2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

MMPI: a Flexible Radiance Field Representation by Multiple Multi-plane Images Blending

Yuze He[1], Peng Wang[2], Yubin Hu[1], Wang Zhao[1], Ran Yi[3], Yong-Jin Liu[1*], *Senior Member, IEEE* and Wenping Wang[4], *Fellow, IEEE*

***Abstract*— This paper presents a flexible representation of neural radiance fields based on multi-plane images (MPI), for high-quality view synthesis of complex scenes. MPI with Normalized Device Coordinate (NDC) parameterization is widely used in NeRF learning for its simple definition, easy calculation, and powerful ability to represent unbounded scenes. However, existing NeRF works that adopt MPI representation for novel view synthesis can only handle simple forward-facing unbounded scenes (e.g., the scenes in the LLFF dataset), where the input cameras are all observing in similar directions with small relative translations. Hence, extending these MPI-based methods to more complex scenes like large-range or even 360-degree scenes is very challenging. In this paper, we explore the potential of MPI and show that MPI can synthesize high-quality novel views of complex scenes with diverse camera distributions and view directions, which are not only limited to simple forward-facing scenes. Our key idea is to encode the neural radiance field with multiple MPIs facing different directions and blend them with an adaptive blending operation. For each region of the scene, the blending operation gives larger blending weights to those advantaged MPIs with stronger local representation abilities while giving lower weights to those with weaker representation abilities. Such blending operation automatically modulates the multiple MPIs to appropriately represent the diverse local density and color information. Experiments on the KITTI dataset and ScanNet dataset demonstrate that our proposed MMPI synthesizes high-quality images from diverse camera pose distributions and is fast to train, outperforming the previous fast-training NeRF methods for novel view synthesis. Moreover, we show that MMPI can encode extremely long trajectories and produce novel view renderings, demonstrating its potential in applications like autonomous driving. Our demo video is available at https://youtube.com/watch?v=mbNKwN5urC8.**

I. INTRODUCTION

Novel view synthesis (NVS) is a long-standing research problem and has been continuously studied over the past decades, with numerous practical applications such as autonomous driving, virtual reality, etc. For the past years, neural radiance field (NeRF) [1], [2], [3] has proven to be a powerful tool to model the 3D scenes for the task of novel view synthesis. Typically, a NeRF model represents a 3D scene by volume densities and view-dependent emissive colors, which is trained by differentiable volume rendering and supervised by the pixel colors of the input posed images. Once training is done, NeRF enables synthesizing photorealistic images from arbitrary viewpoints.

Training a neural radiance field usually requires predefining a bound where the scene is represented. For the rendering of unbounded scenes, a popular strategy is to re-parameterize the unbounded world space to a bounded space, e.g., [1], [4], [5]. Among these methods, a widely used technique is Normalized Device Coordinate (NDC) re-parameterization with the multi-plane images (MPI) data arrangement [6], [7]. The NDC re-parameterization maps an infinitely far view frustum to a unit cube and relocates NeRF's ability to make it consistent with the perspective cameras [5], [1]. MPI benefits from the provided additional depth constraint, and converges more easily than those data structures without a fixed depth, which allows a larger depth range to be modeled. So far, despite of its powerful representation ability, MPI with NDC mapping is only suitable for simple forward-facing scenes, which assumes all the cameras look along with a similar view direction, and the translations between different cameras are small. The reason for this limitation is that NDC needs to predefine a view frustum for calculating the mapping, and if a new camera frustum's orientation or translation is far off the predefined one, the mapping is unreliable, leading to severe degradation of the synthesized image quality.

In this paper we address the problem: is it possible to extend MPI to render more complex scenes, e.g., large-range scenes or even 360-degree scenes? We explore the potential of MPI and present a novel solution to this problem. We propose Multiple Multi-plane Images (MMPI), a flexible representation of neural radiance fields to synthesize high-quality images of complex scenes. As the name indicates, we encode the scene with a set of multiple MPIs. By properly organizing and arranging the positions and orientations of those MPIs, our representation is able to cover a wide, unbounded range of the scene of interest. Based on this representation, we propose a reliability field for each MPI to blend the sampled colors and densities for a proper volume rendering. We further propose a two-stage reliability learning

This work was partially supported by Beijing Natural Science Foundation (L222008), the Natural Science Foundation of China (62332019, 62302297, U2336214), Shanghai Sailing Program (22YF1420300), Young Elite Scientists Sponsorship Program by CAST (2022QNRC001), and Beijing Hospitals Authority Clinical Medicine Development of special funding support (ZLRK202330).

[1]Y. He, Y. Hu, W. Zhao, and Y.-J. Liu are with the Department of Computer Science and Technology, MOE-Key Laboratory of Pervasive Computing, Tsinghua University, China {hyz22, huyb20, zhao-w19}@mails.tsinghua.edu.cn, liuyongjin@tsinghua.edu.cn

[2]P. Wang is with the Department of Computer Science, The University of Hong Kong, Hong Kong pwang3@cs.hku.hk

[3]R. Yi is with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, China ranyi@sjtu.edu.cn

[4]W. Wang is with the Department of Computer Science and Engineering, Texas A&M University, USA wenping@tamu.edu

*Corresponding Author

scheme to effectively train the MPI and its reliability field. At the first stage each MPI is trained individually, after that all the MPIs are then jointly trained using an adaptive blending technique to learn their reliability at each spatial position. The adaptive blending gives larger weights for advantaged MPIs with better local representation abilities and lower weights for MPIs with less representation abilities. In this way, the MPIs collaborate with each other and further increase the rendering quality.

