# NPBG++: Accelerating Neural Point-Based Graphics

Supplementary Material

## A. Datasets

We validate the effectiveness of the proposed method on four different datasets.

- **ScanNet [8]:** a large dataset that offers ample variety within the scenes. We use 78 scenes for training our pipeline and three holdout scenes to measure the generalization of our approach. We select 10 testing frames along the camera trajectory in each holdout scene. From the remaining observations, we obtain the source frames by excluding those closest to the test images to showcase our model's generalization capabilities.
- **NeRF-Synthetic [30]:** a high-quality synthetic dataset with object-centric scenes. We use five of the scenes to fine-tune our method and the three remaining scenes for testing. For the train, the validation, and the test image sets, we follow the original split.
- **H3DS [34]:** a dataset designed for the 3D reconstruction of human heads. It contains images, associated camera poses, as well as accurate foreground masks. We use eight scans for training and two for testing under the 32 view setup.
- **DTU [15]:** is a multi-view stereo dataset with relatively simple objects captured at a resolution of 1200 x 1600, with accurate camera positions. We use the subset of 15 scenes manually annotated with binary segmentation masks for IDR [61]. Out of 15, we use 12 for training and three holdout scenes with nine images for test and five for validation.

DTU scenes for pretraining: scan{37, 40, 55, 63, 65, 69, 83, 97, 105, 106, 122}. ScanNet scenes for pretraining: scene{2-3, 5-7, 9-10, 12-14, 16-21, 24-30, 32-36, 38-40, 42, 44, 46-47, 49-53, 55, 57-58, 60-63, 65, 67-68, 70-78, 80-94, 96-99}. H3DS scenes for pretraining: {1b2a8613401e42a8, 3b5a2eb92a501d54, 5cd49557ea450c89, 444ea0dc5e85ee0b, 609cc60fd416e187, 868765907f66fd85, e98bae39fad2244e, f7e930d8a9ff2091}. NeRF scenes for pretraining: {chair, drums, lego, materials, ship}

## B. Implementation details

We don't use batch normalization in any part of the system. We remove bias in the final dense layer of the feature extractor. The MLP network $H(v)$ predicts learnable basis functions and consists of one hidden layer of width 64. It uses positional encoding similarly to NeRF [30]. We reduce the width of the refiner network compared to the original one [1], reducing it to ≈0.4 mln parameters, as we see a boost in speed without a noticeable drop in quality.

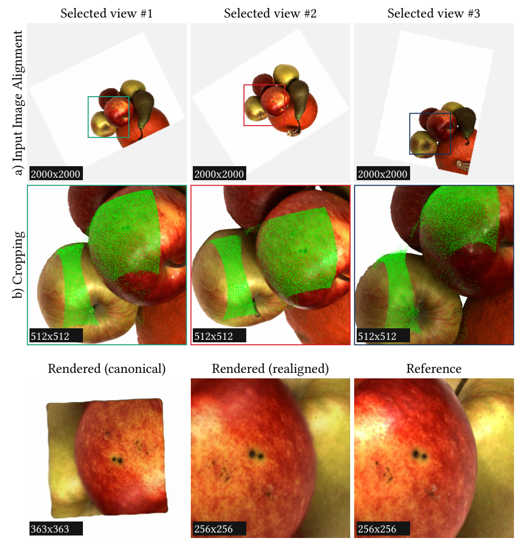


Figure 8. **View selection and cropping.** During training, we randomly crop a target (reference) patch. a) We select three relevant input views and apply input image alignment. b) We make the crops of size twice larger compared to the target patch. Crops are done so that as many points visible in the target patch (marked with *green* color in the Figure) are projected inside the cropped region. We do not resize crops.

We train the system on 4 nVidia Tesla V100 16Gb GPUs. All the experiments were performed using the PyTorch framework [32], its higher-level neural network API PyTorch Lightning [9], and Hydra framework [59] for configuring experiments. We use Adam optimizer [20] with constant learning rate equal to 0.0001. We perform training until convergence. We use the implementation for SPNASNet-100 [47] encoder for the feature extractor network from [55]. We follow the PyTorch3d [35] convention for world coordinates and working with perspective cameras. We use the Kornia library [36] for color augmentatoins and homography transforms.

The visibility reduction factor used for ScanNet data is $r{=}1$ and for all other datasets $r{=}0$ (see visibility definition in Sec. 3.1 - Estimating point's visibility).

