Out of Sight, Out of Mind: 
A Source-View-Wise Feature Aggregation for Multi-View Image-Based Rendering

Geonho Cha¹, Chaehun Shin², Sungroh Yoon²,³, and Dongyoon Wee¹

¹ Clova AI, NAVER Corp., Korea {geonho.cha,dongyoon.wee}@navercorp.com
² Department of Electrical and Computer Engineering, Seoul National University, Korea {chaehun,yoon}@snu.ac.kr
³ Interdisciplinary Program in Artificial Intelligence, Seoul National University, Korea

Abstract. To estimate the volume density and color of a 3D point in the multi-view image-based rendering, a common approach is to inspect the consensus existence among the given source image features, which is one of the informative cues for the estimation procedure. To this end, most of the previous methods utilize equally-weighted aggregation features. However, this could make it hard to check the consensus existence when some outliers, which frequently occur by occlusions, are included in the source image feature set. In this paper, we propose a novel source-view-wise feature aggregation method, which facilitates us to find out the consensus in a robust way by leveraging local structures in the feature set. We first calculate the source-view-wise distance distribution for each source feature for the proposed aggregation. After that, the distance distribution is converted to several similarity distributions with the proposed learnable similarity mapping functions. Finally, for each element in the feature set, the aggregation features are extracted by calculating the weighted means and variances, where the weights are derived from the similarity distributions. In experiments, we validate the proposed method on various benchmark datasets, including synthetic and real image scenes. The experimental results demonstrate that incorporating the proposed features improves the performance by a large margin, resulting in the state-of-the-art performance.

Keywords: multi-view image-based rendering, feature aggregation, neural radiance fields, novel view synthesis, 3D deep learning

1 Introduction

Novel view synthesis is one of the core problems in computer vision, which has various applications like augmented reality and human-computer interaction. Various methods based on explicit scene representations ⁴,¹¹,¹²,¹³,¹⁴,¹⁵,¹⁶ or implicit scene representations ¹⁸,²⁶,³²,³³,³⁶,¹⁶ have been proposed to handle this problem. Recently, neural radiance fields (NeRF) ²⁵, which is based on
Fig. 1. An illustration of the proposed method’s intuition. Let us consider a rendering process of the target image pixel indicated by the green arrow, given the corresponding source image features indicated by ‘x’ marks. The baseline methods [2, 44, 48] utilize a single aggregation range including outlier features so that outlier features influence to get mean and variance of features. Meanwhile, the proposed method uses multiple source-view-wise aggregation ranges, which can suppress the influence of the outlier features. This enables us to figure out the consensus existence in a robust way. Here, the learnable parameter $\lambda$ regulates the aggregation range.

To mitigate the expensive optimization cost, multi-view image-based rendering methods [2, 40, 44, 48] have been introduced to exploit 2D features from source images of the scene during the rendering. They utilize aggregated 2D features to predict the densities and colors of corresponding sampled 3D points. Thus, how to aggregate this feature set is crucial for precise density and color prediction for these methods. To aggregate 2D features, previous multi-view image-based rendering methods typically utilize element-wise global mean and variance, based on the observation that 3D points on surfaces are more likely to have consensus in the 2D feature set. Accordingly, a network learns to reason density by comparing features with global mean based on variance.

A thing to note here is that they give the same weight to each 2D feature when calculating the aggregation features. However, the equally-weighted aggregation features could not be informative in situations which some outlier features, which could frequently occur with occlusions, are in the feature set. For example, in Figure 1 even though the 3D point is on a surface, features extracted from the occluded region of the bucket would be different from the features extracted from the region of the ceiling. In this case, the equally-weighted aggregation features are not suitable for finding a consensus in the feature set.
In this paper, to resolve this issue, we propose a novel source-view-wise feature aggregation method, which facilitates us to find out the consensus existence in a robust way utilizing the local structure of the feature set. Unlike the previous methods that use common equally-weighted aggregation features, our approach incorporates multiple aggregation features which are aggregated based on source-view-wise weights. For the proposed aggregation, we first calculate the source-view-wise distance distribution for each element in the feature set. Thereafter, the distance distribution is converted to similarity distributions through the proposed learnable similarity measure functions. Here, the learnable similarity measure functions are trained in an end-to-end manner with other network parameters to be tailored to the 2D features. Finally, we extract the aggregation features by calculating the weighted element-wise means and variances, where the weights are determined based on the similarity distributions. These features are utilized along with the source image features to infer the volume densities and colors of 3D points in the volume rendering process.

In experiments, we validate the proposed method on several benchmark datasets consisting of synthetic and real scenes. The experimental results show that incorporating the proposed feature set improves the performance by a large margin, which demonstrates the effectiveness of the proposed source-view-wise feature aggregation method.

The contributions of our method can be summarized as:

- We propose a novel source-view-wise feature aggregation method which facilitates us to utilize the consensus existence in a robust way by leveraging the local structures in the feature set.
- We propose novel learnable similarity measure functions for the proposed feature extraction.
- The proposed scheme shows the state-of-the-art performance on the diverse benchmark datasets.

2 Related work

The novel view synthesis problem has been actively dealt with as it has various applications. The methods handling the problem can be categorized into two types based on how they model 3D spaces. The first type models the 3D space with explicit scene representations \cite{11,12,15,21,24,29,35,37,41,42,50}, and the second one models the 3D space based on implicit scene representations \cite{18,26,32,36,46}.

The explicit-representation-based methods utilize voxel grids \cite{11,12,21,29,35,42} or multi-plane images \cite{4,5,15,24,37,41,50} to perform the novel view image synthesis process. However, these methods have some drawbacks: (i) it is hard to infer high-resolution images as the required memory size increases drastically with the increased resolutions because of the explicitly inferred 3D spaces, and (ii) some artifacts could occur comes from the discretized nature.