We have conducted extensive experiments on the KITTI dataset and ScanNet dataset. Experimental results demonstrate that MMPI is fast to train (about 40 minutes to converge on a large range scene on a single Nvidia 3090 GPU), and achieves state-of-the-art novel view synthesis quality among fast-training NeRF methods.

II. RELATED WORKS

Fast Neural Radiance Fields. The original NeRF model [1] is encoded by a multi-layer perceptron (MLP), which usually takes hours or days to converge due to the complex optimization of the deep model and is slow to render an image. Some works focused on accelerating the rendering speed of NeRFs and achieved real-time rendering by baking or distilling from a pre-trained NeRF model [8], [9], [10] but are still slow. Instead of training a deep neural network, recent works showed that the training process can be greatly accelerated by direct optimization of voxels [11], [8], neural hash grids [12], [3], [13] or tensor decomposition [14], [15]. In this paper, following DVGO [11] we use MPI with voxel representation to encode the density and colors for speeding up the training process.

NeRFs for unbounded scenes. Typically, a NeRF model is only able to encode bounded scenes. To render unbounded scenes, previous works adopt different space parameterization methods to map an unbounded scene to a bounded scene. A widely used one is Normalized Device Coordinate (NDC) mapping with multi-plane images (MPI). NDC maps an unbounded view frustum to a bounded cube [1], [8], [14], [11], which can suitably represent a forward-facing scene and is easy to compute. For unbounded 360-degree scenes, some recent works proposed several reparameterization methods [5], [8], [4], [16], which share a similar idea that maps an infinitely large spherical space to a bounded sphere space. More recently, MERF [17] proposed a line-preserving mapping for efficient rendering of 360-degree unbounded scenes, and F2-NeRF [18] proposed an adaptive space reparameterization method called perspective warping for free trajectories. MPI is an efficient 3D scene representation format consisting of L image planes located at a set of fixed depths [19], [20], [21], [22], [23], [24], which has been demonstrated to perform well on forward-facing scenes due to the additional depth constraint, and converges more easily than other alternatives without fixed discretized depths. In this work, we use NDC mapping for its simple definition and easy calculation, and we propose Multiple MPI blending with an adaptive blending operation to render complex unbounded scenes. Nex360 [25] also used multiple MPIs for rendering 360-degree scenes. However, it conducts the blending operation on 2D image space and is slow to train (up to 20 hours), while our method blends the MPIs directly on the 3D space and has a significantly fast convergence speed.

NeRFs for large-scale scenes. Since the emergence of NeRF, recent works have tried to extend NeRF to large-scale scenes, which usually requires training a large number of NeRF models and composing those models for large-scale scene representation [26], [27], [28], [29]. Our method is orthogonal and compatible with such scene composition methods and can be possibly extended to large-scale scenes.

III. METHOD

Our goal is to synthesize high-quality images from novel views given several posed images of unbounded scenes as supervision. We propose Multiple Multi-plane Image Blending (MMPI), a new scene representation based on NeRF [1] for novel view synthesis of complex unbounded scenes. We encode the scene with multiple multi-plane images and design an adaptive blending strategy based on a learnable reliability grid.

In this section, we first introduce the preliminaries of the method in Sec. III-A, which includes the basic NeRF pipeline, voxel grid and MPI scene representations. In Sec. III-B, we introduce our Multi-MPI blending operation and provide an efficient method for fused training and rendering of multiple MPIs. Sec. III-C describes how to improve the rendering effect of Multi-MPI Blending by learning the per-voxel reliability. Moreover, Sec. III-D addresses and overcomes the inherent drawbacks of the MPI format by blending with an extra centered voxel grid.

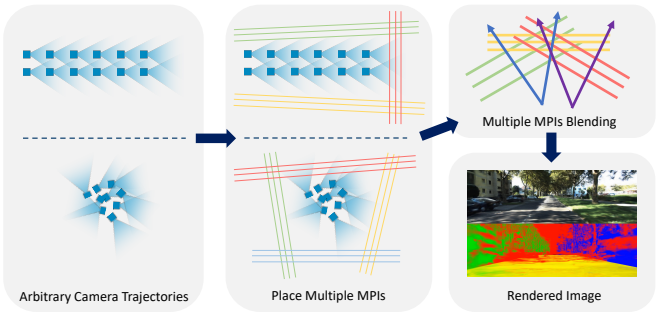

Fig. 1. An overview of our proposed MMPI pipeline. We utilize multiple MPIs facing different directions and jointly render them for novel view synthesis. Our MMPI method support blending and rendering any number of arbitrarily located MPI grids to support a wide variety of camera trajectories.

