BoostMVSNeRFs: Boosting MVS-based NeRFs to Generalizable View Synthesis in Large-scale Scenes

Chih-Hai Su⋆[1], Chih-Yao Hu⋆[2], Shr-Ruei Tsai⋆[1], Jie-Ying Lee⋆[1], Chin-Yang Lin[1], and Yu-Lun Liu[1]

[1] National Yang Ming Chiao Tung University, Hsinchu City 300093, Taiwan (R.O.C.)
[2] National Taiwan University, Taipei 106319, Taiwan (R.O.C.)
https://su-terry.github.io/BoostMVSNeRFs/

Abstract. While Neural Radiance Fields (NeRFs) have demonstrated exceptional quality, their protracted training duration remains a limitation. Generalizable and MVS-based NeRFs, although capable of mitigating training time, often incur tradeoffs in quality. This paper presents a novel approach called **BoostMVSNeRFs** to enhance the rendering quality of MVS-based NeRFs in large-scale scenes. We first identify limitations in MVS-based NeRF methods, such as restricted viewport coverage and artifacts due to limited input views. Then, we address these limitations by proposing a new method that selects and combines multiple cost volumes during volume rendering. Our method does not require training and can adapt to any MVS-based NeRF methods in a feed-forward fashion to improve rendering quality. Furthermore, our approach is also end-to-end trainable, allowing fine-tuning on specific scenes. We demonstrate the effectiveness of our method through experiments on large-scale datasets, showing significant rendering quality improvements in large-scale scenes and unbounded outdoor scenarios.

Keywords: Novel view synthesis · Neural radiance fields · 3D synthesis · Neural rendering

1 Introduction

In computer vision, 3D reconstruction and novel view synthesis are crucial, with widespread applications from photogrammetry to AR/VR. Traditional methods relied on photo-geometry for 3D scene reconstruction using meshes. Recently, the task of novel view synthesis has advanced drastically since the emergence of the Neural Radiance Field (NeRF) and its variants [2–4,12,39,41,60]. NeRF encodes 3D information into a Multi-layer Perceptron (MLP) network to represent a scene. Despite such methods providing photorealistic rendering quality, these models have a huge downside as they require per-scene training with a long training time.

⋆ Authors contributed equally to the paper.

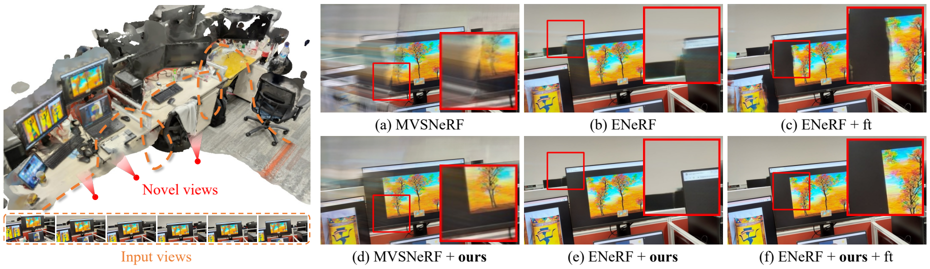
Recent advances in Generalizable NeRFs [6, 9, 70, 78, 84, 86] improve scene adaptation by extracting input image features via 2D CNNs and utilizing large datasets for training, allowing for rapid scene adaptation and enhanced rendering through fine-tuning. MVS-based methods such as MVSNeRF [9] and ENeRF [32] synthesize high-quality novel views by constructing cost volumes from a few input images, leveraging 3D CNNs and volume rendering in a feed-forward fashion. However, they are constrained by using a fixed number of input views and often struggle to reconstruct large-scale and unbounded scenes, resulting in padding artifacts at image boundaries (Fig. 1(a)) and wrongly reconstructed geometry in disocclusion regions (Fig. 1(b)). Furthermore, these issues could hardly be resolved by per-scene fine-tuning (Fig. 1(c)).

To address the problems, we propose BoostMVSNeRFs, a pipeline that is compatible with any MVS-based NeRFs to improve their rendering quality in large-scale and unbounded scenes. We first present 3D visibility scores for each sampled 3D point to indicate the proportion of contributions from individual input views. We then volume render the 3D visibility scores into 2D visibility masks to determine the contribution of each cost volume to the target novel view. Next, we combine multiple cost volumes during volume rendering to effectively expand the coverage of the novel view viewport and reduce artifacts by constructing more consistent geometry and thus alleviate the aforementioned MVS-based NeRFs' issues. Additionally, to optimize the novel view visibility coverage, we further propose a greedy algorithm to approximate the optimal support cost volume set selection for the multiple-cost volume combined rendering. Our proposed pipeline is compatible with any MVS-based NeRFs to improve their rendering quality (Fig. 1(d, e)) and is end-to-end trainable. Therefore, our method also inherits this property from MVS-based NeRFs and can be fine-tuned to a specific scene to further improve the rendering quality (Fig. 1(f)).

We conduct experiments on two large-scale datasets, Free [69] and ScanNet [14] datasets, which contain unbounded scenes with free camera trajectories and large-scale indoor scenes with complex structures, respectively. Experiments demonstrate that our proposed method performs favorably against other per-scene training or generalizable NeRFs in different dataset scenarios. Most importantly, our method is able to improve any MVS-based NeRF rendering quality through our extensive experiments, especially in free camera trajectories and unbounded outdoor scenes, which are the most common use cases in real-world applications.

2 Related Work

Novel View Synthesis. Novel view synthesis is a core challenge in computer vision, addressed through various techniques like image-based rendering [7, 19, 27, 48, 51] or multiplane image (MPI) [18, 30, 40, 58, 63, 93], and explicit 3D representations, including meshs [15, 61, 66, 74], voxels [37, 38, 56], point clouds [1, 78], depth maps [17, 21, 23, 55, 64]. Recently, neural representations [25, 34, 37, 55, 56, 73, 93], particularly Neural Radiance Fields (NeRF) [2–4, 39, 41, 47, 60, 89], have achieved photorealistic rendering by representing scenes with continuous



(a) MVSNeRF (b) ENeRF (c) ENeRF + ft

(d) MVSNeRF + **ours** (e) ENeRF + **ours** (f) ENeRF + **ours** + ft

Fig. 1: Our BoostMVSNeRFs enhances the novel view synthesis quality of MVS-based NeRFs in large-scale scenes. MVS-based NeRF methods often suffer from (a) limited viewport coverage from novel views or (b) artifacts due to limited input views for constructing cost volumes. (c) These drawbacks cannot be resolved even by per-scene fine-tuning. Our approach selects those cost volumes that contribute the most to the novel view and combines multiple selected cost volumes with volume rendering. (d, e) Our method does not require any training and is compatible with existing MVS-based NeRFs in a feed-forward fashion to improve the rendering quality. (f) The scene can be further fine-tuned as our method supports end-to-end fine-tuning.

fields. Despite the advancements in areas like relighting [5, 43, 85, 86, 88, 92], dynamic scenes [29, 35, 47, 50, 77], and multi-view reconstruction [46, 68, 82, 83], these methods although speed up training using hash grid [42] or voxel [8, 59] as representations, still require intensive per-scene optimization, thus limiting their generalizability. In contrast, our generalizable approach balances rendering quality and speed through feed-forward inference efficiently.

