Self-supervised Super-plane for Neural 3D Reconstruction

Botao Ye[1] Sifei Liu[2] Xueting Li[2] Ming-Hsuan Yang[3,4]
[1]University of Chinese Academy of Sciences [2]NVIDIA
[3]University of California, Merced [4]Yonsei University

Abstract

Neural implicit surface representation methods show impressive reconstruction results but struggle to handle texture-less planar regions that widely exist in indoor scenes. Existing approaches addressing this leverage image prior that requires assistive networks trained with large-scale annotated datasets. In this work, we introduce a self-supervised super-plane constraint by exploring the free geometry cues from the predicted surface, which can further regularize the reconstruction of plane regions without any other ground truth annotations. Specifically, we introduce an iterative training scheme, where (i) grouping of pixels to formulate a super-plane (analogous to super-pixels), and (ii) optimizing of the scene reconstruction network via a super-plane constraint, are progressively conducted. We demonstrate that the model trained with super-planes surprisingly outperforms the one using conventional annotated planes, as individual super-plane statistically occupies a larger area and leads to more stable training. Extensive experiments show that our self-supervised super-plane constraint significantly improves 3D reconstruction quality even better than using ground truth plane segmentation. Additionally, the plane reconstruction results from our model can be used for auto-labeling for other vision tasks. The code and models are available at https://github.com/botaoye/S3PRecon.

1. Introduction

Reconstructing 3D scenes from multi-view RGB images is an important but challenging task in computer vision, which has numerous applications in autonomous driving, virtual reality, robotics, *etc*. Existing matching-based methods [30, 31, 50] estimate per-view depth maps, which are then fused to construct 3D representation. However, they do not recover the depth of the scene in texture-less planar areas well (such as walls, floors, and other solid color planes), which are abundant, especially for indoor scenes. Recent data-driven methods [22, 26, 34, 38] alleviate this problem to some extent by automatically learning geometric priors from large-scale training data, but they either require numerous and expensive 3D supervision (*e.g*., depth [22, 38], normal [14], *etc*.) or lack fine-grain details [26, 34].

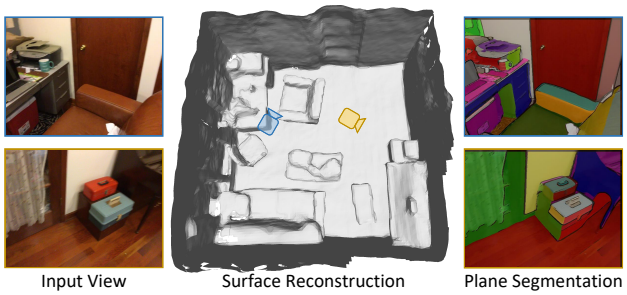


Figure 1. **Reconstruction and Plane Segmentation Results.** Our method can reconstruct smooth and complete planar regions by employing the super-plane constraint and further obtain plane segmentation in an unsupervised manner.

Recently, neural implicit representations have gained much attention and shown impressive reconstruction results without 3D supervision [39, 46, 47]. However, these methods purely rely on the photo-consistency to construct the scene, which leads to texture-geometry ambiguities since there are different plausible interpretations to satisfy this objective. Several approaches address this problem by introducing additional priors obtained from trained models that can be considered as *assistant networks*. For instance, ManhattanSDF [9] introduces the Manhattan assumption on the floor and wall regions, which are predicted by a semantic segmentation model. NeuRIS [37] and MonoSDF [48] adopt additional normal supervision acquired from normal prediction networks trained on large-scale labeled datasets. Although these methods can regularize the reconstruction process on indoor scenes, they all rely on large-scale annotated 2D or 3D datasets. In addition, these pretrained geometric prediction networks are sensitive to different scenes and not friendly across different domains or datasets. For example, MonoSDF [48] reports that different normal prediction networks significantly affect the reconstruction quality. Thus, a natural question arises: *can we improve the RGB-based reconstruction results on texture-less regions without any implicit supervision from assistant networks?*

In this work, we propose a novel neural 3D reconstruction framework with the Self-Supervised Super-Plane con-

straint, called S^3P, which does not require any labeled data or assistant networks. The intuition behind our approach is simple: the constructed results provide a free geometry cue, *i.e.*, surface normal, which can be utilized to guide the reconstruction process of planar regions. Specifically, pixels belonging to parallel planes tend to have similar normal directions (shown in Fig 3). We group pixels sharing similar normal values into the same cluster, which we call a super-plane (analogous to a group of pixels is called superpixel). A super-plane constraint is then applied to force normal directions within the same super-plane to be consistent, thus constraining the reconstruction of large super-plane regions. Due to the ambiguity of the prediction, especially at the early training stages, the grouped super-plane can be inaccurate and introduces noisy self-supervision signals. Therefore, an automatic filtering strategy is further designed to compare the normals of each pixel with the estimated super-plane normal, *i.e.*, the average normal of all pixels belonging to the same super-plane, and outliers with large angular differences will be filtered out. We also detect the discontinuity of the surface according to the geometry and color features and mask out non-plane edge regions in the super-plane segmentation maps.

Notably, by using normals for grouping, multiple parallel planes will be grouped together, so that our super-planes are typically larger than individual planes. This property is particularly beneficial for volume rendering-based training process [25] since planes with more pixels will also have more stable and accurate averaged normals when limited pixels are sampled in each iteration. We experimentally verify this benefit: our super-plane constraint yields better results than adopting ground truth plane segmentation.

As a by-product, self-supervised plane segmentation of the reconstructed scene can be easily obtained from the super-plane masks by extracting connected components separated by the detected non-plane edge regions. Thus, our approach can be extended to reconstruct planes of a scene without ground truth supervision. It can also be applied to label new scenes automatically for applications that require such training data. The main contributions of this work are:

- We introduce a super-plane constraint for neural implicit scene reconstruction by first generating super-planes in an unsupervised manner and then performing automatic outlier filtering.
- Our super-plane segmentation method can be further extended to get unsupervised plane reconstruction results, which can be used as auto-labeling.
- Experimental results show that our method significantly improves the reconstruction quality of texture-less planar regions, and the *unsupervised* plane reconstruction results are comparable to those from state-of-the-art *supervised* methods.