A. Preliminaries

Neural Radiance Fields. Generally, Neural Radiance Fields (NeRF) represent a 3D scene by spatial-variant volume densities with spatial and view-direction-variant emissive colors, which can be modeled as a learnable function F_Θ that takes the 3-dimensional location of a sampled point $\mathbf{x} = (x; y; z)$ and a 2-dimensional viewing direction $\mathbf{d} = (\theta; \phi)$ as inputs, and outputs density σ and color c:

$$(\sigma, c) = F_\Theta(\mathbf{x}, \mathbf{d}). \tag{1}$$

When rendering a pixel color from a ray of view, a volume-rendering-like formula is employed that involves marching along the ray to determine the color of a pixel $\hat{C}(\mathbf{r})$. In the ray marching process, a set of 3D points is sampled along the ray and the synthesized pixel color is integrated by the volume rendering equation from the sampled densities σ_i and colors c_i by:

$$\hat{C}(\boldsymbol{r}) = \sum_{i=1}^{N} T_i(1 - \exp(-\sigma_i \delta_i))c_i,$$
$$\text{where } T_i = \exp\left(-\sum_{j=1}^{i-1} \sigma_j \delta_j\right) \tag{2}$$

where $\delta_i = t_{i+1} - t_i$ is the distance between adjacent samples. This rendering process is differentiable, and therefore, the model can be optimized by minimizing the difference between the observed and rendered colors.

Voxel Grid Representation. To achieve faster convergence, we follow DVGO [11] and incorporate an explicit voxel grid for modeling 3D information in the scene representation. Different from traditional NeRF architecture, which uses MLP to predict density from 3D coordinates, we construct a learnable 3D density voxel grid $\mathbf{V}_\sigma$ with a voxel number of $N_x \times N_y \times N_z$. We calculate the density $\sigma(\mathbf{x})$ at a particular point $\mathbf{x}$ through trilinear interpolation of 3D coordinates:

$$\sigma(\mathbf{x}) = interp(\mathbf{x}, \mathbf{V}_\sigma), \tag{3}$$
$$interp(\mathbf{x}, \mathbf{V}_\sigma) : (R^3, R^{1\times N_x \times N_y \times N_z}) \to \mathbb{R}, \tag{4}$$

To acquire color information, we construct a D-channel learnable 3D feature voxel grid $\mathbf{V}_{feat}$ using an explicit-implicit hybrid representation in order to achieve efficient and view-dependent rendering:

$$\mathcal{F}(\mathbf{x}) = interp(\mathbf{x}, \mathbf{V}_{feat}), \tag{5}$$
$$interp(\mathbf{x}, \mathbf{V}_{feat}) : (R^3, R^{D\times N_x \times N_y \times N_z}) \to \mathbb{R}^D, \tag{6}$$

where $\mathcal{F}(\mathbf{x})$ contains the color information for different viewing angles. The final color is determined by the feature $\mathcal{F}(\mathbf{x})$, the 3D coordinates $\mathbf{x}$ and the viewing-direction $\mathbf{d}$ together by passing a shallow MLP network $MLP_\Theta^{(rgb)}$:

$$c(\mathbf{x}, \mathbf{d}) = MLP_\Theta^{(rgb)}(\mathcal{F}(\mathbf{x}), \mathbf{x}, \mathbf{d}), \tag{7}$$

where c is the view-dependent color emission. Similar to NeRF, we also use the positional encoding strategy for $\mathbf{x}$ and $\mathbf{d}$ as additional input to MLP.

MPI Representation. The voxel grid representation is limited in its ability to accurately represent unbounded 3D scenes, as objects that are located beyond the range of the grids cannot be properly modeled. To address this issue, we adopt the approach proposed by DVGOv2 [7], which reorganizes the voxel grid to the Multi-Plane Images (MPI) format, with a collection of L RGB-density image planes at fixed depths. By linearly sampling disparity from the depth of a specified near plane to ∞ in the original space, we place a learnable image plane at each sampling depth. These planes collectively form a MPI, which allows us to address the challenge of representing infinitely far objects. We leverage the same parameterization method as NeRF to warp the forward-facing scene into the normalized device coordinate (NDC) space. Then, we evenly select several image planes within the $z \in [-1, 1]$ range in the NDC space to achieve the desired effect above.

The representations for density and color features are essentially similar to the voxel grid mentioned previously. However, for obtaining density and feature by coordinates, interpolation is only carried out along the x and y axes, and not along the z-axis (when sampling points for each ray, only points located at the depth of the image planes are selected, and no other points are included).

B. Multiple MPI Blending

Multi-Plane Images (MPI) is an efficient 3D scene representation format consisting of L image planes, each located at a fixed depth d_i. Because of the additional depth constraint provided, MPI performs better for forward-facing scenes shot in nearly the same direction, and converges more easily than those data structures without a fixed depth, which allows a larger depth range to be modeled. However, the MPI data structure also has many inherent defects. First, MPI can only cover one direction of the scene; moreover, the depth of the image plane selected by MPI is discrete, but the depth of the scene, in reality, is continuous. When using MPI to model scenes with greater depth variation, no matter how the image planes are arranged, there will be a region with excessive depth gap, and the depth of the scene in this region cannot be correctly predicted.

