MuRF: Multi-Baseline Radiance Fields

Haofei Xu[1,2]   Anpei Chen[1,2]   Yuedong Chen[3]   Christos Sakaridis[1]   Yulun Zhang[4]
Marc Pollefeys[1,5]   Andreas Geiger[2,†]   Fisher Yu[1,†]

[1]ETH Zurich [2]University of Tübingen, Tübingen AI Center [3]Monash University
[4]Shanghai Jiao Tong University [5]Microsoft [†]Joint last author

Abstract

We present Multi-Baseline Radiance Fields (MuRF), a general feed-forward approach to solving sparse view synthesis under multiple different baseline settings (small and large baselines, and different number of input views). To render a target novel view, we discretize the 3D space into planes parallel to the target image plane, and accordingly construct a target view frustum volume. Such a target volume representation is spatially aligned with the target view, which effectively aggregates relevant information from the input views for high-quality rendering. It also facilitates subsequent radiance field regression with a convolutional network thanks to its axis-aligned nature. The 3D context modeled by the convolutional network enables our method to synthesis sharper scene structures than prior works. Our MuRF achieves state-of-the-art performance across multiple different baseline settings and diverse scenarios ranging from simple objects (DTU) to complex indoor and outdoor scenes (RealEstate10K and LLFF). We also show promising zero-shot generalization abilities on the Mip-NeRF 360 dataset, demonstrating the general applicability of MuRF.

1. Introduction

Novel view synthesis from sparse input views is a critical and practical problem in computer vision and graphics. However, typical optimization-based Neural Radiance Fields (NeRFs) [1, 2, 4, 18] cannot cope well with sparse views, and additional regularizations [19, 29, 33] are usually required to better constrain the optimization process.

In this paper, we aim at learning feed-forward NeRF models [3, 32, 38] that are able to perform feed-forward inference on new unseen data, *without* requiring per-scene optimization. Moreover, inductive biases could be acquired from data and better results can potentially be attained.

To achieve this, existing sparse view methods can be broadly classified into methods that take small baseline images as input and methods that focus on large baseline settings. However, none of the existing methods works well across different baselines (see Fig. 1a and Fig. 1b). In par-
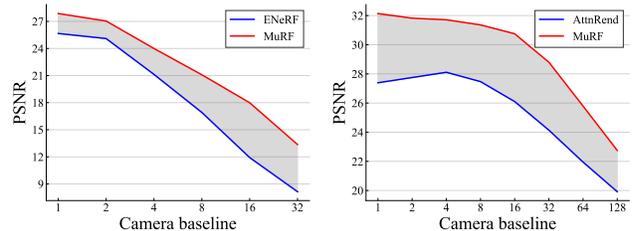
(a) MuRF outperforms previous state-of-the-art small baseline method ENeRF. The larger baseline, the larger performance gap.

(b) MuRF outperforms previous state-of-the-art large baseline method AttnRend. The smaller baseline, the larger performance gap.
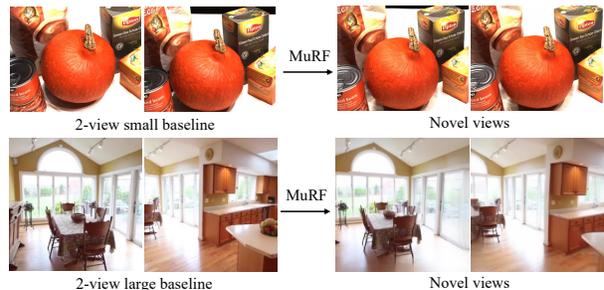

2-view small baseline — MuRF — Novel views

2-view large baseline — MuRF — Novel views

Figure 1. **MuRF supports multiple different baseline settings**. Previous methods are specifically designed for either small (*e.g.*, ENeRF [16]) or large (*e.g.*, AttnRend[1] [8]) baselines. However, no existing method performs well on both (see Table 1).

ticular, small baseline methods [3, 5, 6, 15–17, 32] rely heavily on feature matching to extract relevant information from input views and suffer significantly when there is insufficient image overlap or occlusion occurs. In contrast, large baseline methods [8, 24] are mostly data-driven, and often discard matching cues. Instead, they resort to large-scale datasets and generic architectures (*e.g.*, Transformers [7, 30]) to learn geometric inductive biases implicitly. However, due to their correspondence-unaware and data-driven nature, even state-of-the-art methods [8, 24] tend to produce blurry renderings and their generalization ability is still limited, as we show in Fig. 3 and Table 6.

Our goal is to address sparse view synthesis under both

---

[1]We name the previously unnamed method of Du et al. [8] as AttnRend, according its GitHub repository name.

small and large camera baselines, different number of input views, and diverse scenarios ranging from simple objects to complex scenes. To this end, we introduce Multi-Baseline Radiance Fields (MuRF), a method for sparse view synthesis from multiple different baseline settings (Fig. 1).

More specifically, to render a target novel view, we discretize the 3D space with planes parallel to the target image plane, and accordingly construct a *target view frustum volume* that is spatially aligned with the target view to render. This is a crucial difference to previous volume-based methods like MVSNeRF [3] and GeoNeRF [15], where the volumes are constructed in a *pre-defined reference input view* by straightforwardly following the standard practice in multi-view stereo (MVS). However, a key difference between MVS and novel view synthesis (NVS) is that MVS estimates the depth of a *reference input view*, while NVS aims to render a *novel view*. Taking this difference into account is crucial, since the reference view volume will not be able to effectively aggregate information from input views when the overlap between the reference and target view is small, which results in failure for large baselines (Fig. 5).

