [3](#)
- [29] X. Wang and et al. op43dgs: Optimized projection for 360° gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. [2](#), [3](#), [4](#), [6](#)
- [30] Christopher Choy Animashree Anandkumar Minsu Cho Yoonwoo Jeong, Seokjun Ahn and Jaesik Park. Self-calibrating neural radiance fields. In *ICCV*, 2021. [2](#), [3](#)
- [31] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. [6](#)
- [32] Wei Zhang, Ming Li, and Jian Chen. Depth-aware stitching framework for omnidirectional vision. In *Proceedings of the ICCV Workshops*, 2023. [2](#)
- [33] Weidi Xie Min Chen Victor Adrian Prisacariu Zirui Wang, Shangzhe Wu. Nerf-: Neural radiance fields without camera pose calibration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. [2](#), [3](#)