Fig. 2. Illustration of the rendering quality when using different-facing MPI grids.

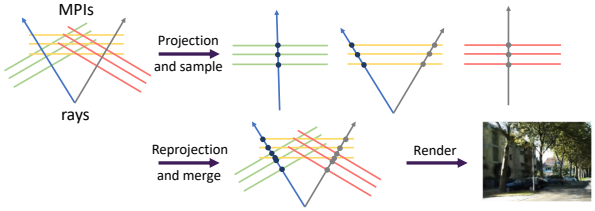

Fig. 3. An 2D illustration of our Multiple MPI blending pipeline. We first project camera rays to each MPI's NDC space, sample points and obtain their alpha and color values, and finally all the points are projected into the world coordinate system and merged by their distance to the camera.

Our MMPI method is based on a key observation that changing the MPI's orientation can result in better modeling of a specific region of the scene. For instance, in the scene of KITTI dataset as shown in Fig. 2, when we use the MPI facing the front of the scene, we find that the cars on the right side are blurrier, while when we use the MPI facing the right side of the scene, the image quality of the cars on the right

side becomes significantly better. This is because the depth change of the cars in the front and back direction is too big, and the cars fall into the sparse area of planes in the MPI facing the front of the scene; while the depth change in the left and right direction is smaller and many objects are closer, the cars fall into the dense area of planes in the MPI facing the right side of the scene. Similarly, small objects such as roadside poles and tree trunks are more easily modeled by MPI facing left or right sides; the surfaces of some buildings need to be predicted with continuously changing depths when using MPI facing the front of the scene, but only a constant depth needs to be predicted by MPI facing left or right sides, thus improving the quality of scene modeling.

Based on the above observation, a reasonable approach to improve the modeling quality of a scene is to leverage *multiple MPIs* facing different directions and render them jointly, which allows for the combination of their individual advantages in a cohesive manner. Thus we propose Multiple MPI Blending (MMPI), which supports any number of MPI grids with any orientation for mixed rendering.

We place K MPIs with different orientations in 3D space, each with a grid in its own NDC space, modeling all objects from the near plane to infinity. For a particular training viewpoint, we first generate a set of rays $\{\mathbf{r}_s\}$ based on all image pixels in that viewpoint. Then, for each ray $\mathbf{r}$ in the set, we need to obtain its intersection points with all MPI planes and arrange them in the order of their distance from the camera. This process can be done using traditional planar intersecstion algorithms (*e.g.*, DDA [30]), but it can significantly slow down the training and rendering process. Instead, we propose the "sample and merge" strategy (Fig. 3), which effectively improves the sampling speed.

"Sample and merge" strategy. For each ray $\mathbf{r} = (\mathbf{o}, \mathbf{d})$, we first project it into each MPI's NDC space and sample the corresponding MPI $\mathcal{M}_i$ to get a set of intersection points $\{p_j\}_i$. The projected ray direction is then denoted as $Proj_i(\mathbf{d})$. Due to the ambiguity of the z-axis direction in the NDC space transformation, we need to perform an additional forward-facing check to mask out the points with z-component less than zero (which are back to the current MPI $\mathcal{M}_i$) in the $Proj_i(\mathbf{d})$ orientation. Then we input $\{p_j\}_i$ and $Proj_i(\mathbf{d})$ into the density and feature grid of the corresponding MPI to obtain the density and color sets $\{\sigma_j\}_i$, $\{c_j\}_i$, and further calculate the alpha set $\{\alpha_j\}_i$ by:

$$\{\sigma_j\}_i = interp(\{p_j\}_i, \mathbf{V}_{\sigma,i}), \tag{8}$$

$$\{\mathcal{F}_j\}_i = interp(\{p_j\}_i, \mathbf{V}_{feat,i}), \tag{9}$$

$$\{c_j\}_i = MLP_{\Theta}^{(rgb)}(\{\mathcal{F}_j\}_i, \{p_j\}_i, Proj_i(\mathbf{d})), \tag{10}$$

$$\{\alpha_j\}_i = \{1 - \exp(-\sigma_j \delta_j)\}_i. \tag{11}$$

Then all point sets $\{p_j\}_i$ are projected to the original 3D space and sorted according to the distance from the camera to obtain an ordered set:

$$\{p_k, \alpha_k\}_{k=1}^{\sum_l^K n(\{p_j\}_l)}, \tag{12}$$

$$\text{where } ||Proj_a^{-1}(p_{k-1}) - \mathbf{o}||_2 < ||Proj_b^{-1}(p_k) - \mathbf{o}||_2,$$
$$p_{k-1} \in \{p_j\}_a,\ p_k \in \{p_j\}_b. \tag{13}$$

Then we render the pixel color by:

$$\hat{\mathbf{C}}(\mathbf{r}) = \sum_{k=1}^{\sum_l^K n(\{p_j\}_l)} T_k \alpha_k c_k, \quad \text{where } T_k = \prod_{j=1}^{k-1}(1 - \alpha_j). \tag{14}$$

The technique of Multi-MPI blending can be expanded to encompass unbounded 360-degree scenes like indoor scenes, where the entire 3D space can be modeled by setting up multiple MPIs and making their frustum range cover the entire 360-degree space.

