MVGSR: Multi-View Consistency Gaussian Splatting for Robust Surface Reconstruction

Chenfeng Hou[1] Qi Xun Yeo[2] Mengqi Guo[2] Yongxin Su[1] Yanyan Li[2†] Gim Hee Lee[2]
[1] Beihang University
[2] National University of Singapore
[†] Corresponding author

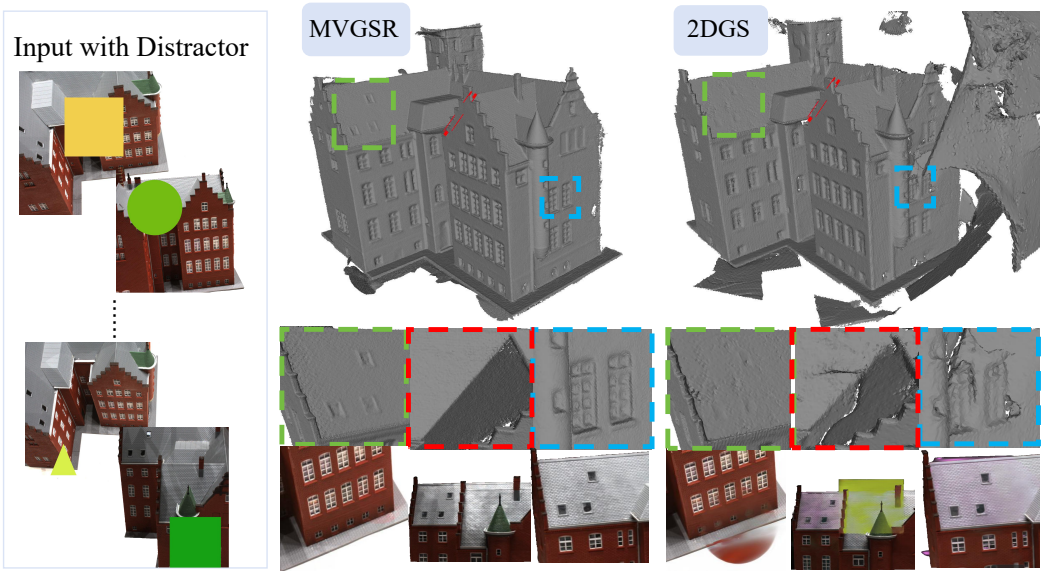


Figure 1. Surface reconstruction in scenes with distractors. The RGB inputs polluted by different types of objects are fed to 2DGS and MVGSR methods. Compared to 2DGS [7], the proposed method, MVGSR, shows robust mesh reconstruction and high-fidelity appearance rendering performance.

Abstract

*3D Gaussian Splatting (3DGS) has gained significant attention for its high-quality rendering capabilities, ultra-fast training, and inference speeds. However, when we apply 3DGS to surface reconstruction tasks, especially in environments with dynamic objects and distractors, the method suffers from floating artifacts and color errors due to inconsistency from different viewpoints. To address this challenge, we propose Multi-View Consistency Gaussian Splatting for the domain of Robust Surface Reconstruction (**MVGSR**), which takes advantage of lightweight Gaussian models and a heuristics-guided distractor masking strategy for robust surface reconstruction in non-static environments. Compared to existing methods that rely on MLPs for distractor segmentation strategies, our approach separates distractors from static scene elements by comparing multi-view feature consistency, allowing us to obtain precise distractor masks early in training. Furthermore, we introduce a pruning measure based on multi-view contributions to reset transmittance, effectively reducing floating artifacts. Additionally, we apply a multi-view consistency loss to achieve high-quality performance in surface reconstruction tasks. Experimental results demonstrate that MVGSR achieves competitive geometric accuracy and rendering fidelity compared to the state-of-the-art surface reconstruction algorithms. More information is available on our project page(https://mvgsr.github.io).*

1. Introduction

In recent years, neural network-based implicit representations [5, 18, 23] of 3D scenes have gained significant popularity due to their impressive modeling capabilities and generalization power. More recently, 3D Gaussian-based representations [7, 13, 15] have emerged as a promising alternative, offering a new approach to addressing this challenge. Compared to neural implicit representations, 3D Gaussian-based scene representations exhibit faster processing and higher rendering quality. Although 3D Gaussian-based scene representations achieve impressive visual results, they have limitations [4, 15] in accurately describing a scene's geometry. This deficiency, particularly in the context of robust surface reconstruction, significantly impacts the overall quality of 3D reconstruction. In casually captured scenes [35, 36], moving pedestrians, vehicles, and other objects in the images are unavoidable distractors. These distractors can occlude the central objects of interest. As shown in Fig. 1, when there are viewpoint-dependent distractors, 3DGS-based surface reconstruction algorithms may model them as artifacts clustered in front of the camera lens or as viewpoint-dependent color representations attached to the object's surface. To address the issue of distractors during reconstruction, we propose a surface reconstruction algorithm that avoids these distractions, preventing artifacts while achieving competitive geometric accuracy and enhanced rendering capabilities.

Different to previous work [24, 25] using photometric residual decomposition to obtain distractor masks, our core insight is that distractors appearing in only a few images lack semantic consistency across different views, resulting in noticeable differences in features extracted by pre-trained base models. We leverage this observation to separate distractors from static scenes early in training. Compared to iterative masks learned by MLP during training, masks obtained through multi-view comparison have two advantages: first, they distinguish distractors from high-frequency details, allowing continuous optimization of high-quality surface reconstruction. Second, strict mask boundaries prevent gradient leakage during Gaussian splitting, avoiding artifacts and viewpoint-dependent color errors in distractor regions.

To address the floating artifacts and ghosting caused by uncertain geometric relationships, we design a pruning measure based on multi-view contributions. This allows for transmittance resetting of objects occluding the camera's view to mitigate the impact of occlusion on gradient flow. This strategy can remove floating artifacts while compressing the number of point clouds, with only a minimal decrease in rendering quality. For high-precision optimization of surface reconstruction, we use a multi-view consistency loss function to enhance the reconstruction capability of Gaussian splats. This is achieved through structural and color consistency constraints on corresponding points from different views. In scenes with distractors, our method achieves optimal rendering quality and competitive reconstruction results. Our contributions can be summarized as follows:

- We propose a method using multi-view consistency to distinguish distractors from static objects before surface reconstruction, achieving strict distractor masking and significantly reducing floating artifacts and color errors.
- We introduce a new Gaussian pruning technique based on multi-view contributions to remove floating artifacts with minimal reduction in rendering capability.
- In scenes with distractors, we achieve high-fidelity rendering quality and high-precision reconstruction results. The code and generated dataset will be released to the community.

2. Related Work

2.1. Traditional 3D Reconstruction

Traditional Surface Reconstruction methods [26, 27, 29, 34] can be roughly grouped into different classes based on their intermediate representations , such as point clouds [26], volumes [34], and depth maps[9]. These methods typically decompose the entire multi-view stereo (MVS) problem into several stages. First, a dense point cloud is extracted from the multi-view images via patch-based matching [19]. Then, the surface structure is constructed through either feature triangulation [16] or implicit surface fitting [8]. These methods are often affected by mismatching or noise introduced during the reconstruction pipeline.

2.2. Neural Surface Reconstruction

NeRF-based methods [1, 18] take 5D rays as input to predict density and color sampled in implicit space, leading to more realistic rendering results. Despite NeRF's impressive performance in 3D reconstruction, its implicit function representation through volume rendering results in poor geometric accuracy and susceptibility to noise. To address these issues, methods such as NeuS [30], BakedSDF [31], and UNISURF [20] represent surfaces using signed distance functions (SDF) to achieve more accurate scene geometry. Meanwhile, Nerf2Mesh [28] introduces an iterative mesh refinement algorithm that adaptively adjusts vertex positions and volume density based on reprojection rendering error. However, while the NeRF-based framework exhibits powerful surface reconstruction capabilities, the stacked MLP layers impose limitations on inference time and representation power.