Methods	Explicit 3D supervision	Implicit 2D/3D supervision	Handle texture-less
Patch Match-based MVS	×	×	×
Data-driven MVS	✓	×	✓
NeuS [39], VolSDF [46]	×	×	×
ManhattanSDF [9]	×	✓(2D)	✓
NeuRIS [37], MonoSDF [48]	×	✓(3D)	✓
Ours	×	×	✓

Table 1. **Comparison between different reconstruction methods.** Our method can handle texture-less planar regions without implicit supervision provided by assistant networks.

2. Related Work

MVS-based 3D Reconstruction. Reconstructing 3D scenes from images is a long-standing computer vision task. Multi-view stereo (MVS) methods [29, 31, 31, 50] first estimate the per-view depth map based on feature matching, followed by depth fusion [4, 24] and meshing [12]. However, these approaches do not handle texture-less planar regions well because dense feature matching is intractable in these regions. Recent learning-based MVS methods can alleviate this problem by obtaining geometric priors from training datasets. For instance, MVSNet [44] and their variants [8, 35, 45] extract image features to build cost volumes, which are further fed into a 3D CNN to predict depth maps. They also resort to exploiting the depth-normal consistency to better handle texture-less regions [14, 22]. Such a two-stage pipeline lacks global scene consistency due to the individual estimation of each view and thus often suffers from noisy and incomplete results. Other approaches [26, 34] address this issue by regressing scene depth values using the Truncated Signed Distance Function (TSDF). TSDF constructs a spatially discretized representation that leads to a limited capability to model fine details, like thin surfaces. Moreover, all these methods require large-scale training datasets with 3D supervision, and the geometric priors are acquired from specific training datasets, making them difficult to generalize due to the domain gap.

Neural Implicit Surface Reconstruction. Recently, neural implicit representation methods [21, 25, 39, 46, 47] encode the scene into light-weight Multi-layer Perceptrons (MLPs). By combining neural radiance field with volume rendering techniques, NeRF [25] shows impressive novel view synthesis results but fails to extract accurate surface due to the ambiguities of the underlying radiance field representation. Therefore, NeuS [39] and VolSDF [46] instead use a signed distance field (SDF) to represent the scene, which largely improves the reconstruction quality. These methods can learn the scene geometry purely from posed 2D images and are able to produce high-resolution reconstruction without large memory consumption. However, similar to the MVS approaches, they cannot handle texture-less regions well due to ambiguities. NeuralRGB-D [2] and Go-Surf [36] use ground truth depth maps during training, and while im-

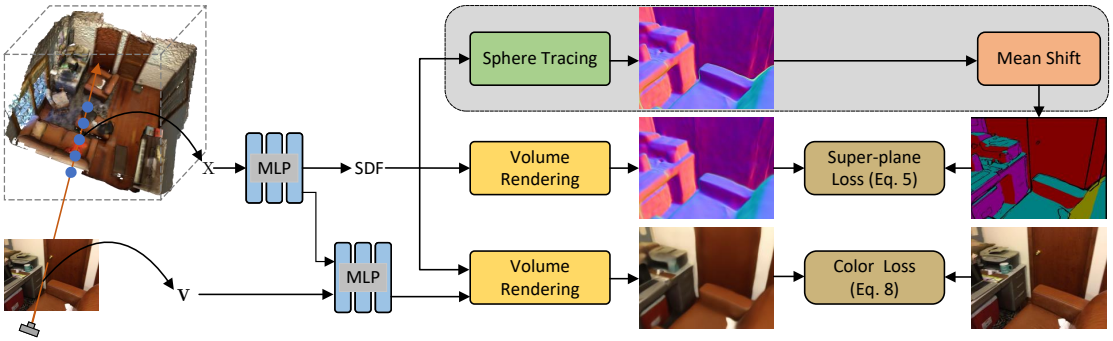


Figure 2. **Overview of S^{3}P.** By exploiting the free geometry cues (*i.e.*, normal direction) existed in the constructed surface, we can generate super-plane segmentation in an unsupervised manner, which is used to constrain the reconstruction process of planar regions.

pressive results are obtained, less common depth cameras are required. Some recent approaches instead introduce additional geometric priors (*e.g.*, normal [37, 48] obtained from assistant networks to guide the optimization of these planar areas. However, these assistant networks need to be carefully selected and trained on large-scale annotated datasets [5] and are also sensitive to the domain gap. In this work, we show that without additional ground truth or assistant networks, the reconstruction quality in texture-less regions can be improved by exploring the self-supervised super-plane constraint. Comparison between the proposed method and existing reconstruction models are summarized in Tab. 1.

Plane Reconstruction. Traditional plane reconstruction methods [3, 15, 33] recover 3D planes by mining the geometric cues in 2D images such as line segments and vanishing points. However, they rely on strong assumptions (*e.g.*, Manhattan world assumption [6]) about the scene and thus cannot generalize well to various application scenarios. Recently, learning-based approaches [18, 19, 41, 43, 49] can directly infer piecewise planar regions by obtaining rich geometric priors in the dataset. Some methods extend bottom-up semantic segmentation [19, 49] or top-town instance segmentation [19] to predict the segmentation mask and parameters of the per-view plane. Recent methods [11, 17, 20] construct planes from multi-view images to take advantage of multi-view geometry. PlanarRecon [41] further constructs scene-level plane reconstruction using the trained normal and voting vector. In contrast to these approaches, we can obtain super-planes in an unsupervised context, and reconstruction models supervised using our super-plane surpass that using the predicted plane.

3. Proposed Method

Given posed multi-view images of an indoor scene, we aim to reconstruct the high-quality surfaces. We represent the scene geometry via a signed distance field (SDF), which is supervised by its multi-view images through volume rendering (Sec. 3.1). We propose an unsupervised super-plane segmentation method by exploiting the free geometric cues in the reconstructed surface. The segmentation results are further used as pseudo labels to regularize the reconstruction network during training (Sec. 3.2). To improve the self-supervised segmentation and stabilize the network training, we introduce two approaches to filter out mis-segmented and non-plane edge regions (Sec. 3.3). We discuss the training details in Sec. 3.4. Furthermore, we extend our method for unsupervised plane reconstruction, which can be used for auto-labeling or reconstruction of the plane in unseen new environments (Sec. 3.5). An overview of our approach is shown in Fig. 2.