C. Reliability Learning

When there is a large overlap among MPIs, training by directly blending them may result in some degree of degradation. One possible explanation for this is that MPIs with a sparse distribution of planes at certain locations tend to learn faster, whereas MPIs with a denser distribution of planes learn at a slower rate but have better modeling ability. When these MPIs are simultaneously trained, those that are learned first may impede the learning of other MPIs, preventing the finer-level learning of certain parts and leading to a decreased ability to model small objects.

To address the issue mentioned above, we propose per-voxel reliability learning that learns the relative confidence of multiple MPIs at each location in an end-to-end manner. The optimized reliability scores are utilized to merge MPIs by an adaptive blending operation, which gives larger blending weights to those advantaged MPIs with better local representation abilities and lower weights to those with less representation abilities.

Reliability grid and adaptive blending. We begin by creating a 1-channel reliability grid $\mathbf{V}_{r,i}$ for each MPI $\mathcal{M}_i$, which has the same size and resolution as the 3D density voxel grid in the NDC space where it is situated. For the i-th MPI $\mathcal{M}_i$, we obtain the reliability $\mathcal{R}_i(p)$ at a point p in its NDC through trilinear interpolation of 3D coordinates:

$$\mathcal{R}_i(p) = interp(p, \mathbf{V}_{r,i}), \tag{15}$$

$$interp(p, \mathbf{V}_{r,i}) : (R^3, R^{1 \times N_x \times N_y \times N_z}) \to \mathbb{R}. \tag{16}$$

For the intersection point p sampled at $\mathcal{M}_i$, to obtain its relative confidence in the j-th MPI $\mathcal{M}_j$, we project it to the NDC coordinate system where $\mathcal{M}_j$ is located, and obtain the corresponding reliability $\mathcal{R}_j(p)$ in $\mathcal{M}_j$ by:

$$\mathcal{R}_j(p) = interp(Proj_j(Proj_i^{-1}(p)), \mathbf{V}_{r,j}),$$
$$\text{where } j \neq i \tag{17}$$

Note that here the point p does not necessarily fall on the image plane of the target MPI $\mathcal{M}_j$ after two reprojections, but since reliability varies continuously, we use trilinear interpolation at this point, interpolating on all x, y, z axes. After that, the relative confidence $P_i(p)$ of point p over all reliabilities is calculated using the softmax function:

$$P_i(p) = \frac{e^{\mathcal{R}_i(p)}}{\sum_{j=1}^K e^{\mathcal{R}_j(p)}}. \tag{18}$$

Then the rendering formula Eq.(14) is updated to:

$$\hat{\mathbf{C}}(\mathbf{r}) = \sum_{k=1}^{\sum_l^K n(\{p_j\}_l)} T_k P_k \alpha_k c_k,$$
$$\text{where} \quad T_k = \prod_{j=1}^{k-1}(1 - P_j \alpha_j). \tag{19}$$

Two-stage reliability learning scheme. To learn the reliability grid, we propose a two-stage learning scheme. Specifically, in the first training stage, each MPI is trained individually without learning the reliability; after this stage, each MPI has received sufficient training and the expressive ability has been fully explored respectively. Then in the second stage, we jointly train all the MPIs and learn their reliability at each spatial position in an end-to-end manner.

D. Blending MMPI with centered voxel grid

The MPI data format has an inherent drawback: it cannot be effectively viewed from the opposite side and is subject to distortion when viewed from an angle nearly parallel to image planes. Consequently, regardless of the number of MPIs used for blending, it becomes challenging to model objects located in the center of the camera trajectory, thereby limiting the scope of application. To address this issue, we propose the extra use of a centered cube voxel grid $\mathcal{C}$ other than the existing MPIs, which represents the scene information in the area surrounding the camera trajectory and nearby regions. This cube grid is created directly in the world coordinate system and interpolates on the x, y, and z axes, in contrast to the MPI grid, which only interpolates on the x and y axes when obtaining density and color features.

In the rendering process, for each ray $\mathbf{r} = (\mathbf{o}, \mathbf{d})$, the point set $\{p_j\}_\mathcal{C}$ is obtained by sampling the cube grid $\mathcal{C}$ directly in the world coordinate system. Color and alpha set $\{c_j\}_\mathcal{C}$, $\{\alpha_j\}_\mathcal{C}$ are obtained by a process similar to that of MPI. Together with other MPI sampled point sets, they are merged and rendered according to the distance from the camera. Note that since the centered cube grid can only represent part of the scene information in one direction, it is not rendered separately, but always in combination with other MPIs.

E. Training loss

Our training loss is defined as:

$$\mathcal{L} = \mathcal{L}_{pho} + \lambda_{pt\_rgb}\mathcal{L}_{pt\_rgb} + \lambda_{bg}\mathcal{L}_{bg} + \lambda_{dist}\mathcal{L}_{dist} + \lambda_{TV}\mathcal{L}_{TV}, \tag{20}$$

where $\mathcal{L}_{pho}$ is the photometric loss, $\mathcal{L}_{bg}$, $\mathcal{L}_{pt\_rgb}$, $\mathcal{L}_{dist}$, $\mathcal{L}_{TV}$ are background entropy loss, per-point color loss, distortion loss, and total variation loss, respectively.