The process of View selection and Cropping, described in Sec. 3.3, is illustrated in Figure 8.

## C. Additional experiments

**Style Transfer**. We show one more possible application of our approach: 3D style transfer. To this end, we modify the loss formulation in Sec. 3.3 by setting $\lambda_2 = 0$, $\lambda_3 = 0$, and introducing a style loss with weight equal to 1. We finetune the system for

Figure 9. **Qualitative results of 3D scene stylization.** The leftmost column shows one of the input views with the input style image on the top-left corner. The other columns represent the stylization result at different novel views.

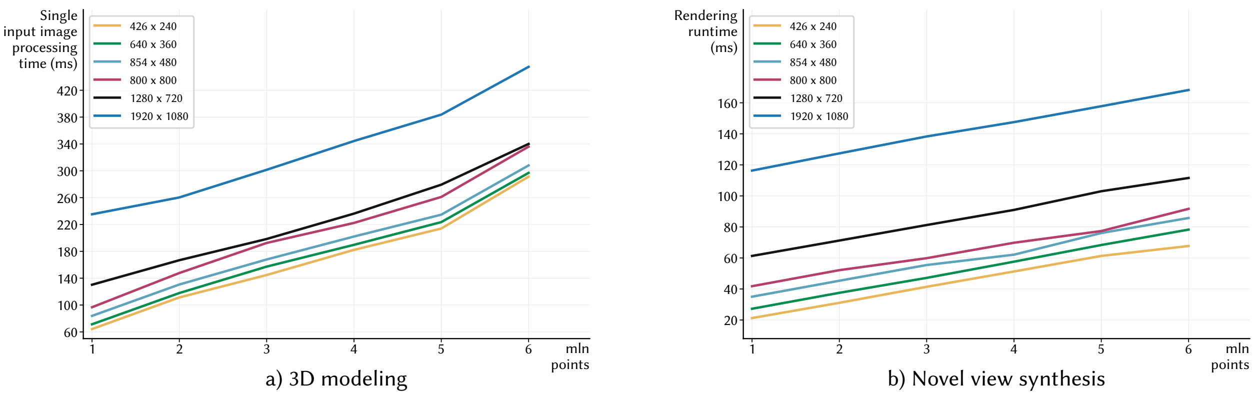


Figure 10. **Runtime evaluation. a)** Single input image processing time (denoted $U$) depends on image size and number of points and includes time for feature extraction plus the time to update intermediate states. If we let $P$ be the time required to obtain the point cloud (e.g. using MVS), and denote with $S$ the time required to compute Eq. 2, Eq. 3, then the 3d modeling stage total runtime is equal to $P +$ #images $\times\ U + S$. $S$ takes from 512 ms to 3059 ms when varying the number of points from 1 mln to 6 mln and does not depend on image size. **b)**. Rendering time depends on image size and number of points. On average, descriptors calculation step takes 0.05%, rasterization step - 37.86%, refinement step - 57.85%, and output image alignment - 4.24% of total rendering time during novel view synthesis. In principle, one can accelerate the rasterization step using an OpenGL implementation instead of a PyTorch-based one. The time was measured using Nvidia GeForce 1080 Ti. The refinement step can also potentially be accelerated by using neural architecture search and other common techniques.

five epochs on ScanNet train scenes and then infer the system on a holdout scene. See qualitative results in Figure 9.

# D. Runtime Details

The runtime of our system depends on the number of points in the point cloud and the image size. We provide the detailed analysis in Figure 10.

# E. Additional results

We include the scores obtained by all methods on specific scenes for all considered datasets. Table 5 contains a detailed version of scores reported in Table 1