3.1. Preliminaries

Similar to [39, 46], we represent the geometry of a scene as an SDF via MLP. An MLP g_θ maps a 3D point $\mathbf{x} \in \mathbb{R}^3$ to a signed distance to its closest surface:

$$\hat{s} = g_\theta(\mathbf{x}), \quad \hat{s} \in \mathbb{R}, \tag{1}$$

where $\hat{s}$ denotes the predicted SDF value. The surface can be further defined as the zero level-set of the SDF: $\mathcal{S}_\theta = \{\boldsymbol{x} \in \mathbb{R}^3 \mid g_\theta(\boldsymbol{x}; \theta) = 0\}$. Aside from 3D geometry, we predict continues radiance function to facilitate photometric-based reconstruction. Another MLP is used to predict radiance value $\hat{\mathbf{c}}$:

$$\hat{\mathbf{c}} = f_\phi(\mathbf{x}, \mathbf{v}, \hat{\mathbf{n}}, \hat{\mathbf{z}}), \quad \hat{c} \in \mathbb{R}^3 \tag{2}$$

where $\mathbf{x}$ and $\mathbf{v}$ are the input 3D point and its corresponding view direction. The normal direction $\hat{\mathbf{n}}$ is computed by the gradient of the SDF g_θ at point $\mathbf{x}$. The feature vector $\hat{\mathbf{z}}$ corresponds to the output geometric feature of g_θ as in [47].

A differentiable volume rendering technique [25] is adopted for training the geometry network. For each image pixel to be rendered, we first sample N points $\{\mathbf{x}_i\}$ along its camera ray $\mathbf{r}$ starting from the camera center $\mathbf{o}$ and pointing to $\mathbf{v}$, namely $\mathbf{x}_i = \mathbf{o} + d_i\mathbf{v}$. Then, the signed distance and color value of $\mathbf{x}_i$ are predicted by Eq. 1 and 2 separately. The color $\hat{\mathbf{C}}(\mathbf{r})$ of ray $\mathbf{r}$ is accumulated by:

$$\hat{\mathbf{C}}(\mathbf{r}) = \sum_{i=1}^{N} T_i \alpha_i \hat{\mathbf{c}}_i, \quad \alpha_i = \left(1 - \exp\left(-\sigma_i \delta_i\right)\right). \tag{3}$$

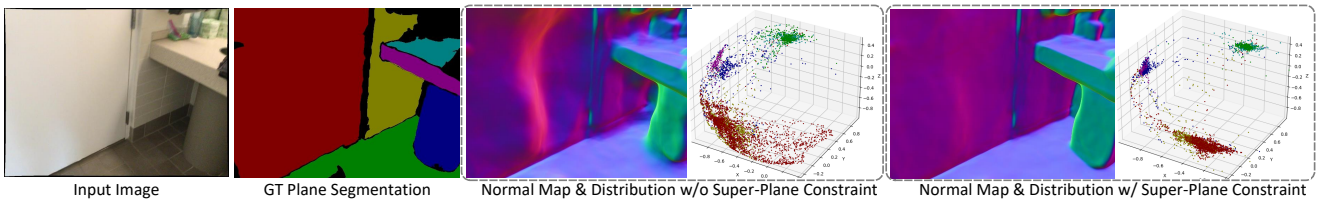


Figure 3. **Motivation of Our Unsupervised Super-plane Segmentation.** The normals from parallel planes are essentially grouped together and can therefore be used to generate super-planes. After applying the super-plane constraint, normals of different super-planes are more separable, which means they are pushed closer. The points with different colors represent normals from different planes.

Here, δ is the volume density transformed from the predicted sign distance value $\hat{s}$ [46], α_i and $T_i = \exp\left(-\sum_{j=1}^{i-1}\sigma_j\delta_j\right)$ represent the alpha value and the accumulated transmittance respectively, and $\delta_i = \|\mathbf{x}_{i+1} - \mathbf{x}_i\|_2$ is the distance between adjacent points. Similarly, we can also obtain the surface depth $\hat{\mathbf{D}}(\mathbf{r})$ and surface normal $\hat{\mathbf{N}}(\mathbf{r})$ corresponding to ray $\mathbf{r}$ as:

$$\hat{D}(\mathbf{r}) = \sum_{i=1}^{N} T_i\alpha_i\hat{d}_i, \quad \hat{\mathbf{N}}(\mathbf{r}) = \sum_{i=1}^{N} T_i\alpha_i\hat{\mathbf{n}}_i \tag{4}$$

During training, we randomly sample a batch of pixels in each iteration and construct the camera ray set $\mathcal{R}$, in which each $\mathbf{r} \in \mathcal{R}$ goes through the corresponding image pixel and calculate $\hat{\mathbf{C}}(\mathbf{r})$, $\hat{D}(\mathbf{r})$, and $\hat{\mathbf{N}}(\mathbf{r})$ using Eq. 3 and 4.

3.2. Self-supervised Super-plane Constraint

As aforementioned, the existing neural scene reconstruction methods [39, 46] are not effective for recovering structures in texture-less planar regions. To deal with this problem, we estimate the planar normal and force normal directions within the same plane to be consistent with it, thereby guiding the reconstruction of these regions. However, how to obtain the plane segmentation is challenging, and a naive solution may be to use ground truth plane segmentation masks or the results predicted by a plane reconstruction network [18, 19]. Such approaches have two main limitations: 1) training the network requires large-scale 3D annotation, *i.e.*, the planes, and 2) as described in Sec. 3.1, due to memory limitations, only a small fraction of pixels can be sampled in each iteration compared to the total number of pixels present in each image, while small-sized planes widely exist. Therefore, small planes do not receive sufficient supervision, and the estimated plane normals are noisy.

In contrast, we propose an unsupervised super-plane segmentation method that groups pixels belonging to parallel planes into the same cluster by exploiting free surface normal information. Thus, no labeled data or pretrained networks, are required and the super-plane structure increases the probability of sampling points in the same cluster at each iteration.

Super-plane Segmentation via Grouping. Our goal is to construct super-planes, and the surface normal $\hat{\mathbf{N}}(\mathbf{r}) \in \mathbb{R}^3$ provides a suitable initial source of super-plane segmentation. As shown in Fig. 3, despite some noise, points belonging to parallel planes generally have much more similar normal directions compared with others. This motivates us to treat the normal as a super-plane embedding vector, which has greater similarity within parallel planes. Since the total number of planes is not determined at the time of grouping, the K-means clustering is not applicable, and we instead adopt the Mean-shift clustering method [49] to obtain the super-plane segmentation from the normal maps. We denote each super-plane mask as $\mathcal{M}_i, i \in [1, K]$, and K is the total number of clusters.

