NeRFVS: Neural Radiance Fields for Free View Synthesis via Geometry Scaffolds Supplementary Material

Chen Yang[1*], Peihao Li[3], Zanwei Zhou[1], Shanxin Yuan[2], Bingbing Liu[2],
Xiaokang Yang[1], Weichao Qiu[2], Wei Shen[1†]
[1] MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University
[2] Huawei Noah's Ark Lab [3]Tsinghua University
{ycyangchen,SJTU19zzw,xkyang,wei.shen}@sjtu.edu.cn; lph21@mails.tsinghua.edu.cn;
{shanxin.yuan,liu.bingbing,qiuweichao1}@huawei.com

1. Datasets

1.1. ScanNet [1]

ScanNet is a widely used indoor scene novel view synthesis dataset. Since it is a real-capture dataset, we cannot acquire ground truth images from arbitrary views. To enable the ScanNet dataset to support the evaluation of FVS tasks, we design a heuristic split algorithm and split the raw dataset into the training, interpolation, and extrapolation sets. The split algorithm ensures the views in the interpolation set are near the training views and the extrapolation views are significantly different from the training views.

For training image selection, we first uniformly sample 10% views from the raw image sequence for each scene following the setting of [3]. We directly select the middle frames among training views for the interpolation set. As for the specific extrapolation set, we manually split 6% views from the training views. These views are less overlapped with the training pixels. We allocate their 8 adjacent views together with themselves as the extrapolation views. To better illustrate the split, we make a simplified example with 0-100 views in the trajectory:

- Training view: 0, 10, 20, 30, 70, 80, 90, 100
- Interpolation: 5, 15, 25, 75, 85, 95
- Extrapolation: 46, 47, 48, 49, 50, 51, 52, 53, 54

The following scenes are used for evaluation: 0050_00, 0084_00, 0580_00 and 0616_00.

1.2. Barbershop

The images from ScanNet often contain motion blur, dark borders, noise, and pose inaccuracy. These shortcomings degrade the rendering quality of NeRFs and hinder our diagnosis among FVS tasks. To this end, we designed a synthetic dataset named Barbershop. Barbershop is a classic open-access demo scene provided by "Blender Animation Studio" in blender website , which contains many small objects, complex reflection, and sizeable low-texture area. We use Blender software to render this indoor scene with the cycles engine. To generate this indoor scene dataset, we convert the Barbershop to a interactive scene where users can move and capture images freely as if capturing real scenes. This way, we collect one trajectory with 543 images and uniformly render 1152 images for the extrapolation set. For the interpolation set, we uniformly select one-sixth views among the trajectory as interpolation set and others as training set. For extrapolation views generation, we first split the whole scene into a $4 \times 4 \times 3$ grid, and then put cameras on the grid corners. On each location on the grid corner, we render $3 \times 8 = 24$ images from different camera directions. The "3" represents the cameras looking at 45 degrees diagonally upward, middle and 45 degrees diagonally downward. The "8" represents the cameras looking evenly in eight directions on the plane parallel to the floor, as shown in Fig. 1.

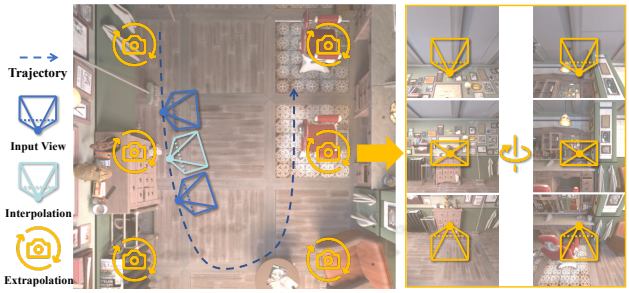

Figure 1. **Data split of the Barbershop dataset.** The extrapolation views are evenly sampled among space.

[*] Work done during an internship at Huawei Noah's Ark Lab.
[†] Corresponding Author.