To handle these issues, implicit-representation-based 3D object modeling methods have been proposed \cite{6,9,23,27}. They modeled the 3D objects based
on signed distance functions [9, 27] or occupancy functions [6, 23]. Incorporating these methods, implicit-representation-based novel view synthesis methods have been proposed [18, 19, 20, 26, 32, 36, 46]. Recently, neural radiance fields [25] and its following work [1, 8, 14, 16, 17, 20, 22, 31, 39, 47] have been proposed. The NeRF utilizes implicit neural radiance fields, and shows solid performance on the novel view synthesis task. They used the classical differentiable volume rendering [10] to synthesize the target image, of which the volume densities and colors are estimated from a multi-layer perceptron (MLP) network. To alleviate the issue that the MLP tends to focus on learning low-frequency information, they utilized a positional encoding technique [30]. Although its satisfactory performance, the need for per-scene optimization was a burden for their practical uses.

To relieve this condition, multi-view image-based rendering methods [2, 3, 40, 44, 48] have been proposed. To render a novel view image of a scene, they utilize source image features of the same scene. Here, figuring out the consensus existence is a key for the on-the-fly estimation of volume densities and colors. To this end, most of the multi-view image-based rendering methods incorporated equally-weighted aggregation features. PixelNeRF [48] proposed an image feature conditioned MLP to handle the generalizable novel view synthesis problem. They aggregated the multi-view MLP outputs by an average pooling, which is fed into another MLP to estimate the volume density and color. MVSNeRF [2] utilized plane-swept cost volumes for the generalized novel view synthesis. The global variance of the source view image features was utilized to establish the cost volumes. IBRNet [44] proposed a MLP-based network with a ray transformer. They utilized globally aggregated features of element-wise mean and variance. On the other hand, GRF [40] utilized a geometry-aware attention module for the feature aggregation, which could be effective for the occlusion handling. However, they did not integrate the local features and the aggregation feature. It could be ineffective to infer surfaces which leads to inferior performance in occlusion problems.

Note here that most of the previous methods utilized equally-weighted aggregation features. However, in this way, it could be hard to discover the surface when some outliers, which could frequently occur by occlusions, are included in the source image feature set. In this paper, to handle this issue, we propose a novel source-view-wise feature aggregation method, which facilitates us to inspect the consensus existence in a robust way by incorporating the local structure in the feature set.

3 Multi-view image-based rendering

We handle the problem of synthesizing the target image $I_t$ in the target camera view $p_t$ given source images $I_s$ along with their corresponding camera parameters $\{I_s^i, K_s^i, p_s^i | i = 1, \ldots, n_s\}$, where $n_s$ is the number of the source images, and $K_s^i$ and $p_s^i$ are the intrinsic and extrinsic camera parameters of the $i$-th source image, respectively.
For the target image rendering, we utilize the differentiable volume rendering scheme \[10\]. First, we shoot ray \( r(t) = o + td \) for each pixel of the target image, where \( o, d, \) and \( t \) are the target camera center, target viewing direction, and depth, respectively. After that, we sample points \( \{ t_i | i = 1, \cdots, n_t \} \) on the ray, where the sampling range is bounded by the given near bound \( t_n \) and the far bound \( t_f \). Here, \( n_t \) is the number of sampled points, and \( t_n \leq t_v \) if \( u \leq v \). The RGB color \( C \) of the pixel is rendered based on the following equation:

\[
C(r) = \sum_{i=1}^{n_t} T_i (1 - \exp(-\sigma(r(t_i))\delta_i)) c(r(t_i), d),
\]

where \( T_i = \exp(-\sum_{j=1}^{i-1} \sigma(r(t_j))\delta_j) \), and \( \delta_i = t_{i+1} - t_i \). Here, \( \sigma(r(t_i)) \) is the volume density of the 3D point \( r(t_i) \), and \( c(r(t_i), d) \) is the RGB color of the 3D point \( r(t_i) \) conditioned on the viewing direction \( d \). Note that for the volume rendering, we need to design the volume density estimator \( \sigma(\cdot) \) and the color estimator \( c(\cdot) \).

In the multi-view image-based rendering approach, the features of the given source images are incorporated to infer the volume densities and colors of sampled 3D points. Let us consider the problem of estimating the volume density and the color of a 3D point \( x \). We first project \( x \) onto each source image as

\[
u^i = \Pi^i x,
\]

where \( \Pi^i \) is the projection matrix of the \( i \)-th source view image, and \( u^i \) is the projected pixel location. We here assume homogeneous coordinates. Based on the projected pixel locations, we compose the feature set \( f = \{ f_i \in \mathbb{R}^{n_f} | i = 1, \cdots, n_s \} \), which consists of the extracted source image features. Here, \( n_f \) is the feature dimension. In this process, we get the bilinearly interpolated features \( f_i \) since the projected pixel may not have integer coordinates. The feature set \( f \) is utilized for the volume density and color estimation.

Most of the multi-view image-based rendering approaches design the estimation process after being inspired by the classical multi-view stereo scheme \[33,38\]. If a 3D point is on a surface, which is the case that the 3D point has a significant impact on the rendered image, there is a consensus in the feature set \( f \). The network is designed to fulfill that the features in the consensus contribute a lot when inferring the volume density and the color of the 3D point. To achieve this, most of the previous methods utilize equally-weighted aggregation features \[2,44,48\]. A typical way for the aggregation is to calculate the statistical information like element-wise mean and variance of the feature set.

