Pseudo-plane Regularized Signed Distance Field for Neural Indoor Scene Reconstruction

Jinpeng Yu[1,2†], Jing Li[1,2†], Ruoyu Wang[1], Shenghua Gao[1*]

[1*]School of Information Science and Technology, ShanghaiTech University, 393 Middle Huaxia Road, Pudong District, Shanghai, 201210, China.
[2] Xiaohongshu Technology Incorporated Company, 388 Madang Road, Huangpu District, 201210, Shanghai, China.

*Corresponding author(s). E-mail(s): gaoshh@shanghaitech.edu.cn;
Contributing authors: yujp1@shanghaitech.edu.cn; lijing1@alumni.shanghaitech.edu.cn; wangry3@shanghaitech.edu.cn;
†These authors contributed equally to this work.

Abstract

Given only a set of images, neural implicit surface representation has shown its capability in 3D surface reconstruction. However, as the nature of per-scene optimization is based on the volumetric rendering of color, previous neural implicit surface reconstruction methods usually fail in the low-textured regions, including floors, walls, etc., which commonly exist for indoor scenes. Being aware of the fact that these low-textured regions usually correspond to planes, without introducing additional ground-truth supervisory signals or making additional assumptions about the room layout, we propose to leverage a novel Pseudo-plane regularized Signed Distance Field (PPlaneSDF) for indoor scene reconstruction. Specifically, we consider adjacent pixels with similar colors to be on the same pseudo-planes. The plane parameters are then estimated on the fly during training by an efficient and effective two-step scheme. Then the signed distances of the points on the planes are regularized by the estimated plane parameters in the training phase. As the unsupervised plane segments are usually noisy and inaccurate, we propose to assign different weights to the sampled points on the plane in plane estimation as well as the regularization loss. The weights come by fusing the plane segments from different views. As the sampled rays in the planar regions are redundant, leading to inefficient training, we further propose a keypoint-guided rays sampling strategy that attends to the informative textured regions with large color variations, and the implicit network gets a better reconstruction, compared with the original uniform ray sampling strategy. Experiments show that our PPlaneSDF achieves competitive reconstruction performance in Manhattan scenes. Further, as we do not introduce any additional room layout assumption, our PPlaneSDF generalizes well to the reconstruction of non-Manhattan scenes.

Keywords: Scene Reconstruction, Neural Surface Reconstruction, Implicit Representation, Plane Regularized Reconstruction

1 Introduction

Reconstructing the 3D geometry of an indoor scene is an appealing and valuable task in computer vision because of its potential applications in computer graphics, virtual reality, etc. As the acquisition of 3D ground truth is expensive, it is promising to reconstruct the 3D scene directly from multiple images without 3D ground truth. Recently, the neural radiance field (NeRF) [33] has achieved promising results in novel view synthesis based on implicit neural representations and volumetric rendering, giving a set of images. However, NeRF does not explicitly regularize the geometry of the scene, resulting in poor reconstruction. To achieve a better geometry estimation, recent methods [37, 50, 56] propose to model the density from the signed distance field instead of directly using the volume density in NeRF. However, indoor scenes have many textureless areas and repetitive patterns. It is difficult for previous methods to optimize the implicit network based on the rendering loss for these regions. As a result, the surface reconstruction of these methods at those planar regions is usually bumpy and noisy. It is necessary to incorporate additional constraints to get better reconstruction results.

To improve the reconstruction for low-textured regions for indoor scenes, Manhattan-SDF [18] proposes to use a pre-trained semantic segmentation network to find the floor and wall regions, then regularize the geometry of these regions. The normal directions of floors and walls are regularized to be aligned with three dominant directions under Manhattan world assumption [9]. However, there are still many scenes with more than three dominant directions, which are the non-Manhattan scenes. Manhattan-SDF is not applicable to the non-Manhattan rooms. It may also perform poorly on scenes cluttered by objects with small visible areas of floors and walls, and its result is restricted by the segmentation network whose performance may degrade on scenes out of the pre-trained domain. In this work, we propose to tackle neural surface reconstruction of indoor scenes by using the pseudo-planes as a regularization. To generate plane segments, we cluster the adjacent pixels with similar colors by super-pixels [13] and consider these regions as pseudo-planes. The planes are not only large floor and wall regions but also small plane regions on objects, e.g., the chair and the piano exterior, as shown in Fig. 1. We then estimate the plane parameters on the fly during training in an effective and efficient two-step scheme and regularize the signed distance field (SDF) with these planes. As we do not make the Manhattan assumption on the indoor scene, our method generalizes well to both Manhattan and non-Manhattan rooms.

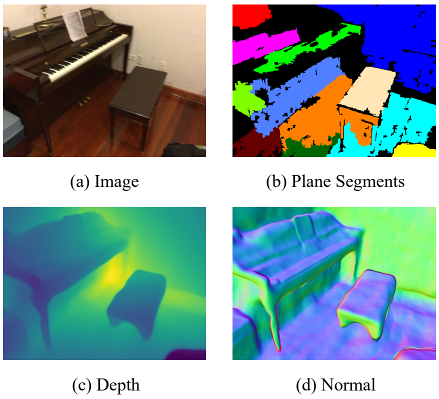


Fig. 1 (a) is the image, (b) is the generated plane segments by super-pixel, (c) is the depth map, and (d) is the normal map from the reconstruction of PPlaneSDF. It can be seen that not only large areas of floor and wall regions but also small planar regions on objects, e.g., the chair and the piano, are segmented and reconstructed to be planar.

To be specific, we first sample only a few points on every plane and get their depth through the accumulated rendering. By back-projecting these points with the estimated depth, we fit the roughly estimated planes with these points by a least square method. The parameters of the roughly estimated plane are noisy and inaccurate as the number of sampled points is limited by the memory in the slow rendering. In the second step, we utilize the roughly estimated plane parameters to get the depth of a new set of sampled points and back-project them to the world space. The coordinates of these points are then fed into the MLPs to get the signed distances and the gradients of signed distances, i.e., normal directions. We assume that the back-projected points on the roughly estimated planes are close enough to the planar surface, therefore the signed distances are also the distance to these planes. We march these points according to the signed distances and the normal directions to get the coordinates of the points on the rectified planes, and the parameters of the rectified plane are estimated with these points. In this step, we avoid using the depth with the expensive accumulated rendering and directly get the signed distances by querying points to get the rectified coordinates. Compared with the first step, this process is much faster, therefore the number of sampled points is much larger than that in the first step. After getting the rectified plane parameters, we regularize the signed distances of the newly sampled points on the roughly estimated planes by their distances to the rectified planes.

As the unsupervised plane segments are often noisy and inaccurate, we propose to assign different weights to the points during the plane fitting and the plane regularization loss. The weights are obtained by fusing plane segments from different views. We conduct the fusion by predicting the segmentation probabilities and using the probabilities as the weights. The segmentation prediction is supervised by the super-pixel segments from different views, therefore considering the cross-view consistency implicitly. Existing MLP in VolSDF [56] is queried by 3D coordinates to get the density and color. We introduce a new segmentation network branch to the MLP. The branch is supervised in the volume rendering by matching the accumulated prediction of the network, and the generated plane segments following [4, 49]. During the fusion process, the segmentation network predicts lower probabilities for those inconsistent areas among different views and higher probabilities for those consistent areas among different views. We utilize the probabilities as the weights assigned to different points during plane estimation and regularization.

Further, we observe that the random ray sampling strategy which samples rays in volumetric rendering randomly is inefficient for indoor scenes. The reason is that the planar regions usually occupy large areas and the rays in the texture-less indoor planar regions are easy to be sampled in random sampling. Their colors are similar in the planar regions, therefore the RGB rendering loss for the sampled ray in the planar region is usually close to 0 when there is still a large reconstruction error and contributes less to the optimization. To avoid such redundant sampling, we suppose that the ray sampling strategy should attend to the more informative textured regions with large color variations. We use Direct Sparse Odometry [12] to extract keypoints for the color image. We then treat the keypoints and the pixels around them as the informative regions and increase the sampling probability of the rays in these regions, therefore making them easier to be sampled. By our proposed keypoint guided rays sampling strategy, the implicit network gets a better reconstruction.

The main contributions of this paper can be summarized as follows. i) We propose to constrain indoor neural surface reconstruction with pseudo-planes, without imposing any assumption on the room layout or introducing additional supervision from external annotation on geometry. Experiments show that our method achieves competitive performance on Manhattan scenes and generalizes well to non-Manhattan scenes. ii) Under the memory and time restriction, we design an effective and efficient two-step strategy to estimate plane parameters, which are then used to regularize the signed distance field. iii) We introduce the weighting schedule on points during plane estimation and regularization. The weights are obtained by fusing the plane segments among views through volumetric rendering. iv) To avoid redundant ray sampling in the planar regions, we propose the keypoint guided sampling strategy to sample rays more efficiently and get a better performance. In this ray sampling strategy, the sampling probability in the informative textured regions is increased, therefore the rays are easier to be sampled. Our method also enables the potential to represent the scenes with the vectorized pseudo planes. The parameters of the vectorized planes can also be calculated efficiently with our method.

The rest of this paper is organized as follows: In Sec. 2, we introduce the work related to the multi-view stereo and the neural scene reconstruction. In Sec. 3, we detail our proposed PPlaneSDF, including pseudo-plane regularization, plane segments fusion based points weighting, and keypoint guided ray sampling strategy. In Sec. 4, extensive experiments are conducted to validate the effectiveness and generalization of our method on the Manhattan scenes and the non-Manhattan scenes. We conclude our work in Sec. 5.

2 Related work

Multi-view stereo. Traditional 3D reconstruction methods from multi-view images [2, 31, 35, 42, 43, 45] first estimate depth map for each view and then fuse the predicted depth from different views to obtain the final reconstruction result.

However, these methods struggle with texture-less areas such as floor and wall regions in indoor scenes due to the mismatching of these regions by the hand-crafted features. To tackle this problem, some depth-based methods [6, 19, 54, 55] incorporate deep learning to extract global features, which is more robust for matching. After extracting global features, cost volumes are built and then fed to 3D CNNs to get final depth maps, and then leverage the extra fusion process to reconstruct the scene. Others [24, 58] enforce the depth-normal consistency in the training process. Such methods usually reconstruct noisy and incomplete surfaces because of the inconsistency of individual predicted depth maps. Many methods also incorporate planar prior in MVS to achieve better performance on the texture-less regions. [17, 52] utilize the second-order priors on the smoothness of 3D surfaces as planar priors to constrain the planar areas in the scene. However, it leads to a triple cliques inference, making the optimization very difficult. [15] chooses to train a planar classifier to distinguish the planar and non-planar areas, then links and fuses plane hypotheses across multiple overlapping views. The non-planar regions are reconstructed with the results of a standard multi-view stereo algorithm. Besides, some methods [16, 53] leverage the triangulation to build piece-wise planar regions. [16] choose to embed the priors into multi-view stereo directly over the disparity space. [53] uses the planar priors with PatchMatch MVS to focus on the geometric constraints of different views. Although PatchMatch-based MVS methods have great performance in scene reconstruction tasks, [41, 48] point that PatchMatch-based methods often failed to reconstruct texture-less planar regions because the similarity measurement methods may be ineffective in these regions. [48] first generates the multiple plane hypotheses using the filtered depth maps on the areas that are not successfully recovered, then uses the Markov Random Field to select the depth hypotheses to improve the completeness of reconstruction results. [41] proposes a novel PatchMatch hypothesis to improve the reconstruction completeness by expanding reliable depth estimation in neighboring texture-less regions. Inspired by these methods which prove the effectiveness of the planar constraints, our methods also utilize planar prior but in the implicit surface reconstruction process.