https://www.blender.org/download/demo-files/

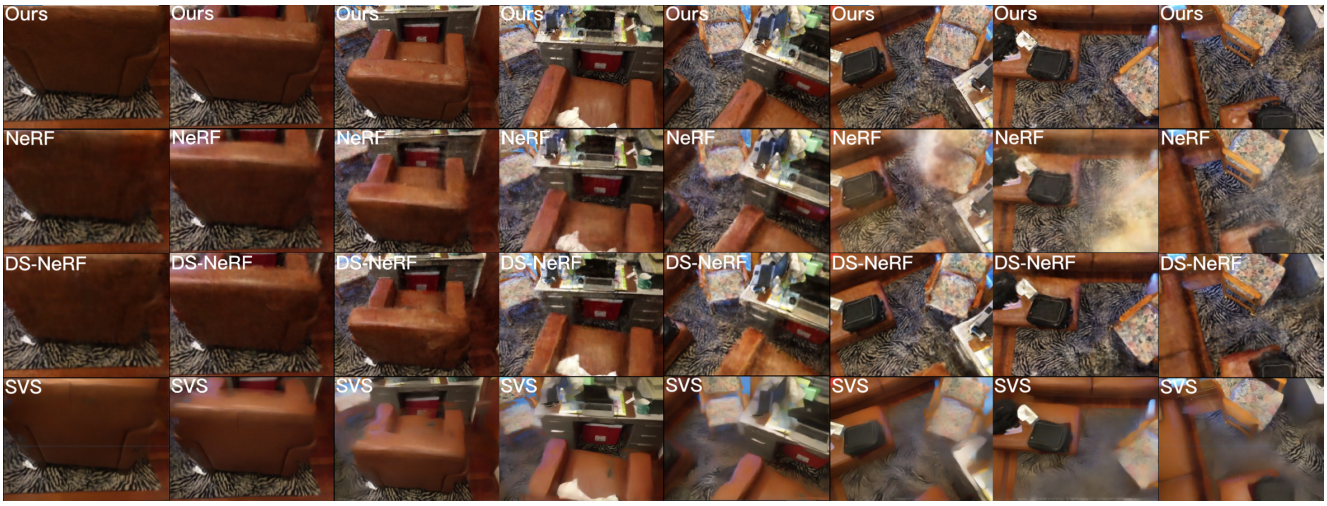

Figure 2. **Video qualitative results.** We show a continuous sequence of extrapolation results to assist evaluation. Please see the back of the sofa which is a less observed region, and the carpet which contains high-frequency details.

2. Implementation Details

2.1. NeRFVS

View Coverage Map Our NeRFVS leverages the geometry scaffold to generate view coverage maps $\mathcal{V}$ (Alg. 1). Specifically, we first render the depth maps $\mathcal{D}$ of training views using camera parameters and the scaffold. Then we use the depth maps to reconstruct a point cloud $\mathcal{P} \in \mathbb{R}^3$. Each point P_i in $\mathcal{P}$ is projected to each view in the training views and gets the corresponding coordinates (u_i, v_i) in the image plane and the corresponding depth z_i. Points with (u_i, v_i) out of image range or $z_i <= 0$ will be discarded. Considering occlusion, we drop the points whose difference between the re-projection depth z_i and the rendered depth D'_i is larger than 1e-2. Finally, we add one hit to the rounded (u_i, v_i) in the $\mathcal{V}$. In this manner, we obtain the view coverage information from the geometry scaffold. We show the view coverage information among the four scenes in Fig. 3.

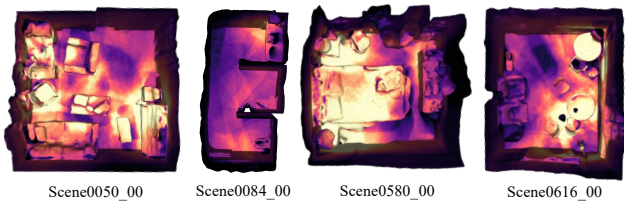

Figure 3. **View coverage of the four scenes in ScanNet.**

Model Architecture Our model is based on NeRF-pytorch [9]. The frequency in positional encoding for point positions is 10, and for view directions is 4. Unlike the vanilla NeRF that adopts a ReLU activation to produce density σ, we choose a softplus activation instead, achieving more stable optimization. In all experiments, we uniformly sample 128 points for the coarse stage and 128 points for the fine stage. We use the loss weights $\lambda_{color} = 1$, $\lambda_{\text{d}} = 0.5$, $\lambda_{\text{w}} = 0.1$ and $\lambda_{\text{c}} = 0.01$ across all ScanNet scenes. For the Barbershop, we use $\lambda_{\text{c}} = 0.075$ for a stronger regularization on the color prediction. We use $\alpha = 9$, $\lambda_{max} = 5$, and $\beta = 0.1$ in all experiments except for relative ablation.

Algorithm 1: View Coverage Map Generation

Input : training camera projection matrix $\mathcal{M}$, geometry scaffold $\mathcal{G}$, view coverage map $\mathcal{V}$

$\mathcal{V} \leftarrow 0^{N \times H \times W}$;
$\mathcal{D} \leftarrow$ rendered depth maps from $\mathcal{G}$;
$\mathcal{P} \leftarrow$ reconstructed pointcloud with $\mathcal{M}$ and $\mathcal{D}$;
for $\{V_j, M_j, D_j\}$ *in* $\{\mathcal{V}, \mathcal{M}, \mathcal{D}\}$ **do**
  for P_i *in* $\mathcal{P}$ **do**
    $(u_i, v_i, z_i) \leftarrow$ project P_i with M_j;
    if (u_i, v_i) *not in valid image region or* $z_i <= 0$ **then**
      continue;
    end
    $D'_i \leftarrow$ bi-linear interpolation of D_j with coordinate (u_i, v_i);
    $\Delta D \leftarrow |D'_i - z_i|$
    /* judge occlusion, $\epsilon = 1e-2$ */
    if $\Delta D > \epsilon$ **then**
      continue;
    end
    $(u'_i, v'_i) \leftarrow$ round the float (u_i, v_i) to be int;
    $V_j[u'_i, v'_i]$ += 1;
  end
end
return $\mathcal{V}$;

A relaxing stage is adopted within the last 10% of the training iterations. We only apply the $\mathcal{L}_{\text{color}}$ on rays whose view coverage is larger than α in this stage. This way, we fine-tune the fully observed regions with only the photometric loss. While on few shot regions, the variance regularization and depth constraint are still applied.

