Remote Sensing Novel View Synthesis with Implicit Multiplane Representations

Yongchang Wu, Zhengxia Zou* and Zhenwei Shi, *Member, IEEE*

***Abstract*—Novel view synthesis of remote sensing scenes is of great significance for scene visualization, human-computer interaction, and various downstream applications. Despite the recent advances in computer graphics and photogrammetry technology, generating novel views is still challenging particularly for remote sensing images due to its high complexity, view sparsity and limited view-perspective variations. In this paper, we propose a novel remote sensing view synthesis method by leveraging the recent advances in implicit neural representations. Considering the overhead and far depth imaging of remote sensing images, we represent the 3D space by combining implicit multiplane images (MPI) representation and deep neural networks. The 3D scene is reconstructed under a self-supervised optimization paradigm through a differentiable multiplane renderer with multi-view input constraints. Images from any novel views thus can be freely rendered on the basis of the reconstructed model. As a by-product, the depth maps corresponding to the given viewpoint can be generated along with the rendering output. We refer to our method as Implicit Multiplane Images (ImMPI). To further improve the view synthesis under sparse-view inputs, we explore the learning-based initialization of remote sensing 3D scenes and proposed a neural network based Prior extractor to accelerate the optimization process. In addition, we propose a new dataset for remote sensing novel view synthesis with multi-view real-world google earth images. Extensive experiments demonstrate the superiority of the ImMPI over previous state-of-the-art methods in terms of reconstruction accuracy, visual fidelity, and time efficiency. Ablation experiments also suggest the effectiveness of our methodology design.**

***Index Terms*—Novel View Synthesis, Multi plane images (MPI), Implicit Neural Network, Remote Sensing.**

I. Introduction

NOVEL view synthesis aims at rendering novel images of a 3D scene from arbitrary query viewpoints given a set of pre-collected multi-view images as input. In remote sensing(RS), novel view synthesis has substantial application potential for various tasks such as 3D scene reconstruction, urban management and disaster assessment.

Generating novel views from multi-view RS images is challenging due to the high complexity, view-sparsity and limited view-perspective variations of RS scenes. Recent approaches to novel view synthesis and 3D scene construction are usually designed based on mesh rendering [1–3] and volumetric representation [4–8] techniques, mainly focusing on small-scale scenes, particularly at object level. Recently, implicit neural representation [9–14], as an emerging technique in computer vision and graphics has brought great attention to novel view synthesis and 3D representations. Implicit neural representation provides a novel way to parameterize continuous differentiable signals with neural networks, including volumes and radiance signals in 3D scenes. Based on implicit neural representation, many approaches have been proposed for novel view synthesis very recently. NeRF [15] is known as a representative of such group of approaches. NeRF is proposed to encode the 3D shapes into the network weights, combined with differentiable rendering to achieve end-to-end optimization, where the inefficiency of scene representation and the complexity of rendering are significantly reduced. CityNeRF [16] extends the NeRF from object-level to city-scale with multi-scale remote sensing images as input. A progressive training paradigm is proposed in CityNeRF to store scene details by gradually adding network modules. However, the implicit neural representation in these methods is still limited by the traditional volume rendering process, and thus may suffer from a slow rendering speed.

In remote sensing image view synthesis, restricted by the camera movement of the mounted platform (e.g., satellites and UAVs), the camera usually has a limited range of perspective variation as it flies over the scene. In addition, when the collected views are very sparse, it will become more difficult

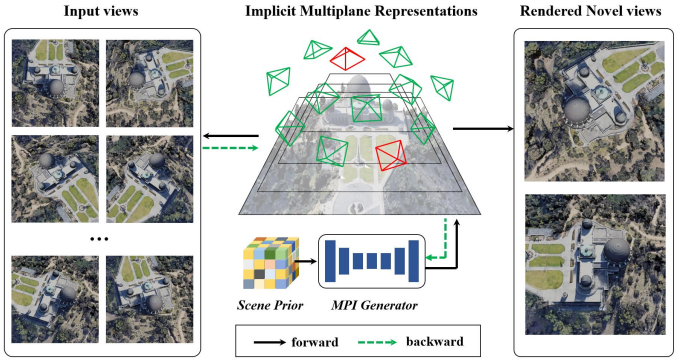

Fig. 1. We propose a new method for remote sensing novel view synthesis. Unlike recent approaches designed based on volume rendering, our method takes advantage of implicit multiplane images (ImMPI) representation to generate novel views from sparse posed images. In this figure, we visualize 11 training views (marked as green) and 2 novel views (marked as red) poses and show corresponding images rendered from optimized ImMPI.

The work was supported by the National Natural Science Foundation of China under the Grant 62125102. *(Corresponding author: Zhengxia Zou (e-mail: zhengxiazou@buaa.edu.cn))*

Yongchang Wu and Zhenwei Shi are with the Image Processing Center, School of Astronautics, Beihang University, Beijing 100191, China, and with the Beijing Key Laboratory of Digital Media, Beihang University, Beijing 100191, China, and also with the State Key Laboratory of Virtual Reality Technology and Systems, School of Astronautics, Beihang University, Beijing 100191, China. Zhengxia Zou is with Department of Guidance, Navigation and Control, School of Astronautics, Beihang University, Beijing 100191, China.

to reconstruct the 3D scene accurately. Recent methods like SinSyn [17] and MINE [18] explore to generate novel view from single image input. However, due to the absence of real scale, it is difficult for the above discussed methods to render high-quality images in remote sensing applications.

To tackle the above challenges, we propose a new method called Implicit Multiplane Images (ImMPI) for RS novel view synthesis. We incorporate multiplane images, an explicit representation naturally suitable for remote sensing with the recent advances in implicit neural representation. In the proposed method, the 3D scene is constructed under a self-supervised optimization paradigm through a differentiable multiplane renderer with multi-view input constraints. Images from any novel view can be thus freely rendered based on the reconstructed 3D scene model. As a by-product, the depth maps corresponding to the given viewpoint can be generated along with the rendering output. In addition, an initialization method is proposed with the motivation that 3D scene priors learned from large remote sensing datasets can be applied across scenes, which further improves the optimization stability and efficiency under sparse-view inputs. Since there are no publicly available datasets for RS novel view synthesis, we build a new dataset for this task using real-world google earth images. Extensive experiments demonstrate the superiority of the ImMPI over previous state-of-the-art methods in terms of reconstruction accuracy, visual fidelity and time efficiency.

The contribution of our work can be summarised as follows:

- We propose implicit MPI representation (ImMPI), a novel method to represent remote sensing 3D scenes. Combining the advantages of implicit neural representation and explicit MPI, the proposed method is naturally suitable for RS novel view synthesis and enables fast rendering.
- We introduce a learning-based network for ImMPI initialization. By extracting 3D scene distribution priors, the optimization process can be significantly accelerated and stabilized.
- We introduce a new dataset for RS novel view synthesis. The dataset consists of 16 real-world 3D scenes collected from Google Earth as well as their multi-view images, including mountains, urban area, buildings, parks, villages, etc. We also made our code publicly available. The dataset and code can be found at https://github.com/wyc-Chang/ImMPI.

II. Related Work

A. 3D Representation for novel view synthesis

Novel view synthesis aims at rendering unobserved viewpoints from a scene given a number of images and camera poses as inputs. It can be modeled as a two-stage process where the first stage recovers geometry from multi-view images and the second one renders images corresponding to given viewpoints. The representation quality of 3D scene is crucial to the quality of the rendered novel views. In this subsection, we introduce common scene representation methods especially for novel view synthesis task, including explicit representations and implicit representations.

Explicit 3D scene representation includes optical flow, mesh, volume, etc. Some early approaches [19–21] reconstruct the optical flow field from multi-view images and achieve view synthesis by interpolation. However, these methods require very dense input views of the scene, which limits their application scope. Some recent methods explore mesh-based representations for novel view synthesis. Liu *et al*.proposed mesh-based novel view synthesis [22] with differentiable rendering applied to reproduce images corresponding to known viewpoints. The 3D meshes are optimized by gradient descent. However, this method requires template meshes for initialization before optimization, which is difficult to obtain due to the complexity of RS scenes. Besides, the images rendered by these methods may suffer from severe artifacts behind occluded areas. Volumetric representation is another approach to representing 3D scenes for novel view synthesis. Early work [23, 24] directly represents RGB color information with voxels. Recently, DeepVoxels [25] proposed a learning-based network to predict 3D feature embedding of each grid in volumetric representation from a set of posed images. Since 3D volume is memory inefficient, the resolution for spatial context needs to be traded off carefully. Although combining with deep convolutional neural networks can compensate for the degradation of rendering high-resolution images from low volume resolutions, the improvement is still limited for remote sensing scenes.

