This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Global and Hierarchical Geometry Consistency Priors for Few-shot NeRFs in Indoor Scenes

Xiaotian Sun[1] Qingshan Xu[2] Xinjie Yang[1] Yu Zang[1*] Cheng Wang[1]
[1] Fujian Key Laboratory of Sensing and Computing for Smart Cities, Xiamen University
[2] School of Computer Science and Engineering, Nanyang Technological University

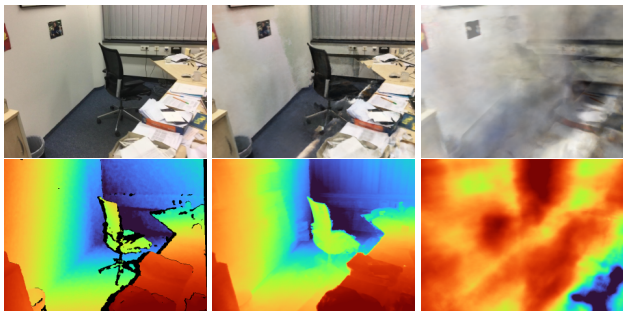
Abstract

It is challenging for Neural Radiance Fields (NeRFs) in the few-shot setting to reconstruct high-quality novel views and depth maps in 360° outward-facing indoor scenes. The captured sparse views for these scenes usually contain large viewpoint variations. This greatly reduces the potential consistency between views, leading NeRFs to degrade a lot in these scenarios. Existing methods usually leverage pretrained depth prediction models to improve NeRFs. However, these methods cannot guarantee geometry consistency due to the inherent geometry ambiguity in the pretrained models, thus limiting NeRFs' performance. In this work, we present P^{2}NeRF to capture global and hierarchical geometry consistency priors from pretrained models, thus facilitating few-shot NeRFs in 360° outward-facing indoor scenes. On the one hand, we propose a matching-based geometry warm-up strategy to provide global geometry consistency priors for NeRFs. This effectively avoids the overfitting of early training with sparse inputs. On the other hand, we propose a group depth ranking loss and ray weight mask regularization based on the monocular depth estimation model. This provides hierarchical geometry consistency priors for NeRFs. As a result, our approach can fully leverage the geometry consistency priors from pretrained models and help few-shot NeRFs achieve state-of-the-art performance on two challenging indoor datasets. Our code is released at https://github.com/XT5un/P2NeRF.

1. Introduction

As an advanced implicit scene representation, Neural Radiance Fields (NeRFs) are widely popular for many tasks, such as novel view synthesis [1, 2, 6, 17–19, 30], 3D reconstruction [12, 21, 34, 37, 39], and 3D generation [8, 27, 40]. It models continuous color and geometry fields of a 3D scene through a neural network to realize a compact 3D scene representation. In general, NeRF-like methods can achieve photo-realistic rendering results with a sufficient amount of training views. However, for sparse views captured in 360° outward-facing indoor scenes, it usually contain large viewpoint variations. This greatly reduces the potential consistency between views, leading the vanilla NeRFs to degrade a lot or even fail. Therefore, it is challenge for few-shot NeRFs to reconstruct high-quality novel views and depth maps in these scenarios.

Recently, geometric constraints have been demonstrated to be one key factor to improve few-shot NeRFs [10, 13, 20, 24]. Some methods [20, 28] impose regularization constraints by mining the geometric properties of the training



(a) Comparison with methods using general monocular depth

(b) Comparison with methods using absolute depth from other domains

Figure 1. Rendering results in 360° outward-facing indoor scenes. In challenging indoor scenes, using general monocular depth cannot meet the global geometry of the scene. And methods using in-domain geometric clues have poor generalization on other scenes.

*Corresponding author.

views, such as local depth smoothing, cross view visibility, *etc*. DS-NeRF [10] uses the sparse depth from Structure-from-Motion (SfM) to supervise the rendering depth directly. MonoSDF [39] aligns monocular depth and rendering depth by least squares. SparseNeRF [13] distills ranking relations from monocular depth in a local window. But these methods are usually effective in scenes with slight changes in viewpoint (Fig. 1a). On the other hand, the approaches with in-domain geometric priors are more effective in indoor environments with large viewpoint variations. DDP [24] trains a depth completion network to regress the dense absolute depth based on the sparse depth from SfM. SCADE [33] considers the ambiguity of depth estimation from a single view, and therefore utilizes multiple absolute depth predictions from a single view to constrain the NeRFs. However, the in-domain prior severely limits the generalization of such methods in out-of-domain scenes (Fig. 1b).

In this work, we propose P^{2}NeRF, to capture global and hierarchical geometry consistency priors from pretrained models for facilitating the few-shot NeRFs. First, we propose a matching-based geometry warm-up strategy to introduce a global geometry consistency prior. Specifically, since the sparse views usually make the 3D point optimization of NeRFs only be constrained by one or two images, NeRFs cannot learn reliable global geometry in early training, further interfering the subsequent optimization. Our matching-based geometry warm-up strategy extracts the sparse correspondences from different pairs of views using deep matching modules. Then, we reconstruct a coarse point cloud and compute the depth of each point. Although the accuracy of this point cloud is poor, it describes the scene's global structure to some extent. Therefore, this strategy provides a global geometry consistency prior to warm up the implicit geometry of NeRFs, preventing the radiance fields falling into geometrical disasters.

