Depth-Regularized Optimization for 3D Gaussian Splatting in Few-Shot Images

Jaeyoung Chung[1] Jeongtaek Oh[2] Kyoung Mu Lee[1,2]
[1]ASRI, Department of ECE, [2]IPAI, Seoul National University, Seoul, Korea
{robot0321,ohjtgood,kyoungmu}@snu.ac.kr

Abstract

This paper presents a method to optimize Gaussian splatting with a limited number of images while avoiding overfitting. Representing a 3D scene by combining numerous Gaussian splats has yielded outstanding visual quality. However, it tends to overfit the training views when only a few images are available. To address this issue, we employ an adjusted depth map as a geometric reference, derived from a pre-trained monocular depth estimation model and subsequently aligned with the sparse structure-from-motion points. We regularize the optimization process of 3D Gaussian splatting with the adjusted depth and an additional unsupervised smooth constraint, thereby effectively reducing the occurrence of floating artifacts. Our method is mainly validated on the NeRF-LLFF dataset with varying numbers of images, and we conduct multiple experiments with randomly selected training images, presenting the average value to ensure fairness. Our approach demonstrates robust geometry compared to the original method, which relied solely on images.

1. Introduction

Reconstructing three-dimensional space from images has long been a challenge in computer vision. Recent advancements show the feasibility of photorealistic novel view synthesis [3, 32], igniting research into reconstructing a complete 3D space from images. Driven by progress in computer graphics techniques and industry demand, particularly in sectors such as virtual reality [14] and mobile [11], research on achieving high-quality and high-speed real-time rendering has been ongoing. Among the recent notable developments, 3D Gaussian Splatting (3DGS) [24] stands out through its combination of high quality, rapid reconstruction speed, and support for real-time rendering. 3DGS employs Gaussian attenuated spherical harmonic splats [12, 39] with opacity as primitives to represent every scene part. It guides the splats to construct a consistent geometry by imposing a constraint on the splats to satisfy multiple images at the same time.

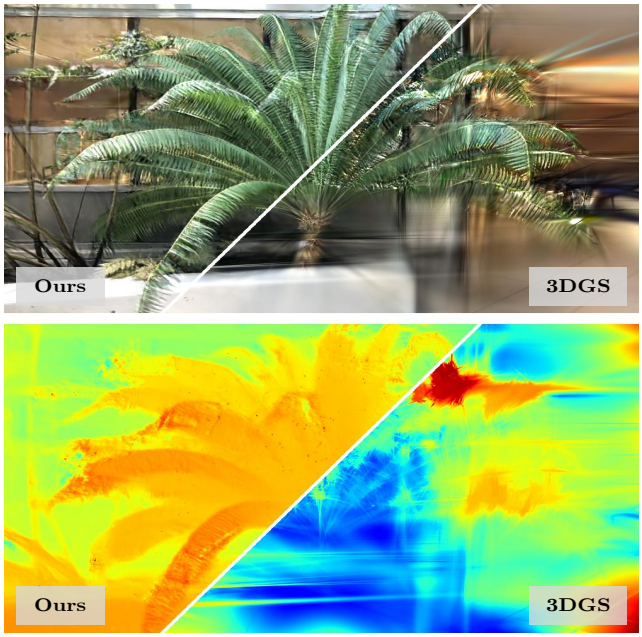


Figure 1. **The effectiveness of depth regularization in a few-shot setting** We present the results optimized with only *two* images. Compared to the degraded 3DGS due to overfitting, our proposed method utilized depth guidance estimated from the images to mitigate overfitting, resulting in geometry of high quality.

The approach of aggregating small splats for a scene provides the capability to express intricate details, yet it is prone to overfitting due to its local nature. 3DGS [25] optimizes independent splats according to multi-view color supervision without global structure. Hence, when there is an insufficient number of images that can not provide global geometric cues, 3DGS experiences overfitting during the optimization process. The limited geometric information from the few images leads to an incorrect convergence toward a local optimum, resulting in optimization failure or floating artifacts as shown in Figure 1. Nevertheless, the capability to reconstruct a 3D scene with a restricted number of images is crucial for practical applications, prompting us to tackle the few-shot optimization problem.

One intuitive solution is to supplement an additional ge-

ometric cue such as depth. In numerous 3D reconstruction contexts [6, 23, 41], depth map proves immensely valuable for reconstructing 3D scenes by providing direct geometric information. Utilizing a depth sensor aligned with an RGB camera enables the direct acquisition of such dense depth maps with minimal errors, yet the requirement for such equipment poses obstacles to practical applications. Structure-from-motion (SfM) is another method for obtaining geometry information, which optimizes camera parameters and 3D points by matching the 2D feature points across multi-view images. 3DGS also utilizes SfM, particularly COLMAP [42], to initialize the cameras and Gaussian splats. However, the 3D feature points estimated from the SfM algorithm encounter a notable scarcity with few images. The sparse nature of the point cloud makes it impractical to guide the global geometry of the Gaussian splats. Hence, a method for inferring dense depth maps is essential. One of the methods to extract dense depth from images is utilizing a monocular depth estimation model or a depth completion model. While the models can infer dense depth maps from individual images based on priors obtained from the data, they produce only relative depth due to scale ambiguity. Since the scale ambiguity leads to critical geometry conflicts in multi-view images, we need to adjust scales to prevent the conflict between independently inferred depths.

In this paper, we propose a method to represent 3D scenes using a small number of RGB images leveraging prior information from a pre-trained monocular depth estimation model [5] and a smoothness constraint. We adapt the scale and offset of the estimated depth to the sparse COLMAP points, solving the scale ambiguity. We use the adjusted depth as a geometry guide to assist color-based optimization, reducing floating artifacts and satisfying geometry conditions. We prevent the overfitting problem by incorporating an early stop strategy, where the optimization process stops when the depth-guide loss starts to rise. Moreover, we apply a smoothness constraint to achieve stability, ensuring neighbor 3D points have similar depths. We adopt 3DGS as our baseline and compare the performance of our method in the NeRF-LLFF [31] dataset. We confirm that our strategy leads to plausible results not only in terms of RGB novel-view synthesis but also 3D geometry reconstruction. Further experiments demonstrate the influence of geometry cues such as depth and initial points on Gaussian splatting. They significantly influence the stable optimization of Gaussian splatting.