Multi-View Stereo and Generalizable Radiance Fields. Neural Radiance Fields (NeRF) offer photorealistic rendering but are limited by costly per-scene optimization. Recently, generalizable NeRFs [6, 9, 70, 78, 84, 86] provide efficient approaches to synthesize novel views without per-scene optimization. Techniques like PixelNeRF [86] and IBRNet [70] merge features from adjacent views for volume rendering, while PointNeRF [78] constructs point-based fields for this purpose. Multi-view stereo (MVS) methods estimate depth using cost volumes [46], with MVSNet [80] utilizing 3D CNNs for feature extraction and cost volume construction, enabling end-to-end training and further novel view synthesis. Despite amazing results from learning-based MVS, these methods are memory-intensive, prompting innovations like plane sweep [81] and coarse-to-fine strategies [10, 22, 87] for efficiency. Other works, such as MVSNeRF [9], ENeRF [32] and Im4D [31], further bridge MVS methods with NeRF, introducing volumetric representations and depth-guided sampling for speed and dynamic reconstruction. Although these works advance the performance of generalizable NeRF, their rendering qualities are hindered by the limited visibility coverage of a single cost volume, leading to poor synthesis quality and visible padding artifacts near the image boundaries on large-scale or unbounded scenes. Additional research endeavors have been suggested to address these challenges. For instance, GeoNeRF [26]) introduces a novel approach to handle occlusions, while Neural Rays [36] presents an occlusion-aware representation aimed at mitigating this problem. Although these methods tackle occlusions issues, the view coverage problem originated from MVS-based methods still exists. Our method overcomes this issue by selecting and combining multiple cost volumes to improve coverage and rendering confidence, enhancing the performance and robustness of MVS-based NeRF methods without any cost compared with previous methods.

Few-Shot NeRFs. Prior work utilized mainly two different approaches to reconstruct scenes with sparse input views [28]: introducing regularization priors and training generalized model. Regularization-based methods [16,24,45,52,53,57,65,67,75,76,79,94] such as Vip-NeRF [57] attempt to tackle this problem by obtaining visibility prior to regularize the scenes' relative depth. Training generalized models [9,11,13,26,33,54,62,70,86] on large datasets such as MVSNeRF [9] constructs cost volume to gain cross-view insight to tackle this goal. Different from this line of work, we present a novel visibility mask in a 3D fashion and serve as a visibility score to blend features while performing volume rendering.

Radiance Fields Fusion. Recently, several works propose to tackle scene fusion and intend to achieve large-scale reconstruction. NeRFusion [91] performs sequential data fusion on voxels with GRU on the image level. SurfelNeRF [20] fuses scenes after converting them to surfels [49] representation. Our approach seamlessly integrates cost volume without requiring training, thereby harnessing the capabilities of all MVS-based pre-trained models. Instead of concentrating solely on large-scale fusion, our method functions as a readily applicable tool to enhance various cost volume-based MVS applications.

3 Method

Given multi-view images in an unbounded scene, the same as other MVS-based NeRF methods (Sec. 3.1), our task is to synthesize novel view images without per-scene training. In order to tackle limited viewport coverage from a single cost volume created by a fixed number of few (*e.g.*, 3) input images, we propose *BoostMVSNeRFs*, an algorithm to consider multiple cost volumes while rendering. We first introduce a 3D visibility score for each sampled 3D point, which is used to render volume into 2D visibility masks (Sec. 3.2). Given a rendered 2D visibility mask for each cost volume, we combine multiple cost volumes in a support set to render novel views (Sec. 3.3). Finally, we present a greedy algorithm to iteratively select cost volumes and update the support set to maximize the viewport coverage and confidence of novel views (Sec. 3.4). Our pipeline is end-to-end trainable and thus can be fine-tuned on a new scene (Sec. 3.5). Our method is model-agnostic and applicable to any MVS-based NeRFs to boost the rendering quality.

3.1 MVS-based NeRFs Preliminaries

Given multi-view images with camera parameters, MVS-based NeRFs [9, 22, 32] use a shared 2D CNN to extract features for input images. Then, following MVSNet [80], we construct a feature volume by warping the input features into the target view. The warped features would be used to construct the encoding volume by computing the variance of multi-view features. Next, we apply a 3D CNN to regularize the encoding volume to build the cost volume CV to smooth the noise in the feature volume. Given a novel viewpoint, we query the color c and density σ using an MLP with sampled 3D point coordinates x, viewing directions v, trilinear interpolated cost volume values at location p, and projected colors from input views $\mathbf{C}_{\text{in}}$ as input:

$$(c, \sigma) = \text{MLP}_\theta(p, v, \text{CV}(p), \mathbf{C}_{\text{in}}), \tag{1}$$

where θ denotes the parameter of the MLP. Finally, we can volume render along rays to get the pixel colors in novel views.

The volume rendering equation in NeRF or MVSNeRF is evaluated by differentiable ray marching for novel view synthesis. A pixel color is computed by accumulating sample point values through ray marching. Here we consider a given ray $\mathbf{r}$ from the camera center o through a given pixel on the image plane as $\mathbf{r} = o + u_j d$, where d is the normalized viewing direction, and u_j is the quadrature point constrained within the bounds of the near plane u_n and the far plane u_f. The final color is given by:

$$C(\mathbf{r}) = \sum_{j=1}^{J} T(j)\alpha(\sigma_j \delta_j) c_j, \tag{2}$$

where $T(j) = \exp(-\sum_{s=1}^{j-1} \sigma_s \delta_s)$ is the accumulated transmittance, $\alpha(x) = 1 - \exp(-x)$ is the opacity of the point, and $\delta_j = u_{j+1} - u_j$ is the distance between two quadrature points.

The existing MVS-based NeRFs only utilize a single cost volume from a few viewpoints (*e.g.* 3 input views). As a result, these methods often fall into limited viewport coverage, wrong geometry, and rendering artifacts (Fig. 1(a, b)). To overcome these problems, a naive solution would be training another MVS-based NeRF with more input views to construct the cost volume. Nevertheless, this solution requires training a new model with larger memory consumption, but even so, the input views could still be insufficient in inference time. Therefore, we proposed a novel method considering multiple cost volumes while rendering novel views.

