Advanced Computer Vision Term Project

[1]*Chung-Hao Liao* (廖崇澔) [1]*Chiou-Shann Fuh* (傅楸善)

[1]Department of Computer Science and Information Engineering,
National Taiwan University, Taipei, Taiwan
[*]E-mail: b09902133@csie.ntu.edu.tw fuh@csie.ntu.edu.tw

ABSTRACT

We modify a channel pruning algorithm and apply it to a generalizable NeRF model, which deals with the problem of novel view synthesis. The model we use consists of two parts, the geometric reasoner used to reason the geometric prior for each scene and the renderer used to predict images from novel view.

Keywords: *NeRF, Multi-view Stereo, Channel Pruning*

1. INTRODUCTION

Neural Radiance Fields (NeRFs) have been demonstrated as a powerful tool in novel view synthesis due to their ability to construct complex scenes and objects with delicate detail. As a result, NeRFs may play a crucial role in many applications such as movies and Augmented/Virtual Reality (AR/VR).

Nevertheless, NeRFs require a huge amount of computational resources during training and rendering. Traditional NeRFs need to be trained from scratch on every scene, which means that the large training cost is required when a new scene is encountered. As a result, the training cost for NeRFs is very remarkable. In addition, the slow rendering process is mainly attributed to two properties of NeRFs. First, they use a volumetric rendering algorithm, which implies that hundreds of queries are needed to render a single pixel. Second, each query involves an inference on a Multi-Layer Perceptron (MLP). The remarkable computational cost has made NeRFs impractical to be widely deployed on edge devices, which have limited computational resources.

There are some works that focus on reducing training cost by using generalized NeRFs models, where a radiance field can be constructed from only several multi-view input images without the need of training from scratch. For example, MVSNeRF [1] and GeoNeRF [4] leverage 2D-CNN and 3D-CNN to construct a *cost volume* and a *neural scene encoding volume* for each novel scene encountered, and an MLP is used to regress volume density and RGB radiance using the features from the neural scene encoding volume. However, despite its powerful generalization ability, the computational cost of its rendering process is more than that of original NeRF model due to the additional step to build the neural encoding volume.

In this work, we modify a current channel pruning algorithm to downsize the original convolutional neural networks for building cost volume and neural scene encoding. Taking advantage of sparser sub-model, we reduce the memory consumption and computational resource needed for predicting novel view with generalized NeRF models.

2. RELATED WORK

Our work is inspired by the following works related to *novel view synthesis*, *multi-view stereo* and *channel pruning*.

2.1 Multi-view Stereo

Multi-view stereo (MVS) is a computer vision problem, whose purpose is to estimate the dense geometric representation of a scene given multiple images captured from different viewpoints. Recently, methods based on deep learning have been adopted to address this problem, and they have been proved to outperform traditional methods. For instance, MVSNet [5] infers the depth map from several input views by aggregating the features from all images and constructing a cost volume, which undergoes 3D convolutions and regresses the depth map. Cascade MVSNet [6] proposes a more time and memory efficient formulation of cost volume by using feature pyramid networks to extract features from all images. Moreover, they narrow the depth range of cost volume by using the prediction from the courser-level cost volume. It has also been shown that replacing the variance-based construction with group-wise correlation similarity can further make the cost volumes smaller in size [8]. MVSTER [7] utilizes the Epipolar Transformer to learn both 2D features and 3D features more efficiently in comparison to [5] and [6]. Besides, the proposed method enhances the use of 3D spatial associations, which makes them achieve even better reconstruction performance. We found that MVS architecture is a appropriate and appliable way to learn

and predict the geometric representation of an unknown scene.

2.2 Novel View Synthesis

Novel View Synthesis (NVS) aims at synthesizing novel view images from several input images associated with their camera poses taken on the same scene. Earlier work [9, 10] tried to address this problem by directly interpolating input images. For example, [9] utilizes a lumigraph approach to generalize to arbitrary camera poses of input images. Researchers [10] studied the problem with light field functions.