In summary, our contributions are as follows:

- We propose a depth-guided Gaussian Splatting optimization strategy that enables optimizing the scene with a few images, mitigating the overfitting issue. We demonstrate that even an estimated depth adjusted with a sparse point cloud, an outcome of the SfM pipeline, can play a vital role in geometric regularization.
- We present a novel early stop strategy: *halting* the training process when depth-guided loss suffers to drop. We illustrate the influence of each strategy through thorough ablation studies.
- We show that adopting a smoothness term for the depth map guides the model to find the correct geometry. Comprehensive experiments reveal enhanced performance attributed to the inclusion of a smoothness term.

2. Related Work

Novel view synthesis Structure from motion (SfM) [47] and Multi-view stereo (MVS) [46] are techniques for reconstructing 3D structures using multiple images, which have been studied for a long time in the computer vision field. Among the continuous developments, COLMAP [42] is a widely used representative tool for SfM. COLMAP performs camera pose calibration and finds sparse 3D key points using the epipolar constraint [22] of multi-view images. For more dense and realistic reconstruction, deep learning based 3D reconstruction techniques have been mainly studied. [21, 32, 52] Among them, Neural radiance fields (NeRF) [32] is a representative method that uses a neural network as a representation method. NeRF creates realistic 3D scenes using an MLP network as a 3D space expression and volume rendering, producing many follow-up papers on 3D reconstruction research. [3, 4, 18, 45, 48, 55] In particular, to overcome slow speed of NeRF, many efforts continues to achieve real-time rendering by utilizing explicit expression such as sparse voxels[16, 28, 44, 57], featured point clouds[53], tensor [10], polygon [11]. These representations have local elements that operate independently, so they show fast rendering and optimization speed. Based on this idea, various representations such as Multi-Level Hierarchies [33, 34], infinitesimal networks [19, 40], triplane [9] have been attempted. Among them, 3D Gaussian splatting [24] presented a fast and efficient method through alpha-blending rasterization instead of time-consuming volume rendering. It optimizes a 3D scene using multi-million Gaussian attenuated spherical harmonics with opacity as a primitive, showing easy and fast 3D reconstruction with high quality.

Few-shot 3D reconstruction Since an image contains only partial information about the 3D scene, 3D reconstruction requires many multi-view images. COLMAP uploads feature points matched between multiple images onto 3D space, thus increasing the number of images enhances the reliability of the 3D points and the camera poses. [17, 42] NeRF also optimizes the color and geometry of a 3D scene based on the pixel colors of many images to obtain high-quality scenes. [49, 58] However, the requirements for many images hindered practical application, sparking research on

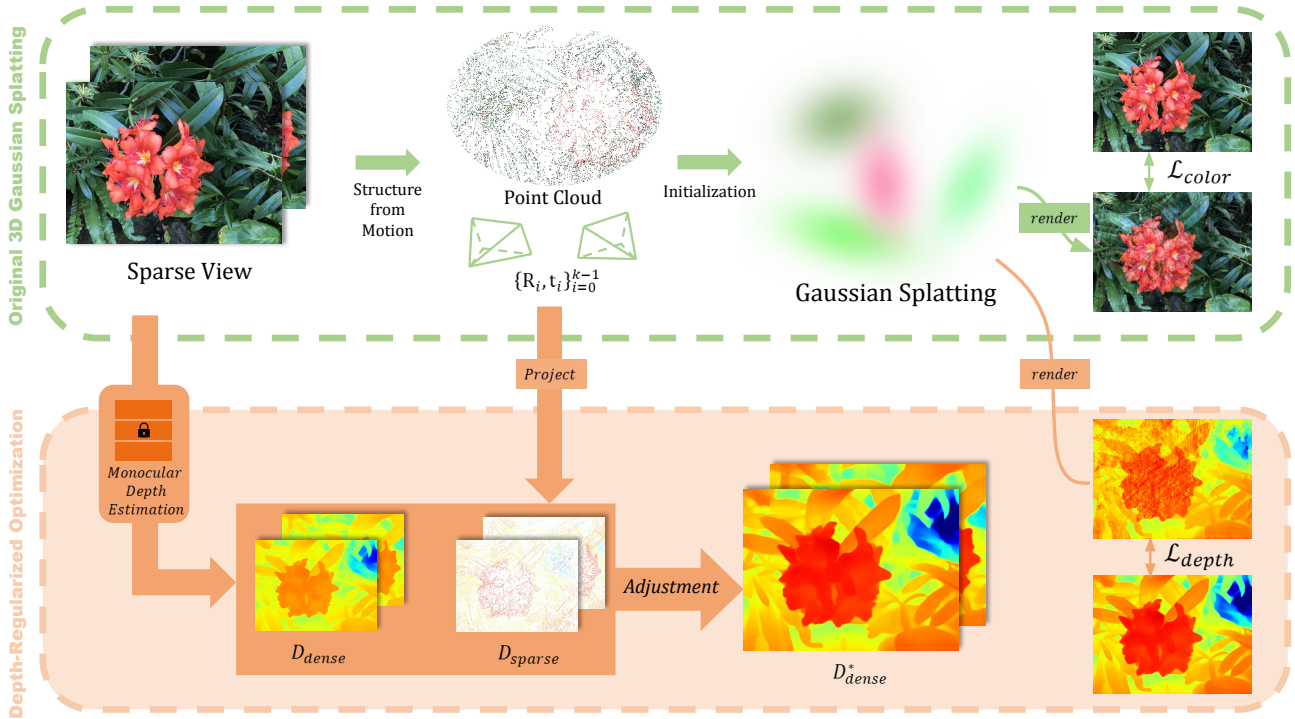


Figure 2. **Overview.** We optimize the 3D Gaussian splatting [24] using dense depth maps adjusted to the point clouds obtained from COLMAP [42]. By incorporating depth maps to regulate the geometry of the 3D scene, our model successfully reconstructs scenes using a limited number of images.

