FullCircle:
Effortless 3D Reconstruction from Casual 360° Captures

Yalda Foroutan*[1] Ipek Oztas*[1,4] Daniel Rebain[2] Aysegul Dundar[4] Kwang Moo Yi[2]
Lily Goli[3] Andrea Tagliasacchi[1,3]

[1]Simon Fraser University [2]UBC [3]University of Toronto [4]Bilkent University

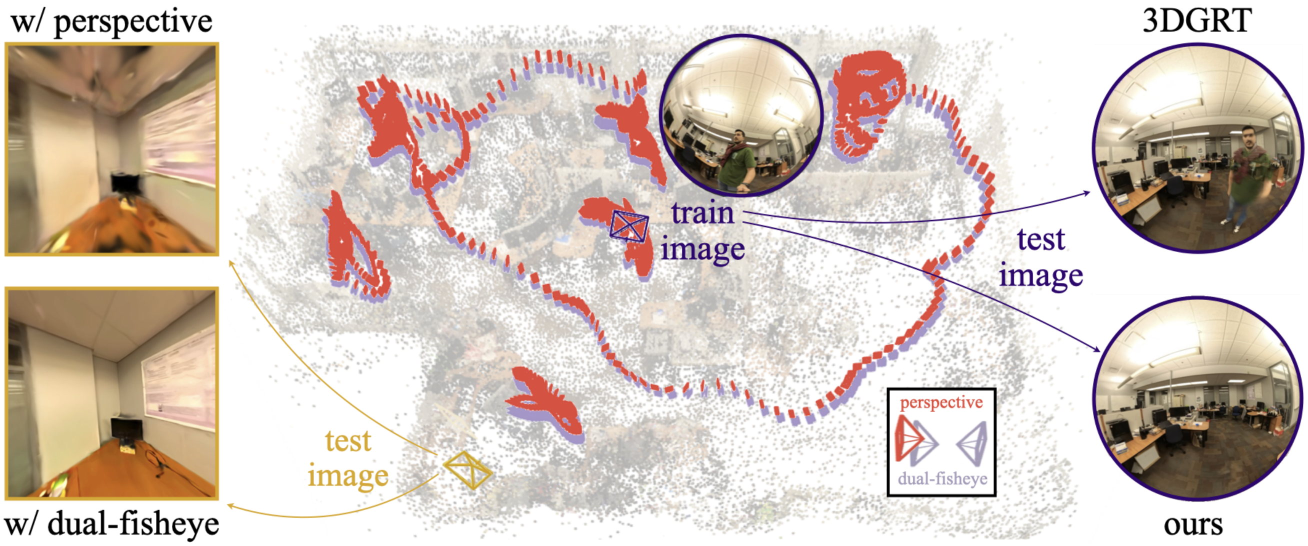


Figure 1. **Teaser** – Using dual-fisheye imagery—common in 360° cameras—we can recover 3D scenes far more effectively than with traditional perspective cameras, resulting in substantial gains in novel view synthesis (left: perspective vs. fisheye). However, 360° training images often include distracting elements, especially the camera operator, which can degrade reconstruction quality if left unaddressed (right: 3DGRT). Our method, FullCircle, overcomes these challenges, enabling fast, high-quality scene capture using 360° cameras.

Abstract

Radiance fields have emerged as powerful tools for 3D scene reconstruction. However, casual capture remains challenging due to the narrow field of view of perspective cameras, which limits viewpoint coverage and feature correspondences necessary for reliable camera calibration and reconstruction. While commercially available 360° cameras offer significantly broader coverage than perspective cameras for the same capture effort, existing 360° reconstruction methods require special capture protocols and pre-processing steps that undermine the promise of radiance fields: effortless workflows to capture and reconstruct 3D scenes. We propose a practical pipeline for reconstructing 3D scenes directly from raw 360° camera captures. Our pipeline requires no special capture protocols or pre-processing, and exhibits robustness to a prevalent source of reconstruction errors: the human operator that is visible in all 360° imagery. To facilitate evaluation, we introduce a multi-tiered dataset of scenes captured as raw dual-fisheye images, establishing a benchmark for robust casual 360° reconstruction. Our method significantly outperforms not only vanilla 3DGS for 360° cameras but also robust perspective baselines when perspective cameras are simulated from the same capture, demonstrating the advantages of 360° capture for casual reconstruction.

1. Introduction

Recent advances in radiance fields [18, 31] have turned 3D reconstruction from a tech demo into practical tools for real-world capture (e.g. deployed products include Niantic's Scaniverse, Polycam's 3D scanner, KIRI's Engine, and Meta's HyperScapes). Scenes reconstructed with radiance fields are not only photorealistic, but also fast to

*Equal contribution

train/render [18, 33], robust to outliers [28, 54], and scalable to large environments [26, 45]. This surge of progress has led to growing efforts in dataset capture for building libraries of 3D reconstructed scenes [8, 51] and objects [35].

Despite the focused effort on building such 3D reconstructable datasets, most capture pipelines rely on perspective cameras (smartphones or DSLRs). While these are convenient for small-scale casual captures, they are fundamentally inefficient for large-scale 3D dataset collection. This is because each image provides a narrow field of view, requiring dense trajectories and redundant coverage – approximately 10× to 30× images depending on desired resolution – to achieve sufficient multi-view overlap. In contrast, 360° cameras can observe the entire scene from a single viewpoint, providing an order-of-magnitude increase in coverage with the same capture effort. For data-hungry methods like 3DGS, this richer coverage provides stronger multi-view constraints for camera calibration and denser observations, leading to more stable radiance field optimization.

Towards this promise, recent works have adapted the 3DGS rasterization workflow to handle *distorted projections* from equirectangular or fisheye inputs [23, 25, 42, 53]. However, a key unresolved challenge is ensuring robustness when working with "real" human-captured 360° scenes. Because 360° cameras are all-seeing, they inevitably *include the human capturer* (i.e. the camera operator) and their shadow, as well as other potential transient distractors [38]. Unless carefully handled, this results in severe noisy artifacts in the reconstructed geometry; see Figure 1. As a result, robust and scalable 360° 3D reconstruction pipelines remain elusive.

We introduce a robust 360° capture and reconstruction pipeline that effectively addresses these challenges. Our pipeline removes the human capturer, while maintaining consistent scene geometry and appearance. We demonstrate that, with identical capture trajectories and compared to perspective cameras, 360° cameras yield: (i) significantly higher reconstruction quality, and (ii) more reliable camera calibration. The advantage is especially pronounced for out-of-distribution novel views (i.e. rendered far from the training trajectory). To show the effectiveness of our method, we collect 9 diverse 360° scenes covering a wide range of conditions (indoor/outdoor, lighting, distractor density), and a dedicated test-set that is free of distractors. We evaluate our approach against both perspective-based and 360° robust 3DGS baselines. Our results show that our method enables high-quality and robust 3D reconstruction directly from casual 360° captures. Our collected dataset provides a focused testbed for robust 360° reconstruction, and our pipeline, both of which we will release publicly, enables scalable 360° data capture for large-scale dataset creation.

2. Related Work

We first briefly review related work on multi-view 3D reconstruction with radiance fields then discuss recent works that focus on non-pinhole (fisheye, 360) cameras. We also discuss works that consider distractors in the scene.

Radiance fields. Multi-view 3D reconstruction has been completely reshaped since the introduction of Neural Radiance Fields (NeRF) [31] and 3D Gaussian Splatting (3DGS) [18]. Both of these methods model radiance fields, from which then images are rendered through volume rendering. NeRFs model the radiance field as a neural field [11], while 3DGS methods use explicit 3D Gaussians which are rasterized into a desired view efficiently. Since the introduction of 3DGS, it has been quickly preferred over NeRFs, due to its much faster inference, and explicit nature. Various methods have been proposed to extend NeRFs and 3DGS, including those that enhance efficiency [22, 32], robustness to transient distractors [28, 54], initialization [19], and even extending to dynamic scenes [27, 50]. In terms of camera compatibility, original 3D Gaussian Splatting is restricted to perspective cameras due to its rasterization pipeline, whereas NeRF is more flexible thanks to its ray-tracing formulation. However, NeRF models for fisheye cameras remain prohibitively slow in practice.

