Supplementary Material for
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

We provide more implementation details, including training and inference (Sec. A), network architecture (Sec. B), sphere tracking (Sec. C), network training (Sec. D), the evaluation matrices (Sec. E), more qualitative results (Sec. F and G), and limitations (Sec. H) in the supplementary material.

A. Training and Inference Details

Training device and time. All experiments are conducted on a single RTX3090, and it takes about 8 hours to train each scene. Note that the average rendering time to segment each view is 0.925s thanks to the usage of sphere tracing, which is much faster than using volume rendering (67.2s). Thus, even for the largest number of views in all evaluated scenes, *i.e.*, 477, it takes only 7.35 min to render all images and all the time used for segmentation is about 30 min, which is only a small fraction of the training time (8 h).

Hyper-parameters. The weights of eikonal loss λ_{eik}, depth loss λ_d and super-plane loss λ_{plane} used in the training process are 1.0, 1.0, and 0.1 respectively. The auto-filtering threshold α is set to 0.9. When evaluating the performance of planar segmentation, the non-planar edge region detection threshold γ is set to 0.85 and set to 0.9 during training to ensure the robustness of edge region detection. Besides, when evaluating the plane segmentation performance, we apply a median blur filter with kernel size 9×9 to smooth the input images.

B. Network Details

Our geometric network g_θ is an 8-layer MLP with a hidden dimension of 256, while the color network f_ϕ is another 4-layer MLP with the same number of hidden dimensions. The complete network architecture is shown in Fig. 1.

C. Sphere Tracing

We adopt the sphere tracing method [1] to produce the normal maps, where the procedure is shown in Algorithm. 1.

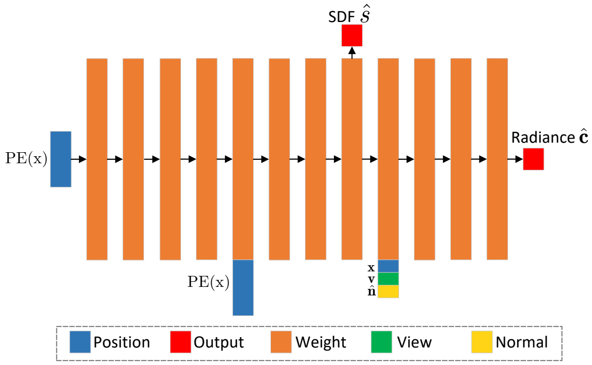

Figure 1. **Network Architecture.** Our network outputs SDF value $\hat{s}$ and radiance value $\hat{\mathbf{c}}$.

D. An Iterative Training Procedure

We summarize the whole training procedure mentioned in Sec. 3 of the main paper in Algorithm. 2. Our method optimizes the surface reconstruction result in an iterative way, where the super-plane segmentation and surface reconstruction quality are progressively improved.

To better understand the effect of our iterative training procedure, we visualize the surface normal map and the segmentation after each update of the segmentation masks. As shown in Fig. 3, the quality of the reconstruction and planar segmentation results are improved iteratively. Fig. 3 also shows the non-plane edge region detection results (black areas in super-plane segmentation maps) during training.

E. Evaluation Metrics

The definition of metrics used for 3D reconstruction evaluation is shown in Tab. 1. The plane and pixel recalls [2] used for evaluating plane reconstruction results are defined as follows: we first treat a ground truth plane as correctly predicted if it satisfies the following two conditions: (1) There exists a predicted plane that has an IoU larger than 0.5 with it. (2) The average depth difference between the

Algorithm 1 Adapted sphere tracing algorithm for a camera ray $\mathbf{x}_i = \mathbf{o} + d_i\mathbf{v}$ over the signed distance fields g_θ.

```
Input: max iteration $N$
Output: surface position $\mathbf{x}_N$
Initialize $n = 0$, $d_0 = 0$, $\mathbf{x}_0 = \mathbf{o}$.
while $n < N$ do:
    Calculate the SDF value $\hat{\mathbf{s}}_n$ of point $\mathbf{x}_n$: $\hat{s}_n = g_\theta(\mathbf{x_n})$
    $d_{n+1} \leftarrow d_n + \hat{\mathbf{s}}_n$
    $\mathbf{x}_{n+1} \leftarrow \mathbf{o} + d_{n+1}\mathbf{v}$, $n \leftarrow n + 1$.
end while
```