Recent work has demonstrated the capability of implicit neural representation for representing 3D shapes. With implicit neural representations, 3D geometric information can be encoded into the neural network weights by learning the mapping between 3D coordinates and occupancy or signed distance functions [9, 11, 13, 14]. By using a MLP model mapping 5D vectors (3D coordinates and 2D view directions) to transparency and color values, NeRF [15] shows superiority over CNN-based volume rendering methods on view synthesis. Later works like NeRF-W [26] and NeRF++ [27] extend NeRF from object-level to unbounded scenes. For very large-scale scenes, Block-NeRF [28] decomposes the scene into blocks and separately optimizes individual NeRF models. By decoupling the rendering and scene size, Block-NeRF can be scalable to large scenes while allowing individual updates of each block. CityNeRF [16] achieves city-scale scene reconstruction and proposes a progressive learning method to solve multi-scale problems. Despite the above progress, NeRF-based methods require optimization scene-by-scene, and need sufficient views for supervision. For RS scenes with sparse views, the above conditions cannot be guaranteed, so these methods are difficult to apply. To improve the reconstruction on sparse-view conditions, PixelNeRF [29] is proposed very recently, which introduces a CNN-based encoder to learn scene prior from one or a few images. PixelNeRF uses a similar idea to the Prior extractor in the proposed method. However, the difference between the proposed method and PixelNeRF is that the former is designed based on implicit multiplane representations while the latter is based on volume rendering and thus suffers from rendering inefficiency.

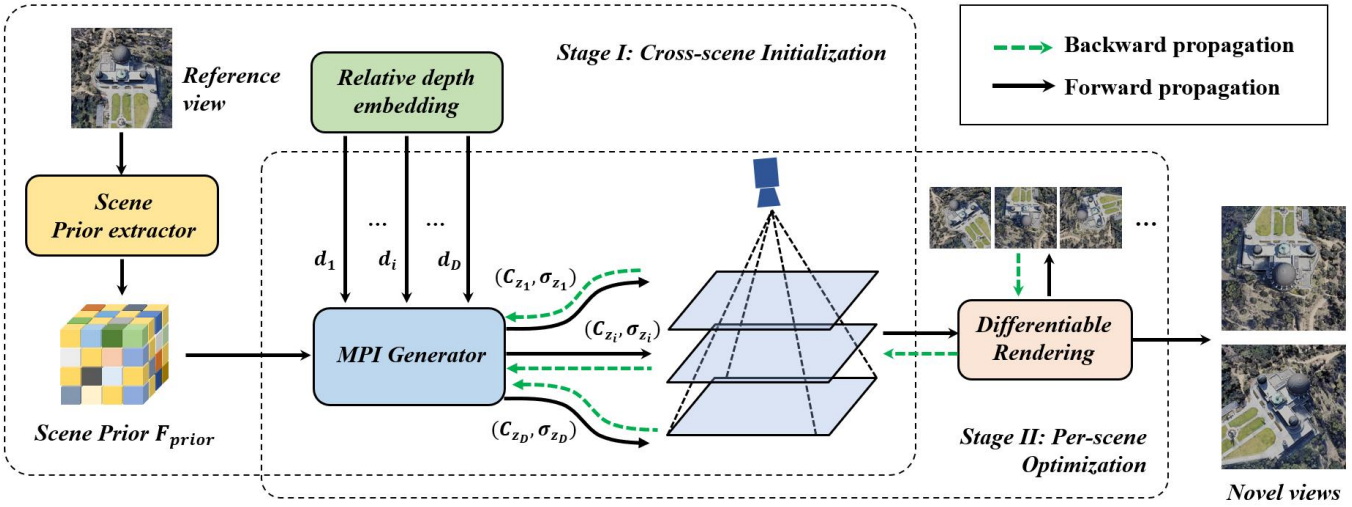

Fig. 2. An overview of our method. The processing pipeline of the proposed method consists of two stages. 1) Cross-scene initialization: Given a selected perspective denoted as reference view, we propose a Prior extractor encoder that produces the latent feature F_{prior} as the initialization of implicit MPI. The encoder is trained on different RS scenes in a self-supervised way. 2) Per-scene optimization: Given a set of input views, we iteratively optimize the ImMPI representation and then render novel views from optimized ImMPI by differentiable rendering.

B. Remote Sensing Image Height/Depth Estimation

In computer vision and photogrammetry remote sensing, depth/height estimation refers to estimating the distance from an object to the camera with single or multi-view input images. Depth/height estimation and view synthesis are closely related since good estimation results can help to produce good view synthesis. Multi-view stereo based methods have achieved accurate results for remote sensing image depth estimation tasks [30–35]. Recently, encoder-decoder based neural networks are introduced to single view height estimation task [36–40]. Multi-task learning is also adopted [41, 42] to increase the accuracy of height estimation by jointly learning from semantic labels. However, these methods require ground-truth depth maps or high-resolution digital surface model(DSM) as supervision, which are not always available in practice. Different from the above methods, in this paper, we take advantage of differentiable rendering and self-supervised learning, where we re-project the rendering views to the original view inputs and enforce them to be similar. This way, with the help of multi-view constraints, the depth and 3d structure of the scene can still be understood properly despite the absence of depth ground truth.

III. Methods

Given a set of multi-view images of a RS scene as input, our method aims at rendering images corresponding to any new viewpoints. An overview of the proposed method is illustrated in Fig.2. There are two main stages in our method:

- Stage I: Cross-scene Neural Network Initialization. We train a scene prior extraction network in order to predict object distribution for ImMPI initialization. The heights of common ground objects in RS scenes have potential regularities, which can be applied to narrow down the solution space of scene reconstruction. Taking single image as input, the scene prior extraction network initializes the implicit MPI representation of the input image. For more details, please refer to Section III-B.
- Stage II: Per-scene Implicit Representation Optimization. After the initialization stage, coarse structure of the scene has been learned in the pre-trained ImMPI model. We then iteratively optimize the parameters of implicit neural network with other viewpoint images. After optimization, accurate novel views can be rendered with the ImMPI model. Details of the per-scene optimization stage can be found in Section III-C.

A. 3D Scene Representation

We combine implicit neural network with explicit multi-plane images to represent RS scenes. In our proposed scene representation, the geometry and appearance information is encoded in convolutional network parameters and the novel view is rendered from multiplane images output by the network. Since the optical axis of an onboard camera is almost perpendicular to the ground in remote sensing platforms, the proposed ImMPI is naturally suitable for remote sensing photography. In addition, its efficiency in rendering images via homography warping and differentiable rendering facilitates real-time applications.

1) Explicit MPI Representation: In our method, we use an implicit neural network, i.e., a deep convolutional neural network (CNN) to generate explicit MPI scene representation, where the multi-planary geometry of the scene is encoded in the weights of the CNN. We follow the algorithm [43] and divide the 3D space into a collection of RGBA layers $\{(C_{z_1}, \sigma_{z_1}), (C_{z_2}, \sigma_{z_2}), \cdots (C_{z_D}, \sigma_{z_D})\}$ in camera frustum, where $C_{z_i}(x, y)$ is a 3-dim vector denoting RGB value at position $[x, y, z_i]^T$ in camera frustum, $\sigma_{z_i}(x, y)$ is a scalar denoting the transmittance of position $[x, y, z_i]^T$, D is the depth sample number. Let $[x, y]^T$ be a 2D pixel coordinate on the plane, with depth hypothesis z_i and pinhole camera

intrinsic K, we can reconstruct the 3D location $[X, Y, Z]^T$ of the point in Cartesian coordinate as follows:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = z_i K^{-1} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = z_i \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \quad (1)$$

2) Novel View Rendering from MPI: Given a viewpoints denoted as target view, MPI renders the corresponding 2D image as follows. First, map all the planes in MPI to novel view camera frustum by differentiable homography warping. The scene MPI representation is constructed in reference-view camera frustum after initialization, whose intrinsic is noted as K_{ref}. Let the novel target-view camera intrinsic be K_{tgt} and the transform matrix between reference-view and target-view be $T_{tgt2ref} = [R_{tgt2ref}, t_{tgt2ref}]$. Then, the correspondence between $[x_{ref}, y_{ref}]$ and $[x_{tgt}, y_{tgt}]$ with respect to pixel coordinate can be calculated as:

$$\begin{bmatrix} x_{ref} \\ y_{ref} \\ 1 \end{bmatrix} = H_{tgt2ref} \begin{bmatrix} x_{tgt} \\ y_{tgt} \\ 1 \end{bmatrix}, \quad (2)$$

where $H_{tgt2ref}$ denotes the homography warping matrix, calculated by transform matrix $[R_{tgt2ref}, t_{tgt2ref}]$ and depth hypothesis z_i as follows:

$$H_{tgt2ref} = K_{ref} \left(R_{tgt2ref} - \frac{t_{tgt2ref} n^T}{z_i} \right) K_{tgt}, \quad (3)$$

where $n^T = [0, 0, 1]^T$ is the normal vector for each plane with respect to reference camera. According to the corresponding relation in Equation 2, MPI representation in target-view camera frustum can be thus sampled from the reference-view.

After warping the MPI representation to target camera frustum, we apply differentiable rendering to get novel view 2D images. For each pixel position (x, y) in novel image plane, RGB pixel values are calculated by following the equation below:

$$I = \sum_{i=1}^{N} T_i \left(1 - exp(-\sigma_{z_i} \delta_{z_i})\right) C_{z_i}. \quad (4)$$

Specifically, $T_i = exp(-\sum_{j=1}^{i-1} -\sigma_{z_j} \delta_{z_j})$ represents the accumulation of transparency from the first plane to the ith plane. δ_{z_i} denotes the Euclidean distance between $[x, y, z_i]$ and $[x, y, z_{i+1}]$ in Cartesian coordinates, which can be calculated by following the Equation 1.

Given the MPI representation of the scene and novel viewpoints, the process of rendering at new viewpoints can be finally expressed as:

$$I_{tgt} = \mathcal{R}\left(f_\Phi, T_{tgt2ref}, K_{ref}, K_{tgt}\right), \quad (5)$$

where $\mathcal{R}$ is the MPI renderer defined by Equation 4. f_Φ is the multiplane images, where in our method is parameterized by a pre-trained CNN model. Since the MPI rendering is essentially a plane-to-plane warping and ray accumulation process, novel views can be rendered very fast.