Neural scene reconstruction. Different from the two-stage MVS methods, Atlas [34] first proposed to end-to-end reconstruct the 3D scenes, where 2D CNN features are back-projected and accumulated into the voxel volume to predict the truncated signed distance function (TSDF) with 3D CNN. NeuralRecon [47] uses recurrent networks to fuse the features from different views, which is much faster. TransformerFusion [3] uses the attention mechanism to attend to the most informative features of the image views in the temporal feature fusion. Recently, [32, 33, 37, 38, 46, 50, 51, 56] represent the scene directly as an implicit neural function and generate high-resolution reconstruction results. [29, 36] propose a new method which can learning 3D shapes from only 2D images by introducing an implicit differentiable renderer. IDR [57] models the geometry with a zero level-set of a neural network and the view-dependent appearance, but needs additional mask information. NeRF [33] achieves excellent novel view synthesis performance by implicit volume rendering but cannot get accurate and clean geometry. VolSDF [56] and NeuS [50] derivate density from SDF in volume rendering, therefore achieving good reconstruction for objects. When reconstructing indoor scenes, these methods suffer from texture-less planar regions. To deal with the problem, [11, 26, 40, 51] utilizes the sparse depth from COLMAP [42] as additional supervision or to guide the sampling process. Manhattan-SDF [18] regularizes the normal directions of pixels in the floor and wall regions under the Manhattan world assumption [9]. The performance of Manhattan-SDF is limited by the pre-trained semantic segmentation network, which predicts the semantic floor and wall regions in indoor scenes. Moreover, it cannot generalize to non-Manhattan scenes and the scenes cluttered by objects with small visible areas of floors and walls. Our method constrains indoor scenes with pseudo-planes obtained by unsupervised superpixel segmentation. Therefore, our approach can handle the aforementioned complex scenes.

3 Method

We tackle indoor reconstruction through neural implicit surface reconstruction. However, the optimization of implicit surface reconstruction is difficult for the texture-less regions. Therefore,

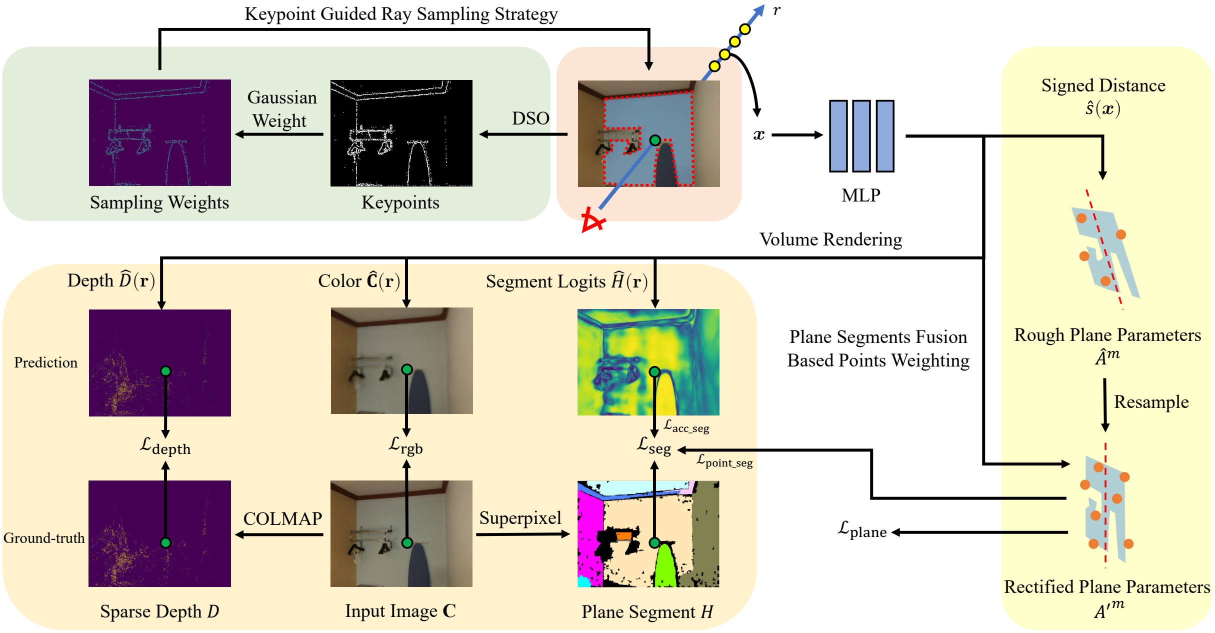


Fig. 2 The overview of our method. We utilize the DSO method to extract keypoints and then calculate the ray sampling weight to guide the sampling. We learn the appearance, signed distance, depth, and segments of 3D scenes with implicit neural representations. For each ray, we use differentiable volume rendering to render its pixel color, depth, and segment probabilities, which are supervised with the input image, sparse depth obtained by COLMAP, and plane segments in 2D by super-pixel, respectively. We propose the novel plane regularization by a two-step plane parameter estimation strategy, which improves the smoothness in planar regions and details in the non-planar parts. We calculate roughly estimated plane parameters according to the accumulated depth of the sampled points and resample more points to compute the rectified plane parameters according to the signed distances and the normal directions. To overcome noisy and inaccurate plane segments, we additionally introduce a plane segments fusion strategy to weight the points in the second rectified plane estimation step and the plane regularization.

we propose pseudo-plane regularization to apply additional regularization on the SDF of textureless regions. As the generated plane segments are noisy and inaccurate, we propose to fuse the plane segments from different views and assign points with varied weights in plane estimation and regularization based on the fusion. Lots of rays are randomly sampled in the planar regions. Therefore, the RGB rendering loss for the sampled ray in the planar region is usually close to 0 with a large reconstruction error; therefore the random ray sampling in these regions is inefficient and contributes less to the optimization. To avoid such redundant sampling, we propose the keypoint guided ray sampling strategy to attend to the informative textured regions with large color variations. As a result, the implicit network gets a better reconstruction result. The overview of our method can be seen in Fig. 2.

3.1 Neural Implicit Surface Reconstruction

We reconstruct the scenes by the neural implicit surface reconstruction methods [37, 50, 56] without 3D supervision. The reconstruction result is represented by a signed distance field (SDF). Following VolSDF [56], given one 3D point $\boldsymbol{x}$, we get its color $\hat{\mathbf{c}}(\boldsymbol{x})$[1] and signed distance $\hat{s}(\boldsymbol{x})$ by querying the MLP network F as:

$$(\hat{\mathbf{c}}(\boldsymbol{x}), \hat{s}(\boldsymbol{x})) = F(\boldsymbol{x}). \tag{1}$$

[1] ^ denotes the prediction from the network.

We derive the density $\sigma(\boldsymbol{x})$ from $\hat{s}(\boldsymbol{x})$ in volume rendering as:

$$\sigma(\boldsymbol{x};\beta) = \begin{cases} \frac{1}{2}\exp\left(\frac{\hat{s}(\boldsymbol{x})}{\beta}\right) & \text{if } \hat{s}(\boldsymbol{x}) \leq 0 \\ 1 - \frac{1}{2}\exp\left(-\frac{\hat{s}(\boldsymbol{x})}{\beta}\right) & \text{if } \hat{s}(\boldsymbol{x}) > 0, \end{cases} \tag{2}$$

where β is a learnable parameter.

The transparency T of the i-th sampled point along the ray $\boldsymbol{r}$ in the discrete form is:

$$T_i = \exp(-\sum_{j=1}^{i-1}\delta_j\sigma_j), \tag{3}$$

where $\delta_j = t_{j+1} - t_j$ is the distance between adjacent samples t_j and t_{j+1} along the ray. The accumulated color $\hat{\mathbf{C}}(\boldsymbol{r})$ along the ray $\boldsymbol{r}$ is:

$$\hat{\mathbf{C}}(\boldsymbol{r}) = \sum_{i=1}^{K} T_i\left(1 - \exp\left(-\sigma_i\delta_i\right)\right)\hat{\mathbf{c}}_i, \tag{4}$$

where K is the number of sampled points. The training loss contains three terms:

$$\mathcal{L} = \mathcal{L}_{\text{rgb}} + \lambda_{\text{sdf}}\mathcal{L}_{\text{sdf}} + \lambda_{\text{depth}}\mathcal{L}_{\text{depth}}, \tag{5}$$

where

$$\mathcal{L}_{\text{rgb}} = \sum_{\boldsymbol{r}\in\mathcal{R}} \|\hat{\mathbf{C}}(\boldsymbol{r}) - \mathbf{C}(\boldsymbol{r})\|, \tag{6}$$

$$\mathcal{L}_{\text{sdf}} = \sum_{\boldsymbol{x}\in\mathcal{X}} \left(\|\nabla_{\boldsymbol{x}} d(\boldsymbol{x})\|_2 - 1\right)^2, \tag{7}$$

$\mathbf{C}(\boldsymbol{r})$ is the ground-truth color of the ray $\boldsymbol{r}$. $\mathcal{L}_{\text{sdf}}$ is the Ekinoal loss [56, 57] to guarantee the property of SDF. $\mathcal{X}$ is the set of points sampled uniformly in the space. $\mathcal{L}_{\text{depth}}$ is the depth loss supervised by the depth $D(\boldsymbol{r})$ from SfM [42] following [11, 18, 40] to regularize the rendered depth $\hat{D}(\boldsymbol{r})$ as:

$$\hat{D}(\boldsymbol{r}) = \sum_{i=1}^{K} T_i\left(1 - \exp\left(-\sigma_i\delta_i\right)\right)t_i, \tag{8}$$

$$\mathcal{L}_{\text{depth}} = \sum_{\boldsymbol{r}\in\mathcal{D}} |\hat{D}(\boldsymbol{r}) - D(\boldsymbol{r})|, \tag{9}$$

where $\mathcal{D}$ is the set of rays that have valid depth values estimated by SfM[42] on the image. Since the depth from SfM is sparse and is not available in the texture-less regions, the reconstruction result is still poor with the loss above. Therefore, it is necessary to incorporate stronger regularization to constrain the indoor scene reconstruction to achieve a better result.

3.2 Plane Regularization

There are many man-made planes in indoor scenes, not only walls, floors, and ceilings but also tables, wardrobes, etc. These areas are also texture-less areas without valid supervision in reconstruction. It is reasonable to improve reconstruction in these areas by regularizing these areas to be flat if we know that these areas are planes.

