StructNeRF: Neural Radiance Fields for Indoor Scenes With Structural Hints

Zheng Chen, Chen Wang, Yuan-Chen Guo, and Song-Hai Zhang, *Member, IEEE*

***Abstract*—Neural Radiance Fields (NeRF) achieve photo-realistic view synthesis with densely captured input images. However, the geometry of NeRF is extremely under-constrained given sparse views, resulting in significant degradation of novel view synthesis quality. Inspired by self-supervised depth estimation methods, we propose StructNeRF, a solution to novel view synthesis for indoor scenes with sparse inputs. StructNeRF leverages the structural hints naturally embedded in multi-view inputs to handle the unconstrained geometry issue in NeRF. Specifically, it tackles the texture and non-texture regions respectively: a patch-based multi-view consistent photometric loss is proposed to constrain the geometry of textured regions; for non-textured ones, we explicitly restrict them to be 3D consistent planes. Through the dense self-supervised depth constraints, our method improves both the geometry and the view synthesis performance of NeRF without any additional training on external data. Extensive experiments on several real-world datasets demonstrate that StructNeRF shows superior or comparable performance compared to state-of-the-art methods (e.g. NeRF, DSNeRF, RegNeRF, Dense Depth Priors, MonoSDF, etc.) for indoor scenes with sparse inputs both quantitatively and qualitatively.**

***Index Terms*—Neural radiance fields, neural rendering, novel view synthesis.**

I. Introduction

Novel view synthesis (NVS) for indoor scenes plays an important role in VR and AR applications, such as virtual navigation through buildings, tourist sites, and game environments. However, people often have to devote extensive efforts to collecting and processing large amounts of input data in order to produce satisfying results [12], [21], [25]. It remains to be a problem how to synthesize photo-realistic novel views given limited indoor images [7], [16], [22], [25], [36]. Recently, Neural Radiance Fields (NeRF) [21] emerges as a promising technique for NVS. NeRF uses a continuous multi-layer perceptron (MLP) to encode the radiance and density of a 3D scene and then synthesizes novel views through differentiable volumetric rendering. It achieves photo-realistic results even when representing some scenes with complicated geometry and appearance. Nevertheless, sparse indoor scene inputs bring several innate challenges to NeRF. First, reconstructing the geometry and appearance of objects or scenes becomes an ill-posed problem with insufficient inputs. Even though NeRF can well fit the training images at the pixel level, the geometry is indeed inaccurate and leads to unsatisfying renderings at test viewpoints [7]. The necessity of "inside-out" view capture for indoor scene images exaggerates this issue [12]. Compared with "outside-in" viewing scenarios for outdoor scenes or standalone objects, adjacent views would have less overlap with each other given the same number of images [25]. Second, indoor scenes contain many textureless regions such as walls, floors, tables, and ceilings, making it hard for NeRF to find enough cross-view 3D correspondences.

Several recent studies [7], [25], [34] leverage depth priors to improve the performance of NeRF in novel view synthesis. DSNeRF [7] adopts the sparse depth point cloud from COLMAP [26] to directly constrain the depth rendered by NeRF. However, the depth from Structure-from-Motion (SfM) is both sparse and noisy. Dense Depth Priors [25] further utilizes a depth completion network to predict dense depth maps, which are then used to guide the sampling and depth prediction of NeRF. However, the depth completion network introduces view inconsistency and generalization issues. To overcome these problems, we present StructNeRF, a technique that takes inspiration from recent self-supervised depth estimation methods [15], [37] and incorporates structural hints naturally contained in multi-view inputs, which turns into easy-to-adapt regularizations for NeRF geometry without any additional networks or data. StructNeRF considers the huge differences between textured and textureless regions and tackles them separately. Inspired by the insight of NeRF++ [39], we notice that the ability of NeRF to model the appearance of view-dependent effects leads to the ambiguity between its 3D shape and radiance (shape-radiance ambiguity). To reduce this ambiguity, we ensure that the same 3D region in different views is view-consistent by leveraging a patch-based multi-view consistent photometric loss based on depth warping. The resulting depth constraints are therefore dense and view-consistent. Patch-based photometric loss works well for textured regions, but it fails to discriminate non-textured regions that are common in indoor scenes, such as floor, walls, tables and ceiling. At the same time, we notice these regions are almost planes, so we further restrict them to be planar. To be more specific, we segment each input view into superpixels and group them as plane priors (Most superpixels are planes as shown in Fig. 4). Then the co-planar constraint [15] is applied to constrain the

Manuscript received 17 August 2022; revised 10 July 2023; accepted 3 August 2023. Date of publication 15 August 2023; date of current version 3 November 2023. This work was supported in part by the Natural Science Foundation of China under Project 62132012 and in part by the Tsinghua-Tencent Joint Laboratory for Internet Innovation Technology. Recommended for acceptance by M. Long. *(Corresponding author: Zheng Chen.)*

The authors are with the BNRist, Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China (e-mail: chenz20@mails.tsinghua.edu.cn; cw.chenwang@outlook.com; guoyc19@mails.tsinghua.edu.cn; shz@tsinghua.edu.cn).

This article has supplementary downloadable material available at https://doi.org/10.1109/TPAMI.2023.3305295, provided by the authors.

Digital Object Identifier 10.1109/TPAMI.2023.3305295

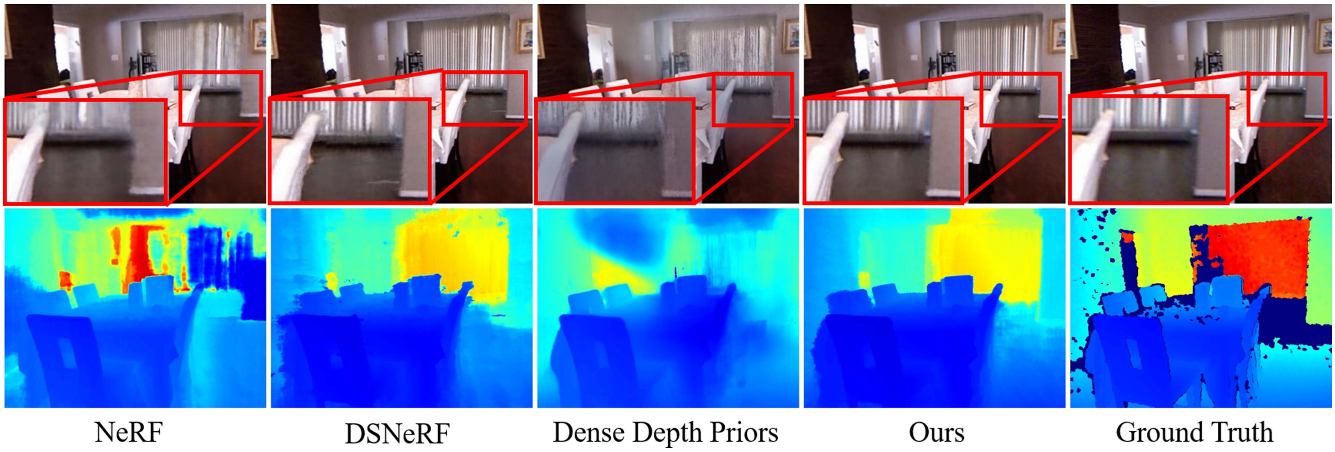

Fig. 1. Qualitative comparison of novel view synthesis and depth estimation given only a sparse set of indoor images. StructNeRF demonstrates superior performance over state-of-the-arts. We propose two structural hints: patch-based multi-view consistency at textured regions and planar consistency at non-textured regions, significantly improving the synthesis and geometry of radiance fields without any additional data or networks.

depth of those regions. Additionally, we apply warm-up training to reduce the negative impact of noisy sparse point clouds from COLMAP, which helps StructNeRF to better utilize the sparse depth information. We evaluate our method on three indoor datasets: ScanNet [6], NYUv2 [29], SUN3D [35], using only sparse inputs. Results show that our proposed structural hints and training strategy together enable StructNeRF outperform existing per-scene training methods (only use data of target scene without additional training). Compared to Dense Depth Priors [25] that utilizes external data to train their depth estimation networks (we call data-driven methods), our method still shows comparable performance on their pretrained datasets and surpass them on other ones. A demonstration of the comparisons can be found in Fig. 1.

The contributions of our paper can be summarized as the following:

1) By introducing patch-based multi-view consistent loss into NeRF, StructNeRF obtains dense and view-consistent depth constraints, without pretraining on external data.

2) StructNeRF re-projects points in textureless regions into the 3D space and enforce them to be planes with the plane consistency loss in NeRF. Therefore, the reconstructed planes are more flat and the rendering quality is also improved.