Algorithm 2 Training procedure of S^{3}P.

```
Input: max training iteration $M$, update interval for super-plane segmentation $t$
for i = 1 to $N$ do
    if $i \,\%\, t == 0$ then
        Update normal maps and super-plane segmentation masks
        Clean up segmentation masks with the auto-filtering and non-plane region detection strategies
        Update super-plane normal
    end if
    if $i \geq t$ then
        Train with super-plane loss in Eq. 5 of the main paper
    else
        Train without super-plane loss                              ▷ Initialize geometric structure
    end if
end for
```

ground truth plane and the corresponding predicted plane is smaller than a threshold varying from 0.05m to 0.6m with an increment of 0.05m. Then, the plane recall is defined as the percentage of correctly predicted planes, and the pixel recall is the percentage of pixels within all correctly predicted planes.

Metric	Definition
Acc	$\text{mean}_{\mathbf{p}\in P}(\text{min}_{\mathbf{p}^*\in P^*} \lvert\lvert\mathbf{p} - \mathbf{p}^*\rvert\rvert_1)$
Comp	$\text{mean}_{\mathbf{p}^*\in P^*}(\text{min}_{\mathbf{p}\in P} \lvert\lvert\mathbf{p} - \mathbf{p}^*\rvert\rvert_1)$
Prec	$\text{mean}_{\mathbf{p}\in P}(\text{min}_{\mathbf{p}^*\in P^*} \lvert\lvert\mathbf{p} - \mathbf{p}^*\rvert\rvert_1 < .05)$
Recal	$\text{mean}_{\mathbf{p}^*\in P^*}(\text{min}_{\mathbf{p}\in P} \lvert\lvert\mathbf{p} - \mathbf{p}^*\rvert\rvert_1 < .05)$
F-score	$\frac{2\cdot\text{Perc}\cdot\text{Recal}}{\text{Prec}+\text{Recal}}$

Table 1. **Evaluation Metrics.** P and P^* are the point clouds sampled from predicted and ground truth mesh.

F. Comparison of Plane and Super-plane

To better understand the difference between plane and super-plane segmentation, we visualize the results produced by our method in Fig. 4. It can be seen that our super-planes are typically larger than individual planes since parallel As mentioned in the main paper, planes are grouped together, which provides more accurate averaged normals and thus facilitates the training process.

G. Plane Reconstruction Visualization

We show the piece-wise planar reconstruction results in Fig 5. It can be seen that our method can produce accurate planar segmentation and further recover accurate planar surfaces.

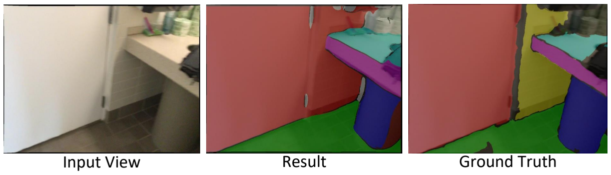

Figure 2. **Plane Segmentation Failer Case.** Our method may fail to separate two closely connected parallel planes.

H. Limitations

1) For some thin non-planar regions, if the baseline model of our method cannot recover it, adding super-plane constraints does not help. 2) When the angle between a pixel's surface normal and the super-plane normal is very small, the auto-filtering strategy may fail. However, We found the reconstruction results are not largely affected because it is acceptable to treat pixels with the tiny surface normal angle differences as the same plane. We will add

these analyses to the revised paper. 3) The planar segmentation result may not be able to separate two closely connected parallel planes as shown in Fig. 2. This may be solved by separating the planes using an edge detection algorithm.

References

[1] John C Hart. Sphere tracing: A geometric method for the antialiased ray tracing of implicit surfaces. *The Vis. Comput.*, 1996. 1

[2] Chen Liu, Jimei Yang, Duygu Ceylan, Ersin Yumer, and Yasutaka Furukawa. Planenet: Piece-wise planar reconstruction from a single rgb image. In *CVPR*, 2018. 1