3.2.1 Plane Segments Generation

Plane segments can be obtained by plane detection [27, 28, 61]. However, plane detection methods need extra annotated planes, which are not easy to obtain. Therefore, we propose to get planes unsupervised by super-pixel segmentation. Given one image during training, we follow [7, 8, 25, 59] to get the plane segments by super-pixel segmentation without supervision. The Felzenszwalb super-pixel segmentation[13] uses the greedy approach and segments areas with low gradients. The pixels with similar colors are clustered together. We treat the super-pixel segments whose areas are larger than the predefined thresholds to be the pseudo-planes, filtering the small segments that are likely to be non-planar out. As shown in Fig. 1, areas such as floors, walls, chair surface, and piano exterior are segmented.

3.2.2 Two-step based Pseudo-plane Parameter Estimation

We use the least square method to fit the 3D coordinates of n_1 points on the m-th plane to get its parameters A^m as [59]. Given a set of 3D points $X = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots \boldsymbol{x}_{n_1}]^\top$ belonging to the m-th plane in one image during training, the relation between the points X and the plane parameters A^m is

$$XA^m = \mathbf{1}, \tag{10}$$

$$A^m = \left(X^\top X + \epsilon I\right)^{-1} X^\top \mathbf{1}, \tag{11}$$

where I is the identity matrix. Enough points are beneficial to get accurate plane parameters. However, as we introduce plane regularization on the fly during training, we need to estimate the plane parameters online. To get the coordinates of 3D

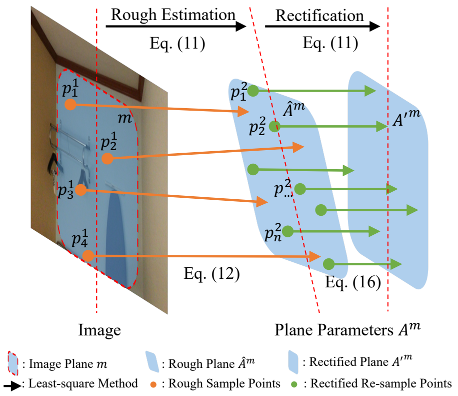


Fig. 3 The illustration of the two-step plane parameter estimation. We first estimate the parameters of the roughly estimated planes by fitting a few points back-projected by the expensive rendering depth in the first step. Further, we randomly sample more points on the roughly estimated plane and then obtain the signed distances and the normal directions by directly querying the MLPs efficiently. Finally, we march the points according to their signed distances and normal directions, and then estimate the parameters of the rectified planes by fitting these points.

points, it is necessary to get the depth of these points, which is rendered by volume rendering as Eq. (8). It is slow and memory-consuming to query every point along the ray to get the depth of only one point. Therefore, we propose to get the plane parameters efficiently and effectively in a two-step scheme, utilizing the property of SDF.

Our two-step plane parameter estimation scheme includes the rough plane parameter estimation and the rectified plane parameter estimation, as can be seen in Fig. 3.

Rough plane parameter estimation. In the first step, we get the roughly estimated plane parameters, which is inaccurate but fast. We sample only four points $\boldsymbol{p}_1^1, \boldsymbol{p}_2^1, \boldsymbol{p}_3^1, \boldsymbol{p}_4^1 \in \mathbb{R}^3$ in the pixel space[2] per plane segment during rendering and get the corresponding rendered depth $d_1^1, d_2^1, d_3^1, d_4^1$ via expensive volume rendering. The corresponding 3D points in the camera space of the training image are calculated as follows:

$$\boldsymbol{x}_i^1 = d_i^1 K^{-1} \boldsymbol{p}_i^1, \tag{12}$$

where K is the camera intrinsics matrix. We then get the roughly estimated plane parameters $\hat{A}^m$ based on Eq. (27). The parameters of the roughly estimated plane are noisy and inaccurate as the number of sampled points is limited by the memory in the slow volume rendering process.

Rectified plane parameter estimation. In the second step, we re-estimate the plane parameters utilizing SDF. We re-sample a new set of points $\boldsymbol{p}_1^2, \boldsymbol{p}_2^2, \dots, \boldsymbol{p}_{n_2}^2$ on the roughly estimated plane in the pixel space. The depth of these points on the roughly estimated plane is calculated as follows:

$$d^2\left(\boldsymbol{p}_i^2\right) = \left(\hat{A}^{m\top} K^{-1} \boldsymbol{p}_i^2\right)^{-1}. \tag{13}$$

We then back-propagate these points to get the coordinates of the 3D points in the world space as:

$$\begin{aligned} \boldsymbol{x}_i^2 &= E^{-1} d^2(\boldsymbol{p}_i^2) K^{-1} \boldsymbol{p}_i^2 \qquad (14)\\ &= E^{-1} \boldsymbol{p}_i^{2^{-1}} K \hat{A}^{m^{-\top}} K^{-1} \boldsymbol{p}_i^2, \qquad (15) \end{aligned}$$

where E is the extrinsic matrix of the camera. We then directly feed the 3D coordinates of these points into the MLP to get the corresponding signed distances $\hat{s}(\boldsymbol{x}_i^2)$ and normals $\hat{\boldsymbol{n}}(\boldsymbol{x}_i^2)$, respectively. The normal is computed by the derivative of the signed distance s. We assume that the points on the roughly estimated planes are close enough to the true surface, i.e., the plane. Therefore, we get the coordinates of the points on the rectified plane by marching these points according to the signed distances and normals as:

$$\bar{\boldsymbol{x}}_i = \boldsymbol{x}_i^2 - \hat{s}(\boldsymbol{x}_i^2)\hat{\boldsymbol{n}}(\boldsymbol{x}_i^2) \tag{16}$$

The rectified plane parameters A'^m are calculated from these points. It should be noted that the coordinates of the newly sampled points are directly from the MLP without the accumulated depth here. The process is much faster. Therefore, the number of sampled points in this step is much larger than in the first step. As a result of the scheme, the calculation of plane parameters is efficient and effective.

3.2.3 Pseudo-plane Regularized SDF

After getting the rectified plane parameters A'^m, we calculate the signed distances $s(\boldsymbol{x}_i^2)$ of the

[2] Here $\boldsymbol{p}_1^1, \boldsymbol{p}_2^1, \boldsymbol{p}_3^1, \boldsymbol{p}_4^1$ are the homogeneous coordinates of the sampled points. The superscript ^{1} and ^{2} are denoted to distinguish the points in the first step rough plane parameter estimation and the second step rectified plane parameter estimation, respectively

sampled points $\boldsymbol{x}_i^2$ to the estimated plane as:

$$s(\boldsymbol{x}_i^2) = \mathrm{sgn}(\bar{\boldsymbol{x}}_i, \boldsymbol{x}_i^2, \boldsymbol{x}_c)\|\frac{{\boldsymbol{x}_i^2}^\top A'^m - 1}{\sqrt{{A'^m}^\top A'^m}}\|, \tag{17}$$

where

$$\mathrm{sgn}(\bar{\boldsymbol{x}}_i, \boldsymbol{x}_i^2, \boldsymbol{x}_c) = \begin{cases} 1, & \text{if } \|\bar{\boldsymbol{x}}_i - \boldsymbol{x}_c\| \geq \|\boldsymbol{x}_i^2 - \boldsymbol{x}_c\|, \\ -1, & \text{if } \|\bar{\boldsymbol{x}}_i - \boldsymbol{x}_c\| < \|\boldsymbol{x}_i^2 - \boldsymbol{x}_c\|, \end{cases} \tag{18}$$

$\boldsymbol{x}_c$ is the coordinate of the camera, and $\mathrm{sgn}(\cdot)$ is the sign function that turns positive if the sampled point is outside the surface and negative if inside. We consider the calculated signed distance as the pseudo 'ground truth' to regularize the SDF. The plane regularization loss for the predicted signed distances $\hat{s}(\boldsymbol{x}_i)$ during training is:

$$\mathcal{L}_{\text{plane}} = \sum_{\boldsymbol{x} \in \mathcal{X}^2} \|s(\boldsymbol{x}) - \hat{s}(\boldsymbol{x})\|. \tag{19}$$

3.3 Plane Segments Fusion Based Points Weighting

The plane segments are usually noisy and inaccurate. If the points outside the true plane are considered in the pseudo-plane, the plane parameters are likely to be poorly estimated and the points are also forced to be close to the plane by the false regularization. Therefore, we propose to weight the points differently in plane estimation and regularization to remedy these problems. The weights are obtained by fusing plane segments from different views. We assume that the inconsistent areas of the segments from different views are more likely to be poorly segmented and the consistent areas are more accurate.

Plane Segments Fusion. We propose and train a segmentation network branch attached to the MLPs to fuse the plane segments. The network predicts sigmoid probability $h^m(\boldsymbol{x})$ of point $\boldsymbol{x}$ on plane m. To get the segmentation prediction $\hat{H}^m$ of plane m on the image, we accumulate the segmentation probability in volume rendering as:

$$\hat{H}^m(\boldsymbol{r}) = \sum_{i=1}^{K} T_i \left(1 - \exp\left(-\sigma_i \delta_i\right)\right) h_i^m, \tag{20}$$

We train the segmentation network branch with the binary cross entropy (BCE) loss:

$$\mathcal{L}_{\text{seg}} = \mathcal{L}_{\text{acc\_seg}} + \mathcal{L}_{\text{point\_seg}}. \tag{21}$$

The segmentation loss contains two parts. $\mathcal{L}_{\text{acc\_seg}}$ is the loss for the accumulated segmentation sigmoid probability, and $\mathcal{L}_{\text{point\_seg}}$ is for the sigmoid probability $\mathbf{h}^i(\boldsymbol{r})$ of the sampled points $\boldsymbol{x}^2$ on the roughly estimated planes on the ray $\boldsymbol{r}$ in the second step.

$$\begin{aligned} \mathcal{L}_{\text{acc\_seg}} = \sum_{\boldsymbol{r} \in \mathcal{R}} \sum_{i=1}^{M} & H^i(\boldsymbol{r}) \log(\hat{H}^{\alpha(i)}(\boldsymbol{r})) \\ & + (1 - H^i(\boldsymbol{r})) \log(1 - \hat{H}^{\alpha(i)}(\boldsymbol{r})), \end{aligned} \tag{22}$$

$$\begin{aligned} \mathcal{L}_{\text{point\_seg}} = \sum_{\boldsymbol{r} \in \mathcal{R}} \sum_{i=1}^{M} & h^i(\boldsymbol{r}) \log(\hat{H}^{\alpha(i)}(\boldsymbol{r})) \\ & + (1 - h^i(\boldsymbol{r})) \log(1 - \hat{H}^{\alpha(i)}(\boldsymbol{r})). \end{aligned} \tag{23}$$

The correspondence between i and $\alpha(i)$ is achieved by matching the accumulated prediction H^i and $\hat{H}^{\alpha(i)}$ with Hungarian matching [4, 23, 49]. We search for a permutation of M elements $\alpha \in \mathfrak{S}_M$ with the lowest cost as:

$$\alpha = \underset{\alpha \in \mathfrak{S}_M}{\arg\min} \sum_{i}^{M} \mathcal{L}_{\text{match}} \left(H^i, \hat{H}^{\alpha(i)}\right) \tag{24}$$

$\mathcal{L}_{\text{match}}$ is measured by pair-wise sIoU and BCE.