3D reconstruction using only a few images. Many studies in few-shot 3D reconstruction utilize depth to provide valuable geometric cues for creating 3D scenes. Depth helps reduce the effort of inferring geometry through color consensus across multiple images in various ways, including a surface smoothness constraint [26, 36], a sparse depth supervision obtained from COLMAP [13, 50], a dense depth map obtained from additional sensors [2, 7, 15], or an estimated dense depth map from the pre-trained neural network. [35, 38, 41] These studies regularize geometry based on the globality of the neural network, so it is difficult to apply them to representations with large locality such as sparse voxel [16] or feature point [53]. Many studies attempted to establish connectivity between local elements in a 3D space through the total variation (TV) loss [16, 54, 60]. Still, it requires exhaustive hyperparameter tuning of the total variation, which varies on the scene and location. 3DGS [24] generates floating artifacts with a small number of images due to its strong locality. Since 3DGS utilizes SfM for initialization, the feature points obtained during this process can be employed as a cost-free depth guide. However, due to the limited availability of images, the sparse nature of the feature points makes it challenging for them to serve as meaningful depth guidance. Hence we use a coarse geometry guide for optimization through a pretrained depth estimation model [5, 30, 59]. The estimated dense depth provides rough guidance to the location of splats, which significantly contributes to optimization stability in few-shot situations and helps eliminate floating artifacts that occur in random places.

3. Method

Our method addresses the optimization problem in the few-shot setting with k images $\{I_i\}_{i=1}^{k}, I_i \in [0,1]^{H\times W\times 3}$, where i denotes the camera number. As a preprocessing, we run SfM (such as COLMAP[42]) pipeline and get the each camera pose $R_i \in \mathbb{R}^{3\times 3}, t_i \in \mathbb{R}^3$, intrinsic parameters $K \in \mathbb{R}^{3\times 3}$, and a point cloud $P \in \mathbb{R}^{3\times n}$. From the result of SfM, we derive a sparse depth $D_{\text{spr},i}$ for each image by projecting all visible points onto the pixel space. Subsequently, we utilize a depth estimation network to estimate a depth map for each image, which is then adjusted to the sparse depth (Section 3.1). We apply regularization to the geometry of Gaussian splats by incorporating the dense depth prior obtained through the rasterization process. (Section 3.2). We add another regularization for smoothness between depths of adjacent pixels (Section 3.3) and refine optimization options for few-shot settings (Section 3.4).

3.1. Preparing Dense Depth Prior

To guide the splats into plausible geometry, we need to provide global geometry information due to the locality of

Gaussian splats. The dense depth is one of the promising geometry prior, but there is a challenge in constructing it. From the result of SfM, we can obtain a sparse depth map $D_{\text{spr},i}$ for each image by projecting all visible points to the pixel space:

$$p = K[\mathrm{R_i}|\mathrm{t_i}]P, \tag{1}$$

$$D_{\text{spr},i} = \{p_z\} \in [0, \infty]. \tag{2}$$

The density of SfM points depends on the number of images, resulting in insufficient valid points to estimate dense depth directly in a few-shot setting. (For example, SfM reconstruction from 19 images creates a sparse depth map with an average of 0.04% valid pixels. [41]) Even the latest depth completion models fail to complete dense depth due to the significant information gap.

When designing the depth prior, it is important to note that even rough depth significantly aids in guiding the splats and eliminating artifacts resulting from splats trapped in incorrect geometry. Hence, we employ a state-of-the-art monocular depth estimation model and incorporate scale matching to offer a coarse, dense depth guide for optimization. From a train image I_i, the monocular depth estimation model F_θ outputs dense depth,

$$D_{\text{den},i} = s \cdot F_\theta(I_i) + t. \tag{3}$$

To resolve the scale ambiguity in the estimated dense depth $D_{\text{den},i}$, we adjust the scale s and offset t of estimated depth to sparse SfM depth $D_{\text{spr},i}$:

$$s^*, t^* = \arg\min_{s,t} \sum_{p \in D_{\text{spr},i}} \left\| w(p) \cdot D_{\text{spr},i}(p) - D_{\text{den},i}(p; s, t) \right\|^2, \tag{4}$$

where $w \in [0, 1]$ is a normalized weight presenting the reliability of each feature point calculated as the reciprocal of the reprojection error from SfM. Finally, we use the adjusted dense depth $D^*_{\text{den}} = s^* \cdot F_\theta(I) + t^*$ to regularize the optimization loss of Gaussian splatting.

3.2. Depth Rendering through Rasterization

3D Gaussian splatting utilizes a rasterization pipeline [1] to render the disconnected and unstructured splats leveraged on the parallel architecture of GPU. Based on differentiable point-based rendering techniques [27, 51, 56], they render an image by rasterizing the splats through α-blending. Point-based approaches exploit a similar equation to NeRF-style volume rendering, rasterizing a pixel color with ordered points that cover that pixel,

$$C = \sum_{j \in N} c_j \alpha_j T_j \tag{5}$$

$$\text{where} \quad T_j = \prod_{k=1}^{j-1} (1 - \alpha_k),$$

C is the pixel color, c is the color of splats, and α here is learned opacity multiplied by the covariance of 2D Gaussian. This formulation prioritizes the color c of opaque splat positioned closer to the camera, significantly impacting the final outcome C. Inspired by the depth implementation in NeRF, we integrate the depth of each splat similar to Eqn. (5). However, in areas with an insufficient number of primitives, alpha integration may not be completed. So we adopt a method of normalizing through alpha integration to ensure appropriate depth calculation as,