3.2 3D Visibility Scores and 2D Visibility Masks

By taking I reference views into account in constructing a single cost volume, the maximum number of cost volumes we can refer to is $C_I^N = \binom{N}{I} = \frac{N(N-1)\cdots(N-I+1)}{I(I-1)\cdots 1}$ for each target view, where N is the number of reference views. However, utilizing all cost volumes results in high memory consumption and also leads to inefficient rendering. To tackle this challenge, we propose a method to select those

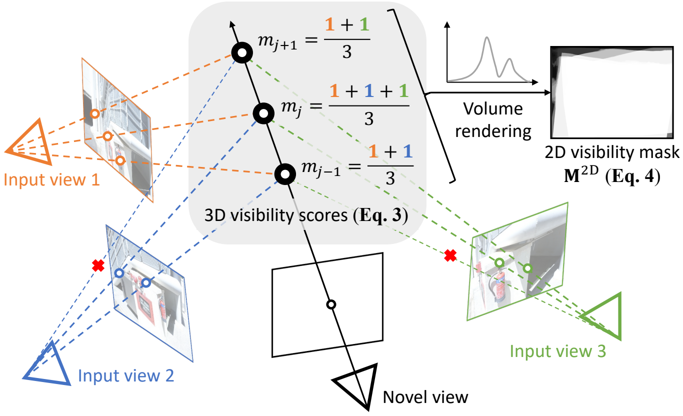

$m_{j+1} = \frac{1+1}{3}$
$m_j = \frac{1+1+1}{3}$
$m_{j-1} = \frac{1+1}{3}$
3D visibility scores (**Eq. 3**)
Volume rendering
2D visibility mask
$\mathbf{M}^{2D}$ (**Eq. 4**)
Input view 1
Input view 2
Input view 3
Novel view

Fig. 2: 3D visibility scores and 2D visibility masks. For a novel view, depth distribution is estimated from three input views, from which 3D points are sampled and projected onto each view to determine visibility. These projections yield 3D visibility scores m_j, normalized across the views, and are subsequently volume rendered into a 2D visibility mask $\mathbf{M}^{2D}$. This mask highlights the contribution of each input view to the cost volume and guides the rendering process, aiding in the selection of input views that optimize rendering quality and field of view coverage.

cost volumes with the largest contribution to viewport coverage and potential enhancement of rendering quality for novel views. To evaluate the contribution of each cost volume, we present *multi-view 3D visibility scores* as a metric.

For each sample point in a cost volume, we calculate its corresponding 3D visibility scores (the gray-shaded part in Fig. 2). These scores quantify the level of observation from various cost volumes, serving as a measurement of visibility. To calculate the 3D visibility scores of a single cost volume in a rendered view, we sample rays from the rendered view and aggregate the visibility weight from the reference views. Let I represent the total number of reference views. We use $\mathbb{1}_i(p)$ to indicate whether a sample point p is in the viewport of reference view i (bottom part in Fig. 2). The 3D visibility scores m_j are calculated using the formula:

$$m_j = \frac{\sum_{i=1}^{I} \mathbb{1}_i(p)}{I}, \tag{3}$$

where the subscript j denotes the sampled 3D point index along the ray, and the output 3D visibility scores range from 0 to 1. Each point on the mask indicates its 3D visibility score, with larger values reflecting higher confidence in the information at a specific sample point. The visibility score can be utilized as the weight for the feature of a point on a specific cost volume. Therefore, with the 3D visibility scores, we can combine the results from different cost volumes when volume rendering.

After obtaining 3D visibility scores for each cost volume, we propose the *2D visibility mask*. The 2D visibility is constructed by volume rendering the 3D metrics scores to novel view, as shown in Fig. 2. Similar to Eq. 2, given ray $\mathbf{r}$

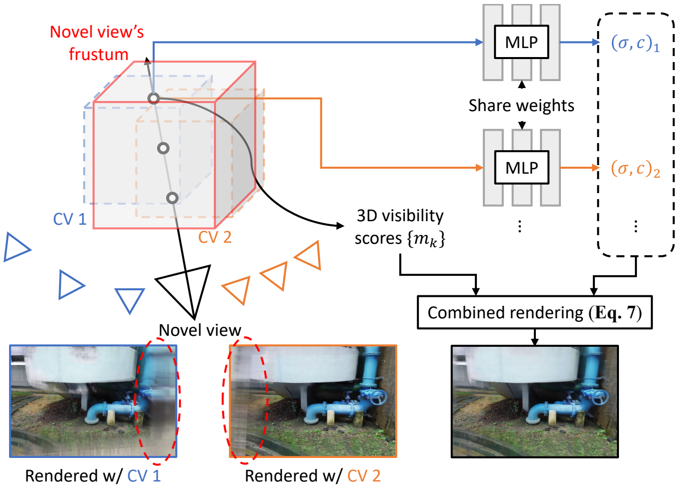


Fig. 3: Combined rendering from multiple cost volumes. Using a single cost volume, as in traditional MVS-based NeRFs, often introduces padding artifacts or incorrect geometry, as indicated by the **red** dashed circles. Our method warps selected cost volumes to the novel view's frustum and applies 3D visibility scores m_j as weights to blend multiple cost volumes during volume rendering. Combined rendering provides broader viewport coverage and combines information from multiple cost volumes, leading to improved image synthesis and alleviating artifacts.

from the camera center o with direction d, the value of 2D visibility mask is given by:

$$\mathbf{M}^{\mathrm{2D}}(\mathbf{r}) = \sum_{j=1}^{J} T'(j)\alpha\left(m_j\delta_j\right)m_j, \tag{4}$$

where $T'(j) = \exp(-\sum_{s=1}^{j-1} m_s\delta_s)$ is the transmitte considering 3D visibility scores. The 2D visibility mask will be used in cost volume selection; we will thoroughly discuss it in Sec. 3.4.

3.3 Rendering by Combining Multiple Cost Volumes

Our proposed rendering differs from the traditional one (Eq. 2) by considering 3D visibility scores and combining multiple cost volumes. Below, we explain the modifications we make. First, let us only consider a single cost volume for simplicity. The pixel color output by considering only a single cost volume is given by:

$$C_{\mathrm{single}}(\mathbf{r}) = \sum_{j=1}^{J} T_{\mathrm{single}}(j)\alpha\left(\sigma_j\delta_j\right)m_jc_j, \tag{5}$$

$$T_{\mathrm{single}}(j) = \exp\left(-\sum_{s=1}^{j-1}\left(\sigma_s\delta_s - \ln m_s\right)\right). \tag{6}$$

To further consider multiple cost volumes and also utilize their corresponding 3D visibility scores, we modify Eq. 5 to combine the result across multiple cost volumes. The final proposed volume rendering is given by:

$$C(\mathbf{r}) = \sum_{k=1}^{K} \sum_{j=1}^{J} T_{\text{combined}}(j) \alpha \left(\sigma_j^k \delta_j \right) M_j^k c_j^k, \tag{7}$$

$$T_{\text{combined}}(j) = \sum_{k=1}^{K} \exp \left(- \sum_{s=1}^{j-1} \left(\sigma_s^k \delta_s - \ln M_s^k \right) \right), \tag{8}$$

where K is the number of selected cost volumes, and $M_j^k = \frac{m_j^k}{\sum_{k=1}^{K} m_j^k}$ is the normalized 3D visibility score so that the summation of 3D visibility scores over selected cost volumes equals 1.

The illustration and effect of combining multiple cost volumes in rendering is shown in Fig. 3. Existing MVS-based NeRFs use a single cost volume to render novel views that contain padding artifacts and wrong geometry. Combining multiple cost volumes in rendering alleviates these artifacts and broadens the viewport coverage of novel views, thus improving the rendering quality.