Super-plane Constraint. With the obtained super-planes, we estimate the super-plane normal $\overline{\mathbf{N}}_i$ by averaging normals of pixels belonging to the same super-plane. The averaged normals filter out noise in the initial individual surface normals and capture the global structure of the super-planes. By enforcing all normals in the same super-plane region to be the same as $\overline{\mathbf{N}}_i$, the accurate planar geometry can be recovered. The super-plane loss is defined as:

$$\mathcal{L}_{\text{plane}} = \sum_{i=1}^{K}\sum_{\mathbf{r}\in\mathcal{R}} \|1 - \overline{\mathbf{N}}_i\hat{\mathbf{N}}(\mathbf{r})\|_1. \tag{5}$$

During training, we detach the gradients of $\overline{\mathbf{N}}_i$ to ease the burden of the optimization process.

Iterative Optimization. We apply the super-plane normal estimation and surface structure optimization in an iterative manner. To be specific, we re-rendering normal maps in every t iterations and also update the super-plane segmentation and the super-plane normal estimation. Note that the first t iterations are used to initialize the geometric structure and the super-plane loss in Eq. 5 will not be added. Through this training process, better reconstruction results bring better segmentation and vice versa. The overall optimization process can be found in the supplementary material.

Accelerating rendering via Sphere Tracing. In Sec. 3.1, we introduce the volume rendering used during training. However, it is computationally expensive to render normal maps of all training images. To solve this problem, we apply sphere tracing [10] to approach surface points $\mathbf{x}_{suf}$ along each ray $\mathbf{r}$ and the corresponding surface normal is used to perform clustering. The detailed algorithm of sphere tracing

can be found in the supplementary material. Sphere Tracing is only used for generating normal maps for plane segmentation, and we still adopt volume rendering during training since it handles occlusions better [39].

3.3. Super-plane Segmentation Refinement

Auto-filtering for Plane Segmentation. The core idea of our super-plane constraint is to enforce a global constraint for the pixels belonging to the same super-plane. However, since the unsupervised super-plane segmentation might include some noisy segmentation results, which can be non-plane regions or wrongly grouped pixels, we propose a self-guided auto-filtering strategy to filter out these outliers. Specifically, we compare the normal of all pixels in the super-plane with the averaged normal $\overline{\mathbf{N}}_i$ mentioned in Sec. 3.2, and all pixels with normal direction differences from $\mathbf{N}_i$ larger than α are considered outliers, which will not be used for estimate super-plane normal and calculate the super-plane loss in Eq. 5.

Non-plane Edge Region Detection. In addition to the auto-filtering strategy, we also directly mask out non-plane edge regions. Typically, these regions will be more color and geometry discontinuous. As such, we use this property to detect object edges and some non-plane regions [16]. Specifically, for each pixel p, we calculate the geometry discontinuity as:

$$d_n(\mathbf{p}) = [\sum_{\mathbf{q}\in\mathcal{Q}} \left\|\overline{\mathbf{N}}(\mathbf{p}) - \overline{\mathbf{N}}(\mathbf{q})\right\|_2], \tag{6}$$

where $\mathcal{Q}$ is the 3×3 adjacent regions of p and $[\cdot]$ is the normalization operation. Similarly, the color discontinuity is defined as:

$$d_c(\mathbf{p}) = [\sum_{\mathbf{q}\in\mathcal{Q}} \|\mathbf{C}(\mathbf{p}) - \mathbf{C}(\mathbf{q})\|_2], \tag{7}$$

where $\mathbf{C}$ is the ground-truth pixel color value. The final discontinuity $d(\mathbf{p})$ is then defined as the maximum value of $d_n(\mathbf{p})$ and $d_c(\mathbf{p})$. Pixels with discontinuity $d(\mathbf{p}) > \gamma$ are seen as non-plane edge regions, which will not be used to calculate the super-plane loss in Eq. 5. Note that some textured planar regions (*e.g.*, painted) may also be detected and masked out in this step. However, our method aims to better reconstruct the untextured planar regions, while the textured regions can already be reconstructed by existing schemes and therefore have little impact on the results.

3.4. Training Objectives

In addition to the super-plane constraint loss in Eq. 5, the following losses are applied during training for each ray $\mathbf{r} \in \mathcal{R}$ described in Sec, 3.1.

Color Loss. The link between the scene geometry is built in Eq. 2 and 3, thus the scene reconstruction process can be supervised by the following color loss:

$$\mathcal{L}_{\text{color}} = \sum_{\mathbf{r}\in\mathcal{R}} \|\hat{\mathbf{C}}(\mathbf{r}) - \mathbf{C}(\mathbf{r})\|_1, \tag{8}$$

where $\mathbf{C}(\mathbf{r})$ is the ground-truth pixel color value.

Eikonal Loss. As suggested by [39,46,47], Eikonal loss [7] is added to regularize the SDF:

$$\mathcal{L}_{\text{eik}} = \sum_{\mathbf{y}\in\mathcal{Y}} \left(\|\nabla g_\theta(\mathbf{y})\|_2 - 1\right)^2, \tag{9}$$

where $\mathcal{Y}$ is a concatenated set of uniformly sampled points and surface points.

Depth Loss. Depth supervision has been proven to be beneficial for the geometric representation [9, 40]. In addition, sparse depth maps are generated as a side effect during camera calibration using COLMAP [29]. Therefore, we also add sparse depth supervision:

$$\mathcal{L}_{\text{d}} = \sum_{\mathbf{r}\in\overline{\mathcal{R}}} |\hat{R}(\mathbf{r}) - D(\mathbf{r})|_1, \tag{10}$$

where $\overline{\mathcal{R}}$ represents the subset of rays which have valid sparse depth $D(\mathbf{r})$ produced by COLMAP [29].

The overall loss for the proposed model is defined by

$$\mathcal{L} = \mathcal{L}_{\text{color}} + \lambda_{\text{eik}}\mathcal{L}_{\text{eik}} + \lambda_{\text{d}}\mathcal{L}_{\text{d}} + \lambda_{\text{plane}}\mathcal{L}_{\text{plane}}, \tag{11}$$

where λ_{eik}, λ_{d}, and λ_{plane} are hyperparameters for weighting each loss term.

3.5. Plane Reconstruction

Although our super-plane fits the training demand well, further process is needed to obtain desired plane segmentation results. To do so, we directly apply the non-plane edge region mask mentioned in Sec. 3.3 to separate the super-planes $\mathcal{M}_i$ into continuous planes. Specifically, each connected component in the super-plane mask separated by the non-plane edge region mask is considered a plane.