2.3 Neural Scene Representations

Recent works done on novel view synthesis problem have focused on deep learning-based methods, namely, using neural networks to represent the geometry and appearance of scenes. To render a novel view image, we can query the corresponding color and opacity of some points through the neural scene representations. In particular, NeRF [13] combines MLPs and differentiable volume rendering algorithm to achieve lifelike synthesis. Some improvements [14, 15, 16] have been done on the original NeRF model, but in most of them, the neural networks need to be trained from scratch for each new scene encountered, which may take hours or days.

MVSNeRF [1] constructs a plane-swept cost volume inspired by MVSNet [5] and utilizes 3D UNet to generate a scene encoding volume, which contains meaningful information about the appearance of scenes. The scene encoding volume contributes to its generalization ability. Namely, rendering of a new scene with few input views without retraining is possible. Both ENeRF [17] and GeoNeRF [4] introduce cascaded cost volumes to obtain the geometric prior of scene with multiple levels of resolution. ENeRF [17] further accelerates the rendering by introducing depth-guided sampling which is learned from cost volumes. On the other hand, GeoNeRF [4] improves the quality of synthesized images by introducing the attention-based renderer which aggregates information from different input views. In addition, they deal with the problem of occlusion by detecting and excluding the occluded views, which takes advantage of depth map prediction inferred from fine-level cost volumes.

2.4 Channel Pruning

To address the over-parametrization problem of convolutional neural networks (CNN), Many approaches have been proposed to compress the model size of CNNs. Generally, there are two kinds of network pruning approaches. Weight pruning (unstructured pruning) eliminates specific weights in filters, but the acceleration only works on some specialized hardware due to the resulting unstructured sparsity. In Channel pruning (filter pruning/ structured pruning), the entire filters are eliminated. As a result, its effect is more flexible and applicable to a wide variety of hardware. Current channel pruning algorithms can be roughly categorized into three categories:

2.4.1. Pruning after training

There are three main steps in these approaches: pre-training a big model, pruning unimportant channels and fine-tuning the pruned sub-model. For example, [18] prunes the filters based on their ℓ_1-norm. [19] introduces a variant of ℓ_2 -norm as the criteria of selecting unimportant channels. [20] claims that filters with smaller norm are not always less important. Hence, they propose a novel criterion based on its distance to the geometric median of all the filters in the same layer. [21] proposes a pruning approach based on structural redundancy reduction, where a graph is established for each convolutional layer to measure the redundancy.

2.4.2 Pruning during training

These approaches execute channel pruning and model training simultaneously. [22] proposes a pruning framework with dynamically updated regularization terms. We mainly modify the idea of [23] to prune the MVSNet. In [23], traditional pruning-only strategy is abandoned. Instead, they repeatedly conduct pruning and regrowing stages in the model training process, which avoid pruning important channels prematurely. In addition, the numbers of remaining channels across different layers are dynamically re-distributed. The redistribution allows the pruned sub-model to explore its structure during training process, which makes the channel pruning more flexible.

2.4.3 Pruning at early stage

These approaches prune the networks at the early stage of training. For example, [24] identified the structured of pruned models in the very stage of training. While this approach makes the model training much more efficiently, the performance also drops more significantly when the original model size is huge.

3. METHOD

The following sections provide the details of our method.

3.1 The Architecture of Generalized NeRF Model

The architecture of our original model without pruning is shown in figure 1. Similar to GeoNeRF [4], the entire architecture can be partitioned into two parts: geometric reasoner and renderer. Given a set of V input views $\{I_v\}_{v=1}^{V}$ with size $H \times W$, the geometric reasoner builds cascaded cost volume for each input view individually, and the renderer interpolates features

obtained from the geometric reasoner to yield the color and density for each sample point. Note that all the sample points are on the camera rays at novel camera poses. Finally, the images at novel camera poses can be rendered by the volume rendering approach. The geometric reasoner is a process of per-scene initialization. Namely, it is only inferred when a new scene is encountered. On the other hand, the renderer is required whenever we want to render a novel view, no matter whether the scene has changed.