3.4 Support Cost Volume Set Selection

As mentioned in Sec. 3.3, we only select K cost volumes for combined rendering to optimize rendering efficiency. Ideally, combining selected K cost volumes should provide maximum coverage for the rendered view. This problem can be formulated as *maximum coverage problem*, which is NP-hard. Thus, to complete view selection in polynomial time, we propose a greedy algorithm to construct a support set $\mathbf{S}$ of K cost volumes in **Algorithm 1**. Nemhauser *et al.* [44] also proved that the greedy algorithm is the optimal algorithm in polynomial time.

We show an example of the proposed selection algorithm in Fig. 4. At the beginning of the algorithm, our method selects the cost volume with the largest coverage score of the corresponding 2D visibility mask. The rendered image contains padding artifacts near the image boundaries as the viewport of this single cost volume is limited. Later on, our selection algorithm gradually selects the cost volumes that could maximize the visibility coverage and, therefore, enlarge the valid region of the rendered view. As a result, the rendering quality of novel views progressively grows as more cost volumes are selected and combined in the volume rendering.

3.5 End-to-end Fine-tuning

Our method is compatible with any MVS-based NeRFs to boost the rendering quality. Moreover, our approach is not optimized for a specific scene and could be generalized to new scenes, allowing it to enhance any end-to-end fine-tunable model. Fine-tuning refines geometry and color consistency within cost volumes and eliminates padding artifacts through combined rendering from multiple cost volumes. Thus, our method could augment the capabilities of advanced MVS-based NeRFs beyond ENeRF and MVS-NeRF.

Algorithm 1 Support cost volume set selection algorithm

```
Input: ${\mathbf{CV}_n}_{n=1}^{N}$: $N$ candidate cost volumes
Input: ${\mathbf{M}_n^{\text{2D}}}_{n=1}^{N}$: 2D visibility masks
Output: $\mathbf{S}$: a support set of $K$ cost volumes
$\mathbf{S} \leftarrow \varnothing$                                  ▷ Initialize the support CV set as an empty set
$\mathbf{P}_0 \leftarrow$ 2D Mask filled with ones                    ▷ Initialize the view coverage
while $|\mathbf{S}| < K$ do
    best_idx $\leftarrow 0$
    max_ratio $\leftarrow 0$
    $i \leftarrow 1$                                                  ▷ Initialize selection iteration
    while $i \leq N$ do
        if $\mathbf{CV}_i \notin \mathbf{S}$ then                     ▷ Consider remaining views only
            ratio $\leftarrow \sum (\mathbf{P}_{i-1} \cdot \mathbf{M}_i^{\text{2D}})$
            if ratio > max_ratio then
                max_ratio $\leftarrow$ ratio
                best_idx $\leftarrow i$
            end if
        end if
        $i \leftarrow i + 1$
    end while
    $\mathbf{P}_i \leftarrow \mathbf{P}_{i-1} \cdot (1 - \mathbf{M}_{\text{best\_idx}}^{\text{2D}})$   ▷ Update the view coverage
    $\mathbf{S} \leftarrow \mathbf{S} \cup \{\mathbf{CV}_{\text{best\_idx}}\}$                    ▷ Add the best CV to the set
end while
```

4 Experiments

4.1 Experimental Settings

Datasets. We evaluate two datasets: (1) the Free dataset collected by F2-NeRF [69] and (2) the ScanNet [14] dataset. The Free dataset consists of seven challenging scenes featuring narrow, long camera trajectories and focused foreground objects. Our evaluations on the Free dataset follow the train/test split in F2-NeRF [69] by using one-eighth of the images for testing and the rest for training. As for the ScanNet dataset, we strictly follow the train/test splits as defined in NeRFusion [91], NerfingMVS [72], and SurfelNeRF [20], with eight large-scale indoor scenes. We assess the rendering quality with PSNR, SSIM [71], and LPIPS [90] metrics.

Baselines. We compare BoostMVSNeRFs with various state-of-the-art NeRFs, including fast per-scene optimization NeRFs such as F2-NeRF [69] and Zip-NeRF [4] and generalizable NeRFs such as MVSNeRF [9], ENeRF [32] and SurfelNeRF [20].

In particular, F2-NeRF excels in outdoor scenes with free camera trajectories. Our method employs cost volume representations similar to MVSNeRF and ENeRF but enlarges valid visible regions by fusing multiple cost volumes. Although SurfelNeRF also proposes fusing multiple surfels as a type of 3D representation,

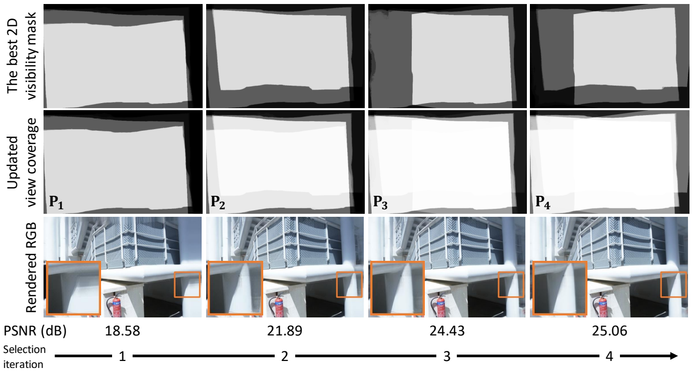


Fig. 4: Support cost volume set selection. Initially, our greedy algorithm selects a single cost volume, providing maximum coverage yet insufficient to prevent padding artifacts (**orange** boxes). Subsequent iterations incorporate additional cost volumes, progressively expanding view coverage, and improving image quality, as indicated by the increasing PSNR values.

the fusion method and its scene representation differ from BoostMVSNeRFs. To ensure fairness, we used the same experimental settings as in previous studies and used official codes where possible. All the training, fine-tuning, and evaluations are done on a single RTX 4090 GPU.

Our method is compatible with MVS-based techniques, allowing us to employ pre-trained models such as MVSNeRF and ENeRF in our experiments. Unless otherwise specified, we use ENeRF as our backbone MVS-based NeRF method in all the experiments. We optimize the parameters, $N = 6$, $I = 3$, and $K = 4$, for efficient rendering and high quality. Our method achieves similar runtime performance in rendering and fine-tuning as other generalizable NeRF methods but renders significantly improved quality.

4.2 Comparison with State-of-the-art Methods

Free Dataset. On the Free dataset, BoostMVSNeRFs emerges as the best among no per-scene and per-scene optimization NeRF methods as shown in Table 1 and Fig. 5. Compared to F2-NeRF and SurfelNeRF, which produced blurred images, BoostMVSNeRFs leverages multiple cost volume fusion and view selection based on visibility maps for superior rendering quality. Our method demonstrates compatibility with various camera trajectories and achieves results comparable to those of existing methods.

Our method outperforms generalizable NeRF techniques like MVSNeRF and ENeRF on the Free dataset (Table 1), enhancing rendering quality through our view selection and multiple cost volume combined rendering approach. Integrated with MVS-based NeRFs, our method achieves a PSNR improvement of 0.5-1.0 dB without requiring additional training. End-to-end fine-tuning on test scenes further enhances rendering quality, particularly in regions where a sin-

Table 1: Quantitative comparisons with state-of-the-art methods on the Free [69] dataset.