However, in this way, it could be hard to identify the consensus due to some outlier features which can frequently arise from occlusions. To resolve this issue, we propose a novel source-view-wise feature aggregation method, which is introduced in the next chapter.
Fig. 2. An overview of the proposed source-view-wise feature aggregation method. We first calculate the distance distribution for each element in the feature set for the feature aggregation. After that, the distance distributions are converted to similarity distributions with the proposed learnable similarity measure functions. Finally, the aggregated features are extracted by calculating weighted element-wise means and variances. Here, $n_s$, $n_k$, and $n_f$ are the number of source images, similarity measure functions, and the feature dimensions, respectively.

Fig. 3. A graphical illustration of $h(\cdot)$ with respect to various values of $\lambda$.

4 Source-view-wise feature aggregation

An overview of the proposed feature aggregation method is visualized in Figure 2. Given the source image feature set $f$, the proposed method aggregates the features in a source-view-wise way to obtain aggregation features for each source feature. We first compute the distance distribution for the $i$-th source feature as

$$d_i = \{d_{i,j} | j = 1, \ldots, n_s\}, \quad d_{i,j} = \|f_i - f_j\|_2, \quad (3)$$

where $\|\cdot\|_2$ denotes the $l_2$-norm. After that, the distance distribution is converted to the similarity distributions based on the proposed learnable similarity measure functions.

4.1 Learnable similarity measure functions

The distance distribution could be converted to a similarity distribution with a predefined function. However, the predefined function might be a sub-optimal solution as they are not designed utilizing the extracted image features. Moreover,
it might be hard to extend the predefined function to a family of the functions, which might be needed to model the complex feature distributions. To mitigate these issues, we propose learnable similarity measure functions. Note that ablation studies with respect to the choice of the similarity measure functions can be found in the experiments section.

The goal of the similarity measure functions is converting the distance distribution \( d_{i,j} \) to a set of similarity distributions \( s_{i} = \{s^k \mid k = 1, \cdots, n_k\} \) where \( n_k \) is the number of similarity measure functions, and \( s^k = \{s^k_{i,j} \mid j = 1, \cdots, n_s\} \). Here, \( s^k_{i,j} \) is the similarity derived from \( d_{i,j} \) based on the \( k \)-th similarity measure function. Each similarity measure function should meet the following conditions: (i) The function should output larger similarity values in case of the smaller distances, and (ii) it should be easy to be trained, i.e., it has smooth gradient functions. With the criteria in consideration, we propose the following functions as the similarity measure:

\[
h_k(d_{i,j}) = s^k_{i,j} = e^{-\lambda_k d_{i,j}}, \tag{4}\]

where \( \lambda_k \) is a learnable scalar parameter. The learnable parameter \( \lambda_k \) is trained with the other network weights in an end-to-end manner. Here, \( \lambda_k \) regulates the aggregation range, i.e., how much the farther features are ignored. Some illustrations of the similarity measure function with respect to various values of \( \lambda \) are visualized in Figure 3. A thing to note here is that \( \lambda_k \) should have a positive value to meet the first condition of the similarity measure function. To achieve this, we parameterize \( \lambda_k \) as \( e^{\alpha_k} \) where \( \alpha_k \) is a learnable scalar parameter.

With the proposed method, unlike a predefined function, the similarity measure function can be designed more tailored to the extracted image features. Furthermore, we can utilize several similarity measure functions by increasing \( n_k \). An ablation study with varying \( n_k \) can be found in the experiments section.

### 4.2 Weighted element-wise means and variances

We aggregate the features based on the weighted statistical distributions of element-wise means and variances. For the \( k \)-th similarity distribution, the weight is derived from the measured similarity as

\[
w^k_{i,j} = s^k_{i,j} / \sum_{j} s^k_{i,j}. \tag{5}\]

Thereafter, the weighted element-wise mean \( m^k_i \) and variance \( v^k_i \) are calculated as

\[
m^k_i[l] = \sum_{j} w^k_{i,j} f^k_j[l], \quad v^k_i[l] = \sum_{j} w^k_{i,j} (f^k_j[l] - m^k_i[l])^2, \tag{6}\]

where \( a[l] \) is the \( l \)-th element of \( a \).

The aggregated feature set \( \{m^k_i, v^k_i \mid k = 1, \cdots, n_k\} \) is concatenated with the \( i \)-th source image feature \( f_i \) to be utilized as an input for a neural network which estimates the volume densities and colors of 3D points.
5 Proposed framework

5.1 Network design

An overview of the proposed network structure incorporating the source-view-wise aggregated features is visualized in Figure 4. We design the network based on the framework of [44] as it shows the state-of-the-art performance. A thing to note here is that the proposed feature aggregation method might be utilized with other multi-view image-based rendering methods which utilize global aggregation features [2,48].

For the image feature extraction network, we use the same network structure used in [44] with the output feature dimension is 16. After the source images are encoded with the feature extraction network, the features and RGB values at the projected pixels are fed into the network along with the relative viewing direction features. Here, the relative viewing direction features model the differences between the target camera viewpoint and the source camera viewpoints. The relative viewing direction features are fed into a MLP to match the feature dimension of the features with that of the image features. The dimension-matched relative viewing direction features are added to the image features, and the features are fed into the source-view-wise feature aggregator introduced in Section 4. The feature aggregator outputs aggregated feature set, which is concatenated with the input feature of the aggregator. The concatenated features are then fed into a MLP, and the output of the MLP is fed into the volume density and color estimator along with the RGB values and relative viewing direction features. The structure of the volume density and color estimator is the same as the lat-
ter part of the IBRNet structure [44]. For more detailed structures, please refer to [44].