3.1.1 Geometric reasoner

First, each input images undergoes a Feature Pyramid Network (FPN) to generate 2D feature maps at three different scales.

$$F_{v,l} = FPN(I_v) \in \mathbb{R}^{\frac{H}{2^l}\times\frac{W}{2^l}\times 2^l C} \quad l = 0,1,2$$

where FPN denotes a feature pyramid network. Then, we follow the same approach in Cascaded MVSNet, where a homography warping $H_v(d)$ warps a pixel (u, v) in the *v*-th view to the reference view at depth d. The construction of hypothesis depth planes also follows the original Cascaded MVSNet. The coarsest level consists of depth planes covering the whole range in the camera's frustum. Then, finer levels narrow the range of hypothesis depth planes based on the depth map prediction from the coarser level. Different from the original Cascaded NVSNet, groupwise correlation similarity is adopted to build cost volumes for each input view. Let $P_{v,l}$ denote the cost volume for *v*-th view with scale level *l*. We further process these cost volumes to obtain the 3D semantic features and depth maps for each input view.

$$D_{v,l}, \Phi_{v,l} = R_l(P_{v,l})$$

where R_l denotes the UNet built by 3D convolution for scale level l, $D_{v,l} \in \mathbb{R}^{\frac{H}{2^l}\times\frac{W}{2^l}\times 1}$ represents depth maps and $\Phi_{v,l} \in \mathbb{R}^{D_l\times\frac{H}{2^l}\times\frac{W}{2^l}\times C}$ represents 3D semantic features maps.

3.1.2 Renderer

For all sample points $\{x_n\}_{n=1}^N$, we combine finest-scale 2D features and the 3D features from all levels to predict its density $\{\sigma_n\}_{n=1}^N$ and color $\{c_n\}_{n=1}^N$. The details are the same as GeoNeRF. Once volume densities and colors are predicted for all sample points, the color of the corresponding pixel is generated via volume rendering algorithm.

3.2 The Pipeline of Pruning MVSNet

We follow and modify the algorithm proposed in [23] to compress *FPN* and the 3D UNet $\{R_l\}_{l=0}^2$ in the geometric reasoner. During training process, the sub-model structure exploration, channel pruning stage and the channel regrowing stage is periodically conducted at every ΔT steps.

3.2.1 Sub-model structure exploration

In this stage, sparsity is re-distributed across the different layers, while the overall sparsity is fixed. We use the learnable scaling factors in Batch Normalization (BN) as the criteria to judge which layers are more important.

For FPN, the algorithm of structure exploration is the same as research [23]. Assume that there are *L* layers in a CNN network and the original number of channels in layer *l* is C_l. We denote the BN scaling factors of channel *c* in layer *l* by $\gamma_{l,c}$ and the overall sparsity by *S*. We rank all of the scaling factors. Let Γ be all of the pairs (l, c) such that $\gamma_{l,c}$ is in the top *1-S* percent of all the BN scaling factors. Then, the sparsity of layer *l*:

$$s_l = \frac{|\{c|c \in \{1,2,\dots,C_l\}\ and\ (l,c) \notin \Gamma\}|}{C_l}$$

Hence, the number of surviving channels in layer *l* is $\tilde{C}_l = \lceil (1 - s_l)C_l \rceil$.

For 3D UNet, the output from two different layers may be an input for the same layer. We call these two layers a "pair". Hence, we need to slightly modify the criterion proposed in [23] and make sure that the channel sparsity is the same within each pair. Assume that there are L_1 pairs and L_2 single layers (layers not in a pair). We denote the BN scaling factors of channel *c* in layer *l* by $\gamma_{l,c}$ and the overall sparsity by *S*. Define $\gamma'_{l,c}$ as the followings:

$$\gamma'_{l,c} = \begin{cases} \gamma_{l,c}, & if\ layer\ l\ is\ a\ single\ layer \\ (\gamma_{l,c} + \gamma_{\tilde{l},c})/2, & if\ layer\ l\ is\ in\ a\ pair\ with\ \tilde{l} \end{cases}$$

We rank all of the $\gamma'_{l,c}$ and use the same method as that in FPN to decide the sparsity of each layer.