II. Related Work

In this section, we briefly review Neural Radiance Field with sparse inputs and self-supervised depth estimation.

A. Neural Radiance Fields With Sparse Inputs

Based on implicit neural representations, Neural Radiance Fields (NeRF) [21] encoded 3D scenes into a continuous multilayer perceptron (MLP) and achieved photorealistic novel view synthesis. A growing number of NeRF extensions then emerged, e.g., reconstructing without camera poses [19], [33], modelling non-rigid scenes [23], [24], unbounded scenes [40], handling reflections [11], [30] and super-resolution [31]. When the scene is observed by sparse views, NeRF would however estimate a wrong density distribution, which is specifically reflected as some artifacts in the rendering process, such as “floaters”. Here we give a detailed review of NeRF-based methods in both object-level and scene-level when the inputs are sparse.

Given sparse object-level views, several recent works [3], [4], [13], [14], [36] synthesized novel views using a pretraining with an optional per-scene optimization strategy. The pretrained network is however not suitable for indoor scenes due to the domain gap. Other methods impose regularizations on NeRF geometry, for example, RegNeRF [22] samples unobserved camera poses and regularizes patches rendered from those views with a depth smoothness loss and a trained normalizing flow model respectively. InfoNeRF [16] utilizes regularization based on information theory to improve view synthesis. However, both InfoNeRF [22] and RegNeRF [22] do not guarantee multi-view consistency. And all these object-level approaches never take into account the characteristics of the indoor scenes that we mentioned in Section I.

With regard to scene-level, recent studies like NerfingMVS [34], DSNeRF [7] and Dense Depth Priors [25] proposed to introduce depth priors to resolve the unconstrained geometry problem in NeRF from different aspects. Without additional network or training, DSNeRF [7] utilizes the sparse depth information from COLMAP [26] directly to constrain the depth rendered. NerfingMVS [34] instead trains a monocular depth estimation network to get scene-specific depth priors for guiding NeRF sampling. Similarly, Dense Depth Priors [25] leverages a pretrained depth completion network to predict dense depth maps for each view individually, which are then used to both supervise the rendered depth and guide NeRF sampling. However, there are two obvious problems in Dense Depth Priors. First, the depth completion network is not view-consistent because each view is processed individually. Second, it also suffers from generalization issues as it relies on labeled training data such as ScanNet [6].

Compared with previous methods, StructNeRF leverages patch-based multi-view consistent depth loss to obtain dense supervision for NeRF without any depth completion network

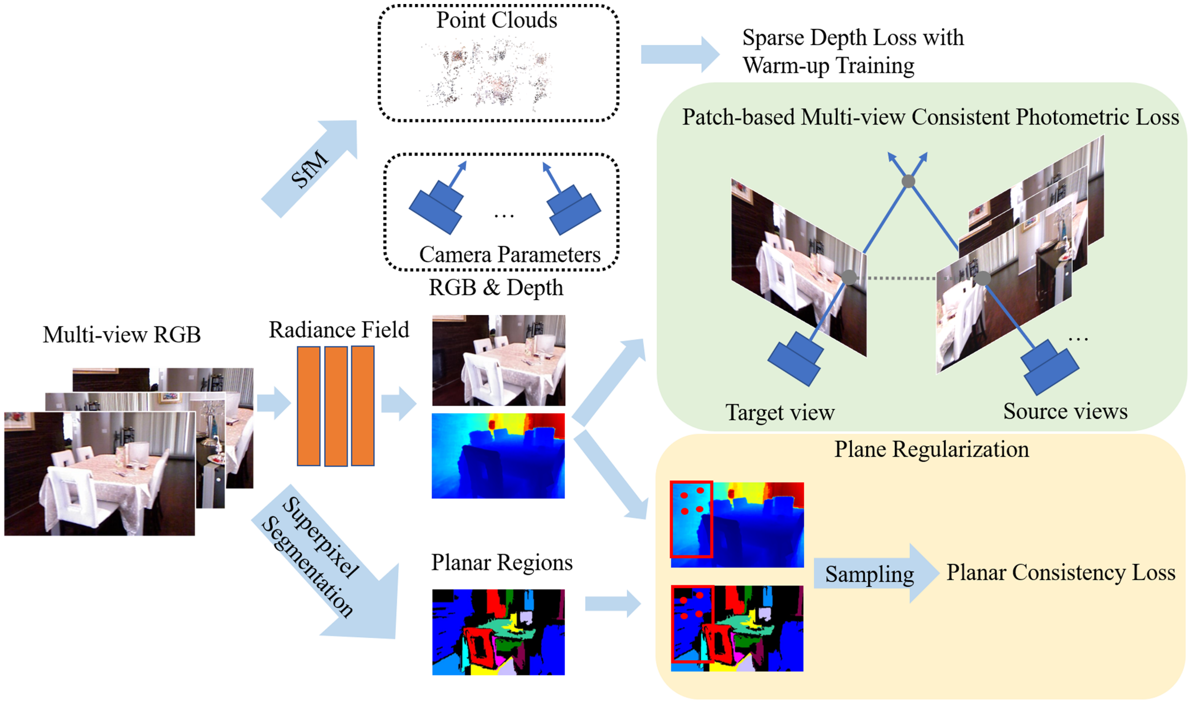

Fig. 2. We exploit the inherent structural hints in sparse views to improve the performance of NeRF in novel view synthesis. First we utilize the structure-from-motion to obtain camera parameters and sparse point clouds. The sparse point clouds are used to constrain the depth of NeRF at keypoints featuring a warm-up training scheme. For regions with rich textures, we utilize the patch-based photometric loss for self-contained dense depth supervision. We also propose a planar consistency loss to regularize the depth of non-texture regions with the assistance of superpixel segmentation.

and additional data (unlike Dense Depth Priors [25]). Besides, we are the first to utilize a 3D planar consistency loss to further improve the quality of view synthesis in texture-less regions, which is also complementary to the multi-view patch-match loss.

B. Self-Supervised Depth Estimation

Self-supervised depth estimation methods are proposed to ease the demand for large-scale labeled training data. SfM-Learner [43] is a pioneering work that supervises the geometry estimations from a depth estimation network by photometric loss. To solve the issue of dynamic objects, optical flow methods are used to compensate for the moving pixels. Semantic masks provided by pretrained semantic segmentation models are also utilized to handle dynamic objects [20]. The approaches do not get satisfactory results in indoor scenes because they do not take into account the non-texture regions.

MovingIndoor [42] is the first self-supervised depth estimation approach focusing on indoor scenes. The authors propose to use the sparse flow via matching with SURF [2] to initialize the optical flow estimation network, SFNet. In the training process, sparse flows are propagated from textured regions to non-textured regions through iterations and finally transformed into dense flows, which are then used to supervise the depth estimation network. P^{2}Net [37] leveraged a patch-based multi-view consistency photometric error to constrain the depths. Other methods also adopt structural regularities such as co-planar constraints to improve depth estimations [15], [18], [37].

Motivated by these self-supervised indoor depth estimation approaches, we propose to utilize the structural hints naturally embedded in indoor scenes to constrain the depth of NeRFs, i.e., the patch-based multi-view consistency loss from [37] and planar consistency loss from [15]. However, previous work mainly focus on the depth estimation task, we are the first to introduce these priors in NeRF and demonstrate that they can significantly resolve the unconstrained NeRF geometry issue and enable higher quality view synthesis.

In detail, our method differs from P2Net [37] and PLNet [15] in the following aspects: (1) StructNeRF is based on NeRF, which operates only on multi-view inputs of a single scene while still demonstrates the effectiveness of using the priors of indoor scenes. However, P2Net [37] and PLNet [15] are based on CNN and require large-scale datasets to train. (2) Previous work and ours focus on different tasks. StructNeRF is primarily used for novel view synthesis by regularizing the geometry. P2Net and PLNet [15] can only be used for depth estimation. (3) P2Net [37] implements patch-based multi-view consistent loss based on sparse key points. We found that the method based on sparse key points is insufficient for NeRF because the key points only account for a small portion of textured regions. In contrast, StructNeRF adopts the method of random sampling in the whole image domain directly.

III. Method

StructNeRF facilitates indoor novel view synthesis given only sparse input images, the framework of which is shown in Fig. 2. First, we obtain sparse point clouds and camera parameters from Structure-from-Motion (SfM). We then incorporate self-supervised depth estimation methods into the optimization of NeRF by imposing patch-based multi-view consistent photometric loss (Section III-B) and planar consistency loss (Section III-C). Lastly, we observe that while point clouds from SfM could serve as sparse depth priors for NeRF, it suffers from noisy estimation, for which we adopt a warm-up training strategy to gradually decay its contribution to the entire optimization (Section III-D). Before introducing our method, we briefly revisit NeRF [21] in Section III-A.