Our target view frustum volume is constructed by sampling multi-view input image colors and features, which provide appearance information to the prediction of the 3D point's color. We also compute the cosine similarities between sampled features to provide multi-view consistency cues to aid the prediction of volume density [5]. The sampled colors and features, and the computed cosine similarities are concatenated together to provide relevant information for both small and large baseline scenarios.

Equipped with this axis-aligned target volume representation, we further propose to use a convolutional neural network (CNN) to reconstruct the radiance field from this volume. Thanks to the context modeling abilities of CNNs, our method yields more accurate scene structures than previous (MLP-based) point-wise [6, 8] and (Ray Transformer-based) ray-wise methods [5, 27, 32] (see Fig. 6).

In practice, we perform $8\times$ subsampling in the spatial dimension when constructing the target volume, which enables our method to efficiently handle high-resolution images. The full resolution radiance field is finally obtained with a lightweight $8\times$ upsampler [25]. We also use a computationally more efficient 3D CNN alternative by factorizing the 3D $3\times3\times3$ convolution to a 2D $3\times3\times1$ convolution on the spatial dimension and a 1D $1\times1\times3$ convolution on the depth dimension, *i.e.*, (2+1)D CNN, a popular strategy in video recognition works [9, 28].

We conduct extensive experiments to demonstrate the effectiveness of our proposed target view frustum volume and 3D context-aware radiance field decoder. More specifically, we outperform previous specialized models on both small (DTU [14], LLFF [18]) and large (RealEstate10K [40]) baselines, and achieve promising zero-short generalization

abilities on the Mip-NeRF 360 [2] dataset, indicating the general applicability of our method.

2. Related Work

Multi-Baseline. The concept of "multi-baseline" dates back to multi-baseline stereo depth estimation [10, 20, 36, 37], where multiple stereo pairs with various baselines are used to surpass matching ambiguities and thus achieve improved precision. In this paper, we aim at designing a generally applicable approach to solving novel view synthesis from multiple different sparse view settings. When considering multiple images as input, our problem setup is somewhat similar to multi-baseline stereo, while with the radiance field as the output for volumetric rendering. Besides, we are also interested in novel view synthesis from very sparse (as few as 2) input views with different baselines. In this sense, our proposed multi-baseline radiance field not only draws connections to classic multi-baseline stereo, but also extends it to view synthesis from 2 input views with both small and large baselines.

Sparse Input Novel View Synthesis. Applying NeRF to novel view synthesis with sparse input views has garnered considerable interest in recent years. Several works improve the optimization-based pipeline via the incorporation of additional training regularizations, *e.g.*, depth [19], correspondence [29] and diffusion model [33]. An alternative area of research turns to the feed-forward designs by learning reliable representations from data, and accordingly removing the need of per-scene optimization. Among them, MVSNeRF [3], GeoNeRF [15] and NeuRay [17] follow the spirit of multi-view stereo (MVS) to construct cost volumes at pre-defined reference viewpoints. However, the reference volume will not be able to effectively aggregate the information for input views when the overlap between the reference and target novel view is small, which accordingly results in failure for large baselines. Besides, ENeRF [16] essentially relies on an MVS network to guide NeRF's sampling process with the estimated depth. However, the depth quality will be unreliable for large baselines and thus severely affects the subsequent rendering process.

Another line of works such as SRF [6], IBRNet [32], GPNR [26], GNT [27] and MatchNeRF [5] mostly perform ray-based rendering, where each ray is rendered independently and no explicit 3D context between different rays is modeled. In contrast, our target view volume representation effectively encodes the scene geometry and naturally enables 3D context modeling with a convolutional network, yielding better scene structures. Despite various motivations and implementations, the above models are all designed and experimented only on small baseline settings, limiting their applications in real-world scenarios. Only a few existing approaches focus on the large baseline settings. Nevertheless, these works [8, 24] resort to large-scale
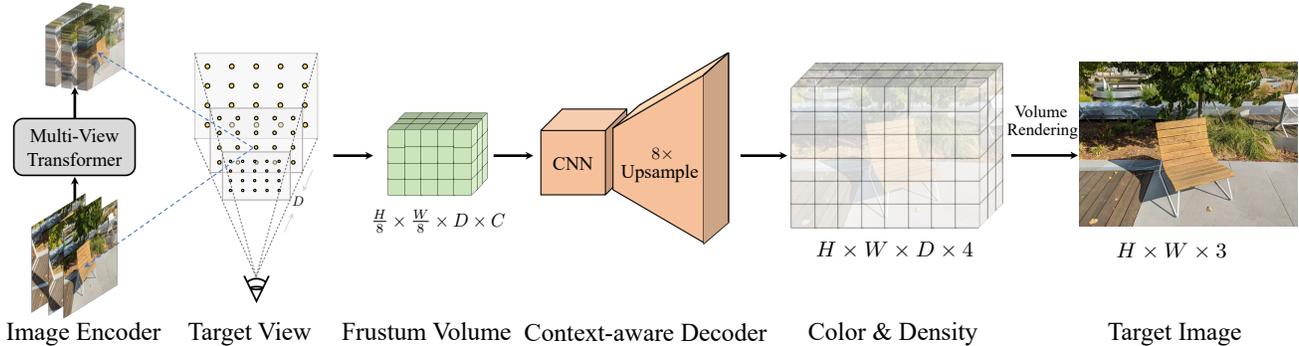
Figure 2. **Overview**. Given multiple input images, we first extract multi-view image features with a multi-view Transformer. To render a target image of resolution $H \times W$, we construct a target view frustum volume by performing $8\times$ subsampling in the spatial dimension while casting rays and sampling D equidistant points on each ray. For each 3D point, we sample feature and color information from the extracted feature maps and input images, which consists of the elements of the target volume $\boldsymbol{z} \in \mathbb{R}^{\frac{H}{8} \times \frac{W}{8} \times D \times C}$. Here, C denotes the channel dimension after aggregating sampled features and colors. To reconstruct the radiance field from the volume, we model the context information in the decoder with a (2+1)D CNN operating on low resolution and subsequently obtain the full-resolution radiance field with a lightweight $8\times$ upsampler. The target image is finally rendered with volumetric rendering.