Weighting. By fusing the plane segments, $h^m(\boldsymbol{x})$ is relatively low in the inconsistent regions, while in the consistent regions, $h^m(\boldsymbol{x})$ is high. We normalize $h^m(\boldsymbol{x_i})$ to $\bar{h}^m(\boldsymbol{x_i}) = \frac{h^m(\boldsymbol{x_i})}{\sum_{i=1}^{n} h^m(\boldsymbol{x_i})} n$. We utilize $h^m(\boldsymbol{x})$ to weight Eq. (27) in the second step rectified plane parameter estimation as:

$$W = \begin{bmatrix} \bar{h}^m(\boldsymbol{x_1}) & & \\ & \ddots & \\ & & \bar{h}^m(\boldsymbol{x_{n_2}}) \end{bmatrix}, \tag{25}$$

$$A = \left(X^\top W X + \epsilon I\right)^{-1} X^\top I. \tag{26}$$

Eq. (19) in the plane regularization is weighted as:

$$\mathcal{L}_{\text{plane}} = \sum_{\boldsymbol{x}\in\tilde{\mathcal{X}}^2} \bar{h}^m(\boldsymbol{x})\|s(\boldsymbol{x}) - \hat{s}(\boldsymbol{x})\|. \tag{27}$$

Here we do not weight the points in the first step of plane parameters estimation. The reason is that we only get the roughly estimated plane parameters in the first step. The second step is more important.

3.4 Overall Loss

To be summerized, our overall loss function is:

$$\begin{aligned}\mathcal{L} =& \mathcal{L}_{\text{rgb}} + \lambda_{\text{sdf}}\mathcal{L}_{\text{sdf}} + \lambda_{\text{depth}}\mathcal{L}_{\text{depth}} + \\ & \lambda_{\text{plane}}\mathcal{L}_{\text{plane}} + \lambda_{\text{seg}}\mathcal{L}_{\text{seg}},\end{aligned} \tag{28}$$

where $\mathcal{L}_{\text{rgb}}, \mathcal{L}_{\text{sdf}}$ and $\mathcal{L}_{\text{depth}}$ is explained in Eq. (5). $\mathcal{L}_{\text{sdf}}$ is our plane regularization loss in Eq. (27), and $\mathcal{L}_{\text{seg}}$ is the plane segmentation loss in Eq. (21).

3.5 Keypoint Guided Ray Sampling Strategy

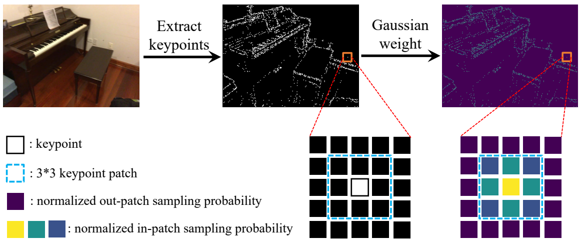

Extract keypoints
Gaussian weight
□ : keypoint
: 3*3 keypoint patch
: normalized out-patch sampling probability
: normalized in-patch sampling probability

Fig. 4 The illustration of keypoint guided ray sampling strategy. We utilize Direct Sparse Odometry (DSO) [12] to extract keypoints from the color images. We initialize the sampling weight of every pixel in the image as 1, then choose the 3×3 patch centered at the keypoints and calculate the sampling weight of pixels in patches by Gaussian distance. We finally normalize the weights of the entire image to sample rays.

Neural volumetric rendering is optimized based on the rendering loss between the predicted RGB color and the ground-truth color. In indoor scenes, ordinary random ray sampling is inefficient and redundant. The reason is that the planar regions usually occupy large areas and the rays in the texture-less regions are easy to be sampled. Considering there is a small offset for a sampled ray but still hitting inside the same plane. The RGB rendering loss is usually small as the RGB colors are similar inside the plane, after some iterations. Though the reconstruction is wrong, the loss may be close to zero. Sampled rays in these regions contribute less to the reconstruction optimization. It is reasonable that the sampling strategy should attend to the informative textured areas with large color variations. Inspired by [12, 14, 42, 43, 59], we propose the keypoint guided ray sampling strategy. To be specific, we extract the keypoints by Direct Sparse Odometry [12]. We calculate the sampling weight G_{p_i} of 2D point p_i by the Gaussian distance to the keypoints in the patch P_{q_j} centered at the 2D keypoints q_j as:

$$G_{p_i} = \begin{cases} 1 + k\exp(\frac{\|p_i - q_j\|_2}{\gamma}), & j = \arg\min \|p_i - q_j\|_2, \\ & \text{and } p_i \in P_{q_j}, \\ 1, & \text{otherwise}, \end{cases} \tag{29}$$

where k and γ are hyper-parameters to control the scale of sampling weights. The final sampling probability $\bar{G}_{p_i}$ for point p_i is normalized as:

$$\bar{G}_{p_i} = \frac{G_{p_i}}{\sum_j G_{p_j}} \tag{30}$$

The illustration of our proposed keypoint guided ray sampling strategy can be seen in Fig. 4.

4 Experiments

4.1 Datasets and Metrics

Datasets. We perform our experiments on ScanNet (V2) [10] and 7-Scenes [45]. ScanNet (V2) is a large-scale challenging real-world indoor scene dataset with ground-truth camera poses and surface reconstructions, containing around 2.5M RGB-D images captured in 1613 indoor scenes. 7-Scenes consists of RGB-D frames captured by a tracked Kinect RGB-D camera and uses KinectFusion [20] to obtain dense 3D models and camera poses. While previous neural indoor scene reconstruction methods with depth and normal priors mainly focused on the scenes with Manhattan world assumption, in our experiments, we prove that our method can perform better in more general scenes. Thus we consider: a) The same real-world indoor scenes following [18]; b)

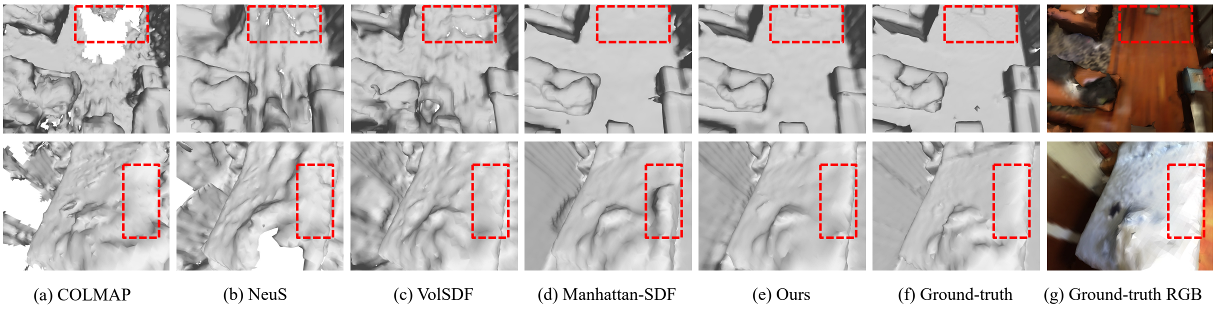


Fig. 5 Visualization compared with other methods on the indoor scenes in Manhattan-SDF [18] from the ScanNet dataset. COLMAP [42] and NeuS [50] cannot produce proper reconstruction results. VolSDF [56] leads to noisy reconstructions without additional constraints. Manhattan-SDF improves the smooth results in the regions such as floor, wall, and bed, but details are missing in some small planar areas such as the sweeping robot. In contrast, our method can get a more detailed surface in these regions, achieving compelling results.

Four additional Manhattan world scenes where the DeepLab V3+ segmentation network predicts poorly on the wall and floor regions; c) Four additional Manhattan world scenes with small visible areas of walls and floors; d) Four non-Manhattan world scenes from the ScanNet dataset.

Metrics. Following [18, 47, 60], we use RGB-D fusion results as 3D reconstruction ground truth and evaluate our method using five standard metrics defined in [34]: accuracy, completeness, precision, recall, and F-score with a threshold of 5cm.

4.2 Implementation Details

Our network is trained for 50 epochs with batches of 1024 rays. We use the Adam optimizer [22] with the initial learning rate 5×10^4 and an exponential learning rate decay strategy. All RGB images are initially resized to 640×480 for scene reconstruction, and we utilize the Felzenszwalb super-pixel segmentation [13] and COLMAP [42] method to extract super-pixel and sparse depth maps from these re-scaled images respectively. We implement our method with PyTorch [39] and train our network on one NVIDIA GeForce RTX 2080Ti GPU. We obtain the camera's intrinsic and extrinsic parameters from the original dataset [10], and adjust them to be consistent with the image scaling. Following [1], we normalize all cameras to be inside a unit sphere and initialize network parameters by representing the initial SDF to a unit sphere with the surface normals facing inside. We evaluate the results by the Marching Cubes algorithm [30] to extract surface mesh from the learned signed distance function. To overcome the effect of unobserved regions in evaluation, suggested by Manhattan-SDF [18], we render depth maps from predicted mesh and re-fuse them with TSDF fusion [20] following [34]. We set λ_{sdf} as 0.1, λ_{depth} as 1.0 following [18], and set λ_{plane} as 0.01, λ_{seg} as 0.05.

4.3 Performance Comparisons

We compare our method with the following methods: traditional MVS reconstruction method: COLMAP [42]. We use screened Poisson Surface reconstruction (sPSR) [21] to reconstruct mesh from extracted point clouds; Neural volume rendering methods: NeRF [33], NeuS [50] and VolSDF [56]; Volume rendering with additional pre-trained segmentation model: Manhattan-SDF [18]. We do not visualize the reconstruction results of NeRF because it fails to reconstruct indoor scenes without geometric constraints. Our method achieves state-of-the-art performance not only in the Manhattan scenes but also in the non-Manhattan scenes. Our method is also unrestricted by the pre-trained network which may perform poorly on some scenes, and can also perform well on the cluttered indoor rooms with small visible areas of floors and walls.

Manhattan scenes:

1. We compare PPlaneSDF with other methods on the four scenes on the ScanNet dataset [10], and the four scenes on the 7-Scenes [45] from Manhattan-SDF [18]. As can be seen in Table 1, our method achieves state-of-the-art results on these scenes. As can be seen in Fig. 5, COLMAP

Table 1 Results on the ScanNet dataset and the 7-Scenes dataset. We compare our method with the classical MVS and the neural volume rendering methods on the four indoor scenes from Manhattan-SDF [18] in each dataset. † indicates the result reproduced by ours. Other results are from Manhattan-SDF. The averaged results show that our method achieves the competitive reconstruction performance on the ScanNet dataset and performs better on the 7-Scenes scenes. The best and the second best results are boldfaced and underlined, respectively.