3.2.2 Channel pruning stage

For FPN, the algorithm of channel pruning is the same as research [23]. Given the target channel sparsity of the *l*-th layer s_l and $\tilde{C}_l = \lceil (1 - s_l)C_l \rceil$, we determine a set of surviving channels. As in [23], we represent the problem as a *Column Subset Selection* (CSS) problem. Namely, we represent the weights of all channels in layer *l* by a matrix, each column of which represents the kernel weights of an output channel. We calculate the top $\tilde{C}_l$ right singular vectors of the matrix and compute the leverage score for each column of the matrix. Then, the channels with $\tilde{C}_l$ highest leverage score are retained.

For 3D UNet, we prune the same set of channels for the layers in each pair. Hence, we also need to modify the algorithm mentioned in the last paragraph. Given the target channel sparsity of the *l*-th layer s_l and $\tilde{C}_l = \lceil (1 - s_l)C_l \rceil$. If *l-th* layer is a single layer, then we can determine the set of surviving channels by the same criterion as that for FPN. If *l-th* layer is in a pair with $\tilde{l}$, we represent the weights of all channels in the pair by a matrix. The *c-th* column of the matrix is the concatenation of the kernel weights of *c-th* output

channels from layer l and that from layer $\tilde{l}$. We calculate the top $\tilde{C}_l$ right singular vectors of the matrix and compute the leverage score for each volume of the matrix. Then, the channels with $\tilde{C}_l$ highest leverage scores are retained.

3.2.3 Channel regrowing stage

Because the model is trained from scratch, the weights in early training steps may not accurately represent the importance of each channel. That is, the channel pruning stage in early training steps may be misguided by immature kernel weights. Hence, we adopt the method proposed in [23] to regrow a subset of channels that are pruned previously.

Once a channel is regrown, the weights related to that channel is restored to the last values before it is pruned. This method further increases the influence of the regrown channels on the training behavior.

We sample the subset of regrown channels based on their orthogonality to other surviving channels. For layers in pair, the orthogonality is calculated based on the concatenation of the weights from two layers. A higher channel orthogonality means the weights of the channel are more independent to those of the surviving channels. Hence, the channel with higher orthogonality has more probability to be sampled because it is difficult to be replaced by some combination of other channels. In addition, we also supervised the training of geometric reasoner with the smooth L1 loss between the depth maps predicted by the geometric reasoner and the ground truth depth maps. For data without ground truth depth, we self -supervise with this loss.

3.3 Loss Function

As in GeoNeRF, we use three loss to train the generalized model. The first one is the mean squared error between rendered pixel color and the ground truth pixel color. For training data with ground truth depth, we also use smooth L1 loss between the predicted depth and the ground truth depth of each pixel.

4. EXPERIMENTS

4.1 Training Datasets

We train our model on the real DTU dataset [25] and real forward-facing datasets from LLFF [26] and IBRNet [27], which is the same as GeoNeRF [4]. The scenes in the DTU dataset we use for training is also the same as GeoNeRF [4]. Ground truth depths of DTU are used for depth supervision. For samples from other datasets, we utilize self-supervision form of depth supervision

4.2 Evaluation Datasets

We evaluate our model on the same scenes as GeoNeRF [4] does. The evaluation scenes include 16 test scenes from DTU MVS [25], 8 test scenes from LLFF [26] and 8 test scenes from NeRF realistic synthetic dataset [13]. We also follow the same evaluation protocol in GeoNeRF [4] for all of the datasets.