datasets and generic architectures, without employing explicit 3D representations, which ultimately leads to blurry rendering and limited generalization. As a comparison, our MuRF is developed with both geometry-aware target view frustum volume representation and 3D context-aware convolutional network, which make it excel at both small and large baseline settings on different datasets.

3. Approach

Our MuRF (Multi-Baseline Radiance Fields) is an encoder-decoder architecture (Fig. 2), where the encoder maps the multi-view input images to features that are used to construct a volume at the target view's camera frustum, and the decoder regresses the radiance field from this volume representation. The target image is finally obtained using volume rendering [18]. The key components are introduced below.

3.1. Multi-View Feature Encoder

To aggregate essential information required as input for learning feed-forward NeRF models, we first extract features $\{\boldsymbol{F}_k\}_{k=1}^{K}$ from K input images $\{\boldsymbol{I}_k\}_{k=1}^{K}$. Our feature encoder consists of a weight-sharing per-image 2D CNN and a joint multi-view Transformer. The CNN consists of 6 residual blocks [12] and every 2 residual blocks include a stride-2 convolutional layer to achieve $2\times$ downsampling. Accordingly, we obtain features at $1/2$, $1/4$ and $1/8$ resolutions. The $1/8$ convolutional features are then fed into a joint multi-view Transformer to obtain features at $1/8$ resolution. Our Transformer is built on GMFlow [34, 35]'s 2-view Transformer architecture, where we extend the 2-view cross-attention to $K(K \geq 2)$ input views by performing cross-attention for all K views simultaneously. This facilitates more efficient processing of additional input views without significantly increasing memory footprint compared to the pair-wise architecture [5]. The multi-scale

features are sampled in next volume reconstruction step and the sampled features are concatenated together.

3.2. Target View Frustum Volume

To render an image for a target novel view, our key difference with previous methods [3, 15, 17] is that we construct our volume aligned with the *target view frustum*, instead of a pre-defined reference input view. Such a spatially-aligned target view frustum volume representation naturally enables effective aggregation of information from input images. This is in contrast to the reference volume construction approach which has a high risk missing the information outside the epipolar lines of the reference view. This phenomenon is more pronounced for large baselines where the scene overlap is small, as illustrated in Fig. 5.

The volume elements are sampled from the input images and features to provide necessary cues to aid the prediction of color and density for volume rendering. More specifically, to render a target image of resolution $H \times W$, we perform $8\times$ subsampling in the spatial dimension while casting rays and uniformly sample D points on each ray. The $8\times$ subsampling enables our method to maintain an acceptable volume resolution for high-resolution images, and we finally obtain the full resolution volume with a lightweight upsampler [25]. Next, we introduce the sampling process.

Color sampling within a window. Due to the $8\times$ subsampling, we might risk losing some information if we only sample a single point compared to full resolution sampling. To remedy this, we instead sample a 9×9 window centered at the 2D projection of the 3D point and thus obtain a color vector of $\{\tilde{\boldsymbol{c}}_k^{\mathrm{w}}\}_{k=1}^{K}$ for K input views, where $\tilde{\boldsymbol{c}}_k^{\mathrm{w}}$ denotes the concatenation of colors within the window.

Feature sampling and matching. We also sample image features $\{\boldsymbol{f}_i\}_{i=1}^{K}$ from K feature maps. To obtain geometric

cues, we quantify the multi-view consistency of these features by computing their pair-wise cosine similarities. We then use the cosine feature similarities to provide geometric cues for volume density prediction, based on the observation that the multi-view features of a surface point tend to exhibit high multi-view consistency [5]. All the pair-wise cosine similarities are aggregated with a learned weighted average, which can be expressed as

$$\hat{s} = w_{ij} \sum_{i<j} \cos(\boldsymbol{f}_i, \boldsymbol{f}_j), \quad i, j = 1, 2, \dots, K \quad (1)$$

where $\boldsymbol{s}_{ij}$ denotes the cosine similarity between pair $(\boldsymbol{f}_i, \boldsymbol{f}_j)$, and (i, j) iterates over all possible $K(K-1)/2$ non-repeated pairs. Following previous work [39], w_{ij} are normalized visibility weights learned from the entropy of the depth distribution on each ray.

It's worth noting that the pair-wise cosine similarities $\cos(\boldsymbol{f}_i, \boldsymbol{f}_j)$ in Eq. (1) can be computed in parallel with a single matrix multiplication. More specifically, we first normalize the features to unit vectors and then collect them as a $K \times M$ matrix, where M is the feature dimension. Finally all $K(K-1)/2$ pair-wise cosine similarities can be obtained by extracting the upper triangle of the multiplication of this matrix with its transpose. In practice, we use group-wise cosine similarities [5, 11] for further improved expressiveness, where the same computation still applies.