Non-pinhole cameras and 3DGS. While NeRF-based methods exist [3, 5, 14, 21], here we focus on 3DGS-based methods [1, 23, 24, 49] capable to optimize radiance fields from datasets consisting of fisheye and 360 images.

Several works [15, 44, 48, 55] undistort 360° images into cubemap faces for reconstruction and depth estimation. Others [1, 23, 24, 49] adapt the 3DGS rasterization to spherical images by approximating 360° projection with modified perspective rasterization. Conversely, 3D-Gaussian Ray Tracing (3DGRT) [53] introduces a ray-tracing formulation for Gaussian primitives, enabling accurate rendering for non-pinhole cameras such as fisheye and 360°.

The Gaussian Unscented Transform (3DGUT) by Li et al. [25] further extends this by introducing unscented transformations, so to properly rasterize 3D Gaussians when lens distortion is present, bringing the ability to not just train, but also render the model without loss of fidelity. Finally, Seam360GS [42] performs seam-aware and exposure-aware 360-degree Gaussian splatting to remove stitching artifacts and photometric inconsistencies in real-world omnidirectional captures. While the primary focus of these methods is to improve the quality of reconstructions, they assume ideal *distraction-free* captures. With 360 cameras, this is clearly never the case, as in casual captures the person holding the camera is *unavoidably* in view.

Reconstructing with distractors. Unless in a controlled environment, reconstruction with NeRFs and 3DGS must account for distractors. In the wild, even if carefully hid-

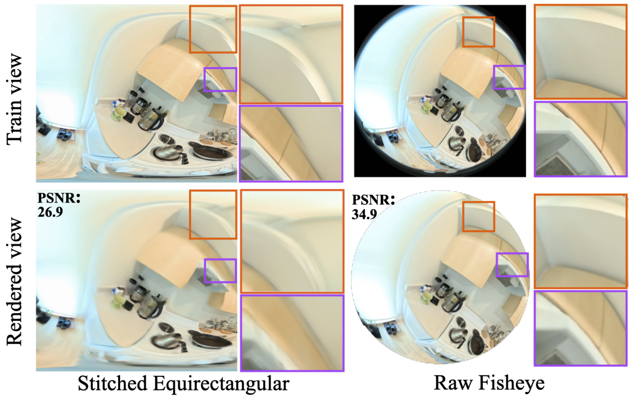


Figure 2. **Stitching artifacts** – Stitched equirectangular inputs contain stitching artifacts (top left) that lead to noisy edges and reduced PSNR in the reconstruction (bottom left). Our method, trained on raw fisheye images, avoids these artifacts (right).

ing behind the camera, the capturer can occasionally cast shadows within the scene, or photograph another person unexpectedly appearing within the frame while attempting to capture a static scene. Robustness to transient distractors and illumination changes has been explored in NeRFs [2, 28, 38]; NeRF-W's per-view embedding was later extended to 3DGS [7, 9, 16, 52], alongside distractor-removal methods [54] inspired by RobustNeRF [38]. In our work, we are interested more in the case specific to the 360 capture setup, where the distractor is the capturer, a highly practical and common scenario for 360 cameras. By focusing on this particular scenario, we are able to tailor our solution to build a more robust pipeline.

Dataset collection efforts. Radiance-field reconstruction datasets range from controlled captures like ScanNet++ [8], designed for accurate calibration and reconstruction, to casual datasets using handheld perspective videos [12, 37, 38, 43]. Existing 360° datasets such as ODIN [47], EgoNeRF [5], 360Roam [14], and FIORD [13] employ controlled capture setups or provide pre-stitched panoramas. To our knowledge, no public dataset offers casual, handheld 360° captures from dual-fisheye cameras; we therefore collect one across diverse scenes to facilitate analysis and benchmarking of reconstruction from casual 360 captures.

3. Method

Narrow field of view capture imposes multiple constraints, which the user needs to respect to acquire high-quality input data for training radiance fields [10] For example, according to the COLMAP tutorial [39]: (i) texture-less images (e.g. a white wall) should be avoided, as well as (ii) specularities introduced by shiny surfaces, (iii) images should be taken so to ensure high visual overlap, with each object observed in at least three views, and (iv) enough images should be taken from a relatively similar viewpoint, and yet near duplicates and pure camera rotations should be avoided (i.e., take a few steps after each photograph is taken). Clearly, such constraints are impractical for casual users. In contrast, professional reconstruction tools (e.g. Meta Horizon Hyperscape [29, 30]), often incorporate AR-based guidance systems that actively assist users in maintaining coverage, overlap, and scene diversity during capture.

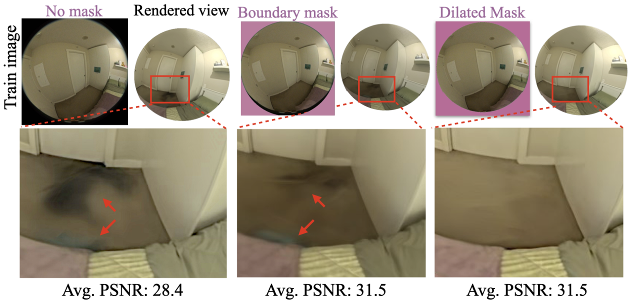


Figure 3. **Boundary masking** – Black boundary pixels contaminate the reconstruction when unmasked (left), while minimal masking leaves blue artifacts from edge distortion and color aberration (middle); our dilated mask prevents these artifacts (right).

Data capture. In contrast to a traditional radiance field capture and reconstruction process, which depends on thousands of manually placed perspective photographs, our capture pipeline is built on a casual video capture from a 360° camera. In order for the process to be employed at scale by non-expert users, our pipeline must not require any tedious, methodical camera placement, and must be robust to a variety of behaviors from the person performing the capture, including cases where the capturer does not actively cover all viewing directions, as well as extended periods where they may remain stationary, such as while capturing close-up images of a small area of interest in high details. These requirements present unique challenges in the context of 360° capture, as "the capturer" will be visible at all times, and must be removed from the final scene reconstruction.

Dataset. To evaluate our technique for casual capture, we capture a dataset of 9 scenes with a single consumer-grade Insta360 X4 360° (dual-fisheye) camera, readily available through mainstream retail channels, to reflect its widespread adoption among casual users. During these captures, we emulate non-expert capturers behaviors, including both periods of motion where the person actively moves around the scene, as well as periods where the person remains static, and only moves the camera. For statistics on the lengths of these captures, including the static and dynamic parts, as well as other details, please refer to Table 1. For the purpose of quantitative evaluation, we also capture a golden test set of images for each scene using a *tripod*, where no people are visible, and the visible parts of the tripod are masked

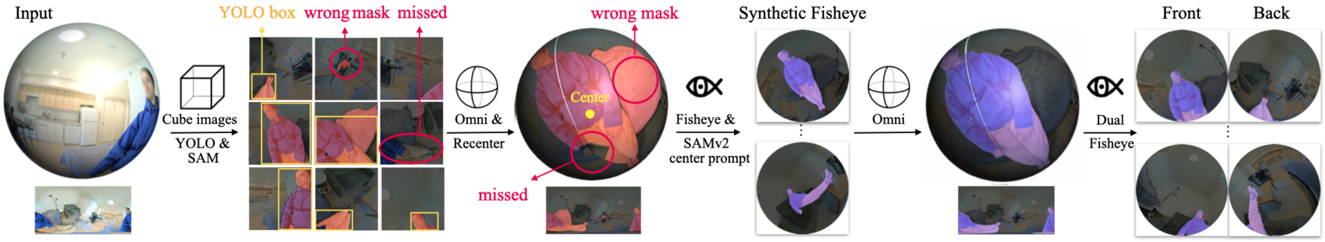


Figure 4. **Capturer mask estimation** – Our masking pipeline first detects the capturer using YOLOv8 [17] and SAMv2 [34] (red masks). These initial masks roughly localize the capturer but include missing or incorrect regions that can degrade reconstruction. We then re-center the omni images on the detected capturer, render synthetic fisheyes, and re-run SAMv2 with a center-point prompt to obtain refined, temporally consistent masks (purple masks), which are mapped back to the original dual-fisheye inputs.

out. This enables unbiased measurement of the novel view quality from views which are not sampled from the training trajectory, and which include areas that were occluded by the capturer during the video.