4. Experiments

4.1. Implementation Details

Architecture. Our method is implemented in Python using PyTorch and trained with the Adam optimizer [13] with an initial learning rate of $5e^{-4}$. We train 50k iterations for each scene and randomly sample 1024 rays per iteration, and we adopt the error-bounded sampling strategy [46] to sample points along each ray. During training, images are resized to 640×480 pixels, and all cameras are normalized to fit inside a unit sphere [1]. The Marching Cubes algorithm [23] is used to extract mesh from the learned SDF. More details on network architecture and the training process are presented in the supplementary material.

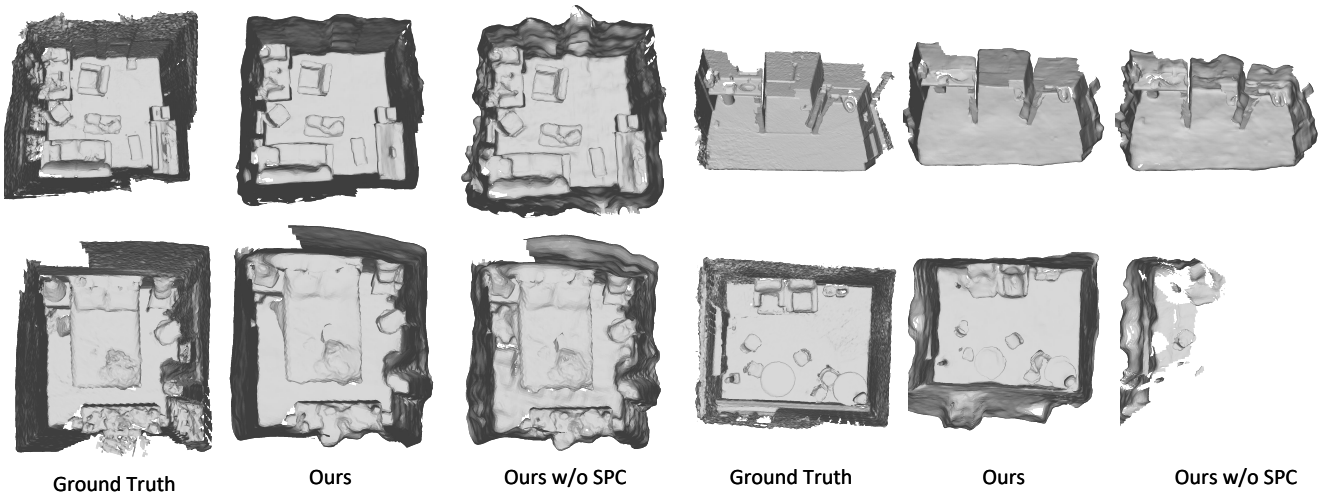


Figure 4. **Ablation of Super-plane Constraint.** By adding the super-plane constraint, the reconstruction quality on large-scale plane regions is significantly improved without any additional supervision. "SPC." represents the super-plane constraint. Zoom in for details.

#Num	Plane Source	Plane Constraint	Super-plane	Filter	Mask	Acc↓	Comp↓	Prec↑	Recall↑	F-score↑
1	×					0.073	0.071	0.604	0.575	0.589
2	GT	✓				0.060	0.062	0.687	0.632	0.658
3	Un	✓	✓			0.060	0.063	0.676	0.619	0.646
4	Un	✓	✓	✓		0.058	0.060	0.693	0.645	0.668
5	Un	✓	✓		✓	0.060	0.062	0.686	0.627	0.655
6	Un	✓		✓	✓	0.059	0.062	0.690	0.641	0.664
7	Un	✓	✓	✓	✓	**0.055**	**0.059**	**0.709**	**0.660**	**0.683**

Table 2. **Ablation studies on ScanNet.** Our method improves the performance notably and shows better performance compared with using ground truth plane segmentation masks. Here, "GT" and "Un" denote ground truth plane segmentation and our unsupervised plane segmentation, respectively.

Datasets. We evaluate our method on Scannet [5] and 7-Scenes [32]. Scannet is a large-scale RGB-D dataset containing 1613 indoor scenes with ground-truth camera parameters and surface reconstructions. 7-Scenes is a collection of RGB-D frames whose camera tracks and dense 3D models are obtained with KinectFusion [27]. We use the eight randomly selected scenes (four from Scannet validation set and four from 7-Scenes) to perform the experiments as in [9]. In each scene, one-tenth of views are uniformly sampled for reconstruction, and we do not use any additional implicit or explicit supervision during training.

Baselines. We evaluate our 3D reconstruction model against 1) MVS methods: COLMAP [29] and variants with the plane fitting [9] (denoted as COLMAP*). 2) MVS method with plane regularization: ACMP [42]. 3) Vanilla neural volume rendering methods: NeRF [25], UNISURF [28], NeuS [39] and VolSDF [46]. 4) Neural volume rendering methods with explicit supervision: ManhattanSDF [9], NeuRIS [37], and MonoSDF [48]. For the plane reconstruction task, we compare with the state-of-the-art supervised Plane R-CNN [18] method.

Evaluation Metrics. For the 3D reconstruction task, we evaluate five standard metrics following [26]: accuracy, completeness, precision, recall, and F-score. Among these, F-score is considered as the main metric following [34]. For the plane segmentation task, plane and pixel recalls are used as evaluation metrics following [18, 19], different depth thresholds over the overlapping region vary from 0.05m to 0.6m are evaluated. Detailed definitions of these metrics can be found in the supplementary material.

4.2. Ablation Study and Analysis

To illustrate the effectiveness of each design in S^{3}P, we conduct ablation studies on ScanNet. We train seven variations: (1) VolSDF* with only image and sparse depth supervision, and this is treated as our baseline, (2) VolSDF* with plane loss (ground truth plane segmentation), (3) VolSDF* with super-plane loss, (4) add auto-filtering strategy to the super-plane loss, (5) add non-plane masks to the super-plane loss, (6) use plane loss instead of super-plane, (7) our full framework. All results are shown in Tab. 2.