Second, based on monocular depth predictions, we propose two hierarchical geometry consistency priors to enhance the reconstruction of details. Because of the reduced consistency between views, NeRFs tends to concentrate the volume density in regions close to the camera, which results in serious ambiguity on the cross-view geometry [20, 36]. Our hierarchical geometry consistency priors leverage the relative ranking clues from monocular depth estimation to alleviate the above ambiguity. On the one hand, We propose a depth group ranking loss for rendering depth. This anchors the depths at different levels to each other, and enables the NeRFs to learn a reasonable geometric layout. On the other hand, we present a ray weight mask regularization that pushes the surface depth towards the right direction by adjusting the ray weights. Hierarchical geometry consistency priors force the volume density of NeRFs to be distributed at reasonable locations, reducing the geometry ambiguity. Therefore, it greatly improves the geometric reconstruction of details.

Our main contributions are summarized as follows:

- A matching-based global geometry consistency prior for warming up the NeRF's geometry, which prevents early overfitting during training.
- Two monocular depth-based hierarchical geometry consistency priors, including a group depth ranking loss and ray weight mask regularization to constrain the rendering depth and ray weights, respectively. They clamp the surface depth to a reasonable location, which enhances the detail of NeRFs.
- Extensive experiments on two challenging indoor datasets demonstrate that P^{2}NeRF achieves state-of-the-art image and depth rendering performance with sparse inputs.

2. Related Work

Neural Radiance Field for Rendering. Compared with traditional representations such as point cloud, voxels, and meshes, NeRFs generally use a multi-layer perceptron (MLP) to parameterize the scene. In NeRFs [18], spatial coordinates and viewing directions are mapped to volume densities and colors, and render pixels by a volume rendering algorithm. Mip-NeRF [1] designs a cone ray passing through the optical center of the camera and the pixel plane, and a Gaussian model is used to fit the conical frustums where the sampling points on the ray are located, which further enhances the network representation. The voxel-based neural fields [17, 19, 30], which combine explicit and implicit representation, dramatically speed up the training and inference. While these approaches can achieve excellent realism rendering, hundreds of training images are often required for a simple scene. With only a few training views, it is difficult for NeRFs to obtain similar rendering quality.

Sparse View Neural Radiance Fields. Some approaches [15, 20, 26, 28, 36] design a series of regularization losses for the training data itself to achieve stable training of the NeRFs in the sparse view case from the point of view of geometry, sampling space, position encoding frequency, and so on. PixelNeRF [38] and MVSNeRF [5] trains an image feature encoder that maps training view features to the camera coordinate system, achieving generalization across scenes. A more general and efficient approach is introducing additional information to supervise the training of NeRFs, especially geometric supervision. DDP [24] and SCADE [33] specifically train a depth prediction network to generate a depth prior with absolute scale. DS-NeRF [10] uses the sparse depth generated by COLMAP [25] to supervise the rendering depth. DietNeRF [14] uses the image encoder of CLIP [22] to extract the semantic features of the rendered views, and the cross-view semantic consistency are calculated by the semantic similarity loss between the training poses and randomized viewpoints. StructNeRF

[7] leverages the sparse depth supervision from SfM and the planar information from superpixel segmentation to enhance the training of NeRF. SparseNeRF [13] proposes a pairwise level local depth ranking loss which distills the relative positional relationships provided by monocular depth estimation to help the NeRFs.

Supervision of geometry has been demonstrated to contribute significantly to the success of few-shot NeRFs. Our P^{2}NeRF introduces global and hierarchical geometry consistency priors from pretrained models to constrain NeRFs, ultimately achieving high-performance yet universal few-shot NeRFs in complex indoor environments.

3. Method

3.1. Preliminaries and Motivation

Neural Radiance Fields. NeRF-like methods follow a general workflow to learn a scene representation: 1) capture a series of posed images covering the whole scene, 2) convert the rays passing through the camera's optical centers and pixel planes into some sample points, 3) query the volume densities and colors of sample points through a neural network, 4) and finally use ray marching algorithm to compose the pixels by weighting all points on the rays. We define the coordinate of the ray origin as $\mathbf{o}$, and the direction vector of the ray marching as $\mathbf{d}$. Given N samples along a ray, the i-th point $\mathbf{p}_i$ at depth t_i can be expressed as: $\mathbf{p}_i = \mathbf{o} + t_i\mathbf{d}$. The query network f with learnable parameters θ_0 and θ_1 maps $\mathbf{p}_i$ and $\mathbf{d}$ to volume density σ_i, hidden features $\mathbf{F}_i$ and color $\mathbf{c}_i$: $f_{\theta_0}(\mathbf{p}_i) = (\sigma_i, \mathbf{F}_i)$ and $f_{\theta_1}(\mathbf{F}_i, \mathbf{d}) = \mathbf{c}_i$. The volume rendering algorithm can be expressed as follows:

$$\hat{\mathbf{C}} = \sum_{i=1}^{N} w_i \mathbf{c}_i \quad \text{and} \quad \hat{D} = \sum_{i=1}^{N} w_i t_i, \tag{1}$$

$$\begin{aligned} &\text{with} \quad w_i = T_i(1 - \exp(-\sigma_i\delta_i)), \\ &T_i = \exp(-\sum_{j=1}^{i}\sigma_j\delta_j), \quad \delta_i = \|\mathbf{p}_{i+1} - \mathbf{p}_i\|_2 \end{aligned} \tag{2}$$

where w_i indicates the weight distribution of each point on the ray, and the rendering color $\hat{\mathbf{C}}$ is composed by weighting the color at each point, and the rendering depth $\hat{D}$ can be obtained in the same way. To optimize the network f, the photometric loss is computed between the rendering color $\hat{\mathbf{C}}$ and the ground truth color $\mathbf{C}$ as: $\mathcal{L}_{\text{color}} = \|\hat{\mathbf{C}} - \mathbf{C}\|_2^2$.

When the training views are dense enough, the sampling points can be jointly optimized by multiple viewpoint images. However, sparse inputs break the potential consistency between views and lead to the degradation of NeRFs.