IV. EXPERIMENTS

A. Datasets and Metrics

We conduct evaluations on two challenging unbounded datasets, KITTI [31] and ScanNet [32]. **1) KITTI** consists of image sequences capturing car trajectories using forward-facing stereo cameras. To evaluate our method, we randomly select five near-static sequential segments from the KITTI sequences, which contain almost no dynamic objects such as moving cars, bicyclists, etc. We use COLMAP [33] to obtain the ground truth of camera poses. All the methods are trained with 22 evenly selected images and evaluated on the remaining 42 images. **2) ScanNet** is a large dataset containing 1,613 indoor scenes. Following the experimental configurations of NeuRIS [34], we select 8 scenes and evenly choose 1/6 of the images in each scene for training. For testing, we randomly choose 500 images except for the ones used for training. For image quality evaluation, we adopt the widely used metrics PSNR, SSIM[35] and LPIPS$_{VGG}$[36]. We further conduct NVS of extremely long trajectories and provide the details in the supplementary video.

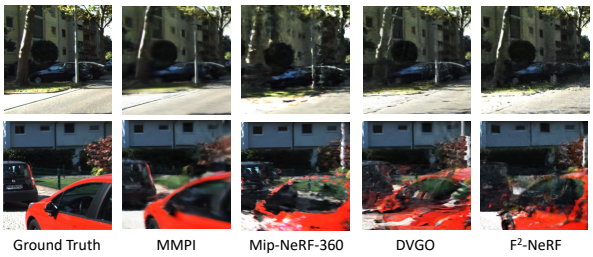

Fig. 4. Visual comparison on the KITTI dataset.

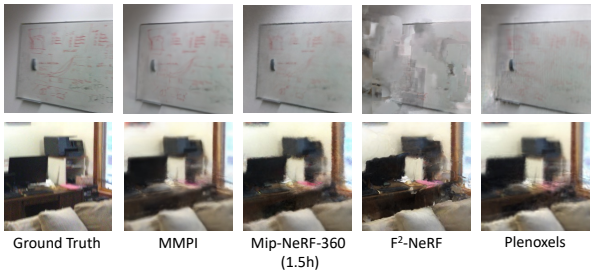

Fig. 5. Visual comparison on the ScanNet dataset.

B. Implementation Details

For the KITTI dataset, we use four MPI grids (towards front, left, right and below, respectively) for blending, each MPI has a resolution of 270×270, and a total of 256 depths are selected. For the ScanNet dataset, we use five MPI grids (towards front, back, left, right and below, respectively) and one centered cube grid for blending, where each MPI grid has a resolution of 192×192 and a total of 128 depths selected, and the centered cube grid has a resolution of 160×160×160. When training on the KITTI dataset, we first train each MPI grid individually for a total of 30k iterations, and then freeze all MPIs except the reliability grid for a total of 10k iterations of reliability blending learning. For training on ScanNet, we omit reliability learning due to the small overlap between each MPI and train a total of 30k iterations directly.

C. Comparisons

We chose several representative works that can synthesize new perspective images of unbounded scenes and conducted quantitative comparisons with our MMPI. The works include NeRF [1], NeRF++ [4], mip-NeRF-360 [5], Plenoxels [8],

TABLE I

RESULTS ON THE KITTI DATASET.

Method	Training Time	PSNR↑	SSIM↑	LPIPS$_{VGG}$ ↓
NeRF [1]	10 hours	13.44	0.362	0.604
NeRF++ [4]	22 hours	10.82	0.297	0.681
mip-NeRF-360 [5]	11 hours	**16.97**	**0.569**	**0.448**
Plenoxels [8]	18 min	13.07	0.262	0.620
TensoRF [14]	40 min	13.33	0.292	0.608
F^2-NeRF [18]	13 min	17.84	0.572	0.465
DVGO[7]	8 min	17.58	0.558	0.495
MMPI	40 min	**19.23**	**0.610**	**0.464**

TABLE II

RESULTS ON THE SCANNET DATASET.

Method	Training Time	PSNR↑	SSIM↑	LPIPS$_{VGG}$ ↓
NeRF++ [4]	28 hours	21.78	0.717	0.556
mip-NeRF-360 [5]	36 hours	**31.11**	**0.881**	**0.277**
Plenoxels [8]	25 min	26.93	0.824	0.441
mip-NeRF-360 (short)	1.5 hours	28.85	0.850	**0.380**
F^2-NeRF [18]	13 min	27.67	0.835	0.387
MMPI (short)	15 min	28.19	0.836	0.445
MMPI	45 min	**29.70**	**0.854**	0.391

TensoRF [14], F^2-NeRF [18], and DVGO [7]. The mip-NeRF-360 experiment was conducted on the A100 GPU due to excessive memory, while other experiments and training time statistics were completed on the Nvidia 3090 GPU.