$$D = \frac{\sum_{j \in N} d_j \alpha_j T_j}{\sum_{j \in N} \alpha_j T_j}, \tag{6}$$

where D is the rendered depth and $d_j = (R_i p_j + T_i)_z$ is the depth of each j-th splat from the i-th camera. Eqn. (6) enables the direct utilization of α_i and T_i calculated in Eqn. (5), facilitating rapid depth rendering with minimal computational load. Finally, we guide the rendered depth to the estimated dense depth using L1 distance,

$$\mathcal{L}_{depth} = \left\| D - D^*_{\text{den}} \right\|_1. \tag{7}$$

This depth loss serves as guidance for geometry, mitigating the risk of overfitting.

3.3. Unsupervised Smoothness Constraint

Even though each estimated depth is fitted to the COLMAP points, the adjusted depths D^*_{den} serve as a rough guide and often lead to conflicts in multi-view consistency, particularly in detailed areas. To alleviate this, we introduce an unsupervised constraint for geometry smoothness inspired by [20] to regularize the conflict. This constraint implies that points in similar 3D positions have similar depths on the image plane. We utilize the Canny edge detector [8] as a mask to ensure that it does not regularize the area with significant differences in depth along the boundaries. For a depth d_i and its adjacent depth d_j, we regularize the difference between them:

$$\mathcal{L}_{smooth} = \sum_{d_j \in \text{adj}(d_i)} \mathbb{1}_{ne}(d_i, d_j) \cdot \left\| d_i - d_j \right\|^2 \tag{8}$$

where $\mathbb{1}_{ne}$ is a indicator function that presents whether both depths are *not* in edge. Here, we assume that neighboring pixels have similar depths excluding the edges. While this assumption may not precisely align with reality, it effectively serves as a regularize.

We conclude the final loss terms by incorporating the depth loss from Eqn. (7) and smoothness loss and smoothness loss from Eqn. (8) with their own hyperparameters λ_{depth} and λ_{smooth}:

$$\begin{aligned} \mathcal{L} = (1 - \lambda_{ssim}) \mathcal{L}_{color} + \lambda_{ssim} \mathcal{L}_{D-SSIM} \\ + \lambda_{depth} \mathcal{L}_{depth} + \lambda_{smooth} \mathcal{L}_{smooth} \end{aligned} \tag{9}$$

			PSNR↑				SSIM↑				LPIPS↓			
			2-view	3-view	4-view	5-view	2-view	3-view	4-view	5-view	2-view	3-view	4-view	5-view
NeRF-LLFF[32]	Fern	3DGS	13.03	14.29	16.73	18.59	0.336	0.408	0.517	0.603	0.476	0.389	0.296	0.217
		Ours	**17.59**	**19.13**	**19.91**	**20.55**	**0.516**	**0.588**	**0.616**	**0.642**	**0.286**	**0.232**	**0.203**	**0.167**
		Oracle	18.18	20.30	20.78	21.81	0.524	0.636	0.654	0.701	0.278	0.201	0.185	0.157
	Flower	3DGS	14.90	17.75	**19.71**	**21.39**	0.351	**0.508**	**0.605**	**0.671**	**0.406**	**0.257**	**0.190**	**0.146**
		Ours	**15.92**	**17.80**	19.15	20.45	**0.395**	0.445	0.538	0.576	0.414	0.376	0.323	0.293
		Oracle	19.71	22.16	23.26	24.65	0.570	0.673	0.714	0.760	0.250	0.163	0.128	0.097
	Fortress	3DGS	13.87	15.98	19.26	19.98	0.363	0.492	0.609	0.631	0.389	0.283	0.201	0.191
		Ours	**19.80**	**21.85**	**23.07**	**23.72**	**0.567**	**0.655**	**0.724**	**0.740**	**0.232**	**0.191**	**0.162**	**0.144**
		Oracle	23.07	24.51	26.39	26.73	0.654	0.728	0.787	0.797	0.159	0.130	0.100	0.093
	Horns	3DGS	11.43	12.48	13.76	14.75	0.264	0.339	0.433	0.498	0.531	0.464	0.395	0.350
		Ours	**15.91**	**16.22**	**18.09**	**18.39**	**0.420**	**0.466**	**0.527**	**0.565**	**0.362**	**0.349**	**0.306**	**0.296**
		Oracle	18.56	20.08	20.88	22.52	0.568	0.644	0.668	0.725	0.259	0.212	0.199	0.168
	Leaves	3DGS	12.33	12.36	12.49	12.26	**0.260**	**0.275**	**0.298**	**0.297**	**0.412**	**0.397**	**0.397**	**0.401**
		Ours	**13.04**	**13.63**	**13.97**	**14.13**	0.235	0.270	0.283	**0.297**	0.460	0.445	0.440	0.438
		Oracle	13.52	14.23	14.78	14.85	0.287	0.353	0.377	0.397	0.380	0.348	0.341	0.356
	Orchids	3DGS	11.78	13.94	**15.41**	16.08	0.182	**0.320**	**0.416**	**0.460**	**0.426**	**0.310**	**0.245**	**0.219**
		Ours	**12.88**	**14.71**	15.40	**16.13**	**0.216**	0.297	0.343	0.391	0.462	0.383	0.366	0.352
		Oracle	14.89	16.45	17.42	18.45	0.365	0.471	0.525	0.576	0.303	0.237	0.200	0.174
	Room	3DGS	10.18	11.51	11.59	12.21	0.404	0.494	0.510	0.552	0.606	0.559	0.556	0.515
		Ours	**17.21**	**18.11**	**18.87**	**19.63**	**0.668**	**0.719**	**0.732**	**0.757**	**0.352**	**0.360**	**0.326**	**0.295**
		Oracle	20.66	22.31	23.80	24.59	0.758	0.801	0.839	0.864	0.217	0.188	0.160	0.156
	Trex	3DGS	10.72	11.72	13.11	14.14	0.322	0.417	0.492	0.548	0.520	0.446	0.394	0.351
		Ours	**14.90**	**15.90**	**16.75**	**17.37**	**0.480**	**0.537**	**0.567**	**0.625**	**0.358**	**0.362**	**0.348**	**0.305**
		Oracle	17.76	19.58	20.84	22.83	0.591	0.669	0.714	0.786	0.284	0.226	0.192	0.134
Mean		3DGS	12.25	13.75	15.26	16.17	0.306	0.407	0.485	0.533	0.471	0.388	0.334	0.299
		Ours	**15.94**	**17.17**	**18.15**	**18.74**	**0.439**	**0.497**	**0.541**	**0.571**	**0.365**	**0.337**	**0.309**	**0.288**
		Oracle	18.29	19.95	21.02	22.05	0.539	0.622	0.660	0.701	0.266	0.213	0.188	0.167