Motivation. Rethinking Eq. (1), we see that the rendering color is a composite of the color $\mathbf{c}_i$ and weight w_i of each point on the ray. When NeRFs are supervised only by the photometric loss in the sparse view setting, it will be difficult to learn both $\mathbf{c}_i$ and w_i well at the same time. The $\mathbf{c}_i$ is the network prediction dependent on both position and direction, making it hard to enforce constraints for $\mathbf{c}_i$ directly to improve few-shot NeRFs. As for the w_i, it only depends on the volume density, *i.e.*, the geometry representation of NeRFs. It will be more feasible to introduce constraints for w_i to improve few-shot NeRFs. Considering that the rendering depth only relies on the w_i, our motivation is, improving the geometry of few-shot NeRFs by constraining the rendering depth, so as to further improve the rendering color. Based on this motivation, we take advantage of prior knowledge from pretrained models and convert them into the geometry supervision of few-shot NeRFs. At first, to avoid the geometry collapse of early training, we introduce a matching-based global geometry consistency prior to warm up the implicit geometry (Sec. 3.2). Then, we propose to extract hierarchical geometry consistency priors from the monocular depth estimation to enhance the structural details for few-shot NeRFs (Sec. 3.3). Both of these together make up our P^{2}NeRF.

(a) GT Point Cloud (b) Triangulation Point Cloud

Figure 2. Visualization of GT point cloud and triangulation point cloud. The depths from triangulation are very inaccurate, with only 46% of the points having an error of less than 0.1 meter.

3.2. Global Geometry Consistency Prior

It is natural to leverage explicit geometric data, *e.g.*, point cloud, depth map, or voxels of the scene to supervise the geometric representation of NeRFs globally. In this way, the early geometry collapse of few-shot NeRFs would disappear. Unfortunately, these data are not always available for every scenario. On the other hand, SfM is usually used to compute image poses for NeRFs, which also outputs sparse point clouds as a by-product. However, SfM also fails to reconstruct enough 3D points stably with only sparse images. Therefore, we propose to introduce global geometry consistency by triangulating two view correspondences using deep matching modules. Although the point cloud obtained by triangulation is of low accuracy, the number of points is sufficiently high. This can provide global geometry priors sufficiently to overcome early geometry collapse of NeRFs.

We first compute the frustums of the training poses and find the images who might intersect to build image pairs.

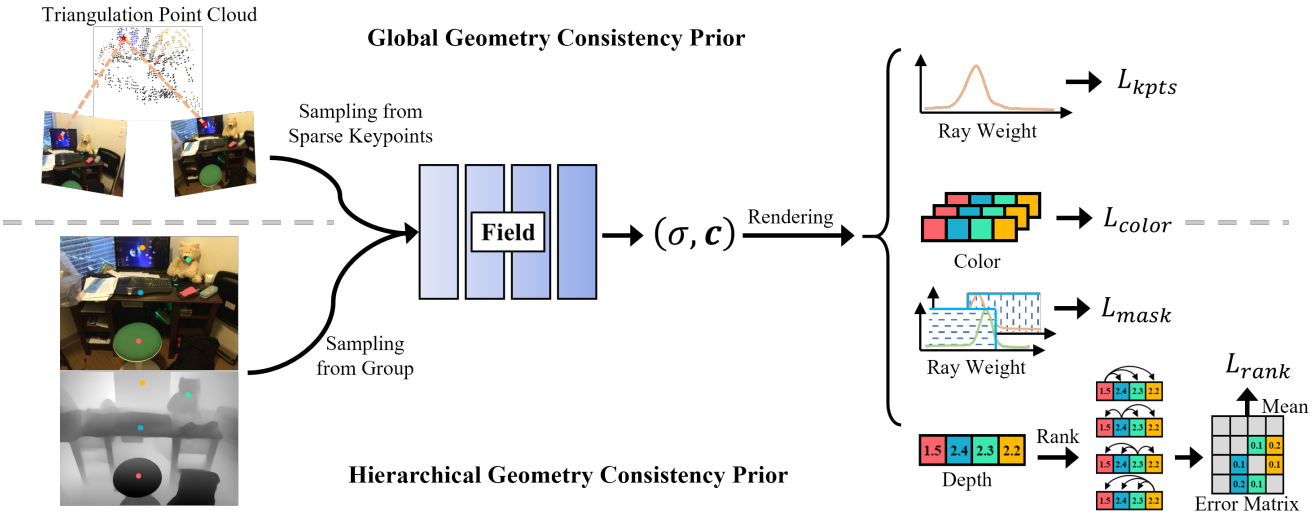


Figure 3. The pipeline of P^{2}NeRF. Top: using image matching to obtain sparse and coarse geometry prior computation L_{kpts} to warm up the model. Bottom: sampling a set of points based on monocular depth, computes their group depth ranking loss L_{rank} and applies mask regularization L_{mask} to the ray weights. The color loss L_{color} is computed throughout the training process.

Then LoFTR [31] is used to match them to generate sparse correspondences. Since the image matching is not exactly correct, we filter the extremely wrong keypoints using the epipolar constraint. Next, we use the least squares method to compute the point in 3D space that has the minimum distance from the rays where the keypoint pair is located. Finally, this 3D point is projected onto the ray, and the depth from the projected point to the camera is our keypoint depth prior, denoted D_k. Fig. 2 visualizes the point cloud from sensors and triangulation for a scene in the ScanNet [9] dataset, and it shows that the point cloud obtained using only triangulation is extremely coarse, but it still describes the approximate shape of the scene.