5.2 Loss function

Like previous volume-rendering-based methods [25, 44, 48], the whole network is trained in an end-to-end manner leveraging the differentiable volume rendering scheme. The rendering loss is defined as

\[ L_{\text{render}} = \sum_{r \in R} \| \hat{C}(r) - C(r) \|^2_F, \tag{7} \]

where \( \hat{C}(r) \) and \( C(r) \) are the estimated and the ground truth RGB values, respectively. \( R \) is the sampled ray batch in a training iteration.

5.3 Implementation details

We incorporate all the techniques proposed in [44] like neighboring view selection scheme, ray transformer, and temporal visual consistency improvement scheme. For the effective point sampling on a ray, we adopt the hierarchical sampling scheme following the general practices in [25, 44]. Hence, we train two networks simultaneously for the coarse and fine point samplings. In the coarse point sampling, we uniformly sample \( n_t = 64 \) points. After that, in the fine point sampling, we conduct importance sampling based on the estimated volume densities from the coarse network. As a result, we additionally sample 64 points, resulting in \( n_t = 128 \) for the fine network. To build our framework, we use PyTorch [28]. For optimization, we use Adam [13] optimizer. The initial learning rate is set to \( 1 \times 10^{-3} \) for the feature extraction network, and \( 5 \times 10^{-4} \) for the other part of the network. The learning rate is decreased by a factor of 0.5 for every 50k iteration. The whole training ends at 250k iterations. The network is trained on four V100 GPUs with a batch size of 2000 rays, which takes about a day to finish. For the network configuration, we use \( n_f = 19, n_k = 5, n_s = 10 \), unless stated otherwise. For the finetuning, we finetune the whole network with a lower learning rate of \( 5 \times 10^{-4} \) and \( 2 \times 10^{-4} \) for the feature extraction network and the other part of the network, respectively. Here, the learning rate is decreased by a factor of 0.5 for every 20k iteration. The network is finetuned on two V100 GPUs with a batch size of 1000 rays. The finetuning process ends at 60k iterations, which takes about six hours to finish.

5.4 Comparison with other methods

Comparison with GRF. Though GRF [40] follows the same motivation which handles occlusion problems, its approach is different from our method. While GRF uses attention to aggregate source 2D features into globally aggregated 3D point feature, our method uses MLP to integrate both local and global information to infer the volume density and color of a 3D point. Because GRF does
not integrate local and global information, it can be ineffective to infer surfaces which leads to inferior performance in occlusion problems. A quantitative comparison result can be found in the experiments section.

**Comparison with self-attention.** The proposed method may seem similar to the self-attention mechanism [43] in that the element-wise similarities are utilized. However, the proposed feature aggregation method is different from the self-attention scheme in two ways: (i) We incorporate $l_2$-norm as the distance metric, which is commonly used in the multi-view image-based rendering approach. This might enable us to use the proposed method as a universal technique. (ii) The number of additional parameters to utilize the self-similarities is much fewer than that of the self-attention mechanism. For example, in GRF, AttSets needs about 1000k additional parameters, and Slot Attention needs about 300k additional parameters. However, the proposed method just needs $n_k = 5$ additional parameters.

6 Experiments

In this section, we show experimental results on two evaluation protocols. On the first protocol of “pretrained”, the model is trained on training scenes and tested on unobserved scenes without any further optimization. On the second protocol of “finetuned”, the model trained with the first protocol is finetuned using the training images of given test scenes before rendering the novel view images of the test scenes.

6.1 Dataset

For training our model based on the “pretrained” protocol, we incorporate five public datasets: object-centrically rendered scenes using a subset of the Google Scanned Objects dataset [7] which is provided by [44], the RealEstate10K dataset [50], the Spaces dataset [4], the LLFF dataset [24], and the IBRNet collected dataset [44]. The training datasets consist of various synthetic and real scenes, which facilitates us to train the generalizable model.

To evaluate the proposed model, we utilize three datasets: the Diffuse Synthetic 360$^\circ$ dataset [35], the Realistic Synthetic 360$^\circ$ dataset [25], and the Forward-Facing [24] dataset. The test datasets consist of real scene [24] and synthetic scenes [25, 35], which allows us to evaluate the proposed method in various situations. Following the practice in [44], we evaluate the proposed model on the sub-sampled images with a factor of 10, 8, and 8 for the Diffuse Synthetic 360$^\circ$ dataset, the Realistic Synthetic 360$^\circ$ dataset, and the Forward-Facing dataset, respectively. For all the datasets, we utilize the camera parameters and the near and far bounds estimated using COLMAP [34].

6.2 Quantitative results

In this section, we show the quantitative comparison results. For the evaluation metric, we incorporate peak signal to noise ratio (PSNR), structural similar-
Table 1. Performance comparison results on the several benchmark datasets including real and synthetic scenes. For the protocol, “P” means the “pretrained” protocol, and “F” means the “finetuned” protocol. The performances of the other methods are quoted from [44]. In each category, the best results are in **bold**, and the second-best are underlined.

| Method | Protocol | Diffuse Synthetic 360° | Realistic Synthetic 360° | Real Forward-Facing | PSNR | SSIM | LPIPS | PSNR | SSIM | LPIPS |
|--------|----------|------------------------|--------------------------|--------------------|------|------|-------|------|------|-------|
|        |          | [35]                   | [25]                     | [24]               |      |      |       |      |      |       |
| LLFF   | P        | 34.38 0.985 0.048       | 24.88 0.911 0.114        | 24.13 0.798        | 0.212|
| IBRNet | P        | 37.17 0.990 0.017       | 25.49 0.916 0.100        | 25.13 0.817        | 0.205|
| Ours   | P        | 37.45 0.990 0.017       | 27.13 0.930 0.083        | 25.47 0.826        | 0.190|
| SRN    | F        | 33.20 0.983 0.001       | 22.26 0.816 0.119        | 22.84 0.668        | 0.378|
| NV     | F        | 29.62 0.929 0.059       | 26.05 0.893 0.160        | -                 | -    |
| NeRF   | F        | 40.15 0.991 0.023       | 31.01 0.947 0.081        | 26.50 0.811        | 0.250|
| IBRNet | F        | 42.93 0.997 0.009       | 28.14 0.942 0.072        | 26.73 0.851        | 0.175|
| Ours   | F        | 43.21 0.997 0.008       | 30.89 0.956 0.051        | 26.66 0.849        | 0.175|