Multi-view aggregation. Next, the sampled colors and features from K input views are aggregated with a learned weighted average. More specifically, the aggregated color and features are

$$\hat{\boldsymbol{c}} = w_i \sum_{i=1}^{K} \tilde{\boldsymbol{c}}_i^{\mathrm{w}}, \quad \hat{\boldsymbol{f}} = w_i \sum_{i=1}^{K} \boldsymbol{f}_i, \quad (2)$$

where w_i are normalized learned weights [16]. In particular, we use a small MLP network to predict w_i from the concatenation of sampled features and view direction difference between the source and target views.

The cosine similarity $\hat{s}$ in Eq. (1), colors $\hat{\boldsymbol{c}}$ and features $\hat{\boldsymbol{f}}$ in Eq. (2) are then concatenated and projected to a C-dimensional vector with a linear layer. Accordingly we obtain a target volume $\boldsymbol{z} \in \mathbb{R}^{\frac{H}{8} \times \frac{W}{8} \times D \times C}$ for all $\frac{H}{8} \times \frac{W}{8} \times D$ 3D points. This volume encodes appearance and geometry information from multi-view images and features, which is next fed as input to the decoder for radiance field prediction.

3.3. Context-aware Radiance Field Decoder

Given the target view frustum volume $\boldsymbol{z} \in \mathbb{R}^{\frac{H}{8} \times \frac{W}{8} \times D \times C}$, our decoder learns to predict the 4-dimensional (color and density) radiance field $\boldsymbol{R} \in \mathbb{R}^{H \times W \times D \times 4}$ of all $H \times W \times D$ 3D points. In order to perform well on both small and large baseline input views, we observe that it's crucial to model the 3D context between different 3D points. The context information can help learn useful inductive biases from data

and accordingly leads to better scene structures.

To achieve this, we model the 3D context with a convolutional network. A straightforward approach would be using a 3D CNN. In this paper, we explore an alternative for better memory and parameter efficiency, while maintaining similar performance (Fig. 7). More specifically, we factorize the 3D ($3 \times 3 \times 3$) convolution to a 2D ($3 \times 3 \times 1$) convolution in the spatial dimension and a 1D ($1 \times 1 \times 3$) convolution in the depth dimension, i.e., (2+1)D CNN, which is a popular strategy in video recognition works [9, 28].

The full decoder consists of two major components: a (2+1)D CNN which operates on the low resolution volume and a lightweight upsampler which achieves $8\times$ upsampling and thus outputs a full-resolution radiance filed. The low-resolution (2+1)D CNN architecture is composed of 12 stacked (2+1)D residual blocks [12]. The final color and density predictions are obtained using two linear layers, with output channels of 3 and 1, respectively. More architectural details are presented in the supplementary material.

3.4. Hierarchical Volume Sampling

Like other NeRF methods [1, 18, 32], our model also supports hierarchical volume sampling for further improved performance. When hierarchical sampling is used, the model presented before can be viewed as a coarse model, and the hierarchical stage as a fine model. The fine model has a very similar overall architecture as the coarse stage, with the ray sampling process as the key difference. More specifically, given the density prediction from the coarse model, we first compute the Probability Distribution Function (PDF) [18] on each ray by normalizing the color composition weights in the volume rendering equation. Next, we sample a new set of points on each ray according to this distribution. Thanks to the coarse geometry predicted by the coarse network, the fine model requires sampling fewer points since the influence of empty space or occluded regions can be removed. In our implementation, we uniformly sample 64 points on each ray in the coarse model, and only sample 16 points in the fine stage. Due to the smaller number of sampling points, we are able to directly construct a target view frustum volume at the full resolution. We also reduce the volume's channel dimension from 128 to 16, which makes it more efficient for subsequent regression. Thus the volume at the fine stage is more compact. We use a lightweight 3D U-Net [22, 23] to predict the color and density from this volume, and the final color prediction is similarly obtained using volume rendering [18]. More details are presented in the supplementary material.

3.5. Training Loss

We use random crops from the full image for training, the training loss is an addition of ℓ_1, SSIM and LPIPS losses between rendered and ground truth image color.

4. Experiments

Implementation details. We implement our method using PyTorch [21] and we adopt a two-stage training process, where we first train the coarse model only, and then train the fine model with the coarse model frozen. For all the experiments, we sample 64 points in the coarse model, and 16 points in the fine model. For experiments on the RealEstate10K [40] dataset, we follow AttnRend [8]'s setting to train and evaluate on the 256×256 resolution. On this resolution, our method can easily use $4\times$ subsampling when constructing the volume, unlike the default $8\times$ subsampling. We also reduce the number of channels of the volume from 128 to 64 when $4\times$ subsampling is used. We don't include the additional hierarchical sampling stage for experiments on the RealEstate10K dataset since the single scale model already performs very competitively with $4\times$ subsampling. Our code is available at https://github.com/autonomousvision/murf.

Evaluation settings. Our evaluations include both small and large baselines, different number of views, and diverse scenarios, including object-centric, indoor and unbounded outdoor scenes, to test the model's general applicability.

4.1. Main Results

Both small & large baselines. To compare with existing methods on *both* small and large baselines, we choose previous state-of-the-art small baseline method ENeRF [16] and large baseline method AttnRend [8] as two representative approaches. The evaluations are conducted on DTU [14] and RealEstate10K [40] datasets. Since ENeRF only reported its results on DTU and AttnRend only on RealEstate10K, not both, we re-train ENeRF on RealEstate10K and AttnRend on DTU, which allows for more comprehensive comparisons. The results in Table 1 demonstrate that previous state-of-the-art methods are specialized to either small or large baselines, but they can not work well on both. In contrast, our MuRF performs consistently better on both small and large baselines. Next, we conduct comparisons in different specialized settings.