In the early stages of training, these coarse depths are used to warm up the implicit geometry. Because of the limited number of keypoints, we sample patches of size P centered on keypoints. For each ray passing through the patch, we compute its weight distribution using Eq. (2). Although D_k has a bias, it should be near the true depth. So, we set a window of radius ϕ on the sampling ray centered on D_k. For points located inside the window, we will maximize the sum of their weights, and for points outside the window, we will minimize the weight of each of them. This loss will force the implicit geometry of the radiance field towards the coarse scene geometry. This loss can be computed by:

$$\mathcal{L}_{\text{kpts}} = \begin{cases} l(1 - \sum_{i}^{N} w_i), & \text{if } |t_i - D_k| \leq \phi \\ l \sum_{i}^{N} w_i, & \text{if } |t_i - D_k| > \phi \end{cases} \tag{3}$$

$$\text{with} \quad l = [s < S] \times [s \bmod T = 0], \tag{4}$$

where $[\cdot]$ is Iverson bracket, s is the current iteration, and we only optimize $\mathcal{L}_{\text{kpts}}$ at an interval of T iterations in the first S iterations at the beginning.

3.3. Hierarchical Geometry Consistency Prior

Monocular depth estimation models [3, 11, 23, 32] that are fully trained on large-scale datasets have good generalization capability. Although they cannot provide absolute depth, they can characterize the hierarchical relationship of different objects in space. Based on this, we design two losses to leverage this hierarchical geometry consistency prior constrain the geometry of NeRFs. The first is a group depth ranking loss, which constrains the scene surface to be in a reasonable position by anchoring the rendering depths from different groups to each other. The second is the ray weight mask regularization loss, which pushes the rendering depth to the correct direction by adjusting the distribution of ray weights in different intervals. In this work, we use the monocular depth prior from the DPT [23].

Group Depth Ranking Loss. We categorize the pixels in the monocular depth map into M depth groups according to percentile, where pixels with smaller group order numbers are closer to the camera. We randomly sample rays in each depth group and combine them into batch, and then get the rendering depths using Eq. (1). Next, the rendering depths are compared one by one to generate a error matrix. The rays with incorrectly depth ranking will be penalized. Fig. 3 explains the process of calculating the group depth ranking loss, which is shown in Eq. (5):

$$\mathcal{L}_{\text{rank}} = \frac{1}{M^2} \sum_{i=1}^{M} \sum_{j=1}^{M} \max(\text{sign}(j - i)(\hat{D}_i - \hat{D}_j), 0), \tag{5}$$

where sign$(\cdot)$ is a sign function, and $\hat{D}_i$ is the rendering depth of the ray from the i-th group. When $i < j$, if $\hat{D}_i$ is also less than $\hat{D}_j$, it means that the ranking is correct in these two places, and the error matrix has the value of 0 at (i, j). However, if $\hat{D}_i$ is greater than $\hat{D}_j$, it means that the ranking is wrong between them, and the error matrix produces a loss value at (i, j). In the same way, we can introduce that when $i > j$, a loss occurs only if $\hat{D}_i$ is less than $\hat{D}_j$. In addition, for the case of $i = j$, it is obvious that there will be no loss. Our group depth ranking loss can achieve more effective geometric constraints by calculating a error matrix that anchors depths at different positions to each other.

Ray Weight Mask Regularization. Assume that there are two pixels a and b, and the monocular depth of a is smaller than that of b. Let their ground truth depths be D^a and D^b, and their rendering depths be $\hat{D}^a$ and $\hat{D}^b$.

When a ranking error occurs in their rendering depth, the cases between the rendering depth and the ground truth depth can be summarized in three cases: 1) $\hat{D}^b \gg D^b$, 2) $\hat{D}^a \ll D^a$, 3) other cases. We find that, in either case we can at least determine the relationship between the rendering depth and the ground truth depth for a or b. For the first case, $\hat{D}^a$ must be greater than D^a. For the second case, $\hat{D}^b$ must be less than D^b. For the third case, there must be a margin Δ such that $\hat{D}^a + \Delta \geq D^a$ and $\hat{D}^b - \Delta \leq D^b$. Also, since we can infer the positional relationship between the ground true depth and the rendering depth, we can adjust the ray weight distribution directly to constrain the implicit geometry. Therefore, we define a new hierarchical geometric constrain, ray weight mask regularization loss, as:

$$\mathcal{L}_{\text{mask}} = 1 - \sum_{i=1}^{N} m_i w_i + \sum_{i=1}^{N} (1 - m_i) w_i, \tag{6}$$

$$m_i^a = \begin{cases} 1, & \text{if} \quad t_i^a \leq \hat{D}^a + \Delta \\ 0, & \text{if} \quad t_i^a > \hat{D}^a + \Delta \end{cases} \tag{7}$$

$$m_i^b = \begin{cases} 0, & \text{if} \quad t_i^b < \hat{D}^b - \Delta \\ 1, & \text{if} \quad t_i^b \geq \hat{D}^b - \Delta \end{cases} \tag{8}$$

where m is a mask for splitting unreasonable regions on the ray, whose value is set to 0 for regions where depth must not exist and 1 for regions where depth might exist.

Fig. 4 explains the mechanism of $\mathcal{L}_{\text{mask}}$. For a ray passing through a, any weight that is located behind $\hat{D}^a + \Delta$ is a negative contributor to the depth rendering, so we would like to reduce the weight in this area. The term $\sum_{i=1}^{N}(1 - m_i)w_i$ in the loss function achieves this effect. Also, the item $1 - \sum_{i=1}^{N} m_i w_i$ ensures that the weight distribution is concentrated in the area in front of $\hat{D}^a + \Delta$. $\mathcal{L}_{\text{mask}}$ works similarly for pixel b, except that the mask takes the opposite value.