PSNR, SSIM, and LPIPS are peak signal-to-noise ratio, structural similarity index measure (SSIM) [45], and learned perceptual image patch similarity (LPIPS) [49], following the practices in [44]. We first compare the performance of the proposed method based on the “pretrained” protocol with state-of-the-art methods of LLFF [24] and IBRNet [44]. The comparison results are summarized in the top part of Table 1. From the results, we can demonstrate that the proposed method shows the state-of-the-art performance in most cases. On the Diffuse Synthetic 360° dataset, the proposed method shows comparable performances with IBRNet on the SSIM and LPIPS measures. We conjecture that the proposed method has little room for improvement on these cases as the Diffuse Synthetic 360° dataset consists of simple scenes. However, we note that the proposed method improves the PSNR measure by a large margin. In other cases, which are relatively hard, we can see that the proposed method shows the state-of-the-art performances.

We also report the performance of the proposed method based on the “finetuned” protocol. We compare the performance of the proposed method with various methods of SRN [36], NV [21], NeRF [25], and IBRNet [44]. The comparison results are summarized in the bottom part of Table 1. The results demonstrate that the proposed method shows better or competitive performances compared with one of the state-of-the-art methods, IBRNet.

In addition, we report the comparison result with NeRF and GRF [40] in Table 2 based on the both of the protocols. For the fair comparison, we have followed the practices of [40]. Specifically, the network is trained on Chair, Mic, Ship, and Hotdog scenes of the Realistic Synthetic 360° dataset, and tested on Drums, Lego, Materials, and Ficus scenes of the Realistic Synthetic 360° dataset. In this case, we use four neighboring source view images, i.e., $n_s=4$. From the results, we can see that the proposed method shows much better performance than GRF.

### 6.3 Qualitative results

We provide qualitative comparison results on the “pretrained” protocol with IBRNet [44] which is one of the state-of-the-art methods. Note that the qualita-
Table 2. Performance comparison results on the Realistic Synthetic 360° dataset. The performance of the other methods are quoted from [40]. In each category, the best results are in **bold**, and the second-best are **underlined**.

|                      | Pretrained | Finetuned (100 iters) | Finetuned (1k iters) | Finetuned (10k iters) |
|----------------------|------------|-----------------------|----------------------|-----------------------|
|                      | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ |
| NeRF                 | 15.15 | 0.752 | 0.359 | 19.81 | 0.809 | 0.228 | 23.35 | 0.875 | 0.137 |
| GRF                  | 13.62 | 0.763 | 0.246 | 19.69 | 0.835 | 0.169 | 22.00 | 0.876 | 0.128 | 25.10 | 0.916 | 0.089 |
| Ours                 | **22.37** | **0.904** | **0.096** | **25.58** | **0.928** | **0.076** | **26.86** | **0.941** | **0.061** | **27.95** | **0.950** | **0.053** |

The comparison results on diverse scenes are visualized in Figure 5. We can see that the proposed method synthesizes the novel view image more accurately than IBRNet in all the cases. Specifically, in the Diffuse Synthetic 360° dataset, the proposed method shows better results in the edge regions. The results of IBRNet have some artifacts, which is not the case for the proposed method. The source image feature set in the edge region is easy to include outlier features of the background, which is handled by incorporating the proposed aggregation features in the proposed method. In the case of the Realistic Synthetic 360° dataset, the proposed method synthesizes the image more accurately in the region which is easy to be self-occluded. The proposed method successfully synthesizes the pochlain belt and the inside structure of the microphone. In the case of the Real Forward-Facing dataset, the proposed method shows more accurate results in the region which is easy to be self-occluded (the first example) and in the edge region (the second example). These results demonstrate that the proposed method facilitates us to utilize robust aggregation features in situations where some outliers are included in the source image feature set.

6.4 Ablation study

**Various values of \( n_k \).** We report the ablative results with respect to various values of \( n_k \) in Table 3. We can see that incorporating the proposed source-view-wise aggregation features improves the baseline [44] performance on the Realistic Synthetic 360° and Real Forward-Facing datasets even when we utilize only one learnable similarity measure function. From the results, we can demonstrate the effectiveness of the proposed scheme over the global feature aggregation method. In the case of the Diffuse Synthetic 360° dataset, the proposed method shows comparable performance with the baseline. We speculate that this comes from the fact that the Diffuse Synthetic 360° dataset consists of simple scenes, resulting in little room for improvement. On the one hand, we can also validate the robustness of the proposed framework with respect to the choice of \( n_k \).