Raw fisheye input. The manufacturer-provided software with the 360° camera by default outputs *stitched* omni-directional/panorama images in *equirectangular* projection; see Fig. 2-(left). However, using these stitched images directly for training is problematic due to the inevitable *optical misalignment* between the two original fisheye views in consumer cameras. This misalignment results in artifacts around the stitching boundary, and thus results in lower quality reconstructions when used as ground truth; see Figure 2. Instead, we extract the original (un-stitched) fisheye frames, and use these as input to training. While these frames do not show any artifacts from stitching, they do show *distortion* and *chromatic aberration* effects around the outer boundary of the camera's field of view. We mask out these pixels with a boundary mask $\mathcal{M}_B$, before continuing to camera estimation and reconstruction; see Figure 3.

3.1. Capturer mask estimation

Because the training views include the person holding the camera during capture, we generate masks to remove the person before proceeding to reconstruction. Rather than treating the presence of the capturer as a nuisance, we take advantage of this consistency and propose a simple, robust masking strategy that leverages the fact that the capturer *appears in all views*, and *moves continuously* through time. This stands in contrast to previous perspective-robust methods that rely on analyzing photometric error during reconstruction to infer masks [20, 38, 54]. To identify pixels belonging to the capturer, we employ a two-stage masking process. First, we estimate the approximate location of the person, and then we generate a synthetic fisheye view centered on them which is used to robustly estimate the final mask. The entire process illustrated in Fig. 4.

Finding the capturer. We start by extracting a set of 16 overlapping virtual 90° pinhole camera views that cover the sphere. Independently for each of these frames, a mask is predicted using SAMv2 [34] at the location of the capturer in the frame predicted by YOLOv8 [17] with a "person" prompt. An approximate global direction for the capturer is then obtained as the *average direction* of every pixel in each pinhole frame, which was classified as part of the capturer. Synthetic 180° fisheye frames centered on this direction are then generated from the omni-directional images.

Segmenting capturer (through time). After synthesizing fisheye images such that the capturer remains centered, we can automatically prompt SAMv2 [34] by selecting the image center as the prompt location. This setup allows us to exploit temporal consistency, as mask propagation through time becomes robust when the capturer stays near the image center. We further leverage the observation that auto-stitching artifacts, though detrimental to 3DGS reconstruction, remain largely tolerable for pre-trained video segmentation models like SAMv2 [34]. To account for uncertainties in the segmenter predictions, we also dilate the predicted final masks by a negligible amount (4 pixels). After intermediate conversions between omnidirectional, synthetic fisheye, and cubemap representations, the final capturer masks are propagated back to the original fisheye domain, with all RGB data preserved in their raw form throughout the process. We refer to capturer masks as $\mathcal{M}_C$.

Handling shadows. In addition to parts of the scene directly occluded by the capturer, secondary effects caused by the capturer can also introduce noise into the reconstruction. The most prominent of these is the capturer's shadow, which can significantly affect reconstruction quality if not properly masked. Since video segmentation models such as SAMv2 typically do not account for secondary effects like shadows, we design an auxiliary method to explicitly handle them. We assume no hard lighting conditions that produce

This capture procedure mirrors an expert capture session, which requires much longer capture time, and deliberate choices during capture

This procedure relies on the person detector correctly localizing the capturer in at least a few frames, with minimal false positives, and on the capturer being the only human distractor present in most frames.

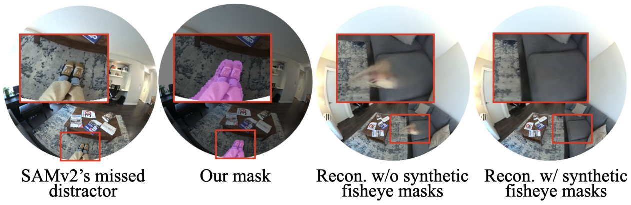

SAMv2's missed distractor
Our mask
Recon. w/o synthetic fisheye masks
Recon. w/ synthetic fisheye masks

Figure 5. **Synthetic fisheye** – Without re-centering the fisheye images synthetically, SAMv2 [34] misses the capturer in some frames, resulting in a noisy reconstruction.

long cast shadows. To this end, we define a luminance-based mask M_L, where each pixel is activated if the predicted RGB luminance exceeds that of the corresponding ground-truth image; such that $\mathcal{M}_L = \mathcal{I}(L^*_{\text{pred}} > L^*_{\text{gt}})$, where $\mathcal{I}$ is the indicator function. Here, L^* denotes the perceptual lightness component of the pixels in the image in the CIELAB color space, computed from sRGB values via the standard RGB→XYZ→Lab conversion [6]. Applying M_L globally across the entire image could suppress not only capturer's cast shadows but also specular highlights or view-dependent brightness variations. To avoid this, we restrict shadow removal to the vicinity of the capturer by intersecting $\mathcal{M}_L$ with a dilated capturer mask $\hat{\mathcal{M}}_C$, yielding the localized shadow mask $\hat{\mathcal{M}}_L = \mathcal{M}_L \wedge \hat{\mathcal{M}}_C$. The final mask is then obtained as the logical union of all components, denoting invalid regions: $\mathcal{M}_{\text{final}} = \mathcal{M}_B \vee \mathcal{M}_C \vee \hat{\mathcal{M}}_L$.

3.2. Training the radiance field

After having masked input fisheye images, we continue with a mostly standard 3DGS training process. We estimate camera poses with COLMAP [40, 41], using raw front and back fisheye images, and generate a SfM point cloud to initialize 3DGS training. The composed masks are used in both steps to avoid failures due to the presence of the capturer in the training images; see Sec. 4.1 and Tab. 1. Classical 3DGS training undistorts fisheye images with COLMAP before training. Conversely, we build upon 3D Gaussian Ray Tracing (3DGRT) [53], which is capable to render non-pinhole images, and instead use the masked fisheye frames for training. Importantly, we show that using raw fisheye images with 3DGRT achieves higher reconstruction quality compared to training a 3DGS model on undistorted images, due to its higher coverage; see Fig. 7. The 3DGRT model is trained on all views from the camera trajectory which were successfully estimated by COLMAP, with the separately captured tripod views used as a test set.

4. Experiments

Table 1. We collect a dataset of dual-fisheye scene captures with different levels of difficulty. We report the number of frames where the capturer is static/dynamic, and the number of COLMAP pose failures (w/ and w/o transient masks as MCF and UCF). The full image set is released to support future work on robust calibration.

Scene	#training	#test	#total	#MCF	#UCF	duration	difficulty
Room 1	442 / 117	38	597	–	–	88s / 23s	Easy
Flat 1	440 / 153	46	639	–	–	88s / 31s	Easy
Flat 2	434 / 107	26	567	–	–	87s / 21s	Easy
Room 2	252 / 82	28	362	–	–	50s / 16s	Easy
Room 3	257 / 100	30	387	1	1	51s / 20s	Medium
Lab	431 / 116	34	581	–	–	86s / 23s	Medium
Lounge	413 / 120	22	555	–	555	83s / 24s	Medium
Persons	448 / 129	34	611	110	611	90s / 26s	Hard
Shadow	434 / 113	35	582	–	–	87s / 23s	Hard

We first show the effectiveness of dual-fisheye capture compared to conventional perspective capture under similar camera trajectories (Sec. 4.1). We show that 360° cameras are inherently more suitable for casual scene acquisition due to their wider coverage and consequently easier camera calibration. Next, we discuss why undistorting 360° images before reconstruction is suboptimal, showing using raw fisheye inputs yields higher-fidelity reconstructions (Sec. 4.2). We then show that, when operating directly on raw fisheye images, our method effectively suppresses the distractor capturer and reconstructs high-quality scenes, outperforming baselines designed for fisheye inputs (Sec. 4.3). Finally, we ablate our pipeline design in (Sec. 4.4).

