electronics

MDPI

Article

MM-NeRF: Large-Scale Scene Representation with Multi-Resolution Hash Grid and Multi-View Priors Features

Bo Dong [1,2,3,4], Kaiqiang Chen [1,4,*], Zhirui Wang [1,4], Menglong Yan [1,4,5,6], Jiaojiao Gu [7] and Xian Sun [1,4]

[1] Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100190, China; dongbo21@mails.ucas.ac.cn (B.D.)
[2] University of Chinese Academy of Sciences, Beijing 100190, China
[3] School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing 100190, China
[4] Key Laboratory of Network Information System Technology (NIST), Institute of Electronics, Chinese Academy of Sciences, Beijing 100190, China
[5] Jigang Defence Technology Company, Ltd., Jinan 250132, China
[6] Cyber Intelligent Technology (Shandong) Co., Ltd., Jinan 250100, China
[7] Coastal Defense College, Naval Aeronautical University, Yantai 264001, China
* Correspondence: chenkq@aircas.ac.cn

Abstract: Reconstructing large-scale scenes using Neural Radiance Fields (NeRFs) is a research hotspot in 3D computer vision. Existing MLP (multi-layer perception)-based methods often suffer from issues of underfitting and a lack of fine details in rendering large-scale scenes. Popular solutions are to divide the scene into small areas for separate modeling or to increase the layer scale of the MLP network. However, the subsequent problem is that the training cost increases. Moreover, reconstructing large scenes, unlike object-scale reconstruction, involves a geometrically considerable increase in the quantity of view data if the prior information of the scene is not effectively utilized. In this paper, we propose an innovative method named MM-NeRF, which integrates efficient hybrid features into the NeRF framework to enhance the reconstruction of large-scale scenes. We propose employing a dual-branch feature capture structure, comprising a multi-resolution 3D hash grid feature branch and a multi-view 2D prior feature branch. The 3D hash grid feature models geometric details, while the 2D prior feature supplements local texture information. Our experimental results show that such integration is sufficient to render realistic novel views with fine details, forming a more accurate geometric representation. Compared with representative methods in the field, our method significantly improves the PSNR (Peak Signal-to-Noise Ratio) by approximately 5%. This remarkable progress underscores the outstanding contribution of our method in the field of large-scene radiance field reconstruction.

Keywords: NeRF; scene representation; view synthesis; hash grid feature; multi-view prior

check for updates

Citation: Dong, B.; Chen, K.; Wang, Z.; Yan, M.; Gu, J.; Sun, X. MM-NeRF: Large-Scale Scene Representation with Multi-Resolution Hash Grid and Multi-View Priors Features. *Electronics* **2024**, *13*, 844. https://doi.org/10.3390/electronics13050844

Academic Editor: Beiwen Li

Received: 16 January 2024
Revised: 17 February 2024
Accepted: 20 February 2024
Published: 22 February 2024

1. Introduction

With the development of deep learning technology, learning-based neural networks are beginning to replace traditional methods in various industries, such as medicine [1], finance [2], manufacturing [3,4], etc. However, accurate modeling of 3D scenes has always been a challenging problem. In recent times, the succession of methods employing Neural Radiance Fields (NeRFs) [5] for the representation of large-scale 3D scenes [6–9] has achieved notable success. These studies have greatly promoted the development of the meta-universe, virtual reality, animations, and more. Existing methods mainly use a progressive update scheme [6,7] to construct the final 3D representation or divide the scene into multiple partitions, each represented by a multi-layer perception (MLP) model [8,9]. However, these methods based on MLP architecture have the problem of losing details when simulating large and complex scenes due to the limited model capacity and can only generate blurry renderings [6,8,9]. Furthermore, the MLP model learns from zero knowledge, lacking some prior information input. This results in each scene requiring a large amount of view source data [6,8], further limiting their application in the real world.

Lately, we noticed that some NeRF variants [10–12] offer insights that may address the aforementioned challenges. Specifically, some methods focusing on accelerating object-scale NeRF optimization propose storing local features in a three-dimensional dense voxel grid [10,11] or hash grid [12]. Grid features make it easy to fit local scene content with explicitly and independently learned features, replacing extensive MLP computations with fast feature interpolation. However, using a dense voxel grid to represent large-scale scenes, the number of parameters will grow cubically as the scene increases, so existing methods [11] often use smaller resolutions during the optimization process. The multi-resolution hash grid is another structure that has been used, which applies a hash function to randomly map three-dimensional points into a hash table. The resolution can be set to a larger number. However, a failure to provide additional information can lead to suboptimal results in the presence of hash collisions.

Another distinctive variant that motivated us is the generalizable NeRF [13–16], which aims to give NeRF the ability to model general scene structures by inputting additional information from images. Existing methods [13–16] usually employ a pipeline of an image encoder to embed multi-view images into a prior $\mathbf{z}$ and a NeRF as the decoder input 3D position conditioned on $\mathbf{z}$ to generate the target view image. These variants perform well in object-scale scenes, requiring only a few (e.g., three [13,14]) camera views to synthesize new views without any retraining. We try to transfer the capabilities of the generalizable NeRF, using the image encoder to provide scene priors for large-scale scenes, thereby reducing the dependence on the number of views. Simultaneously, the priors information can serve as a supplement to the hash grid to solve the suboptimization problem under hash collisions.

To summarize, we integrate the multi-resolution hash grid feature with the generalizable NeRF encoder–decoder pipeline and apply it to large-scale scenes, proposing a high-resolution refined neural representation method that does not require a large number of multi-views, called MM-NeRF. Our major contributions can be summarized as follows:

- We propose a new optimization NeRF variant, called MM-NeRF, that is specifically designed for large-scale unbounded scene modeling.
- We introduce a new pipeline that integrates complementary features from 3D hash grids and scene priors to achieve efficient and accurate large-scene modeling.
- Our MM-NeRF achieves good scene synthesis representation without requiring a large number of views, indicating the superior performance of our model.

2. Related Work

2.1. NeRF

NeRFs [5] represent 3D scenes as a radiance field approximated using multi-layer perception (MLP). The MLP takes the position and viewing direction of 3D points as input to predict their color and density. Combined with volume rendering [17], NeRF achieves a photo-realistic rendering quality and has attracted considerable attention. Many follow-up methods have been developed to improve the quality of synthesized views [18–20], training and inference speed [12,21–23], explore model generalization based on sparse views [13,14,24,25], and pose estimation [26–28]. Further, some recent works have developed NeRF for more complex tasks, such as dynamic scenes [29,30], controllable editing [31,32], multi-modality [33], etc.

2.2. Large-Scale Scene NeRF

Although the vanilla NeRF [5] was designed only to handle object-scale scenes, scaling up NeRFs to large-scale scenes such as cities will enable a wider range of applications. NeRF-W [34] was the first attempt to apply NeRF to outdoor scenes. BungeeNeRF [6] and NeRFusion [7] propose a progressively updated reconstruction scheme to reconstruct large indoor and outdoor scenes, respectively. Mega-NeRF [8] and Block-NeRF [9] adopt a divide-

and-conquer strategy to handle large-scale scenes, decomposing the scene into multiple regions, each of which is represented by a single NeRF. However, these methods merely consider the reconstruction results of large scenes, and there is insufficient improvement in the model framework. Therefore, like most MLP-based NeRFs, the loss of details and a large number of views in these methods when dealing with large and complex scenes are still challenging problems to be solved.

2.3. Grid-Based NeRF

NeRFs use MLP to approximate implicit functions for representing 3D scenes, with the benefit of occupying minimal memory. However, each sampling point in the space needs to undergo forward calculation by MLP, resulting in a very low efficiency. Instant-ngp [12] introduces a hash encoding strategy that utilizes hash searches to obtain 3D features and then connects the NeRF pipeline to achieve rendering output. Hash searches are much faster than MLP calculation, greatly speeding up NeRF. Plenoxels [10] and DVGO [11] take a more aggressive approach by directly substituting a dense voxel grid for MLP and performing volume rendering on the interpolated 3D features. However, both of these approaches have their limitations. Specifically, hash encoding may encounter issues with search conflicts, while the representation using only voxel grids becomes memory-intensive as the scene scale increases. Therefore, in NeRFs of large-scale scenes [35,36], grid-based features are often used as one of the multiple branching features. Our method also adopts a similar strategy.