As for the depth hypothesis, z_i can be estimated from the sparse points by following the structure-from-motion method like COLMAP [44] when calculating camera poses. In practice, the depth range $[z_{near}, z_{far}]$ of RS scenes can be much larger. Therefore, the depth sampling strategy of MPI is crucial. With a pre-defined depth sample number D, we evenly sample hypotheses on reciprocal depth space by following the strategy [45]:

$$\frac{1}{z_i} = \frac{1}{z_{far}} + \frac{i-1}{D} \left(\frac{1}{z_{near}} - \frac{1}{z_{far}} \right). \quad (6)$$

The above operation helps to make the rendering process applicable to complex and large depth range RS scenes. Finally, the depth map I_{depth} under the query view can also be calculated in a similar way as the rendered image:

$$I_{depth} = \sum_{i=1}^{N} T_i (1 - exp(-\sigma_{z_i} \delta_{z_i})) z_i. \quad (7)$$

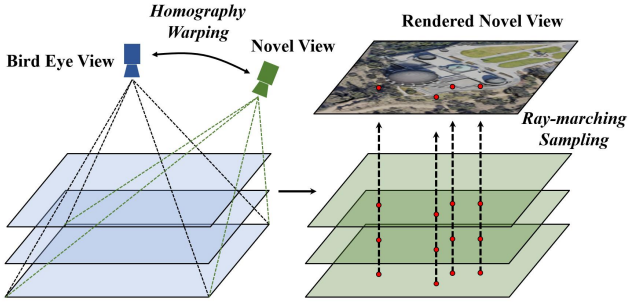

Fig. 3. Illustration of the view rendering process from MPI. Reference-view and target-view are marked with blue and green cameras. The MPI representation is constructed with respect to reference-view frustum initially. To render novel views, the MPI is firstly transformed to the target-view coordinate by using Homography warping. We then apply ray-marching sampling to render the image.

B. Cross-scene Neural Network Initialization

In this subsection, we introduce the Cross-scene Neural Network Initialization method. We introduce a scene prior extraction network and present a self-supervised learning method to learn 3D distribution priors from a single image. The key is the use of differentiable rendering and enforcing the projected views to be similar to the source views. The MPI can be thus roughly estimated with the above view constraints. The overview process is shown in Fig. 2.

1) Network Design and Training Process: Given an image denoted as the reference view, the prior extractor encodes the 2D image feature as F_{prior}. Then, the feature is input to the MPI Generator to obtain the initial MPI representation of the scene. Details of the network design are illustrated in Fig. 4. Specifically, we adopt ResNet18 [46] as the backbone of our Prior extractor. The Prior extractor takes in a single image and produces multi-scale features. Then, the MPI generator takes in multi-scale features and produces multi-scale MPI representations. We set the number of features scales to 5 and the number of MPI scales to 4 in our method.

Relative Depth Embedding. RS scenes may have very different depth ranges. However, we expect to train a generic model that captures depth priors as comprehensively as possible for MPI initialization. Considering that it's difficult to recover the accurate absolute depth from a single image

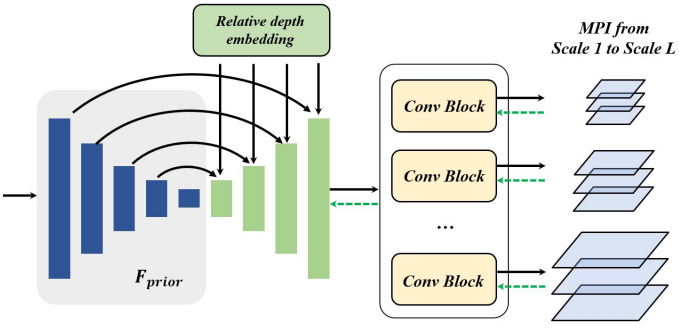

Fig. 4. Architecture of the MPI generator. Depth embedding vectors are concatenated with five-scale F_{prior}. Each scale feature passes through the convBlock to generate corresponding explicit MPI.

directly, we adopt relative depth as the depth value z_i of different planes in MPI. Specifically, suppose the depth sample number is D, we apply a 1-dim positional embedding to $d_i = \{0, 1, 2, ..., D-1\}$ by following the Equation 8 presented in [47]:

$$\gamma(d_i) = [\sin(2^0\pi d_i), \cos(2^0\pi d_i), ..., \sin(2^{L-1}\pi d_i), \cos(2^{L-1}\pi d_i)]. \tag{8}$$

where the hyperparameter L is set to 10 with the dimension of depth position embedding vector equal to 20. Then the depth embedding vectors are merged with multi-scale features F_{prior}. When rendering the target view image, we map the d_i of MPI plane to the absolute depth values z_i with respect to the depth range of reference image.

Training Process. The scene prior extraction network is trained by using a large number of images from different scenes. The network can be trained simply on any remote sensing multi-view stereo datasets. Taking multi-scale MPI representation generated from the corresponding reference image as input, we can render target-view images according to the procedure illustrated in III-A.

2) Loss functions for Cross-scene Training: We train our scene prior extraction network on the WHU MVS/Stereo Dataset [48]. This dataset provides four posed neighbor images for each reference image. We train the model with reference-target paired images. Note that only 2D RGB images in the WHU MVS/Stereo dataset are used for training, and no 3D supervision such as depth map is introduced. To optimize the Prior extractor and MPI generator, we minimize the L1 loss $\mathcal{L}_{L1}$ and Structure Similarity Index Measure (SSIM) loss [49] $\mathcal{L}_{ssim}$ between the rendered image and corresponding ground-truth:

$$\mathcal{L}_{prior}(\theta, \phi) = \sum_{s=1}^{L} \lambda_1 \mathcal{L}_{L1}(I_s^{tgt}, I_s^{gt}) + \lambda_2 \mathcal{L}_{ssim}(I_s^{tgt}, I_s^{gt}), \tag{9}$$

where θ, ϕ are the network parameters of the Prior extractor and MPI generator. s refers to the scale of MPI and L is the total number of scales. λ_{L1} and λ_{ssim} are pre-defined weights to balance the two loss terms. We set $\lambda_1 = 2.0$ and $\lambda_2 = 1.0$ respectively. Since the networks and the MPI rendering process are all differentiable, the networks can be trained in an end-to-end fashion with the above losses.

Algorithm 1: Novel View Synthesis with the proposed ImMPI model.

Input: $\mathbf{B} = \{(\mathbf{I}_n^{src}, \mathbf{T}_n^{src}, \mathbf{K}_n^{src}) | n = 1 : N\}$ (images and camera parameters of training views)
Input: $(\mathbf{T}^{tgt}, \mathbf{K}^{tgt})$ (novel viewpoints)
Input: **Iteration** (iterate number during optimization)
Output: **ImMPI** (scene representation)
Output: $\mathbf{I}^{tgt}$ (novel view RGB image)

// step1: extract priors and initialize ImMPI
  // select a training view denoted as reference view
  $(\mathbf{I}^{ref}, \mathbf{T}^{ref}, \mathbf{K}^{ref}) = Sample(\mathbf{B})$
  // extract prior with pretrained Prior extractor f_θ
  $\mathbf{F}_{prior} = f_{\theta^\star}(\mathbf{I}^{ref})$
// step2: optimize ImMPI with training views
for i *in* $1 :$ **Iteration do**
  for *(*$\mathbf{I}_n^{src}, \mathbf{T}_n^{src}, \mathbf{K}_n^{src}$*) in* $\mathbf{B}$ **do**
    // render images
    $\mathbf{MPI} = f_\phi(\mathbf{F}_{prior})$
    $\mathbf{I}^{syn} = \mathcal{R}(\mathbf{MPI}, \mathbf{T}_n^{src}, \mathbf{K}_n^{src}, \mathbf{K}^{ref})$
    // gradient descent to optimize ImMPI
  end
end
  // **ImMPI***: scene info encoded in F_{prior} and $\phi^\star$
  $\mathbf{MPI}^\star = f_{\phi^\star}(f_{\theta^\star}(\mathbf{F}_{prior}))$
// step3: render novel view
  $\mathbf{I}^{tgt} = \mathcal{R}(\mathbf{MPI}^\star, \mathbf{T}^{tgt}, \mathbf{K}^{tgt}, \mathbf{K}^{ref})$

C. Per-scene Implicit Representation Optimization

Since the initialization stage only utilizes information from one perspective, the initialized MPI representation is not accurate enough and may produce artifacts in occluded areas. In the optimization stage, we further optimize the implicit MPI representation of a specific scene iteratively with images from other viewpoints.

1) Optimization: The per-scene optimization process is illustrated in Fig.2. During the optimization process, the parameters of the Prior extractor are fixed and the weights of the MPI generator are updated through information from other training perspectives. With scene prior F_{prior} inferred from the reference image, we can generate an initial implicit MPI representation and reconstruct the 3D scene under a self-supervised optimization paradigm. Specifically, taking the camera parameters of training views as input, corresponding synthetic images can be rendered according to Equation 5. By calculating the difference between the synthetic images and the ground-truth images, the weights of the MPI Generator are optimized in the gradient descent way. Then the geometry and appearance of the scene are encoded in the network after traversing the train-views several times. We do not directly optimize the MPI pixels for two reasons. On one hand, it's hard to converge because of too many parameters of explicit MPI. On the other hand, when directly optimizing the RGBA values of MPI, each position is treated individually ignoring the local similarity. In comparison, optimizing the parameters of the CNN-based MPI generator preserves spatial continuity, which in turn can bring smoothness to the rendered image.