Method	ScanNet					7-Scenes				
	Acc ↓	Comp ↓	Prec ↑	Recall ↑	**F-score** ↑	Acc ↓	Comp ↓	Prec ↑	Recall ↑	**F-score** ↑
NeRF [33]	0.735	0.177	0.131	0.290	0.176	0.573	0.321	0.159	0.085	0.083
NeuS [50]	0.179	0.208	0.313	0.275	0.291	0.151	0.247	0.313	0.229	0.262
VolSDF [56]	0.414	0.120	0.321	0.394	0.346	0.285	**0.140**	0.220	**0.285**	0.246
COLMAP [44]	**0.047**	0.235	**0.711**	0.441	0.537	**0.069**	0.417	**0.536**	0.202	0.289
Manhattan-SDF† [18]	0.050	**0.055**	0.705	0.660	0.681	0.161	0.288	0.360	0.182	0.239
Ours	0.057	0.062	0.703	**0.664**	**0.683**	0.104	0.301	0.511	0.209	**0.295**

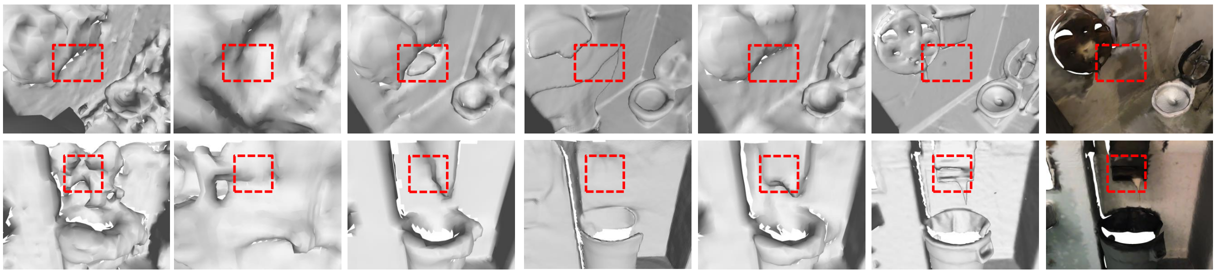
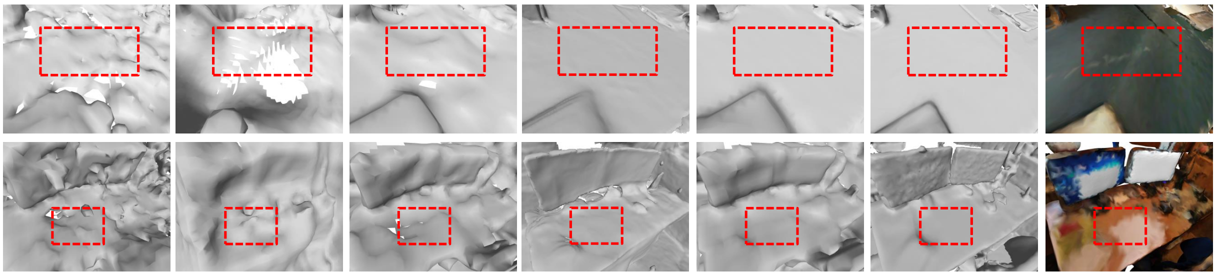

(a) COLMAP (b) VolSDF (c) Manhattan-SDF (d) MonoSDF (e) Ours (f) Ground-truth (g) Ground-truth RGB

Fig. 6 Visualization compared with other methods on Manhattan scenes where the DeepLab V3+ [5] segmentation network predicts poorly on the wall and floor regions. COLMAP and VolSDF cannot achieve competitive results. Manhattan-SDF and MonoSDF produces unreasonable reconstruction results in the poorly predicted segmentation regions, e.g., the wall region near the paper holder. Compared with these approaches, our approach can produce more details and smooth surfaces, achieving better results.

and NeuS cannot produce proper reconstruction results. VolSDF leads to noisy reconstructions without additional constraints. Manhattan-SDF improves the smooth results in the regions such as floor, wall, and bed, but details are missing in some small planar areas such as the sweeping robot. In contrast, our method can get a more detailed surface in these regions while achieving compelling results.

2. We also conduct a comparison on the other scenes in the ScanNet dataset. In Table 2, we select 4 scenes where DeepLab V3+ [5] segmentation network predicts poorly on the wall and floor regions in the ScanNet dataset. The results show that Manhattan-SDF [18] degrades on these scenes restricted by the inaccurate predictions of the pre-trained segmentation network. As can be seen in Fig. 6, COLMAP and VolSDF cannot achieve competitive results. The depth estimation network of MonoSDF [60] also performs pool. Manhattan-SDF and MonoSDF produces unreasonable reconstruction results in the poorly predicted regions, e.g., the wall region near the paper holder. Compared with these approaches, our approach can produce more details and smooth surfaces, achieving better results. The comparison between our generated planes and the semantic segmentation results used in Manhattan-SDF can be seen in Fig. 8. DeepLab V3+ predicts poorly in some scenes, while ours gives a better segmentation result.

3. In Table 3, we select the scenes where the walls and floors are heavily occupied by other objects. Manhatan-SDF degrades due to insufficient constraints on the small area of walls and floors, while our proposed plane regularization can still work on the small areas of planar regions on the objects



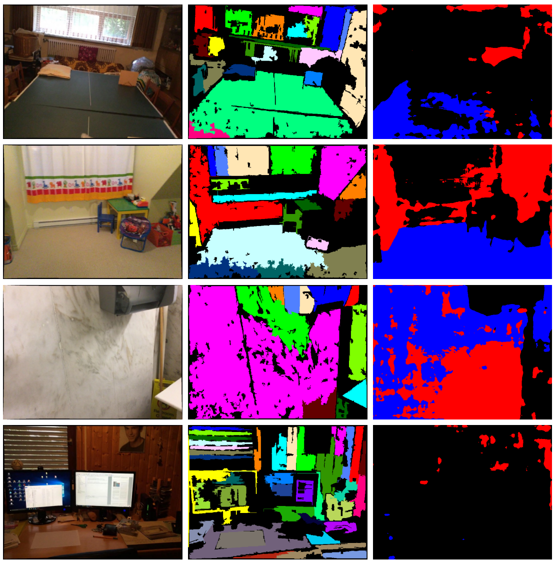
(a) COLMAP (b) VolSDF (c) Manhattan-SDF (d) MonoSDF (e) Ours (f) Ground-truth (g) Ground-truth RGB

Fig. 7 Visualization compared with other methods on the scenes with small visible areas of walls and floors. COLMAP and VolSDF do not lead to complete reconstructions. Manhattan-SDF cannot put effective Manhattan world assumption constraints on these scenes and produce poor reconstructions on some objects, e.g., the table tennis table and computer desk. Instead, our method relies not only on the wall and floor regions but also on these small areas of planes on objects more generally. So we can achieve more smooth and more accurate results with more details and obtain results competitive with the MonoSDF method.

Table 2 Results compared with other methods on Manhattan scenes. It shows that DeepLab V3+ [5] segmentation network predicts poorly on the wall and floor regions. We chose four scenes from the ScanNet dataset. Our approach can still perform well while Manhattan-SDF [18] degrades. The best and the second best results are boldfaced and underlined, respectively.

Method	Acc↓	Comp↓	Prec↑	Recall↑	F-score↑
NeRF [33]	0.420	0.612	0.041	0.017	0.023
NeuS [50]	0.283	0.482	0.182	0.136	0.153
VolSDF [56]	0.254	0.306	0.239	0.179	0.198
MonoSDF [60]	0.199	**0.123**	0.426	0.396	0.407
COLMAP [44]	0.131	<u>0.149</u>	0.482	<u>0.452</u>	0.460
Manhattan-SDF [18]	**0.104**	0.171	**0.563**	0.442	<u>0.493</u>
Ours	<u>0.107</u>	0.168	**0.563**	**0.472**	**0.512**

and performs better. In Fig. 8, DeepLab V3+ also predicts nearly nothing in the scenes where walls and floors are heavily occupied. In contrast, the super-pixel approach performs better in these scenes. Results validate the effectiveness of our proposed PPlaneSDF. The visualization can be in Fig. 7. COLMAP and VolSDF do not lead to complete reconstructions. Manhattan-SDF cannot put effective Manhattan world assumption constraints on these scenes and produce poor reconstructions on some objects, e.g., the regions on the table tennis table and computer desk. Instead, our method benefits not only the wall and floor regions but also these small areas of planes on objects more generally. So we can achieve more smooth and more accurate results with more details results.

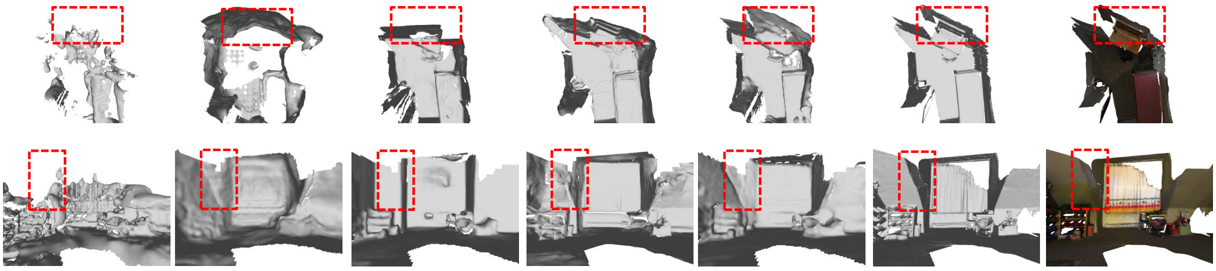


(a) Image (b) Superpixel Segments (c) Semantic Segments

Fig. 8 The comparison between plane segments by superpixel (b) and the semantic segments for wall and floor by DeepLab V3+ (c). The floors are masked with red color, and the walls are masked with blue color in (c). DeepLab V3+ predicts poorly in some scenes. It also predicts nearly nothing in the scenes where walls and floors are heavily occupied. In contrast, the super-pixel approach we adopt gives a better segmentation result in these scenes.

MonoSDF [60] performs better than us as the monocular depth estimation network performs well on these scenes.

Non-Manhattan scenes: Except Manhattan scenes, there exist lots of non-Manhattan scenes



(a) COLMAP (b) VolSDF (c) Manhattan-SDF (d) MonoSDF (e) Ours (f) Ground-truth (g) Ground-truth RGB

Fig. 9 Visualization compared with other methods on the non-Manhattan scenes. COLMAP and VolSDF can only produce rough surface reconstruction results. Manhattan-SDF wrongly predicts the sloping wall as zigzag because of the Manhattan world assumption. MonoSDF cannot achieve competitive results due to its heavy reliance on the monocular depth prediction model. Without Manhattan world assumption and any pretrained model, our method achieves reasonable and detailed surfaces, getting better performance and generalizing well.

Table 3 Results on the cluttered Manhattan scenes with small visible areas of walls and floors. We choose four scenes from the ScanNet dataset. Results show that our method can still achieve the second best performance in these challenging scenes. The best and the second best results are boldfaced and underlined, respectively.