2.4. Generalizable NeRF

Pioneer works [13–16] mix the 2D features independently extracted from each input view and inject them into the MLP, providing an intuitive mechanism to adapt NeRF. However, these methods struggle to handle complex scenes effectively due to the lack of explicit geometric awareness encoding in the features. Following methods [24,37,38] verify that introducing geometric priors can improve generalization. Particularly, MVSNeRF [24] constructs a cost volume and then applies a 3D CNN to reconstruct a neural encoding volume with per-voxel neural features. GeoNeRF [37] further enhances the architecture by using a cascaded cost volume and incorporating attention modules. NeuRay [38] calculates a visibility feature map with the cost volumes or depth maps to select whether the 3D point is visible. All these geometry priors based on cost volumes are sensitive to the choice of the reference view. In contrast, we introduce a matching-based strategy to incorporate geometric priors without requiring cost volumes or 3D CNNs.

3. Methods

To effectively represent large-scale scenes, we propose MM-NeRF, which combines the expertise of grid representation-based methods and prior feature-based methods. We leverage multi-resolution hash grids to capture as much 3D detail as possible, then we let the views encoder supplement the missing prior information and finally produce high-quality renderings with NeRF.

Figure 1 illustrates our overall pipeline. Our method comprises two branches, namely the multi-resolution hash feature branch and the multi-view priors feature branch. First, we sample 3D points along rays cast from pixels. Second, the sampling points undergo the multi-resolution hash branch to obtain multi-resolution grid features with geometric significance. Simultaneously, they pass through the multi-view prior branch to obtain prior features. These features, along with position encoding (PE), are then fed into the decoder, which predicts density σ and color values c. Finally, the image colors can be computed through volume rendering.

In Section 3.1, we describe the feature branch of multi-resolution grid representation. In Section 3.2, we introduce the prior feature extraction branch based on the image encoder and attention mechanism. Finally, Section 3.3 provides detailed insights into how NeRFs implement the rendering output.

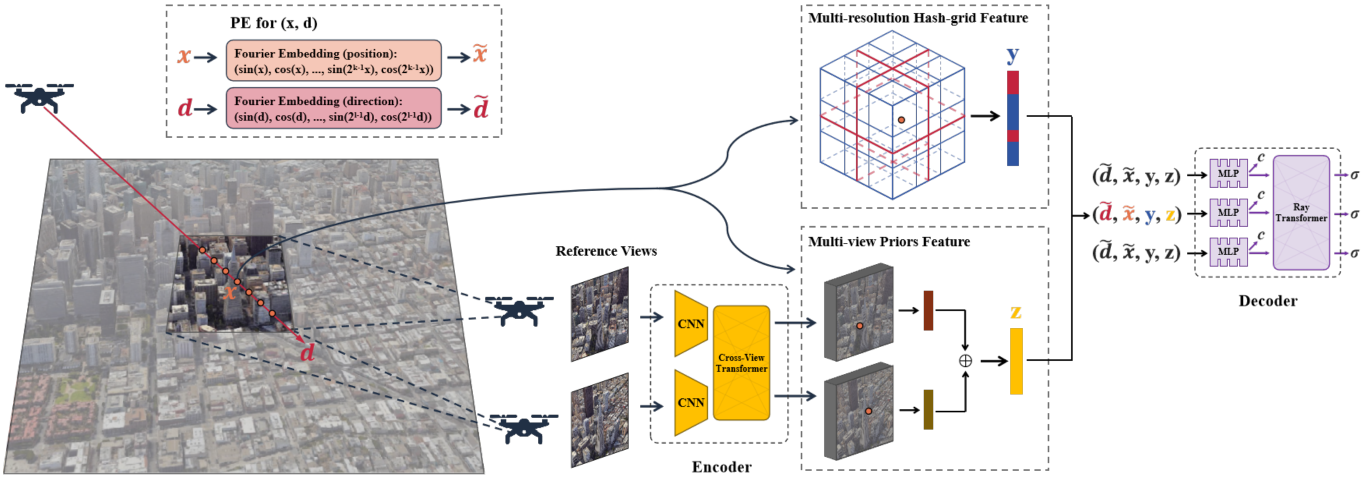
PE for (x, d)
x
Fourier Embedding (position):
(sin(x), cos(x), ..., sin(2^{k-1}x), cos(2^{k-1}x))
x̃
d
Fourier Embedding (direction):
(sin(d), cos(d), ..., sin(2^{l-1}d), cos(2^{l-1}d))
d̃
x
d
Multi-resolution Hash-grid Feature
y
Reference Views
CNN
CNN
Cross-View Transformer
Encoder
Multi-view Priors Feature
z
⊕
(d̃, x̃, y, z)
(d̃, x̃, y, z)
(d̃, x̃, y, z)
MLP
MLP
MLP
c
c
c
Ray Transformer
σ
σ
σ
Decoder

Figure 1. Overview. Our method involves two branches: a multi-resolution hash grid feature branch and a multi-view prior feature branch, the output of which is combined with position encoding (PE) and is fed into in the decoder to predict density σ and color value c.

3.1. Multi-Resolution Hash Grid Feature

Recall that NeRFs predict point density and color by passing point coordinates' position encoding (PE) [5] into an 8-layer MLP. The compact model encodes the entire scene content in an MLP that takes PE embeddings as input, but it is difficult to expand the scene due to the limitation of the model capacity. In contrast, the multi-resolution hash grid [12] is an efficient data structure, which divides space into closely adjacent small cubic cells, similar to a voxel grid. However, the features of each unit area are not stored on the cube's vertices but instead stored centrally in the form of a hash table. And the space is repeatedly divided at different resolutions. Therefore, unlike mapping 3D points to a fixed-size voxel grid, a multi-resolution hash grid does not significantly increase the number of parameters when dealing with an increase in the scale of the scene.

The number of parameters of the multi-resolution hash grid is bounded by $L \cdot T \cdot F$, where L is the number of resolutions and T and F are the hash table size and feature dimension of each resolution. We set $L = 16$, $T = 2^{19}$, and $F = 2$ to balance the trade-off between capacity and efficiency, so the total number of parameters is 2^{24}. Figure 2 illustrates the steps for obtaining features for our multi-resolution hash grid. For a given input coordinate $\mathbf{x}$, first, obtain the indices $\mathbf{V} = \{(\mathbf{p}_i)_{i=1}^{8} \mid \mathbf{p}_i \in \mathbb{R}^3\}$ of surrounding voxel vertices under the grid at different resolutions (red and blue in Figure 2 represent two different resolutions). According to the hash function $h : \mathbf{V} \to \mathbf{Y}$, fetch the features from the hash table and perform linear interpolation based on the relative position of $\mathbf{x}$ in different resolution grids. Then, we concatenate the results of each level together to form multi-resolution hash features $\mathbf{y}$ of point $\mathbf{x}$, as one of the branch inputs of NeRFs. For a vertex $\mathbf{p} = (p_x, p_y, p_z)$, we adopt the hash function and resolution settings used in Instant-ngp [12]:

$$h(\mathbf{p}) = (\oplus\, p_i \pi_i) \bmod T, i = x, y, z, \tag{1}$$

where $\oplus$ represents the bitwise XOR operation and π_i is the unique large prime number, $\pi_x = 1$, π_y = 2,654,435,761, and π_z = 805,459,861, respectively. The resolution of each level is chosen to be a geometric progression between the coarsest and finest resolutions $[N_{min}, N_{max}]$:

$$N_l = \left\lfloor N_{min} \cdot b^l \right\rfloor, \tag{2}$$

$$b = \exp(\frac{\ln N_{max} - \ln N_{min}}{L - 1}). \tag{3}$$

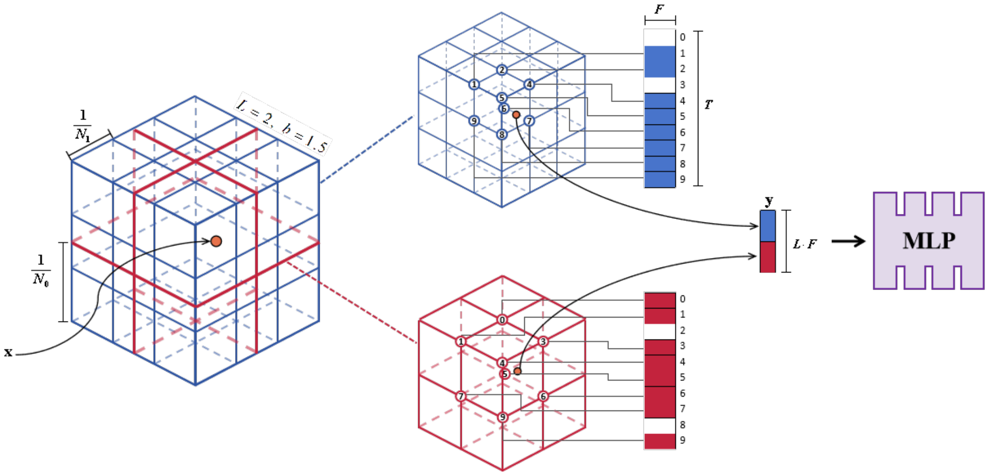

Figure 2. Illustration of multi-resolution hash features. For a given input coordinate $\mathbf{x}$, we locate surrounding vertices at different resolution levels and fetch their $\mathbf{F}$-dimensional feature vectors in a hash table then linearly interpolate to obtain the features of $\mathbf{x}$. The features of the vertices are updated via the gradients returned by the MLP.