3-view small baseline on DTU. In this setting, to render a target view, 3 nearest views are selected from all source views based on the distance to the target camera location. Thus this constitutes a small baseline setting. As shown in Table 2, we achieve more than 1dB PSNR improvement compared to previous best method ENeRF [16]. The visual comparisons are shown in Fig. 3, where our method produces significantly better scene structures.

2-view large baseline on RealEstate10K. The RealEstate10K is a very large dataset, which consists of more than 66K training scenes and more than 7K testing scenes. We follow the evaluation setting of AttnRend [8] for comparisons. More specifically, the two input views are

Method	DTU (small baseline)			RealEstate10K (large baseline)		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
ENeRF [16]	27.61	0.956	0.091	19.52	0.739	0.341
AttnRend[8]	18.57	0.732	0.419	21.38	0.839	0.262
MuRF	**28.76**	**0.961**	**0.077**	**24.20**	**0.865**	**0.170**

Table 1. **Comparison on both small and large baselines**. Previous state-of-the-art methods are specialized to either small (ENeRF) or large (AttnRend) baselines, but can not work well on both.

Method	PSNR↑	SSIM↑	LPIPS↓
PixelNeRF [38]	19.31	0.789	0.382
SRF [6]	22.12	0.845	0.292
IBRNet [32]	26.04	0.917	0.190
MVSNeRF [3]	26.63	0.931	0.168
GeoNeRF [15]	26.76	0.893	0.150
MatchNeRF [5]	26.91	0.934	0.159
ENeRF [16]	27.61	0.956	0.091
MuRF	**28.76**	**0.961**	**0.077**

Table 2. **DTU 3-view small baseline**.

Method	PSNR↑	SSIM↑	LPIPS↓
PixelNeRF [38]	13.91	0.460	0.591
SRF [6]	15.40	0.486	0.604
GeoNeRF [15]	16.65	0.511	0.541
IBRNet [32]	15.99	0.484	0.532
GPNR [26]	18.55	0.748	0.459
AttnRend [8]	21.38	0.839	0.262
MuRF	**24.20**	**0.865**	**0.170**

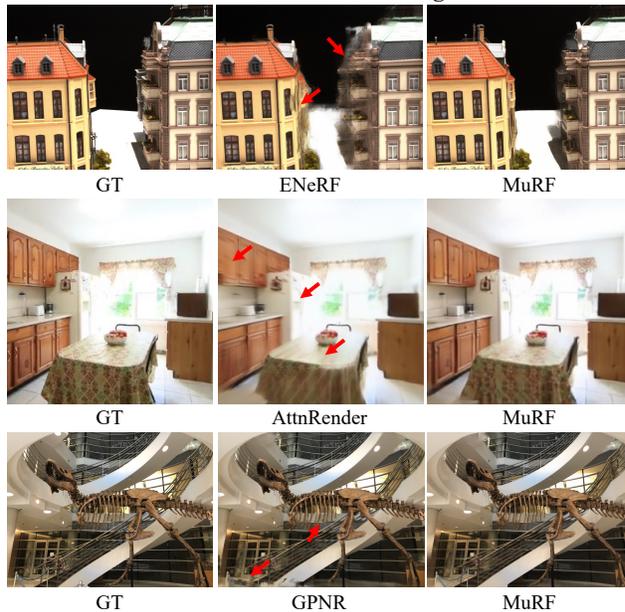Table 3. **RealEstate10K 2-view large baseline**.

Figure 3. **Visual comparisons with previous best methods** on DTU, RealEstate10K and LLFF datasets.

selected from a video with a distance of 128 frames, and the target view to synthesis is an intermediate frame. This is a challenging setting since the overlap between two input

Method	Setting	3-view			6-view			9-view		
		PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Mip-NeRF [1]		8.68	0.571	0.353	16.65	0.741	0.198	23.58	0.879	0.092
DietNeRF [13]		11.85	0.633	0.314	20.63	0.778	0.201	23.83	0.823	0.173
RegNeRF [19]	per-scene optimization	18.89	0.745	0.190	22.20	0.841	0.117	24.93	0.884	0.089
DiffusioNeRF [33]		16.20	0.698	0.207	20.34	0.818	0.139	25.18	0.883	0.095
SPARF [29]		21.01	0.870	**0.100**	-	-	-	-	-	-
SRF [6]		15.32	0.671	0.304	17.54	0.730	0.250	18.35	0.752	0.232
PixelNeRF [38]		16.82	0.695	0.270	19.11	0.745	0.232	20.40	0.768	0.220
MVSNeRF [3]	feed-forward inference	18.63	0.769	0.197	20.70	0.823	0.156	22.40	0.853	0.135
ENeRF† [16]		19.84	0.856	0.171	21.82	0.894	0.131	23.49	0.919	0.106
MuRF		**21.31**	**0.885**	0.127	**23.74**	**0.921**	**0.095**	**25.28**	**0.936**	**0.084**

Table 4. **DTU 3, 6 and 9 input views**. †Enhanced ENeRF baseline by doubling the number of sampling points on each ray, otherwise we were not able to obtain meaningful results. We again outperform ENeRF by 1∼2dB PSNR even compared to this enhanced baseline.

views is usually small, useful priors should be acquired from the model and the data in order to obtain reliable synthesis results. In this large baseline setting, we achieve significant improvement (∼3dB PSNR) than previous best method AttnRend [8], as shown in Table 3. The visual comparison in Fig. 3 indicates that our method produces clearer rendering results than AttnRend [8]. We attribute such large improvements to our target view frustum volume and the convolutional radiance field decoder, which are two key missing components in AttnRend [8]. The target view frustum volume effectively encodes the geometric structure of the view synthesis task and the convolutional decoder enables context-modeling between neighbouring 3D points, both significantly contributing to our final performance. Their effects are more thoroughly analyzed in Sec. 4.2.