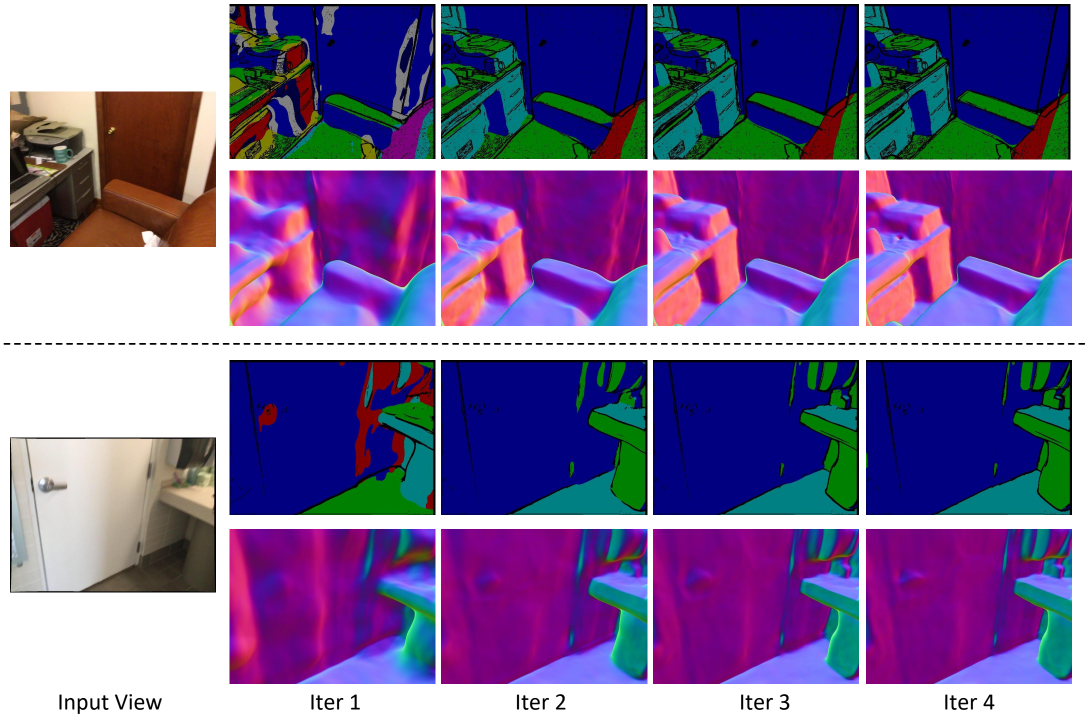

Figure 3. **Visualization of Iterative Refinement Procedure.** The first and second rows in each group represent the super-plane segmentation and surface normal results after each update. The black areas in the super-plane segmentation maps represent the area where the non-planar edge regions are detected.

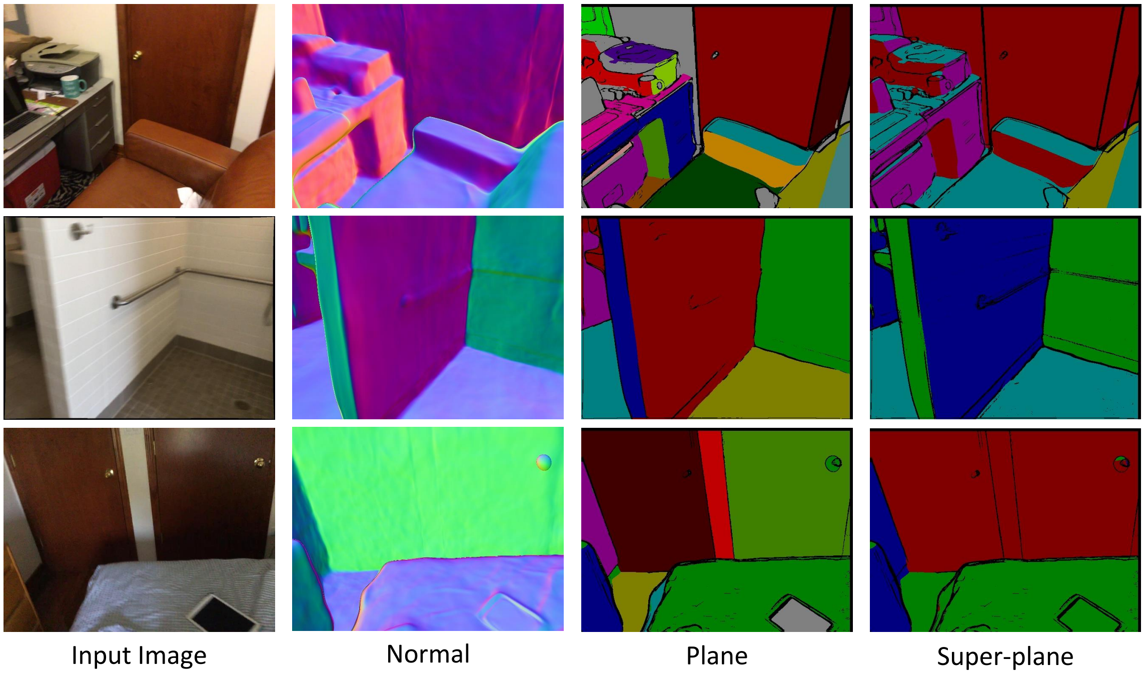

Figure 4. **Comparison of Plane and Super-plane.** Different colors in the segmentation map represent different clusters. Our super-plane segmentation groups the pixels belonging to parallel planes into the same cluster.