Some works [12,36] have shown that multi-resolution hash grid representation is significantly better than dense voxel representation when dealing with scene scale expansion. However, due to collision issues in hash mapping, the interpolated features inevitably contain information from multiple distinct surface points, limiting the performance of the NeRF model. An intuitive solution is to increase the hash table size T to achieve improvements, at the cost of a significant increase in the number of parameters and a longer optimization time. This result prompted us to introduce an effective strategy to enhance the hash grid features for large-scale scenes by introducing prior information.

3.2. Multi-View Priors Feature

Previous generalized NeRF [13–16,24,37,38] methods typically used the CNN+MLP architecture. The CNN serves as an encoder for extracting 2D features from input views. These features are then aggregated in various ways and propagated backward [13–16] or used to construct an intermediate product [24,37,38] (e.g., a cost volume). MLP works as a decoder to output color and density. Our goal is to develop a similar architecture, but, different from others, we propose to use the Transformer for cross-view interactions for CNN features, followed by projecting 3D points onto them for interpolation. The lower branch in Figure 1 shows the specific details of our framework, which consists of an encoder f_θ consisting of a CNN and a Transformer to extract cross-view aligned features. The decoder g_ϕ adopts the structure of IBRNet [14], including MLP and the Transformer, which predict color and density, respectively, for volume rendering.

Regarding the encoder, we first use a weight-shared CNN [39] to extract downsampled convolutional features $\{\mathbf{F}_i^c\}_{i=1}^N$ from N input views $\{\mathbf{I}_i\}_{i=1}^N$. Features between different views are interacted with through a Transformer with cross-attention to further enhance the feature quality. The process can be described as follows:

$$\mathcal{T} : (\mathbf{F}_1^c, \mathbf{F}_2^c, \cdots, \mathbf{F}_N^c) \rightarrow (\mathbf{F}_1, \mathbf{F}_2, \cdots, \mathbf{F}_N), \tag{4}$$

where $\mathcal{T}$ represents the Transformer. For its structure, we followed GMFlow [40]. The convolutional features $\{\mathbf{F}_i^c\}_{i=1}^N$ are input to the Transformer through shift windows. To mitigate the impact of noise, we perform a summation followed by averaging on the Transformer features. As shown in Figure 3, for a given 3D point position $\mathbf{x}$, we first project it onto the 2D Transformer features $\{\mathbf{F}_i\}_{i=1}^N$ of the views $\{\mathbf{I}_i\}_{i=1}^N$ using the camera parameters $\{\mathbf{M}_i\}_{i=1}^N$ and then perform bilinear sampling to obtain features $\{\mathbf{f}_i\}_{i=1}^N$. We divide the feature vectors $\{\mathbf{f}_i\}_{i=1}^{N}$ into G groups along the channel dimension and then sum and average the features of each group:

$$\mathbf{z} = \frac{\sum\limits_{g=1}^{G}\sum\limits_{i=1}^{N}\mathbf{f}_i^{(g)}}{N \cdot G}, \tag{5}$$

where $\mathbf{z}$ represents the cross-view feature of 3D point $\mathbf{x}$, which is used as a prior in our method to capture view-consistent matching information.

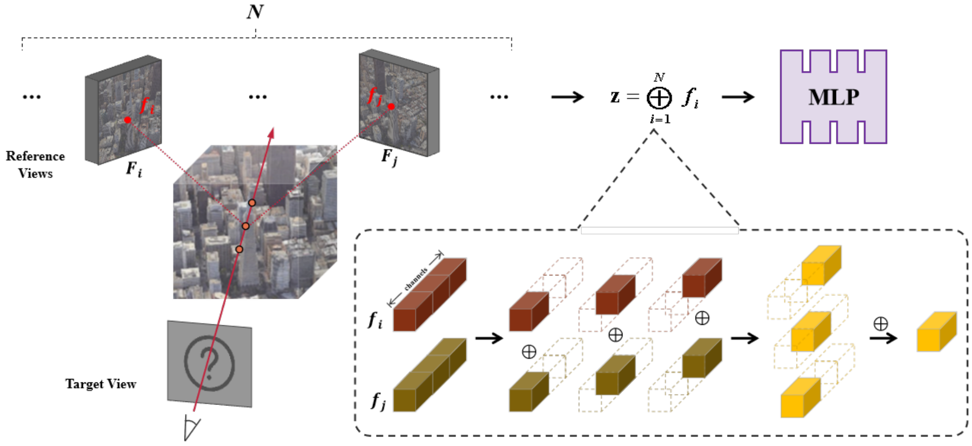

Figure 3. The encoder generates 2D features for cross-view interaction on N input images. Project 3D points onto 2D feature planes for bilinear sampling. Aggregating the sampled features of different views in the channel direction forms a prior fed to the MLP.

3.3. NeRF Render Network

For the original input of the render network, we are the same as the vanilla NeRF [5], which involves hierarchical sampling and positional encoding (PE). Building upon this, we added the grid feature $\mathbf{y}$ from Section 3.1 and the prior $\mathbf{z}$ from Section 3.2 as additional inputs to the decoder g_ϕ for predicting the color and density of 3D point $\mathbf{x}$. $\mathbf{y}$ and $\mathbf{z}$ represent the features in 3D space and the features of the 2D view, respectively. By fusing these two features we are able to not only capture the three-dimensional structure in space but also obtain prior information in the 2D perspective. The input–output representation of the decoder can be expressed as follows:

$$g_\phi : (\tilde{\mathbf{d}}, \tilde{\mathbf{x}}, \mathbf{y}, \mathbf{z}) \rightarrow (\mathbf{c}, \sigma), \tag{6}$$

where $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{d}}$ represent the high-frequency encoding results of the 3D point and the viewing direction, $\mathbf{y}$ denotes the multi-resolution hash-grid features, and $\mathbf{z}$ is the prior capturing view-consistent information. The output is a pair of color $\mathbf{c}$ and density value σ.

Considering the limited decoding capabilities of a simple MLP, we construct a rendering network including both MLP and a Transformer, following the previous work [14]. As shown in Figure 1, the Transformer can introduce cross-point interactions by fusing the rendered information along a ray, predicting the density σ. And the MLP predicts color $\mathbf{c}$. Using the predicted color $\mathbf{c}$ and volume density σ from the decoder, a new view can be synthesized via volume rendering. Volume rendering calculates the color $\mathbf{C}$ for a pixel by accumulating colors according to the density for all sampling points on the corresponding rays passing through the pixel:

$$\mathbf{C} = \sum_{i=1}^{K} T_i (1 - \exp(-\sigma_i \delta_i)) \mathbf{c}_i, \tag{7}$$

$$T_i = \exp(-\sum_{j}^{i-1} \sigma_i \delta_i), \tag{8}$$

where $\mathbf{c}_i$ and σ_i refer to the color and density of the i-th sampled 3D point on the ray. T_i is the volume transmittance, and δ_i denotes the distance between adjacent points. K is the total number of 3D points sampled on a ray.