Different number of input views on DTU and LLFF. We also evaluate on different number of input views to further understand the effect of different baselines. Firstly, we follow RegNeRF [19]'s setting of 3, 6 and 9 input views on DTU. Different from the aforementioned 3-view small baseline setting on DTU, the baseline of this setting is larger. More specifically, for each test scene, different numbers of views (3, 6 and 9) are sampled from a fixed set of 9 views. Each scene has 25 test views and they all share the same input views. This is different from the small baseline DTU setting before where the nearest views (instead of shared fixed views for all test views in each scene) are selected for each test view. Thus this setting is more close to the large baseline setting, especially when the number of input views is small. In this setting, our MuRF significantly outperforms previous representative methods like PixelNeRF [38] and MVSNeRF [3], as shown in Table 4.

To better understand the performance of different methods on different baselines, we re-train ENeRF [16], the previous state-of-the-art method on DTU's small baseline setting, on this large baseline setting with their official code. Since the key component of ENeRF is to use depth-guided sampling to skip empty space in NeRF's volume rendering process, its performance heavily relies on the quality of the

Method	#views	PSNR↑	SSIM↑	LPIPS↓
PixelNeRF [38]	10	18.66	0.588	0.463
MVSNeRF [3]	10	21.18	0.691	0.301
IBRNet [32]	10	25.17	0.813	0.200
NeuRay [17]	10	25.35	0.818	0.198
GPNR [26]	10	25.72	0.880	0.175
GNT [31]	10	25.53	0.885	0.218
	4	25.95	0.897	0.149
MuRF	6	26.04	0.900	0.153
	10	**26.49**	**0.909**	**0.143**

Table 5. **LLFF**. Our 4-view model already outperforms previous 10-view methods. It improves further with more views.

estimated depth. In this large baseline setting, our re-trained ENeRF is not able to produce reasonable results with their original hyper-parameters since the small overlap between input views makes the depth estimation quality lower and thus negatively affect NeRF's rendering process. Thus we train an enhanced ENeRF baseline by doubling the number of sampling points (from original 8 and 2 to 16 and 4 for 2-stage NeRFs) on each ray such that the model is more tolerant to the depth estimation error. The results are shown in Table 4. Our MuRF again outperforms ENeRF by 1∼2dB PSNR in different number of input views.

Compared with per-scene optimization methods, our MuRF achieves similar performance with the state-of-the-art 3-view method SPARF [29] in the 3-view setting. It's worth noting that SPARF specifically focuses on 3 input views and they didn't report the performance on 6 and 9 views. In contrast, we purse thorough evaluation on different number of views and we show clear improvement than previous per-scene optimization methods with 6 and 9 input views, despite without using any per-scene optimization.

We also compare with previous methods on the LLFF dataset [18]. Since this is a forward-facing dataset, the baselines between different images are typically small. Previous methods usually evaluate on this dataset with 10 input views following IBRNet [32]'s setting. Surprisingly, our MuRF trained with 4 input views already outperforms all previous 10-view methods (Table 5), suggesting our model is also more data-efficient. Our performance gets further improve-

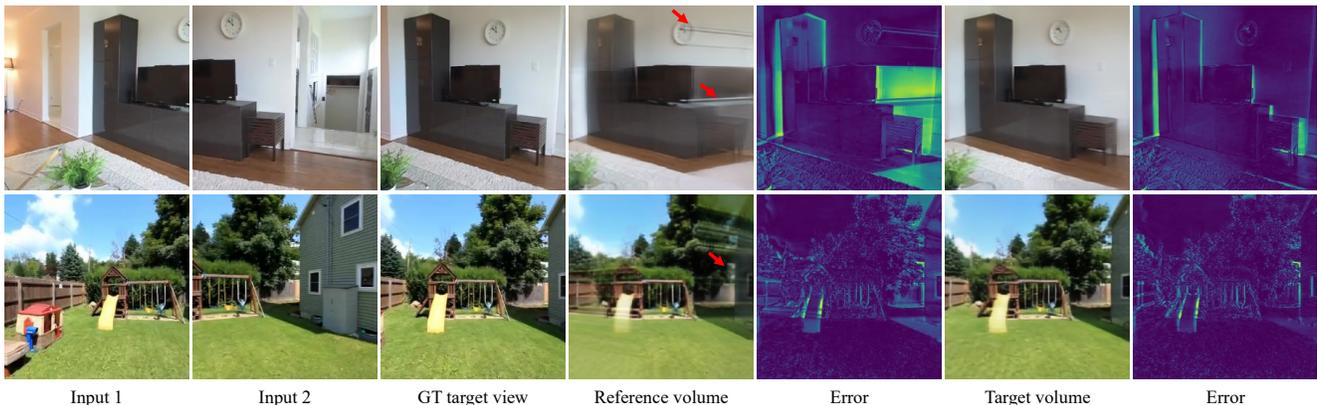Figure 4. **Zero-shot generalization on Mip-NeRF 360 dataset**.

| Input 1 | Input 2 | GT target view | Reference volume | Error | Target volume | Error |

Figure 5. **Our target view volume *vs*. reference (first image) view volume**. Constructing the volume at the pre-defined reference view space might miss relevant information (*e.g.*, red arrows regions) in other views since such information could be far away from the the reference view's epipolar lines and thus hard to be sampled. In contrast, we construct the volume in the target view, which more effectively aggregates information from all input views and thus maximizes the information usage.