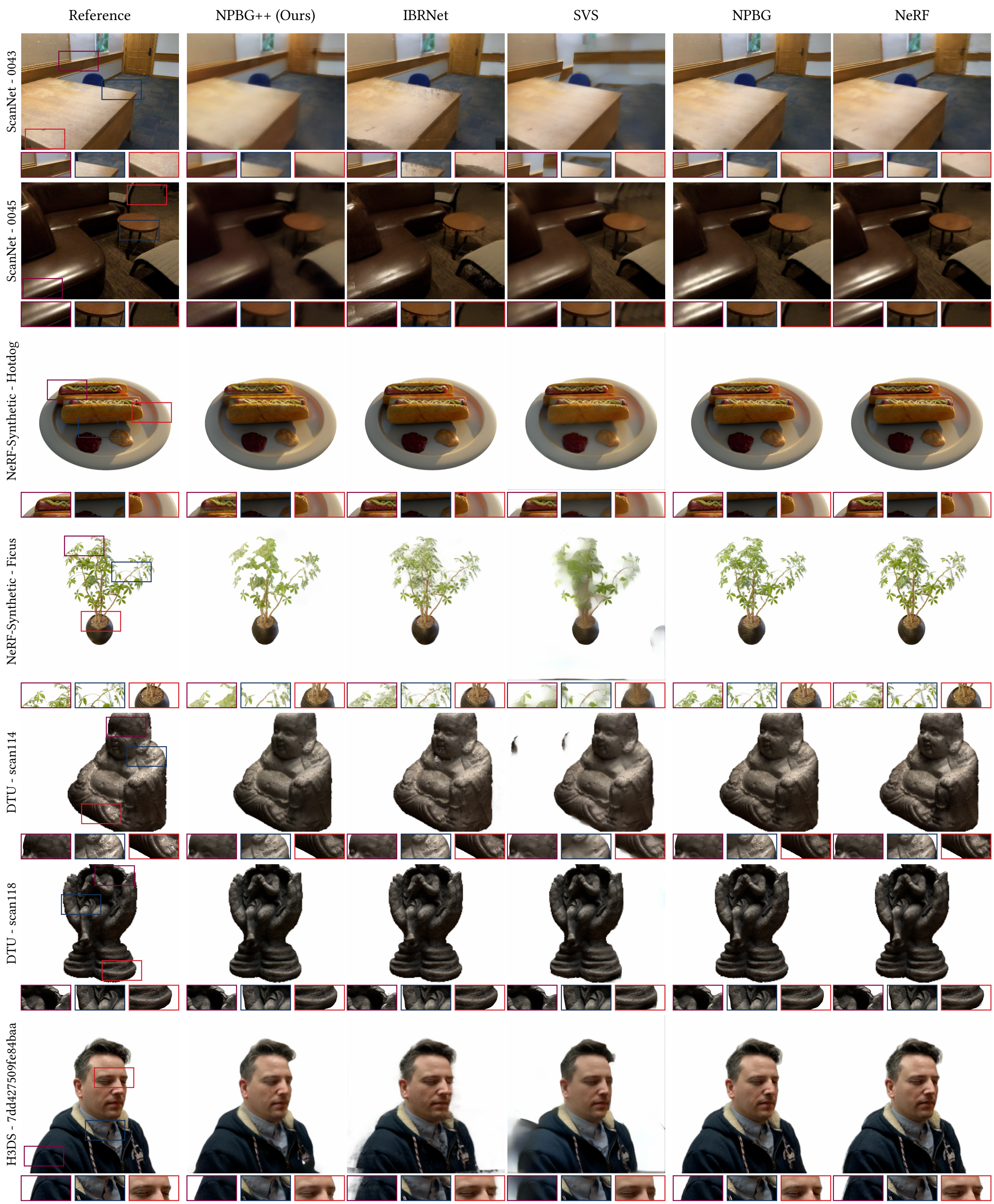


Figure 11. **Additional qualitative evaluations.** Comparisons with otimization-based approaches (NPBG [1], NeRF [30]) and learning based approaches (IBRNet [53], SVS [38]) on ScanNet [8], NeRF-Synthetic [30], DTU [15], H3DS [34] scenes.