We train the model end-to-end using only the photometric loss function, without requiring any other ground-truth geometric data:

$$\mathcal{L} = \sum_{\mathbf{p} \in \mathcal{P}} \| C_p - \tilde{C}_p \|_2^2, \tag{9}$$

where $\mathcal{P}$ denotes the set of pixels within one training batch and C_p and $\tilde{C}_p$ refer to the rendered color and the ground-truth color of pixel p, respectively.

Algorithm 1 shows the pseudocode of our proposed algorithm to better understand our method and implementation process.

Algorithm 1 Optimization process of MM-NeRF.

```
Input: multi-view images $\{\boldsymbol{I}_i\}_{i=1}^{N}$, camera poses $\{\boldsymbol{M}_i\}_{i=1}^{N}$, system initialization parameters $\mathcal{S}$
Parameter: number of sampling points $N$, max frequency for PE $M_f$, angle threshold of the reference view $\theta$, hash parameter $L$, $T$, $F$, grid resolutions range $N_{min}$, $N_{max}$
$\{N_l\} \leftarrow N_{min}, N_{max}$                                   ▷ Formulas (2) and (3)
$\{\boldsymbol{o}, \boldsymbol{d}, t\} \leftarrow \{\boldsymbol{M}_i\}_{i=1}^{N}$                              ▷ Ray parameters
for iter=1,2,... do
    $\{\boldsymbol{x}\} \leftarrow \{\boldsymbol{o}, \boldsymbol{d}, t\}$                                ▷ Sampling points
    $\boldsymbol{V} \leftarrow \boldsymbol{x}$                                          ▷ Surrounding voxel vertices
    $\boldsymbol{F}^{\boldsymbol{V}} \leftarrow hash(\boldsymbol{V})$                                   ▷ Formula (1)
    $\boldsymbol{y} \leftarrow \boldsymbol{F}^{\boldsymbol{V}}$                                         ▷ Grid Feature
    $(\boldsymbol{I}_j, \boldsymbol{M}_j)_{j=1}^{K} \leftarrow (\{\boldsymbol{I}_i\}_{i=1}^{N}, \{\boldsymbol{M}_i\}_{i=1}^{N}, \theta)$     ▷ Reference views
    $\{\boldsymbol{F}_j^c\}_{j=1}^{K} \leftarrow CNN(\{\boldsymbol{I}_j\}_{j=1}^{K})$
    $\{\boldsymbol{F}_j\}_{j=1}^{K} \leftarrow \mathcal{T}(\{\boldsymbol{F}_j^c\}_{j=1}^{K})$                     ▷ Formula (4)
    $\boldsymbol{z} \leftarrow (\boldsymbol{F}_j, \boldsymbol{M}_j)_{j=1}^{K}$                               ▷ Priors feature
    $\tilde{\boldsymbol{d}}, \tilde{\boldsymbol{x}} = \mathrm{PE}(\boldsymbol{d}, \boldsymbol{x})$                                 ▷ Position encoding
    $\boldsymbol{c}, \sigma \leftarrow (\tilde{\boldsymbol{d}}, \tilde{\boldsymbol{x}}, \boldsymbol{y}, \boldsymbol{z})$                              ▷ Formula (6)
    $\boldsymbol{I'}_i \leftarrow (\boldsymbol{c}, \sigma)$                                    ▷ Formulas (7) and (8)
    $\mathcal{L} \leftarrow (\{\boldsymbol{I}_i\}_{i=1}^{N}, \{\boldsymbol{I'_i}\}_{i=1}^{N})$                     ▷ Formula (9)
    $\mathcal{S} \leftarrow \mathcal{S} + \nabla_S \mathcal{L}$
end for
Output: $\{\boldsymbol{I'}_i\}_{i=1}^{N}$
```

4. Results

4.1. Data

To evaluate our proposed method, we used the public large-scale dataset provided by BungeeNeRF [6]. The dataset was synthesized using Google Earth Studio, capturing multi-scale city images from drone to satellite height using specified camera positions. And the data quality is sufficient to simulate real-world challenges. We used two of these scenes for experiments.

To further verify our method's generalizability in the real world, we conducted experiments on three real-world scenes of the UrbanScene3D dataset [41]. In addition, we also created our dataset, which includes multi-view images of four architectural scenes, each taken at a different height range. Please see Table 1 for details. We employed COLMAP [42] to obtain the initial camera pose.

Table 1. Details of the real-world scene dataset we created: all four scenes were captured at various heights from the drone perspectives, and then the video was framed to obtain a certain number of views. Finally, we used COLMAP to estimate camera poses.

Scene [1,2]	Building Height (m)	Viewing Height (m)	Number of Views
Aerospace Information Museum, Jinan	21	20–30	65
Yellow River Tower, Binzhou	55.6	10–80	153
Meixihu Arts Center, Changsha	46.8	60–80	81
Greenland Xindu Mall, Hefei	188	100–200	169

[1] In the following, AIM, YRT, MAC, and GXM are used to refer to the Aerospace Information Museum, Yellow River Tower, Meixihu Arts Center, and Greenland Xindu Mall, respectively. [2] AIM is derived from our drone collection, while YRT, MAC, and GXM are sourced from internet videos.

4.2. Evaluation

To assess the effectiveness of our method, we employed three metrics: the PSNR, SSIM [43], and LPIPS [44]. The results are presented in Tables 2–4. We first compared our method with the classical NeRF [5] and Mip-NeRF [18]. As expected, these general methods, not specifically optimized for large scenes, fall short compared to ours.

We then compared it with large-scale NeRF variants (BungeeNeRF [6], Mega-NeRF [8]). Compared to these purely MLP-based methods, our method brings sharper geometry and finer details. In particular, due to the inherent limitations in the capacity of MLP, it often fails to simulate rapid and diverse changes in geometry and color, such as building exterior walls with multiple textures. Although dividing the scene into small regions (Mega-NeRF [8]) or increasing the structure size of MLPs (BungeeNeRF [6]) can be somewhat helpful, the rendered results still appear overly smooth. In contrast, guided by the learned grid features, the sampling points are effectively compressed close to the scene surface and coupled with multi-view priors, providing rich geometric and surface information and supplementing the missing scene details in grid features.

In addition, we also extended the comparative analysis to a wider range of non-NeRF methods [11,45] to verify the superiority of our method. The results are shown in Tables 2–4.

For Google scenes, as shown in Table 2, our method outperforms other methods in PSNR, LPIPS, and SSIM. Specifically, our method achieves 24.963 dB and 24.778 dB values for the PSNR for 56Leonard and Transamerica, respectively, which is an average improvement of 2.5 dB compared to the optimal method. The rendering results presented in Figure 4 indicate that our method produces more refined novel views. For large urban scenes with either a distant (top row in Figure 4) or a closer view (bottom row in Figure 4), the results from other methods could exhibit blurriness, while our method ensures detail preservation, resulting in clearer, less noisy outcomes that excel in overall quality and detail.

For real-world scenes, Tables 3 and 4 outlines the metrics for our method and others. We still outperform others across three metrics. Specifically, our method achieves an average PSNR of 24.296 dB. There is an average improvement of 1 dB compared to the optimal method. In Figures 5 and 6, we present the rendering results for real-world scenes, showcasing the notable superiority of our method in terms of details compared to other methods. As shown in the figures, BungeeNeRF and Mega-NeRF generate blurry textures and smooth boundaries. In contrast, our method can synthesize novel views with finer textures and clear boundaries that are very close to the ground truth.

Table 2. Quantitative comparison on Google scenes dataset. We report PSNR (↑), LPIPS (↑), and SSIM (↓) metrics on the test view. We highlighted the best and second-best results.

	56Leonard (Avg.)			Transamerica (Avg.)		
	PSNR↑ [1]	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑
NeRF [5]	21.107	0.335	0.611	21.420	0.344	0.625
Mip-NeRF [18]	21.642	0.299	0.695	21.820	0.331	0.687
DVGO [11]	21.317	0.323	0.631	21.467	0.337	0.606
TensoRF [45]	22.289	0.310	0.658	22.023	0.303	0.664
Mega-NeRF [8]	22.425	0.372	0.680	22.546	0.283	0.707
BungeeNeRF [6]	23.058 [3]	0.245	0.736	23.232	0.232	0.721
MM-NeRF (ours)	**24.963** [2]	**0.182**	**0.814**	**24.778**	**0.197**	**0.802**

[1] The upward arrow indicates that the higher the metric, the better. The downward arrow indicates that the lower the metric, the better. [2] Bold indicates the best results. [3] Underlined indicates the second-best results.