Method	DTU			Mip-NeRF 360 Dataset		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
AttnRend [8]	11.35	0.567	0.651	14.00	0.474	0.712
MuRF	**22.19**	**0.894**	**0.211**	**23.98**	**0.800**	**0.293**

Table 6. **Zero-shot generalization after trained on RealEstate10K**. Our method outperforms AttnRend [8] by significantly large margins.

ment with more input views. The visual comparison with previous state-of-the-art method GPNR [26] in Fig. 3 shows that our rendering has better structures.

Generalization on DTU and Mip-NeRF 360 dataset. For feed-forward models, we are also interested in their zero-shot generalization abilities on unseen datasets, which is a practical problem setting. For this evaluation, we compare with AttnRend [8] using their released model trained on the same RealEstate10K dataset. The results are reported on the object-centric dataset DTU and scene-level Mip-NeRF 360 dataset with 2 input views. From Table 6, we observe our MuRF generalizes significantly better than AttnRend. The lack of geometric inductive biases makes AttnRend more data-hungry and prone to overfitting to the training data.

To further explore the zero-shot performance limit on Mip-NeRF 360 dataset, we conduct additional fine-tuning with our RealEstate10K pre-trained model on a mixed dataset collected by IBRNet [32]. We achieve further improvement and the promising visual results shown in Fig. 4 indicate the general applicability of our method.

4.2. Ablation and Analysis

For fast ablation experiments, we only train coarse models on both DTU and RealEstate10K datasets. Since the full RealEstate10K dataset is very large, which would take significant compute to train all the ablations, we use the subset provided by AttnRend [8] for training and evaluation.

Volume Orientation. One key difference of our method with previous approaches is that we construct the volume in the *target view frustum*, unlike the popular reference view-based volume in multi-view stereo related methods [3, 15]. Such a volume representation essentially aggregates information from multi-view input images, where the reference view volume would suffer from information loss since the regions outside its epipolar lines are less likely to be sampled. This issue would be more pronounced for large base-

Module	Method	Large Baseline (RealEstate10K)			Small Baseline (DTU)		
		PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Volume Orientation	target view	**21.29**	**0.808**	**0.246**	**26.89**	**0.925**	**0.119**
	reference view	20.20	0.776	0.309	26.50	0.922	0.138
Volume Elements	cosine & feature & color	**21.29**	**0.808**	**0.246**	**26.89**	**0.925**	**0.119**
	w/o cosine	21.06	0.800	0.255	26.68	0.919	0.124
	w/o feature	20.91	0.794	0.275	26.79	0.923	0.132
	w/o color	20.04	0.789	0.275	25.80	0.916	0.133
Radiance Decoder	(2+1)D CNN (spatial + depth)	21.29	0.808	0.246	**26.89**	**0.925**	**0.119**
	3D CNN (spatial & depth)	**21.39**	**0.809**	**0.243**	26.86	0.924	0.121
	2D CNN (spatial only)	21.06	0.801	0.256	26.67	0.921	0.124
	1D CNN (depth only)	20.78	0.788	0.272	26.37	0.920	0.124
	Ray Transformer (depth only)	20.57	0.780	0.283	26.22	0.912	0.129
	MLP (point only)	20.41	0.778	0.298	26.24	0.918	0.126

Table 7. **Ablations**. Settings used in our final model are underlined. The 3D CNN model has more than 50% parameters (16.6M *vs*. 10.4M) than our factorized (2+1)D CNN model, and the performance is similar, thus we choose to use the (2+1)D CNN.

Figure 6. **CNN *vs*. MLP *vs*. Ray Transformer**. Our CNN-based decoder yields sharper structures thanks to its context modeling abilities in both spatial and depth dimensions.

lines, as illustrated in Fig. 5. In Table 7, we can observe a 1dB PSNR performance drop by replacing our target view volume with the reference view volume on the large baseline RealEstate10K dataset. Even on the small baseline DTU dataset, our target view volume is still better since it maximizes the information usage from input views.

Volume Elements. We construct the volume using information from the sampled colors, features, and features' cosine similarities, they are concatenated together as the volume elements. We ablate different inputs in Table 7 to understand their roles. It's interesting to see that for the large baseline setting, the features are more important than the cosine similarities, while cosine similarities become more important for small baselines. This could be expected since the feature matching information would be less reliable for large baselines due to insufficient scene overlap. It's also worth noting that for both small and large baselines, the sampled colors also contribute to the final performance. The image color provides valuable cues for predicting the 3D points' color, thus leading to improved performance.

Radiance Decoder. We compare different approaches to predict the radiance field from the target view frustum vol-

ume in Table 7. It can be observed that the spatial context modeled by the 2D CNN is more important to the final performance than that on the depth dimension. Previous methods like IBRNet [32] and GNT [27] use Ray Transformer to only model the context information on the depth dimension. However, we observe in our experiments that the Ray Transformer not always produces better results than MLP (see Fig. 6), while our (2+1)D CNN is consistently better. We also compare with the straightforward 3D CNN decoder and the performance is similar. However, 3D CNN has more than 50% more parameters than our (2+1)D CNN, and thus we choose to use the (2+1)D CNN for better efficiency. The visual comparison in Fig. 6 illustrates that our CNN-based decoder produces clearly better structures.

5. Conclusion

We present MuRF, a feed-forward method to sparse view synthesis from multiple different baseline settings. Key to our approach are the proposed target view frustum volume and CNN-based radiance field decoder. We achieve state-of-the-art performance on various evaluation settings, demonstrating the general applicability of our method.