**Choice of the distance metric.** In order to validate the benefit of incorporating \( l_2 \)-norm as the distance metric, we check the performance of the proposed scheme in case of utilizing the cosine distance as the distance metric. Note that the \( l_2 \)-norm is commonly used in the multi-view image-based rendering approach [2, 44], and the cosine distance is widely utilized in the self-attention mechanism [43]. To this end, we measure the distance between two features \( f_i \),
Table 3. Ablation results with respect to various values of \(n_k\) on the benchmark datasets. For the protocol, “P” means the “pretrained” protocol. The best results are in **bold**, and the second-best are underlined.

| Method  | Protocol | Diffuse Synthetic 360° [35] | Realistic Synthetic 360° [25] | Real Forward-Facing [24] |
|---------|----------|-----------------------------|-----------------------------|--------------------------|
|         |          | PSNR | SSIM | LPIPS | PSNR | SSIM | LPIPS | PSNR | SSIM | LPIPS |
| Baseline [44] | P | 37.17 | 0.990 | 0.017 | 25.49 | 0.916 | 0.100 | 25.13 | 0.817 | 0.205 |
| \(n_k = 1\) | P | 37.01 | 0.989 | 0.020 | 26.81 | 0.927 | 0.085 | 25.31 | 0.821 | 0.201 |
| \(n_k = 2\) | P | 37.20 | 0.989 | 0.019 | 26.96 | 0.928 | 0.083 | 25.37 | 0.823 | 0.201 |
| \(n_k = 3\) | P | 37.28 | 0.989 | 0.018 | 27.24 | 0.931 | 0.082 | 25.36 | 0.822 | 0.200 |
| Ours (\(n_k = 5\)) | P | **37.45** | **0.990** | **0.017** | **27.13** | **0.930** | **0.083** | **25.47** | **0.826** | **0.198** |

Table 4. Ablation results with respect to various settings on the benchmark datasets. For the protocol, “P” means the “pretrained” protocol. The best results are in **bold**.

| Method | Protocol | Diffuse Synthetic 360° [35] | Realistic Synthetic 360° [25] | Real Forward-Facing [24] |
|--------|----------|-----------------------------|-----------------------------|--------------------------|
|         |          | PSNR | SSIM | LPIPS | PSNR | SSIM | LPIPS | PSNR | SSIM | LPIPS |
| Exp1 (cosine distance) | P | 37.26 | 0.989 | 0.018 | 27.09 | 0.930 | 0.082 | 25.39 | 0.824 | 0.199 |
| Exp2 (1/\(\lambda d\)) | P | 37.42 | 0.989 | 0.018 | 27.10 | 0.930 | 0.080 | 25.33 | 0.823 | 0.200 |
| Exp3 (fixed \(\lambda\)) | P | 37.15 | 0.989 | 0.019 | 26.93 | 0.928 | 0.083 | 25.39 | 0.824 | 0.198 |
| Ours | P | **37.45** | **0.990** | **0.017** | **27.13** | **0.930** | **0.083** | **25.47** | **0.826** | **0.196** |

\(f_j\) as \(d_{i,j} = 1 - f_i \cdot f_j/\|f_i\|_2 \|f_j\|_2\). Here, \(\cdot\) means the dot product operation and \(\|\cdot\|_2\) is the \(l_2\)-norm. The result is summarized in Table 6 (Exp1). We can see that incorporating the cosine distance as the distance metric degrades the performance in most of the cases.

**Choice of the similarity measure function.** We conduct an ablative study with the mapping functions in the form of \(1/(1+\lambda d_{i,j})\). The result is summarized in Table 6 (Exp2). We can see that incorporating this form slightly decreases the performance.

**Learnability of \(\lambda\).** To check the efficacy of the learnability of the similarity measure functions, we report the performance with evenly distributed fixed lambdas (\(\lambda = 0.05, 1.2875, 2.525, 3.7625, 5\)). The result is summarized in Table 6 (Exp3). We can see the decreased performance, which validates the importance of the learnability.

## 7 Conclusions

In this paper, we have proposed a novel source-view-wise feature aggregation method for the multi-view image-based rendering problem. Unlike the previous methods that incorporate equally-weighted aggregation features to figure out surfaces in the estimation process, the proposed method uses a set of locally aggregated features to facilitate us to leverage the local structures in the feature set. In experiments, incorporating the proposed features improves the performance by a large margin, resulting in the state-of-the-art performances. This demonstrates the effectiveness of the proposed scheme. A thing to note here is that the proposed method can be applied to other baseline methods other than the one we used in the implementation. The proposed method could be more effective when used with more powerful baselines, which is left as future work.
Fig. 5. Qualitative comparison results on the benchmark datasets with the “pre-trained” protocol. The proposed method shows more accurate results compared to the other method.
References