Method	Setting	PSNR ↑	SSIM ↑	LPIPS ↓	FPS ↑
MVSNeRF [9]	No per-scene optimization	20.06	0.721	0.469	1.79
MVSNeRF + Ours		20.52	0.722	0.470	1.26
ENeRF [32]		23.24	0.844	0.225	**9.90**
ENeRF+Ours		**24.21**	**0.862**	**0.218**	5.51
F2-NeRF [69]	Per-scene optimization	25.55	0.776	0.278	3.75
Zip-NeRF [4]		25.90	0.772	0.241	0.66
MVSNeRF$_{\text{ft}}$ [9]		20.49	0.698	0.425	1.79
MVSNeRF + Ours$_{\text{ft}}$		21.59	0.759	0.265	1.26
ENeRF$_{\text{ft}}$ [32]		25.19	0.880	0.180	**9.90**
ENeRF+Ours$_{\text{ft}}$		**26.14**	**0.894**	**0.171**	5.51

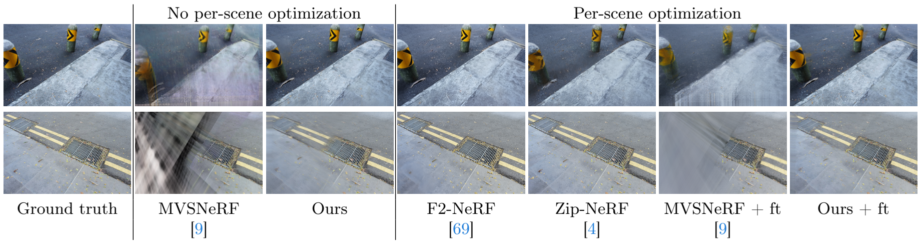


Fig. 5: Qualitative comparisons of rendering quality on the Free [69] dataset.

gle cost volume falls short. This highlights the benefit of multiple-cost volume fusion. For detailed visual comparisons, please refer to Fig. 6.

ScanNet Dataset. We conducted a comprehensive comparison of BoostMVSNeRFs with other state-of-the-art methods in no per-scene and per-scene optimization settings on the ScanNet dataset in Table 2. BoostMVSNeRFs demonstrates superior performance with a PSNR of 31.73 dB in no per-scene optimization, outperforming SurfelNeRF due to its cost volume fusion and efficient view selection strategy. In per-scene optimization, BoostMVSNeRFs excels again with a PSNR of 32.87 dB, indicating its effectiveness in cost volume fusion and per-scene adaptation. We also compare our method with two generalizable NeRF methods, MVSNeRF and ENeRF, on the ScanNet dataset in Table 2. Our method achieves better rendering quality over existing MVS-based NeRF methods in SSIM without per-scene optimization and in PSNR and LPIPS with per-scene fine-tuning.

Furthermore, our approach showed impressive results on large-scale scenes, outperforming SurfelNeRF in both direct inference and per-scene fine-tuning. Unlike SurfelNeRF, which suffered from artifacts due to its surfel-based rendering approach, our model's multiple cost volume fusion and efficient view information selection and aggregation led to high-quality and consistent renderings, as shown

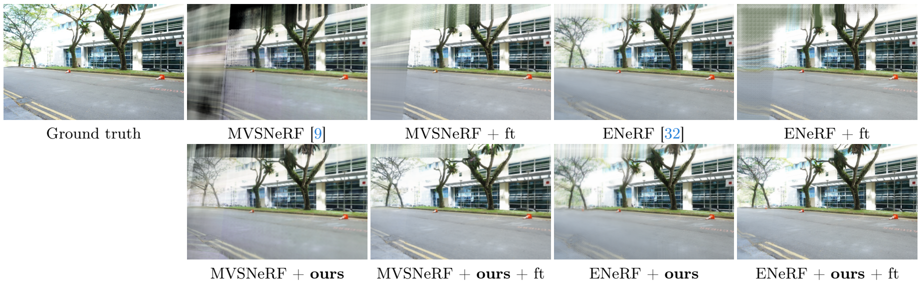


Fig. 6: Qualitative rendering quality improvements of integrating our method into MVS-based NeRF methods on the Free dataset.

Table 2: Quantitative comparisons with state-of-the-art methods on the ScanNet [14] dataset.

Method	Setting	PSNR ↑	SSIM ↑	LPIPS ↓	FPS ↑
SurfelNeRF [20]	No per-scene optimization	19.28	0.623	0.528	1.25
MVSNeRF [9]		23.40	0.862	0.367	1.99
MVSNeRF + Ours		23.66	0.872	0.365	1.41
ENeRF [32]		**31.73**	0.955	**0.206**	**11.03**
ENeRF + Ours		31.01	**0.957**	0.219	6.14
F2-NeRF [69]	Per-scene optimization	28.11	0.894	0.230	4.18
SurfelNeRF$_{\text{ft}}$ [20]		20.04	0.653	0.504	1.25
Zip-NeRF [4]		32.24	0.917	0.214	0.74
MVSNeRF$_{\text{ft}}$ [9]		24.69	0.872	0.316	1.99
MVSNeRF + Ours$_{\text{ft}}$		24.63	0.880	0.320	1.41
ENeRF$_{\text{ft}}$ [32]		32.70	**0.960**	0.174	**11.03**
ENeRF + Ours$_{\text{ft}}$		**32.87**	0.955	**0.173**	6.14

in Fig. 5. This indicates our cost volume fusion's effectiveness in reconstructing large-scale scenes efficiently and accurately.

4.3 Ablation Study

Cost Volume Selection Scheme. In Sec. 3.4, we propose a greedy method to select the cost volumes that will approximately maximize the view coverage. To validate the effectiveness of our method, We conducted experiments comparing two other cost volume selection methods. These two methods are: (a) selecting K cost volumes that are closest to the render view pose, which is adopted by ENeRF [32] and (b) selecting corresponding cost volumes directly with the highest contribution of 2D visibility mask. In particular, method (b) is a degenerate version of method our proposed selection method (c), which is based on view coverage. Table 3 shows that our greedy cost volume selection method performs better than the other two methods.

Table 3: Ablation of the cost volume selection. We compare three different strategies for cost volume selection on all scenes of the Free [69] dataset: (a) ENeRF's method, which is based on pose distance, (b) direct selection of cost volumes with maximum visibility, and (c) Our proposed greedy method, which maximizes the visibility coverage.

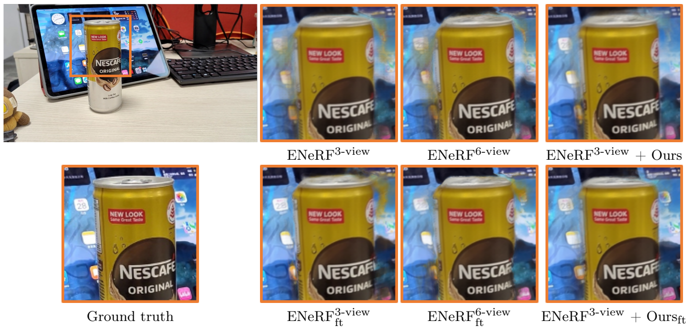
Method	PSNR ↑	SSIM ↑	LPIPS ↓
(a) ENeRF [32]	24.09	0.861	0.220
(b) Maximize 2D visibility $\mathbf{M}_i^{2D}$	24.19	0.861	0.218
(c) Maximize view coverage $\mathbf{P}_i$	**24.21**	**0.862**	**0.218**

Table 4: Different ways of combining more input views. We compare training an MVS-based NeRF with a larger number of input views (6 input views here) and our proposed cost volume selection and combined rendering on all scenes of the Free [69] dataset.