A. Preliminaries

Neural Radiance Fields (NeRF) represents a scene as a continuous neural volume using a multi layer perceptron (MLP) $f_\theta : (\mathbf{x}, \mathbf{d}) \to (\mathbf{c}, \sigma)$ with θ as the learnable parameters, where $\mathbf{x} \in \mathbb{R}^3$ and $\mathbf{d}$ denotes a 3D position abd the view direction, $\sigma \in \mathbb{R}$ and $c \in \mathbb{R}^3$ the corresponding density and radiance.

NeRF is an emission-only model, which means the color of a pixel only depends on the radiance along a ray with no other lighting factors. Therefore, according to volume rendering, the color along the camera ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ that shots from the camera center $\mathbf{o}$ in direction $\mathbf{d}$ can be approximated by numerical quadrature

$$\hat{\mathbf{C}}(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(t)\mathbf{c}(t)dt, \tag{1}$$

where $T(t) = \exp(-\int_{t_n}^{t} \sigma(s)ds)$.

NeRF is optimized by sampling random rays from all training images and minimizing the rendered and ground truth pixel color in L2 norm

$$L_{\text{Color}} = \sum \|\hat{\mathbf{C}}(\mathbf{r}) - \mathbf{C}(\mathbf{r})\|_2^2 \tag{2}$$

B. Patch-Based Multi-View Consistency

The ability of NeRF to model the appearance of view-dependent appearance leads to the ambiguity between its 3D shape and radiance [39]. To reduce the shape-radiance ambiguity, we leverage multi-view consistency explicitly to supervise the depth of every pixel for each view.

To begin with, we render the depth of a given pixel with the formulation proposed in the original NeRF paper. The depth of the ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ can thus be calculated as the following:

$$\hat{\mathbf{D}}(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(t)tdt, \tag{3}$$

After we sample points (or pixels) p_i^t from the pixels of the target view I_t, the original point-based warping process back-projects the extracted points to the source views I_s by

$$p_i^{t\to s} = KM^s M^{t^{-1}}(\hat{\mathbf{D}}(p_i) \odot (K^{-1}p_i^t)), \tag{4}$$

where K denotes camera intrinsic parameters. M_s and M_t are the camera extrinsic parameters of the source view I_s and the target view I_t respectively. $\hat{\mathbf{D}}(p_i)$ is the rendered depth at the pixel p_i. $\odot$ represents Hadamard Product. (Here $\hat{\mathbf{D}}(p_i) = \hat{\mathbf{D}}(p_i^t)$ and we ignore the subscript t of $\hat{\mathbf{D}}(p_i)$ for convenience.)

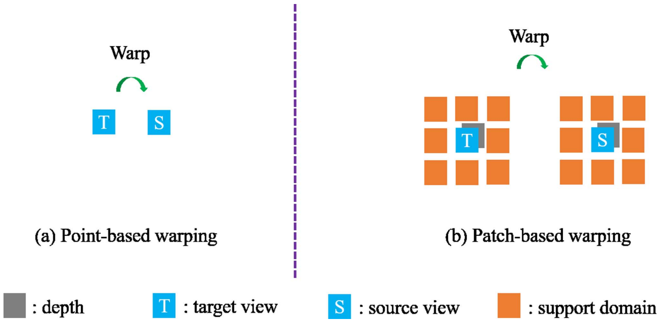

Fig. 3. Point-based and patch-based warping operations. Point-based operation warps pixel-by-pixel and suffers from severe false matching. However, patch-based operation warps a pixel together with its support domain from the target view to the source view, leading to more robust representations in the depth estimation task. This figure is adopted from [37].

Nevertheless, point-based representation is not discriminative enough and may cause false matching because many pixels have the same intensity values in an image. Similar to self-supervised depth estimation [37], we define a *support domain* Ω_{p_i} as the local window surrounding the sampled point p_i. Photometric loss is calculated over each supported domain instead of an isolated point, with which the depth of NeRF can be more accurate because the sampled points combined with their support domains are more unique (Fig. 3).

With more robust depth warping and cross-view matching enabled by the support domain, like [10], [37], we propose to adopt a photometric loss over the support domain Ω_{p_i}, which is the combination of an L1 loss and a structure similarity loss SSIM [32].

$$L_{SSIM} = SSIM(I_t[\Omega_{p_i}^t], I_s[\Omega_{p_i}^{t\to s}]) \tag{5}$$

$$L_{L1} = \left\|I_t[\Omega_{p_i}^t] - I_s[\Omega_{p_i}^{t\to s}]\right\|_1 \tag{6}$$

$$L_{ph} = \alpha L_{SSIM} + (1-\alpha)L_{L1}, \tag{7}$$

where *support domain* Ω_{p_i} is defined as the local window surrounding the sampled point p_i. $I_t[\Omega_{p_i}^t]$ defines the pixel values at $\Omega_{p_i}^t$ in the target view I_t via a bilinear interpolation. $\Omega_{p_i}^{t\to s}$ defines the region after warping the support domain $\Omega_{p_i}^t$ from the target view I_t to the source view I_s. And α is a weighting factor that is set to 0.85 empirically. By definition, L_{ph} is patch-based and multi-view consistent.

Dense depth constraints are proved to be more beneficial to the geometry of neural radiance fields [25]. Therefore, unlike P ^{2} Net [37], we sample points directly from the whole image instead of the keypoints [8]. Our experiments also show that dense sampling results in better performance than that of sampling from keypoints (See Section IV-E). More importantly, in contrast to Dense Depth Priors [25], we achieve dense depth constraints free of any depth completion network which relies on external dataset training and have potential generalization problem.

C. Planar Regularization With Superpixels

Although patch-based photometric loss works well for textured regions, it fails to discriminate non-textured regions that are common in indoor scenes, such as floor, walls, tables and

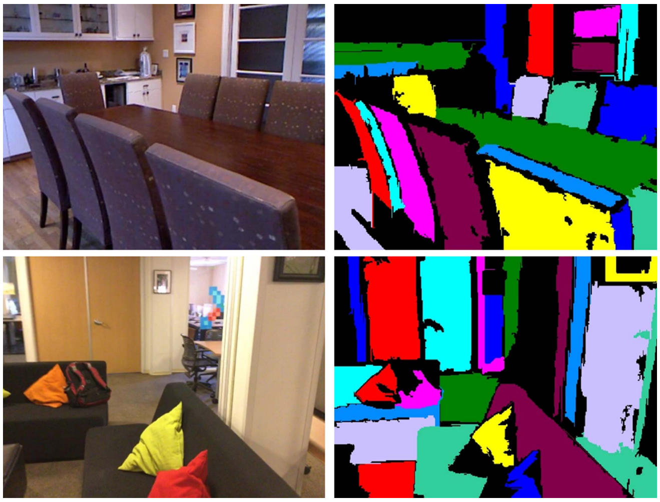

Fig. 4. Superpixel extraction (right) of two indoor images (left), colors represent different regions. We can see that most extracted regions are planes.

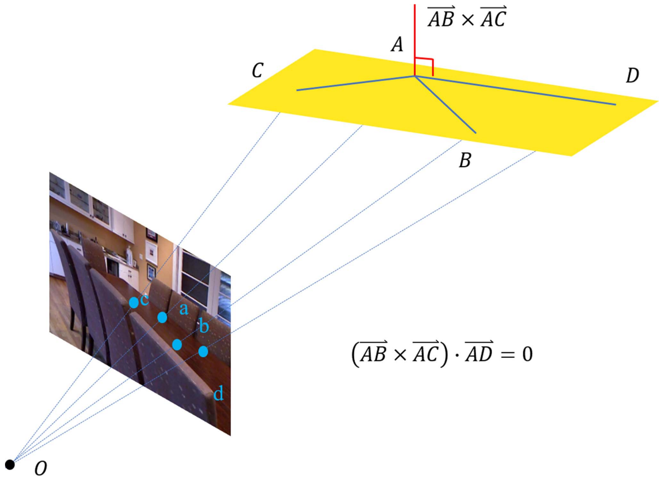
$\overrightarrow{AB} \times \overrightarrow{AC}$
A
C
D
B
c
a
b
d
O
$(\overrightarrow{AB} \times \overrightarrow{AC}) \cdot \overrightarrow{AD} = 0$

Fig. 5. Camestration process. We first re-project points a, b, c, d in a 2D plane to the 3D coordinates, then enforce the cross product of $AB^{\longrightarrow}$ and $AC^{\longrightarrow}$ to be perpendicular to $AD^{\longrightarrow}$. This figure is inspired by [37]

ceiling. We further observe that those non-textured regions are mostly planar. Therefore, how to inform StructNeRF of the planar constraints of a scene is the core concern.