2.2. NeRFVS (NGP)

Our NeRFVS (NGP) is mainly built on the torch-NGP and tiny-cuda-nn [4]. Specifically, we use the 16 levels grids with 2 dimension features per entry. The hash table size is 2^{19}. We apply spherical harmonics encoding with 4 degrees for the view direction encoding. Since the convergence of instant-NGP is much faster than vanilla NeRF, we only use 50k iterations compared to the 200k iterations of NeRF. With super-fast instant-NGP, our method can achieve fast indoor scene free navigation.

3. Baseline Method Details

We compared our results with several state-of-the-art indoor synthesis methods, and here we present the implementation details of these methods.

3.1. Dense Depth Priors [6]

We run COLMAP [7] with training images to get sparse depth and use the officially released depth completion network to compute dense depth priors, which is pre-trained on the whole ScanNet dataset. Following the original setting, we set the depth loss weight to 0.004 for ScanNet and Barbershop scenes.

3.2. Stable View Synthesis [5]

Stable View Synthesis leverages a geometry scaffold from the COLMAP MVS. We follow its data generation procedure and use the training, interpolation, and extrapolation images together to generate data. SVS applies a perceptual loss for optimization, which is similar to the computation of the LPIPS score. Thus the rendered images are with high LPIPS scores compared to other methods.

We show more qualitative comparison results among extrapolation views in Fig. 4. The images synthesized by SVS are inconsistent with the scene, resulting in artifacts and distortions, especially in inaccurate geometry areas. Our NeRFVS treat the geometry scaffold as unreliable, using multi-view consistency to assist the scene reconstruction among fully observed regions. As for few-shot regions, we mainly rely on the geometry priors and variance regularization. In this manner, NeRFVS significantly reduces the inconsistency, resulting in high fidelity results.

3.3. NerfingMVS [8]

NerfingMVS run COLMAP MVS with both training images and test images, which is not reasonable since test images are unreachable in practice. Therefore, We run COLMAP only with training images to obtain depth priors for its depth network's training, which makes NerfingMVS perform a little worse but more practical.

Figure 4. **Qualitative comparison with SVS.** The images synthesized by SVS are lower fidelity compared with ours, e.g., the window blind in the first row is twisted.

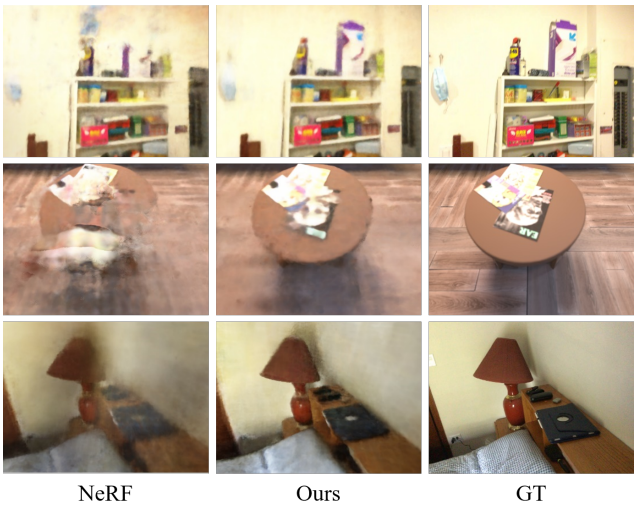

Figure 5. **Performance on few-shot regions.** The wall and the downside of the table are observed with only several times.

3.4. DS-NeRF [2]

Similar to our implementation in Dense Depth Priors, we compute sparse depth priors with training images using COLMAP. The depth loss weight is set to 0.1 for both ScanNet dataset and Barbershop dataset.

4. Performance on few-shot regions

We show the effectiveness of our NeRFVS on few-shot regions (the wall and the downside of the table) compared with the NeRF in Fig. 5. Our NeRFVS significantly improve the rendering quality by reducing the ambiguity among these regions.

References

[1] 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

[2] 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. 3

[3] 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. 1

[4] Thomas Müller. tiny-cuda-nn, 4 2021. 3

[5] Gernot Riegler and Vladlen Koltun. Stable view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12216–12225, 2021. 3

[6] 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. 3

[7] 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. 3

[8] 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. 3

[9] Lin Yen-Chen. Nerf-pytorch. https://github.com/yenchenlin/nerf-pytorch/, 2020. 2