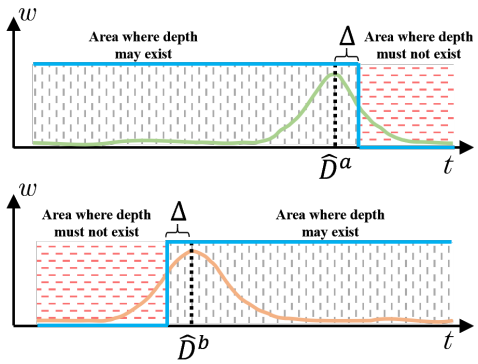


Figure 4. Ray weight mask regularization works on a pair of rays which are incorrectly ranked. L_{mask} moves the weight distribution in the right direction by minimizing the weights that appear in incorrect regions.

	Case	$\mathcal{L}^a_{\text{mask}}$	$\mathcal{L}^b_{\text{mask}}$
1)	$D^a < D^b < \hat{D}^b - \Delta < \hat{D}^a$	✓	✗
2)	$\hat{D}^b < \hat{D}^a + \Delta < D^a < D^b$	✗	✓
3)	$\hat{D}^a + \Delta \geq D^a$ and $\hat{D}^b - \Delta \leq D^b$	✓	✓

Table 1. Effectiveness of $\mathcal{L}_{\text{mask}}$ in different cases.

We list the various cases where $\mathcal{L}_{\text{mask}}$ is valid or not in Tab. 1. Although it suffers from negative optimization when $\hat{D}^a \ll D^a$ or $\hat{D}^b \gg D^b$, as long as margin Δ is not overly strict, these extreme cases are rare. Meanwhile, if the extreme case occurs, the group depth ranking also produces a large loss value, further reducing the impact of this case.

Combining all our designs, the total loss of P^{2}NeRF is:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{color}} + \lambda_0 \mathcal{L}_{\text{kpts}} + \lambda_1 \mathcal{L}_{\text{rank}} + \lambda_2 \mathcal{L}_{\text{mask}}. \tag{9}$$

4. Experiments

4.1. Experimental Setting

Datasets and Evaluation Metrics. We evaluate the performance of P^{2}NeRF with sparse views on indoor datasets ScanNet [9] and Replica [29]. For the experiments on ScanNet, we followe the configuration of the DDP [24], sampling 18 to 20 images in three rooms for training and using 8 images for testing. Replica is a synthetic dataset, we use 8 scenes rendered by NICE-SLAM [42], and sample 20 images from 2000 frames of each scene at intervals of 100 for training and testing respectively. Meanwhile, in order to ensure the difference between the training and testing sets, the training set begin from 0th frame, and the testing set begin from 50th frame. We adopt four metrics to evaluate the test results, where PSNR, SSIM [35], and LPIPS [41] are used to evaluate the novel view synthesis, and depth RMSE is used to evaluate the geometric reconstruction capability.