Table 1. Quantitative results in NeRF-LLFF [31] dataset. The best performance except oracle is **bolded**.

where the preceding two loss terms $\mathcal{L}_{color}, \mathcal{L}_{D-SSIM}$ correspond to the original 3D Gaussian splatting losses. [24]

3.4. Modification for Few-Shot Learning

We modify two optimization techniques from the original paper to create 3D scenes with limited images. The techniques employed in 3DGS were designed under the assumption of utilizing a substantial number of images, potentially hindering convergence in a few-shot setting. Through iterative experiments, we confirm this and modify the techniques to suit the few-shot setting. Firstly, we set the maximum degree of spherical harmonics (SH) to 1. This prevents the overfitting of spherical harmonic coefficients responsible for high frequencies due to insufficient information. Secondly, we implement an early-stop policy based on depth loss. We configure Eqn. (9) to be primarily driven by the color loss while employing the depth loss and the smoothness loss as guiding factors. Hence, overfitting emerges due to the predominant influence of color loss. We use a moving averaged depth loss to halt optimization when the splats start to deviate from the depth guide. Lastly, we remove the periodic reset process. We observe that resetting all splats' opacity α leads to irreversible and detrimental consequences. Due to a lack of information from the limited images, the inability to restore the opacity of splats led to scenarios where all splats were removed or trapped in local optima, causing unexpected outcomes and optimization failures. As a result of the techniques above, we achieve stable optimization in few-shot learning.

4. Experiment

4.1. Experiment settings

Datasets. We evaluate our method on NeRF-LLFF [31] dataset. NeRF-LLFF includes eight scenes with forward-facing cameras, and we split the images of each scene into train and test sets. We use the image outer edge of the camera group as the train set based on the convex hull algorithm [37] due to the forward-facing camera distribution. We optimize the scene with k-shot (k=2,3,4,5) images randomly selected from the train set and evaluate on the same test set. We use ten randomly selected seeds and report the average of ten experiments.

Implementation details. For a fair comparison among different options, using unified coordinates in each scene and standardizing the evaluation values is essential. We achieve this by processing all the images of a scene through COLMAP to obtain consistent camera poses and feature points, selecting those relevant to each k-shot experiment.

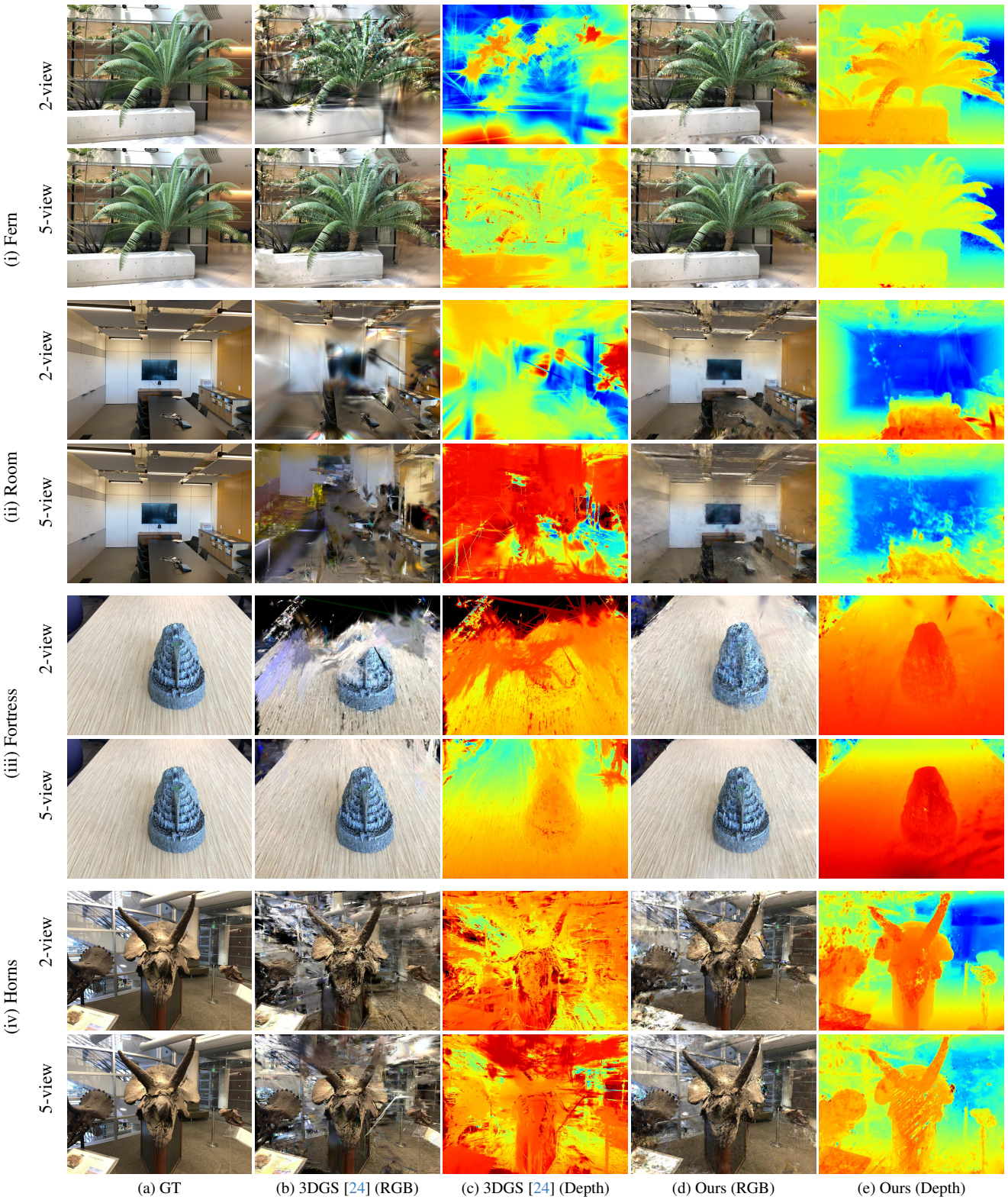
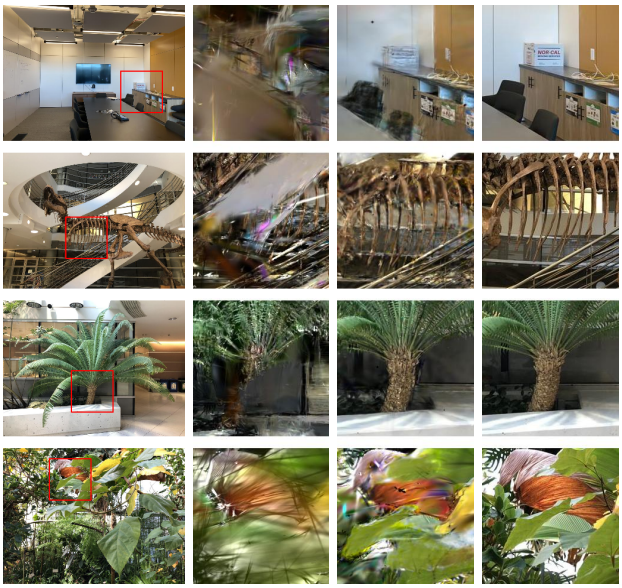