1. 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: Proc. IEEE Int’l Conf. Computer Vision (October 2021)
2. Chen, A., Xu, Z., Zhao, F., Zhang, X., Xiang, F., Yu, J., Su, H.: Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. arXiv preprint arXiv:2103.15595 (2021)
3. Chibane, J., Bansal, A., Lazova, V., Pons-Moll, G.: Stereo radiance fields (srf): Learning view synthesis for sparse views of novel scenes. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition (June 2021)
4. Flynn, J., Broxton, M., Debevec, P., DuVall, M., Fyffe, G., Overbeck, R., Snively, N., Tucker, R.: DeepView: View synthesis with learned gradient descent. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition (June 2019)
5. Flynn, J., Neulander, I., Philbin, J., Snively, N.: Deepstereo: Learning to Predict New Views from the World’s Imagery. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition (June 2016)
6. Genova, K., Cole, F., Sud, A., Sarna, A., Funkhouser, T.: Local Deep Implicit Functions for 3D Shape. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition (June 2020)
7. Google Research: Google scanned objects. https://app.ignitionrobotics.org/GoogleResearch/fuel/collections/GoogleScannedObjects
8. Jain, A., Tancik, M., Abbeel, P.: Putting NeRF on a Diet: Semantically Consistent Few-Shot View Synthesis. In: Proc. IEEE Int’l Conf. Computer Vision (October 2021)
9. Jiang, C., Sud, A., Makadia, A., Huang, J., Nießner, M., Funkhouser, T., et al.: Local Implicit Grid Representations for 3D Scenes. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition (June 2020)
10. Kajiya, J.T., Von Herzen, B.P.: Ray tracing volume densities. ACM SIGGRAPH computer graphics 18(3), 165–174 (1984)
11. Kalantari, N.K., Wang, T.C., Ramamoorthi, R.: Learning-Based View Synthesis for Light Field Cameras. ACM TOG (2016)
12. Kar, A., Häne, C., Malik, J.: Learning a Multi-View Stereo Machine. In: Advances in neural information processing systems (2017)
13. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization (2017)
14. Li, Z., Niklaus, S., Snively, N., Wang, O.: Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition (June 2021)
15. Li, Z., Xian, W., Davis, A., Snively, N.: Crowdsampling the Plenoptic Function. In: Proc. European Conference on Computer Vision (August 2020)
16. Lin, C.H., Ma, W.C., Torralba, A., Lucey, S.: BARF: Bundle-Adjusting Neural Radiance Fields. In: Proc. IEEE Int’l Conf. Computer Vision (October 2021)
17. Lindell, D.B., Martel, J.N., Wetzstein, G.: AutoInt: Automatic Integration for Fast Neural Volume Rendering. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition (June 2021)
18. Liu, L., Gu, J., Lin, K.Z., Chua, T.S., Theobalt, C.: Neural Sparse Voxel Fields. In: Advances in neural information processing systems (2020)
19. Liu, S., Zhang, Y., Peng, S., Shi, B., Pollefeys, M., Cui, Z.: Dist: Rendering Deep Implicit Signed Distance Function with Differentiable Sphere Tracing. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition (June 2020)
20. Liu, S., Zhang, X., Zhang, Z., Zhang, R., Zhu, J.Y., Russell, B.: Editing conditional radiance fields. In: Proc. IEEE Int’l Conf. Computer Vision (October 2021)
21. Lombardi, S., Simon, T., Saragih, J., Schwartz, G., Lehrmann, A., Sheikh, Y.: Neural Volumes: Learning Dynamic Renderable Volumes from Images. ACM TOG (2019)
22. Martin-Brualla, R., Radwan, N., Sajjadi, M.S., Barron, J.T., Dosovitskiy, A., Duckworth, D.: NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition (June 2021)
23. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy Networks: Learning 3D Reconstruction in Function Space. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition (June 2019)
24. 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)
25. 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: Proc. European Conference on Computer Vision (August 2020)
26. Niemeyer, M., Mescheder, L., Oechsle, M., Geiger, A.: Differentiable Volumetric Rendering: Learning Implicit 3D Representations without 3D Supervision. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition (June 2020)
27. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition (June 2019)
28. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Wallach, H., Larochelle, H., Beygelzimer, A., d’Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems 32, pp. 8024–8035. Curran Associates, Inc. (2019)
29. Penner, E., Zhang, L.: Soft 3D Reconstruction for View Synthesis. SIGGRAPH Asia (2017)
30. Rahaman, N., Baratin, A., Arpit, D., Draxler, F., Lin, M., Hamprecht, F., Bengio, Y., Courville, A.: On the spectral bias of neural networks. In: International Conference on Machine Learning. pp. 5301–5310. PMLR (2019)
31. Rebain, D., Jiang, W., Yazdani, S., Li, K., Yi, K.M., Tagliasacchi, A.: DeRF: Decomposed Radiance Fields. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition (June 2021)
32. Saito, S., Huang, Z., Natsume, R., Morishima, S., Kanazawa, A., Li, H.: PIFu: Pixel--aligned Implicit Function for High-resolution Clothed Human Digitization. In: Proceedings of the IEEE International Conference on Computer Vision (October 2019)
33. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. International journal of computer vision 47(1), 7–42 (2002)
34. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition (June 2016)
35. Sitzmann, V., Thies, J., Heide, F., Nießner, M., Wetzstein, G., Zollhofer, M.: DeepVoxels: Learning Persistent 3D Feature Embeddings. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition (June 2019)
36. Sitzmann, V., Zollhöfer, M., Wetzstein, G.: Scene Representation Networks: Continuous 3D-Structure-Aware Neural Scene Representations. In: Advances in neural information processing systems (2019)
37. Srinivasan, P.P., Tucker, R., Barron, J.T., Ramamoorthi, R., Ng, R., Snavely, N.: Pushing the Boundaries of View Extrapolation with Multiplane Images. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition (June 2019)
38. Szelski, R., et al.: Image alignment and stitching: A tutorial. Foundations and Trends® in Computer Graphics and Vision 2(1), 1–104 (2007)
39. Tretschk, E., Tewari, A., Golyanik, V., Zollhöfer, M., Lassner, C., Theobalt, C.: Non-Rigid Neural Radiance Fields: Reconstruction and Novel View Synthesis of a Dynamic Scene From Monocular Video. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition (June 2021)
40. Trevithick, A., Yang, B.: GRF: Learning a General Radiance Field for 3D Representation and Rendering. In: Proc. IEEE Int’l Conf. Computer Vision (October 2021)
41. Tucker, R., Snavely, N.: Single-View View Synthesis with Multiplane Images. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition (June 2020)
42. Tulsiani, S., Zhou, T., Efros, A.A., Malik, J.: Multi-view Supervision for Single-view Reconstruction via Differentiable Ray Consistency. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition (July 2017)
43. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems (2017)
44. 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: Proc. IEEE Conf. Computer Vision and Pattern Recognition (June 2021)
45. Wang, Z., Bovik, A., Sheikh, H., Simoncelli, E.: Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing 13(4), 600–612 (2004). [https://doi.org/10.1109/TIP.2003.819861]
46. Yariv, L., Kasten, Y., Moran, D., Galun, M., Atzmon, M., Basri, R., Lipman, Y.: Multiview Neural Surface Reconstruction by Disentangling Geometry and Appearance. In: Advances in neural information processing systems (2020)
47. Yu, A., Li, R., Tancik, M., Li, H., Ng, R., Kanazawa, A.: PlenOctrees for Real-time Rendering of Neural Radiance Fields. In: Proc. IEEE Int’l Conf. Computer Vision (October 2021)
48. Yu, A., Ye, V., Tancik, M., Kanazawa, A.: pixelnerf: Neural radiance fields from one or few images. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition (June 2021)
49. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition (June 2018)
50. Zhou, T., Tucker, R., Flynn, J., Fyffe, G., Snavely, N.: Stereo Magnification: Learning view synthesis using multiplane images. SIGGRAPH (2018)
A Appendix