Inspired by self-supervised depth estimation [15], [37], we aim to first identify 2D planes in input images by adopting the Felzenszwalb superpixel segmentation algorithm [9]. Specifically, we extract superpixels from each view and define regions with area larger than a threshold as planes (We set it to be 1000 pixels in our experiments empirically) because those non-textured regions often span over a larger area. Fig. 4 provides examples that most of the segmented regions are planes.

Without specific regularization, NeRFs may fail to preserve the planar properties across different views, i.e. the depth map of planar regions is not flat. We propose to further impose the planar constraint to StructNeRF for non-textured regions using the planar consistency loss [15]. From each plane, we randomly sample 4 pixels, i.e., a, b, c and d. With the rendered depth of StructNeRF, we then transform them to 3D points A, B, C, and D in the camera coordinate with the following equation

$$P = \hat{\mathbf{D}}(p_i) \odot (K^{-1}p_i), p_i \in \{a, b, c, d\}, P \in \{A, B, C, D\}, \quad (8)$$

where K denotes the matrix of camera intrinsic parameters. p_i is the selected pixel and P is the corresponding 3D point.

As shown in Fig. 5, the cross product of $AB^{\longrightarrow}$ and $AC^{\longrightarrow}$ should be perpendicular to the plane where A, B, C and D is located. Therefore, $AD^{\longrightarrow}$ should be perpendicular to $AB^{\longrightarrow} \times AC^{\longrightarrow}$. The planar consistency loss is computed by

$$L_{pc} = \frac{1}{N_p} \sum_{i=1}^{N_p} \left| \overrightarrow{A_iB_i} \times \overrightarrow{A_iC_i} \cdot \overrightarrow{A_iD_i} \right|, \quad (9)$$

where N_p denotes the number of 4-point sets we randomly select from planes.

As shown in the experiments, StructNeRF achieves better performances both in terms of depth estimation and view synthesis for planar regions with the proposed plane regularization.

D. Training Strategy

As introduced in DS-NeRF [7], we also leverage the depth of sparse keypoints extracted by COLMAP [26], [27] to supervise the geometry of the neural radiance field, which is view-consistent in nature

$$L_{sparse} = \sum_{x_i \in \mathbf{X}_j} w_i \left| \hat{\mathbf{D}}(\mathbf{r}_{ij}) - (\mathbf{M}_j x_i) \cdot [0, 0, 1]^T \right|^2, \quad (10)$$

where the keypoint in camera j is reprojected using camera extrinsic parameters $\mathbf{M}_j$ and then is projected onto its unit camera axis [0,0,1]. We also introduce a hyperparameter w_i to adaptively adjust the weights of keypoints to reduce the negative influence of unreliable keypoints, determined by the reprojected error from COLMAP estimation. Supposing a 3D keypoint x_i is visible in camera j, the reprojected error e_{ij} is the distance in pixels between the camera coordinates KM_jx_i and detected 2D keypoint in camera j. Thus, the confidence weight of a keypoint can then be measured by the total reprojected error $e_i = \sum_j e_{ij}$

$$w_i = exp\left(-\left(\frac{e_i}{\bar{e}}\right)^2\right), \quad (11)$$

where $\bar{e}$ denotes the mean absolute error of all the keypoints in a scene.

We optimize the depth of neural radiance field by the weighted combination of patch-based multi-view consistency photometric loss, planar consistency loss and sparse depth loss from COLMAP as follows:

$$L_{\text{Depth}} = \lambda_{ph}L_{ph} + \lambda_{pc}L_{pc} + \lambda_{sparse}L_{sparse} \quad (12)$$

The overall loss function of StructNeRF is therefore

$$L_{\text{Total}} = L_{\text{Depth}} + L_{\text{Color}}, \quad (13)$$

Warm-up Training: During our training, we found that when we use fixed weights for L_{sparse} over all iterations, the results were disturbed by the inaccurate points. Naively reducing L_{sparse} would only diminish the benefits of the accurate points,

TABLE I
COMPARISONS WITH OTHER METHODS

Dataset	Method	Additional Training?	PSNR↑	SSIM↑	LPIPS↓	Depth RMSE↓	Plane Mean Dev↓
NYUv2	MonoSDF [38]	Yes	18.67	0.6313	0.4186	-	-
	RegNeRF [22]	No	**27.35**	0.8442	0.1681	-	-
	NeRF [21]	No	24.44	0.8009	0.2140	0.9846	0.0371
	DSNeRF [7]	No	26.48	0.8325	0.1885	0.3642	0.0322
	Dense Depth Priors [25]	Yes	24.26	0.7862	0.2059	0.8504	**0.0245**
	Ours	No	27.13	**0.8533**	**0.1660**	**0.3250**	0.0286
SUN3D	MonoSDF [38]	Yes	17.29	0.5922	0.4629	-	-
	RegNeRF [22]	No	22.59	0.7157	0.3214	-	-
	NeRF [21]	No	18.85	0.6435	0.3933	1.4919	0.0435
	DSNeRF [7]	No	23.00	0.7401	0.3020	0.4811	0.0372
	Dense Depth Priors [25]	Yes	20.43	0.6653	0.4146	0.8225	**0.0278**
	Ours	No	**23.63**	**0.7677**	**0.2819**	**0.4480**	0.0319
ScanNet	MonoSDF [38]	Yes	18.62	0.6475	0.4504	-	-
	RegNeRF [22]	No	24.71	0.7806	0.2861	-	-
	NeRF [21]	No	24.11	0.7753	0.2923	0.4932	0.0392
	DSNeRF [7]	No	25.06	0.7972	0.2906	0.2207	0.0362
	Dense Depth Priors [25]	Yes	25.24	0.7910	0.2746	**0.1246**	**0.0158**
	Ours	No	**25.34**	**0.8095**	**0.2710**	0.2050	0.0288

The performance of the thickened is the best. The underlined ranks the second in performance.

so we adopt a warm-up training strategy. Specifically, we introduce the sparse depth loss item L_{sparse} only in the first half of the training ($\lambda_{sparse} = 0.05$). In the remaining iterations, we set $\lambda_{sparse} = 0$ and let L_{ph} and L_{pc} refine the depths of pixels where noisy points are located. By setting $\lambda_{sparse} = 0$, the noisy point clouds will not have an impact on NeRF anymore in the later training process. With warm-up training, we strike a balance of utilizing the sparse depth priors and avoiding noises of point clouds.

IV. Experiments

A. Evaluation Metrics

We use peak signal-to-noise (PSNR), the structural similarity index measure (SSIM) [32] and the learned perceptual image patch similarity (LPIPS) [41] to measure the quality of synthesized RGB novel views by comparing them with the ground truth. Besides, we also include two other metrics to demonstrate the effectiveness of our reconstructed geometry over previous methods. We use depth root-mean-square error (Depth RMSE) for measuring the predicted depth maps and the ground truth. Also, the Plane Mean Deviation is used to evaluate the flatness of planes for the predicted depth [15]. It is defined the distance of the measured point cloud to the fitted plane. We use the mean deviation to measure the flatness of planes for the predicted depth. Since the depth from NeRF is in a relative scale, different from the absolute ground truth depth from sensors. We therefore align the predicted depth to the ground truth according to conventional practice [43] because the deviation is scale-variant.

B. Datasets

To evaluate our method, We train and test our model on three multi-view indoor scene datasets in terms of novel view synthesis: ScanNet [6], NYUv2 [29] and SUN3D [35]. We use only the RGB data for training and the ground truth depth are only used for evaluation.

For each scene of the datasets above, We take one frame every 10 or 20 frames from each video in the datasets evenly for training and evaluation. And we run COLMAP [26] over no more than 28 frames to obtain the poses and point clouds. Among the sampled frames, 8-th, 16-th and 24-th frames are used as the test views, and the rest are used as the training views. The interval frames are 20 frames between testing views and the nearest training views. In our experiments, we found that other baselines struggle in this setting because the views are sparse while our method can still get good results.

For ScanNet [6] and SUN3D [35], the image resolution is 468×624 after we down-sample and crop the dark borders from calibration. For NYUv2 [29], the image resolution is 545×415.

C. Implementation Details

We set $\lambda_{ph} = 0.025$ and $\lambda_{pc} = 0.025$ and $\lambda_{sparse} = 0.05$ in all the scenes of each dataset. N is set to 2, and we take the previous two frames and the posterior two frames of the target view as the source views, which is the same as [37]. We use the Adam optimizer [17] with learning rate 0.0005 and sample rays in batches of 1024. The radiance field is optimized for 100 k iterations. We use the same MLP architecture in all experiments as NeRF [21] for fair comparison.