| Method | Per scene optimization | Nerf-Synthetic - Hotdog | | | Nerf-Synthetic - Ficus | | | Nerf-Synthetic - Mic | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | **PSNR↑** | **SSIM↑** | **LPIPS↓** | **PSNR↑** | **SSIM↑** | **LPIPS↓** | **PSNR↑** | **SSIM↑** | **LPIPS↓** |
| SVS [38] | ✗ | 25.70 | 0.933 | 0.107 | 20.29 | 0.883 | 0.132 | 22.44 | 0.940 | 0.072 |
| IBRNet [53] | ✗ | 33.33 | 0.969 | 0.144 | 25.47 | 0.926 | 0.186 | 29.60 | 0.969 | 0.140 |
| **NPBG++ (Ours)** | ✗ | 28.84 | 0.949 | 0.078 | 22.48 | 0.903 | 0.090 | 26.87 | 0.957 | 0.045 |
| NPBG [1] | ✓ | 32.26 | 0.957 | 0.059 | 24.71 | 0.920 | 0.078 | 28.88 | 0.962 | 0.036 |
| NeRF [30] | ✓ | 36.08 | 0.975 | 0.045 | 29.29 | 0.958 | 0.049 | 32.10 | 0.977 | 0.030 |
| SVS$_{ft}$ [38] | ✓ | 26.56 | 0.934 | 0.105 | 20.62 | 0.879 | 0.128 | 22.93 | 0.943 | 0.071 |
| IBRNet$_{ft}$ [53] | ✓ | 36.46 | 0.980 | 0.137 | 28.66 | 0.957 | 0.146 | 32.40 | 0.980 | 0.148 |
| **NPBG++$_{ft\text{-}system}$ (Ours)** | ✓ | 27.16 | 0.948 | 0.072 | 23.48 | 0.911 | 0.082 | 28.08 | 0.962 | 0.037 |
| **NPBG++$_{ft}$ (Ours)** | ✓ | 32.31 | 0.964 | 0.050 | 24.61 | 0.925 | 0.070 | 29.08 | 0.967 | 0.029 |

| Method | Per scene optimization | ScanNet - 0000 | | | ScanNet - 0043 | | | ScanNet - 0045 | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | **PSNR↑** | **SSIM↑** | **LPIPS↓** | **PSNR↑** | **SSIM↑** | **LPIPS↓** | **PSNR↑** | **SSIM↑** | **LPIPS↓** |
| SVS [38] | ✗ | 22.17 | 0.752 | 0.442 | 21.74 | 0.833 | 0.429 | 26.06 | 0.727 | 0.465 |
| IBRNet [53] | ✗ | 19.54 | 0.703 | 0.535 | 24.42 | 0.859 | 0.434 | 26.07 | 0.719 | 0.513 |
| **NPBG++ (Ours)** | ✗ | 20.66 | 0.738 | 0.530 | 22.63 | 0.845 | 0.464 | 26.04 | 0.716 | 0.511 |
| NPBG [1] | ✓ | 22.24 | 0.695 | 0.474 | 25.27 | 0.830 | 0.421 | 27.75 | 0.686 | 0.482 |
| NeRF [30] | ✓ | 22.08 | 0.729 | 0.588 | 25.98 | 0.869 | 0.466 | 29.15 | 0.743 | 0.558 |
| SVS$_{ft}$ [38] | ✓ | 21.30 | 0.559 | 0.535 | 21.24 | 0.737 | 0.531 | 24.38 | 0.533 | 0.562 |
| IBRNet$_{ft}$ [53] | ✓ | 20.14 | 0.714 | 0.528 | 25.56 | 0.868 | 0.427 | 27.57 | 0.741 | 0.523 |
| **NPBG++$_{ft\text{-}system}$ (Ours)** | ✓ | 21.04 | 0.738 | 0.517 | 22.31 | 0.846 | 0.454 | 27.08 | 0.720 | 0.498 |
| **NPBG++$_{ft}$ (Ours)** | ✓ | 22.05 | 0.742 | 0.457 | 25.51 | 0.859 | 0.410 | 28.26 | 0.716 | 0.477 |

| Method | Per scene optimization | DTU - 110 | | | DTU - 114 | | | DTU - 118 | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | **PSNR↑** | **SSIM↑** | **LPIPS↓** | **PSNR↑** | **SSIM↑** | **LPIPS↓** | **PSNR↑** | **SSIM↑** | **LPIPS↓** |
| SVS [38] | ✗ | 19.22 | 0.872 | 0.178 | 22.04 | 0.893 | 0.165 | 21.69 | 0.927 | 0.143 |
| IBRNet [53] | ✗ | 23.77 | 0.923 | 0.238 | 27.42 | 0.910 | 0.231 | 26.24 | 0.938 | 0.225 |
| **NPBG++ (Ours)** | ✗ | 21.13 | 0.907 | 0.161 | 24.57 | 0.904 | 0.164 | 24.00 | 0.933 | 0.137 |
| NPBG [1] | ✓ | 24.65 | 0.916 | 0.123 | 26.74 | 0.891 | 0.137 | 26.62 | 0.932 | 0.114 |
| NeRF [30] | ✓ | 25.55 | 0.917 | 0.194 | 27.42 | 0.894 | 0.217 | 27.78 | 0.928 | 0.182 |
| SVS$_{ft}$ [38] | ✓ | 17.57 | 0.842 | 0.208 | 21.57 | 0.860 | 0.192 | 23.01 | 0.889 | 0.171 |
| IBRNet$_{ft}$ [53] | ✓ | 23.69 | 0.915 | 0.238 | 25.43 | 0.913 | 0.193 | 22.28 | 0.923 | 0.237 |
| **NPBG++$_{ft\text{-}system}$ (Ours)** | ✓ | 22.30 | 0.916 | 0.150 | 25.16 | 0.906 | 0.162 | 24.70 | 0.935 | 0.128 |
| **NPBG++$_{ft}$ (Ours)** | ✓ | 24.84 | 0.929 | 0.122 | 26.72 | 0.911 | 0.134 | 26.67 | 0.945 | 0.113 |