Method	Acc↓	Comp↓	Prec↑	Recall↑	F-score↑
NeRF [33]	0.448	0.566	0.042	0.010	0.015
VolSDF [56]	0.322	0.236	0.243	0.159	0.184
NeuS [50]	0.124	0.156	0.370	0.285	0.320
Manhattan-SDF [18]	<u>0.103</u>	0.148	0.442	0.361	0.396
COLMAP [44]	0.160	0.136	0.404	0.406	0.402
MonoSDF [60]	0.105	**0.118**	**0.514**	**0.490**	**0.501**
Ours	**0.093**	<u>0.132</u>	<u>0.494</u>	<u>0.408</u>	<u>0.445</u>

Table 4 Results on the non-Manhattan scenes. We choose four non-Manhattan scenes from the ScanNet dataset. Our method surpasses other methods on challenging scenes. The best and the second best results are boldfaced and underlined, respectively.

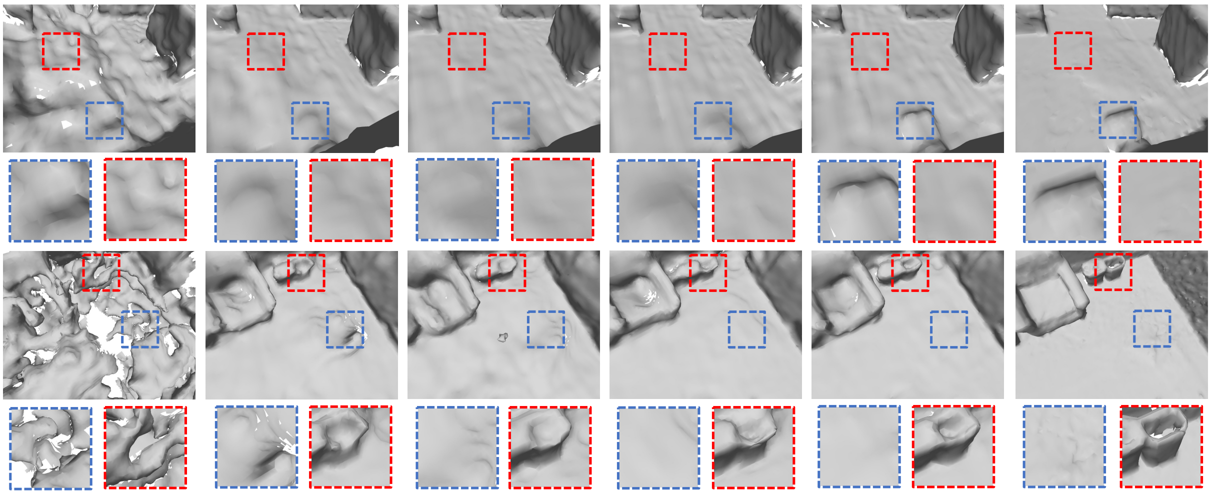
	Non-Manhattan				
Method	Acc↓	Comp↓	Prec↑	Recall↑	F-score↑
NeRF [33]	0.538	0.760	0.036	0.011	0.016
NeuS [50]	0.246	0.349	0.173	0.120	0.140
VolSDF [56]	0.419	0.223	0.217	0.199	0.204
COLMAP [44]	0.134	0.195	0.440	0.358	0.389
MonoSDF [60]	0.122	**0.090**	0.441	0.444	0.441
Manhattan-SDF [18]	<u>0.109</u>	0.132	<u>0.516</u>	<u>0.450</u>	<u>0.478</u>
Ours	**0.105**	<u>0.131</u>	**0.542**	**0.495**	**0.515**

where there are more than three dominant directions. In Table 4, we choose four non-Manhattan scenes from the ScanNet dataset. The results show that our method performs well on the non-Manhattan scenes and surpasses Manhattan-SDF a lot. The reason is that our method is not constrained to the Manhattan world assumption while Manhattan-SDF still imposes the Manhattan world assumption and gets unreasonable results. Fig. 9 shows that COLMAP and VolSDF can only produce rough surface reconstruction results. Manhattan-SDF wrongly predicts the sloping wall as zigzag because of the Manhattan world assumption. MonoSDF cannot achieve competitive results due to its heavy reliance on the monocular depth prediction model. Without Manhattan world assumption and any pretrained model, our method achieves reasonable and detailed surfaces, getting better performance and generalizing well.

4.4 Ablation Studies

4.4.1 The Importance of Different Modules in PPlaneSDF

To validate the effectiveness of our proposed method, we conduct ablation studies on the four Manhattan scenes from Manhattan-SDF [18]. The results are shown in Table 5. In Table 5, the baseline is the original VolSDF [56]. Depth means utilizing the sparse depth from COLMAP [42] as



(a) VolSDF (b) VolSDF-D (c) VolSDF-D-P (d) VolSDF-D-P-W (e) PPlaneSDF (VolSDF-D-P-W-S) (f) Ground-truth

Fig. 10 Visualization of the ablation studies of PPlaneSDF on the Manhattan scenes in the ScanNet dataset. Our proposed method can effectively improve reconstruction quality and produce much more detailed and smooth reconstruction results than the baselines. Note that VolSDF-D-S can reconstruct more clear contours of objects such as the sweeping robot and doorframe compared to VolSDF-D, and VolSDF-D-P can reconstruct smoother planes such as the wall region compared to VolSDF-D. VolSDF-D-P-W is also better than VolSDF-D-P. Our PPlaneSDF can maintain the details while also producing smooth and high-quality reconstruction results. (Note: *-D: w/ sparse depth; *-P: w/ our proposed plane regularization; *-W: w/ plane segments fusion based points weighting; *-S: w/ keypoint guided sampling)

Table 5 The ablation study of PPlaneSDF. The result is averaged on the four Manhattan scenes from Manhattan-SDF [18]. We retrain the VolSDF and add components to it. Results show that every design in our method works and gets the best performance with the combination of all designs. The best and the second best results are boldfaced and underlined, respectively.

Depth	Plane Reg	Weighting	Sampling	Acc ↓	Comp ↓	Prec ↑	Recall ↑	**F-score ↑**
				0.270	0.145	0.416	0.387	0.397
✓				0.062	0.065	0.637	0.611	0.623
✓			✓	0.067	0.065	0.652	0.629	0.640
✓	✓			0.059	0.067	0.680	0.636	0.657
✓	✓	✓		**0.055**	**0.062**	<u>0.689</u>	<u>0.650</u>	<u>0.669</u>
✓	✓	✓	✓	**0.055**	**0.062**	**0.703**	**0.664**	**0.683**

additional supervision following [18, 26, 40, 51], which is also adopted in our method. Plane Reg means our proposed plane regularization. Weighting means our proposed plane segments fusion based points weighting. Sampling means the keypoint guided ray sampling strategy. Results show that every design in our proposed PPlaneSDF works, and the combination of all designs gets the best performance. Compared with VolSDF with COLMAP depth as supervision, our proposed plane regularization improves by 0.034 on F-score. F-score further increases by 0.012 by our proposed plane segments fusion based points weighting. The final F-score increases by 0.014 by our proposed keypoint guided ray sampling strategy.

The visualization can also be seen in Fig. 10. The results verify the effectiveness of our method. The visualization of the reconstruction by PPlaneSDF in the training process can be seen in Fig. 11. It shows that the regions are becoming planar in the early training.

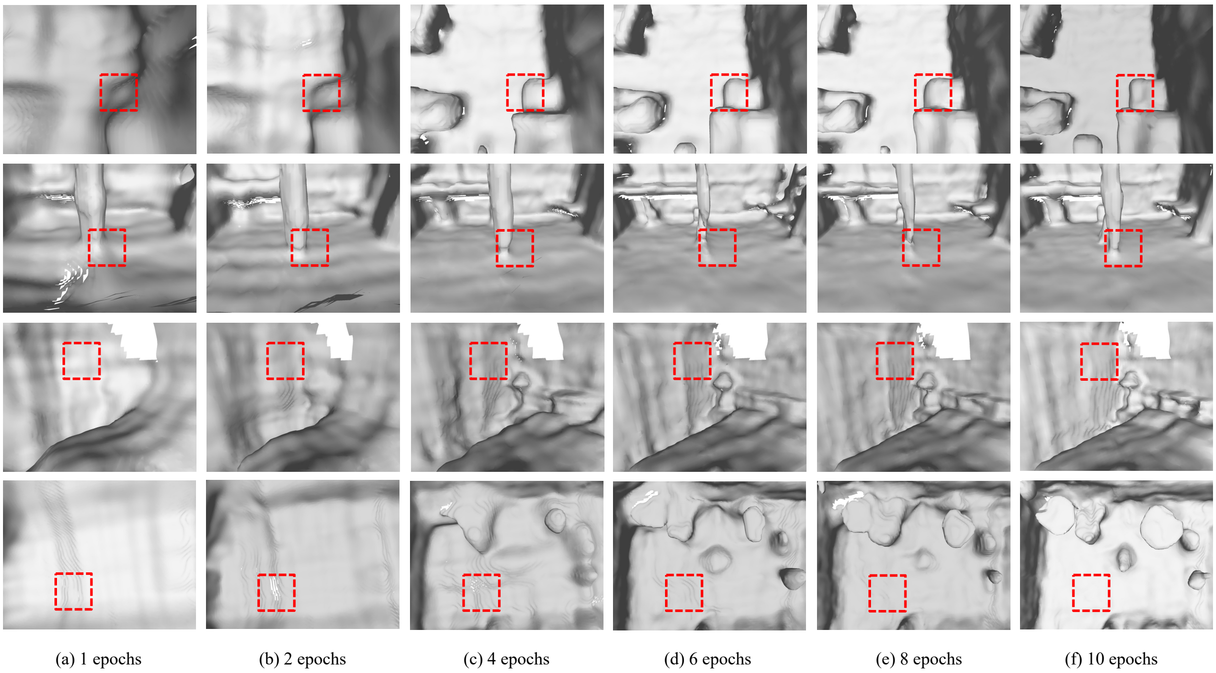


Fig. 11 Visualization of the reconstruction results of PPlaneSDF on the Manhattan scenes in the ScanNet dataset at early training epochs. It shows that the contour of the objects such as the boxes and piano become more clear, and some planar areas such as the walls, floor, bed, and table become flatter, gradually. The quality of reconstruction is gradually improving in the training process.

4.4.2 Number of Points in Plane Parameter Estimation.

The number of points is important to estimate an accurate plane parameter, therefore, we propose the two-step plane parameter estimation. In Table 6, we study the effect of the different number of points in the rectified plane parameter estimation. It can be seen that when the number of sample points is too small, the performance degrades a lot. The reason is that the few sample points make the plane estimation inaccurate. Accurate estimation is in need of sampling sufficient points. When the number of sample points is large, the performance also degrades. The reason may be that the unsupervised plane segments by super-pixel are noisy and inaccurate, therefore it may need some randomness.

4.4.3 Visualization of Plane Segments Fusion Based Points Weighting

We visualize the plane segments fusion based points weighting. As can be seen in Fig. 12, the weights are small in the mistakenly segmented regions, such as the schoolbag and lamp. In contrast, the weights are large in the accurately segmented regions, such as the ceiling and wall. The weights are distributed differently in the whole image, showing the effectiveness of our method. Benefiting from our fusion strategy, we are able to predict plane parameters effectively and impose corresponding constraints for generating reasonable results even in the case of over-segmented pseudo-planar regions, seen in Fig. 13.