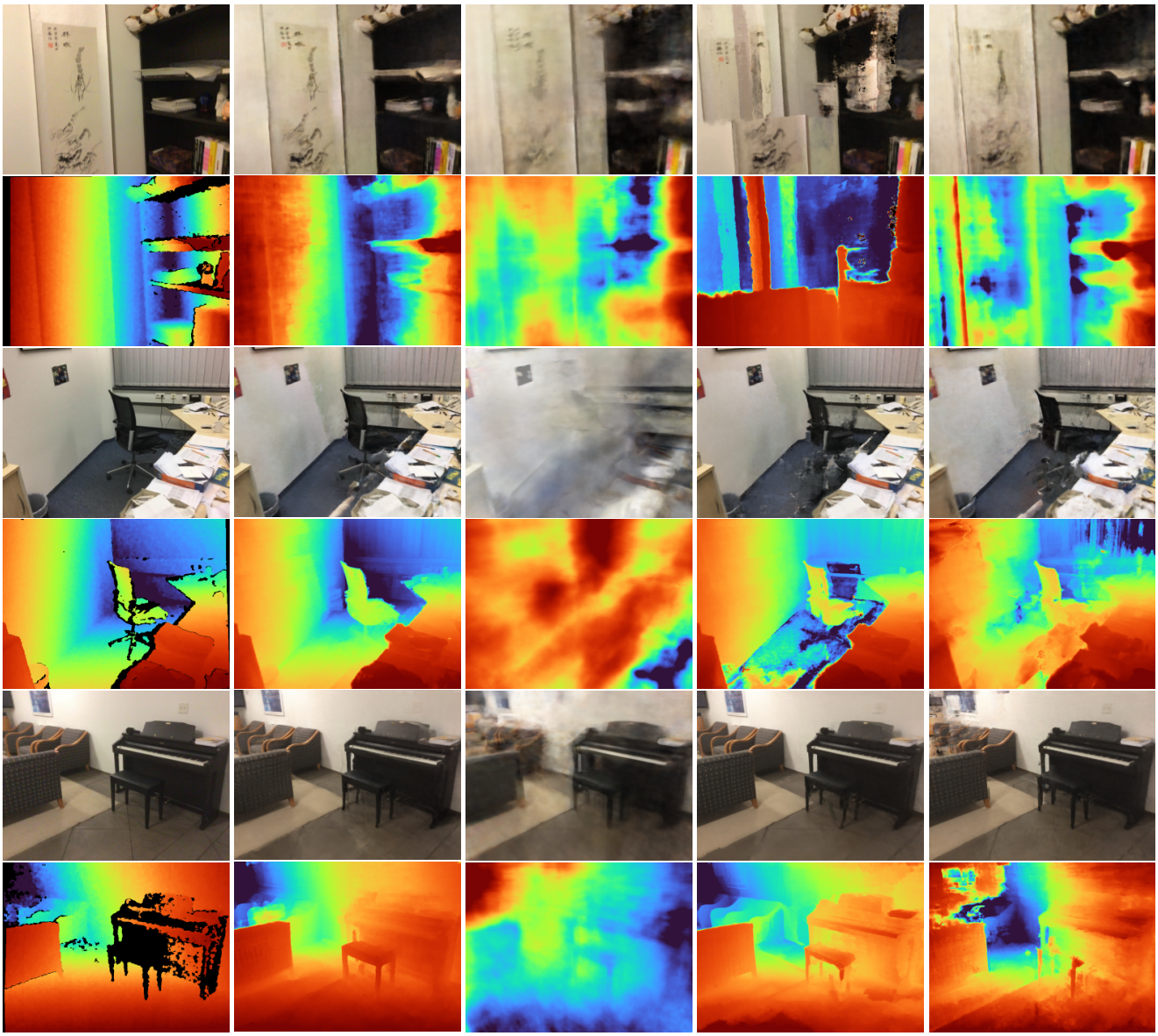


Figure 5. Comparison of rendering images and rendering depth visualizations for three scenes in the ScanNet dataset.

Implementation Details. We implement our P^{2}NeRF using Jax-based [4] Mip-NeRF [1] as a backbone network. We use the Adam optimizer [16] with a learning rate decaying exponentially from 2×10^{-3} to 2×10^{-5}. The batch size is 4096. Each scene is trained 100k iterations on a NVIDIA RTX3090 GPU, taking about 6.5 hours. For our geometric warm-up, T and S are set to 3 and 1536, respectively. The patch size is set to 16. And, the radius ϕ is $0.1\times(f-n)$, f being the depth from the far plane to the camera and n being the depth from the near plane to the camera. For the group depth ranking loss, M is set to 32. For ray weight mask regularization, we use a margin of 0.1m to ensure that training can proceed properly. In the total loss, the coefficients λ_0, λ_1, and λ_2 are 0.1, 0.02, and 0.002, respectively.

Comparing Methods. We compare with four recent state-of-the-art few-shot NeRFs methods, including DS-NeRF [10], FreeNeRF [36], SparseNeRF [13], and SCADE [33]. DS-NeRF uses the sparse depth obtained from COLMAP [25] reconstruction as a prior. FreeNeRF does not use any external information. SparseNeRF also uses a monocular depth prior, but it computes pairwise ranking losses at the local level. SCADE specifically trains an ambiguity-aware model to regress absolute depth. For FreeNeRF, we performed experiments with 50% and 80%-schedule frequency regularization, and report the better 50%-schedule results. In the official implementation of SCADE, the rendering im-

Method	PSNR↑	SSIM↑	LPIPS↓	RMSE↓
DS-NeRF [10]	20.85	0.713	0.344	0.447
InfoNeRF [15]	16.97	0.615	0.361	1.212
RegNeRF [20]	18.17	0.621	0.310	0.600
FreeNeRF [36]	19.37	0.652	0.302	0.516
SCADE [33]	20.75	0.703	0.306	0.481
SparseNeRF [13]	17.54	0.624	0.429	0.991
Ours	21.03	0.719	0.209	0.213

Table 2. Quantitative results on ScanNet. DS-NeRF results are from previous literature [24]. The PSNR, SSIM, and LPIPS reported in the SCADE paper are 21.54, 0.732, and 0.292, respectively. As mentioned above, in order to ensure fair comparisons, we only report the uncorrected rendered image metrics. Red, orange and yellow: the best, scenond-best and third-best.

Method	PSNR↑	SSIM↑	LPIPS↓	RMSE↓
DS-NeRF [10]	27.35	0.866	0.167	0.411
FreeNeRF [36]	28.32	0.887	0.117	0.299
SCADE [33]	22.49	0.757	0.294	1.822
SparseNeRF [13]	29.86	0.887	0.163	0.151
Ours	31.15	0.909	0.061	0.185

Table 3. Quantitative results on Replica.

Model	$\mathcal{L}_{\text{kpts}}$	$\mathcal{L}_{\text{rank}}$	$\mathcal{L}_{\text{mask}}$	PSNR ↑	SSIM ↑	LPIPS ↓	RMSE ↓
(1)	Baseline			18.71	0.637	0.302	0.722
(2)	✓			19.63	0.697	0.253	0.376
(3)		✓	✓	15.99	0.537	0.408	0.976
(4)	✓		✓	20.27	0.702	0.216	0.238
(5)	✓	✓		20.64	0.706	0.208	0.204
(6)	✓	✓	✓	21.03	0.719	0.209	0.213

Table 4. Ablation study results on ScanNet.

ages are post-processed with ground truth. For fair comparisons, we just report the results without post-processing. In addition, we also report quantitative results for InfoNeRF [15] and RegNeRF [20] on ScanNet.

4.2. Comparisons

ScanNet Dataset. Tab. 2 shows the quantitative results of different methods on the ScanNet [9] dataset. We achieve the best rendering quality and the most accurate geometry estimation on this dataset. Fig. 5 shows the visualization results. SparseNeRF suffers from significant overfitting, and only using a monocular depth prior for ranking is not an effective way to reconstruct the geometry. FreeNeRF's rendering depth without the assistance of additional information is clearly affected by the local image texture. Although SCADE's quantitative results are close to ours, its qualitative results contain obvious artifacts.