Experimental results are shown in Tab.III.

2) Loss Functions for Optimization: Unlike NeRF [15] that randomly samples points in training images, the proposed ImMPI renders the entire image of training views. The losses can therefore be applied at the image level. Given multi-scale MPIs and transformation matrix of training view, multi-scale images $\{I_s|s = 1, ..., L\}$ can be rendered according to Equation 5. Following the objective function designed in [43], the loss function is calculated between rendered and ground-truth images. The loss function is defined as:

$$\begin{aligned}\mathcal{L}_{opt}(\phi) = \sum_{s=1}^{L}\sum_{i=1}^{N} \beta_1 \mathcal{L}_{L1}(I_{s,i}^{tgt},\ I_{s,i}^{gt}) \\ + \beta_2 \mathcal{L}_{ssim}(I_{s,i}^{tgt},\ I_{s,i}^{gt}) + \beta_3 \mathcal{L}_{lpips}(I_{s,i}^{tgt},\ I_{s,i}^{gt})\end{aligned} \tag{10}$$

where N is the number of target views and L is the number of scales. $\mathcal{L}_{L1}$, $\mathcal{L}_{ssim}$, and $\mathcal{L}_{lpips}$ represent pixel-wise L1 loss, SSIM loss, and the Learned Perceptual Image Patch Similarity (LPIPS) loss [50], respectively. The LPIPS loss is computed as the distance between two images on their multi-scale features produced by the VGG-19 networks [51], which aims at improving the visual fidelity of the rendering outputs. Pre-defined balancing weights β_1, β_2, and β_3 are set to 2.0, 1.0, 1.0 respectively.

Finally, at novel view rendering phase, given a novel viewpoint, the rendering details of our method are shown in Algorithm 1.

IV. Experiments

In this section, we first introduce the dataset used for training cross-scene initialization and our new dataset for per-scene optimization. Then, experiments are conducted on our new dataset and compare with other view synthesis methods. Finally, controlled experiments and ablation analysis are given to verify the effectiveness of our method.

A. Experimental Setup

WHU MVS/Stereo Dataset[48] is a public large-scale Earth surface reconstruction dataset. It consists of 1776 images captured in 11 strips by UAV. The covered area contained dense and tall buildings, sparse factories, mountains covered with forests, and some bare ground and rivers. This dataset is mainly used for multi-view depth estimation tasks. We train the cross-scene initialization based on this dataset.

LEVIR-NVS Dataset is our newly proposed dataset for remote sensing image novel view synthesis[1]. We use Blender to acquire multi-view 2D images of 3D scene models captured from Google Earth. The dataset consists of 16 scenes, including mountains, cities, villages, buildings, etc. Each scene has 21 multi-view images of size 512x512, 11 views are used for training and the rest are used for testing. Pose transformations such as wrapping and swinging in actual aerial photography are included during the simulation process. The depth range of scenes in LEVIR-NVS varies from 60 meters to 150 meters and each scene covers an area of about ten square kilometers.

[1]LEVIR is the laboratory's name where the authors of this paper are in. NVS is short for "Novel View Synthesis"

Implementation details. Our model is implemented on Pytorch and is trained using a single GeForce RTX 3090 GPU. For the cross-scene initialization training, we apply Adam Optimizer[52] to optimize the model. The learning rate is set to 0.0001 initially and decays 0.5 times per 40 epochs during 200 training epochs. During the per scene optimization, we also apply the Adam optimizer and the learning rate is set to 0.001. For each scene in LEVIR-NVS, the optimization can converge in less than 500 iterations.

Evaluation Metrics. Similar to the metrics adopted in previous novel view synthesis literature [15, 18, 27, 29], we apply PSNR, SSIM, LPIPS [50] to evaluate the rendering accuracy. PSNR and SSIM evaluate pixel-level differences between rendered images and ground-truth images, and LPIPS utilizes a VGG network to evaluate image similarity at feature level.

B. Comparison to other methods

We compare our method with two state-of-the-art novel view synthesis methods: NeRF [15], NeRF++ [27].

- NeRF [15] is a neural implicit representation based method for novel view synthesis. NeRF represents the scene volume with a MLP model mapping 5D vectors (3D coordinates and 2D view directions) to transparency and color values. NeRF optimizes the MLP representation by a set of posed images during training. The optimized MLP then can be used to render novel views with conventional volume rendering approaches. NeRF assumes the entire scene to be contained in a bounded volume and the training views are captured from 360-degree viewpoints distributed on a hemisphere. This assumption makes NeRF hard to be used in large-scale scenes.
- NeRF++ [27] propose to apply NeRF to 360 degree captures of objects within large-scale, unbounded scenes. The authors proposed a novel spatial parameterization scheme called inverted sphere parameterization as a remedy for vanilla NeRF. Specifically, NeRF++ models the scene space with two separate NeRFs, an inner unit sphere and an outer volume, representing foreground and background respectively. After optimizing the models individually, the render results are composited together to generate final novel view images.

Fig. 6 and Fig. 7 show qualitative comparison between our method and the two comparison methods. We can see that our method brings a significant increase in rendering fidelity. Although NeRF and NeRF++ can achieve decent results in the training views, the rendering results on test views are not satisfactory. Both NeRF and NeRF++ suffer from noticeable blurring, loss of details and severe artifacts on test viewpoints. We attribute this phenomenon to two reasons. On one hand, sparse perspectives in RS scenes cannot provide enough supervision for NeRF-based methods and makes it difficult to recover accuracy geometry. For positions in the air, the transparency values are only supervised from overhead with limited view-perspective variations. Therefore the network is easy to converge to a solution that meets all observations but does not conform to the actual scene geometry. On the other

TABLE I
QUANTITATIVE COMPARISON OF TRAIN-VIEW/TEST-VIEW WITH DIFFERENT NOVEL VIEW SYNTHESIS METHODS ON LEVIR-NVS DATASET.

Scenes	PSNR↑			SSIM↑			LPIPS↓		
	NeRF [15]	NeRF++ [27]	ImMPI (Ours)	NeRF [15]	NeRF++ [27]	ImMPI (Ours)	NeRF [15]	NeRF++ [27]	ImMPI (Ours)
Building#1	20.76 / 12.21	22.76 / 14.19	24.92 / 24.77	0.533 / 0.147	0.649 / 0.280	0.867 / 0.865	0.530 / 0.652	0.434 / 0.601	0.150 / 0.151
Building#2	20.34 / 12.38	24.72 / 18.48	23.31 / 22.73	0.444 / 0.184	0.711 / 0.483	0.783 / 0.776	0.573 / 0.668	0.384 / 0.507	0.217 / 0.218
College	23.65 / 12.16	26.66 / 18.57	26.17 / 25.71	0.591 / 0.181	0.746 / 0.471	0.820 / 0.817	0.475 / 0.645	0.319 / 0.474	0.201 / 0.203
Mountain#1	25.17 / 16.29	26.40 / 23.65	30.23 / 29.88	0.565 / 0.290	0.616 / 0.563	0.854 / 0.854	0.556 / 0.674	0.484 / 0.508	0.187 / 0.185
Mountian#2	24.84 / 12.48	26.08 / 24.45	29.56 / 29.37	0.482 / 0.168	0.563 / 0.519	0.844 / 0.843	0.551 / 0.631	0.481 / 0.501	0.172 / 0.173
Mountain#3	27.65 / 18.86	30.43 / 24.14	33.02 / 32.81	0.603 / 0.395	0.737 / 0.541	0.880 / 0.878	0.506 / 0.602	0.365 / 0.483	0.156 / 0.157
Observation	21.42 / 12.48	24.71 / 16.98	23.04 / 22.54	0.498 / 0.168	0.701 / 0.386	0.728 / 0.718	0.501 / 0.631	0.353 / 0.505	0.267 / 0.272
Church	18.75 / 10.59	23.70 / 13.50	21.60 / 21.04	0.421 / 0.126	0.701 / 0.302	0.729 / 0.720	0.563 / 0.654	0.374 / 0.564	0.254 / 0.258
Town#1	21.05 / 12.79	25.89 / 18.26	26.34 / 25.88	0.451 / 0.191	0.747 / 0.401	0.849 / 0.844	0.553 / 0.631	0.328 / 0.510	0.163 / 0.167
Town#2	19.62 / 11.43	25.22 / 17.03	25.89 / 25.31	0.434 / 0.187	0.738 / 0.417	0.855 / 0.850	0.573 / 0.654	0.336 / 0.500	0.156 / 0.158
Town#3	20.44 / 12.25	26.52 / 18.30	26.23 / 25.68	0.470 / 0.230	0.786 / 0.546	0.840 / 0.834	0.561 / 0.658	0.307 / 0.453	0.187 / 0.190
Stadium	21.40 / 14.11	26.64 / 19.26	26.69 / 26.50	0.497 / 0.286	0.753 / 0.504	0.878 / 0.876	0.540 / 0.649	0.313 / 0.466	0.123 / 0.125
Factory	20.26 / 11.79	25.51 / 13.13	28.15 / 28.08	0.495 / 0.242	0.749 / 0.296	0.908 / 0.907	0.572 / 0.656	0.363 / 0.597	0.109 / 0.109
Park	20.04 / 13.86	25.41 / 17.70	27.87 / 27.81	0.427 / 0.228	0.736 / 0.407	0.896 / 0.896	0.576 / 0.653	0.362 / 0.528	0.123 / 0.124
School	20.94 / 12.33	24.90 / 19.50	25.74 / 25.33	0.503 / 0.268	0.673 / 0.516	0.830 / 0.825	0.559 / 0.669	0.399 / 0.479	0.163 / 0.165
Downtown	19.26 / 11.62	24.69 / 15.17	24.99 / 24.24	0.427 / 0.163	0.718 / 0.331	0.825 / 0.816	0.585 / 0.665	0.393 / 0.578	0.201 / 0.205
mean	21.02 / 12.68	25.59 / 17.96	**26.34 / 25.95**	0.481 / 0.201	0.705 / 0.425	**0.835 / 0.831**	0.546 / 0.647	0.371 / 0.514	**0.172 / 0.173**