Overall Super-plane Constraint. We first demonstrate the effectiveness of our overall super-plane constraint by showing qualitative results in Fig. 4 and quantitative results in Tab. 2. Comparing #1 and #7 in Tab. 2 shows that the proposed super-plane constraint brings about 0.094 F-score im-

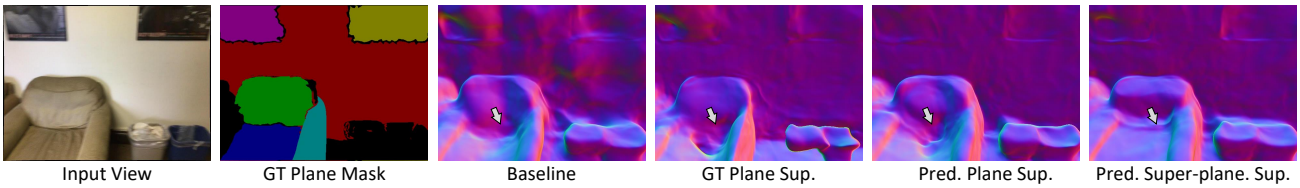


Figure 5. **Effectiveness of Super-plane Structure.** Introducing of super-plane structure can provide more ample and stable constraints. Therefore, the proposed super-plane constraint better recovers the structure in small plane regions compared with ground truth plane segmentation-assisted plane constraint.

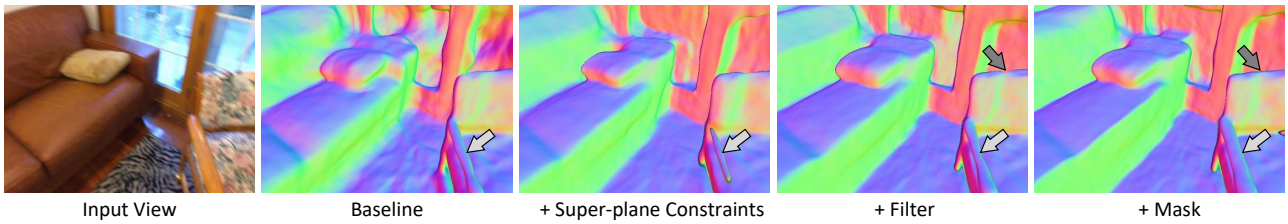


Figure 6. **Reconstruction of thin edge regions.** By adding the super-plane constraint, the reconstruction quality of planar regions is significantly improved but some thin edge regions disappeared due to noisy segmentation. Our auto-filter and edge detection mechanism can eliminate this negative effect.

provements. Fig. 4 shows that after adding the proposed super-plane supervision, the reconstruction quality on both large-scale texture-less regions (*e.g.*, wall, floor) and small plane regions (*e.g.*, tables, sofa) can be largely improved since ambiguities caused by the lack of texture can be resolved. In addition, our method can complement some incomplete reconstructed regions of the baseline model, as there are additional cues available. It is worth noting that these are cues at our disposal that do not require additional labeled data or trained networks.

Super-plane vs. Plane Segmentation. We then analyze how the introduced super-plane structure affects the reconstruction performance. Fig. 5 shows that a smoother surface can be reconstructed in large planes (walls) using either ground truth plane segmentation, predicted plane segmentation, or the constraints provided by our super-plane segmentation compared to the VolSDF* baseline. However, simply using ground truth plane segmentation or predicted plane segmentation does not construct small planes well (sofa as indicated by arrows). These results show that simply taking ground truth plane segmentation to provide plane constraint is insufficient for the volume rendering-based optimization since only a small portion of pixels is sampled in each iteration. Thus, the sampled pixels may not be sufficient to recover the plane structure. In contrast, our super-plane constraint provides more supervision by grouping parallel planes into the same cluster. Tab. 2 shows that using super-plane constraint gives better performance compared with ground truth or predicted plane segmentation.

Noise Reduction. We demonstrate that the adopted non-plane edge region detection and self-filtering mechanism can filter out noisy non-plane regions in the super-plane segmentation results. Fig. 6 shows VolSDF* can faithfully reconstruct the thin structure in the edge regions (chair armrests indicated by light grey arrows) but shows noisy reconstruction results in planar regions like floor and window. Directly adding our super-plane constraint reconstructs these planar regions better but, unfortunately, makes the results worse for the non-planar regions at these edges, as the segmentation results are far from perfect. Adapting the self-guided filtering strategy automatically filters out wrongly segmented pixels and thus can keep most of the fine details. Then, masking out the detected edge regions further improves the reconstruction quality of edges (as indicated by dark grey arrows). Tab. 2 also shows that both the non-plane region detection and self-filtering mechanism improve the performance over only adding the super-plane constraint, and adding both of them gives the best performance.

4.3. Comparison with State-of-the-arts

3D Reconstruction. Tab. 3 shows the quantitative results compared with other methods on Scannet [5] and 7-Scenes [32]. The first four methods in Tab. 3 adopt assistive networks to provide additional priors, while the remaining methods do not need. The results show that our method significantly outperforms other assistant network-free MVS and volume rendering methods. In addition, our model performs better than the method requiring a segmentation network trained on annotated 2D datasets (ManhattanSDF [9]) and on par with the scheme requiring a normal network trained on 3D datasets (NeuRIS [37]).

Fig. 7 shows qualitative scene-level reconstruction results of evaluated methods. Both the MVS-based method (COLMAP [29]) and the vanilla volume rendering-based method (VolSDF [46]) do not reconstruct complete or smooth planar structures. ManhattenSDF [9] achieves compelling results by introducing an assistive segmentation network to find floors and walls, then applying Manhattan as-

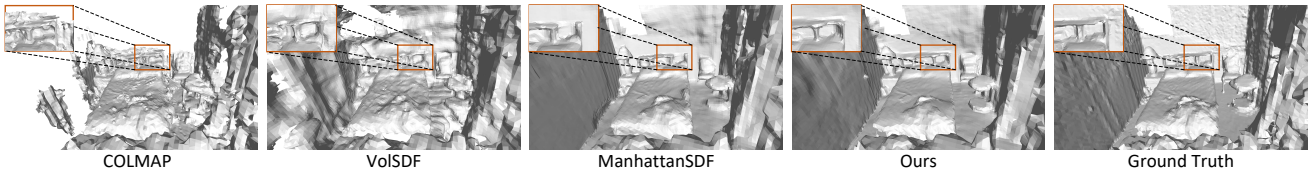
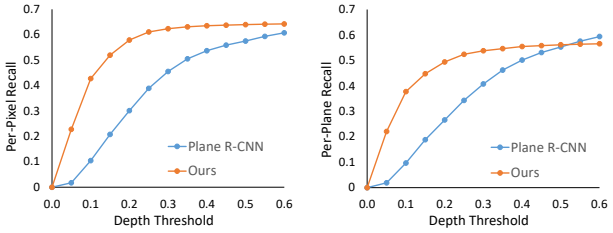


Figure 7. **Qualitative reconstruction comparisons.** Our method can construct smooth and complete structures in both large and small planar regions compared with others. Zoom in for details.