Limitation and Discussion. Currently, large-scale scene-level datasets are still scarce, we expect more diverse training data would improve our performance further. Besides, our model currently assumes known camera parameters and static scenes, extending our method to pose-free scenarios and dynamic scenes could be interesting future directions.

References

[1] Jonathan T Barron, 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 *ICCV*, 2021. 1, 4, 6

[2] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *CVPR*, 2022. 1, 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. In *ICCV*, 2021. 1, 2, 3, 5, 6, 7

[4] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *ECCV*, 2022. 1

[5] Yuedong Chen, Haofei Xu, Qianyi Wu, Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. Explicit correspondence matching for generalizable neural radiance fields. In *arXiv*, 2023. 1, 2, 3, 4, 5

[6] Julian Chibane, Aayush Bansal, Verica Lazova, and Gerard Pons-Moll. Stereo radiance fields (srf): Learning view synthesis for sparse views of novel scenes. In *CVPR*, 2021. 1, 2, 5, 6

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1

[8] Yilun Du, Cameron Smith, Ayush Tewari, and Vincent Sitzmann. Learning to render novel views from wide-baseline stereo pairs. In *CVPR*, 2023. 1, 2, 5, 6, 7

[9] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *ICCV*, 2019. 2, 4

[10] David Gallup, Jan-Michael Frahm, Philippos Mordohai, and Marc Pollefeys. Variable baseline/resolution stereo. In *CVPR*, 2008. 2

[11] Xiaoyang Guo, Kai Yang, Wukui Yang, Xiaogang Wang, and Hongsheng Li. Group-wise correlation stereo network. In *CVPR*, 2019. 4

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3, 4

[13] Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. In *ICCV*, 2021. 6

[14] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engin Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *CVPR*, 2014. 2, 5

[15] Mohammad Mahdi Johari, Yann Lepoittevin, and François Fleuret. Geonerf: Generalizing nerf with geometry priors. In *CVPR*, 2022. 1, 2, 3, 5, 7

[16] Haotong Lin, 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*, pages 1–9, 2022. 1, 2, 4, 5, 6

[17] Yuan Liu, Sida Peng, Lingjie Liu, Qianqian Wang, Peng Wang, Christian Theobalt, Xiaowei Zhou, and Wenping Wang. Neural rays for occlusion-aware image-based rendering. In *CVPR*, 2022. 1, 2, 3, 6

[18] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 1, 2, 3, 4, 6

[19] Michael Niemeyer, Jonathan T Barron, Ben Mildenhall, Mehdi SM Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *CVPR*, 2022. 1, 2, 6

[20] Masatoshi Okutomi and Takeo Kanade. A multiple-baseline stereo. *TPAMI*, 15(4):353–363, 1993. 2

[21] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *NeurIPS*, 2019. 5

[22] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 4

[23] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 4

[24] Mehdi SM Sajjadi, Henning Meyer, Etienne Pot, Urs Bergmann, Klaus Greff, Noha Radwan, Suhani Vora, Mario Lučić, Daniel Duckworth, Alexey Dosovitskiy, et al. Scene representation transformer: Geometry-free novel view synthesis through set-latent scene representations. In *CVPR*, 2022. 1, 2

[25] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *CVPR*, 2016. 2, 3

[26] Mohammed Suhail, Carlos Esteves, Leonid Sigal, and Ameesh Makadia. Generalizable patch-based neural rendering. In *ECCV*. Springer, 2022. 2, 5, 6, 7

[27] Mukund Varma T, Peihao Wang, Xuxi Chen, Tianlong Chen, Subhashini Venugopalan, and Zhangyang Wang. Is attention all that neRF needs? In *ICLR*, 2023. 2, 8

[28] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, 2018. 2, 4

[29] Prune Truong, Marie-Julie Rakotosaona, Fabian Manhardt, and Federico Tombari. Sparf: Neural radiance fields from sparse and noisy poses. In *CVPR*, 2023. 1, 2, 6

[30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NIPS*, 2017. 1

[31] Peihao Wang, Xuxi Chen, Tianlong Chen, Subhashini Venugopalan, Zhangyang Wang, et al. Is attention all nerf needs? *arXiv preprint arXiv:2207.13298*, 2022. 6

[32] 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 *CVPR*, 2021. 1, 2, 4, 5, 6, 7, 8

[33] Jamie Wynn and Daniyar Turmukhambetov. Diffusionerf: Regularizing neural radiance fields with denoising diffusion models. In *CVPR*, 2023. 1, 2, 6

[34] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, and Dacheng Tao. Gmflow: Learning optical flow via global matching. In *CVPR*, 2022. 3

[35] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, Fisher Yu, Dacheng Tao, and Andreas Geiger. Unifying flow, stereo and depth estimation. *PAMI*, 2023. 3

[36] Ruigang Yang and Marc Pollefeys. Multi-resolution real-time stereo on commodity graphics hardware. In *CVPR*, 2003. 2

[37] Ruigang Yang, Greg Welch, and Gary Bishop. Real-time consensus-based scene reconstruction using commodity graphics hardware. In *10th Pacific Conference on Computer Graphics and Applications*, 2002. 2

[38] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *CVPR*, 2021. 1, 5, 6

[39] Jingyang Zhang, Yao Yao, Shiwei Li, Zixin Luo, and Tian Fang. Visibility-aware multi-view stereo network. *arXiv preprint arXiv:2008.07928*, 2020. 4

[40] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: learning view synthesis using multiplane images. *ACM Trans. Graph.*, 37(4): 65, 2018. 2, 5