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

Supplementary Material

A. Implementation Details

Our implementation is based on FreeNeRF's codebase[*]. We use the plain Mip-NeRF [1] as our backbone and the maximum input frequency for positional encoding of coordinates is 16. The learning rate is warmed up with a multiplier of 0.01 in the first 512 iterations.

B. Datasets and Metrics

ScanNet Dataset. We use three ScanNet [2] rooms provided by DDP [4] for our experiments, which are scene0710_00, scene0758_00 and scene0781_00. The scene0758_00 contains 20 training views and the other scenes contain 18 training views. All scenes are tested with 8 views. And the image size is 468×624.

Replica Dataset. Experiments on the Replica [5] dataset use trajectories from eight scenes rendered by NICE-SLAM [6]. They are office0, office1, office2, office3, office4, room0, room1 and room2. Each raw trajectory contains 2000 frames, and we select 20 frames for training at 100 frame intervals starting at the 0-th frame, and test images at 100 frame intervals starting at the 50-th frame. All images are downsampled to size 340×600.

Metrics. We normalize the RGB values between 0 and 1 to calculate the PSNR, SSIM, and LPIPS scores. The formula of PSNR is $-10 \cdot \log_{10}(\text{MSE})$. We compute the SSIM score using the structural_similarity function from the scikit-image library[†]. In addition, LPIPS uses the computation method based on an AlexNet [3] provided in the open-source library PerceptualSimilarity[‡].

C. Geometry Consistency Prior Generation

Global Geometry Consistency Prior. Each camera's corresponding region in 3D space is a square frustum, consisting of eight vertices. Four of them lie in the near plane and the remaining four in the far plane. We compute the 3D frustums of all training views and check if the vertices of each frustum are inside the other. The purpose of this is to find pairs of images that might have overlapping regions. And we only perform image matching on such image pairs. We use the LoFTR implementation in Kornia[§] for image matching. Then we calculate the epipolar line of each keypoint and calculate the distance from the corresponding point to the epipolar line. A match with an excessive distance is obviously wrong, and we filter these wrong matches using a threshold of 3 pixels. Next, we use least squares to compute the 3D point that minimizes the distance to the rays where the keypoint pair is located, and discard the points with distances greater than 5cm. Finally all remaining points are projected onto their own rays to obtain our geometric prior D_k.

Hierarchical Geometry Consistency Prior. We estimate the monocular depth using the dpt_hybrid_384 model from the MiDaS repository[¶]. Since the output of DPT is relative inverse depth, we use the formula $1-(D_{\text{dpt}}-V_{\text{min}})/(V_{\text{max}}-V_{\text{min}})$ to reverse the depth and normalize it. The V_{min} and V_{max} are the minimum and maximum values of the DPT output, respectively.

Model	setting	PSNR ↑	SSIM ↑	LPIPS ↓	RMSE ↓
(1)	$ps = 8$	20.92	0.709	0.208	0.213
(2)	$ps = 32$	20.82	0.687	0.218	0.229
(3)	$M = 16$	20.50	0.680	0.221	0.233
(4)	$M = 64$	21.01	0.713	0.203	0.205
(5)	$\Delta = 0.05$	20.62	0.709	0.211	0.221
(6)	$\Delta = 0.2$	20.97	0.713	0.204	0.210
(7)	default	21.03	0.719	0.209	0.213

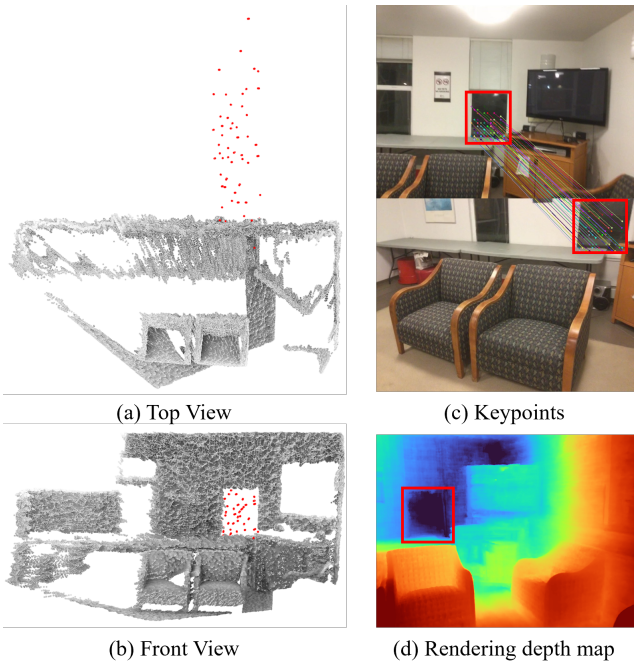
Table 1. ScanNet results for different hyperparameters. The default model uses the settings from the paper, *i.e.* $ps = 16$, $M = 32$, and $\Delta = 0.1$. Models (1) to (6) ablate a single variable separately.

D. Hyperparameters

We perform additional ablation experiments for the patch size in the global geometry consistency prior, the group number in the group depth ranking loss, and the margin in the ray weight mask regularization. The experimental results with different hyperparameter settings are demonstrated in Tab. 1.