Table 6 The performance of PPlaneSDF with the different number of sample points in rectified plane parameter estimation. The result is the averaged F-score on the four Manhattan scenes from Manhattan-SDF [18]. The best performance is achieved with 8192 points. The larger or smaller number of sample points degrades the performance. The best and the second best results are boldfaced and underlined, respectively.

# of samples	1024	2048	4096	8192	16384	32768
F-score	0.633	0.638	<u>0.642</u>	**0.657**	0.638	0.634

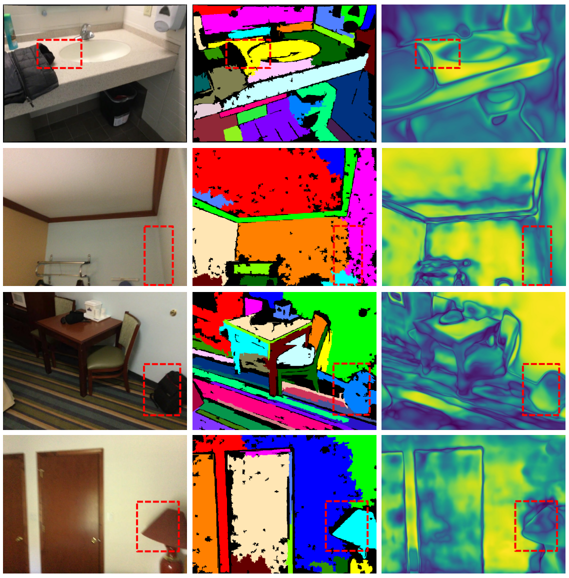


Fig. 12 The illustration of the plane segments by superpixel (b) and the weights of points with plane segments fusion (c). The weights are obtained by plane segment fusion from different views. The weights are distributed differently in the whole image. The weights are very small in the mistakenly segmented regions, such as the schoolbag and lamp. In contrast, the weights are large in the accurately segmented regions, such as the ceiling and wall.

Table 7 The performance of PPlaneSDF with different loss weights of plane regularization (λ_{plane}). The result is the averaged F-score on the four Manhattan scenes from Manhattan-SDF [18]. The best and the second best results are boldfaced and underlined, respectively.

λ_{plane}	0.01	0.05	0.1	0.2	0.5	1.0
F-score	0.618	0.632	<u>0.649</u>	**0.657**	0.640	0.592

4.4.4 Loss Weights

We conduct ablations on different loss weights in PPlaneSDF. Appropriate loss weights in the proposed plane regularization and plane segmentation get the best performance, seen in Table 7 and Table 8. Too large loss weights for plane regularization and plane segmentation may hinder the optimization process of neural surface reconstruction, while too small loss weights may lose the effect.

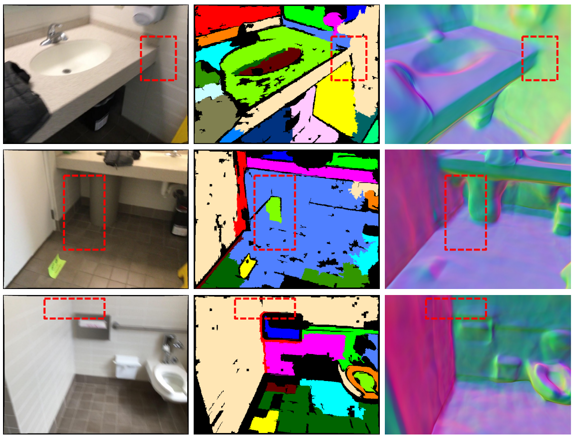


Fig. 13 The illustration of some cases about the plane over-segments by super-pixel (b) and the normal maps our model predicted (c). The accurate normal maps are obtained benefiting from our plane segments fusion strategy from different views. As for the over-segmented pseudo-planar regions, we can also predict accurate plane parameters effectively.

Table 8 The performance of PPlaneSDF with different loss weights of plane segmentation (λ_{seg}). The result is the averaged F-score on the four Manhattan scenes from Manhattan-SDF [18].

λ_{seg}	0.001	0.005	0.01	0.05	0.1
F-score	0.649	0.650	**0.669**	<u>0.653</u>	0.647

Table 9 The ablation of the keypoint guided ray sampling strategy. The result is the averaged F-score on the four Manhattan scenes from Manhattan-SDF [18]. The best and the second best results are boldfaced and underlined, respectively.

Method	k			
	0.5	1	1.5	2
Uniform	0.607	0.608	0.617	0.608
Gaussian	0.612	0.612	**0.640**	<u>0.614</u>

4.4.5 Keypoint Guided Ray Sampling

We conduct ablation for keypoint guided ray sampling strategy in Table 9. We try different sampling weight strategies. Gaussian means that the weights in the patches are set corresponding to the Gaussian distance to the keypoints in Eq. (29). Weights are large near the keypoints and small far from them. Uniform means that the weights in the

Table 10 The ablation of the threshold of pseudo-plane segmentation. The result is the averaged F-score on the four Manhattan scenes from Manhattan-SDF [18]. The best and the second best results are boldfaced and underlined, respectively.

seg. threshold	1000	2000	3000	4000
F-score	<u>0.664</u>	**0.683**	0.659	0.644

patches centered at keypoints are set uniform and equal to the mean of the corresponding Gaussian weights. The results show that Gaussian performs better than the Uniform weights.

4.5 The Threshold of Pseudo-plane Segmentation

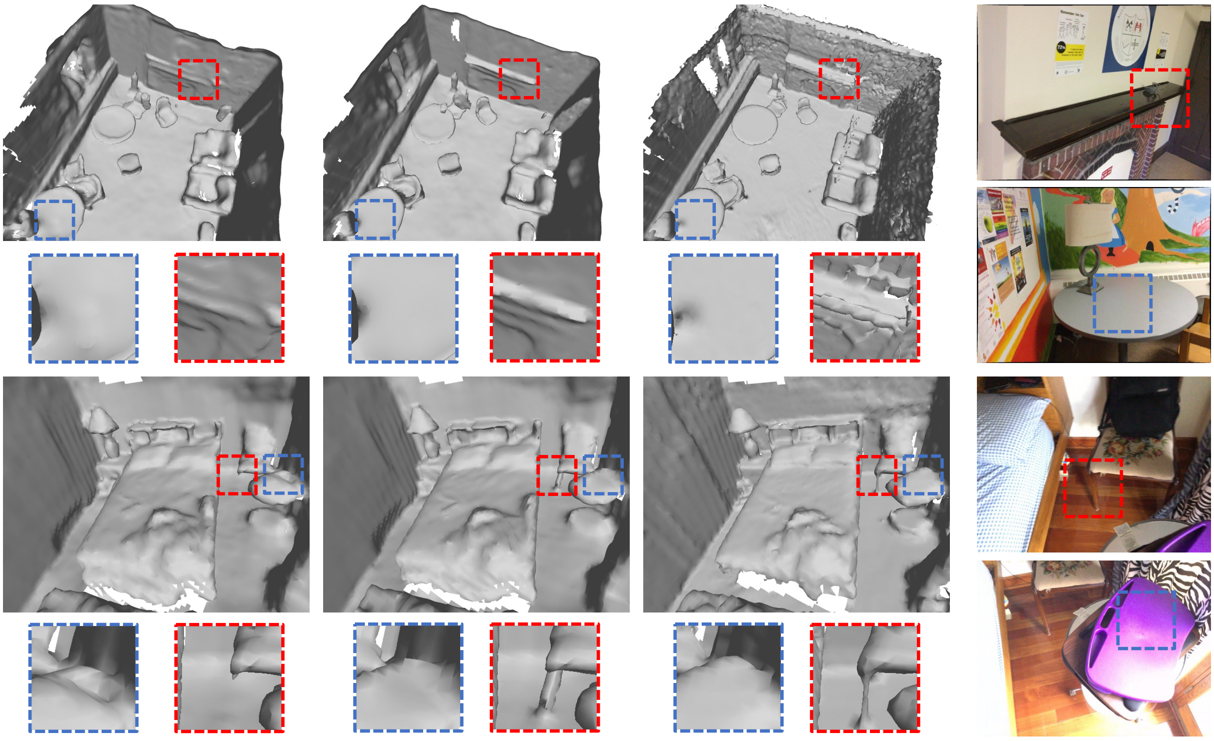
We treat the super-pixel segments whose areas are larger than the predefined thresholds to be the pseudo-planes, filtering the small segments that are likely to be non-planar out. We conduct ablation on the segmentation threshold in Table 10. The results show that too small or too large thresholds lower the performance. A small threshold results in over-segmentation, therefore imposing incorrect plane regularity on the non-plane areas. In contrast, large threshold results in under-segmentation, therefore the plane regularity is lacking in the plane areas. As a result, the performance decreases. Finally, we choose the 2000 threshold to extract pseudo-planes in the scene four Manhattan scenes from Manhattan-SDF [18]. The selection of the threshold is related to the resolution of input RGB images and the size of planar regions in the indoor scene. For the 640 × 480 resolution training images of the ScanNet dataset, smaller pseudo-plane segmentation threshold (e.g., 1000) are usually chosen for indoor scenes with smaller planar regions (e.g. the utility rooms) to achieve better pseudo-plane constraints. On the other hand, larger pseudo-plane segmentation threshold (e.g., 2000) are preferred for indoor scenes with larger planar regions (such as bathrooms) for better results.

Table 11 Averaged results of the combination of our proposed PPlaneSDF and Manhattan-SDF [18] on the four Manhattan scenes from Manhattan-SDF. Results show that our proposed methods are compatible and can get better performance combined with Manhattan-SDF. The best and the second best results are boldfaced and underlined, respectively. (Note: *+P: w/ our proposed plane regularization; *+W: w/ plane segments fusion based points weighting; *+S: w/ keypoint guided sampling; *+PPlaneSDF = *+P+W+S)

Method	Acc↓	Comp↓	Prec↑	Recall↑	F-score↑
M-SDF	0.050	0.055	0.705	0.660	0.681
M-SDF + P	0.049	<u>0.053</u>	0.714	0.665	0.688
M-SDF + P + W	<u>0.048</u>	0.054	<u>0.727</u>	<u>0.674</u>	<u>0.699</u>
M-SDF + PPlaneSDF	**0.046**	**0.052**	**0.737**	**0.683**	**0.709**

4.6 Combination with Our Method with Manhattan-SDF

Though our method achieves comparable performance on the four Manhattan scenes in ScanNet dataset from Manhattan-SDF [18] and performs better on other scenes, we combine our method with Mahattan prior from Manhattan-SDF to achieve better performance. We combine the Manhattan-SDF loss on the segmented floors and walls with our method. Manhattan-SDF can deal with large areas of planar regions, while our plane regularization can serve as an enhancement on the small areas of plane regions. As can be seen in Table 11, with our proposed plane regularization, the F-score improves by 0.007. Adding plane segments fusion based points weighting, the F-score further improves by 0.011. With our keypoint guided sampling, the F-score further improves by 0.01. The results validate that the effectiveness of our method and our designs are compatible with Manhattan-SDF. The visualization can be seen in Fig. 14. Manhattan-SDF only focuses on the floor and wall regions under the Manhattan world assumption and ignores the planar areas of objects. Combined with our proposed PPlaneSDF method, Manhattan-SDF + PPlaneSDF can not only reconstruct smoother planes on the wall and floor regions but also produce more accurate planes and precise contours of objects, e.g., the fireplace and the chair leg. The reason is that our plane regularization also works on the planar areas of objects and our keypoint guided ray sampling strategy attends to the object contours with large color variations.