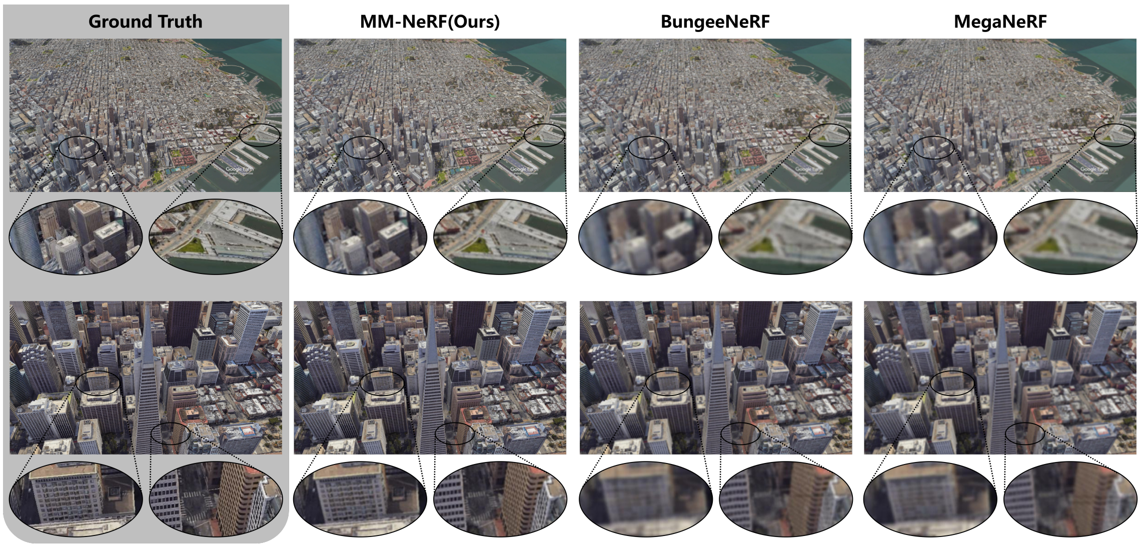

Figure 4. A qualitative comparison between our method and others. The MLP-based methods (BungeeNeRF and Mega-NeRF) suffer from severe blurring in different distances of views. Our method achieves a photorealistic quality at novel views compared to ground-truth images.

Table 3. Quantitative comparison on UrbanScene3D dataset. We report PSNR(↑), LPIPS(↑), and SSIM(↓) metrics on the test view. We highlighted the best and second-best results.

	UrbanScene3D-Campus			UrbanScene3D-Residence			UrbanScene3D-Sci-Art		
	PSNR↑ [1]	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑
NeRF [5]	21.276	0.357	0.579	20.937	0.415	0.528	21.104	0.456	0.580
Mip-NeRF [18]	21.322	0.298	0.607	21.193	0.394	0.585	21.284	0.418	0.542
DVGO [11]	22.105	0.254	0.643	21.919	0.344	0.628	22.312	0.427	0.629
TensoRF [45]	22.683	0.228	0.689	22.563	0.270	0.680	22.425	0.337	0.618
Mega-NeRF [8]	23.417 [3]	0.171	0.751	22.468	0.243	0.673	22.861	0.244	0.711
BungeeNeRF [6]	22.917	0.189	0.722	22.342	0.285	0.598	22.632	0.308	0.620
MM-NeRF (ours)	**24.126** [2]	**0.158**	**0.807**	**23.514**	**0.164**	**0.757**	**23.965**	**0.166**	**0.802**

[1] The upward arrow indicates that the higher the metric, the better. The downward arrow indicates that the lower the metric, the better. [2] Bold indicates the best results. [3] Underlined indicates the second-best results.

We compared MM-NeRF with other methods in the mentioned scenes. Unlike other NeRF methods that lack 3D grid features for explicit geometry learning, MM-NeRF avoids local geometric deformation issues. For instance, in row 2 of Figure 4, BungeeNeRF and MegaNeRF exhibit misaligned building exterior walls. This error is even more noticeable in the enlarged view of a street lamp in row 3 of Figure 6. In addition, due to large-scale scenes with limited views and substantial view differences, methods without a priors feature input struggle to synthesize new view RGB values, resulting in numerous artifacts, especially in complex texture areas (e.g., Figure 5, rows 2 and 3).

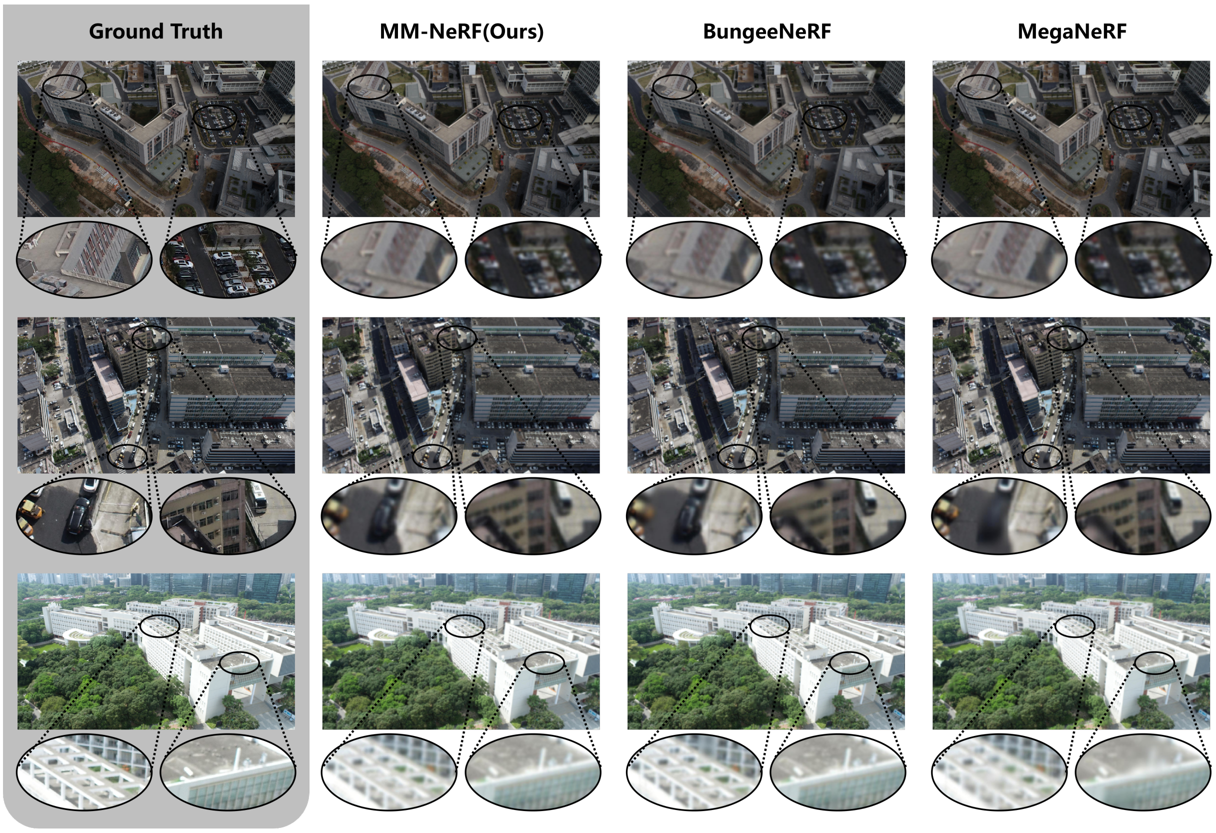

Figure 5. Qualitative comparison shows that our method achieves better visual quality and is more photorealistic in three UrbanScene3D scenes.

Table 4. Quantitative comparison on our real-world scenes dataset. We report PSNR(↑), LPIPS(↑), and SSIM(↓) metrics on the test view. We highlighted the best and second-best results.