2.3. GS-based Surface Reconstruction

Different from neural surface reconstruction methods, GS-based approaches optimize point-based radiance fields, including 3D Gaussians, 2D Gaussians, and Gaussian surfels. SuGaR [6] extracts meshes from the positions of 3D Gaussians and introduces a regularization term to encourage

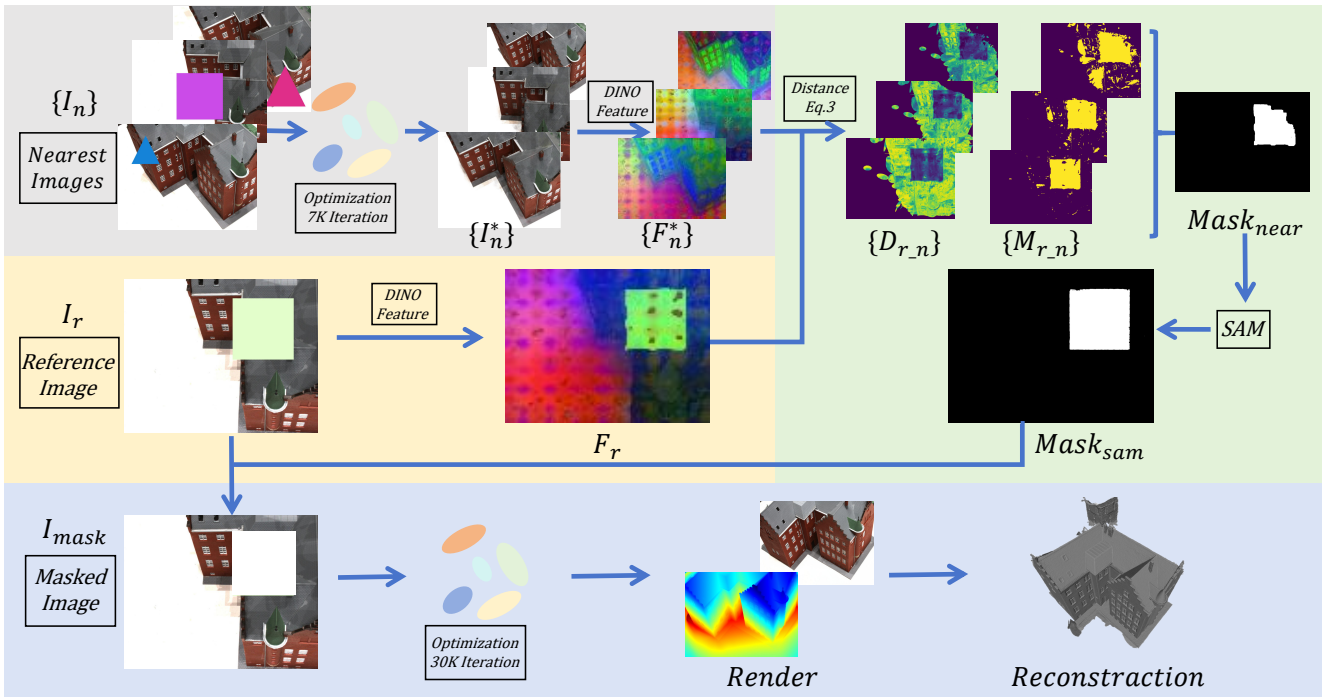


Figure 2. The detailed architecture of our MVGSR Framework. Images with distractors are fed to the system that makes use of multi-view consistency Gaussian Splatting algorithm to achieve robust surface reconstruction for non-static environments.

Gaussians to align with the scene surface based on sampled 3D point clouds. While this alignment improves geometric reconstruction accuracy, the irregular shapes of 3D Gaussians pose challenges in modeling smooth geometric surfaces. 2DGS [7] achieves view-consistent geometry by collapsing 3D volumes into a set of 2D Gaussian. GOF [33] constructs Gaussian opacity fields and extracts 3D models directly from them. However, these 3DGS-based methods typically produce biased depth estimates and struggle with maintaining multi-view geometric consistency. To address the inconsistent issue in scene reconstruction, PGSR [2] flattens the Gaussian function into a planar shape based on the assumption that scenes compose of several smaller planar regions. Leveraging these 3D relationships, PGSR introduces a new depth computation strategy to accurately extract geometric parameters from Gaussian surfels. Instead of assuming planar surfaces, our method proposes a more general architecture for object reconstruction, which incrementally transforms inconsistent ellipsoids to Gaussian surfels during the densification process.

2.4. Distractor Removal

For reconstruction in non-static scenes, distractor removal is one of the most important task. There are two main strategies in dealing with this problem namely segmentation-based methods and heuristics-based methods. For the former, deep semantic or segmentation models are used to detect object with potential dynamic characteristics or to recognize static scenarios. Based on pre-trained models, DynaMoN [12] further utilized semantic maps to reconstruct dynamic scenes via Motion-Aware Fast and Robust Camera Localization for in dynamic NeRF. However, for the latter, approaches [3, 17, 24] make use of heuristics generated from multi-view geometry algorithms to separate dynamic objects from static scenes. To be specific, NeRF-W [17] assumes that most of transient objects are generally small during the NeRF training process, while RobustNeRF [24] tries to define transient objects from static background based on the photometric residuals during the optimization module. Compared to RobustNeRF, NeRF-HUGS [3] leverages on masks estimated based pre-trained model with residual mask obtained from RobustNeRF to achieve more accurate distractor masking performance.

3. Preliminary of Gaussian Splatting

3DGS [13] models 3D scenes by employing a set of 3D Gaussians $\{\mathcal{G}_i\}$. A Gaussian function defines each of these Gaussians at point $\boldsymbol{p}_i \in \mathcal{P}$:

$$\mathcal{G}_i(\boldsymbol{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma_i}) = e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)^\top \boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_i)},$$

Each point $\boldsymbol{p}_i \in \mathcal{P}$ is centered at $\boldsymbol{\mu}_i \in \mathbb{R}^3$ with a corresponding 3D covariance matrix $\boldsymbol{\Sigma}_i \in \mathbb{R}^{3\times3}$. This covariance matrix $\boldsymbol{\Sigma}_i$ is factorized into a scaling matrix $\boldsymbol{S}_i \in \mathbb{R}^{3\times3}$ and

a rotation matrix $\boldsymbol{R}_i \in \mathbb{S}O(3)$:

$$\boldsymbol{\Sigma}_i = \boldsymbol{R}_i \boldsymbol{S}_i \boldsymbol{S}_i^\top \boldsymbol{R}_i^\top.$$

3DGS facilitates quick α-blending for rendering views. The transformation matrix W and intrinsic matrix $\boldsymbol{K}$ allow $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$ to be converted into camera coordinates related to W, and subsequently projected into 2D coordinates using the following functions:

$$\boldsymbol{\mu}_i^{'} = \boldsymbol{K}\boldsymbol{W}[\boldsymbol{\mu}_i, 1]^\top, \quad \boldsymbol{\Sigma}_i^{'} = \boldsymbol{J}\boldsymbol{W}\boldsymbol{\Sigma}_i\boldsymbol{W}^\top\boldsymbol{J}^\top,$$

where $\boldsymbol{J}$ represents the Jacobian of the affine approximation for the projective transformation. The color $\boldsymbol{C} \in \mathbb{R}^3$ of a pixel $\boldsymbol{u}$ can be rendered using α-blending.:

$$\boldsymbol{C} = \sum_{i \in N} T_i \alpha_i \boldsymbol{c}_i, \quad T_i = \prod_{j=1}^{i-1}(1 - \alpha_i),$$

Here, α_i is determined by evaluating $\mathcal{G}_i(\boldsymbol{u}|\boldsymbol{\mu}_i^{'}, \boldsymbol{\Sigma}_i^{'})$ and multiplying it with a learnable opacity associated with $\mathcal{G}_i$. The view-dependent color $\boldsymbol{c}_i$ is expressed using spherical harmonics (SH) from the Gaussian $\mathcal{G}_i$. T_i represents the cumulative opacity, and N denotes the number of Gaussians the ray intersects.

The center $\boldsymbol{\mu}_i$ of a Gaussian $\mathcal{G}_i$ can be transformed into the camera coordinate system as:

$$\left[x_i, y_i, z_i, 1\right]^\top = [\boldsymbol{W}|\boldsymbol{t}][\boldsymbol{\mu}_i, 1]^\top,$$

And previous methods [4, 11] in depth rendering under the current viewpoint can be denoted as:

$$\boldsymbol{D} = \sum_{i \in N} T_i \alpha_i z_i.$$

For 3D Gaussians, the direction of the minimum scale factor corresponds to the normal n_i of the Gaussian. The normal map under the current viewpoint is achieved through α-blending:

$$\boldsymbol{N} = \sum_{i \in N} T_i \alpha_i \boldsymbol{R}_c^T \boldsymbol{n}_i$$

where R_c is the rotation from the camera to the global world.

4. Methodology

Given a set of images of a scene containing outliers, the goal of our method is to achieve high-precision surface reconstruction while maintaining robust view rendering performance. In existing work on outlier masking [24, 25], mask estimation typically relies on photometric errors, but these methods cannot accurately distinguish between outliers and reconstruction details, leading to potential inaccuracies. To address this issue, we propose a novel method that leverages feature similarity across multiple views to differentiate outliers. Specifically, we utilize features extracted from a self-supervised 2D foundation model [21] for feature extraction. With fewer training iterations, we can generate a rough scene representation to compute the mapping between the original view and adjacent views. By comparing feature similarities, we can derive an initial mask that helps identify outliers. To refine the mask boundaries and enhance its accuracy, we sample points from the mask and apply a basic segmentation model to obtain precise mask results. Retraining with this refined mask helps mitigate gradient leakage during the optimization process. Additionally, to further reduce artifacts, we apply pruning based on multi-view contributions, ensuring the removal of unreliable information from the reconstruction. The process of obtaining the initial mask will be detailed in Sec. 4.1, while Sec. 4.2 will discuss multi-view pruning, and Sec. 4.3 will describe the loss function used in the optimization.