(a) Manhattan-SDF (b) Manhattan-SDF + PPlaneSDF (c) Ground-truth (d) Ground-truth Image

Fig. 14 Visualization of the combination with Manhattan-SDF in scenes from the ScanNet dataset. We combine our approach with Manhattan-SDF and further improve the quality and accuracy of the reconstruction. Manhattan-SDF only focuses on the floor and wall regions under the Manhattan world assumption and ignores the planar areas of objects. Combined with our proposed PPlaneSDF method, Manhattan-SDF + PPlaneSDF can not only reconstruct smoother planes on the wall and floor regions but also produce more accurate planes and precise contours of objects, e.g., the fireplace and the chair leg, which makes a better result. The reason is that our plane regularization also works on the planar areas of objects and our keypoint guided ray sampling strategy attends to the object contours with large color variations.

4.6.1 Runtime and Memory Usage

Table 12 Comparison of training time (1 iteration) and GPU memory consumption for each components. All test experiments are conducted on a 2080Ti GPU with a batch size of 1024 rays.

Plane Reg	Weighting	Sampling	Time(s)	Mem.(MB)	F-score
		✓	0.5967	8599.24	0.640
✓			0.7171	8599.97	0.657
✓	✓		0.8145	9211.14	0.669
✓	✓	✓	0.8228	9211.85	**0.683**

The runtime of 1 iteration and the GPU memory usage of our method is listed in Table 12. The number of total iterations are the same for different methods. The plane regularization costs extra time due to the plane estimation and the plane regularization loss. The segmentation branch in the Plane Segments Fusion Based Points Weighting costs extra time and GPU memory. The Keypoint Guided Ray Sampling Strategy costs a little extra time. Comparison to the baselines can be seen in Table 13. Our method costs extra time

Table 13 Comparison of training time (1 iteration) and GPU memory consumption with the baseline methods. All test experiments are conducted on a 2080Ti GPU with a batch size of 1024 rays.

Methods	Time (s)	Mem. (MB)	F-Score
VolSDF [56]	0.5501	8561.33	0.397
VolSDF [56] + Depth	0.5895	8598.01	0.623
Manhattan-SDF [18]	0.6743	8974.46	0.681
Ours	0.8128	9210.85	**0.683**

and memory compared to VolSDF [56], VolSDF [56] + Depth and Manhattan-SDF [18].

5 Conclusion

In this work, we present the pseudo-plane regularized signed distance field for indoor scene reconstruction. We consider adjacent pixels with similar colors to be on the same pseudo-planes and regularize the signed field of the neural implicit surface representation by these planes. To estimate the plane parameters on the fly during training effectively and efficiently, we propose a two-step estimation scheme. The singed distances of the sampled points are regularized to be the distances to the planes. To remedy the noisy and inaccurate plane segments, different weights are assigned to the sampled points in plane estimation as well as the regularization loss. The weights come by fusing the plane segments from different views. The keypoint guided ray sampling strategy is also proposed to avoid redundant sampling in the planar regions and attend to the informative textured regions. The implicit network gets the better reconstruction by this strategy. Experiments show that our method not only achieves competitive reconstruction results in Manhattan scenes but also generalizes well to the non-Manhattan scenes.

Data Availability Statements

The manuscript has no associated data.

References

[1] Matan Atzmon and Yaron Lipman. Sal: Sign agnostic learning of shapes from raw data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2565–2574, 2020.

[2] Michael Bleyer, Christoph Rhemann, and Carsten Rother. Patchmatch stereo-stereo matching with slanted support windows. In *Bmvc*, volume 11, pages 1–11, 2011.

[3] Aljaz Bozic, Pablo Palafox, Justus Thies, Angela Dai, and Matthias Nießner. Transformerfusion: Monocular rgb scene reconstruction using transformers. *Advances in Neural Information Processing Systems*, 34:1403–1414, 2021.

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[5] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.

[6] Shuo Cheng, Zexiang Xu, Shilin Zhu, Zhuwen Li, Li Erran Li, Ravi Ramamoorthi, and Hao Su. Deep stereo using adaptive thin volume representation with uncertainty awareness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2524–2534, 2020.

[7] Alejo Concha and Javier Civera. Using superpixels in monocular slam. In *2014 IEEE international conference on robotics and automation (ICRA)*, pages 365–372. IEEE, 2014.

[8] Alejo Concha and Javier Civera. Dpptam: Dense piecewise planar tracking and mapping from a monocular sequence. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5686–5693. IEEE, 2015.

[9] James M Coughlan and Alan L Yuille. Manhattan world: Compass direction from a single image by bayesian inference. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 941–947. IEEE, 1999.

[10] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer*

vision and pattern recognition, pages 5828–5839, 2017.

[11] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12882–12891, 2022.

[12] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE transactions on pattern analysis and machine intelligence*, 40(3):611–625, 2017.

[13] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International journal of computer vision*, 59(2):167–181, 2004.

[14] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE transactions on pattern analysis and machine intelligence*, 32(8):1362–1376, 2009.

[15] David Gallup, Jan-Michael Frahm, and Marc Pollefeys. Piecewise planar and non-planar stereo for urban scene reconstruction. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 1418–1425. IEEE, 2010.

[16] Andreas Geiger, Martin Roser, and Raquel Urtasun. Efficient large-scale stereo matching. In *ACCV (1)*, pages 25–38, 2010.

[17] Michael A Gennert. Brightness-based stereo matching. In *1988 Second International Conference on Computer Vision*, pages 139–140. IEEE Computer Society, 1988.

[18] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5511–5520, 2022.

[19] Sunghoon Im, Hae-Gon Jeon, Stephen Lin, and In So Kweon. Dpsnet: End-to-end deep plane sweep stereo. *arXiv preprint arXiv:1905.00538*, 2019.

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

[21] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics (ToG)*, 32(3):1–13, 2013.

[22] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[23] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[24] Uday Kusupati, Shuo Cheng, Rui Chen, and Hao Su. Normal assisted stereo depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2189–2199, 2020.

[25] Boying Li, Yuan Huang, Zeyu Liu, Danping Zou, and Wenxian Yu. Structdepth: Leveraging the structural regularities for self-supervised indoor depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12663–12673, 2021.

[26] Haotong Lin, Sida Peng, Zhen Xu, Hujun Bao, and Xiaowei Zhou. Efficient neural radiance fields with learned depth-guided sampling. *arXiv preprint arXiv:2112.01517*, 2021.

[27] Chen Liu, Kihwan Kim, Jinwei Gu, Yasutaka Furukawa, and Jan Kautz. Planercnn: 3d plane detection and reconstruction from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4450–4459, 2019.

[28] Chen Liu, Jimei Yang, Duygu Ceylan, Ersin Yumer, and Yasutaka Furukawa. Planenet: Piece-wise planar reconstruction from a single rgb image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2579–2588, 2018.

[29] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2019–2028, 2020.

[30] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM siggraph computer graphics*, 21(4):163–169, 1987.

[31] Paul 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 *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. Ieee, 2007.

[32] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4460–4470, 2019.

[33] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

[34] 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 *European conference on computer vision*, pages 414–431. Springer, 2020.

[35] 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 *2011 10th IEEE international symposium on mixed and augmented reality*, pages 127–136. Ieee, 2011.

[36] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3504–3515, 2020.

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

[38] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174, 2019.

[39] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[40] Barbara Roessle, Jonathan T Barron, Ben Mildenhall, Pratul P Srinivasan, and Matthias Nießner. Dense depth priors for neural radiance fields from sparse input views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12892–12901, 2022.

[41] Andrea Romanoni and Matteo Matteucci. Tapa-mvs: Textureless-aware patchmatch multi-view stereo. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10413–10422, 2019.

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

[43] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *European conference on computer vision*, pages 501–518. Springer, 2016.

[44] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[45] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2930–2937, 2013.

[46] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. *Advances in Neural Information Processing Systems*, 32, 2019.

[47] Jiaming Sun, Yiming Xie, Linghao Chen, Xiaowei Zhou, and Hujun Bao. Neuralrecon: Real-time coherent 3d reconstruction from monocular video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15598–15607, 2021.

[48] Shang Sun, Yunan Zheng, Xuelei Shi, Zhenyu Xu, and Yiguang Liu. Phi-mvs: Plane hypothesis inference multi-view stereo for large-scale scene reconstruction. *arXiv preprint arXiv:2104.06165*, 2021.

[49] Bing Wang, Lu Chen, and Bo Yang. Dm-nerf: 3d scene geometry decomposition and manipulation from 2d images. *arXiv preprint arXiv:2208.07227*, 2022.

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

[51] Yi Wei, Shaohui Liu, Yongming Rao, Wang Zhao, Jiwen Lu, and Jie Zhou. Nerfingmvs: Guided optimization of neural radiance fields for indoor multi-view stereo. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5610–5619, 2021.

[52] Oliver Woodford, Philip Torr, Ian Reid, and Andrew Fitzgibbon. Global stereo reconstruction under second-order smoothness priors. *IEEE transactions on pattern analysis and machine intelligence*, 31(12):2115–2128, 2009.

[53] Qingshan Xu and Wenbing Tao. Planar prior assisted patchmatch multi-view stereo. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12516–12523, 2020.

[54] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *Proceedings of the European conference on computer vision (ECCV)*, pages 767–783, 2018.

[55] Yao Yao, Zixin Luo, Shiwei Li, Tianwei Shen, Tian Fang, and Long Quan. Recurrent mvsnet for high-resolution multi-view stereo depth inference. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5525–5534, 2019.

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

[57] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry

and appearance. *Advances in Neural Information Processing Systems*, 33:2492–2502, 2020.

[58] Wei Yin, Yifan Liu, Chunhua Shen, and Youliang Yan. Enforcing geometric constraints of virtual normal for depth prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5684–5693, 2019.

[59] Zehao Yu, Lei Jin, and Shenghua Gao. P^{2}net: Patch-match and plane-regularization for unsupervised indoor depth estimation. In *European Conference on Computer Vision*, pages 206–222. Springer, 2020.

[60] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. *arXiv preprint arXiv:2206.00665*, 2022.

[61] Zehao Yu, Jia Zheng, Dongze Lian, Zihan Zhou, and Shenghua Gao. Single-image piecewise planar 3d reconstruction via associative embedding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1029–1037, 2019.