A.1 Analysis of the learned parameters

In order to provide an extra analysis of the proposed method, we report the learned parameter values of the proposed similarity measure functions. The values of the pretrained model are summarized in Table 5. As we incorporate larger $n_k$, the learned parameters tend to be evenly distributed, which facilitates the diverse similarity measure functions to model the complex feature distributions.

Table 5. Learned parameter values of the proposed similarity measure functions with respect to various values of $n_k$.

| $n_k$ | $\lambda$ of the coarse network | $\lambda$ of the fine network |
|------|---------------------------------|-------------------------------|
| 1    | 1.15                            | 0.36                          |
| 2    | 0.40, 1.59                      | 0.10, 0.41                    |
| 3    | 0.18, 0.66, 2.19                | 0.08, 0.39, 1.05              |
| 4    | 0.11, 0.41, 0.92, 2.68          | 0.13, 0.51, 1.27, 3.67        |
| 5    | 0.08, 0.32, 0.52, 0.90, 2.36     | 0.05, 0.32, 0.67, 1.48, 3.99  |

A.2 Additional qualitative results

We provide qualitative comparison results with IBRNet [44] based on the “finetuned” protocol. The comparison results on various scenes are visualized in Figure 6. In the case of the Diffuse Synthetic 360° and Realistic Synthetic 360° datasets, the proposed method synthesizes the novel view images more accurately compared with IBRNet. Specifically, in the first, second, and fourth cases, the proposed method shows better quality in edge regions. In the third case, the proposed method synthesizes the ceiling region more clearly. In the case of the Real Forward-Facing dataset, the proposed method shows comparable performances compared with IBRNet. Note that we also submit the video comparison result on the “pretrained” protocol.

A.3 Additional ablation study

Various aggregation methods. For a further validation, we conduct ablation studies with different aggregation methods. For the experiments, we change the source feature aggregation method of IBRNet (element-wise mean+var aggregation) to element-wise mean aggregation (Exp1) and attention-based aggregation (Exp2). For the attention aggregation, we utilize Slot Attention [?] of GRF [40]. Experimental results (Exp1, Exp2, and Ours) in Table 6 show that our method is effective compared to different aggregation methods.

Neighboring view selection. The proposed scheme improves the baseline performance by handling occlusions from similar viewpoints, which is qualitatively
Table 6. Ablation results with respect to various settings on the benchmark datasets. For the protocol, “P” means the “pretrained” protocol. The best results are in **bold**.

| Method          | Protocol | Diffuse Synthetic 360° PSNR | SSIM | LPIPS | Realistic Synthetic 360° PSNR | SSIM | LPIPS | Real Forward-Facing 25° PSNR | SSIM | LPIPS |
|-----------------|----------|-----------------------------|------|-------|-------------------------------|------|-------|-------------------------------|------|-------|
| Exp1 (mean aggre) | P        | 36.94                       | 0.988| 0.021 | 26.66                         | 0.924| 0.091 | 25.10                         | 0.816| 0.207 |
| Exp2 (atten aggre) | P        | 36.92                       | 0.997| 0.032 | 26.82                         | 0.920| 0.105 | 25.12                         | 0.813| 0.207 |
| Exp3 (harsh; mean+var aggre) | P    | 33.99                       | 0.979| 0.034 | 22.82                         | 0.877| 0.145 | 23.20                         | 0.765| 0.264 |
| Exp4 (harsh; our aggre) | P      | 34.72                       | 0.982| 0.028 | 23.61                         | 0.890| 0.130 | 23.50                         | 0.774| 0.256 |
| Ours            | P        | **37.45**                   | **0.990**| **0.017** | **27.13**                         | **0.930**| **0.083** | **25.47**                         | **0.826**| **0.196** |

shown in Figure 5 of the main paper. For a further validation, we conduct a comparison experiment of the baseline aggregation method and the proposed method in more harsher setting which does not select the most neighboring three views. In this setting, the occlusion might occur more frequently. The results are summarized in Table 6 (Exp3 and Exp4). The proposed method shows the better performance, validating the occlusion handling efficacy of our method.

A.4 Additional implementation details

We build the proposed framework based on the official IBRNet code[4]. The proposed framework is trained on four V100 GPUs. Each GPU utilizes a batch of 500 rays which are sampled from a randomly selected scene. For the feature extraction network, the coarse and fine networks share the same network except for the last layer, i.e., the coarse and fine networks have their own convolutional weights for the last layer. In the feature extraction process, we only utilize the features extracted from the valid image region. We also submit the code snippet of the proposed method for reference.

[4] https://github.com/googleinterns/IBRNet
Fig. 6. Qualitative comparison results on the benchmark datasets. The proposed method shows more accurate or comparable results compared with IBRNet.