Implementation details. Our implementation is based on 3DGRT [53], trained with the Adam optimizer and default learning rates for 30k iterations. 3DGS perspective baselines follow their official implementations, and NeRF baselines use the Nerfacto [46] model, with robust extensions and different camera models integrated. Camera calibration (for both fisheye and undistorted fisheye) and image undistortion are performed using COLMAP v3.12 [40, 41]. Unless otherwise specified, we use images downsampled by a factor of 4× for faster processing and reduced memory usage. Human detection is done with YOLOv8s [17], and segmentation with mask propagation in SAMv2.1 [34] (large hierarchical backbone) model. The capturer's mask is dilated by 80 pixels to get $\hat{\mathcal{M}}_C$.

4.1. Dual-fisheye vs. perspective – Fig. 6

One of our main claims is that dual-fisheye cameras provide a more suitable setup for casual data capture than conventional perspective cameras for 3D reconstruction. To illustrate this, we conduct a controlled experiment comparing camera calibration as well as reconstruction quality between fisheye and perspective imagery. The images are captured along identical camera trajectories to emulate two comparable 80-second casual video captures by a non-expert user.

Data. We collect a dataset of 191 dual-fisheye images corresponding to approximately 80 seconds of casual, handheld capture using a dual-fisheye camera. However, the camera is operated like a classical perspective camera, always directed toward scene structures of interest (there are no human distractors in the front fisheye view). To ensure perfectly consistent exposure and color balance between

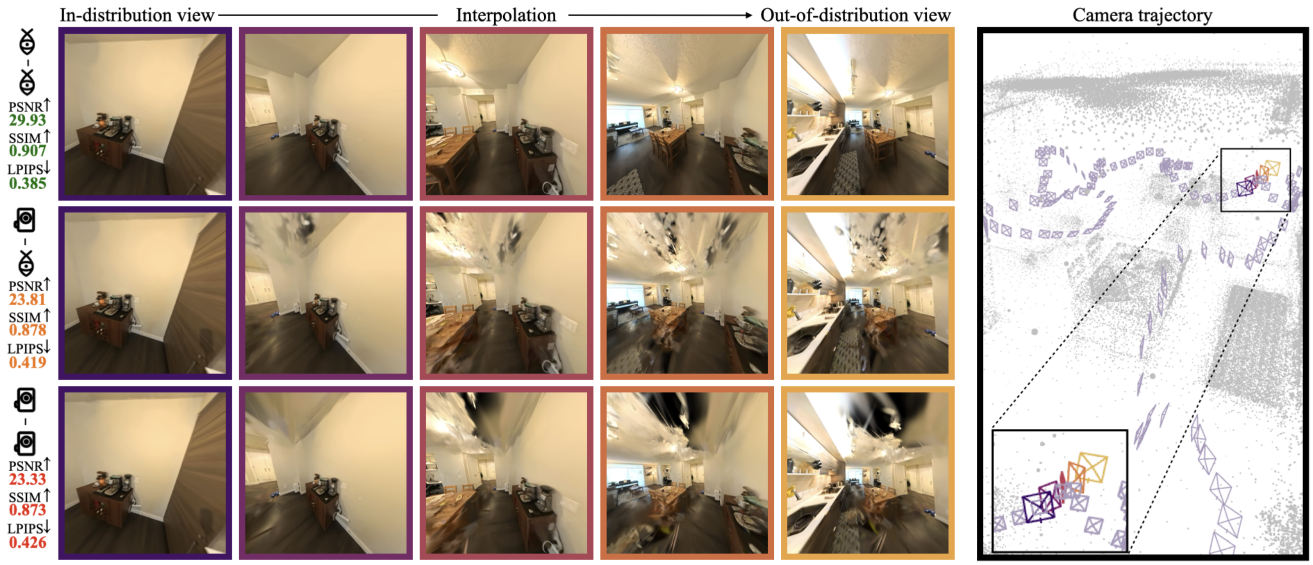


Figure 6. **Dual-fisheye vs. perspective capture** – Using dual-fisheye images for both calibration and reconstruction (👁–👁) yields higher-quality novel views and stable geometry than when reconstruction (👁–📷), or both calibration and reconstruction (📷–📷), relies on perspective images. This advantage becomes most apparent when moving beyond the in-distribution training trajectory (purple cameras) toward out-of-distribution test views (yellow camera), where (👁–📷) and (📷–📷) exhibit a gradual degradation in reconstruction quality.

fisheye and perspective settings, we do not perform a separate perspective capture; instead, in this controlled experiment, we use undistorted images from the front fisheye to simulate perspective views.

Evaluation variants. We evaluate three different settings:

- (👁–👁): both calibration and reconstruction are performed directly on the raw dual-fisheye images using COLMAP followed by our robust reconstruction pipeline, with no undistortion preprocessing.
- (👁–📷): camera calibration is performed on the raw dual-fisheye images, but reconstruction uses only the undistorted perspective images from the front camera.
- (📷–📷): calibration and reconstruction are performed on the front camera's undistorted perspective images.

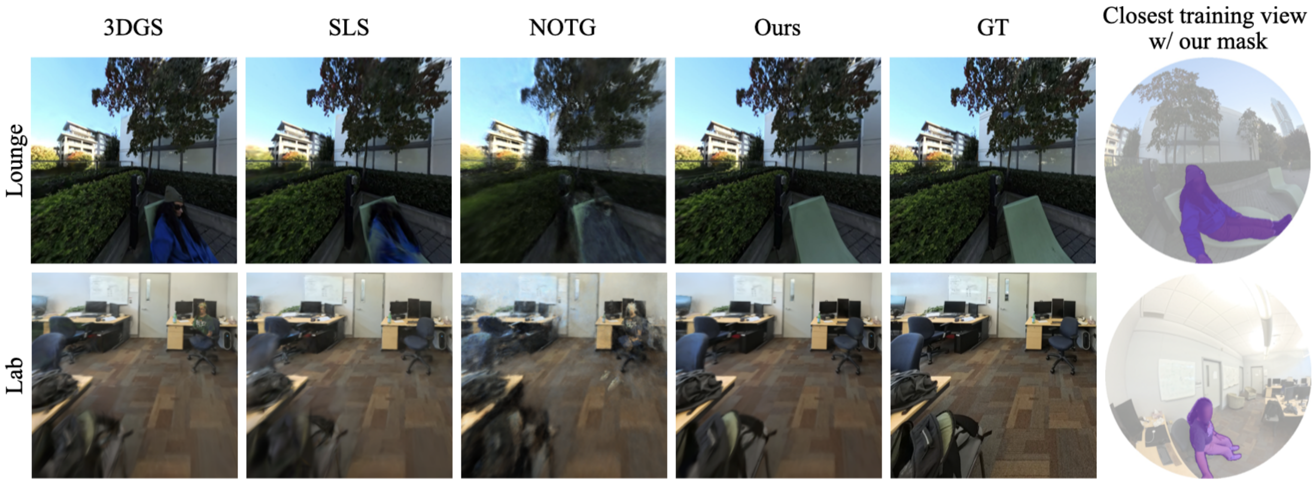
Metrics. We assess both calibration and reconstruction quality through novel view synthesis performance, reporting PSNR on full-resolution images of 2880 × 2880 as a quantitative image quality metric. We assess the reconstruction quality at novel views, outside the distribution of training camera views. We report PSNR, SSIM and LPIPS.