On the KITTI dataset, MMPI outperforms all baseline methods on metrics except LPIPS (Table I). As shown in Fig. 4, the original NeRF and NeRF++ give very vague prediction results, while Plenoxels, DVGO, and TensoRF show structural artifacts, indicating that simple parameterization cannot solve the long-range NVS problem. The perspective warping of F^2-NeRF is helpful in solving the NVS of long-range scenes, but there are still many wrong textures. Mip-NeRF-360, due to its own parameterization, predicts some regions too smoothly, loses detailed information, and is also accompanied by artifacts. In contrast, our MMPI can better restore the overall information of the scene while maintaining details such as roadside poles and road shadows.

We also conducted an evaluation of our method using the challenging 360-degree dataset ScanNet. The results demonstrate that our MMPI approach achieved superior rendering performance in quantitative comparison to all other fast-training NeRF methods (Table II). As shown in Fig. 5, Plenoxels exhibited noticeable banding artifacts, while F^2-NeRF presented a large area of false patches. Meanwhile, we observed that mip-NeRF-360 also produced satisfactory results on ScanNet after an extended training period. This is due to its utilization of a larger MLP that can recognize and refine finer textured regions over an extended period of training. However, its prolonged training time poses a limitation on its practical application. An early-stopped mip-NeRF-360 with a training time of 1.5 hours will result in a fuzzier outcome.

TABLE III

ABLATION STUDIES ON MULTIPLE MPIS AND RELIABILITY LEARNING.

Settings	PSNR↑	SSIM↑	LPIPS$_{VGG}$ ↓
Single MPI	16.73	0.506	0.482
Multiple MPI w/o reliability	17.75	0.549	0.461
Multiple MPI with reliability	**18.87**	**0.584**	**0.458**

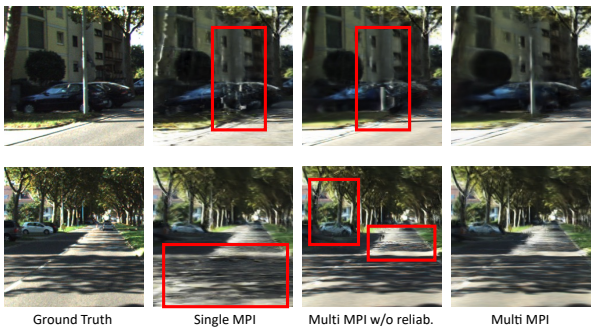

Fig. 6. Visual comparison of ablation study.

D. Ablation Studies

For our ablation studies, we selected a sequence (0096) from the KITTI dataset. We compared the use of a single MPI for training and novel view synthesis (equivalent to using DVGO) with our MMPI approach. To further illustrate the effectiveness of our proposed per-voxel reliability learning for scenes with large overlap among MPIs, we also compared the MMPI method with directly training multiple MPIs without incorporating reliability.

The results in Table III indicate that after blending multiple MPIs, the novel view synthesis quality has improved significantly compared to using only a single MPI. Moreover, the introduction of per-voxel reliability learning has further improved image quality. This demonstrates the scene modeling capability of our proposed Multiple MPI blending, as well as the effectiveness of the training pipeline. We provide more qualitative comparisons in Fig. 6.

E. Limitations

Although a single set of MMPI can handle most capturing situations, the irregularly shaped long trajectories are still challenging. Future research into the methodologies for segmenting trajectories and the placement of multiple sets of MMPIs holds the potential for resolving this issue.

V. CONCLUSION

We propose Multiple Multi-plane Images (MMPI), a flexible representation of neural radiance fields for novel view synthesis of complex unbounded scenes. We encode the scene with multiple multi-plane images with different orientations, and design an adaptive blending strategy based on a learnable reliability grid to boost synthesis quality. By properly organizing and arranging the positions and orientations of those MPIs, our representation is able to cover a wide, unbounded range of the scene of interest. Extensive experiments on two challenging unbounded datasets demonstrate that our MMPI is fast to train and has superior rendering performance than existing state-of-the-art fast-training NeRF methods.

REFERENCES

[1] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," in *ECCV*, 2020.

[2] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan, "Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5855–5864.

[3] M. Tancik, E. Weber, E. Ng, R. Li, B. Yi, J. Kerr, T. Wang, A. Kristoffersen, J. Austin, K. Salahi, *et al.*, "Nerfstudio: A modular framework for neural radiance field development," *arXiv preprint arXiv:2302.04264*, 2023.

[4] K. Zhang, G. Riegler, N. Snavely, and V. Koltun, "Nerf++: Analyzing and improving neural radiance fields," *arXiv preprint arXiv:2010.07492*, 2020.

[5] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman, "Mip-nerf 360: Unbounded anti-aliased neural radiance fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5470–5479.

[6] S. Wizadwongsa, P. Phongthawee, J. Yenphraphai, and S. Suwajanakorn, "Nex: Real-time view synthesis with neural basis expansion," in *CVPR*, 2021.

[7] C. Sun, M. Sun, and H.-T. Chen, "Improved direct voxel grid optimization for radiance fields reconstruction," 2022.