Figure 3. **Qualitative comparison in NeRF-LLFF [31] dataset.** We visualize the distinction between 3D Gaussian Splatting (3DGS) [24] and our method in both 2-view and 5-view settings. Driven primarily by color loss, 3DGS struggled to achieve desirable geometry. Our approach consistently established plausible geometric structures with depth guidance, resulting in superior reconstruction outcomes.



(a) (b) (c) (d)

Figure 4. **Details in cropped patches.** (a) Input View (b) 3DGS [24] (c) Ours (d) Ground Truth. Our method produces superior reconstruction results compared to 3DGS [24], leveraging additional geometric cues. Our method establishes stable geometry, outperforming 3DGS in reconstruction quality.

We select k cameras from the train set and filter the feature points that are visible in at least three out of the k cameras. We use these feature points as depth guidance D_{sparse} in Eqn. (4) and initial points for Gaussian splatting optimization. In the baseline(3DGS), we use the same k camera poses and filtered initial points, reporting the evaluation values at 30k iterations like in the original setup. For the *oracle*, we aim to illustrate the effectiveness of precise depth. We create a pseudo ground truth (GT) depth map by optimizing the entire train and test images and then utilize it instead of the estimated depth. Lastly, based on CUDA, we implement the differentiable depth rasterizer outlined in Eqn. (6).

4.2. Experiment results

We present the comparison results of 3DGS, our method, and oracle for NeRF-LLFF scenes in Table 1. Across all methods and scenes, a decrease in the number of used images consistently results in lower visual quality. Our method typically demonstrates superior results compared to 3DGS, mainly when the number of images is limited. Figure 3 visualize the difference between 3DGS and our method. The depth map highlights the geometry failure of 3DGS in the few-shot case. For instance, the 2-view of the *Fern* displays entirely erroneous geometry compared to the similarity in RGB. In harsh conditions like the 2-view scenario, 3DGS often fails to form appropriate geometry. In contrast, our method forms plausible geometry while generating an attractive image. We present additional examples in Figure 4. The cropped patches demonstrate that our method achieves better results through depth guidance. Hence, we confirm that the geometric cues provided by depth significantly benefit the reconstruction in Gaussian splatting, especially when the number of images is limited. This fact is reaffirmed by the remarkably high performance of the oracle, which employs accurate geometry. The example images of the oracle demonstrate the effectiveness of accurate depth, as depicted in Figure 5. The rich information provided by pseudo-GT depth enables the creation of detailed and reliable results even with limited images.

An important observation to note is the substantial reliance of our approach on the pre-trained monocular depth estimation model. We exploit the pre-trained model of ZoeDepth [5] trained on the indoor dataset NYU Depth v2 [43] and urban dataset KITTI [29]. As a result, our model reports relatively higher performance in indoor scenes (Fortress, Room, Fern) and comparatively worse results for natural scenes (Orchids, Flower). Note that the *Leaves* presents challenges for COLMAP, leading to generally unsuccessful Gaussian splatting training.