D. Comparison With Existing Methods on Indoor Scenes

We compare StructNeRF with existing methods for novel view synthesis given sparse inputs in the indoor scenes. The baselines include two categories: methods trained by per-scene optimization without external data and methods with data-driven depth priors (we call data-driven methods).

1) Comparison With Per-Scene Optimization Methods: Per-scene optimization methods include vanilla NeRF [21], DSNeRF [7], RegNeRF [22]. We ran all the methods on the three datasets, ScanNet, SUN3D and NYU. The results are shown in Table I and Fig. 8.

StructNeRF is comparable with RegNeRF [22] on NYUv2. Although the optimizations of RegNeRF on unseen poses can still work for indoor scenes, StructNeRF is more robust for indoor scenes relatively because it takes the priors of indoor

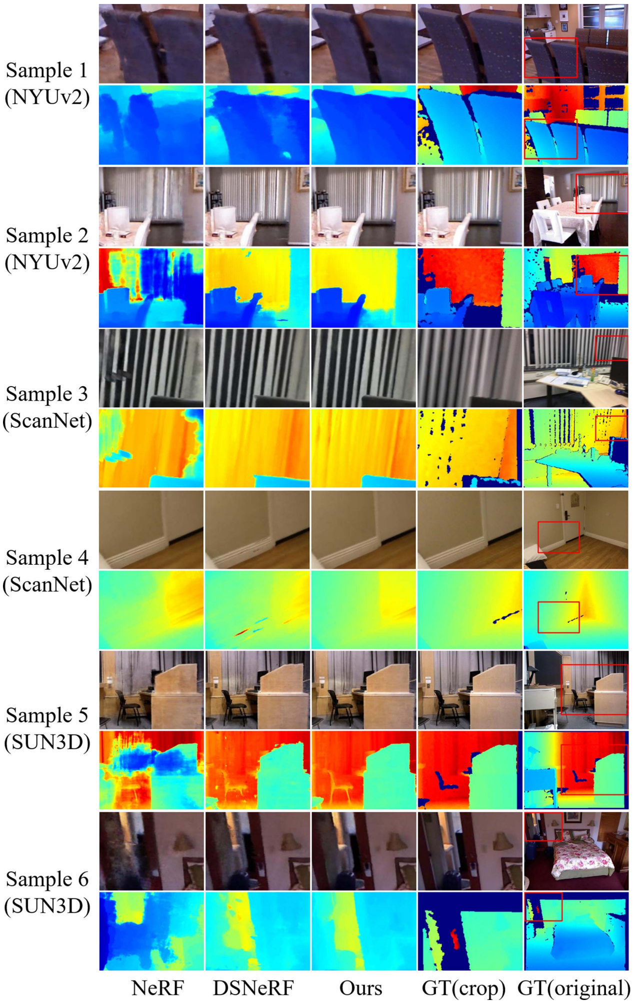

Fig. 6. Comparisons with per-scene optimization method on the test views from NYUv2, ScanNet and SUN3D scenes. “GT” denotes Ground Truth. “GT (original)” and “GT (crop)” mean the original and cropped ground truth respectively.

scenes into consideration. This experiment demonstrates the effectiveness of StructNeRF.

Quantitative comparison in Table I shows that our method outperforms NeRF and DSNeRF in all the metrics. The visualized results of comparisons are listed in Fig. 6. NeRF produces the worst results because its geometry is extremely unconstrained (See Fig. 6). DSNeRF has only sparse depth priors from COLMAP, it often produces artifacts in the depth-unconstrained areas. Wrong color and geometry are produced by DSNeRF, e.g., visible in the chairs (Sample 5 in Fig. 6). In contrast, StructNeRF renders more accurate depths and colors because StructNeRF learns two structural hints which supervise the geometry of NeRF at textured and non-texture regions respectively. Besides, We found that StructNeRF is more robust to the outliers in the sparse depth input (Example 2 and 4 in Fig. 6), while the unnecessary floaters are very obvious for DSNeRF.

2) Comparison With Data-Driven Methods: In addition to the per-scene optimization methods, we also compared

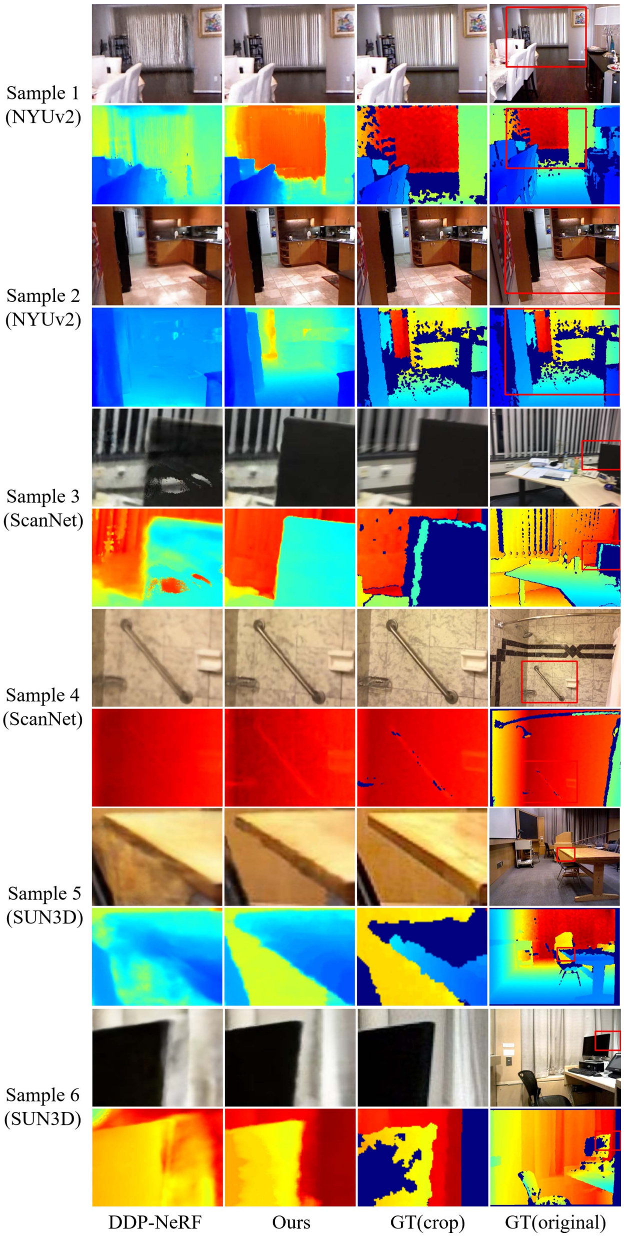

Fig. 7. Comparisons with data-driven method on the test views from NYUv2, ScanNet and SUN3D scenes. DDP-NeRF denotes Dense Depth Priors [25]. “GT” denotes Ground Truth. “GT(original)” and “GT(crop)” mean the original and cropped ground truth respectively.

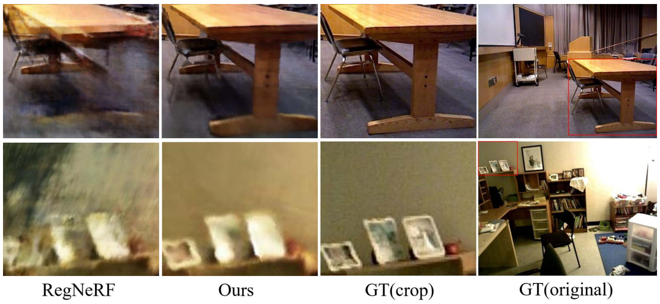

Fig. 8. Comparisons with RegNeRF [22].