| Method | Per scene optimization | H3DS - 5ae021f2805c0854 | | | H3DS - 7dd427509fe84baa | | |
|---|---|---|---|---|---|---|---|
| | | **PSNR↑** | **SSIM↑** | **LPIPS↓** | **PSNR↑** | **SSIM↑** | **LPIPS↓** |
| SVS [38] | ✗ | 19.24 | 0.763 | 0.230 | 18.68 | 0.833 | 0.189 |
| IBRNet [53] | ✗ | 21.23 | 0.756 | 0.303 | 19.37 | 0.826 | 0.255 |
| **NPBG++ (Ours)** | ✗ | 22.26 | 0.782 | 0.202 | 21.33 | 0.854 | 0.151 |
| NPBG [1] | ✓ | 24.59 | 0.783 | 0.170 | 24.77 | 0.871 | 0.122 |
| NeRF [30] | ✓ | 23.81 | 0.797 | 0.202 | 23.95 | 0.868 | 0.153 |
| SVS$_{ft}$ [38] | ✓ | 20.95 | 0.738 | 0.206 | 19.29 | 0.801 | 0.188 |
| IBRNet$_{ft}$ [53] | ✓ | 25.13 | 0.811 | 0.224 | 24.22 | 0.889 | 0.165 |
| **NPBG++$_{ft\text{-}system}$ (Ours)** | ✓ | 24.08 | 0.795 | 0.182 | 23.49 | 0.876 | 0.127 |
| **NPBG++$_{ft}$ (Ours)** | ✓ | 25.08 | 0.805 | 0.158 | 24.73 | 0.885 | 0.116 |

Table 5. **Detailed quantitative evaluations.** For each scene, we compute the metrics [67] on holdout frames. Subscript *ft* indicates finetuned versions of the methods. In the case of NPBG++$_{ft}$ we directly finetune coefficients ($\beta$, $\beta_0$) and the refiner. In the case of NPBG++$_{ft\text{-}system}$ we finetune the feature extractor, aggregator (MLP: neural basis functions), and refiner.

## References


[1] Kara-Ali Aliev, Artem Sevastopolsky, Maria Kolos, Dmitry Ulyanov, and Victor Lempitsky. Neural point-based graphics. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII 16*, pages 696–712. Springer, 2020. 1, 2, 3, 5, 6, 7, 4

[2] Giang Bui, Truc Le, Brittany Morago, and Ye Duan. Point-based rendering enhancement via deep learning. *The Visual Computer*, 34(6):829–841, 2018. 2

[3] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. *arXiv preprint arXiv:2103.15595*, 2021. 2

[4] Shenchang Eric Chen and Lance Williams. View interpolation for image synthesis. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 279–288, 1993. 1

[5] Julian Chibane, Aayush Bansal, Verica Lazova, and Gerard Pons-Moll. Stereo radiance fields (srf): Learning view synthesis for sparse views of novel scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7911–7920, 2021. 2

[6] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014. 4

[7] Charles Csuri, Ron Hackathorn, Richard Parent, Wayne Carlson, and Marc Howard. Towards an interactive high visual complexity animation system. *Acm Siggraph Computer Graphics*, 13(2):289–299, 1979. 2

[8] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017. 6, 7, 2, 4

[9] et al. Falcon, WA. Pytorch lightning. *GitHub. Note: https://github.com/PyTorchLightning/pytorch-lightning*, 3, 2019. 2