	AIM (Avg.)			YRT (Avg.)			MAC (Avg.)			GXM (Avg.)		
	PSNR↑ [1]	**LPIPS↓**	**SSIM↑**	**PSNR↑**	**LPIPS↓**	**SSIM↑**	**PSNR↑**	**LPIPS↓**	**SSIM↑**	**PSNR↑**	**LPIPS↓**	**SSIM↑**
NeRF [5]	21.390	0.259	0.666	21.577	0.223	0.603	22.580	0.193	0.701	20.976	0.279	0.523
Mip-NeRF [18]	22.257	0.202	0.696	21.624	0.241	0.650	22.518	0.199	0.710	22.976	0.183	0.714
DVGO [11]	22.190	0.227	0.629	21.997	0.242	0.655	23.140	0.188	0.723	23.428	0.177	0.760
TensoRF [45]	22.374	0.211	0.729	22.224	0.189	0.715	23.304	0.177	0.731	23.576	0.169	0.784
Mega-NeRF [8]	22.612	0.172	0.769	22.641	0.209	0.677	23.381	0.174	0.726	24.316	0.156	0.807
BungeeNeRF [6]	22.955 [3]	0.185	0.716	23.525	**0.147**	0.774	23.465	0.167	0.742	24.119	0.161	0.814
MM-NeRF (ours)	**24.125** [2]	**0.152**	**0.834**	**24.872**	0.150	**0.801**	**24.322**	**0.133**	**0.884**	**25.149**	**0.137**	**0.844**

[1] The upward arrow indicates that the higher the metric, the better. The downward arrow indicates that the lower the metric, the better. [2] Bold indicates the best results. [3] Underlined indicates the second-best results.

Contrastingly, MM-NeRF's grid features focus on geometry learning, while the prior feature branch encodes texture space. Utilizing both as additional inputs ensures accurate and consistent rendering, yielding precise geometry and detailed texture colors. Row 2 of Figure 6 illustrates MM-NeRF's ability to restore detailed information in distant, dense buildings, significantly enhancing the rendering quality for complex areas.

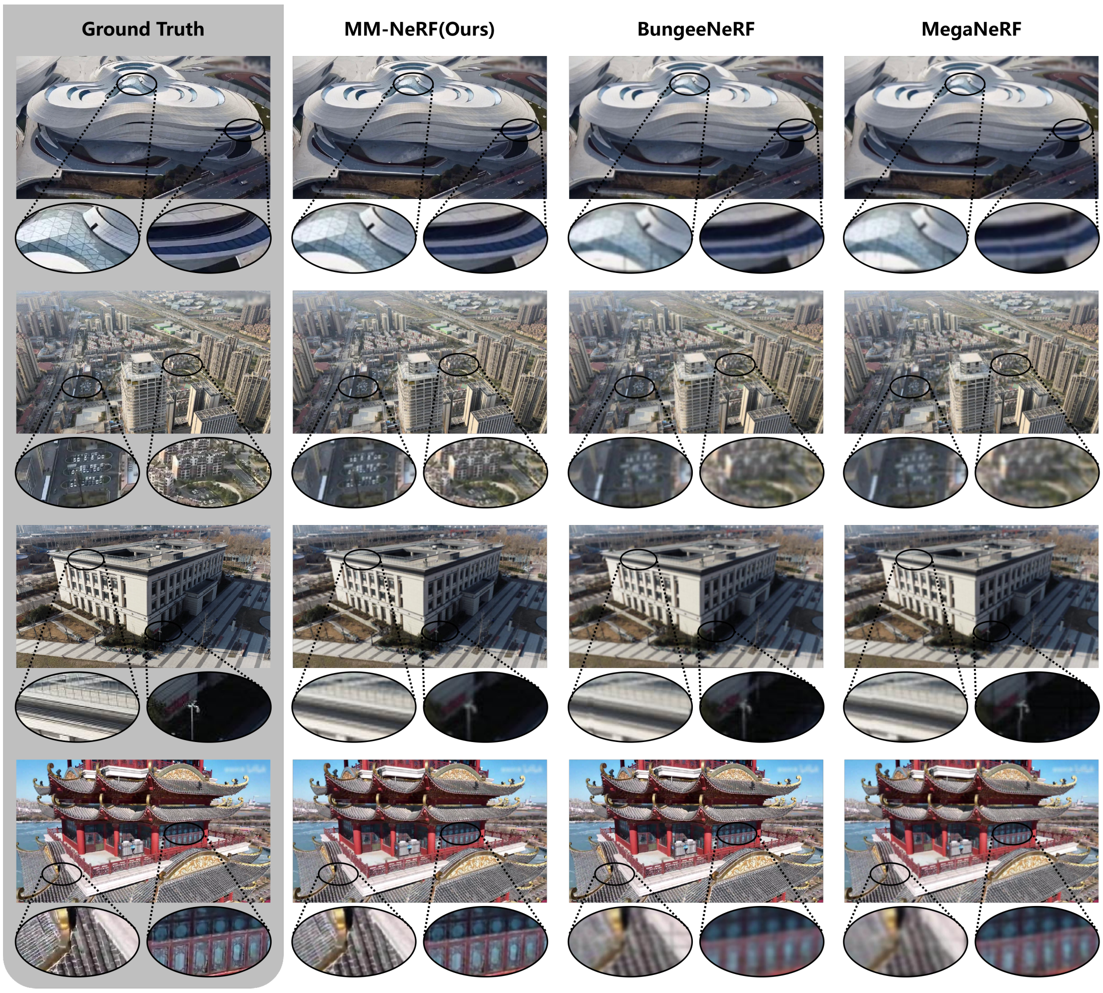

Figure 6. Qualitative comparisons show that our method still performs best on the four real-world scene datasets we created.

4.3. Ablation

To validate the effectiveness of our network, we conducted ablation experiments on Google scenes. First, we proved the impact of the resolution hash grid (Table 5). Further, we studied the effect of the hash grid resolution parameter L (Table 6). Then, we analyzed the impact of multi-view prior features (Table 7). In addition, we also explored the effect of the number of reference views on multi-view prior features (Table 8).

In Table 5, we performed ablation experiments on multi-resolution hash grid features. Figure 7 clearly shows that our proposed multi-resolution hash grid branch can match local scene content explicitly and accurately. Experiments show that NeRFs can benefit from the local features encoded in grid features, and the PSNR is improved by about 1 dB. This result confirms the effectiveness of our multi-resolution hash features in improving the quality of radiation field rendering.

Table 5. Comparison with and without multi-resolution grid feature on Google scenes.

	PSNR↑ [1]	LPIPS↓	SSIM↑
Without multi-resolution grid feature	22.947	0.261	0.599
With multi-resolution grid feature	23.879	0.205	0.735

[1] The upward arrow indicates that the higher the metric, the better. The downward arrow indicates that the lower the metric, the better.

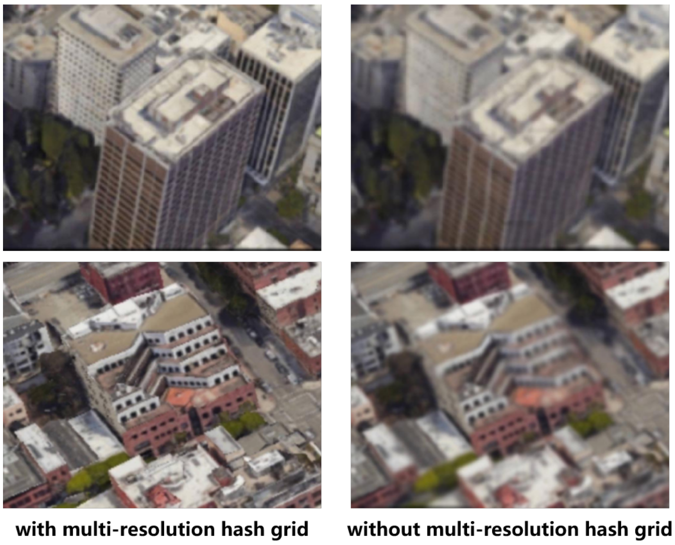
with multi-resolution hash grid
without multi-resolution hash grid

Figure 7. Visualization of branches with/without multi-resolution grid. Adding multi-resolution grid can render more details.

To enhance the stability of multi-resolution hash grids, we explore the impact of different resolution grids on the results. Table 6 illustrates the impact of the grid resolution on rendering results. We find that a higher grid resolution does not necessarily lead to better results, as convergence issues may arise with an increasing resolution. During our experiments, the optimal resolution was $L = 16$, which can better balance the rendering quality of the training time.

Table 6. Impact of different hash resolution grid settings on Google scenes.