Figure 3. Example of distractors in various scenarios of DTU-Robust dataset.

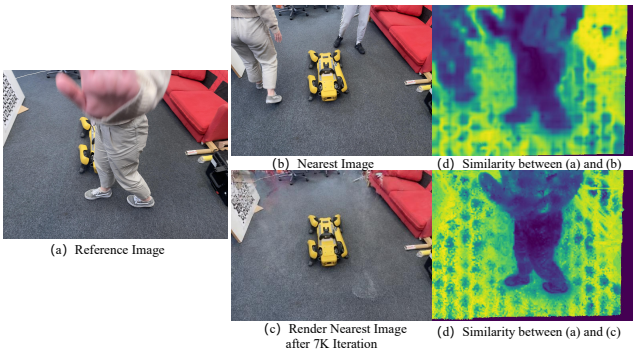


Figure 4. Reference image (a), nearest image after 7000 iteration (b), rendered image (c), feature similarity map between reference and nearest viewpoints (d), and feature similarity map between reference and rendered images (e). The rendered image can remove some distracting objects, preventing interference with the features of the reference image.

4.1. Distractor Detection

Similar to the traditional 3D Gaussian Splatting methods [13, 15], scenes containing distractors are processed in the same initialization stage. This stage leverages sparse point clouds and camera parameters generated from structure-from-motion (SfM) techniques. Using this initial setup, a rough scene reconstruction is performed, resulting in the computation of the initial depth map D, surface nor-

mals N, and an initial rendered image I^*. This initialization provides a foundational representation of the scene, which is then refined in subsequent stages of the method to improve accuracy and handle outliers effectively.

Then, three steps are proposed to achieve an effective distractor detection strategy. First, the Gaussian surfels observed from the viewpoint I_r are associated with the corresponding surfels detected from neighboring viewpoints I_n, based on the relative camera pose $\mathbf{R}, \mathbf{t}$ between the two viewpoints. For a pixel $\mathbf{p}_r$ on image I_r, the corresponding pixel $\mathbf{p}_n$ in a neighboring viewpoint I_n can be calculated using the homographic relationship in Eqn. 1.

$$\mathbf{H}_{nr} = \mathbf{K}_n \mathbf{R}_{nr} \left(\mathbf{I} + \frac{1}{d_r} \cdot \mathbf{t}_{rn} \mathbf{n}_r^\top \right) \mathbf{K}_r^{-1} \tag{1}$$

$$\mathbf{p}_n = \mathbf{H}_{nr} \mathbf{p}_r \tag{2}$$

This allows us to establish correspondences between pixels across different viewpoints, which is essential for identifying and handling potential distractors in the scene.

For a given reference image I_r and an initially rendered neighboring viewpoint I^*, we use DINOv2[21] to extract image features following the Nerf-on-the-Go[22] method, resulting in feature maps F_r and F_n^*. To remove the transient distracting objects that may also be present in the neighboring viewpoints which misleads the feature similarity calculation, we use the initially rendered image I^* instead of the original neighboring viewpoint image I_n. This issue is illustrated in Fig. 4, where distracting objects in the original neighboring viewpoint image I_n interferes with accurate feature matching and distractor detection.

Based on the two associated points, we estimate the feature distance between them shown in Eqn. 3. The number of channels in the feature map is 384. To compute the feature vectors for two points on the corresponding feature maps, we first apply bilinear interpolation to obtain the features $F_r(\mathbf{p}_r), F_n^*(\mathbf{p}_n)$, respectively. This is necessary because the feature map is downsampled by a factor of 14 from the original image, and bilinear interpolation allows us to extract the feature values at specific pixel locations. Pixels with a distance lower than a certain threshold δ_{near} are used as a mask for the adjacent view. Then, the computation process of the distance is denoted as:

$$distance(\mathbf{p}_r, \mathbf{p}_n) = abs\left(\frac{F_r(\mathbf{p}_r) \cdot F_n^*(\mathbf{p}_n)}{|F_r(\mathbf{p}_r)||F_n^*(p_n)|}\right) \tag{3}$$

where M_n is defined based on $distance_{rn}(\mathbf{p}_r) < \delta_{near}$.

Second, each mask M_n obtained from the first step contains many incorrect segmentations due to rough geometric estimates and noise. Using the multi-view mapping relationships, we calculate the visibility of the 3D spatial points corresponding to the pixels classified as clutter in the mask and the current view. If a pixel is identified as clutter by at least two visible adjacent views, it is retained as the final segmentation result M_{nest}.

Third, the quality of segmentation in some boundary regions of the coarse mask is not very accurate. Therefore, we continue to refine these regions based on the segment anything model (SAM) [14] which is a prompt-based segmentation model. We perform uniform sampling on the M_{near} to obtain positive sample points on the clutter and negative sample points in the background region. These are computed with the original image using the outputs of SAM, resulting in a clutter mask M_{sam} with precise edges.

$$M_{sam} = SAM(I_r, \mathcal{P}, \mathcal{N}) \tag{4}$$

4.2. Multi-view Pruning

In the process of reconstructing clutter, the optimization often causes Gaussians to cluster near the camera. These floaters are typically incorrect models of viewpoint-dependent phenomena for clutter that only exists for a few frames. Once these floaters appear during optimization, they force the line of sight to reach the transmittance limit prematurely, preventing gradient propagation. 3DGS is to reset the opacity of all Gaussians every few iterations, using it as a control mechanism. This allows gradients to flow again and prunes Gaussians that cannot regain higher opacity.

However, in regions with clutter, resetting opacity is not fully effective. First, due to the presence of masks, this area tends to split into more Gaussians after opacity reset. For masks that cannot be perfectly segmented, this will lead to further aggravation of floaters. Therefore, we propose multi-view contribution-based pruning (MV-Prune). We define multi-view contribution as:

$$\mathbf{C}_{MV}(p) = \sum_{\mathbb{V}_k \in \mathbb{V}} \left(\sum_{p \in \mathbb{V}_k} \alpha_{i(p)} \prod_{j=1}^{i(p)-1} (1 - \alpha_j) > \delta_{opacity} \right) \tag{5}$$

where V_k is the training viewpoint, and p is the Gaussian for which contribution needs to be calculated. The contribution of Gaussian p for viewpoint V_k is measured by the cumulative transmittance of the pixel associated with Gaussian p. When the cumulative transmittance along the ray exceeds the threshold δ_{opacity}, the contribution of Gaussian p in this viewpoint is incremented by one.

When the opacity of a Gaussian in a certain view exceeds a threshold, the contribution is raised by one. Once $C_{\text{MV}}(p)$ exceeds the threshold δ_{prune}, its transmittance is reset to a lower value to allow gradients to flow again. Experimental results show that MV-Prune effectively handles floaters, compressing by 60% with comparable rendering quality.

4.3. Multi-view Consistency

To enhance geometric consistency, we use photometric consistency constraints based on neighboring patches, similar to

MVS algorithms. We use a homography matrix to compute the 11×11 pixel patch P_r around pixel $\mathbf{p}_r$, mapping it to the corresponding region P_n in the neighboring view. To measure consistency, we use the normalized cross correlation (NCC) [32] coefficient as a loss metric:

$$\boldsymbol{L}_{mv} = \frac{1}{V} \sum_{\boldsymbol{p}_r \in V} \left(1 - NCC\left(\boldsymbol{I}_r\left(\boldsymbol{p}_r\right), \boldsymbol{I}_n\left(\boldsymbol{p}_n\right)\right)\right). \quad (6)$$

To focus on these areas with reconstruction errors, we calculate per-pixel geometric reconstruction accuracy weights. This is done by reprojecting the corresponding pixel $\mathbf{p}_n$ from the neighboring view back using the homography matrix to obtain $\mathbf{p}_n^{'}$, then calculating the reprojection error and weight. This weight can also handle out-of-bounds and potential occlusion issues in different views.

$$\boldsymbol{E}_{repro} = \frac{1}{V} \sum_{\boldsymbol{p}_r \in V} \left\| \boldsymbol{p}_r - \boldsymbol{H}_{rn} \boldsymbol{p}_n \right\| \quad (7)$$

$$\boldsymbol{w}_{repro} = \frac{1}{1 + \boldsymbol{E}_{repro}} \quad (8)$$

Unlike MVS algorithms, we do not directly optimize the reprojection error, as the lack of neighborhood information can impair the gradient propagation of Gaussian errors. Using color error alone is sufficient for depth optimization.