hand, NeRF-based methods use MLP to encode the geometry and appearance of RS scenes. Although introducing position encoding for 3D coordinates, these algorithms directly optimized MLP parameters ignoring connections between adjacent positions, which contains the geometry prior implicitly. In comparison, the implicit MPI retains the geometry information by combining convolutional network and multiplane images to represent 3D scenes. Benefiting from the efficient ImMPI initialization, our method does not overfit the training viewpoint and renders realistic images corresponding to given test viewpoints. The edges and textures of objects in the scene are clear and highly similar to the ground-truth.

Quantitative evaluations of different methods are given in Tab. I. In Train-view, NeRF and NeRF++ can produce relatively high quantitative accuracy in terms of PSNR, SSIM and LPIPS. However, we see a significant drop in the accuracy of test views. Overfitting to training viewpoints leads to inaccurate scene reconstruction, which in turn affects the synthesis results of new viewpoints. As a comparison, our method performs much better than other algorithms in test views, which suggests that ImMPI achieves more accurate reconstruction of RS scenes and can produce high-quality novel view images.

C. Efficiency in Optimization and Rendering

In this section, we analyze the efficiency of our method during Per Scene Optimization and rendering. Tab. II shows the time consumption of different methods. In this table, "Pre-Training" refers to the pre-training of the methods. "Optimization" denotes the per-scene optimization process. "Rendering" means the time consumption of synthesizing novel views. We can see that NeRF and NeRF++ are purely optimization-based methods without pre-training, and the proposed ImMPI needs to be pre-trained with extra 21h beforehand. At the optimization phase, both NeRF and NeRF++ take more than an hour to optimize on each scene while the proposed ImMPI only takes 30 minutes. At the rendering phase, our method only takes only 1 seconds to render per frame from the

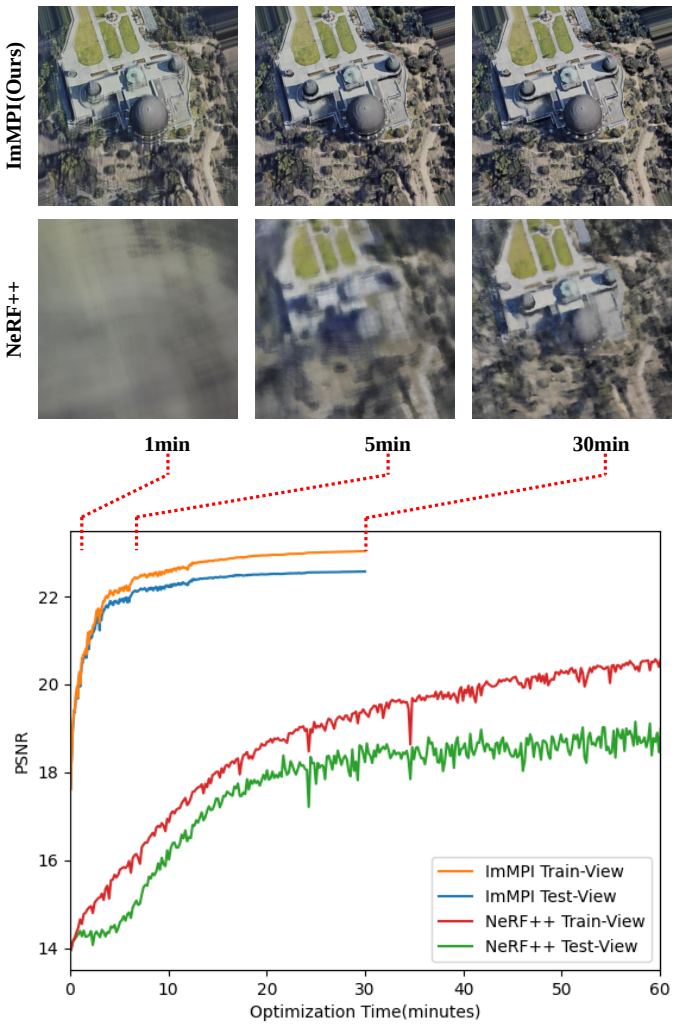

Fig. 5. Optimization comparison between the proposed ImMPI and NeRF++ [27]. ImMPI is 2 times faster than NeRF++ [27] and performs better in test view rendering quality while NeRF++ [27] is prone to overfit with more iterations due to sparse view inputs.

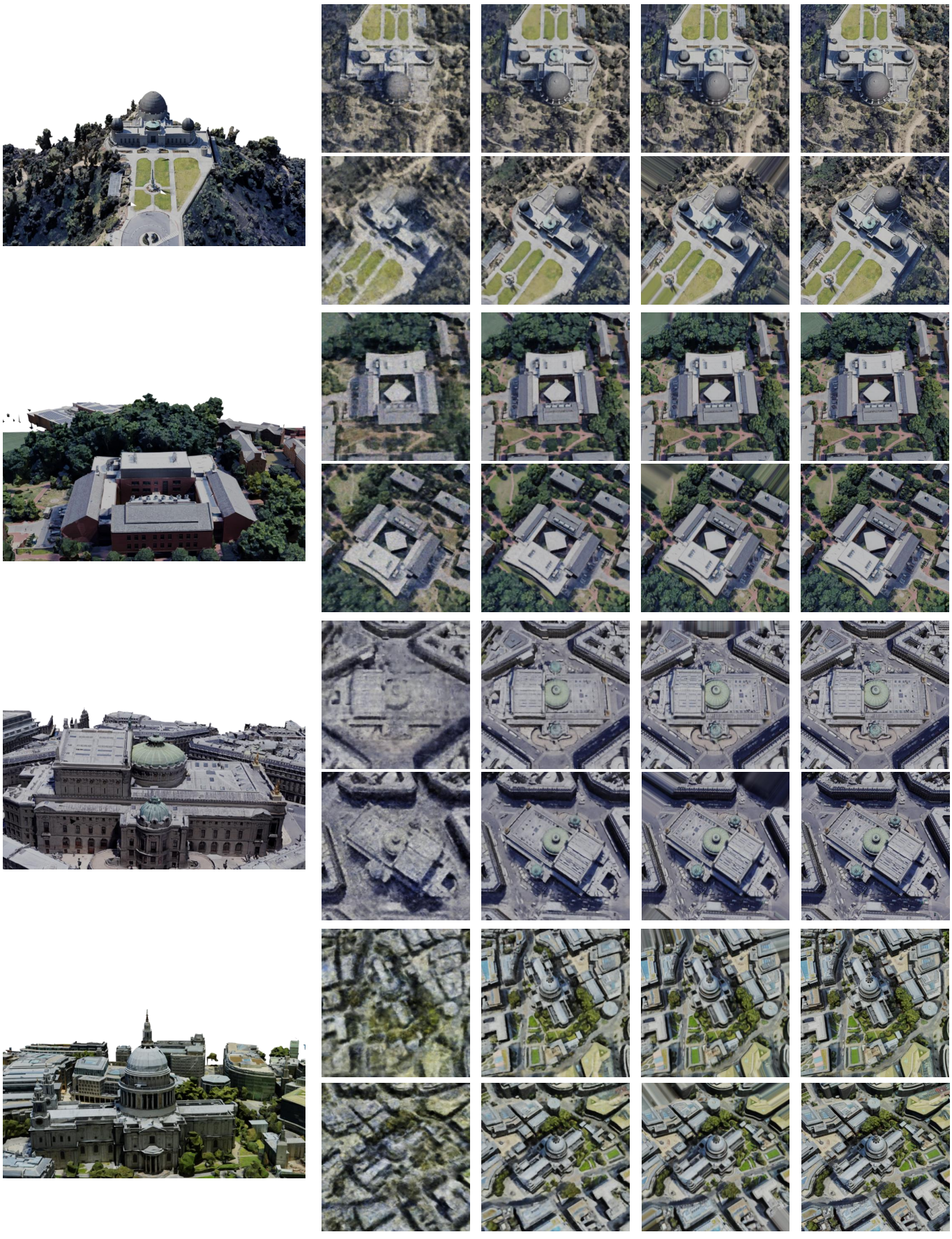

Fig. 6. Qualitative comparison of view rendering with training camera poses on LEVIR-NVS dataset.