	PSNR↑ [1]	**LPIPS↓**	**SSIM↑**
$L = 2$	22.866	0.301	0.563
$L = 4$	22.890	0.289	0.616
$L = 8$	23.496	0.253	0.688
$L = 16$	23.879	0.205	0.735

[1] The upward arrow indicates that the higher the metric, the better. The downward arrow indicates that the lower the metric, the better.

To effectively utilize the features existing in multi-view images, we designed a multi-view prior feature branch. Experimental results confirm the benefits of adding multi-view prior features. The quantitative comparisons provided in Table 7 strongly support the superior performance achieved by integrating multi-view prior features into our method. There is a gain of approximately 1.5 dB in the PSNR.

Table 7. Comparison with and without multi-view prior feature on Google scenes.

	PSNR↑ [1]	**LPIPS↓**	**SSIM↑**
Without multi-view prior feature	22.518	0.288	0.629
With multi-view prior feature	24.079	0.193	0.791

[1] The upward arrow indicates that the higher the metric, the better. The downward arrow indicates that the lower the metric, the better.

Further, we explore the performance of the network under different numbers of reference views. Table 8 presents the quantitative results of this experiment, demonstrating that the more reference views within a certain angular range, the better the performance.

Table 8. Impact of different number of reference view settings within 120° viewing angle on Google scenes.

	PSNR↑ [1]	LPIPS↓	SSIM↑
$n = 1$	23.264	0.279	0.702
$n = 2$	23.613	0.255	0.691
$n = 3$	23.892	0.223	0.727
$n = 4$	23.950	0.214	0.751

[1] The upward arrow indicates that the higher the metric, the better. The downward arrow indicates that the lower the metric, the better.

5. Discussion

Previous NeRF methods [6–9] have certain limitations in large scene view synthesis, including insufficient detail and the need for a large amount of view source data. To address these challenges, we introduce a multi-resolution hash grid feature branch and a multi-view prior feature branch into the classic NeRF framework. The multi-view prior feature branch maximizes the ability to extract as much information as possible from 2D images and then uses multi-resolution grids with geometric properties for modeling, which improves the overall representation ability of large scene NeRF.

Our method can be applied well in real-world scenes. For example, in terms of virtual tourism, by creating realistic digital twins of attractions, users can learn about the destination through a virtual travel experience without actually going there. In the dataset we created, YRT is a tourist attraction, and using our method can generate realistic views from any angle, allowing users to freely tour the virtual environment. Furthermore, our method can be applied to the metaverse. Based on the current local area modeling, it can be expanded to the city level in the future to build a three-dimensional model of the entire city as a digital map of the metaverse.

In addition, since the multi-resolution hash grid stores explicit geometric information and texture feature characteristics, the mesh and texture can be generated by combining classic algorithms (e.g., Marching cubes [46]) and 3D tools (e.g., Xatlas [47]), which can integrate well with existing 3D rendering pipelines, expanding their use in downstream applications.

6. Conclusions

The method we propose, MM-NeRF, represents an advancement in the field of large-scale scene modeling for NeRF. Previous methods of handling large-scale reconstructions often employed divide-and-conquer strategies or increased the network size. In contrast, we propose a novel architecture that integrates efficient hybrid feature input based on the NeRF architecture, including 3D mesh features based on explicit modeling and scene priors obtained from multi-views. The injection of these mixed features into the NeRF network brings supplementary information, which makes up for the limitations of the general NeRF, such as low fitting and insufficient refinement due to the sparsity of large scene views. We addressed several key challenges and made several contributions:

(1) We combined an MLP-based NeRF with explicitly constructed feature grids and introduced a multi-resolution hash grid feature branch to effectively encode local and global scene information, significantly improving the accuracy of large-scale scene modeling.

(2) We noticed that previous NeRF methods do not fully utilize the potential of multi-view prior information. We designed a view encoder to extract and integrate features from multiple views to obtain better results.

Despite the fact that our proposed method improves the rendering quality to a certain extent, our model inherits some limitations of NeRF-based methods:

(1) A slow training phase: although hash mapping is faster than MLP queries, the entire system requires more training epochs (about 200–300 epochs for different scenes) since the other feature branch has a more complex encoder structure.

(2) Handling a large number of high-resolution images: we adopt the existing mixed-ray batch sampling method for training, which is very inefficient without distributed training.

In conclusion, we propose a new variant of optimized NeRF, MM-NeRF, specifically designed for large-scale scene modeling, which takes a step forward in solving the challenges of the large-scene NeRF. MM-NeRF combines a multi-resolution hash grid and cross-view prior feature acquisition to solve the problems of previous methods that are not precise enough in large scenes and rely on a large number of views. But MM-NeRF can be further explored and improved. For example, by capturing and modeling dynamic objects in a scene or exploring the use of prompts to enable controllable view synthesis, these studies could help improve the overall usability of NeRF in large scenes.

Author Contributions: Conceptualization, B.D., K.C., Z.W. and J.G.; investigation and analysis, B.D. and K.C.; resources, M.Y. and X.S.; software, B.D.; validation, B.D., J.G. and K.C.; visualization, B.D.; writing—original draft preparation, B.D. and K.C.; writing—review and editing, Z.W., M.Y. and X.S.; supervision, M.Y. and X.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Key R&D Program of China (2022ZD0118402) and the National Nature Science Foundation of China under Grant 62331027 and Grant 62076241. (Corresponding author: Xian Sun).

Data Availability Statement: In this paper, the Google scenes dataset was downloaded from BungeeNeRF homepage (https://city-super.github.io/citynerf/, accessed on 21 February 2024), and the UrbanScene3D dataset was downloaded from UrbanScene3D dataset homepage (https://vcc.tech/UrbanScene3D, accessed on 21 February 2024). In addition, data collected by the authors are available on request from the corresponding author (accurate declaration of purpose).

Conflicts of Interest: Author M.Y. was employed by the company Jigang Defence Technology Company, Ltd. and Cyber Intelligent Technology (Shandong) Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Abbreviations

The following abbreviations are used in this paper:

NeRF	Nerual Radiance Field
MLP	Multi-Layer Perception
CNN	Convolutional Neural Network
PE	Position Encoding
PSNR	Peak Signal-to-Noise Ratio
SSIM	Structural Similarity
LPIPS	Learned Perceptual Image Patch Similarity

References

1. Rudnicka, Z.; Szczepanski, J.; Pregowska, A. Artificial Intelligence-Based Algorithms in Medical Image Scan Segmentation and Intelligent Visual Content Generation—A Concise Overview. *Electronics* **2024**, *13, 746*. [CrossRef]
2. Mhlanga, D. Industry 4.0 in Finance: The Impact of Artificial Intelligence (AI) on Digital Financial Inclusion. *Int. J. Financ. Stud.* **2020**, *8, 45*. [CrossRef]
3. Zhang, J.; Huang, C.; Chow, M.Y.; Li, X.; Tian, J.; Luo, H.; Yin, S. A Data-Model Interactive Remaining Useful Life Prediction Approach of Lithium-Ion Batteries Based on PF-BiGRU-TSAM. *IEEE Trans. Ind. Inform.* **2024**, *20*, 1144–1154. [CrossRef]
4. Zhang, J.; Tian, J.; Yan, P.; Wu, S.; Luo, H.; Yin, S. Multi-hop graph pooling adversarial network for cross-domain remaining useful life prediction: A distributed federated learning perspective. *Reliab. Eng. Syst. Saf.* **2024**, *244*, 109950. [CrossRef]
5. Mildenhall, B.; Srinivasan, P.P.; Tancik, M.; Barron, J.T.; Ramamoorthi, R.; Ng, R. NeRF: Representing scenes as neural radiance fields for view synthesis. *Commun. ACM* **2021**, *65*, 99–106. [CrossRef]