Method	2-views PSNR↑	2-views SSIM↑	2-views LPIPS↓	5-views PSNR↑	5-views SSIM↑	5-views LPIPS↓
w/o Adjustment (Section 3.1)	7.86	0.319	0.740	10.01	0.346	0.761
w/o $\mathcal{L}_{depth}$ (Section 3.2)	11.49	0.344	0.533	12.97	0.506	0.418
w/o $\mathcal{L}_{smooth}$ (Section 3.3)	14.75	0.415	0.391	17.79	0.561	0.297
w/o early stop (Section 3.4)	13.99	0.345	0.433	17.28	0.494	0.333
Ours	15.91	0.420	0.362	18.39	0.565	0.296

Table 2. **Ablations.** We describe the ablation studies on each element of the proposed method. We also present an experiment supervising with sparse depth from COLMAP instead of dense depth. The reported values are evaluated in *Horns*.

Initialization points	2-views PSNR↑	2-views SSIM↑	2-views LPIPS↓	5-views PSNR↑	5-views SSIM↑	5-views LPIPS↓
COLMAP from sparse-view (Ours)	19.80	0.567	0.232	23.72	0.740	0.144
Unprojected from D^{*}_{dense}	16.39	0.457	0.281	19.15	0.569	0.222
COLMAP from all-view	21.18	0.681	0.200	24.09	0.778	0.127

Table 3. **Comparison of Initialization Methods.** We describe the ablation studies on each element of the proposed method. We also present an experiment supervising with sparse depth from COLMAP instead of dense depth. The reported values are evaluated in *Fortress*.

Figure 5. **Example results utilizing pseudo-GT depth (*oracle*).** Accurate depth facilitates high-quality 3D reconstruction, even with a limited number of images. Fine details are perceptible in both RGB and depth.

4.3. Ablations

We present ablation studies for each component of our proposed method in Table 2. The first and second rows demonstrate the necessity of absolute depth guidance. Without the adjustment process in Section 3.1, the dense Depth D_{dense} has an incorrect scale from the monocular depth estimation model. The depth is misaligned with the camera intrinsic parameters from COLMAP, leading to complete training failure. We also observed optimization failure when solely utilizing unsupervised smooth constraints without depth supervision introduced in Section 3.2. Applying smoothness constraints without absolute geometry supervision yields worse results than the baseline. The third and fourth rows of Table 2 demonstrate the degree of performance enhancement from additional techniques. With the depth supervision D^*_{dense}, the smoothness constraints in Section 3.3 contribute to performance improvement by providing additional geometric cues. Notably, the early stop mechanism introduced in Section 3.4 is pivotal in preventing performance degradation within our approach. By leveraging depth loss, it scrutinizes the divergence of splats from the prescribed geometry guide, effectively halting potential instances of overfitting.

In Table 3, we compared the performances between the different Gaussian splatting initializations. The second row illustrates the outcomes when utilizing a point cloud produced by unprojecting dense depth D^*_{dense} as initialization points. The numerous initial points generated through unprojection are not effectively merged or pruned, resulting in lower performance than the sparse COLMAP initialization. On the other hand, the third row showcases the results when all COLMAP points are used, presenting a much better quality than the second row. Employing many unattainable initial points with k images enhances the outcome via dense depth adjustment and Gaussian splatting initialization.

5. Limitation and Future Work

Our approach demonstrated the feasibility of Gaussian splatting optimization in a few-shot setting through depth guidance, yet it has limitations. Firstly, it heavily relies on the estimation performance of the monocular depth estimation model. Moreover, this model's depth estimation performance can vary based on the learned data domain, consequently affecting the performance of Gaussian splatting optimization. Additionally, relying on fitting the estimated depth to COLMAP points means a dependency on COLMAP's performance, rendering it incapable of handling textureless plains or challenging surfaces where COLMAP might fail. We leave the optimization of 3D scenes as a future work by interdependent estimated depths rather than COLMAP points. Also, exploring methods to regularize geometry across various datasets, including areas where depth estimation, such as the sky, might be challenging, is another avenue for future work.

6. Conclusion

In this work, we introduce Depth-Regularized Optimization for 3D Gaussian Splatting in Few-Shot Images, a model for learning 3D Gaussian splatting with a small number of images. Our model regularizes the splats using depth, demonstrating the effectiveness of such geometric guidance. We exploit a monocular depth estimation model to acquire dense depth guidance and adjust the depth scale based on SfM points. We examined the effectiveness of our proposed depth loss, unsupervised smooth constraint, and early stop technique in the NeRF-LLFF dataset. Our method outperforms 3D Gaussian splatting in a few-shot setting, creating plausible geometry. Finally, we demonstrated through additional experiments that improved depth and initialization points significantly enhance Gaussian splatting-based 3D reconstruction performance.

References

[1] Tomas Akenine-Möller, Eric Haines, and Naty Hoffman. *Real-time rendering*. Crc Press, 2019. 4

[2] Dejan Azinović, Ricardo Martin-Brualla, Dan B Goldman, Matthias Nießner, and Justus Thies. Neural rgb-d surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 3

[3] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. 1, 2

[4] 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*, 2022. 2

[5] Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288*, 2023. 2, 3, 7

[6] Wenjing Bian, Zirui Wang, Kejie Li, Jiawang Bian, and Victor Adrian Prisacariu. Nope-nerf: Optimising neural radiance field with no pose prior. 2023. 2

[7] Hongrui Cai, Wanquan Feng, Xuetao Feng, Yan Wang, and Juyong Zhang. Neural surface reconstruction of dynamic scenes with monocular rgb-d camera. *Advances in Neural Information Processing Systems*, 2022. 3

[8] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, 1986. 4

[9] Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In *CVPR*, 2022. 2

[10] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *ECCV*, 2022. 2

[11] Zhiqin Chen, Thomas Funkhouser, Peter Hedman, and Andrea Tagliasacchi. Mobilenerf: Exploiting the polygon rasterization pipeline for efficient neural field rendering on mobile architectures. In *CVPR*, 2023. 1, 2

[12] Roger A Crawfis and Nelson Max. Texture splats for 3d scalar and vector field visualization. In *Proceedings Visualization'93*, 1993. 1