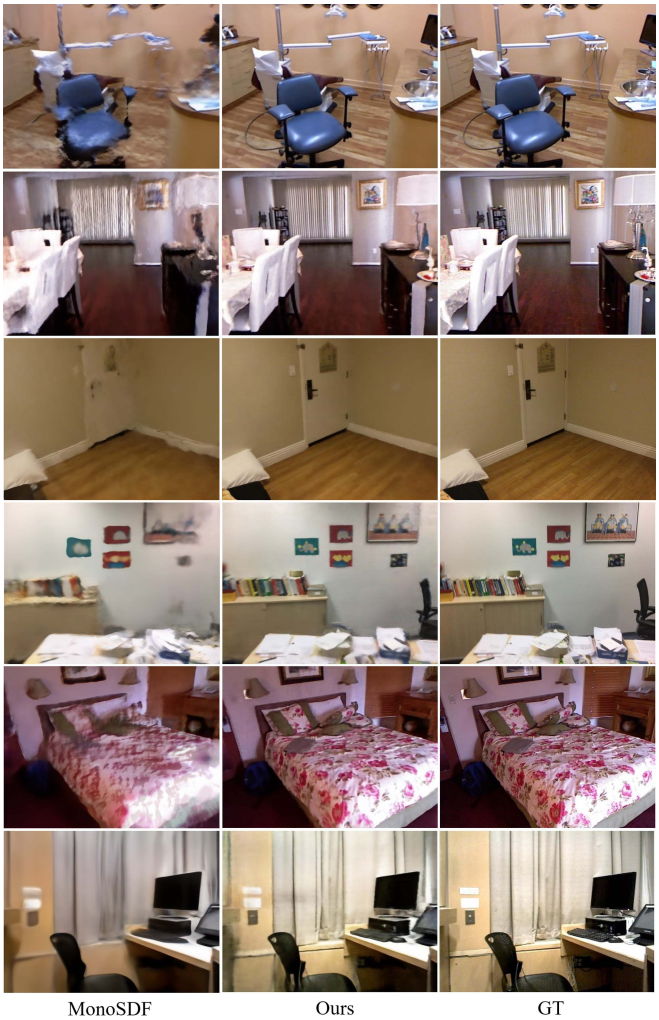

Fig. 9. Comparisons with MonoSDF on the test views from NYUv2, ScanNet and SUN3D scenes. “GT” denotes Ground Truth. The first two rows are from NYUv2. The next two rows are from ScanNet. The last two rows are from SUN3D.

StructNeRF with some data-driven methods: MonoSDF [38] and Dense Depth Priors (DDP-NeRF) [25].

The results of MonoSDF are presented in Table I and Fig. 9 demonstrate that StructNeRF significantly outperforms MonoSDF. Despite MonoSDF's incorporation of monocular depth and normal priors to enhance the geometric quality of neural SDF, these priors are estimated per-scene and thus are not multi-view consistent. In contrast, our approach incorporates a patch-based multi-view consistency loss, which guarantees cross-view consistency and is conducive to the view synthesis task. And StructNeRF renders more accurate colors because StructNeRF is also constrained by co-planar consistency at non-texture regions.

We also compared our method to the recent work (Dense Depth Priors [25]) that uses the depth completion network to complete depths from sparse depth inputs and then uses them to supervise the geometry of NeRF. Different from our method, the depth completion network of Dense Depth Priors needs to be pretrained on a large-scale indoor dataset ScanNet in a supervised way. The completed dense depths for new scenes are pre-calculated by the depth completion network and used to supervise the depth of NeRF when training NeRF on the new scenes. Like NeRF [21], our method is only optimized on the specific scene without additional training. Note we didn't include a comparison with another relevant work NerfingMVS [34] because it fails on most scenes for reasons we explained in the appendix, available online.

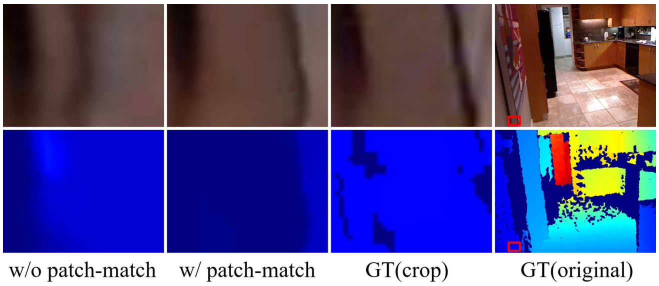

Fig. 10. w/o patchmatch and w/ patchmatch. ((7) of Section III-B).

The results in Table I show that, with the dataset prior, Dense Depth Priors is better than StructNeRF on ScanNet in terms of Depth RMSE and Plane Mean Dev. But StructNeRF still has comparable PSNR, SSIM and LPIPS compared with Dense Depth Priors. We also find that our method is more view-consistent than Dense Depth Priors (shown in the Example 1 and 4 of Fig. 7) because the predicted depths of the depth completion network are not view-consistent.

E. Ablation Study

We conduct ablation studies on NYUv2 to further validate the effectiveness of the different components in StructNeRF for view synthesis and depth estimation. The quantitative results can be found in Table II.

Patch-match: Omitting patch-match leads to inaccurate depth and color in high-frequency areas. The ability of NeRF to model the appearance of view-dependent appearance leads to the ambiguity between its 3D shape and radiance [39]. With multi-view consistency, patch-match improves the geometry of textured regions and reduces the artifacts in the edges such as the edges of the billboard shown in Fig. 10. When it is omitted, the black edge appears in a wrong place and its shape is wrongly estimated because it lacks the corresponding geometry constraints. L_{sparse} only provides the depth constraints ats sparse keypoints and L_{pc} at non-textured regions. Patch-based multi-view consistency is necessary for the textured regions other than keypoints.

Plane Regularization: Removing plane regularization causes the geometry of textureless regions becomes less constrained, leading to less sharp edges in RGB, as shown in Fig. 11. As a result, the Depth RMSE and Plane Mean Dev degrades. In detail, after we remove the plane regularization, the left white area is mistaken for two planes and the colored edge between the white and brown area is less clear. Since the white area corresponds to a plane in StructNeRF represented by a superpixel, our proposed

TABLE II
ABLATION STUDIES ON NYUv2 DATASETS

Method	Additional training?	PSNR↑	SSIM↑	LPIPS↓	Depth RMSE↓	Plane Mean Dev↓
w/o dense-sampling	No	26.31	0.8264	0.1887	0.3394	0.0297
w/o patch	No	27.12	0.8513	0.1680	0.3685	0.0289
w/o warm-up training	No	26.67	0.8427	0.1754	0.3635	0.0301
w/o sparse depth priors	No	26.31	0.8450	0.1770	0.4567	0.0334
w/o patch-match	No	26.60	0.8440	0.1789	0.3625	**0.0284**
w/o plane regularization	No	26.92	0.8453	0.1718	0.3413	0.0295
full	No	**27.13**	**0.8533**	**0.1660**	**0.3250**	0.0286

The performance of the thickened is the best.

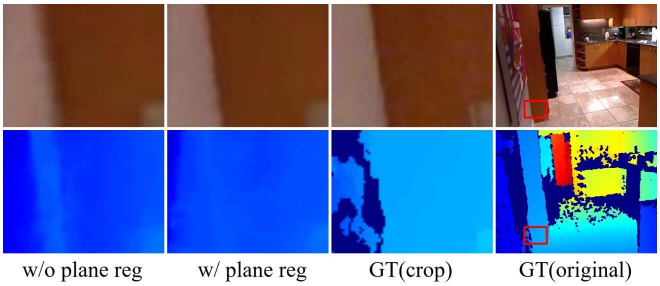

Fig. 11. w/o plane regularization and w/ plane regularization. ((9) of Section III-C).

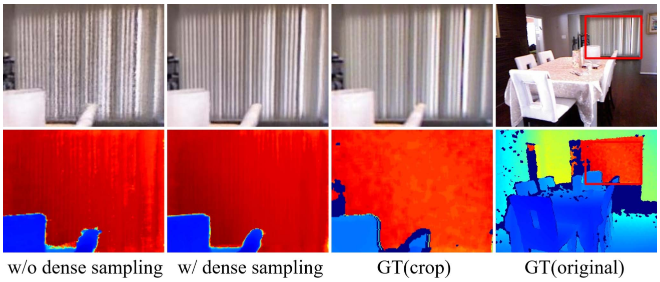

Fig. 13. w/o dense sampling and w/ dense sampling. (See the last paragraph of Section III-B for details).

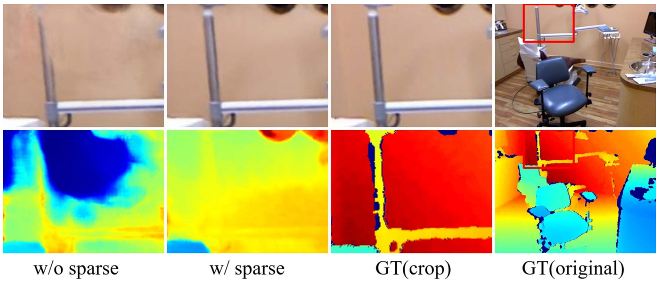

Fig. 12. w/o sparse and w/ sparse. ((10) of Section III-D).