4.3 Implementation Details and Hyperparameters

We use the model checkpoint provided in GeoNeRF [4] as the pretrained model. We train our model for 100k iterations in total. For each iteration, one scene is randomly sampled, and a training batch contains 512 rays selected in the scene. For fair comparison, we use 6 input views for training the generalizable model and 9 input views for evaluation, which is the same setting as GeoNeRF [4]. The total number of sample points on each ray is 128 for all scenes, which is also the same as GeoNeRF [4].

The period of channel pruning and regrowing step is set at 2,500 iterations. We try three different channel sparsity s: 0.2, 0.3 and 0.4. The initial ratio of regrown channels is equal to channel sparsity s, and it decays over training steps following the following formula:

$$\kappa_t = s \times \frac{1 + \cos\frac{\pi t}{100000}}{2}$$

where t represents the number of training steps and κ_t represents the ratio of regrown channels.

We utilize Adam optimizer with a learning rate 5×10^{-4} and a cosine learning rate scheduler.

4.3 Experimental Results

We evaluate our model and provide a comparison with the original GeoNeRF model [4]. We use generalized model to inference on each test scene and so not optimize on each scene. The result is provided in Table 1 and Table 2 in terms of PSNR, SSIM and LPIPS. The result shows that while the total number of channels decreases by 0.4, the qualitive of rendered images still does not worsen a lot on Real Forward-Facing dataset.

Table 1. Evaluation of the pruned model on Real Forward-Facing dataset [26]

	PSNR ↑	SSIM ↑	LPIPS ↓
GeoNeRF [4]	25.44	0.839	0.180
GeoNeRF + sparsity 0.2	23.57	0.782	0.241
GeoNeRF + sparsity 0.3	23.48	0.776	0.248
GeoNeRF + sparsity 0.4	23.28	0.766	0.263

Table 2 Evaluation of the pruned model on Realistic Synthetic dataset [26]

	PSNR ↑	SSIM ↑	LPIPS ↓
GeoNeRF [4]	28.33	0.938	0.087
GeoNeRF + sparsity 0.2			
GeoNeRF + sparsity 0.3	22.62	0.850	0.242
GeoNeRF + sparsity 0.4	23.67	0.867	0.232

REFERENCES

[1] Chen, A., Xu, Z., Zhao, F., Zhang, X., Xiang, F., Yu, J., & Su, H., "Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo," *Proceedings of IEEE/CVF International Conference on Computer Vision*, Virtual, pp. 14124-14133, 2021.

[2] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engil Tola, & Henrik Aanas., "Large scale multi-view stereopsis evaluation," *Proceedings of CVPR*, pp. 406 – 413, 2014.

[3] He, Y., Ding, Y., Liu, P., Zhu, L., Zhang, H., & Yang, "Learning filter pruning criteria for deep convolutional neural networks acceleration," *the IEEE/CVF conference on computer vision and pattern recognition (pp. 2009-2018)*, 2020.

[4] Johari, Mohammad Mahdi, Yann Lepoittevin, and François Fleuret, "Geonerf: Generalizing nerf with geometry priors." *the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. (2022)

[5] 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)*, pp. 767-783. 2018.

[6] Gu, Xiaodong, Zhiwen Fan, Siyu Zhu, Zuozhuo Dai, Feitong Tan, and Ping Tan, "Cascade cost volume for high-resolution multi-view stereo and stereo matching." In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2495-2504. 2020.

[7] Wang, Xiaofeng, Zheng Zhu, Guan Huang, Fangbo Qin, Yun Ye, Yijia He, Xu Chi, and Xingang Wang, "MVSTER: epipolar transformer for efficient multi-view stereo." In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXI*, pp. 573-591. Cham: Springer Nature Switzerland, 2022.

[8] Xu, Qingshan, and Wenbing Tao. "Learning inverse depth regression for multi-view stereo with correlation cost volume." In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, pp. 12508-12515. 2020.

[9] Buehler, Chris, Michael Bosse, Leonard McMillan, Steven Gortler, and Michael Cohen. "Unstructured lumigraph rendering." In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pp. 425-432. 2001.

[10] Levin, Anat, and Fredo Durand, "Linear view synthesis using a dimensionality gap light field prior." In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1831-1838. IEEE, 2010.