Another loss function is the regularization term L_s that minimizes the shortest axis, modeling the Gaussian as a thin surface. The image reconstruction loss is L_{rgb}. In summary, the loss function is:

$$\boldsymbol{L} = L_{rgb} + \lambda_1 L_s + \lambda_2 w_{repro} L_{mv} \quad (9)$$

For the surface loss ,We set λ_1 = 100. For the geometric loss, we set λ_2 = 0.2.

5. Experiments

In this section, we present both qualitative and quantitative comparisons of our method with state-of-the-art Gaussian Splatting reconstruction approaches [2, 7] on public datasets.

5.1. Implementation

All experiments were conducted on a single NVIDIA A800 GPU. The maximum number of nearest images in multi-view scenarios was set to 8, with a maximum angular difference of 60 degrees between adjacent view cameras and a maximum distance of 1.5. The novel view rendering task was performed using the training results from $7,000$ iterations, and the corresponding relationships between adjacent views were calculated using Eqn. 3. When computing the multi-view masks in Sec. 4.1, the cosine similarity threshold δ_{near} is set to 0.5. For SAM prompt points, 20 segmentation prompt points and 1 exclusion prompt point are used, with the segmentation results being considered valid if the top 10 votes exceeded 2, serving as the refined mask segmentation results. The masks are then used for re-training, resulting in high-quality surface reconstruction after $30,000$ iterations. In the multi-view pruning phase, the transmittance contribution was set to 0.5, and only views with contributions exceeding the threshold of 8 were retained.

5.2. Datasets and Metrics

Datasets. To evaluate the reconstruction and rendering performance, we use the DTU dataset [10] for quantitative assessment of reconstruction quality. The DTU dataset consists of 15 object-centric scenes, each accompanied by ground truth 3D models. Since the DTU scenes contain only static objects and lack dynamic elements, we extend the dataset by creating DTU-Robust. This version incorporates distractors into the images to simulate fast-moving objects, as shown in Fig. 3. The updated DTU-Robust dataset is publicly available, and details on the addition of distractors are provided in this section.

In the DTU-Robust dataset, random shapes such as*squares*, *circles*, or *triangles* of varying sizes are added at random positions within the scenes to simulate distractors. To assess the impact of distractor density, we set the proportion of occluded images to a rate r, with values of 0.3, 0.5, 0.8, and 1.0, representing the fraction of the total training set images that contain distractors. Additionally, we evaluate our method on the NeRF-on-the-go dataset [22], which includes six casually captured scenes. The occlusion in these scenes primarily consist of moving humans in outdoor environments. Each scene contains both images with and without distractors, and the evaluation metrics are calculated using the data without distractors.

Metrics for rendering and reconstruction evaluations. In the experimental section, we employ standard photometric rendering quality metrics to assess the performance of novel view rendering, including Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), and Learned Perceptual Image Patch Similarity (LPIPS). For LPIPS, we use normalized VGG features, as is commonly done following the methods in this area.

For surface quality evaluation, we use Chamfer distance to assess the similarity between point clouds. We compute a bidirectional loss denoted as CD during evaluation, considering both the destination and source point sets.

5.3. Reconstruction

We compare the reconstruction performance of our proposed method to PGSR [2] and 2DGS [7] on the DTU-Robust dataset, across different scans and varying occlusion rates. As shown in Tab. 5, our method achieves superior and comparable performance under various evaluation protocols (CD,

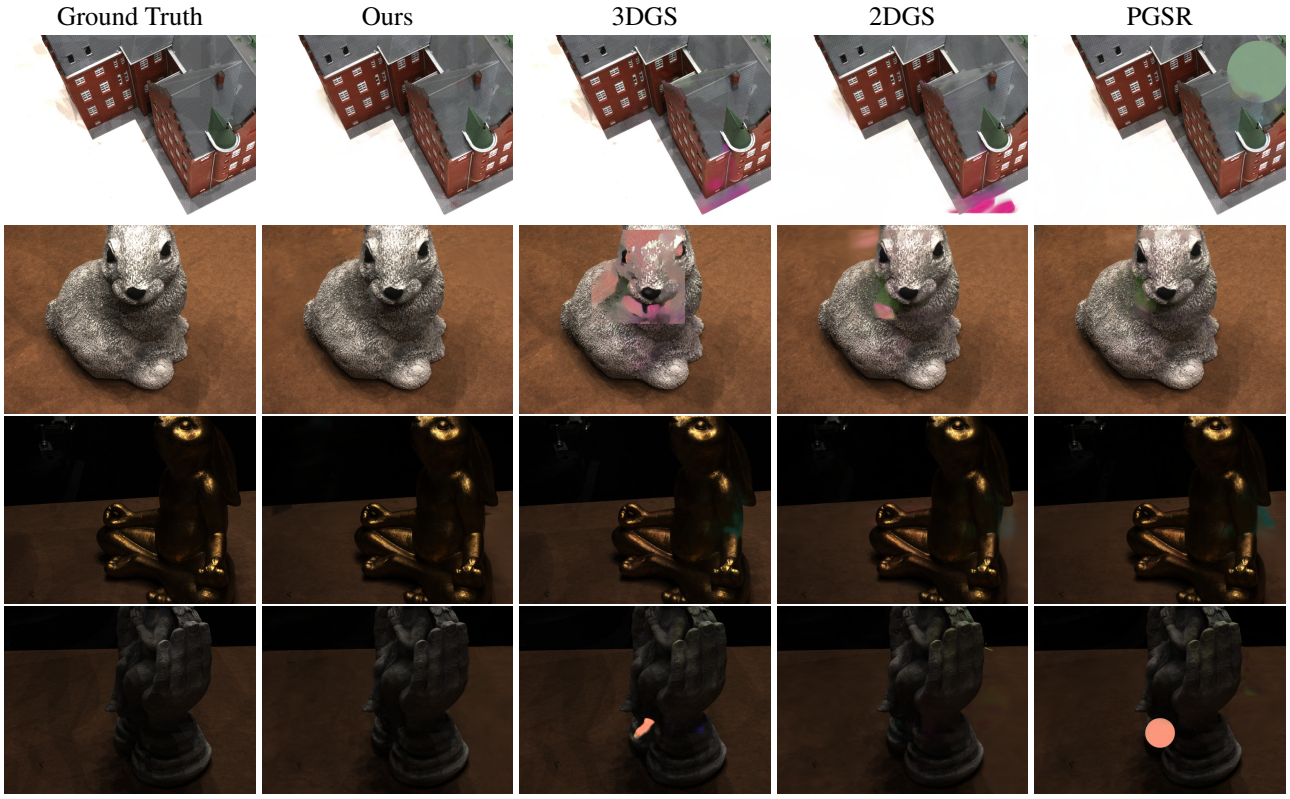


Figure 5. Comparison of view rendering on the DTU-Robust dataset. We select scan scene 24 (building), 55 (rabbit), and 118 (statue) to compare between the different methods. More rendering results are provided in Sec. A.3.

scan		24				55				106				118				Avg.
rate		0.3	0.5	0.8	1.0	0.3	0.5	0.8	1.0	0.3	0.5	0.8	1.0	0.3	0.5	0.8	1.0	
CD	PGSR [2]	0.38	0.36	0.83	0.53	**0.35**	**0.36**	**0.34**	**0.35**	**0.47**	0.46	**0.46**	**0.48**	0.37	**0.36**	**0.43**	0.38	0.43
	2DGS [7]	0.50	0.52	0.48	0.57	0.38	0.39	0.45	0.56	0.70	0.69	0.70	0.73	0.68	0.71	0.69	0.71	0.69
	MVGSR	**0.34**	**0.34**	**0.37**	**0.38**	0.36	0.37	0.37	0.37	**0.47**	**0.45**	0.49	0.49	**0.37**	0.38	0.44	**0.37**	**0.39**
PSNR↑	3DGS [13]	26.21	25.46	25.26	24.74	30.33	29.35	28.07	25.98	33.97	34.20	33.24	31.18	35.12	34.88	32.51	32.59	30.19
	PGSR [2]	31.85	31.48	30.68	**31.31**	32.94	32.11	31.72	30.44	35.40	35.43	35.05	33.86	37.23	37.03	35.71	35.67	33.61
	2DGS [7]	**33.47**	28.94	27.87	30.96	**33.22**	32.99	31.47	30.36	35.61	35.78	35.27	34.46	**37.29**	37.12	35.56	35.18	33.66
	MVGSR	32.21	**31.91**	**31.33**	31.29	33.17	**33.57**	**33.27**	**32.93**	**35.85**	**35.95**	**36.16**	**35.98**	37.26	**37.47**	**37.04**	**37.13**	**34.53**

Table 1. Quantitative results of reconstruction performance on DTU-Robust dataset. We select scan scene 24, 55, 106 and 118 to compare between the different methods. More quantitative results are illustrated in Tab. 5.

d2s, and s2d metrics). Notably, our method outperforms both PGSR and 2DGS in *scan 24*. Specifically, at an occlusion scale of 0.8, our method achieves a CD score that is more than twice as good as PGSR, improving from 0.83 to 0.37. Compared to 2DGS, our method exhibits greater robustness across all scenes, with fewer performance fluctuations. On average, our method outperforms 2DGS and PGSR by 43.48% and 9.30%, with a CD score of 0.39. These results demonstrate that our method maintains robust reconstruction quality across varying occlusion rates and different scenes.