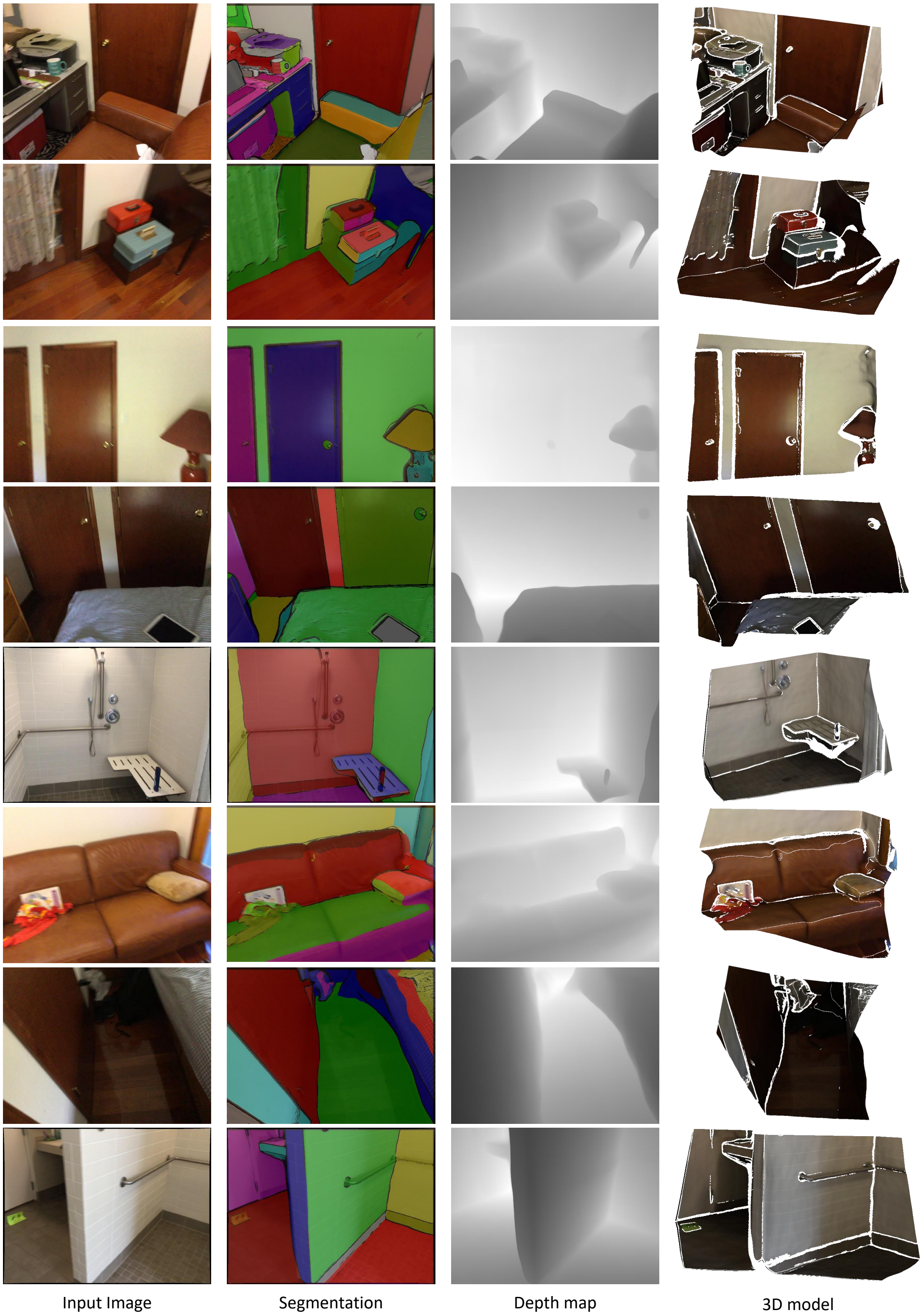

Figure 5. **Unsupervised Piece-wise Planar Reconstruction Results by S^{3}P.** The depth map is generated using the sphere tracing algorithm mentioned in Sec. C.