Dense Depth Priors for Neural Radiance Fields from Sparse Input Views Supplementary Material

Barbara Roessle[1] Jonathan T. Barron[2] Ben Mildenhall[2] Pratul P. Srinivasan[2] Matthias Nießner[1]

[1]Technical University of Munich [2]Google Research

1. Datasets

1.1. ScanNet [3]

Motion Blur Detection We consider motion blur when sampling a small subset of images to be used in NeRF: From each window of n consecutive video frames the sharpest one is selected according to the following metric, where high values indicate sharpness: first, an image is converted to grayscale, then it is convolved with a discrete Laplacian kernel; finally, the variance is computed. n is set to 10 or 20, depending on how densely the video samples the scene.

Train/Test Image Selection After removing images with severe motion blur, we consider the following criteria: 1) SfM successfully registers the set of images. 2) Surfaces to be reconstructed are observed from at least one input view. In practice, images are removed if their content is visible by other images and the remaining set fulfils 1). This way, 22% of the train pixels are not observed by any other train view, 31% are observed by one other, 47% by two or more. Test views have on average 66% overlap with their most overlapping train view.

Image Resolution The image resolution is 468×624 after downsampling and cropping dark borders from calibration.

Test Scenes We ensure that the test scenes are complete, sufficiently large rooms. The following scenes are used for evaluation:

- scene0710_00
- scene0758_00
- scene0781_00

SfM Quality on Few Views Figure 1 shows the mean absolute error (MAE) of the SfM points against the sensor depth. It is computed on the 6291 points from the three ScanNet evaluation scenes. The maximal error is 5.85m. We do not filter the COLMAP SfM output, i.e., all points are projected to the corresponding input views and used as input to the depth completion.

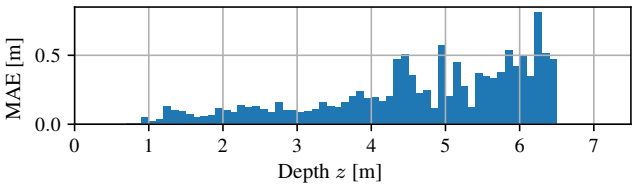

Figure 1. SfM depth error on ScanNet.

1.2. Matterport3D [1]

Train/Test Image Selection Similar to ScanNet, it is ensured that surfaces are observed from at least one input view. 25% of the train pixels are not observed by any other train view, 45% are observed by one other, 30% by two or more. Test views have on average 67% overlap with their most overlapping train view.

Image Resolution The image resolution is 504×630 after downsampling and cropping dark borders from calibration.

Test Scenes We avoid unbounded open space, which is challenging for NeRF approaches. The following scenes are used for evaluation:

- Region 5, house VzqfbhrpDEA
- Region 2, house Vvot9Ly1tCj
- Region 19, house Vvot9Ly1tCj

2. Impact of Sparse Depth Density

We investigate the impact of the sparse depth density on Matterport3D by decreasing it from 0.1% to 0.05% and 0.01% (Tab. 1). While reduced sparse depth lowers performance, it clearly shows that depth completion increases the value of very sparse depth input: With just one tenth of the sparse depth our method still performs better, than the version without completion. Despite 0.01% being very sparse—just 32 points per image on average—we expect that using monocular depth estimation is challenging as view-consistent depth is needed.

Method	Sparse depth density	PSNR↑	SSIM↑	LPIPS↓	Depth RMSE ↓
Ours w/o completion	0.10%	16.90	0.615	0.521	0.427
Ours	0.10%	**18.33**	**0.673**	**0.402**	**0.114**
Ours	0.05%	18.10	0.662	0.414	0.136
Ours	0.01%	17.99	0.662	0.437	0.140

Table 1. Impact of sparse depth density on Matterport3D. Depth RMSE is in meters.