Next, we present a qualitative comparison with the baselines on the DTU-Robust dataset. As shown in Fig. 6, our method consistently produces high-quality reconstructions compared to 2DGS and PGSR. Specifically, our results exhibit smoother, more continuous surfaces with fewer holes and distractors. In contrast, 2DGS outputs lack detail, while PGSR results contain numerous distractors and fragmentary artifacts in most scenes. Our approach generates more complete and smoother surface reconstructions, with fewer artifacts. Furthermore, both 2DGS and PGSR suffer from the appearance of holes and artifacts in the neck of the rabbit (*scan 55*), while our method accurately reconstructs the object in fine detail. These high-quality reconstruction results highlight the robustness of our method to distractors.

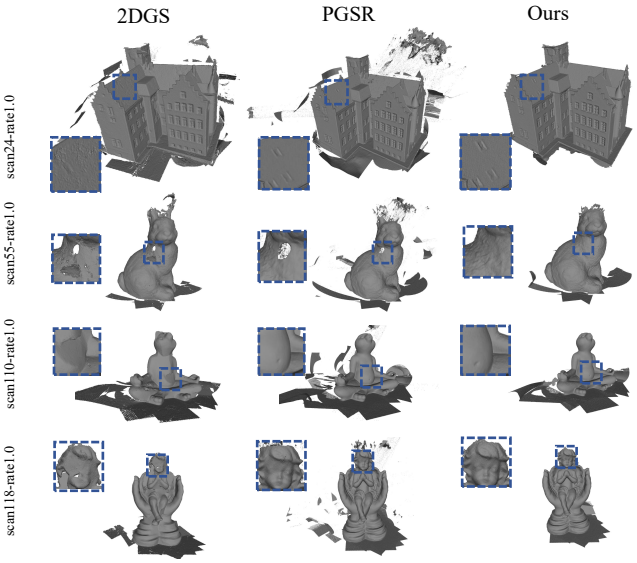


Figure 6. Comparison of reconstruction on DTU-Robust dataset.

(a) no mask (b) MV-mask (c) MV-mask+SAM

Figure 7. Comparison of different mask strategies.

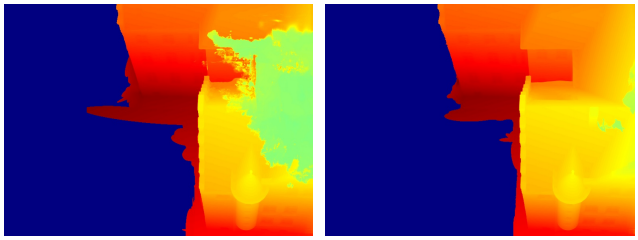
5.4. View rendering

We show the rendering performance on the NeRF on-the-go dataset [22] compared against PGSR [2] and 2DGS [7]. As shown in Tab. 4, our method shows consistently higher PSNR and maintains better or competitive SSIM and LPIPS on different scenes. Especially in *fountain*, ours surpasses the 2DGS by 1.08 and PGSR by 1.13 in terms of PSNR. On average, ours achieves 23.55 dB in terms of PSNR and demonstrates better performance than baselines. The higher results across scenes indicate that our method has robust and strong novel view synthesis quality.

We also evaluated our method qualitatively in comparison to other competing methods as shown in Fig. 5. Our method clearly removes most distractors and distortion compared to the other methods. Our method clearly resembles the ground truth the closest without distractors and this validates the effectiveness of our pruning strategy in alias removing visualized in Fig. 8.

5.5. Multi-view Pruning

Pruning Analysis Quantitatively, we compare the effectiveness of our MV-Prune in Tab. 2. With MV-Prune, the storage requirement of the model is decreased by 30.19% on average while maintaining 99.4% of the original performance according to PSNR↑ and a 0.01 decrease in CD↓.



(a) Before MV-prune (b) After MV-prune

Figure 8. Proposed pruning strategy MV-prune improves artifact removal, comparing results before and after MV-prune application.

	MVPSR			MVGSR+MV-Prune		
	PSNR↑	CD↓	SIZE(MB)	PSNR↑	CD↓	SIZE(MB)
0.0	35.71	0.47	27.28	35.49	0.47	23.57
0.3	35.85	0.47	55.78	35.86	0.46	47.65
0.5	35.95	0.46	56.74	35.70	0.46	47.68
0.8	36.16	0.49	60.78	35.94	0.47	48.24
1.0	35.98	0.49	105.63	35.66	0.49	46.60
Avg.	35.93	0.48	61.24	35.73	0.47	42.75

Table 2. The average metrics on DTU-Robust scan106 before and after MV-Prune, including PSNR, CD, and Gaussian file size.

Qualitatively, when we compare the images visualized in Fig. 8, we observe that the image on the right is cleaner with fewer distractors. Firstly, the green artifact present on the right of Fig. 8a is largely removed in Fig. 8b with only a few small specks remaining. Moreover, the dark red jagged edges in the middle of Fig. 8a are smoothed to become less pronounced in Fig. 8b. Lastly, the details in the scene such as the structure of the object in yellow to orange are preserved in both pictures. These three observations show the efficacy of pruning with fewer distractors and better outlier removal whilst maintaining the high-frequency details of the scene.

Mask Analysis We show the analysis of the effectiveness of our proposed masking strategies in Fig. 7. The baseline 'no mask' renders the scene with a noticeable artifact - the distractor as shown in Fig. 7a, and 'MV-mask' improves the geometry consistency and renders the details behind the distractor with small artifacts, as shown in Fig. 7b. The 'MV-mask+SAM' achieves further enhancement by combining the results from SAM, and provides clear and accurate rendering results, as shown in Fig 7c. The analysis proves that our integrated masking approach effectively mitigates visual artifacts caused by the distractor and improves the overall rendering quality.

6. Discussion and Conclusion

In this paper, we introduced Multi-View Consistency Gaussian Splatting for Robust Surface Reconstruction (MVGSR), a novel approach that addresses the common issues of float-

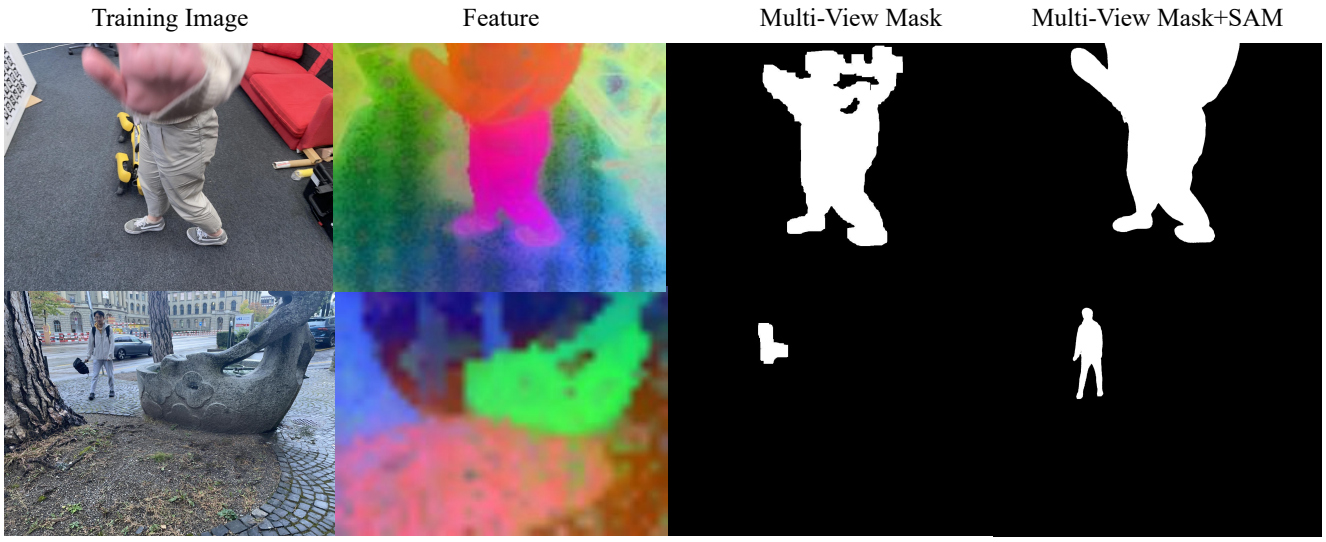


Figure 9. Disturbed images, corresponding semantic features, and multi-view mask results

Dataset	Metric	2DGS [7]	PGSR [2]	Ours
fountain	PSNR↑	20.74	20.69	21.82
	SSIM↑	0.583	0.657	0.659
	LPIPS↓	0.424	0.353	0.355
corner	PSNR↑	26.36	25.17	25.84
	SSIM↑	0.778	0.799	0.799
	LPIPS↓	0.369	0.325	0.338
spot	PSNR↑	22.57	22.72	22.99
	SSIM↑	0.576	0.593	0.586
	LPIPS↓	0.415	0.369	0.379
Avg.	PSNR↑	23.22	22.86	23.55
	SSIM↑	0.645	0.683	0.681
	LPIPS↓	0.402	0.349	0.357

Table 3. Comparison of novel view synthesis for Gaussian Splatting reconstruction methods on NeRF-on-the-go [22]. More results are provided in Supplementary.