Method	Setting	PSNR ↑	SSIM ↑	LPIPS ↓
$\text{ENeRF}^{\text{3-view}}$ [32]	No per-scene optimization	23.24	0.844	0.225
$\text{ENeRF}^{\text{6-view}}$ [32]		23.53	0.770	0.231
$\text{ENeRF}^{\text{3-view}}$ + Ours		**24.21**	**0.862**	**0.218**
$\text{ENeRF}^{\text{3-view}}_{\text{ft}}$ [32]	Per-scene optimization	25.19	0.880	0.180
$\text{ENeRF}^{\text{6-view}}_{\text{ft}}$ [32]		25.61	0.840	0.172
$\text{ENeRF}^{\text{3-view}}$ + Ours_{ft}		**26.14**	**0.894**	**0.171**

Single Cost Volume with More Input Views vs. Combining Multiple Cost Volumes In our method, we select multiple cost volumes and combine them in volume rendering, while ENeRF only forms one cost volume. To examine our method's effectiveness, we train ENeRF (originally three input views) with more input views (6 in this ablation, in order to evenly compare with our proposed method). The results are shown in Table 4 and Fig. 7. We can see an increase in the number of input views which requires time-consuming training to construct a single cost volume. However, the rendering quality improvements are subtle both with or without per-scene fine-tuning. In contrast, our cost volume selection method and combined rendering scheme improve the rendering quality by a large margin and could be further optimized with per-scene fine-tuning.

Robustness with Sparse Input Views. Our proposed combined rendering from multiple cost volumes addresses the challenges of reconstructing large-scale and unbounded scenes due to broader viewport coverage. Therefore, our method could be more robust to sparse input views as more and farther cost volumes are considered during rendering. We conduct an experiment comparing performance across various degrees of sparse views to demonstrate the robustness of our method with sparse input views. Specifically, we uniformly sub-sample the training views and evaluate the rendering quality. The results show a more sig-



ENeRF$^{\text{3-view}}$ ENeRF$^{\text{6-view}}$ ENeRF$^{\text{3-view}}$ + Ours

Ground truth ENeRF$^{\text{3-view}}_{\text{ft}}$ ENeRF$^{\text{6-view}}_{\text{ft}}$ ENeRF$^{\text{3-view}}$ + Ours$_{\text{ft}}$

Fig. 7: Visual effects of different ways of combining more input views. Artifacts in disocclusion regions cannot be resolved by including more input views for a single cost volume. Our method could alleviate these artifacts by combining more cost volumes in rendering.

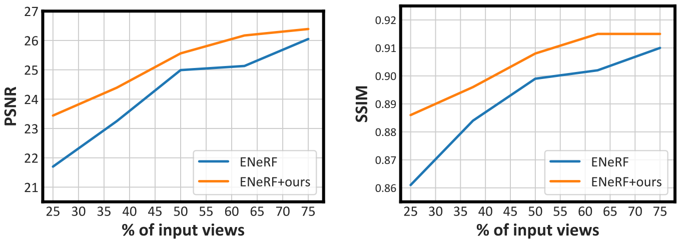


Fig. 8: Robustness with sparse input views. With more sparse input views, the performance drop of our method is less severe than ENeRF, demonstrating the robustness of our method against sparse input views by combining multiple cost volumes in rendering.

nificant decline in both PSNR and SSIM for ENeRF compared to ours while input views become sparse, as indicated by the curve in figure 8.

5 Conclusion

In summary, our BoostMVSNeRFs enhances MVS-based NeRFs, tackling large-scale and unbounded scene rendering challenges. Utilizing 3D visibility scores for multi-cost volume integration, BoostMVSNeRFs synthesizes significantly better novel views, enhancing viewport coverage and minimizing typical single-cost volume artifacts. Compatible with current MVS-based NeRFs, BoostMVS-NeRFs supports end-to-end training for scene-specific enhancement. Experimental results validate the efficacy of our method in boosting advanced MVS-based NeRFs, contributing to more scalable and high-quality view synthesis. Future work will focus on reducing MVS dependency and optimizing memory usage, furthering the field of neural rendering for virtual and augmented reality applications.

References

1. Aliev, K.A., Sevastopolsky, A., Kolos, M., Ulyanov, D., Lempitsky, V.: Neural point-based graphics. In: ECCV (2020) 2
2. Barron, J.T., Mildenhall, B., Tancik, M., Hedman, P., Martin-Brualla, R., Srinivasan, P.P.: Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In: ICCV (2021) 1, 2
3. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In: CVPR (2022) 1, 2
4. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Zip-nerf: Anti-aliased grid-based neural radiance fields. In: ICCV (2023) 1, 2, 9, 11, 12
5. Boss, M., Braun, R., Jampani, V., Barron, J.T., Liu, C., Lensch, H.: Nerd: Neural reflectance decomposition from image collections. In: ICCV (2021) 3
6. Cao, A., Rockwell, C., Johnson, J.: Fwd: Real-time novel view synthesis with forward warping and depth. In: CVPR (2022) 2, 3
7. Chaurasia, G., Duchene, S., Sorkine-Hornung, O., Drettakis, G.: Depth synthesis and local warps for plausible image-based navigation. ACM TOG (2013) 2
8. Chen, A., Xu, Z., Geiger, A., Yu, J., Su, H.: Tensorf: Tensorial radiance fields. In: ECCV (2022) 3
9. Chen, A., Xu, Z., Zhao, F., Zhang, X., Xiang, F., Yu, J., Su, H.: Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In: ICCV (2021) 2, 3, 4, 5, 9, 11, 12
10. Chen, R., Han, S., Xu, J., Su, H.: Point-based multi-view stereo network. In: ICCV (2019) 3
11. Chen, Y., Xu, H., Wu, Q., Zheng, C., Cham, T.J., Cai, J.: Explicit correspondence matching for generalizable neural radiance fields. arXiv preprint arXiv:2304.12294 (2023) 4
12. Cheng, B.Y., Chiu, W.C., Liu, Y.L.: Improving robustness for joint optimization of camera poses and decomposed low-rank tensorial radiance fields. In: AAAI (2024) 1
13. Chibane, J., Bansal, A., Lazova, V., Pons-Moll, G.: Stereo radiance fields (srf): Learning view synthesis for sparse views of novel scenes. In: CVPR (2021) 4
14. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: CVPR (2017) 2, 9, 12
15. Debevec, P.E., Taylor, C.J., Malik, J.: Modeling and rendering architecture from photographs: A hybrid geometry-and image-based approach. In: Seminal Graphics Papers: Pushing the Boundaries, Volume 2 (2023) 2
16. Deng, K., Liu, A., Zhu, J.Y., Ramanan, D.: Depth-supervised nerf: Fewer views and faster training for free. In: CVPR (2022) 4
17. Dhamo, H., Tateno, K., Laina, I., Navab, N., Tombari, F.: Peeking behind objects: Layered depth prediction from a single image. Pattern Recognition Letters (2019) 2
18. Flynn, J., Broxton, M., Debevec, P., DuVall, M., Fyffe, G., Overbeck, R., Snavely, N., Tucker, R.: Deepview: View synthesis with learned gradient descent. In: CVPR (2019) 2
19. Flynn, J., Neulander, I., Philbin, J., Snavely, N.: Deepstereo: Learning to predict new views from the world's imagery. In: CVPR (2016) 2
20. Gao, Y., Cao, Y.P., Shan, Y.: Surfelnerf: Neural surfel radiance fields for online photorealistic reconstruction of indoor scenes. In: CVPR (2023) 4, 9, 12