Analysis. As shown in Fig. 6, all three pipelines produce high-quality reconstructions near the training trajectory, but performance diverges as the viewpoint moves toward out-of-distribution test views. The (👁–👁) pipeline maintains stable quality across this trajectory, while (👁–📷) and (📷–📷) degrade significantly. Notably, (👁–📷) consistently outperforms (📷–📷), even though the same set of undistorted perspective images is used for reconstruction. This indicates that calibration performed on raw dual-fisheye images is more accurate and has a direct impact on 3D reconstruction robustness. Overall, this experiments confirms that, if we were able to remove the capturer from input images, fisheye images are significantly more effective than perspective images in reconstructing scenes with high fidelity (better calibration, and better scene coverage). We now, therefore, shift our focus to dual-fisheye data.

4.2. Reconstruction via undistortion – Fig. 7

While performing capture and calibration in the dual-fisheye domain offers the benefits shown in the previous section, reconstruction from such images can be approached in different ways. A common strategy is to first undistort the front and back fisheye views and then perform reconstruction on the resulting perspective images. This conveniently enables the use of robust perspective-based methods [37, 54] designed to handle distractors. To highlight the advantages of reconstructing directly from dual-fisheye inputs, we compare our pipeline, which is trained on raw fisheye images from our dataset, against perspective baselines trained on undistorted front and back views. We further evaluate the robustness of these perspective baselines in the presence of the capturer, compared to our method.

Dataset. Due to the lack of publicly available dual-fisheye datasets captured casually with the human capturer visible, we collected our own dataset of nine scenes (eight indoor and one outdoor) spanning a range of reconstruction difficulties. Detailed statistics are provided in Table 1. The videos are recorded at 5 FPS and a resolu-



Method	Robust	Room 1	Flat 1	Flat 2	Room 2	Room 3	Lab	Lounge	Persons	Shadow	**Mean**
3DGS [18]	✗	26.8	24.3	23.3	25.7	23.9	24.6	20.6	24.7	24.1	24.2
SLS-MLP [54]	✓	29.9	25.2	26.3	28.4	25.5	25.6	20.8	26.9	**28.2**	26.3
NOTG [37]	✓	25.1	20.3	23.4	22.4	21.9	22.5	20.1	26.3	24.3	22.9
FullCircle (Ours)	✓	**31.6**	**26.9**	**28.4**	**30.6**	**28.3**	**27.8**	**22.5**	**28.9**	27.2	**28.0**

Figure 7. **Effect of undistortion** – Using raw fisheye images with FullCircle yields higher-quality reconstructions, both qualitatively and quantitatively, compared to perspective baselines—robust and non-robust—that operate on preprocessed, undistorted images.

tion of 2880 × 2880 pixels using an Insta360 X4 dual-fisheye camera. Easy scenes correspond to typical indoor captures, medium scenes include view-dependent effects, specular surfaces, and long-horizon outdoor environments, while hard scenes feature long-cast shadows and multiple human distractors. The dataset will be publicly released to support future research. For experiments in this section, the front and back fisheye images are undistorted using COLMAP [40, 41], and then used as input data to reconstruction. All baselines use poses estimated from the dual-fisheye images using COLMAP.

Metrics. We report PSNR as the main novel-view synthesis metric on perspective renderings from the test camera views. Test fisheye images are captured from a tripod to minimize distractor interference. The tripod body is masked out when computing evaluation metrics. SSIM and LPIPS are included in the supplementary material.

Baselines. We compare against 3DGS [18] and its robust variant SpotLessSplats [54] for perspective-based gaussian splatting baselines. From SLS, we use its MLP variant trained with Stable Diffusion features for distractor segmentation. We also provide comparison against NeRF On-the-go (NOTG) [37] as the state-of-the-art robust NeRF method. We re-implement NOTG on top of Nerfacto [46] for faster training and rendering. All methods are trained on undistorted front and back images from the 360 camera, except ours which is trained on raw fisheye images.

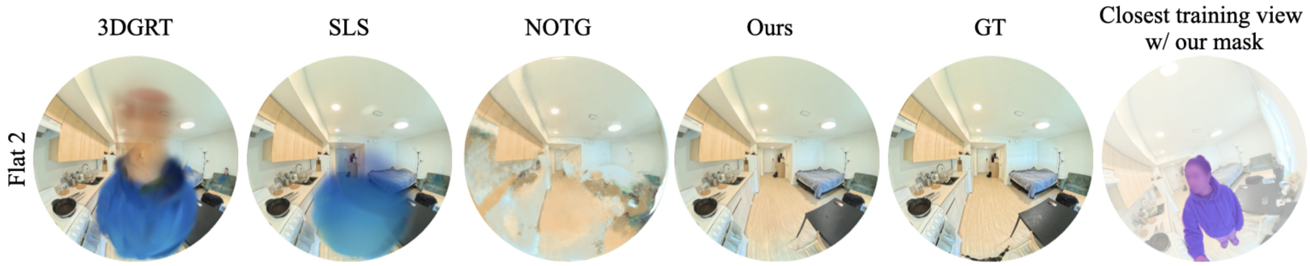
Analysis. 3DGS is a non-robust baseline and frequently reconstructs the human capturer as noisy distractor artifacts. SLS-MLP [54] improves robustness to distractors but often exhibits visible artifacts due to reduced coverage and distortions introduced during the undistortion and resampling phase (Fig. 7, second row). In scenes where the capturer remains static for extended periods, SLS also tends to partially reconstruct the distractor itself (Fig. 7, first row). In contrast, our method produces clean, high-fidelity reconstructions without visible distractor artifacts and outperforms baselines quantitatively across most scenes. Remaining failure cases primarily arise in challenging settings with long-cast shadows or multiple moving distractors, where methods such as SLS perform better.

4.3. Dual-fisheye reconstruction – Fig. 8

Following Sec. 4.2, we focus on methods capable of reconstructing directly from fisheye images without undistortion preprocessing, and can achieve higher reconstruction fidelity. We evaluate our 3D reconstruction pipeline that works on raw dual-fisheye casual captures, on our collected dataset, where the capturer is present in all frames, either in the front or back or both cameras, emulating a non-expert capture. We compare our method to other robust and non-robust methods that operate on raw fisheye images.

Dataset. We use the same dataset introduced in Sec. 4.2, following the same train/test split. All images are calibrated using COLMAP on the dual-fisheye inputs.

Metrics. We evaluate all methods on clean test fisheye images captured from a tripod, with the tripod body masked out. PSNR results are reported in Fig. 8. SSIM and LPIPS



Method	Robust	Room 1	Flat 1	Flat 2	Room 2	Room 3	Lab	Lounge	Persons	Shadow	Mean
3DGRT [53]	✗	27.4	26.8	25.1	27.9	28.4	26.7	23.4	27.6	25.7	26.5
SLS (MLP) [54]	✓	29.5	28.7	28.0	31.3	29.1	28.6	23.7	29.1	26.7	28.3
NOTG [37]	✓	24.2	19.4	19.2	27.7	22.1	17.4	19.8	24.6	20.9	21.7
FullCircle (Ours)	✓	**32.6**	**29.1**	**30.5**	**31.6**	**29.9**	**29.5**	**24.4**	**31.0**	**27.8**	**29.6**

Figure 8. **Fisheye reconstruction** – Our method reliably removes human distractors and reconstructs high-quality 3D scenes. The table shows PSNR metric, where out method outperforms both robust and non-robust radiance-field baselines designed for fisheye images.

are included in the supplementary material.