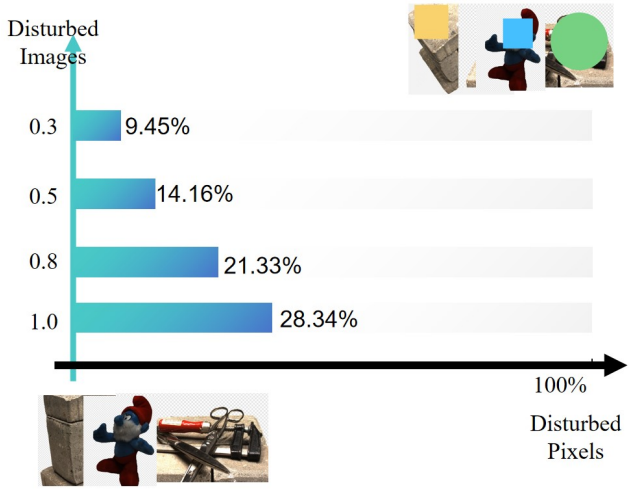


Figure 10. The average ratio of disturbed pixels to all object pixels under different disturbed image proportions on the DTU-Robust dataset.

ing artifacts and color errors in 3D Gaussian Splatting when applied to surface reconstruction. While 3DGS has gained popularity due to its high-quality rendering and fast training speeds, it is prone to distortions from distractors across different viewpoints, which can severely affect the accuracy and quality of surface reconstruction. Our method mitigates these issues by focusing on multi-view consistency to identify and separate distractors from the static elements in the scene.

A. More Results on DTU-Robust Dataset

A.1. Dataset Analysis

The DTU-Robust dataset introduces distractors with random positions, colors, and shapes to a given proportion (rate) of DTU training images. The relationship between the number of disturbed images and the number of disturbed pixels is shown in Fig. 10. Specifically, at a *rate 0.3*, the distractor regions occupy 9.45% of the image, while this increases to 28.34% at *rate 1.0*. Therefore, from the perspective of the occupied pixels in the distractor areas, the challenges increase from *rate 0.3* to *rate 1.0* for the same sequence. The

scan		**fountain**	**corner**	**spot**	**statue**	**train_station**	**tree**	**drone**	**mountain**	**patio_high**	**Avg.**
PSNR↑	PGSR [2]	20.69	25.17	22.72	20.49	19.70	19.03	16.47	20.89	20.27	20.60
	2DGS [7]	20.74	26.36	22.57	16.40	17.51	20.32	19.06	18.27	21.46	20.30
	MVGSR	21.82	25.84	22.99	20.48	20.14	18.76	17.40	20.20	20.67	20.92
SSIM↑	PGSR [2]	0.65	0.79	0.59	0.85	0.73	0.67	0.71	0.36	0.68	0.67
	2DGS [7]	0.58	0.77	0.57	0.57	0.59	0.68	0.67	0.48	0.65	0.62
	MVGSR	0.65	0.79	0.58	0.88	0.77	0.66	0.72	0.33	0.68	0.67
LPIPS↓	PGSR [2]	0.35	0.32	0.36	0.15	0.25	0.20	0.23	0.46	0.23	0.28
	2DGS [7]	0.42	0.36	0.41	0.55	0.53	0.27	0.35	0.52	0.32	0.41
	MVGSR	0.35	0.33	0.37	0.15	0.26	0.19	0.22	0.46	0.24	0.29

Table 4. Comparison of novel view synthesis for Gaussian Splatting reconstruction methods on NeRF-on-the-go [22].

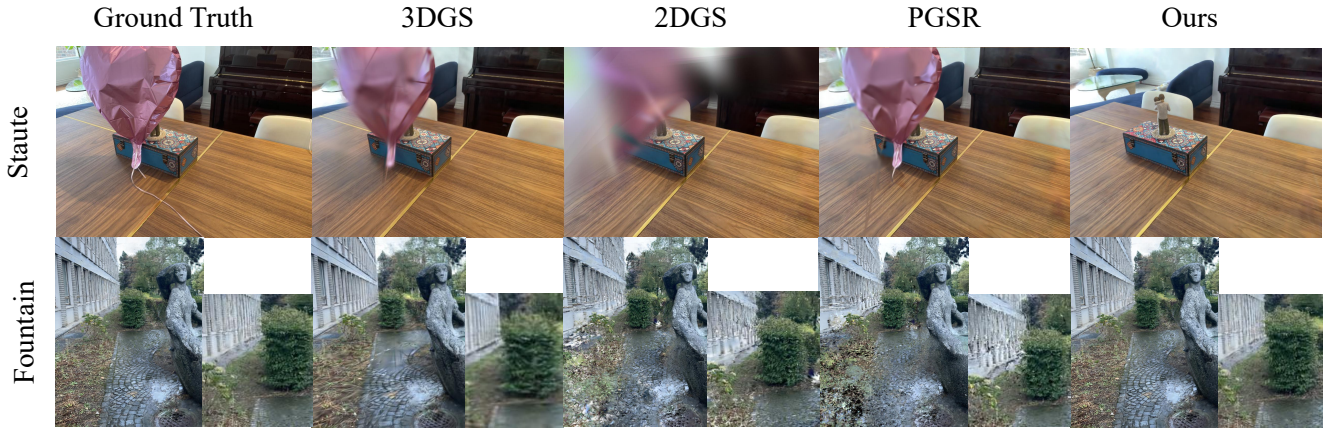


Figure 11. Visualization results of the t_balloon_statue on the dataset provided by RobustNeRF and the fountain on the NeRF-On-the-GO

DTU-Robust dataset effectively demonstrates the impact of occlusions, highlighting reconstruction errors and novel view rendering errors in the presence of these distractions.

A.2. Reconstruction

For surface quality evaluation, we use two other metrics, in addition to the bidirectional Chamfer distance CD used in the main paper, to assess the similarity between point clouds. We denote the Chamfer distance from the destination to the source as d2s, and from the source to the destination as s2d.

We compare the reconstruction performance of our proposed method to PGSR [2] and 2DGS [7] on the DTU-Robust dataset, across different scans and varying occlusion rates. As shown in Tab. 5, our method achieves superior and comparable performance under various evaluation protocols (CD, d2s, and s2d metrics) for the additional scans in the dataset. Notably, our method outperforms both PGSR and 2DGS significantly in *scan 24*, *scan 37*, and *scan 63*. Specifically for *scan 24*, at an occlusion scale of 0.8, our method achieves a CD score that is more than twice as good as PGSR, improving from 0.83 to 0.37. Compared to 2DGS and PGSR, our method exhibits greater robustness across all scenes, with fewer performance fluctuations. We achieve comparable performance to PGSR for all other settings.

On average across all settings, our method outperforms 2DGS and PGSR by 47.4% and 11.8%, with an average CD score of 0.53. These results demonstrate that our method maintains robust reconstruction quality across varying occlusion rates and different scenes.

A.3. Rendering

To evaluate the novel view rendering performance of our proposed method, we compare with PGSR [2], 2DGS [7] and, additionally, 3DGS [13] for the novel view rendering task on the same DTU-Robust dataset. As shown in Tab. 6, our method is largely able to outperform the above competing methods. When comparing PSNR, our method achieves top 2 performance on average under all settings for scenes that are evaluated on.