Method	ScanNet					7-Scenes				
	Acc↓	Comp↓	Prec↑	Recall↑	**F-score↑**	Acc↓	Comp↓	Prec↑	Recall↑	**F-score↑**
COLMAP*	0.396	0.081	0.271	0.595	0.368	0.670	0.215	0.116	0.215	0.149
ManhattanSDF	0.072	0.068	0.621	0.586	0.602	0.112	0.133	0.351	0.326	0.336
NeuRIS	0.050	0.049	0.717	0.669	0.692	-	-	-	-	-
MonoSDF	0.035	0.048	0.799	0.681	0.733	-	-	-	-	-
COLMAP	**0.047**	0.235	**0.711**	0.441	0.537	**0.069**	0.417	**0.536**	0.202	0.289
NeRF	0.735	0.177	0.131	0.290	0.176	0.573	0.321	0.159	0.085	0.083
UNISURF	0.554	0.164	0.212	0.362	0.267	0.407	**0.136**	0.195	0.301	0.231
NeuS	0.179	0.208	0.313	0.275	0.291	0.151	0.247	0.313	0.229	0.262
VolSDF	0.414	0.120	0.321	0.394	0.346	0.285	0.140	0.220	0.285	0.246
Ours	0.055	**0.059**	0.709	**0.660**	**0.683**	0.108	0.147	0.493	**0.480**	**0.483**

Table 3. **Reconstruction Results Comparison.** Note that the first four methods require well-trained assistant networks to provide additional geometric priors. Our method achieves the best performance among all assistant network-free methods while being on par with methods using additional normal information.



(a) Per-Pixel Recall (b) Per-Plane Recall

Figure 8. **Plane Reconstruction Comparisons.** The unsupervised plane reconstruction performance are comparable with state-of-the-art supervised method Plane R-CNN [18].

sumption on these areas, but does not recover small planar regions. In contrast, thanks to the super-plane constraint, our approaches can reconstruct smooth and complete structures on both small and large planar regions without any assistant networks.

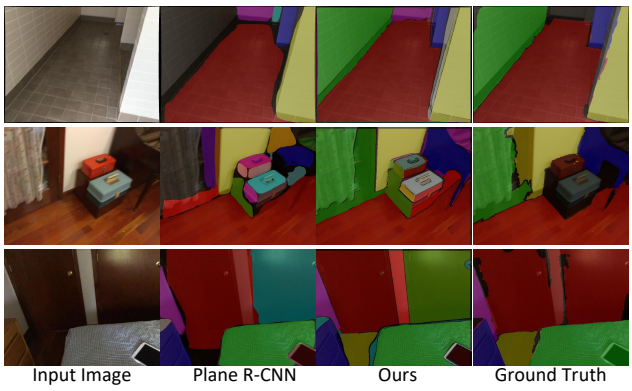


Figure 9. **Plane Segmentation Comparisons.** Our method can generate way more accurate segmentation results in edge regions compared with Plane R-CNN [18].

Plane Reconstruction. To demonstrate the capability of our method on the unsupervised plane reconstruction task, we compare its reconstruction performance with the state-of-the-art supervised Plane R-CNN [18] method. Quantitative results in Fig. 8 show that our method outperforms the Plane R-CNN in terms of per-pixel and per-plane recall when accurate depth prediction is required (low depth threshold). When using a loose depth threshold, we obtain competitive results in terms of per-plane recall and still outperform Plane R-CNN in per-pixel recall even without exposure to the ground truth plane segmentation during training. This quality of unsupervised plane segmentation results can also be analyzed in Fig. 9. Compared to the supervised baseline, we can obtain more accurate and complete segmentation results.

5. Conclusion

This work presents a novel neural scene reconstruction method based on the super-plane constraint. The key idea is to enforce all pixels in parallel planes to have the same normal orientation. We first group pixels belonging to parallel planes into the same cluster in an unsupervised manner, and then design a self-guided filtering and non-planar edge region detection strategy to filter out outliers. The remaining clean pixels are further used to compute super-plane normal and super-plane loss. Experimental results show that our method can reconstruct accurate and complete planar regions with missing texture information without additional implicit or explicit supervision. Furthermore, our method can be extended to obtain plane reconstruction results, which can be used to automatically label new scenes.

References

[1] Matan Atzmon and Yaron Lipman. Sal: Sign agnostic learning of shapes from raw data. In *CVPR*, 2020. 5

[2] Dejan Azinović, Ricardo Martin-Brualla, Dan B Goldman, Matthias Nießner, and Justus Thies. Neural rgb-d surface reconstruction. In *CVPR*, 2022. 2

[3] Olga Barinova, Vadim Konushin, Anton Yakubenko, KeeChang Lee, Hwasup Lim, and Anton Konushin. Fast automatic single-view 3-d reconstruction of urban scenes. In *ECCV*, 2008. 3

[4] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *SIGGRAPH*, 1996. 2

[5] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017. 3, 6, 7

[6] Erick Delage, Honglak Lee, and Andrew Y Ng. Automatic single-image 3d reconstructions of indoor manhattan world scenes. In *ISRR*, 2007. 3

[7] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. In *ICML*, 2020. 5

[8] Xiaodong Gu, Zhiwen Fan, Siyu Zhu, Zuozhuo Dai, Feitong Tan, and Ping Tan. Cascade cost volume for high-resolution multi-view stereo and stereo matching. In *CVPR*, 2020. 2

[9] Haoyu Guo, Sida Peng, Haotong Lin, Qianqian Wang, Guofeng Zhang, Hujun Bao, and Xiaowei Zhou. Neural 3d scene reconstruction with the manhattan-world assumption. In *CVPR*, 2022. 1, 2, 5, 6, 7

[10] John C Hart. Sphere tracing: A geometric method for the antialiased ray tracing of implicit surfaces. *The Vis. Comput.*, 1996. 4