21. Gortler, J.S.S., He, L.w., Szeliski, R., et al.: Layered depth images. In: SIGGRAPH (1998) 2
22. Gu, X., Fan, Z., Zhu, S., Dai, Z., Tan, F., Tan, P.: Cascade cost volume for high-resolution multi-view stereo and stereo matching. In: CVPR (2020) 3, 5
23. Hedman, P., Srinivasan, P.P., Mildenhall, B., Barron, J.T., Debevec, P.: Baking neural radiance fields for real-time view synthesis. In: ICCV (2021) 2
24. Jain, A., Tancik, M., Abbeel, P.: Putting nerf on a diet: Semantically consistent few-shot view synthesis. In: ICCV (2021) 4
25. Jiang, Y., Ji, D., Han, Z., Zwicker, M.: Sdfdiff: Differentiable rendering of signed distance fields for 3d shape optimization. In: CVPR (2020) 2
26. Johari, M.M., Lepoittevin, Y., Fleuret, F.: Geonerf: Generalizing nerf with geometry priors. In: CVPR (2022) 4
27. Kalantari, N.K., Wang, T.C., Ramamoorthi, R.: Learning-based view synthesis for light field cameras. ACM TOG (2016) 2
28. Kim, M., Seo, S., Han, B.: Infonerf: Ray entropy minimization for few-shot neural volume rendering. In: CVPR (2022) 4
29. Li, T., Slavcheva, M., Zollhoefer, M., Green, S., Lassner, C., Kim, C., Schmidt, T., Lovegrove, S., Goesele, M., Newcombe, R., et al.: Neural 3d video synthesis from multi-view video. In: CVPR (2022) 3
30. Li, Z., Xian, W., Davis, A., Snavely, N.: Crowdsampling the plenoptic function. In: ECCV (2020) 2
31. Lin, H., Peng, S., Xu, Z., Xie, T., He, X., Bao, H., Zhou, X.: Im4d: High-fidelity and real-time novel view synthesis for dynamic scenes. arXiv preprint arXiv:2310.08585 (2023) 3
32. Lin, H., Peng, S., Xu, Z., Yan, Y., Shuai, Q., Bao, H., Zhou, X.: Efficient neural radiance fields for interactive free-viewpoint video. In: SIGGRAPH Asia (2022) 2, 3, 5, 9, 11, 12, 13
33. Lin, K.E., Lin, Y.C., Lai, W.S., Lin, T.Y., Shih, Y.C., Ramamoorthi, R.: Vision transformer for nerf-based view synthesis from a single input image. In: WACV (2023) 4
34. Liu, L., Xu, W., Zollhoefer, M., Kim, H., Bernard, F., Habermann, M., Wang, W., Theobalt, C.: Neural rendering and reenactment of human actor videos. ACM TOG (2019) 2
35. Liu, Y.L., Gao, C., Meuleman, A., Tseng, H.Y., Saraf, A., Kim, C., Chuang, Y.Y., Kopf, J., Huang, J.B.: Robust dynamic radiance fields. In: CVPR (2023) 3
36. Liu, Y., Peng, S., Liu, L., Wang, Q., Wang, P., Theobalt, C., Zhou, X., Wang, W.: Neural rays for occlusion-aware image-based rendering. In: CVPR (2022) 4
37. Lombardi, S., Simon, T., Saragih, J., Schwartz, G., Lehrmann, A., Sheikh, Y.: Neural volumes: Learning dynamic renderable volumes from images. ACM TOG (2019) 2
38. Lombardi, S., Simon, T., Schwartz, G., Zollhoefer, M., Sheikh, Y., Saragih, J.: Mixture of volumetric primitives for efficient neural rendering. ACM TOG (2021) 2
39. Meuleman, A., Liu, Y.L., Gao, C., Huang, J.B., Kim, C., Kim, M.H., Kopf, J.: Progressively optimized local radiance fields for robust view synthesis. In: CVPR (2023) 1, 2
40. 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 TOG (2019) 2

41. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: ECCV (2020) 1, 2
42. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM TOG (2022) 3
43. Munkberg, J., Hasselgren, J., Shen, T., Gao, J., Chen, W., Evans, A., Müller, T., Fidler, S.: Extracting triangular 3d models, materials, and lighting from images. In: CVPR (2022) 3
44. Nemhauser, G.L., Wolsey, L.A., Fisher, M.L.: An analysis of approximations for maximizing submodular set functions—i. Mathematical programming (1978) 8
45. Niemeyer, M., Barron, J.T., Mildenhall, B., Sajjadi, M.S., Geiger, A., Radwan, N.: Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In: CVPR (2022) 4
46. Oechsle, M., Peng, S., Geiger, A.: Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In: ICCV (2021) 3
47. Park, K., Sinha, U., Barron, J.T., Bouaziz, S., Goldman, D.B., Seitz, S.M., Martin-Brualla, R.: Nerfies: Deformable neural radiance fields. In: ICCV (2021) 2, 3
48. Penner, E., Zhang, L.: Soft 3d reconstruction for view synthesis. ACM TOG (2017) 2
49. Pfister, H., Zwicker, M., Van Baar, J., Gross, M.: Surfels: Surface elements as rendering primitives. In: Proceedings of the 27th annual conference on Computer graphics and interactive techniques (2000) 4
50. Pumarola, A., Corona, E., Pons-Moll, G., Moreno-Noguer, F.: D-nerf: Neural radiance fields for dynamic scenes. In: CVPR (2021) 3
51. Riegler, G., Koltun, V.: Free view synthesis. In: ECCV (2020) 2
52. Roessle, B., Barron, J.T., Mildenhall, B., Srinivasan, P.P., Nießner, M.: Dense depth priors for neural radiance fields from sparse input views. In: CVPR (2022) 4
53. Seo, S., Han, D., Chang, Y., Kwak, N.: Mixnerf: Modeling a ray with mixture density for novel view synthesis from sparse inputs. In: CVPR (2023) 4
54. Shi, Y., Rong, D., Ni, B., Chen, C., Zhang, W.: Garf: Geometry-aware generalized neural radiance field. arXiv preprint arXiv:2212.02280 (2022) 4
55. Shih, M.L., Su, S.Y., Kopf, J., Huang, J.B.: 3d photography using context-aware layered depth inpainting. In: CVPR (2020) 2
56. Sitzmann, V., Thies, J., Heide, F., Nießner, M., Wetzstein, G., Zollhofer, M.: Deepvoxels: Learning persistent 3d feature embeddings. In: CVPR (2019) 2
57. Somraj, N., Soundararajan, R.: Vip-nerf: Visibility prior for sparse input neural radiance fields (2023) 4
58. Srinivasan, P.P., Tucker, R., Barron, J.T., Ramamoorthi, R., Ng, R., Snavely, N.: Pushing the boundaries of view extrapolation with multiplane images. In: CVPR (2019) 2
59. Sun, C., Sun, M., Chen, H.T.: Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In: CVPR (2022) 3
60. Tancik, M., Casser, V., Yan, X., Pradhan, S., Mildenhall, B., Srinivasan, P.P., Barron, J.T., Kretzschmar, H.: Block-nerf: Scalable large scene neural view synthesis. In: CVPR (2022) 1, 2
61. Thies, J., Zollhöfer, M., Nießner, M.: Deferred neural rendering: Image synthesis using neural textures. ACM TOG (2019) 2
62. Trevithick, A., Yang, B.: Grf: Learning a general radiance field for 3d representation and rendering. In: ICCV (2021) 4