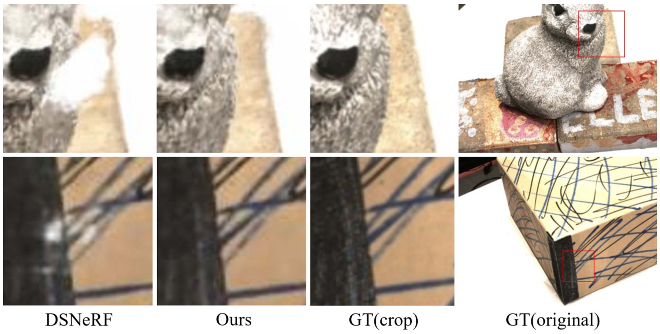

Fig. 14. Comparisons with DSNeRF for DTU [1]. “GT” denotes Ground Truth. “GT(original)” and “GT(crop)” mean the original and cropped ground truth respectively.

TABLE III
COMPARISONS WITH DSNERF ON OBJECT DATASETS

Dataset	Method	PSNR↑	SSIM↑	LPIPS↓
Redwood	DSNeRF [7]	**19.53**	0.5194	0.4605
	Ours	**19.53**	**0.5408**	**0.4389**
DTU	DSNeRF [7]	18.39	0.8138	0.1555
	Ours	**18.83**	**0.8392**	**0.1382**

The performance of the thickened is the best.

planar consistency loss enforces flat geometry in this region, which reduces the artifacts in both the color and depth. It makes up for the insufficiency of patch-match in the non-texture regions.

Patch-based versus Point-based Photometric Loss: We replace the patch-based multi-view consistency photometric loss with the point-based one. It can be seen from Table II that patch-based loss leads to a robust rendering quality.

Sparse Depth Priors: Excluding the sparse depth priors, we observe that NeRF is more likely to fall into the local optimal as shown in Fig. 12. Therefore, although sparse depth priors from COLMAP contain many noises, our experiments show that they are still indispensable for NeRF with sparse views.

Warm-up Training: Omitting the warm-up training strategy and using the same λ_{sparse} across all iterations makes all the metrics worse since the noises of point clouds become much more obvious in the rendering results without the warm-up training.

Dense-sampling: In this experiment, we sample at the keypoints extracted by [8] in patch-match to supervise the depth of NeRF. We observe that sparse keypoint sampling leads to under-constrained depth and worse rendering results (as shown in Fig. 13). Dense sampling instead helps patch-match supervise the geometry of most regions.

F. Comparisons on Other Datasets

1) Object Dataset: StructNeRF was also compared to DSNeRF under Redwood and DTU datasets. The results presented in Figs. 14, 15 and Table III clearly demonstrate that StructNeRF consistently outperforms DSNeRF on object-level datasets. In the DTU dataset, all metrics of our method are better than DSNeRF. When considering the Redwood dataset,

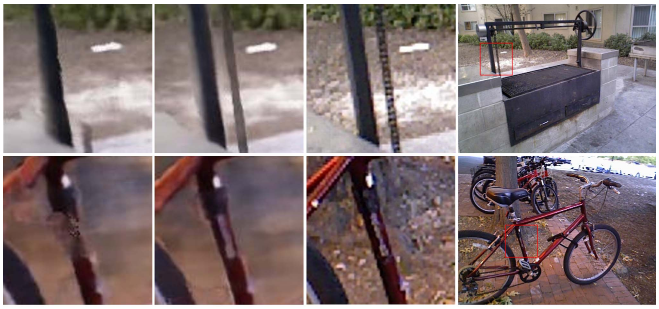

Fig. 15. Comparisons with DSNeRF for Redwood [5]. "GT" denotes Ground Truth. "GT(original)" and "GT(crop)" mean the original and cropped ground truth respectively.

TABLE IV
COMPARISONS ON OUTDOOR SCENES OF ETH3D

Method	Need Pretraining?	PSNR↑	SSIM↑	LPIPS↓
NeRF	No	16.69	0.5597	0.5858
DSNeRF	No	18.26	0.5313	0.6186
Ours	No	**18.60**	**0.5948**	**0.5806**

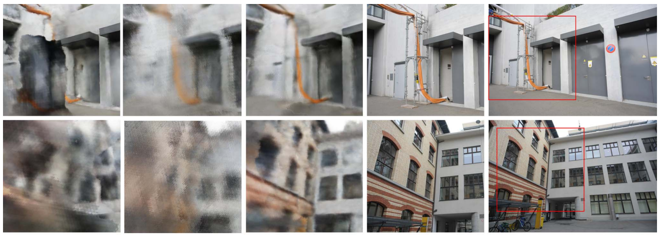

Fig. 16. Comparisons on outdoor scenes of ETH3D.

StructNeRF achieves higher results in terms of SSIM and LPIPS, and comparable PSNR values compared to DSNeRF. The inclusion of patch-based multi-view consistency loss in NeRF proves to be advantageous for enhancing NeRF's geometric estimation.

2) Outdoor Dataset: We apply StructNeRF to three outdoor scenes of ETH3D [28] (courtyard, electro and terrace).

As shown in Tab IV, all the three methods struggles on ETH3D, but all the metrics of our method are still higher than NeRF and DSNeRF. And the visualization results (Fig. 16) also show that StructNeRF is superior to NeRF and DSNeRF on outdoor scenes. StructNeRF is more suitable for indoor scenes because indoor scenes contains more textureless regions which can be improved by plane consistency loss. Outdoor scenes contain much less textureless regions. Besides, StructNeRF cannot work on the reflective areas such as windows where NeRF and DSNeRF cannot work as well with sparse views.

G. Time Analysis of StructNeRF

We ran NeRF and StructNeRF on scene0753_00 of ScanNet for example. If we train NeRF for 1 million iterations, only LPIPS will be slightly better while PSNR and SSIM became worse due to overfitting (See Table V).

TABLE V
TIME ANALYSIS FOR NERF AND STRUCTNERF

Method	PSNR↑	SSIM↑	LPIPS↓	time(hours)
NeRF(100k)	18.64	0.6666	0.3579	4.3
NeRF(200k)	18.29	0.6609	0.3507	8.58
NeRF(300k)	18.48	0.6617	0.3475	12.77
NeRF(1million)	18.39	0.6582	0.3344	42.93
Ours(100k)	21.09	0.7297	0.3041	9.3

We list the amount of training time NeRF and StructNeRF needs on one 2080ti GPU. Our method needs the double amount of time compared to NeRF because we need to perform extra loss calculation.

It is a little hard to train a NeRF model and StructNeRF model with exactly the same time. So we try to run NeRF for more iterations to make approximate comparison. We noticed that NeRF(200 k) and NeRF(300 k) are still inferior to StructNeRF(100 k) because it already converges when training with 100 k iterations.

V. CONCLUSION AND FUTURE WORK

This paper proposes StructNeRF, neural radiance field with self-supervised depth constraints for indoor scene novel view synthesis with sparse input views. We are the first to apply structural hints from multi-view inputs to NeRF for view synthesis and geometry estimation, specifically, patch-match and plane regularization to constrain the depth of textured and textureless regions respectively. In this way, it learns a view-consistent geometry with dense depth constraints. Most importantly, we doesn't have the generalization problem which occurred in data-driven methods, e.g., Dense Depth Priors [25]. Besides, we adopt a warm-up training strategy to reduce the influence of noisy point clouds from Structure-from-Motion. StructNeRF outperforms state-of-the-arts without additional training [7], [21] both in depth estimation and novel view synthesis. In terms of comparison to data-driven methods, i.e., Dense Depth Priors [25], it still achieves comparable performance on the pretrained dataset (ScanNet) and superior performance on other datasets (SUN3D and NYUv2). StructNeRF raises the upper bound of rendering quality of NeRF without external data given sparse input views. Our work also motivates future research to further exploit the structural hints in multi-view inputs for view synthesis and other related tasks.

Limitations of StructNeRF include limited view-dependent effects since the surfaces are observed by only a few input views, which also happens in other baselines [7], [21], [25]. Also, our method in plane reconstruction is still inferior to supervised data-driven methods, albeit we already surpassed per-scene optimization ones significantly. In the future, We will consider how to model the view-dependent effect in the sparse input setting. We would also investigate how to incorporate the rich priors of indoor datasets, possibly with a more generalized NeRF trained across large-scale datasets.

References

[1] H. Aanæs, R. Ramsbøl Jensen, G.E. VogiatzisTola, and A. B. Dahl, "Large-scale data for multiple-view stereopsis," *Int. J. Comput. Vis.*, vol. 120, no. 2, pp. 153–168, 2016.

[2] H. Bay, T. Tuytelaars, and L. V. Gool, "SURF: Speeded up robust features," in *Proc. 9th Eur. Conf. Comput. Vis.*, Springer, 2006, pp. 404–417..