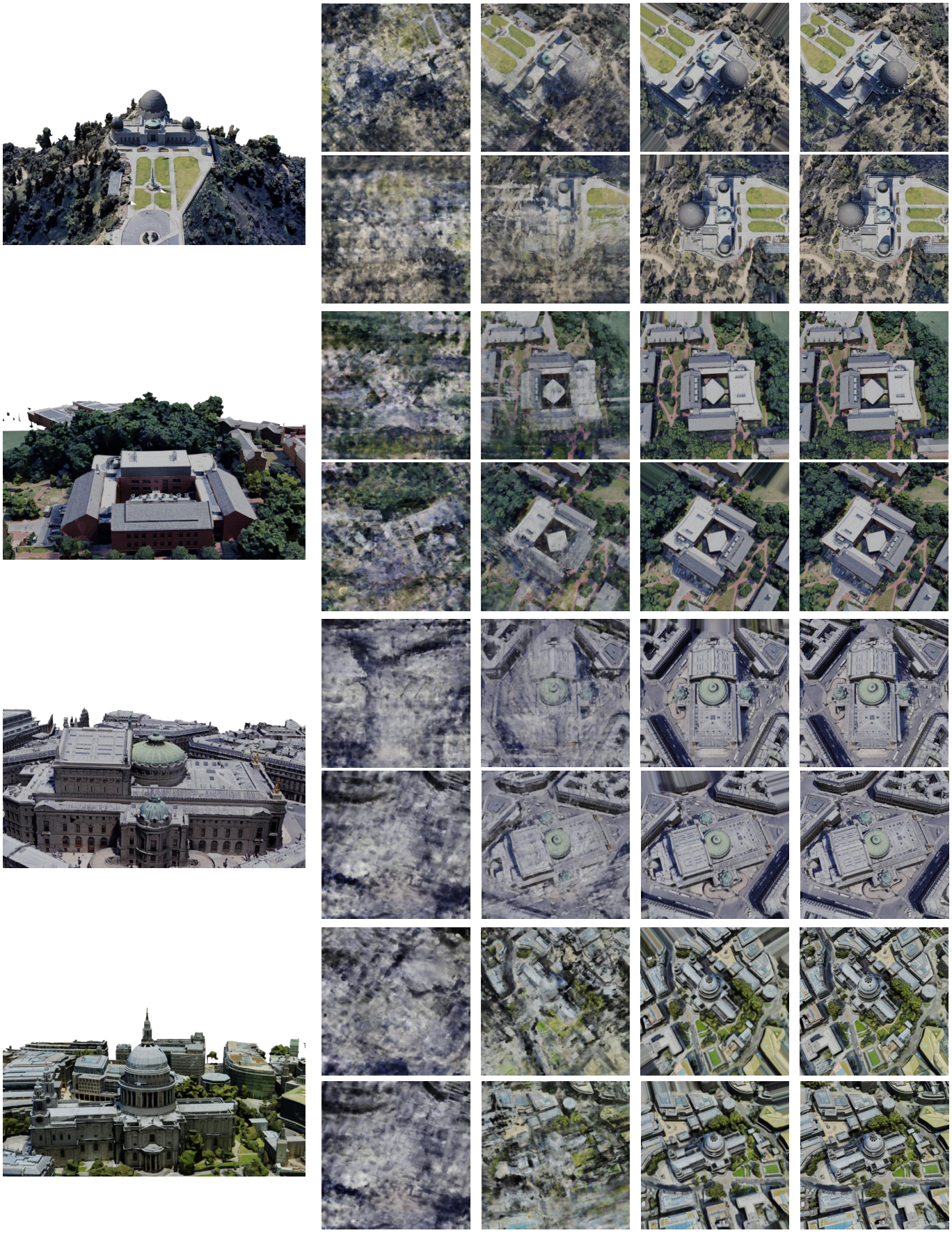

Fig. 7. Qualitative comparison of test view rendering (novel view synthesis) on LEVIR-NVS dataset. The views in this figure have not been seen in the pre-training and optimization process.

TABLE II
QUANTITATIVE COMPARISON OF OPTIMIZATION AND RENDERING SPEED BETWEEN DIFFERENT METHODS.

Method	imageSize	Pre-Train	Optimization	Rendering
NeRF [15]	512x512	-	>90min	>20s
NeRF++ [27]	512x512	-	>60min	>20s
ImMPI (ours)	512x512	21h	<30min	<1s

optimized 3D representation, which is at least 20 times faster than other methods.

NeRF-based methods draw the image pixel-by-pixel in ray-tracing manner. Each pixel in novel view needs to sample points along the ray to calculate the RGB value following the Equation 4. On the one hand, every point on the ray needs the MLP network forward once to obtain the final value C_z, σ_z requiring extensive computation. Purely implicit 3D scene representation and unaccelerated ray-tracing rendering lead to inefficiencies in the optimization process and image rendering. On the other hand, the weights of the MLP network in NeRF and NeRF++ are randomly initialized resulting in large solution space during optimization, which not only leads to time consumption, but also is prone to degenerate geometry estimation of the input scene.

We attribute the faster convergence speed of ImMPI during per-scene optimization to the following reasons: a) ImMPI generates explicit MPI representations whose high efficiency speeds up optimization and rendering process. b) With the learning-based initialization, cross-scene priors are encoded in the ImMPI model, and thus fewer iterations are required in the optimization stage. c) our method inherits the advantages of implicit representation encoding the scene information in network weights, with less optimal variables than explicit representations.

D. Controlled Experiment

Ablation study. We perform ablation studies on 1) cross-scene initialization and 2) per-scene optimization to validate their effectiveness.

Corresponding to the second line of Tab.III, we supplement the ablation experiments in which the explicit multiplane representation is directly optimized. All three metrics have a significant decrease due to ignoring the physical meaning of parameters and lack of sufficient constraints to accurately recover scene geometry. When directly optimizing explicit MPI, the spatial constraints of adjacent positions are ignored and the parameters at each position on each plane are optimized in isolation. In addition, according to the physical meaning of the MPI parameters, RGB should be in the range [0, 1] and σ should be non-negative. Ignoring these constraints leads to inaccurate synthetic images and erroneous gradient computations during optimization.

For cross-scene initialization, we remove this step from the pipeline and optimize the implicit MPI representation directly. For the input F_{prior} of the MPI generator, we adopt two schemes: random initialization and setting as a learnable variable, corresponding to the third and fourth lines of Tab. III.

TABLE III
ABLATION STUDY ON CROSS SCENE INITIALIZATION (INIT) AND PER SCENE OPTIMIZATION (OPTM)

Init	Optm	Iteration	Variable	PSNR↑	SSIM↑	LPIPS↓
✓	×	-	-	16.04	0.3990	0.4134
×	✓	1000	MPI	16.46	0.2917	0.5051
×	✓	1000	ϕ	24.77	0.7568	0.2828
×	✓	1000	$\phi + F_{prior}$	24.81	0.7663	0.2698
✓	✓	200	ϕ	24.89	0.7922	0.2082
✓	✓	500	ϕ	**25.95**	**0.8306**	**0.1734**

TABLE IV
COMPARISON OF MEMORY USAGE, PARAMETER AMOUNT, AND COMPUTATION CONSUMPTION (FLOPS) WHEN RENDERING AN IMAGE WITH DIFFERENT DEPTH HYPOTHESIS NUMBER D IN OUR METHOD.

D	PSNR↑	SSIM↑	LPIPS↓	Memory↓	Params↓	FLOPs↓
8	22.99	0.6419	0.3215	6981MB	16.64M	78.57G
16	24.25	0.7419	0.2600	10774MB	16.64M	147.53G
32	**25.95**	**0.8306**	**0.1734**	18446MB	16.64M	285.46G
48	24.78	0.7869	0.2286	23872MB	16.64M	423.38G

According to quantitative comparison results, the network needs more iterations to converge without the learning-based initialization step. Optimizing F_{prior} and ϕ simultaneously brings better performance, but introduces more parameters at the same time. In comparison, ImMPI converges faster with learning-based initialization and reaches a higher accuracy. This indicates that the cross-scene initialization not only can speed up per-scene optimization, but regularize the optimization to avoid local minimum solutions.

As for the per-scene optimization, we directly render novel views from the ImMPI after the initialization. From the first line of Tab. III, we can see all the three metrics decrease a lot when removing the per scene optimization step. Although there are no significant perspective changes in RS scenes, a single image is still insufficient to support the novel view synthesis. Combined with multi-view information, per-scene optimization significantly improves the quality of rendered images.

Effect of depth hypothesis number.

The accuracy of reconstruction result can be affected by the number of depth hypothesis D in ImMPI. Here we quantitatively analyze the impact by experiment. Specifically, we construct explicit MPI representations with 8, 16, 32 and 48 layers produced by different depth hypothesis numbers in MPI generator. As D increases, the parameters of ImMPI remain the same, but the memory and computation consumption during rendering increases significantly, reducing the speed of per-scene optimization. The results are shown in Tab. IV: PSNR and SSIM increase when D grows from 8 to 32, while the FLOPs almost quadruple increases. In the case of $D = 48$, the quality of rendered image drops slightly. We attribute the decrease to overfitting to training views due to excessive depth sampling. The further increase of depth hypothesis number brings limited accuracy improvement, but the computational resource consumption cannot be ignored. Therefore, we finally set D to 32 in our method.