63. Tucker, R., Snavely, N.: Single-view view synthesis with multiplane images. In: CVPR (2020) 2
64. Tulsiani, S., Tucker, R., Snavely, N.: Layer-structured 3d scene inference via view synthesis. In: ECCV (2018) 2
65. Uy, M.A., Martin-Brualla, R., Guibas, L., Li, K.: Scade: Nerfs from space carving with ambiguity-aware depth estimates. In: CVPR (2023) 4
66. Waechter, M., Moehrle, N., Goesele, M.: Let there be color! large-scale texturing of 3d reconstructions. In: ECCV (2014) 2
67. Wang, G., Chen, Z., Loy, C.C., Liu, Z.: Sparsenerf: Distilling depth ranking for few-shot novel view synthesis. In: ICCV (2023) 4
68. Wang, P., Liu, L., Liu, Y., Theobalt, C., Komura, T., Wang, W.: Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. In: NeurIPS (2021) 3
69. Wang, P., Liu, Y., Chen, Z., Liu, L., Liu, Z., Komura, T., Theobalt, C., Wang, W.: F2-nerf: Fast neural radiance field training with free camera trajectories. In: CVPR (2023) 2, 9, 11, 12, 13
70. Wang, Q., Wang, Z., Genova, K., Srinivasan, P.P., Zhou, H., Barron, J.T., Martin-Brualla, R., Snavely, N., Funkhouser, T.: Ibrnet: Learning multi-view image-based rendering. In: CVPR (2021) 2, 3, 4
71. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE TIP (2004) 9
72. Wei, Y., Liu, S., Rao, Y., Zhao, W., Lu, J., Zhou, J.: Nerfingmvs: Guided optimization of neural radiance fields for indoor multi-view stereo. In: ICCV (2021) 9
73. Wizadwongsa, S., Phongthawee, P., Yenphraphai, J., Suwajanakorn, S.: Nex: Real-time view synthesis with neural basis expansion. In: CVPR (2021) 2
74. Wood, D.N., Azuma, D.I., Aldinger, K., Curless, B., Duchamp, T., Salesin, D.H., Stuetzle, W.: Surface light fields for 3d photography. In: Seminal Graphics Papers: Pushing the Boundaries, Volume 2 (2023) 2
75. Wu, R., Mildenhall, B., Henzler, P., Park, K., Gao, R., Watson, D., Srinivasan, P.P., Verbin, D., Barron, J.T., Poole, B., et al.: Reconfusion: 3d reconstruction with diffusion priors. arXiv preprint arXiv:2312.02981 (2023) 4
76. Wynn, J., Turmukhambetov, D.: Diffusionerf: Regularizing neural radiance fields with denoising diffusion models. In: CVPR (2023) 4
77. Xian, W., Huang, J.B., Kopf, J., Kim, C.: Space-time neural irradiance fields for free-viewpoint video. In: CVPR (2021) 3
78. Xu, Q., Xu, Z., Philip, J., Bi, S., Shu, Z., Sunkavalli, K., Neumann, U.: Point-nerf: Point-based neural radiance fields. In: CVPR (2022) 2, 3
79. Yang, J., Pavone, M., Wang, Y.: Freenerf: Improving few-shot neural rendering with free frequency regularization. In: CVPR (2023) 4
80. Yao, Y., Luo, Z., Li, S., Fang, T., Quan, L.: Mvsnet: Depth inference for unstructured multi-view stereo. In: ECCV (2018) 3, 5
81. Yao, Y., Luo, Z., Li, S., Shen, T., Fang, T., Quan, L.: Recurrent mvsnet for high-resolution multi-view stereo depth inference. In: CVPR (2019) 3
82. Yariv, L., Gu, J., Kasten, Y., Lipman, Y.: Volume rendering of neural implicit surfaces. In: NeurIPS (2021) 3
83. Yariv, L., Kasten, Y., Moran, D., Galun, M., Atzmon, M., Ronen, B., Lipman, Y.: Multiview neural surface reconstruction by disentangling geometry and appearance. In: NeurIPS (2020) 3
84. Yu, A., Fridovich-Keil, S., Tancik, M., Chen, Q., Recht, B., Kanazawa, A.: Plenoxels: Radiance fields without neural networks. In: CVPR (2022) 2, 3

85. Yu, A., Li, R., Tancik, M., Li, H., Ng, R., Kanazawa, A.: Plenoctrees for real-time rendering of neural radiance fields. In: ICCV (2021) 3
86. Yu, A., Ye, V., Tancik, M., Kanazawa, A.: pixelnerf: Neural radiance fields from one or few images. In: CVPR (2021) 2, 3, 4
87. Yu, Z., Gao, S.: Fast-mvsnet: Sparse-to-dense multi-view stereo with learned propagation and gauss-newton refinement. In: CVPR (2020) 3
88. Zhang, K., Luan, F., Wang, Q., Bala, K., Snavely, N.: Physg: Inverse rendering with spherical gaussians for physics-based material editing and relighting. In: CVPR (2021) 3
89. Zhang, K., Riegler, G., Snavely, N., Koltun, V.: Nerf++: Analyzing and improving neural radiance fields. arXiv preprint arXiv:2010.07492 (2020) 2
90. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR (2018) 9
91. Zhang, X., Bi, S., Sunkavalli, K., Su, H., Xu, Z.: Nerfusion: Fusing radiance fields for large-scale scene reconstruction. In: CVPR (2022) 4, 9
92. Zhang, X., Srinivasan, P.P., Deng, B., Debevec, P., Freeman, W.T., Barron, J.T.: Nerfactor: Neural factorization of shape and reflectance under an unknown illumination. ACM TOG (2021) 3
93. Zhou, T., Tucker, R., Flynn, J., Fyffe, G., Snavely, N.: Stereo magnification: Learning view synthesis using multiplane images (2018) 2
94. Zhu, B., Yang, Y., Wang, X., Zheng, Y., Guibas, L.: Vdn-nerf: Resolving shape-radiance ambiguity via view-dependence normalization. In: CVPR (2023) 4