Patch Size. The results with $ps = 8$ are very close to those with $ps = 16$. However, when the patch size is increased by 32, there is a little performance degradation because the less precise global geometry consistency constraints are used

[*] FreeNeRF
[†] scikit-image
[‡] PerceptualSimilarity
[§] LoFTR
[¶] MiDaS



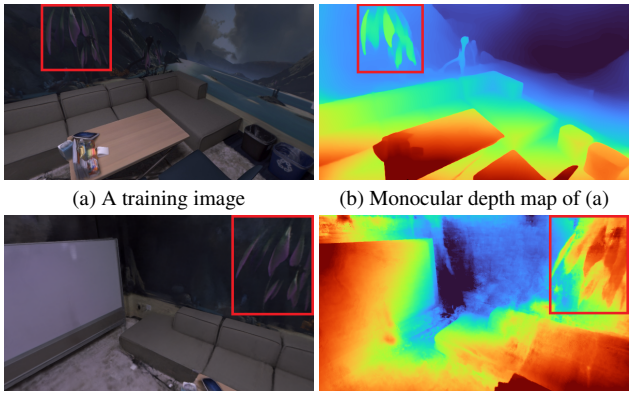
(a) Top View (c) Keypoints

(b) Front View (d) Rendering depth map

Figure 1. The effect of global geometric consistency errors on NeRFs. (a) and (b) are the top and front views of the point cloud, respectively, where the gray points are from the sensor and the red points are from image matching. (c) is a visualization of correspondences. (d) is a rendering depth map of a test view. The rendering depth map shows that the transparent glass is much deeper than the surrounding wall. There is a typical mistake.

setting	PSNR↑	SSIM↑	LPIPS↓	RMSE↓
40-views	35.12	0.921	0.052	0.186
30-views	33.55	0.901	0.054	0.208
20-views	32.14	0.902	0.053	0.217
10-views	25.53	0.787	0.115	0.250

Table 2. Quantitative results at different levels of sparseness.



(a) A training image (b) Monocular depth map of (a)

(c) A rendering image (d) A rendering depth map

Figure 2. The effect of hierarchical geometric consistency errors on NeRFs. (a) is an image from the training set and (b) is the monocular depth estimation result of (a). (c) and (d) are the rendering image and depth map of a test view, respectively. Since the monocular depth estimation network incorrectly predicts the depth of the leaf-like texture on the wall, it causes the NeRFs to produce incorrect rendering depth as well.

over a larger area. It indicates that the patch size should not be set too large.

Group Number. In the experiment with $M = 16$, the performance of both novel view synthesis and rendering depth shows a significant degradation. Since only a few groups are used, there will be depth values with large variance within the same group, and these depths cannot be ranked against each other, which limits the representation of the model. And in the experiment with $M = 64$, the results simliar to the default settings are got, and even the depth RMSE metric is better than the default experiment. In fact, increasing the number of groups is equivalent to increasing the discrimination of depth. However, the group depth ranking loss needs to compute a matrix of size $M \times M$, and we also have to balance between the size of M and the amount of computation.

Margin. Comparing the values of Δ to 0.05, 0.1, and 0.2 respectively, we find that the performance degrades at the strictest margin, *i.e.*, $\Delta = 0.05$. It is consistent with our analysis about ray weight mask regularization that overly strict threshold settings can produce extreme cases where there is a risk of negative optimization of this loss.

E. Effect of view sparsity

To further explore the effect of training view sparsity on performance, we conduct experiments on view sparsity effects in *office0* and *room0* scenes from the Replica dataset, following sparse view training. The results are reported in Tab. 2, with 20-views as the paper's default setting. Observations reveal that with sufficient training views (40-views and 30-views), performance is similar. Conversely, inadequate training views covering the scene (10-views) lead to a notable performance decline.

F. Limitations

Although our P^{2}NeRF achieves stable and robust few-shot NeRFs by leveraging prior knowledge from pretrained models, we find that biases from the pretrained models are also introduced into the NeRFs.

The first effect comes from the coarse point cloud. Fig. 1 shows a typical bad case. Because of the strong specular reflection on the glass windows, the point cloud from image matching has a large error. After warming up, the volume density distribution in this region presents a sickly pattern. And since the true depth of the region is also large in the

training view, the hierarchical geometry consistency is not able to constrain it effectively, which ultimately leads to a significant offset in the depth of the glass window.

The second limitation results from the bias of monocular depth estimation. Since monocular depth estimation suffers from image texture, the hierarchical geometry consistency constraint will distill it into NeRFs in the few-shot setting. We visualize this phenomenon in Fig. 2. P^2NeRF learns the wrong knowledge when distilling relative ranking relations from monocular depth. Since the training view of the scene is sparse, the NeRFs themselves can't optimize this mistake away and eventually react in the rendering results.

References

[1] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021. 1

[2] 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. 1

[3] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 1

[4] Barbara Roessle, Jonathan T Barron, Ben Mildenhall, Pratul P Srinivasan, and Matthias Nießner. Dense depth priors for neural radiance fields from sparse input views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12892–12901, 2022. 1

[5] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 1

[6] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R Oswald, and Marc Pollefeys. Nice-slam: Neural implicit scalable encoding for slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12786–12796, 2022. 1