Effect of training view number. In this part we analyze

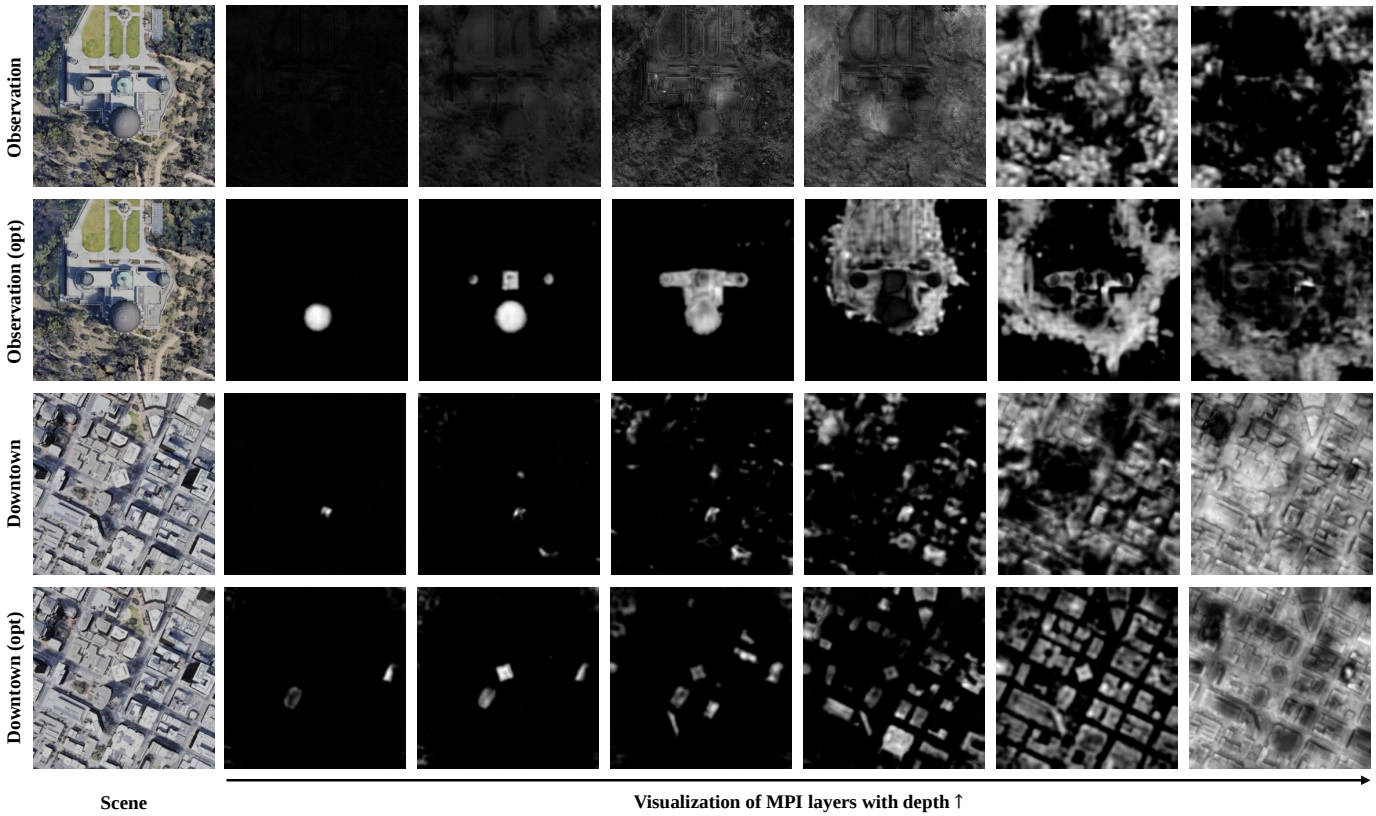

Fig. 8. Visualization of MPI layers with different depths after the cross-scene initialization and per-scene optimization (opt). For 6 of the 32 MPI layers per scene, the σ_{z_i} value is visualized in this figure.

TABLE V
VIEW SYNTHESIS ACCURACY WITH DIFFERENT NUMBER OF TRAINING VIEWS. N DENOTES THE NUMBER OF TRAINING VIEWS USED DURING THE PER-SCENE OPTIMIZATION.

N	Train-View			Test-View		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
3	23.32	0.7445	0.2294	23.16	0.7424	0.2304
5	24.66	0.8016	0.1973	24.70	0.8019	0.1971
7	25.46	0.8168	0.1850	25.53	0.8175	0.1843
9	25.95	0.8226	0.1794	25.56	0.8183	0.1812
11	**26.34**	**0.8351**	**0.1717**	**25.95**	**0.8306**	**0.1734**

the sensitivity of our method to the number of training views adopted during the per-scene optimization. For each scene in the LEVIR-NVS dataset, we experiment with 3, 5, 7, 9 and 11 training views for optimization. The selection of training perspectives follows the principle that the distance between each camera is as far as possible to cover as much scene content as possible. We manually assign the training views under different hyperparameter conditions, and the specific selection settings have been released together with LEVIR-NVS dataset. From Tab. V, all three metrics in train-view and test-view increased with the increase of the number of training views. This shows that more perspectives prompt more accurate reconstruction. Note that ImMPI outputs satisfactory results even with only 3 training views, indicating our method can generalize well to very sparse views in RS scenes.

E. MPI Visualization

To verify the effectiveness of the cross-scene initialization and per-scene optimization, we visualize the σ_{z_i} value of several layers of MPI in Fig. 8 for qualitative analysis. The distribution of σ_{z_i} values along the depth direction implies the geometric information of the scene. It can be seen that ground objects such as buildings and trees of different heights in the same scene appear in different MPI layers after cross-scene initialization. As the depth increases (away from the camera), the content in the scene from the roof to the ground gradually emerges. This indicates that our model can successfully learn depth information from a single image. Also, note that the Prior extractor is trained using the WHU MVS/Stereo dataset, while test image is from the LEVIR-NVS dataset. The visualization results show that our method can accommodate the domain gap between different datasets to some degree. Although the initial ImMPI has some errors in detail, it's sufficient as a good initialization to circumvent some suboptimal solutions. Corresponding to the second and the fourth lines of Fig.8, the geometry of the scene is further accurately reconstructed after per-scene optimization. With less noise and sharper edges, photo-realistic novel views can be synthesized from optimized ImMPI.

F. Depth Estimation

In addition to novel view synthesis, our method can estimate the depth map of the corresponding view according

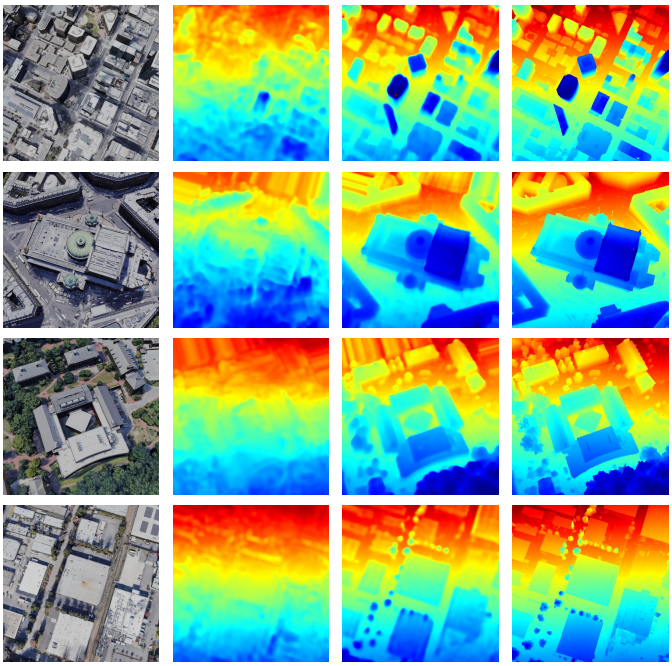

Fig. 9. Visualization of the rendered depth map with our method. "w/o optimize" refers to the depth maps directly rendered from MPI with only initialization. "optimized" refers to the depth maps obtained after the per-scene optimization. Cool color means objects are closer to the camera while warm color means the opposite.

to Equation 7. Depth map qualitative results are shown in Fig. 9. For each scene, we visualize one of the test view depth estimation results. As we can see from the figure, the MPI only with initialization can roughly predict the distance between the ground objects and the camera. Per-scene optimization introduces information from other perspectives, further improving depth estimation accuracy. Compared with the actual depth value, the depth map obtained from MPI still has errors at pixels where the depth value changes sharply. Nevertheless, since we mainly focus on the novel view synthesis task and the depth map is a by-product, our approach is still a feasible way for depth estimation.

V. Conclusion

We propose a new method named ImMPI and a new dataset named LEVIR-NVS for remote sensing novel view synthesis. Given a set of images from a scene, novel view RGB images and corresponding depth maps can be rendered from optimized ImMPI by differentiable rendering. ImMPI combines the advantages of implicit neural network and explicit MPI representation and is naturally suitable for RS images. The Implicit representation encodes the scene geometry to network weights with fewer parameters and the explicit MPI achieves faster rendering speed. We also propose a learning-based cross-scene initialization method to extract scene priors, which dramatically speeds up the per scene optimization and improves accuracy under sparse view inputs. Compared with NeRF-based methods, ImMPI shows significant improvement with 2 times optimization speed and 20 times rendering speed.

References

[1] A. Kanazawa, S. Tulsiani, A. A. Efros, and J. Malik, "Learning category-specific mesh reconstruction from image collections," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 371–386.

[2] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, and Y.-G. Jiang, "Pixel2mesh: Generating 3d mesh models from single rgb images," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 52–67.