6. Xiangli, Y.; Xu, L.; Pan, X.; Zhao, N.; Rao, A.; Theobalt, C.; Dai, B.; Lin, D. BungeeNeRF: Progressive Neural Radiance Field for Extreme Multi-scale Scene Rendering. In Proceedings of the Computer Vision—ECCV 2022, Tel Aviv, Israel, 23–27 October 2022; pp. 106–122.
7. Zhang, X.; Bi, S.; Sunkavalli, K.; Su, H.; Xu, Z. NeRFusion: Fusing Radiance Fields for Large-Scale Scene Reconstruction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 5449–5458.
8. Tancik, M.; Casser, V.; Yan, X.; Pradhan, S.; Mildenhall, B.; Srinivasan, P.P.; Barron, J.T.; Kretzschmar, H. Block-NeRF: Scalable Large Scene Neural View Synthesis. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 8248–8258.
9. Turki, H.; Ramanan, D.; Satyanarayanan, M. Mega-NERF: Scalable Construction of Large-Scale NeRFs for Virtual Fly-Throughs. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 12922–12931.
10. Fridovich-Keil, S.; Yu, A.; Tancik, M.; Chen, Q.; Recht, B.; Kanazawa, A. Plenoxels: Radiance Fields without Neural Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 5501–5510.
11. Sun, C.; Sun, M.; Chen, H.T. Direct Voxel Grid Optimization: Super-Fast Convergence for Radiance Fields Reconstruction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 5459–5469.
12. Müller, T.; Evans, A.; Schied, C.; Keller, A. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. *ACM Trans. Graph.* **2022**, *41*, 1–15. [CrossRef]
13. Yu, A.; Ye, V.; Tancik, M.; Kanazawa, A. pixelNeRF: Neural Radiance Fields From One or Few Images. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Online, 19–25 June 2021; pp. 4578–4587.
14. 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 Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Online, 19–25 June 2021; pp. 4690–4699.
15. Trevithick, A.; Yang, B. GRF: Learning a General Radiance Field for 3D Representation and Rendering. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Online, 11–17 October 2021; pp. 15182–15192.
16. Chibane, J.; Bansal, A.; Lazova, V.; Pons-Moll, G. Stereo Radiance Fields (SRF): Learning View Synthesis for Sparse Views of Novel Scenes. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Online, 19–25 June 2021; pp. 7911–7920.
17. Kajiya, J.T.; Von, H.B.P. Ray tracing volume densities. *ACM SIGGRAPH Comput. Graph.* **1984**, *18*, 165–174. [CrossRef]
18. 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 Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Online, 11–17 October 2021; pp. 5855–5864.
19. Verbin, D.; Hedman, P.; Mildenhall, B.; Zickler, T.; Barron, J.T.; Srinivasan, P.P. Ref-NeRF: Structured View-Dependent Appearance for Neural Radiance Fields. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 5481–5490.
20. Kai, Z.; Gernot, R.; Noah, S.; Vladlen, K. NeRF++: Analyzing and Improving Neural Radiance Fields. *arXiv* **2020**, arXiv:2010.07492.
21. Garbin, S.J.; Kowalski, M.; Johnson, M.; Shotton, J.; Valentin, J. FastNeRF: High-Fidelity Neural Rendering at 200FPS. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Online, 11–17 October 2021; pp. 14346–14355.
22. Reiser, C.; Peng, S.; Liao, Y.; Geiger, A. KiloNeRF: Speeding Up Neural Radiance Fields With Thousands of Tiny MLPs. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Online, 11–17 October 2021; pp. 14335–14345.
23. Wadhwani, K.; Kojima, T. SqueezeNeRF: Further Factorized FastNeRF for Memory-Efficient Inference. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, New Orleans, LA, USA, 18–24 June 2022; pp. 2717–2725.
24. Chen, A.; Xu, Z.; Zhao, F.; Zhang, X.; Xiang, F.; Yu, J.; Su, H. MVSNeRF: Fast Generalizable Radiance Field Reconstruction From Multi-View Stereo. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Online, 11–17 October 2021; pp. 14124–14133.
25. Jain, A.; Tancik, M.; Abbeel, P. Putting NeRF on a Diet: Semantically Consistent Few-Shot View Synthesis. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Online, 11–17 October 2021; pp. 5885–5894.
26. Yen-Chen, L.; Florence, P.; Barron, J.T.; Rodriguez, A.; Isola, P.; Lin, T.Y. iNeRF: Inverting Neural Radiance Fields for Pose Estimation. In Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 28–30 September 2021; pp. 1323–1330.
27. Lin, C.H.; Ma, W.C.; Torralba, A.; Lucey, S. BARF: Bundle-Adjusting Neural Radiance Fields. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Online, 11–17 October 2021; pp. 5741–5751.
28. Zirui, W.; Shangzhe, W.; Weidi, X.; Min, C.; Victor, A.P. NeRF–: Neural Radiance Fields without Known Camera Parameters. *arXiv* **2022**, arXiv:2102.07064.
29. Pumarola, A.; Corona, E.; Pons-Moll, G.; Moreno-Noguer, F. D-NeRF: Neural Radiance Fields for Dynamic Scenes. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Online, 19–25 June 2021; pp. 10318–10327.

30. Fridovich-Keil, S.; Meanti, G.; Warburg, F.R.; Recht, B.; Kanazawa, A. K-Planes: Explicit Radiance Fields in Space, Time, and Appearance. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 18–22 June 2023; pp. 12479–12488.
31. Niemeyer, M.; Geiger, A. GIRAFFE: Representing Scenes As Compositional Generative Neural Feature Fields. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Online, 19–25 June 2021; pp. 11453–11464.
32. Mirzaei, A.; Aumentado-Armstrong, T.; Derpanis, K.G.; Kelly, J.; Brubaker, M.A.; Gilitschenski, I.; Levinshtein, A. SPIn-NeRF: Multiview Segmentation and Perceptual Inpainting With Neural Radiance Fields. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 18–22 June 2023; pp. 20669–20679.
33. Wang, C.; Chai, M.; He, M.; Chen, D.; Liao, J. CLIP-NeRF: Text-and-Image Driven Manipulation of Neural Radiance Fields. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 3835–3844.
34. Martin-Brualla, R.; Radwan, N.; Sajjadi, M.S.M.; Barron, J.T.; Dosovitskiy, A.; Duckworth, D. NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Online, 19–25 June 2021; pp. 7210–7219.
35. Xu, L.; Xiangli, Y.; Peng, S.; Pan, X.; Zhao, N.; Theobalt, C.; Dai, B.; Lin, D. Grid-Guided Neural Radiance Fields for Large Urban Scenes. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 18–22 June 2023; pp. 8296–8306.
36. Yuqi, Z.; Guanying, C.; Shuguang, C. Efficient Large-scale Scene Representation with a Hybrid of High-resolution Grid and Plane Features. *arXiv* **2023**, arXiv:2303.03003.
37. Johari, M.M.; Lepoittevin, Y.; Fleuret, F. GeoNeRF: Generalizing NeRF With Geometry Priors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 18365–18375.
38. Liu, Y.; Peng, S.; Liu, L.; Wang, Q.; Wang, P.; Theobalt, C.; Zhou, X.; Wang, W. Neural Rays for Occlusion-Aware Image-Based Rendering. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 7824–7833.
39. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.
40. Xu, H.; Zhang, J.; Cai, J.; Rezatofighi, H.; Tao, D. GMFlow: Learning Optical Flow via Global Matching. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 8121–8130.
41. Lin, L.; Liu, Y.; Hu, Y.; Yan, X.; Xie, K.; Huang, H. Capturing, Reconstructing, and Simulating: The UrbanScene3D Dataset. In Proceedings of the Computer Vision—ECCV 2022, Tel Aviv, Israel, 23–27 October 2022; pp. 93–109.
42. Schonberger, J.L.; Frahm, J.M. Structure-From-Motion Revisited. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 4104–4113.
43. Sitzmann, V.; Zollhoefer, M.; Wetzstein, G. Scene Representation Networks: Continuous 3D-Structure-Aware Neural Scene Representations. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS) 32, Vancouver, BC, Canada, 8–14 December 2019; Volume 32.
44. Zhang, R.; Isola, P.; Efros, A.A.; Shechtman, E.; Wang, O. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 19–21 June 2018; pp. 586–595.
45. Chen, A.; Xu, Z.; Geiger, A.; Yu, J.; Su, H. TensoRF: Tensorial Radiance Fields. In Proceedings of the Computer Vision—ECCV 2022, Tel Aviv, Israel, 23–27 October 2022; pp. 106–122.
46. Lorensen, W.E.; Cline, H.E. Marching cubes: A high resolution 3D surface construction algorithm. In *Seminal Graphics: Pioneering Efforts That Shaped the Field*; ACM SIGGRAPH: Chicago, IL, USA, 1998; pp. 347–353.
47. Xatlas. Available online: https://github.com/jpcy/xatlas (accessed on 3 February 2024).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.