[11] Linyi Jin, Shengyi Qian, Andrew Owens, and David F Fouhey. Planar surface reconstruction from sparse views. In *ICCV*, 2021. 3

[12] Michael M. Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *SGP*, 2006. 2

[13] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 5

[14] Uday Kusupati, Shuo Cheng, Rui Chen, and Hao Su. Normal assisted stereo depth estimation. In *CVPR*, 2020. 1, 2

[15] David C Lee, Martial Hebert, and Takeo Kanade. Geometric reasoning for single image structure recovery. In *CVPR*, 2009. 3

[16] Boying Li, Yuan Huang, Zeyu Liu, Danping Zou, and Wenxian Yu. Structdepth: Leveraging the structural regularities for self-supervised indoor depth estimation. In *CVPR*, 2021. 5

[17] Zhi-Hao Lin, Wei-Chiu Ma, Hao-Yu Hsu, Yu-Chiang Frank Wang, and Shenlong Wang. Neurmips: Neural mixture of planar experts for view synthesis. In *CVPR*, 2022. 3

[18] Chen Liu, Kihwan Kim, Jinwei Gu, Yasutaka Furukawa, and Jan Kautz. Planercnn: 3d plane detection and reconstruction from a single image. In *CVPR*, 2019. 3, 4, 6, 8

[19] Chen Liu, Jimei Yang, Duygu Ceylan, Ersin Yumer, and Yasutaka Furukawa. Planenet: Piece-wise planar reconstruction from a single rgb image. In *CVPR*, 2018. 3, 4, 6

[20] Jiachen Liu, Pan Ji, Nitin Bansal, Changjiang Cai, Qingan Yan, Xiaolei Huang, and Yi Xu. Planemvs: 3d plane reconstruction from multi-view stereo. In *CVPR*, 2022. 3

[21] Shaohui Liu, Yinda Zhang, Songyou Peng, Boxin Shi, Marc Pollefeys, and Zhaopeng Cui. Dist: Rendering deep implicit signed distance function with differentiable sphere tracing. In *CVPR*, 2020. 2

[22] Xiaoxiao Long, Lingjie Liu, Christian Theobalt, and Wenping Wang. Occlusion-aware depth estimation with adaptive normal constraints. In *ECCV*, 2020. 1, 2

[23] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *SIGGRAPH*, 1987. 5

[24] Paul C. Merrell, Amir Akbarzadeh, Liang Wang, Philippos Mordohai, Jan-Michael Frahm, Ruigang Yang, David Nistér, and Marc Pollefeys. Real-time visibility-based fusion of depth maps. In *ICCV*, 2007. 2

[25] 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 *ECCV*, 2020. 2, 3, 6

[26] Zak Murez, Tarrence van As, James Bartolozzi, Ayan Sinha, Vijay Badrinarayanan, and Andrew Rabinovich. Atlas: End-to-end 3d scene reconstruction from posed images. In *ECCV*, 2020. 1, 2, 6

[27] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *ISMAR*, 2011. 6

[28] Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *ICCV*, 2021. 6

[29] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 2, 5, 6, 7

[30] Johannes L. Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *ECCV*, 2016. 1

[31] Shuhan Shen. Accurate multiple view 3d reconstruction using patch-based stereo for large-scale scenes. *IEEE TIP*, 2013. 1, 2

[32] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *CVPR*, 2013. 6, 7

[33] Sudipta Sinha, Drew Steedly, and Rick Szeliski. Piecewise planar stereo for image-based rendering. In *ICCV*, 2009. 3

[34] Jiaming Sun, Yiming Xie, Linghao Chen, Xiaowei Zhou, and Hujun Bao. Neuralrecon: Real-time coherent 3d reconstruction from monocular video. In *CVPR*, 2021. 1, 2, 6

[35] Fangjinhua Wang, S. Galliani, Christoph Vogel, and Marc Pollefeys. Itermvs: Iterative probability estimation for efficient multi-view stereo. In *CVPR*, 2022. 2

[36] Jingwen Wang, Tymoteusz Bleja, and Lourdes de Agapito. Go-surf: Neural feature grid optimization for fast, high-fidelity rgb-d surface reconstruction. *ArXiv*, abs/2206.14735, 2022. 2

[37] Jiepeng Wang, Peng Wang, Xiaoxiao Long, Christian Theobalt, Taku Komura, Lingjie Liu, and Wenping Wang. Neuris: Neural reconstruction of indoor scenes using normal priors. In *ECCV*, 2022. 1, 2, 3, 6, 7

[38] Kaixuan Wang and Shaojie Shen. Mvdepthnet: Real-time multiview depth estimation neural network. In *3DV*, 2018. 1

[39] 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. In *NeurIPS*, 2021. 1, 2, 3, 4, 5, 6

[40] Yi Wei, Shaohui Liu, Yongming Rao, Wang Zhao, Jiwen Lu, and Jie Zhou. Nerfingmvs: Guided optimization of neural radiance fields for indoor multi-view stereo. In *CVPR*, 2021. 5

[41] Yiming Xie, Matheus Gadelha, Fengting Yang, Xiaowei Zhou, and Huaizu Jiang. Planarrecon: Real-time 3d plane detection and reconstruction from posed monocular videos. In *CVPR*, 2022. 3

[42] Qingshan Xu and Wenbing Tao. Planar prior assisted patchmatch multi-view stereo. In *AAAI*, 2020. 6

[43] Fengting Yang and Zihan Zhou. Recovering 3d planes from a single image via convolutional neural networks. In *ECCV*, 2018. 3

[44] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *ECCV*, 2018. 2

[45] Yao Yao, Zixin Luo, Shiwei Li, Tianwei Shen, Tian Fang, and Long Quan. Recurrent mvsnet for high-resolution multi-view stereo depth inference. In *CVPR*, 2019. 2

[46] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. In *NeurIPS*, 2021. 1, 2, 3, 4, 5, 6, 7

[47] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. In *NeurIPS*, 2020. 1, 2, 3, 5

[48] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. In *NeurIPS*, 2022. 1, 2, 3, 6

[49] Zehao Yu, Jia Zheng, Dongze Lian, Zihan Zhou, and Shenghua Gao. Single-image piece-wise planar 3d reconstruction via associative embedding. In *CVPR*, 2019. 3, 4

[50] Enliang Zheng, Enrique Dunn, V. Jojic, and Jan-Michael Frahm. Patchmatch based joint view selection and depthmap estimation. In *CVPR*, 2014. 1, 2