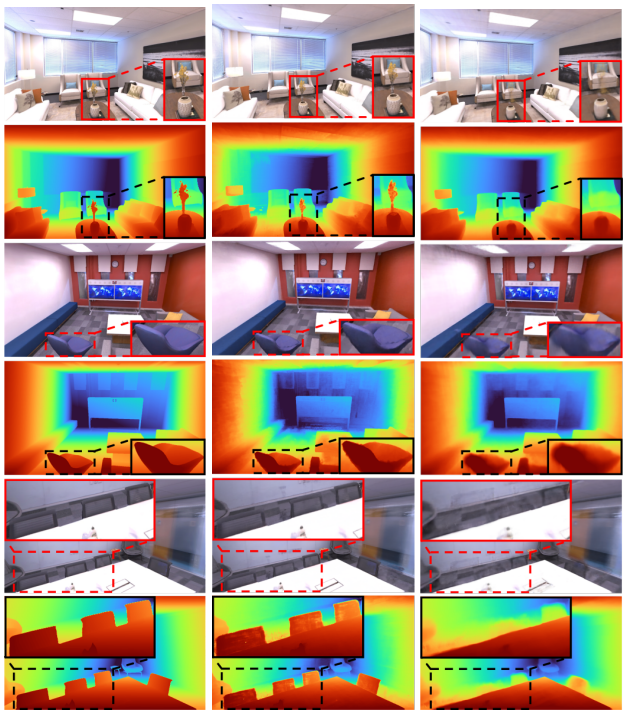


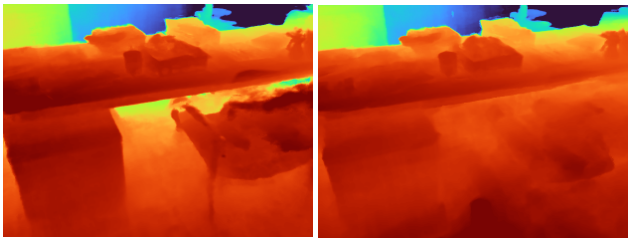
Figure 6. Visualization of our P^{2}NeRF and SparseNeRF on the Replica dataset. The boxes show detail areas. P^{2}NeRF can reconstruct richer texture and sharper depth, while SparseNeRF cannot.

Replica Dataset. We show the quantitative results on the Replica dataset in Tab. 3. Our approach achieves the best novel view synthesis outcomes, and the second best geometric accuracy after SparseNeRF. By analyzing the dataset, we find that the scenes in Replica have more smooth regions, *e.g.*, walls and floors. The local smoothing technique used in SparseNeRF greatly improves its performance in these regions. However, we can still retain more details than SparseNeRF through hierarchical geometric constraints. Fig. 6 visualizes the difference in detail between our and SparseNeRF's rendering images and depths, and our approach reconstructs richer textures and geometry.

4.3. Ablation Study

In this section, we ablate our proposed global and hierarchical consistency constraints on ScanNet dataset. All hyperparameters are set as the previous section. The results are reported in Tab. 4.

Ablation of Geometric Warm-up. Experiment (2) in Tab. 4 studies the effect of geometric warm-up and shows a very significant improvement in the depth RMSE metric, demonstrating its effectiveness. Experiment (3) removes this technique and uses two hierarchical constraints directly, which results in heavy degradation. It is similar to the performance of SparseNeRF as we observed. If there is not a good initial geometry, the depth ranking loss will make the



(a) P^{2}NeRF (b) w/o group ranking

Figure 7. Rendering depth map with & without group depth ranking. In the experiment without group depth ranking, the position of the chair and its surroundings appeared to be mixed up.

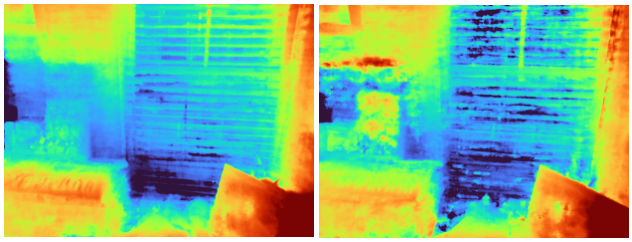
	PSNR ↑	SSIM ↑	LPIPS ↓	RMSE ↓
SparseNeRF	17.54	0.624	0.429	0.991
SparseNeRF+Warm-up	20.30	0.702	0.235	0.234
(2) (SfM)	19.14	0.672	0.282	0.409
(2) (matching)	19.63	0.697	0.253	0.376
(4)+Local depth ranking	20.43	0.691	0.226	0.230
(4)+Group depth ranking	21.03	0.719	0.209	0.213

Table 5. Comparison of SparseNeRF with & without geometric warm-up (top), different warm-up (middle) and depth ranking (bottom) strategies on ScanNet dataset. (2) and (4) means the Model (2) and (4) in Table 4.

geometry increasingly worse. In Tab. 5, we also report the results of applying geometric warm-up to the SparseNeRF as well as warm-up using the SfM point cloud. With our warm-up, SparseNeRF can successfully reconstruct ScanNet scenes. In addition, the RMSE in the experiments using the SfM point cloud are all higher than the experiments using the matching-based point cloud. These fully demonstrate the effectiveness and generalization of our global geometric consistency.

Ablation of Group Depth Ranking. Comparing experiment (2) and (5) in Tab. 4, it can be seen that there is an overall improvement, especially for geometry, after using the group depth ranking constraints. Indeed, we also observe structural ambiguity in the visualization comparison between experiment (4) and (6) (Fig. 7). In addition, we compare the local depth ranking used in SparseNeRF with our group depth ranking. The results exhibited in Tab. 5 show that group depth ranking produces stronger geometric constraints. All of these suggest that group depth ranking is very useful for helping NeRFs learn scene structures.

Ablation of Ray Weight Mask Regularization. In Tab. 4, experiment (4) adds $\mathcal{L}_{\text{mask}}$ to experiment (2), and the performance improvement indicates the effectiveness of this constraint. In particular, comparing the results of experiments (5) and (6), there is a certain decrease in RMSE when ray weight mask regularization is used. It is consistent with our analysis in Tab. 1, that there is an optimization blind spot for $\mathcal{L}_{\text{mask}}$. However, we should still not ignore its effect. In the visualization, we find the depth map with the ray weight mask regularization has less noise (Fig. 8). Meanwhile, the best results for novel view synthesis are obtained by the combination of ray weight mask regularization and group depth ranking.