[13] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 3

[14] Nianchen Deng, Zhenyi He, Jiannan Ye, Budmonde Duinkharjav, Praneeth Chakravarthula, Xubo Yang, and Qi Sun. Fov-nerf: Foveated neural radiance fields for virtual reality. *IEEE Transactions on Visualization and Computer Graphics*, 2022. 1

[15] Arnab Dey, Yassine Ahmine, and Andrew I Comport. Mip-nerf rgb-d: Depth assisted fast neural radiance fields. *arXiv preprint arXiv:2205.09351*, 2022. 3

[16] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *CVPR*, 2022. 2, 3

[17] Yasutaka Furukawa, Carlos Hernández, et al. Multi-view stereo: A tutorial. *Foundations and Trends® in Computer Graphics and Vision*, 2015. 2

[18] 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

[19] Stephan J Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. Fastnerf: High-fidelity neural rendering at 200fps. In *ICCV*, 2021. 2

[20] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017. 4

[21] Xian-Feng Han, Hamid Laga, and Mohammed Bennamoun. Image-based 3d object reconstruction: State-of-the-art and trends in the deep learning era. *PAMI*, 2019. 2

[22] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 2

[23] Nikhil Keetha, Jay Karhade, Krishna Murthy Jatavallabhula, Gengshan Yang, Sebastian Scherer, Deva Ramanan, and Jonathon Luiten. Splatam: Splat, track & map 3d gaussians for dense rgb-d slam. *arXiv preprint arXiv:2312.02126*, 2023. 2

[24] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *TOG*, 2023. 1, 2, 3, 5, 6, 7

[25] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 2023. 1

[26] Mijeong Kim, Seonguk Seo, and Bohyung Han. Infonerf: Ray entropy minimization for few-shot neural volume rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 3

[27] Georgios Kopanas, Julien Philip, Thomas Leimkühler, and George Drettakis. Point-based neural rendering with per-view optimization. In *Computer Graphics Forum*, 2021. 4

[28] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *NIPS*, 2020. 2

[29] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015. 7

[30] Alican Mertan, Damien Jade Duff, and Gozde Unal. Single image depth estimation: An overview. *Digital Signal Processing*, 2022. 3

[31] Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and

Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)*, 2019. 2, 5, 6

[32] 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*, 2021. 1, 2, 5

[33] Thomas Müller, Fabrice Rousselle, Jan Novák, and Alexander Keller. Real-time neural radiance caching for path tracing. *arXiv preprint arXiv:2106.12372*, 2021. 2

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

[35] Thomas Neff, Pascal Stadlbauer, Mathias Parger, Andreas Kurz, Joerg H Mueller, Chakravarty R Alla Chaitanya, Anton Kaplanyan, and Markus Steinberger. Donerf: Towards real-time rendering of compact neural radiance fields using depth oracle networks. In *Computer Graphics Forum*, 2021. 3

[36] Michael Niemeyer, Jonathan T Barron, Ben Mildenhall, Mehdi SM Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 3

[37] Franco P Preparata and Michael I Shamos. *Computational geometry: an introduction*. Springer Science & Business Media, 2012. 5

[38] Malte Prinzler, Otmar Hilliges, and Justus Thies. Diner: Depth-aware image-based neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 3

[39] Michael Reed. *Methods of modern mathematical physics: Functional analysis*. Elsevier, 2012. 1

[40] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In *ICCV*, 2021. 2

[41] Barbara Roessle, Jonathan T Barron, Ben Mildenhall, Pratul P Srinivasan, and Matthias Nießner. Dense depth priors for neural radiance fields from sparse input views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 2, 3, 4

[42] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 2, 3

[43] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part V 12*, 2012. 7

[44] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *CVPR*, 2022. 2

[45] Ayush Tewari, Justus Thies, Ben Mildenhall, Pratul Srinivasan, Edgar Tretschk, Wang Yifan, Christoph Lassner, Vincent Sitzmann, Ricardo Martin-Brualla, Stephen Lombardi, et al. Advances in neural rendering. In *Computer Graphics Forum*, 2022. 2

[46] Carlo Tomasi and Takeo Kanade. Shape and motion from image streams under orthography: a factorization method. *International journal of computer vision*, 1992. 2

[47] Shimon Ullman. The interpretation of structure from motion. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 1979. 2

[48] 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

[49] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. Nerf–: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*, 2021. 2

[50] Yi Wei, Shaohui Liu, Yongming Rao, Wang Zhao, Jiwen Lu, and Jie Zhou. Nerfingmvs: Guided optimization of neural radiance fields for indoor multi-view stereo. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. 3

[51] Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. Synsin: End-to-end view synthesis from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 4

[52] Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. Neural fields in visual computing and beyond. In *Computer Graphics Forum*, 2022. 2

[53] Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. Point-nerf: Point-based neural radiance fields. In *CVPR*, 2022. 2, 3

[54] Chen Yang, Peihao Li, Zanwei Zhou, Shanxin Yuan, Bingbing Liu, Xiaokang Yang, Weichao Qiu, and Wei Shen. Nerfvs: Neural radiance fields for free view synthesis via geometry scaffolds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 3

[55] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *Advances in Neural Information Processing Systems*, 2021. 2

[56] Wang Yifan, Felice Serena, Shihao Wu, Cengiz Öztireli, and Olga Sorkine-Hornung. Differentiable surface splatting for point-based geometry processing. *ACM Transactions on Graphics (TOG)*, 2019. 4

[57] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. In *ICCV*, 2021. 2

[58] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020. 2

[59] Wenliang Zhao, Yongming Rao, Zuyan Liu, Benlin Liu, Jie Zhou, and Jiwen Lu. Unleashing text-to-image diffusion models for visual perception. *arXiv preprint arXiv:2303.02153*, 2023. 3

[60] Chao Zhou, Hong Zhang, Xiaoyong Shen, and Jiaya Jia. Unsupervised learning of stereo matching. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017. 3