[3] A. Chen et al., "MVSNeRF: Fast generalizable radiance field reconstruction from multi-view stereo," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Montreal, QC, Canada, 2021, pp. 14104–14113.

[4] J. Chibane, A. Bansal, V. Lazova, and G. Pons-Moll, "Stereo radiance fields (SRF): Learning view synthesis for sparse views of novel scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 7911–7920.

[5] S. Choi, Q.-Y. Zhou, S. Miller, and V. Koltun, "A large dataset of object scans," 2016, *arXiv:1602.02481*.

[6] A. Dai, A. X. Chang, M. Savva, M. Halber, T. A. Funkhouser, and M. Nießner, "ScanNet: Richly-annotated 3D reconstructions of indoor scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Honolulu, HI, USA, 2017, pp. 2432–2443.

[7] K. Deng, A. Liu, J.-Y. Zhu, and D. Ramanan, "Depth-supervised NeRF: Fewer views and faster training for free," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, New Orleans, LA, USA, 2022, pp. 12872–12881.

[8] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," 2016, *arXiv:1607.02565*.

[9] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vis.*, vol. 59, no. 2, pp. 167–181, 2004.

[10] C. Godard, O. M. Aodha, M. Firman, and G. J. Brostow, "Digging into self-supervised monocular depth estimation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis., Seoul, Korea (South)*, 2019, pp. 3827–3837.

[11] Y. Guo, D. Kang, L. Bao, Y. He, and S.-H. Zhang, "NeRFReN: Neural radiance fields with reflections," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, New Orleans, LA, USA, 2022, pp. 18388–18397.

[12] P. Hedman, T. Ritschel, G. Drettakis, and G. J. Brostow, "Scalable inside-out image-based rendering," *ACM Trans. Graph.*, vol. 35, no. 6, pp. 231:1–231:11, 2016.

[13] A. Jain, M. Tancik, and P. Abbeel, "Putting NeRF on a diet: Semantically consistent few-shot view synthesis," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Montreal, QC, Canada, 2021, pp. 5865–5874.

[14] W. Jang and L. Agapito, "CodeNeRF: Disentangled neural radiance fields for object categories," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Montreal, QC, Canada, 2021, pp. 12929–12938.

[15] H. Jiang, L. Ding, J. Hu, and R. Huang, "PLNet: Plane and line priors for unsupervised indoor depth estimation," in *Proc. Int. Conf. 3D Vis.*, London, United Kingdom, 2021, pp. 741–750.

[16] M. Kim, S. Seo, and B. Han, "InfoNeRF: Ray entropy minimization for few-shot neural volume rendering," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, New Orleans, LA, USA, 2022, pp. 12902–12911.

[17] P. DiederikKingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Representations*, San Diego, CA, USA, 2015.

[18] B. Li, Y. Huang, Z. Liu, D. Zou, and W. Yu, "StructDepth: Leveraging the structural regularities for self-supervised indoor depth estimation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Montreal, QC, Canada, 2021, pp. 12643–12653.

[19] C.-H. Lin, W.-C. Ma, A. Torralba, and S. Lucey, "BARF: Bundle-adjusting neural radiance fields," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Montreal, QC, Canada, 2021, pp. 5721–5731.

[20] Y. Meng et al., "SIGNet: Semantic instance aided unsupervised 3D geometry perception," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Long Beach, CA, USA, 2019, pp. 9810–9820.

[21] B. Mildenhall et al., "NeRF: Representing scenes as neural radiance fields for view synthesis," in *Proc. 16th Eur. Conf.*, Glasgow, U.K., 2020, pp. 405–421.

[22] M. Niemeyer et al., "RegNeRF: Regularizing neural radiance fields for view synthesis from sparse inputs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, New Orleans, LA, USA, 2022, pp. 5470–5480.

[23] K. Park et al., "Nerfies: Deformable neural radiance fields," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Montreal, QC, Canada, 2021, pp. 5845–5854.

[24] K. Park et al., "HyperNeRF: A higher-dimensional representation for topologically varying neural radiance fields," *ACM Trans. Graph.*, vol. 40, no. 6, pp. 238:1–238:12, 2021.

[25] B. Roessle, J. T. Barron, B. Mildenhall, P. P. Srinivasan, and M. Nießner, "Dense depth priors for neural radiance fields from sparse input views," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, New Orleans, LA, USA, 2022, pp. 12882–12891.

[26] L. Johannes Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, 2016, pp. 4104–4113.

[27] J. L. Schönberger, E. Zheng, J.-M. Frahm, and M. Pollefeys, "Pixelwise view selection for unstructured multi-view stereo," in *Proc. Eur. Conf. Comput. Vis.*, Amsterdam, The Netherlands, 2016, pp. 501–518.

[28] T. Schöps et al., "A multi-view stereo benchmark with high-resolution images and multi-camera videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Honolulu, HI, USA, 2017, pp. 2538–2547.

[29] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *Proc. Eur. Conf. Comput. Vis.*, Florence, Italy, 2012, pp. 746–760.

[30] D. Verbin, P. Hedman, B. Mildenhall, T. E. Zickler, J. T. Barron, and P. P. Srinivasan, "Ref-NeRF: Structured view-dependent appearance for neural radiance fields," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, New Orleans, LA, USA, 2022, pp. 5481–5490.

[31] C. Wang, X. Wu, Y. Guo, S.-H. Zhang, Y.-W. Tai, and S.-M. Hu, "NeRF-SR: High quality neural radiance fields using supersampling," in *Proc. 30th ACM Int. Conf. Multimedia*, Lisboa, Portugal, 2022, pp. 6445–6454.

[32] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[33] Z. Wang, S. Wu, W. Xie, M. Chen, and V. A. Prisacariu, "NeRF-: Neural radiance fields without known camera parameters," 2021, *arXiv:2102.07064*.

[34] Y. Wei, S. Liu, Y. Rao, W. Zhao, J. Lu, and J. Zhou, "NerfingMVS: Guided optimization of neural radiance fields for indoor multi-view stereo," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Montreal, QC, Canada, 2021, pp. 5590–5599.

[35] J. Xiao, A. Owens, and A. Torralba, "SUN3D: A database of big spaces reconstructed using sfm and object labels," in *Proc. IEEE Int. Conf. Comput. Vis.*, Sydney, Australia, 2013, pp. 1625–1632.

[36] A. Yu, V. Ye, M. Tancik, and A. Kanazawa, "pixelNeRF: Neural radiance fields from one or few images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 4578–4587.

[37] Z. Yu, L. Jin, and S. Gao, "P2net: Patch-match and plane-regularization for unsupervised indoor depth estimation," in *Proc. Eur. Conf. Comput. Vis.*, Glasgow, U.K., 2020, pp. 206–222.

[38] Z. Yu, S. Peng, M. Niemeyer, T. Sattler, and A. Geiger, "MonoSDF: Exploring monocular geometric cues for neural implicit surface reconstruction," *Proc. Int. Conf. Neural Inf. Process. Syst.*, vol. 35, pp. 25018–25032, 2022.

[39] K. Zhang, G. Riegler, N. Snavely, and V. Koltun, "NeRF : Analyzing and improving neural radiance fields," 2020, *arXiv: 2010.07492*.

[40] K. Zhang, G. Riegler, N. Snavely, and V. Koltun, "NeRF : Analyzing and improving neural radiance fields," 2020, *arXiv: 2010.07492*.

[41] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, 2018, pp. 586–595.

[42] J. Zhou, Y. Wang, K. Qin, and W. Zeng, "Moving indoor: Unsupervised video depth learning in challenging environments," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Seoul, Korea (South), 2019, pp. 8617–8626.

[43] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Honolulu, HI, USA, 2017, pp. 6612–6619.

Zheng Chen received the BS degree in computer science and technology from Jilin University, Changchun, in 2020. He is currently working toward the PhD degree with Tsinghua University, Beijing. His research interests include computer graphics and computer vision.

Chen Wang received the bachelor's and master's degrees in computer science and economics from Tsinghua University. He is working toward the PhD degree with the University of Pennsylvania. His research interests include computer graphics and computer vision.

Yuan-Chen Guo received the bachelor's degree from Tsinghua University, in 2019. He is currently working toward the PhD degree with the Department of Computer Science and Technology, Tsinghua University. His research interests include computer graphics and computer vision.

Song-Hai Zhang (Member, IEEE) received the PhD degree in computer science and technology from Tsinghua University, in 2007. He is currently an associate professor with the Department of Computer Science and Technology, Tsinghua University. His research interests include computer graphics, virtual reality and image/video processing.