Baselines. We compare against 3DGRT [53]: a non-robust Gaussian Splatting baseline that supports fisheye cameras. For a robust variant, we augment 3DGRT with SpotLessSplats (SLS) [54] and use its MLP variant. Additionally, we include a NeRF-based robust baseline, since NeRFs natively support fisheye camera models (ray-based training). Specifically we use Nerf On-the-go (NOTG) [37], reimplemented on Nerfacto [46] base model for faster convergence and rendering. In all methods, the outer ring of the fisheye images is masked out.

Analysis. Our method outperforms all fisheye-compatible baselines both qualitatively and quantitatively. The vanilla 3DGRT baseline [53] fails to handle dynamic distractors, frequently reconstructing the human capturer as noisy artifacts. Adding the SLS robust loss [54] improves results, but struggles when the capturer remains momentarily immobile, failing to separate them from the static background and leaking into reconstruction early-on in training. NOTG often overestimates uncertainty maps when paired with a fisheye camera model, leading to under-reconstruction in regions near the capturer. In contrast, our method explicitly models the persistent presence of the human capturer across frames and reliably segments them out, producing clean, high-fidelity reconstructions even in complex or reflective environments. For the hard scenes, our method occasionally fails to suppress long cast shadows or reconstructs partial geometry of other moving distractors; see Fig. 10.

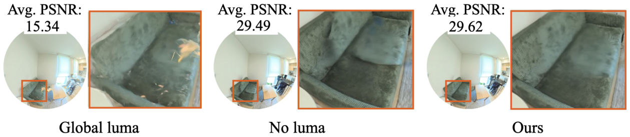


Figure 9. Local luminance mask helps remove distractor's shadow artifacts, while global luminance mask results in global brightness and under-reconstruction.

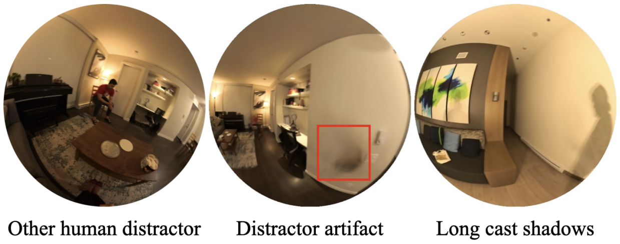


Figure 10. **Limitations** – Our pipeline struggles with additional human distractors (left) and long cast shadows (right).

4.4. Ablation study – Fig. 9

We ablate our design choices for incorporating the luminance mask. Applying it globally alters the overall brightness of the reconstructed scene and often over-masks valid regions, producing floaters and under-reconstruction. Conversely, omitting the luminance mask allows the capturer's shadow to persist as a noisy artifact, degrading PSNR. Our localized luminance masking avoids both failure modes. SSIM and LPIPS are reported in supplementary material.

5. Conclusion

We present a robust pipeline for 3D reconstruction from casually captured 360° images using a consumer-grade dual-fisheye camera. We demonstrate that data collection for reconstruction can be performed more efficiently with dual-fisheye imagery than with commonly used perspective cameras due to their wider angular coverage. To leverage this advantage for scalable dataset collection, we address the central challenge of casual 360° capture—the always-visible capturer that violates photometric consistency—by reliably locating and masking capturer for high-quality reconstruction. To benchmark this, we collect a dataset that provides a focused testbed for robust 360° reconstruction.

While effective, our pipeline currently assumes fixed exposure and does not handle abrupt brightness changes, extreme shadows, or multiple moving distractors; see Fig. 10. Nevertheless, it enables robust reconstruction across a wide range of casual captures, paving the way for scalable 360° reconstructable datasets and training feed-forward reconstruction networks from 360° imagery [4, 36].

References

[1] 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

[2] Jiahao Chen, Yipeng Qin, Lingjie Liu, Jiangbo Lu, and Guanbin Li. Nerf-hugs: Improved neural radiance fields in non-static scenes using heuristics-guided segmentation. In *CVPR*, 2024. 3

[3] 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. 2

[4] Zheng Chen, Chenming Wu, Zhelun Shen, Chen Zhao, Errui Ding, and Song-Hai Zhang. Splatter-360: Generalizable 360○ gaussian splatting for wide-baseline panoramic images. In *CVPR*, 2025. 9

[5] Changwoon Choi, Sang Min Kim, and Young Min Kim. Balanced spherical grid for egocentric view synthesis. In *CVPR*, 2023. 2, 3

[6] Commission Internationale de l'Eclairage. Colorimetry. Technical report, 1976. 5

[7] Hiba Dahmani et al. Swag: Splatting in the wild images with appearance-conditioned gaussians. *ECCV*, 2024. 3

[8] Angela Dai, Daniel Ritchie, and Matthias Niessner. Scannet++: A high-fidelity dataset of 3d indoor scenes. *CVPR*, 2024. 2, 3

[9] Zhang Dongbin et al. Gaussian in the wild: 3d gaussian splatting for unconstrained image collections. *arXiv preprint arXiv:2403.15704*, 2024. 3

[10] Radiance Fields. How to capture gaussian splatting. `https://radiancefields.com/how-to-capture-radiance-fields`, 2024. 3

[11] Kyle Gao, Yina Gao, Hongjie He, Dening Lu, Linlin Xu, and Jonathan Li. Nerf: Neural radiance field in 3d vision, a comprehensive review. *arXiv preprint arXiv:2210.00379*, 2022. 2

[12] L. Goli, S. Sabour, M. Matthews, M. Brubaker, D. Lagun, A. Jacobson, D. J. Fleet, S. Saxena, and A. Tagliasacchi. RoMo: Robust Motion Segmentation Improves Structure from Motion. In *ICCV*, 2025. 3

[13] Ulas Gunes, Matias Turkulainen, Xuqian Ren, Arno Solin, Juho Kannala, and Esa Rahtu. Fiord: A fisheye indoor-outdoor dataset with lidar ground truth for 3d scene reconstruction and benchmarking. In *Scandinavian Conference on Image Analysis*, 2025. 3

[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

[15] Hyeonjoong Jang, Andreas Meuleman, Dahyun Kang, Donggun Kim, Christian Richardt, and Min H. Kim. Egocentric scene reconstruction from an omnidirectional video. *SIGGRAPH*, 2022. 2

[16] Xu Jiacong et al. Wild-gs: Real-time novel view synthesis from unconstrained photo collections. *arXiv preprint arXiv:2406.10373*, 2024. 3

[17] Glenn Jocher, Jing Qiu, and Ayush Chaurasia. Ultralytics YOLO, 2023. 4, 5

[18] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *SIGGRAPH*, 2023. 1, 2, 7

[19] Shakiba Kheradmand et al. 3d gaussian splatting as markov chain monte carlo. *NeurIPS*, 2024. 2

[20] Jonas Kulhanek, Songyou Peng, Zuzana Kukelova, Marc Pollefeys, and Torsten Sattler. WildGaussians: 3D gaussian splatting in the wild. In *NeurIPS*, 2024. 4

[21] Shreyas Kulkarni, Peng Yin, and Sebastian Scherer. 360fusionnerf: Panoramic neural radiance fields with joint guidance. In *IROS*, 2023. 2

[22] Joo Chan Lee, Daniel Rho, Xiangyu Sun, Jong Hwan Ko, and Eunbyung Park. Compact 3d gaussian representation for radiance field. In *CVPR*, 2024. 2

[23] Suyoung Lee, Jaeyoung Chung, Jaeyoo Huh, and Kyoung Mu Lee. Odgs: 3d scene reconstruction from omnidirectional images with 3d gaussian splattings. *NeurIPS*, 2024. 2

[24] 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

[25] Ming Li et al. 3d-gut: Distortion-aware 3d gaussian unwrapping for 360° reconstruction. *arXiv preprint arXiv:2405.06789*, 2024. 2

[26] Zheng Liu, He Zhu, Xingyang Li, Yirun Wang, Yujiao Shi, Wei Li, Jingwen Leng, Minyi Guo, and Yu Feng. Voyager: Real-time splatting city-scale 3d gaussians on your phone. *arXiv*, 2025. 2