	ScanNet	Matterport3D
Ours w/o Completion	1.0	0.25
Ours w/o Uncertainty	0.001	0.007
Ours w/o GNLL	0.04	0.03
Ours w/o Latent Code	0.003	0.007
Ours	0.003	0.007

Table 2. Depth loss weights λ.

3. Implementation Details

3.1. Our Method

Radiance Fields Our model architecture is based on NeRF [7]. The encoded position $\gamma(\mathbf{x})$ is provided as input to the first of 8 layers as well as to the fifth, by concatenating it with the activations from the fourth layer. Layers 1–8 each have 256 neurons and ReLU activations. The output of layer 8 is passed through a single layer with softplus activation to produce density σ. The output of layer 8 is also passed through a 256-channel layer without activation, whose output is concatenated with the viewing direction $\mathbf{d}$ and the latent code $\boldsymbol{\ell}$. The concatenated vector is fed to a 128-channel layer with ReLU activation, before the final layer producing the color $\mathbf{c}$. The latent codes $\boldsymbol{\ell}$ have a size of 4 on ScanNet and 16 on Matterport3D. Due to the different characteristics of the depth input on the two datasets, a suitable depth loss weight λ is determined for each approach and dataset and used across all scenes of the same dataset (Tab. 2).

Depth Completion The depth completion network is based on the architecture from Cheng *et al*. [2]. We use a ResNet-18 [5] encoder and add a second upsampling branch for uncertainty estimation. It equally consists of up-projection layers with skip connections to the same down-sampling layers as the depth prediction branch. To increase performance on very sparse input depth, both branches use a CSPN module, configured to 48 iterations in the depth branch and 24 iterations in the standard deviation branch. The depth completion network is trained at a lower resolution of 256×320 on Matterport3D, and 240×320 on ScanNet. We use the Adam optimizer [6] with a learning rate of 0.0001 and a batch size of 8. We train for 50 epochs on Matterport3D and 12 epochs on ScanNet. On Matterport3D 80 houses are used for training, 5 houses for validation, and 5 houses for testing. On ScanNet we use the provided data split. We ensure that the scenes used for NeRF are not included during training, and are instead in the test sets.

3.2. NerfingMVS [8]

The error map calculation used by NerfingMVS was not sufficiently robust to by applied to entire rooms, so to improve this baseline's performance we adapted it as follows:

Original Calculation For each input view an error map is computed by projecting the 3D points according to the depth prior to all other views, where a depth reprojection error is computed and normalized with the projected depth. The mean of the 4 smallest errors are used as values in the error map.

Problem on Entire Rooms When applying the computation on entire rooms as opposed to a local region, the projected 3D points from other views frequently lie behind the camera. As a result the computed mean is often negative. Similarly, the computation of the near and far planes of the scenes is not suited for entire rooms, leading to a negative near plane in our case. Negative near plane and negative error map content lead to invalid sampling ranges, where the far bound lies in front of the near bound. to address this, we set the near and far planes (t_n and t_f) of each scene such that all depth prior values are contained. In the error map calculation, we assign a maximal error $t_f - t_n$ for projected points that lie behind the camera. Afterwards, the error map values are still computed as the mean of the smallest 4 errors.

3.3. DS-NeRF [4]

We used the same positional encoding frequencies as described for our method in the main paper for this baseline, which improved its performance. A depth loss weight of 0.1 was suitable for the ScanNet scenes.

3.4. NeRF [7]

As in DS-NeRF, we used our own positional encoding frequencies for this baseline, which improved its performance.

References

[1] Angel X. Chang, Angela Dai, Thomas A. Funkhouser, Maciej Halber, Matthias Nießner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *3DV*, 2017.

[2] Xinjing Cheng, Peng Wang, and Ruigang Yang. Learning depth with convolutional spatial propagation network. *IEEE TPAMI*, 42, 2020.

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

[4] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. *ArXiv*, abs/2107.02791, 2021.

[5] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CVPR*, 2016.

[6] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

[7] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

[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 *ICCV*, 2021.