[10] John Flynn, Ivan Neulander, James Philbin, and Noah Snavely. Deepstereo: Learning to predict new views from the world's imagery. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5515–5524, 2016. 2

[11] Stephan J Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. Fastnerf: High-fidelity neural rendering at 200fps. *arXiv preprint arXiv:2103.10380*, 2021. 2

[12] Jeffrey P Grossman and William J Dally. Point sample rendering. In *Eurographics Workshop on Rendering Techniques*, pages 181–192. Springer, 1998. 2

[13] Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. Deep blending for free-viewpoint image-based rendering. *ACM Transactions on Graphics (TOG)*, 37(6):1–15, 2018. 2

[14] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 4

[15] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engin Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 406–413, 2014. 6, 7, 2, 4

[16] Nima Khademi Kalantari, Ting-Chun Wang, and Ravi Ramamoorthi. Learning-based view synthesis for light field cameras. *ACM Transactions on Graphics (TOG)*, 35(6):1–10, 2016. 2

[17] Takeo Kanade, PJ Narayanan, and Peter W Rander. Virtualized reality: Concepts and early results. In *Proceedings IEEE Workshop on Representation of Visual Scenes (In Conjunction with ICCV'95)*, pages 69–76. IEEE, 1995. 1

[18] Abhishek Kar, Christian Häne, and Jitendra Malik. Learning a multi-view stereo machine. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 2

[19] Sagi Katz, Ayellet Tal, and Ronen Basri. Direct visibility of point sets. *ACM Trans. Graph.*, 26(3):24–es, July 2007. 4

[20] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 2

[21] Maria Kolos, Artem Sevastopolsky, and Victor Lempitsky. Transpr: Transparency ray-accumulating neural 3d scene point renderer. In *2020 International Conference on 3D Vision (3DV)*, pages 1167–1175. IEEE, 2020. 2, 3

[22] Georgios Kopanas, Julien Philip, Thomas Leimkühler, and George Drettakis. Point-based neural rendering with per-view optimization. *Computer Graphics Forum (Proceedings of the Eurographics Symposium on Rendering)*, 40(4), June 2021. 2

[23] Christoph Lassner and Michael Zollhofer. Pulsar: Efficient sphere-based neural rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1440–1449, 2021. 2, 3, 4, 5, 7

[24] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 31–42, 1996. 2

[25] Marc Levoy and Turner Whitted. *The use of points as a display primitive*. Citeseer, 1985. 2

[26] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *Advances in Neural Information Processing Systems*, 33, 2020. 2

[27] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7210–7219, 2021. 2

[28] Leonard McMillan and Gary Bishop. Head-tracked stereoscopic display using image warping. In *Stereoscopic Displays and Virtual Reality Systems II*, volume 2409, pages 21–30. International Society for Optics and Photonics, 1995. 1

[29] Moustafa Meshry, Dan B Goldman, Sameh Khamis, Hugues Hoppe, Rohit Pandey, Noah Snavely, and Ricardo Martin-Brualla. Neural rerendering in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6878–6887, 2019. 2

[30] 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 *European conference on computer vision*, pages 405–421. Springer, 2020. 2, 6, 7, 4, 5

[31] Thomas Neff, Pascal Stadlbauer, Mathias Parger, Andreas Kurz, Joerg H. Mueller, Chakravarty R. Alla Chaitanya, Anton S. Kaplanyan, and Markus Steinberger. DONeRF: Towards Real-Time Rendering of Compact Neural Radiance Fields using Depth Oracle Networks. *Computer Graphics Forum*, 40(4), 2021. 2

[32] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 2

[33] Francesco Pittaluga, Sanjeev J Koppal, Sing Bing Kang, and Sudipta N Sinha. Revealing scenes by inverting structure from motion reconstructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 145–154, 2019. 2

[34] Eduard Ramon, Gil Triginer, Janna Escur, Albert Pumarola, Jaime Garcia, Xavier Giro-i Nieto, and Francesc Moreno-Noguer. H3d-net: Few-shot high-fidelity 3d head reconstruction. *arXiv preprint arXiv:2107.12512*, 2021. 6, 7, 2, 4

[35] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv:2007.08501*, 2020. 2

[36] E. Riba, D. Mishkin, D. Ponsa, E. Rublee, and G. Bradski. Kornia: an open source differentiable computer vision library for pytorch. In *Winter Conference on Applications of Computer Vision*, 2020. 2