[3] T. Groueix, M. Fisher, V. G. Kim, B. C. Russell, and M. Aubry, "A papier-mâché approach to learning 3d surface generation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 216–224.

[4] A. Brock, T. Lim, J. M. Ritchie, and N. Weston, "Generative and discriminative voxel modeling with convolutional neural networks," *arXiv preprint arXiv:1608.04236*, 2016.

[5] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1912–1920.

[6] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2367–2376.

[7] A. Kar, C. Häne, and J. Malik, "Learning a multi-view stereo machine," *Advances in neural information processing systems*, vol. 30, 2017.

[8] S. Tulsiani, T. Zhou, A. A. Efros, and J. Malik, "Multi-view supervision for single-view reconstruction via differentiable ray consistency," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2626–2634.

[9] Z. Chen and H. Zhang, "Learning implicit fields for generative shape modeling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5939–5948.

[10] K. Genova, F. Cole, D. Vlasic, A. Sarna, W. T. Freeman, and T. Funkhouser, "Learning shape templates with structured implicit functions," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 7154–7164.

[11] K. Genova, F. Cole, A. Sud, A. Sarna, and T. Funkhouser, "Local deep implicit functions for 3d shape," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4857–4866.

[12] C. Jiang, A. Sud, A. Makadia, J. Huang, M. Nießner, T. Funkhouser *et al.*, "Local implicit grid representations for 3d scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6001–6010.

[13] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3d reconstruction in function space," in *Proceedings of the*

IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 4460–4470.

[14] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "Deepsdf: Learning continuous signed distance functions for shape representation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 165–174.

[15] 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 *European conference on computer vision*. Springer, 2020, pp. 405–421.

[16] Y. Xiangli, L. Xu, X. Pan, N. Zhao, A. Rao, C. Theobalt, B. Dai, and D. Lin, "Citynerf: Building nerf at city scale," *arXiv preprint arXiv:2112.05504*, 2021.

[17] O. Wiles, G. Gkioxari, R. Szeliski, and J. Johnson, "Synsin: End-to-end view synthesis from a single image," 2019.

[18] J. Li, Z. Feng, Q. She, H. Ding, C. Wang, and G. H. Lee, "Mine: Towards continuous depth mpi with nerf for novel view synthesis," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 12 578–12 588.

[19] M. Levoy and P. Hanrahan, "Light field rendering," in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 1996, pp. 31–42.

[20] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen, "The lumigraph," in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 1996, pp. 43–54.

[21] A. Davis, M. Levoy, and F. Durand, "Unstructured light fields," in *Computer Graphics Forum*, vol. 31, no. 2pt1. Wiley Online Library, 2012, pp. 305–314.

[22] S. Liu, T. Li, W. Chen, and H. Li, "Soft rasterizer: A differentiable renderer for image-based 3d reasoning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 7708–7717.

[23] K. N. Kutulakos and S. M. Seitz, "A theory of shape by space carving," *International journal of computer vision*, vol. 38, no. 3, pp. 199–218, 2000.

[24] S. M. Seitz and C. R. Dyer, "Photorealistic scene reconstruction by voxel coloring," *International Journal of Computer Vision*, vol. 35, no. 2, pp. 151–173, 1999.

[25] V. Sitzmann, J. Thies, F. Heide, M. Nießner, G. Wetzstein, and M. Zollhofer, "Deepvoxels: Learning persistent 3d feature embeddings," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2437–2446.

[26] R. Martin-Brualla, N. Radwan, M. S. Sajjadi, J. T. Barron, A. Dosovitskiy, and D. Duckworth, "Nerf in the wild: Neural radiance fields for unconstrained photo collections," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7210–7219.

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

[28] 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," *arXiv preprint arXiv:2202.05263*, 2022.

[29] A. Yu, V. Ye, M. Tancik, and A. Kanazawa, "pixelnerf: Neural radiance fields from one or few images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4578–4587.

[30] G. L. K. Morgan, J. G. Liu, and H. Yan, "Precise subpixel disparity measurement from very narrow baseline stereo," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 48, no. 9, pp. 3424–3433, 2010.

[31] M. Mahato, S. Gedam, J. Joglekar, and K. M. Buddhiraju, "Dense stereo matching based on multiobjective fitness function—a genetic algorithm optimization approach for stereo correspondence," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 6, pp. 3341–3353, 2019.

[32] J. Li, Y. Liu, S. Du, P. Wu, and Z. Xu, "Hierarchical and adaptive phase correlation for precise disparity estimation of uav images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 12, pp. 7092–7104, 2016.

[33] J. Liu, L. Zhang, Z. Wang, and R. Wang, "Dense stereo matching strategy for oblique images that considers the plane directions in urban areas," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 7, pp. 5109–5116, 2020.

[34] G. Facciolo, C. De Franchis, and E. Meinhardt-Llopis, "Automatic 3d reconstruction from multi-date satellite images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 57–66.

[35] Y. Hou, J. Peng, Z. Hu, P. Tao, and J. Shan, "Planarity constrained multi-view depth map reconstruction for urban scenes," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 139, pp. 133–145, 2018.

[36] H. A. Amirkolaee and H. Arefi, "Height estimation from single aerial images using a deep convolutional encoder-decoder network," *ISPRS journal of photogrammetry and remote sensing*, vol. 149, pp. 50–66, 2019.

[37] V.-C. Miclea and S. Nedevschi, "Monocular depth estimation with improved long-range accuracy for uav environment perception," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–15, 2021.

[38] X. Li, M. Wang, and Y. Fang, "Height estimation from single aerial images using a deep ordinal regression network," *IEEE Geoscience and Remote Sensing Letters*, 2020.

[39] C.-J. Liu, V. A. Krylov, P. Kane, G. Kavanagh, and R. Dahyot, "Im2elevation: Building height estimation from single-view aerial imagery," *Remote Sensing*, vol. 12, no. 17, p. 2719, 2020.

[40] S. Xing, Q. Dong, and Z. Hu, "Gated feature aggregation for height estimation from single aerial images," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2021.

[41] M. Carvalho, B. Le Saux, P. Trouvé-Peloux, F. Champagnat, and A. Almansa, "Multitask learning of height and semantics from aerial images," *IEEE Geoscience and*

Remote Sensing Letters, vol. 17, no. 8, pp. 1391–1395, 2019.

[42] S. Srivastava, M. Volpi, and D. Tuia, "Joint height estimation and semantic labeling of monocular aerial images with cnns," in *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. IEEE, 2017, pp. 5173–5176.

[43] T. Zhou, R. Tucker, J. Flynn, G. Fyffe, and N. Snavely, "Stereo magnification: Learning view synthesis using multiplane images," *arXiv preprint arXiv:1805.09817*, 2018.

[44] 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.

[45] Q. Xu and W. Tao, "Learning inverse depth regression for multi-view stereo with correlation cost volume," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 12 508–12 515.

[46] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[47] M. Tancik, P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. Barron, and R. Ng, "Fourier features let networks learn high frequency functions in low dimensional domains," *Advances in Neural Information Processing Systems*, vol. 33, pp. 7537–7547, 2020.

[48] J. Liu and S. Ji, "A novel recurrent encoder-decoder structure for large-scale multi-view stereo reconstruction from an open aerial dataset," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6050–6059.

[49] 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.

[50] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 586–595.

[51] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[52] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

Yongchang Wu received his B.S. degree from the Image Processing Center School of Astronautics, Beihang University in 2020. He is currently pursuing his M.S. degree in the Image Processing Center, School of Astronautics, Beihang University. His research interests include machine learning, deep learning and 3D reconstruction.

Zhengxia Zou received his B.S. degree and his PhD degree from the Image Processing Center, School of Astronautics, Beihang University in 2013 and 2018, respectively. He is currently an Associate Professor at the School of Astronautics, Beihang University. During 2018-2021, he was a postdoc research fellow at the University of Michigan, Ann Arbor. His research interests include computer vision and related problems in remote sensing and autonomous driving. He has published more than 20 peer-reviewed papers in toptier journals and conferences, including TPAMI, TIP, TGRS, CVPR, ICCV, AAAI. His research has been featured in more than 30 global tech media outlets and adopted by multiple application platforms with over 50 million users worldwide. His personal website is https://zhengxiazou.github.io/.

Zhenwei Shi (Member, IEEE) received his Ph.D. degree in mathematics from Dalian University of Technology, Dalian, China, in 2005. He was a Postdoctoral Researcher in the Department of Automation, Tsinghua University, Beijing, China, from 2005 to 2007. He was Visiting Scholar in the Department of Electrical Engineering and Computer Science, Northwestern University, U.S.A., from 2013 to 2014. He is currently a professor and the dean of the Image Processing Center, School of Astronautics, Beihang University. His current research interests include remote sensing image processing and analysis, computer vision, pattern recognition, and machine learning.

Dr. Shi serves as an Editor for the Pattern Recognition, the ISPRS Journal of Photogrammetry and Remote Sensing, and the Infrared Physics and Technology, etc. He has authored or co-authored over 200 scientific papers in refereed journals and proceedings, including the IEEE Transactions on Pattern Analysis and Machine Intelligence, the IEEE Transactions on Image Processing, the IEEE Transactions on Geoscience and Remote Sensing, the IEEE Geoscience and Remote Sensing Letters, the IEEE Conference on Computer Vision and Pattern Recognition and the IEEE International Conference on Computer Vision. His personal website is http://levir.buaa.edu.cn/.