[8] A. Yu, S. Fridovich-Keil, M. Tancik, Q. Chen, B. Recht, and A. Kanazawa, "Plenoxels: Radiance fields without neural networks," *arXiv preprint arXiv:2112.05131*, 2021.

[9] P. Hedman, P. P. Srinivasan, B. Mildenhall, J. T. Barron, and P. Debevec, "Baking neural radiance fields for real-time view synthesis," in *CVPR*, 2021.

[10] C. Reiser, S. Peng, Y. Liao, and A. Geiger, "Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps," in *CVPR*, 2021.

[11] C. Sun, M. Sun, and H.-T. Chen, "Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction," *arXiv preprint arXiv:2111.11215*, 2021.

[12] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *arXiv preprint arXiv:2201.05989*, 2022.

[13] R. Li, M. Tancik, and A. Kanazawa, "Nerfacc: A general nerf acceleration toolbox," *arXiv preprint arXiv:2210.04847*, 2022.

[14] A. Chen, Z. Xu, A. Geiger, J. Yu, and H. Su, "Tensorf: Tensorial radiance fields," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXII*. Springer, 2022, pp. 333–350.

[15] J. Tang, X. Chen, J. Wang, and G. Zeng, "Compressible-composable nerf via rank-residual decomposition," *arXiv preprint arXiv:2205.14870*, 2022.

[16] T. Neff, P. Stadlbauer, M. Parger, A. Kurz, J. H. Mueller, C. R. A. Chaitanya, A. Kaplanyan, and M. Steinberger, "Donerf: Towards real-time rendering of compact neural radiance fields using depth oracle networks," in *Computer Graphics Forum*, 2021.

[17] C. Reiser, R. Szeliski, D. Verbin, P. P. Srinivasan, B. Mildenhall, A. Geiger, J. T. Barron, and P. Hedman, "Merf: Memory-efficient radiance fields for real-time view synthesis in unbounded scenes," *arXiv preprint arXiv:2302.12249*, 2023.

[18] P. Wang, Y. Liu, Z. Chen, L. Liu, Z. Liu, T. Komura, C. Theobalt, and W. Wang, "F^2-nerf: Fast neural radiance field training with free camera trajectories," *arXiv preprint arXiv:2303.15951*, 2023.

[19] T. Zhou, R. Tucker, J. Flynn, G. Fyffe, and N. Snavely, "Stereo magnification: learning view synthesis using multiplane images," *ACM TOG*, vol. 37, no. 4, p. 65, 2018.

[20] P. P. Srinivasan, R. Tucker, J. T. Barron, R. Ramamoorthi, R. Ng, and N. Snavely, "Pushing the boundaries of view extrapolation with multiplane images," in *CVPR*, 2019.

[21] B. Mildenhall, P. P. Srinivasan, R. Ortiz-Cayon, N. K. Kalantari, R. Ramamoorthi, R. Ng, and A. Kar, "Local light field fusion: Practical view synthesis with prescriptive sampling guidelines," *ACM TOG*, 2019.

[22] J. Flynn, M. Broxton, P. Debevec, M. DuVall, G. Fyffe, R. Overbeck, N. Snavely, and R. Tucker, "Deepview: View synthesis with learned gradient descent," in *CVPR*, 2019.

[23] J. Li, Y. He, Y. Hu, Y. Han, and J. Wen, "Learning to compose 6-dof omnidirectional videos using multi-sphere images," in *ICIP*, 2021.

[24] J. Li, Y. He, J. Jiao, Y. Hu, Y. Han, and J. Wen, "Extending 6-dof VR experience via multi-sphere images interpolation," in *ACM MM*, 2021.

[25] P. Phongthawee, S. Wizadwongsa, J. Yenphraphai, and S. Suwajanakorn, "Nex360: Real-time all-around view synthesis with neural basis expansion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[26] M. Tancik, V. Casser, X. Yan, S. Pradhan, B. Mildenhall, P. P. Srinivasan, J. T. Barron, and H. Kretzschmar, "Block-nerf: Scalable large scene neural view synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8248–8258.

[27] K. Rematas, A. Liu, P. P. Srinivasan, J. T. Barron, A. Tagliasacchi, T. Funkhouser, and V. Ferrari, "Urban radiance fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 932–12 942.

[28] H. Turki, D. Ramanan, and M. Satyanarayanan, "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*, 2022, pp. 12 922–12 931.

[29] Y. Xiangli, L. Xu, X. Pan, N. Zhao, A. Rao, C. Theobalt, B. Dai, and D. Lin, "Bungeenerf: Progressive neural radiance field for extreme multi-scale scene rendering," in *European Conference on Computer Vision*. Springer, 2022, pp. 106–122.

[30] S. Marschner and P. Shirley, *Fundamentals of computer graphics*. CRC Press, 2018.

[31] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 3354–3361.

[32] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5828–5839.

[33] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4104–4113.

[34] J. Wang, P. Wang, X. Long, C. Theobalt, T. Komura, L. Liu, and W. Wang, "Neuris: Neural reconstruction of indoor scenes using normal priors," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXII*. Springer, 2022, pp. 139–155.

[35] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[36] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *CVPR*, 2018.