[37] Gernot Riegler and Vladlen Koltun. Free view synthesis. In *European Conference on Computer Vision*, pages 623–640. Springer, 2020. 2

[38] 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. 2, 6, 7, 4, 5

[39] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 3, 5

[40] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1

[41] Johannes Lutz Schönberger, True Price, Torsten Sattler, Jan-Michael Frahm, and Marc Pollefeys. A vote-and-verify strategy for fast spatial verification in image retrieval. In *Asian Conference on Computer Vision (ACCV)*, 2016. 6

[42] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016. 1

[43] Artem Sevastopolsky, Savva Ignatiev, Gonzalo Ferrer, Evgeny Burnaev, and Victor Lempitsky. Relightable 3d head portraits from a smartphone video. *arXiv preprint arXiv:2012.09963*, 2020. 5

[44] Harry Shum and Sing Bing Kang. Review of image-based rendering techniques. In *Visual Communications and Image Processing 2000*, volume 4067, pages 2–13. International Society for Optics and Photonics, 2000. 2

[45] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015. 5

[46] Peter-Pike Sloan, Jan Kautz, and John Snyder. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 527–536, 2002. 3

[47] Dimitrios Stamoulis, Ruizhou Ding, Di Wang, Dimitrios Lymberopoulos, Bodhi Priyantha, Jie Liu, and Diana Marculescu. Single-path nas: Designing hardware-efficient convnets in less than 4 hours. *arXiv preprint arXiv:1904.02877*, 2019. 2

[48] Matthew Tancik, Ben Mildenhall, Terrance Wang, Divi Schmidt, Pratul P Srinivasan, Jonathan T Barron, and Ren Ng. Learned initializations for optimizing coordinate-based neural representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2846–2855, 2021. 2

[49] Ayush Tewari, Justus Thies, Ben Mildenhall, Pratul Srinivasan, Edgar Tretschk, Yifan Wang, Christoph Lassner, Vincent Sitzmann, Ricardo Martin-Brualla, Stephen Lombardi, et al. Advances in neural rendering. *arXiv preprint arXiv:2111.05849*, 2021. 3

[50] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: Image synthesis using neural textures. *ACM Transactions on Graphics (TOG)*, 38(4):1–12, 2019. 2, 4

[51] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 4

[52] Fangjinhua Wang, Silvano Galliani, Christoph Vogel, Pablo Speciale, and Marc Pollefeys. Patchmatchnet: Learned multi-view patchmatch stereo, 2021. 6

[53] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo

Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2021. 2, 6, 7, 4, 5

[54] Maurice Weiler, Fred A Hamprecht, and Martin Storath. Learning steerable filters for rotation equivariant cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 849–858, 2018. 4

[55] Ross Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019. 2

[56] Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. Synsin: End-to-end view synthesis from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7467–7477, 2020. 3, 4, 5, 7

[57] Suttisak Wizadwongsa, Pakkapon Phongthawee, Jiraphon Yenphraphai, and Supasorn Suwajanakorn. Nex: Real-time view synthesis with neural basis expansion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8534–8543, 2021. 2, 3, 4

[58] Minye Wu, Yuehao Wang, Qiang Hu, and Jingyi Yu. Multi-view neural human rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1682–1691, 2020. 2

[59] Omry Yadan. Hydra - a framework for elegantly configuring complex applications. Github, 2019. 2

[60] 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. 5

[61] 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, 2020. 2

[62] Lin Yen-Chen. Nerf-pytorch. `https://github.com/yenchenlin/nerf-pytorch/`, 2020. 6

[63] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. PlenOctrees for real-time rendering of neural radiance fields. In *ICCV*, 2021. 2, 3

[64] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2021. 2

[65] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4471–4480, 2019. 5

[66] Igor Zacharov, Rinat Arslanov, Maksim Gunin, Daniil Stefonishin, Andrey Bykov, Sergey Pavlov, Oleg Panarin, Anton Maliutin, Sergey Rykovanov, and Maxim Fedorov. "zhores"—petaflops supercomputer for data-driven modeling, machine learning and artificial intelligence installed in skolkovo institute of science and technology. *Open Engineering*, 9(1):512–520, 2019. 1

[67] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 6, 5

[68] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1851–1858, 2017. 2

[69] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018. 2