Similar to the task of reconstruction, our method exhibits greater robustness across all scenes, with fewer performance fluctuations compared to other competing methods. On average, we outperform 3DGS most significantly by close to 4.71 dB with a PSNR of 32.33 dB while outperforming other methods by a smaller margin. These results demonstrate that our method can improve the quality of rendered images across varying occlusion rates and different scenes.

scan		24				37				40				55			
rate		0.3	0.5	0.8	1.0	0.3	0.5	0.8	1.0	0.3	0.5	0.8	1.0	0.3	0.5	0.8	1.0
CD	PGSR [2]	0.39	0.37	0.83	0.53	0.66	0.96	1.22	1.16	0.44	0.51	0.46	0.62	0.36	0.36	0.35	0.36
	2DGS [7]	0.50	0.52	0.48	0.57	0.83	0.85	0.85	0.92	0.34	0.69	0.92	0.43	0.38	0.39	0.45	0.56
	MVGSR	0.35	0.34	0.37	0.38	0.59	0.61	0.63	0.60	0.32	0.32	0.32	0.40	0.36	0.38	0.37	0.38
d2s	PGSR [2]	0.38	0.36	1.23	0.70	0.78	1.39	1.91	1.79	0.51	0.66	0.54	0.87	0.33	0.32	0.29	0.31
	2DGS [7]	0.50	0.54	0.47	0.58	0.99	1.02	1.03	1.18	0.35	0.48	0.59	0.53	0.35	0.36	0.46	0.65
	MVGSR	0.34	0.33	0.39	0.41	0.61	0.66	0.69	0.63	0.31	0.31	0.31	0.46	0.29	0.32	0.31	0.32
s2d	PGSR [2]	0.40	0.37	0.44	0.37	0.54	0.54	0.54	0.53	0.37	0.37	0.38	0.38	0.38	0.40	0.41	0.40
	2DGS [7]	0.50	0.51	0.49	0.57	0.67	0.67	0.66	0.66	0.33	1.24	0.90	0.34	0.41	0.43	0.44	0.48
	MVGSR	0.35	0.36	0.36	0.36	0.57	0.57	0.57	0.57	0.33	0.33	0.33	0.35	0.43	0.44	0.43	0.44

scan		63				65				69				83			
rate		0.3	0.5	0.8	1.0	0.3	0.5	0.8	1.0	0.3	0.5	0.8	1.0	0.3	0.5	0.8	1.0
CD	PGSR [2]	0.75	1.17	1.17	1.46	0.60	0.64	0.61	0.59	0.48	0.49	0.49	0.51	0.82	0.85	1.02	0.96
	2DGS [7]	1.07	1.02	1.16	1.22	0.97	1.09	0.92	1.00	0.79	0.81	0.85	0.81	1.32	1.33	1.39	1.42
	MVGSR	0.74	0.86	0.76	0.84	0.59	0.61	0.61	0.59	0.50	0.48	0.50	0.51	0.91	0.84	1.04	0.87
d2s	PGSR [2]	1.09	1.93	1.91	2.52	0.58	0.58	0.55	0.57	0.50	0.50	0.51	0.51	0.55	0.84	0.90	1.07
	2DGS [7]	1.18	1.13	1.25	1.30	0.89	1.08	0.91	1.02	0.79	0.77	0.84	0.79	1.00	0.95	1.00	1.04
	MVGSR	1.08	1.33	1.10	1.28	0.55	0.55	0.53	0.56	0.52	0.50	0.53	0.51	0.57	0.82	0.95	1.00
s2d	PGSR [2]	0.41	0.42	0.43	0.41	0.62	0.70	0.67	0.61	0.47	0.47	0.48	0.51	1.09	0.86	1.14	0.84
	2DGS [7]	0.96	0.91	1.07	1.14	1.06	1.11	0.93	0.97	0.80	0.84	0.85	0.84	1.63	1.71	1.78	1.81
	MVGSR	0.40	0.40	0.42	0.40	0.63	0.67	0.70	0.62	0.48	0.47	0.48	0.50	1.26	0.86	1.14	0.74

scan		97				105				106				110			
rate		0.3	0.5	0.8	1.0	0.3	0.5	0.8	1.0	0.3	0.5	0.8	1.0	0.3	0.5	0.8	1.0
CD	PGSR [2]	0.63	0.63	0.62	0.63	0.58	0.59	0.61	0.73	0.47	0.47	0.47	0.49	0.51	0.46	0.45	0.44
	2DGS [7]	1.22	1.19	1.22	1.13	0.68	0.68	0.69	0.69	0.70	0.69	0.70	0.73	0.70	0.69	0.70	0.73
	MVGSR	0.69	0.63	0.68	0.67	0.58	0.59	0.64	0.69	0.47	0.46	0.49	0.49	0.52	0.47	0.54	0.47
d2s	PGSR [2]	0.63	0.64	0.61	0.65	0.49	0.50	0.54	0.79	0.36	0.37	0.36	0.39	0.63	0.55	0.52	0.51
	2DGS [7]	1.09	1.01	1.08	0.96	0.60	0.59	0.61	0.60	0.50	0.48	0.50	0.58	1.44	1.60	1.53	1.63
	MVGSR	0.70	0.64	0.71	0.71	0.49	0.50	0.58	0.71	0.38	0.36	0.39	0.39	0.64	0.54	0.69	0.56
s2d	PGSR [2]	0.62	0.62	0.62	0.61	0.68	0.68	0.69	0.68	0.58	0.57	0.57	0.58	0.38	0.38	0.38	0.38
	2DGS [7]	1.35	1.36	1.36	1.30	0.77	0.78	0.77	0.78	0.90	0.90	0.90	0.88	1.23	1.56	1.60	1.57
	MVGSR	0.69	0.62	0.65	0.63	0.67	0.68	0.69	0.67	0.57	0.56	0.59	0.59	0.39	0.39	0.39	0.39

scan		114				118				122				Avg.			
rate		0.3	0.5	0.8	1.0	0.3	0.5	0.8	1.0	0.3	0.5	0.8	1.0	0.3	0.5	0.8	1.0
CD	PGSR [2]	0.32	0.31	0.32	0.36	0.37	0.37	0.44	0.38	0.35	0.34	0.37	0.37	0.52	0.57	0.63	0.64
	2DGS [7]	0.38	0.39	0.48	0.39	0.68	0.71	0.69	0.71	0.51	0.54	0.56	0.64	0.74	0.77	0.80	0.80
	MVGSR	0.30	0.30	0.34	0.41	0.37	0.38	0.44	0.38	0.42	0.36	0.43	0.52	0.51	0.51	0.54	0.55
d2s	PGSR [2]	0.33	0.33	0.33	0.33	0.40	0.40	0.42	0.43	0.38	0.38	0.38	0.41	0.53	0.65	0.73	0.79
	2DGS [7]	0.33	0.35	0.43	0.34	0.55	0.57	0.60	0.60	0.49	0.50	0.56	0.61	0.74	0.76	0.79	0.83
	MVGSR	0.27	0.28	0.35	0.49	0.35	0.37	0.48	0.34	0.48	0.34	0.46	0.61	0.51	0.52	0.56	0.60
s2d	PGSR [2]	0.32	0.31	0.32	0.36	0.37	0.37	0.44	0.38	0.35	0.34	0.37	0.37	0.51	0.50	0.53	0.49
	2DGS [7]	0.44	0.43	0.53	0.44	0.81	0.86	0.78	0.81	0.54	0.59	0.55	0.67	0.83	0.93	0.91	0.88
	MVGSR	0.33	0.33	0.33	0.33	0.40	0.39	0.49	0.41	0.37	0.38	0.40	0.42	0.52	0.50	0.53	0.49

Table 5. Quantitative results of reconstruction performance on the DTU-Robust dataset. The Best results are highlighted.

B. Comparison in Masking

As shown in Fig. 9, the semantic features of disturbed images can visually distinguish the distractors. By comparing features from multiple views, it is possible to obtain hints of disturbances before training. SAM is then used to refine the distractor masks. These optimized masks can enhance the model's accuracy in handling complex scenes. This approach not only allows us to identify potential disturbances before training but also enables dynamic adjustments during the training process, improving overall performance and robustness. Experiments on the DTU-Robust dataset demonstrate that this method achieves significant improvements across various scenarios.

scan		24				37				40				55			
rate		0.3	0.5	0.8	1.0	0.3	0.5	0.8	1.0	0.3	0.5	0.8	1.0	0.3	0.5	0.8	1.0
PSNR↑	3DGS [13]	26.21	25.46	25.26	24.74	24.56	24.92	23.84	22.25	22.86	23.55	22.84	22.05	30.34	29.35	28.07	25.99
	PGSR [2]	31.86	31.49	30.68	31.31	27.51	27.07	26.56	25.73	30.83	30.56	29.48	28.66	32.95	32.11	31.73	30.44
	2DGS [7]	33.47	28.94	27.87	30.96	30.96	31.07	30.86	29.94	29.94	27.89	32.77	21.02	21.02	23.92	30.44	33.22
	MVGSR	32.22	31.91	31.34	31.30	29.28	28.56	28.69	27.21	30.65	30.44	30.13	29.64	33.18	33.57	33.28	32.94

scan		63				65				69				83			
rate		0.3	0.5	0.8	1.0	0.3	0.5	0.8	1.0	0.3	0.5	0.8	1.0	0.3	0.5	0.8	1.0
PSNR↑	3DGS [13]	26.91	26.78	24.56	23.20	29.32	27.69	25.53	24.57	28.64	27.29	25.54	24.38	25.96	25.70	25.06	23.39
	PGSR [2]	33.33	32.77	32.54	32.00	32.94	31.88	30.44	29.80	31.79	30.99	30.96	30.02	32.79	32.05	32.00	31.22
	2DGS [7]	33.22	32.99	31.47	30.37	36.38	34.86	33.52	31.82	31.82	33.57	31.94	31.04	31.04	29.93	32.34	32.16
	MVGSR	32.38	32.25	31.94	31.07	32.03	31.49	30.97	29.55	30.98	30.72	30.24	29.41	31.40	31.40	31.09	30.53