[11] Davis, Abe, Marc Levoy, and Fredo Durand. "Unstructured light fields." In *Computer Graphics Forum*, vol. 31, no. 2pt1, pp. 305-314. Oxford, UK: Blackwell Publishing Ltd, 2012.

[12] Xu, Qiangeng, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. "Point-nerf: Point-based neural radiance fields." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5438-5448. 2022.

[13] Mildenhall, Ben, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. "Nerf: Representing scenes as neural radiance fields for view synthesis." *Communications of the ACM* 65, no. 1 (2021): 99-106.

[14] Barron, Jonathan T., 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*, pp. 5855-5864. 2021.

[15] Meng, Quan, Anpei Chen, Haimin Luo, Minye Wu, Hao Su, Lan Xu, Xuming He, and Jingyi Yu. "Gnerf: Gan-based neural radiance field without posed camera." In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6351-6361. 2021.

[16] Park, Keunhong, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B. Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. "Nerfies: Deformable neural radiance fields." In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5865-5874. 2021.

[17] Lin, Haotong, Sida Peng, Zhen Xu, Yunzhi Yan, Qing Shuai, Hujun Bao, and Xiaowei Zhou. "Efficient Neural Radiance Fields for Interactive Free-viewpoint Video." In *SIGGRAPH Asia 2022 Conference Papers*, pp. 1-9. 2022.

[18] Li, Hao, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. "Pruning filters for efficient convnets." *arXiv preprint arXiv:1608.08710* (2016).

[19] Zhuang, Zhuangwei, Mingkui Tan, Bohan Zhuang, Jing Liu, Yong Guo, Qingyao Wu, Junzhou Huang, and Jinhui Zhu. "Discrimination-aware channel pruning for deep neural networks." *Advances in neural information processing systems* 31 (2018).

[20] He, Yang, Ping Liu, Ziwei Wang, Zhilan Hu, and Yi Yang. "Filter pruning via geometric median for deep convolutional neural networks acceleration." In *Proceedings of the*

IEEE/CVF conference on computer vision and pattern recognition, pp. 4340-4349. 2019.

[21] Wang, Zi, Chengcheng Li, and Xiangyang Wang. "Convolutional neural network pruning with structural redundancy reduction." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14913-14922. 2021.

[22] Zhang, Tianyun, Xiaolong Ma, Zheng Zhan, Shanglin Zhou, Caiwen Ding, Makan Fardad, and Yanzhi Wang. "A unified dnn weight pruning framework using reweighted optimization methods." In *2021 58th ACM/IEEE Design Automation Conference (DAC)*, pp. 493-498. IEEE, 2021.

[23] Hou, Zejiang, Minghai Qin, Fei Sun, Xiaolong Ma, Kun Yuan, Yi Xu, Yen-Kuang Chen, Rong Jin, Yuan Xie, and Sun-Yuan Kung. "Chex: channel exploration for CNN model compression." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12287-12298. 2022.

[24] You, Haoran, Chaojian Li, Pengfei Xu, Yonggan Fu, Yue Wang, Xiaohan Chen, Richard G. Baraniuk, Zhangyang Wang, and Yingyan Lin. "Drawing early-bird tickets: Towards more efficient training of deep networks." *arXiv preprint arXiv:1909.11957* (2019).

[25] Jensen, Rasmus, 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*, pp. 406-413. (2014).

[26] Mildenhall, B., Srinivasan, P. P., Ortiz-Cayon, R., Kalantari, N. K., Ramamoorthi, R., Ng, R., & Kar, A. " Local light field fusion: Practical view synthesis with prescriptive sampling guidelines." *ACM Transactions on Graphics (TOG)*, *38*(4), 1-14. (2019)

[27] Wang, Q., Wang, Z., Genova, K., Srinivasan, P. P., Zhou, H., Barron, J. T., ... & Funkhouser, T. "Ibrnet: Learning multi-view image-based rendering." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4690-4699. (2021)