[27] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In *3DV*, 2024. 2

[28] Ricardo Martin-Brualla, Noha Radwan, Mehdi S M Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. *CVPR*, 2021. 2, 3

[29] Inc. Meta Platforms. Meta horizon hyperscape capture (beta) — capture your real-world space and create an immersive digital replica. `https://www.meta.com/experiences/meta-horizon-hyperscape-capture-beta/8798130056953686/`, 2025. 3

[30] Inc. Meta Platforms. Help – how to capture with meta horizon hyperscape (beta). `https://www.meta.com/help/quest/1088536553019177/`, 2025. 3

[31] 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. *ECCV*, 2020. 1, 2

[32] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *SIGGRAPH*, 2022. 2

[33] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *SIGGRAPH*, 2022. 2

[34] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 4, 5, 1

[35] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, et al. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. *ICCV*, 2021. 2

[36] Jiahui Ren, Mochu Xiang, Jiajun Zhu, and Yuchao Dai. Panosplatt3r: Leveraging perspective pretraining for generalized unposed wide-baseline panorama reconstruction. *ArXiv*, 2025. 9

[37] Weining Ren, Zihan Zhu, Boyang Sun, Jiaqi Chen, Marc Pollefeys, and Songyou Peng. Nerf on-the-go: Exploiting uncertainty for distractor-free nerfs in the wild. In *CVPR*, 2024. 3, 6, 7, 8, 2

[38] Sara Sabour, Suhani Vora, Daniel Duckworth, Ivan Krasin, David J. Fleet, and Andrea Tagliasacchi. Robustnerf: Ignoring distractors with robust losses. In *CVPR*, 2023. 2, 3, 4

[39] Johannes L. Schönberger. Colmap tutorial. `https://colmap.github.io/tutorial.html`, 2025. 3

[40] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 5, 7

[41] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *ECCV*, 2016. 5, 7

[42] Changha Shin, Woong Oh Cho, and Seon Joo Kim. Seam360gs: Seamless 360deg gaussian splatting from real-world omnidirectional images. In *ICCV*, 2025. 2

[43] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. *SIGGRAPH*, 2006. 3

[44] Cheng Sun, Chi-Wei Hsiao, Ning-Hsu Wang, Min Sun, and Hwann-Tzong Chen. Indoor panorama planar 3d reconstruction via divide and conquer. In *CVPR*, 2021. 2

[45] Matthew Tancik, Ethan Weber, Ren Ng, Angjoo Kanazawa, Pratul P Srinivasan, and Jonathan T Barron. Block-nerf: Scalable large scene neural view synthesis. *CVPR*, 2022. 2

[46] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, David Mcallister, Justin Kerr, and Angjoo Kanazawa. Nerfstudio: A modular framework for neural radiance field development. In *SIGGRAPH*, 2023. 5, 7, 8

[47] Matthew Wallingford et al. From an image to a scene: Learning to imagine the world from a million 360○ videos. *arXiv preprint arXiv:2412.07770*, 2024. 3

[48] Fu-En Wang, Yu-Hsuan Yeh, Min Sun, Wei-Chen Chiu, and Yi-Hsuan Tsai. Bifuse: Monocular 360 depth estimation via bi-projection fusion. In *CVPR*, 2020. 2

[49] X. Wang et al. op43dgs: Optimized projection for 360° gaussian splatting. In *CVPR*, 2024. 2

[50] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *CVPR*, 2024. 2

[51] Tianhao Wu et al. Gaussianworld: Scaling 3d gaussian splatting to internet-scale datasets. *arXiv preprint arXiv:2407.12345*, 2024. 2

[52] Wang Yuze et al. We-gs: An in-the-wild efficient 3d gaussian representation for unconstrained photo collections. *arXiv preprint arXiv:2406.02407*, 2024. 3

[53] Wei Zhang et al. 3d-grt: 3d gaussian ray tracing for 360° radiance fields. *arXiv preprint arXiv:2403.09876*, 2024. 2, 5, 8, 3

[54] Jiahao Zheng, Jiemin Chen, Yujun Li, Hanxiao Zhao, and John Smith. Spotless splats: Real-time radiance field rendering with clean 3d gaussian splatting. *arXiv preprint arXiv:2401.10968*, 2024. 2, 3, 4, 6, 7, 8

[55] Nikolaos Zioulis, Antonis Karakottas, Dimitrios Zarpalas, and Petros Daras. Omnidepth: Dense depth estimation for indoors spherical panoramas. In *ECCV*, 2018. 2

6. Supplementary materials

The supplementary materials are organized as follows. First, we provide additional **qualitative video results** accessible via the `index.html` file; *please open this file in a modern web browser (e.g., Safari or Google Chrome).*

We then report the corresponding quantitative results (LPIPS and SSIM) for our main experiments. Further, we present additional experiments that complement the descriptions in the main paper. Finally, we present a detailed description of our masking pipeline together with relevant implementation details Sec. 6.4.

6.1. Additional image quality metrics

We compliment our results in Sec. 4.2 and Sec. 4.3 with SSIM and LPIPS metrics. Specifically, we report Tabs. 2 and 3 as additional image quality metrics to Fig. 7 in the main paper, and Tabs. 4 and 5 as an addition to Fig. 8.

Further, we report image quality metrics including PSNR, SSIM and LPIPS in Tabs. 6 to 8 for the ablation study on luminance masking in Fig. 9. While quantitative improvements are generally consistent across scenes with modest gains, the luminance mask provides clearly visible qualitative benefits in suppressing distractor shadows.

6.2. Using alternative tracking models (DINOv3 [a])

In the main paper, we obtain synthetic fisheye masks by automatically prompting SAMv2 [34] using the center of the first synthetic fisheye frame and propagating the mask over time. As an alternative, we also experiment with DINOv3 [a] for segmentation and tracking. However, DINOv3 requires an instance segmentation mask for the first frame, which prevents a fully automatic pipeline compared to our automated point-based SAMv2 prompting. For completeness, we report the DINOv3 results in Tab. 9, but do not adopt this variant as our main approach.

6.3. Performance with unmasked calibration

For the main experiments, we provide COLMAP with both the fisheye frames and their corresponding masks. Without masking, we observed that in some scenes COLMAP extracts features on the capturer (e.g., in Room3 or Persons), which leads to incorrect pose estimates once the person moves. To mitigate this, we apply the masks during calibration. For completeness, we also evaluate COLMAP with unmasked calibration; Tab. 10 shows that masked calibration consistently improves the 3D reconstruction quality.

6.4. Mask conversions

We provide a more complete description of our masking pipeline, including details on converting masks between perspective, fisheye, and omnidirectional camera models required by our method.

Sampling 16 pinhole cameras. For each omnidirectional image obtained from the camera software, we sample 16 virtual 90° pinhole cameras that together cover the full sphere. On each pinhole view, we run YOLOv8 followed by SAMv2 to detect and segment the person. From the resulting mask, we compute the center of mass and the mask area, since small false-positive masks can affect later steps. We then map these quantities back to the omnidirectional and the world space.

Reorienting the omni with the found direction. Once we have the centers of mass and their corresponding areas, we compute a weighted average of these centers in world coordinates to obtain a dominant direction for the distractor. We then reorient the omnidirectional image so that this direction is centered.

Going to fisheye and back. We map the rotated 3D points xyz to a synthetic fisheye frame, treating the rear fisheye as the reference and converting from the omnidirectional image to fisheye via

$$\phi = \arcsin(-y), \qquad \lambda = \operatorname{atan2}(x, -z). \tag{1}$$

$$u = \left(1 + \frac{-\lambda}{\pi}\right)\frac{W}{2}, \qquad v = \left(1 - \frac{2\,\phi}{\pi}\right)\frac{H}{2}, \tag{2}$$

where λ and ϕ represent longitude and latitude, and (u, v) represent omni image coordinates. We then perform segmentation and tracking on these synthetic fisheye frames using SAMv2 (or DINOv3) to obtain fisheye masks. Finally, we map the masks back to the omnidirectional domain.