scan		97				105				106				110			
rate		0.3	0.5	0.8	1.0	0.3	0.5	0.8	1.0	0.3	0.5	0.8	1.0	0.3	0.5	0.8	1.0
PSNR↑	3DGS [13]	23.35	25.67	24.89	24.19	24.21	24.36	23.33	22.89	33.98	34.21	33.24	31.19	35.31	34.57	34.50	33.98
	PGSR [2]	30.80	30.89	30.30	29.49	33.63	33.09	32.38	31.92	35.40	35.43	35.05	33.86	34.35	32.71	33.96	33.28
	2DGS [7]	32.16	31.74	30.59	33.58	33.58	32.99	19.52	18.21	31.01	30.68	30.33	29.84	29.84	33.80	33.42	32.31
	MVGSR	30.19	29.61	29.70	29.48	33.04	32.76	31.40	31.20	35.85	35.95	36.17	35.99	34.63	34.99	34.77	34.74

scan		114				118				122				Avg.			
rate		0.3	0.5	0.8	1.0	0.3	0.5	0.8	1.0	0.3	0.5	0.8	1.0	0.3	0.5	0.8	1.0
PSNR↑	3DGS [13]	31.59	27.46	28.19	29.55	35.13	34.89	32.51	32.60	32.69	31.54	29.48	27.09	28.74	28.23	27.12	26.33
	PGSR [2]	32.47	32.54	32.02	31.61	37.23	37.04	35.71	35.67	36.71	36.01	35.67	34.46	32.97	32.44	31.97	31.30
	2DGS [7]	32.73	32.47	26.54	31.52	37.30	37.12	35.56	35.18	36.74	35.89	35.52	35.32	32.08	31.86	30.84	30.43
	MVGSR	32.17	32.20	31.46	31.20	37.26	37.47	37.04	37.13	36.10	35.78	35.16	34.23	32.76	32.61	32.22	31.71

Table 6. Quantitative results of novel view synthesis (PSNR↑) on DTU-Robust. The Best and Second Best results are highlighted, respectively.

C. NeRF On-the-Go Dataset

We show the rendering performance on the NeRF on-the-go dataset [22] compared against PGSR [2] and 2DGS [7]. As shown in Tab. 4, our method shows consistently higher PSNR and maintains better or competitive SSIM and LPIPS on different scenes. Especially in *fountain*, ours surpasses the 2DGS by 1.08 and PGSR by 1.13 in terms of PSNR. On average, ours achieves 20.92 dB in terms of PSNR and demonstrates better performance than PGSR and 2DGS. The higher results across scenes indicate that our method has robust and strong novel view synthesis quality.

References

[1] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5470–5479, 2022. 2

[2] Danpeng Chen, Hai Li, Weicai Ye, Yifan Wang, Weijian Xie, Shangjin Zhai, Nan Wang, Haomin Liu, Hujun Bao, and Guofeng Zhang. Pgsr: Planar-based gaussian splatting for efficient and high-fidelity surface reconstruction. *arXiv preprint arXiv:2406.06521*, 2024. 3, 6, 7, 8, 9, 10, 11, 12

[3] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19436–19446, 2024. 3

[4] Kai Cheng, Xiaoxiao Long, Kaizhi Yang, Yao Yao, Wei Yin, Yuexin Ma, Wenping Wang, and Xuejin Chen. Gaussianpro: 3d gaussian splatting with progressive propagation. In *Forty-first International Conference on Machine Learning*, 2024. 2, 4

[5] Yan Di, Chenyangguang Zhang, Chaowei Wang, Ruida Zhang, Guangyao Zhai, Yanyan Li, Bowen Fu, Xiangyang Ji, and Shan Gao. Shapematcher: Self-supervised joint shape canonicalization segmentation retrieval and deformation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21017–21028, 2024. 2

[6] Antoine Guédon and Vincent Lepetit. Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5354–5363, 2024. 2

[7] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024. 1, 2, 3, 6, 7, 8, 9, 10, 11, 12

[8] Slobodan Ilic and Pascal Fua. Implicit meshes for surface reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(2):328–333, 2005. 2

[9] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568, 2011. 2

[10] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engin Tola, and Henrik Aanæs. Large scale multi-view stereopsis evalu-

ation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 406–413, 2014. 6

[11] Yingwenqi Jiang, Jiadong Tu, Yuan Liu, Xifeng Gao, Xiaoxiao Long, Wenping Wang, and Yuexin Ma. Gaussianshader: 3d gaussian splatting with shading functions for reflective surfaces. *arXiv preprint arXiv:2311.17977*, 2023. 4

[12] Mert Asim Karaoglu, Hannah Schieber, Nicolas Schischka, Melih Görgülü, Florian Grötzner, Alexander Ladikos, Daniel Roth, Nassir Navab, and Benjamin Busam. Dynamon: Motion-aware fast and robust camera localization for dynamic nerf. *arXiv preprint arXiv:2309.08927*, 2023. 3

[13] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):1–14, 2023. 2, 3, 4, 7, 10, 12

[14] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023. 5

[15] Yanyan Li, Chenyu Lyu, Yan Di, Guangyao Zhai, Gim Hee Lee, and Federico Tombari. Geogaussian: Geometry-aware gaussian splatting for scene rendering. *arXiv preprint arXiv:2403.11324*, 2024. 2, 4

[16] Fangchang Ma and Sertac Karaman. Sparse-to-dense: Depth prediction from sparse depth samples and a single image. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 4796–4803. IEEE, 2018. 2

[17] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7210–7219, 2021. 3

[18] 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

[19] Raúl Mur-Artal and Juan D Tardós. Probabilistic semi-dense mapping from highly accurate feature-based monocular slam. In *Robotics: Science and Systems*. Rome, 2015. 2

[20] Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5589–5599, 2021. 2

[21] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 4, 5

[22] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8931–8940, 2024. 5, 6, 8, 9, 10, 12

[23] Antoni Rosinol, John J Leonard, and Luca Carlone. Nerf-slam: Real-time dense monocular slam with neural radiance fields. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3437–3444. IEEE, 2023. 2

[24] Sara Sabour, Suhani Vora, Daniel Duckworth, Ivan Krasin, David J Fleet, and Andrea Tagliasacchi. Robustnerf: Ignoring distractors with robust losses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20626–20636, 2023. 2, 3, 4

[25] Sara Sabour, Lily Goli, George Kopanas, Mark Matthews, Dmitry Lagun, Leonidas Guibas, Alec Jacobson, David J. Fleet, and Andrea Tagliasacchi. SpotLessSplats: Ignoring distractors in 3d gaussian splatting. *arXiv:2406.20055*, 2024. 2, 4

[26] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. 2

[27] Steven M Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, pages 519–528. IEEE, 2006. 2

[28] Jiaxiang Tang, Hang Zhou, Xiaokang Chen, Tianshu Hu, Errui Ding, Jingdong Wang, and Gang Zeng. Delicate textured mesh recovery from nerf via adaptive surface refinement. *International Conference on Computer Vision (ICCV)*, 2023. 2

[29] Shimon Ullman. The interpretation of structure from motion. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 203(1153):405–426, 1979. 2

[30] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021. 2

[31] Lior Yariv, Peter Hedman, Christian Reiser, Dor Verbin, Pratul P Srinivasan, Richard Szeliski, Jonathan T Barron, and Ben Mildenhall. Bakedsdf: Meshing neural sdfs for real-time view synthesis. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–9, 2023. 2

[32] Jae-Chern Yoo and Tae Hee Han. Fast normalized cross-correlation. *Circuits, Systems and Signal Processing*, 28: 819–843, 2009. 6

[33] Zehao Yu, Torsten Sattler, and Andreas Geiger. Gaussian opacity fields: Efficient and compact surface reconstruction in unbounded scenes. *arXiv preprint arXiv:2404.10772*, 2024. 3

[34] Raza Yunus, Yanyan Li, and Federico Tombari. Manhattanslam: Robust planar tracking and mapping leveraging mixture of manhattan frames. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6687–6693. IEEE, 2021. 2

[35] Dongbin Zhang, Chuming Wang, Weitao Wang, Peihao Li, Minghan Qin, and Haoqian Wang. Gaussian in the wild: 3d gaussian splatting for unconstrained image collections. In

European Conference on Computer Vision, pages 341–359. Springer, 2025. 2

[36] Ruida Zhang, Chengxi Li, Chenyangguang Zhang, Xingyu Liu, Haili Yuan, Yanyan Li, Xiangyang Ji, and Gim Hee Lee. Street gaussians without 3d object tracker, 2024. 2