(a) P^{2}NeRF (b) w/o weight mask

Figure 8. Rendering depth map with & without ray weight mask regularization. With ray weight mask regularization, there is less noise and smoother depth on windows and walls.

4.4. Limitations

Our P^{2}NeRF distills global and hierarchical priors from image matching and monocular depth estimation, respectively, which makes the NeRFs more stable and robust in indoor environments with large viewpoint variations. But both of these also introduce their own biases into the NeRFs. First, since the point cloud we used is very coarse, excessive depth errors may lead to a local poor geometric distribution during warm-up. Second, because monocular depth estimation models are affected by texture, illumination, *etc*., and generate erroneous surfaces. The same errors may happen in the rendering depth. Solving these two problems could further release the potential of the few-shot NeRFs, which will be our future work.

5. Conclusion

We present P^{2}NeRF, a method for achieving few-shot NeRFs using global and hierarchical geometry consistency priors extracted from pretrained models. We use matching-based global priors to warm up the implicit geometry, enabling NeRFs to reliably construct the approximate shape of the scene during the early stages of training. To further enhance the representation of the geometric structure, we introduce hierarchical priors from the monocular depth estimation model. By anchoring the rendering depths of the different hierarchies to each other, and adjustments for unreasonable weight distributions, we render high-quality images and depths. Overall, our P^{2}NeRF is a cheap yet efficient NeRFs solution for challenging indoor scenes with sparse views.

Acknowledgements This work was supported in part by the Fundamental Research Funds for the Central Universities (No. 20720230033) and PDL (2022-PDL-12).

References

[1] 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*, pages 5855–5864, 2021. 1, 2, 6

[2] 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. 1

[3] 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. 4

[4] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, et al. Jax: composable transformations of python+ numpy programs. 2018. 6

[5] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14124–14133, 2021. 2

[6] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *European Conference on Computer Vision*, pages 333–350. Springer, 2022. 1

[7] Zheng Chen, Chen Wang, Yuan-Chen Guo, and Song-Hai Zhang. Structnerf: Neural radiance fields for indoor scenes with structural hints. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 3

[8] Gene Chou, Yuval Bahat, and Felix Heide. Diffusion-sdf: Conditional generative modeling of signed distance functions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2262–2272, 2023. 1

[9] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 4, 5, 7

[10] 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*, pages 12882–12891, 2022. 1, 2, 6, 7

[11] Ainaz Eftekhar, Alexander Sax, Jitendra Malik, and Amir Zamir. Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3d scans. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10786–10796, 2021. 4

[12] Qiancheng Fu, Qingshan Xu, Yew Soon Ong, and Wenbing Tao. Geo-neus: Geometry-consistent neural implicit surfaces learning for multi-view reconstruction. *Advances in Neural Information Processing Systems*, 35:3403–3416, 2022. 1

[13] Guangcong, Zhaoxi Chen, Chen Change Loy, and Ziwei Liu. Sparsenerf: Distilling depth ranking for few-shot novel view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. 1, 2, 3, 6, 7

[14] Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5885–5894, 2021. 2

[15] 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*, pages 12912–12921, 2022. 2, 7

[16] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. 6

[17] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *Advances in Neural Information Processing Systems*, 33:15651–15663, 2020. 1, 2

[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. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part I*, pages 405–421, 2020. 2

[19] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15, 2022. 1, 2

[20] 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*, pages 5480–5490, 2022. 1, 2, 7

[21] 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. 1

[22] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 2

[23] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE transactions on pattern analysis and machine intelligence*, 44(3):1623–1637, 2020. 4

[24] 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*, pages 12892–12901, 2022. 1, 2, 5, 7

[25] 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, 6

[26] Seunghyeon Seo, Donghoon Han, Yeonjin Chang, and Nojun Kwak. Mixnerf: Modeling a ray with mixture density for novel view synthesis from sparse inputs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20659–20668, 2023. 2

[27] J Ryan Shue, Eric Ryan Chan, Ryan Po, Zachary Ankner, Jiajun Wu, and Gordon Wetzstein. 3d neural field generation using triplane diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20875–20886, 2023. 1

[28] Nagabhushan Somraj and Rajiv Soundararajan. Vip-nerf: Visibility prior for sparse input neural radiance fields. In *ACM SIGGRAPH 2023 Conference Proceedings*, 2023. 1, 2

[29] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 5

[30] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5459–5469, 2022. 1, 2

[31] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8922–8931, 2021. 4

[32] Libo Sun, Jia-Wang Bian, Huangying Zhan, Wei Yin, Ian Reid, and Chunhua Shen. Sc-depthv3: Robust self-supervised monocular depth estimation for dynamic scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 4

[33] Mikaela Angelina Uy, Ricardo Martin-Brualla, Leonidas Guibas, and Ke Li. Scade: Nerfs from space carving with ambiguity-aware depth estimates. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16518–16527, 2023. 1, 2, 6, 7

[34] 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. *Advances in neural information processing systems*, 2021. 1

[35] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 5

[36] Jiawei Yang, Marco Pavone, and Yue Wang. Freenerf: Improving few-shot neural rendering with free frequency regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8254–8263, 2023. 2, 6, 7

[37] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *Advances in Neural Information Processing Systems*, 34:4805–4815, 2021. 1

[38] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2021. 2

[39] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. *Advances in neural information processing systems*, 35:25018–25032, 2022. 1, 2

[40] Biao Zhang, Jiapeng Tang, Matthias Nießner, and Peter Wonka. 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. *ACM Trans. Graph.*, 42(4), 2023. 1

[41] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 5

[42] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R Oswald, and Marc Pollefeys. Nice-slam: Neural implicit scalable encoding for slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12786–12796, 2022. 5