Going to original fisheye. For each fisheye camera, we recover its viewing directions and rotate them into world coordinates using the extrinsics obtained from a checkerboard calibration between the omnidirectional and fisheye frames. We then convert the resulting 3D points for the front and rear cameras, $\mathbf{x}_f$ and $\mathbf{x}_r$, into spherical coordinates (ϕ_f, λ_f) and (ϕ_r, λ_r). This yields the corresponding pixels in the omnidirectional image, which we map back to the original fisheye frames as

$$u_f = \left(1 + \frac{-\lambda_f + \pi}{\pi}\right)\frac{W}{2}, \quad v_f = \left(1 - \frac{2\phi_f}{\pi}\right)\frac{H}{2},$$
$$u_r = \left(1 + \frac{-\lambda_r}{\pi}\right)\frac{W}{2}, \qquad v_r = \left(1 - \frac{2\phi_r}{\pi}\right)\frac{H}{2}.$$

References

[a] M. Oquab *et al.*, “DINOv3” 2023. 1, 2

Method	Robust	Room 1	Flat 1	Flat 2	Room 2	Room 3	Lab	Lounge	Persons	Shadow	**Mean**
3DGS [18]	✗	0.90	0.83	0.85	0.90	0.85	0.80	0.72	0.82	0.87	0.84
SLS-MLP [54]	✓	0.90	0.83	0.86	0.91	0.86	0.80	0.71	0.83	0.89	0.84
NOTG [37]	✓	0.80	0.69	0.86	0.80	0.80	0.70	0.57	0.84	0.84	0.77
FullCircle (Ours)	✓	**0.92**	**0.86**	**0.89**	**0.93**	**0.88**	**0.83**	**0.75**	**0.86**	**0.90**	**0.87**

Table 2. **Perspective** – SSIM ↑ metric for reconstruction quality using undistorted fisheye images vs. ours.

Method	Robust	Room 1	Flat 1	Flat 2	Room 2	Room 3	Lab	Lounge	Persons	Shadow	**Mean**
3DGS [18]	✗	0.22	0.35	0.26	0.25	0.31	0.30	0.26	0.37	0.30	0.29
SLS-MLP [54]	✓	**0.19**	0.31	0.22	0.23	0.28	0.30	0.28	0.33	**0.25**	0.27
NOTG [37]	✓	0.31	0.50	0.23	0.41	0.34	0.41	0.45	**0.30**	0.28	0.36
FullCircle (Ours)	✓	**0.19**	**0.30**	**0.18**	**0.21**	**0.26**	**0.27**	**0.23**	0.31	**0.25**	**0.25**

Table 3. **Perspective** – LPIPS ↓ metric for reconstruction quality using undistorted fisheye images vs. ours.

Method	Robust	Room 1	Flat 1	Flat 2	Room 2	Room 3	Lab	Lounge	Persons	Shadow	**Mean**
3DGRT [53]	✗	0.94	0.90	0.91	0.93	0.91	0.89	0.85	0.89	0.91	0.90
SLS-MLP [54]	✓	0.94	**0.91**	0.93	0.94	**0.92**	**0.90**	0.85	0.90	**0.92**	0.91
NOTG [37]	✓	0.86	0.73	0.83	0.92	0.86	0.76	0.69	0.84	0.81	0.81
FullCircle (Ours)	✓	**0.95**	**0.91**	**0.94**	**0.95**	**0.92**	**0.90**	**0.87**	**0.91**	**0.92**	**0.92**

Table 4. **Fisheye** – SSIM ↑ metric for reconstruction quality using fisheye images.

Method	Robust	Room 1	Flat 1	Flat 2	Room 2	Room 3	Lab	Lounge	Persons	Shadow	**Mean**
3DGRT [53]	✗	0.10	0.19	0.12	0.14	0.16	0.16	0.12	0.20	0.16	0.15
SLS-MLP [54]	✓	0.09	0.17	0.10	**0.11**	**0.15**	**0.14**	0.12	0.19	**0.14**	0.14
NOTG [37]	✓	0.19	0.47	0.22	0.14	0.23	0.30	0.35	0.25	0.30	0.27
FullCircle (Ours)	✓	**0.08**	**0.16**	**0.09**	**0.11**	**0.15**	**0.14**	**0.10**	**0.18**	**0.14**	**0.13**

Table 5. **Fisheye** – LPIPS ↓ metric for reconstruction quality using fisheye images.

Method	Room 1	Flat 1	Flat 2	Room 2	Room 3	Lab	Lounge	Persons	Shadow	**Mean**
Global luma	17.06	15.22	21.12	23.1	15.47	15.5	5.23	15.87	9.46	15.34
No luma	32.17	29.10	30.39	31.57	29.79	**29.58**	24.27	30.72	**27.81**	29.49
FullCircle (Ours)	**32.64**	**29.11**	**30.55**	**31.63**	**29.91**	29.54	**24.40**	**30.99**	27.79	**29.62**

Table 6. **Luminance masking** – PSNR ↑ metric reported for reconstruction with/without local/global luminance masking.

Method	Room 1	Flat 1	Flat 2	Room 2	Room 3	Lab	Lounge	Persons	Shadow	**Mean**
Global luma	0.80	0.76	0.87	0.88	0.78	0.74	0.50	0.75	0.68	0.75
No luma	**0.95**	**0.91**	**0.94**	**0.95**	**0.92**	**0.90**	**0.87**	**0.91**	**0.92**	**0.92**
FullCircle (Ours)	**0.95**	**0.91**	**0.94**	**0.95**	**0.92**	**0.90**	**0.87**	**0.91**	**0.92**	**0.92**

Table 7. **Luminance masking** – SSIM ↑ metric reported for reconstruction with/without local/global luminance masking.

Method	Room 1	Flat 1	Flat 2	Room 2	Room 3	Lab	Lounge	Persons	Shadow	**Mean**
Global luma	0.22	0.32	0.17	0.21	0.29	0.31	0.67	0.33	0.40	0.32
No luma	0.09	**0.16**	**0.08**	**0.11**	**0.15**	**0.14**	**0.10**	**0.18**	**0.14**	**0.13**
FullCircle (Ours)	**0.08**	**0.16**	0.09	**0.11**	**0.15**	**0.14**	**0.10**	**0.18**	**0.14**	**0.13**

Table 8. **Luminance masking** – LPIPS ↓ metric reported for reconstruction with/without local/global luminance masking.

Method	Room 1	Flat 1	Flat 2	Room 2	Room 3	Lab	Lounge	Persons	Shadow	**Mean**
DINOv3 [a]	32.59	**29.21**	30.50	31.61	29.87	29.43	24.35	30.86	**27.99**	29.60
FullCircle (Ours)	**32.64**	29.11	**30.55**	**31.63**	**29.91**	**29.54**	**24.40**	**30.99**	27.79	**29.62**

Table 9. **DINOv3** – Reconstruction quality reported in PSNR ↑ metric for DINOv3 masking alternative to SAMv2.

Method	COLMAP	Room 1	Flat 1	Flat 2	Room 2	Room 3	Lab	Lounge	Persons	Shadow	**Mean**
3DGRT [53]	Unmasked	27.56	25.75	24.99	27.88	27.48	26.90	23.43	failed	25.51	-
FullCircle (Ours)	Unmasked	32.20	28.31	30.20	31.30	28.80	28.68	24.38	failed	27.77	-
3DGRT [53]	Masked	27.37	26.81	25.09	27.92	28.41	26.66	23.41	27.56	25.71	26.55
FullCircle (Ours)	Masked	**32.64**	**29.11**	**30.55**	**31.63**	**29.91**	**29.54**	**24.40**	**30.99**	**27.79**	**29.62**

Table 10. **Unmaksed calibration** – Reconstruction performance reported in PSNR ↑ metric using unmasked calibration.