Mega-NeRF++: An Improved Scalable NeRFs for High-resolution Photogrammetric Images

Yiwei Xu, Tengfei Wang, Zongqian Zhan, Xin Wang*

School of Geodesy and Geomatics, Wuhan University, Wuhan 430079, China
(xywjohn_SGG2020, tf.wang)@whu.edu.cn, (zqzhan, xwang)@sgg.whu.edu.cn

Keywords: NeRF, Mega-NeRF, High Resolution images, Photogrammetric Images.

Abstract:

Over the last few years, implicit 3D representation has attracted more and more research endeavors, typified by the so-called Neural Radiance Fields (NeRF). The original NeRF and some relevant variants mostly address on small-scale scene (such as, indoor or tiny toys), which already show good novel views rendering results. It still remains challenging when dealing with wide coverage area that is captured by large number of high-resolution images, the time efficiency and rendering quality is generally limited. To cope with large-scale scenario, recently, Mega-NeRF was proposed to divide the area into several overlapping sub-area and train corresponding sub-NeRFs, respectively. Mega-NeRF adopts the method of parallel training of multiple sub-modules, which means sub-modules are absolutely independent of each other, which might in principle not be an optimal solution, as two sub-NeRFs of adjacent sub-models obtained by parallel training are likely to get different rendering results for the overlapping area, and the final rendering result is supposed to be negative affected. Therefore, we present Mega-NeRF++, and our goal is to improve Mega-NeRF by implementing extra sub-models optimization that alleviate the rendering discrepancy of overlapping sub-NeRFs. More specifically, we further fine tune the original Mega-NeRFs by considering the consistency of adjacent overlapping area, which means the training data used in the optimization are only from the overlapping region, and we also proposed a novel loss, so that it not only takes into account the difference between the prediction of each sub-model and the true value, but also considers the consistency of the predicted results between various adjacent sub-modules in the overlapping region. The experimental results show that, for the overlapping area, our Mega-NeRF++ can qualitatively render better images with higher fidelity and quantitively have higher PNSR and SSIM compare to original Mega-NeRF.

1. Introduction

Nowadays, novel view synthesis based on collected images has again become a research hotspot (SfM, NeRF, Mip-NeRF, Mega-NeRF etc.), mainly due to the development of neuron-based deep learning techniques. The way that traditional photogrammetry does typically contains several very complex processing, including feature extraction and matching, image orientation and sparse point cloud generation, dense matching, delaunay triangulation and mesh model generation etc., and novel view is synthesized via back-projection based on mesh model and image orientation parameters (Schonberger et al., 2016). Recently, the emergence of NeRF has made the neuron-based implicit representation of 3D scene be possible (Mildenhall et al., 2020). Based on the input images with known orientation parameters, NeRF train a MLP and implicitly learns 3D information of the scene, and it can be then used to predicts the color information and volume density at any sample point along a specific ray, which are integrated to obtain the color of the corresponding pixel, thus completing the synthesis of new view from arbitrary pose.

The original NeRF has already been demonstrated to be able to achieve good rendering results when the scene is controlled (e.g., desktop-sized) and the image resolution is not very high (Turki et al., 2022). However, for the photogrammetric dataset that is typically with high resolution of abundant detail information, large number of images covering wide ground area, it is hardly and feasible for original NeRF to be trained on a common consumer computer, regarding training time and hardware memory storage; Moreover, due to the limitations of original NeRF in rendering unbounded scenes and the change of illumination conditions, it is difficult to directly apply traditional NeRF technology to large-scale and high-resolution UAV images.

To address the abovementioned issues, recently, Turki et al. (2022) proposed Mega-NeRF that successfully trained NeRF on UAV images. It mainly partitions the whole scene into several sub-blocks and trains smaller sub-NeRF for each sub-blocks, after that, rendering new views only needs to splice all the relevant sub-NeRFs rendering results together. As the sub-NeRF of each sub-block is a self-contained MLP, it is typically trained individually in parallel with no inter-block communication (Turki et al., 2022), which means all sub-models are independent of each other. This may result in ambiguous rendering, i.e., the adjacent sub-models obtained by parallel training are likely to get different results when rendering a certain overlapping area. Although Mega-NeRF has already applied a weighted averaging strategy to filter out the discrepancy in overlapping region, we can still find a relatively clear brightness discontinuous variation, as Figure 1 illustrates.

Therefore, we present Mega-NeRF++, whose main goal is to improve Mega-NeRF by implementing extra fine tuning (or optimization) for the original sub-NeRFs. This extra fine tuning is akin to Mega-NeRF when training each sub-NeRFs, but we have made the following improvements: 1) To perform the fine tuning, for any two adjacent sub-models, only rays from corresponding overlapping region are sampled as training data;
2) To optimize the original Mega-Nerf so that it takes into account the consistency of the different predicted results from two adjacent sub-models, we proposed an improved original loss function which considers the rendering consistency existing in overlapping area. The overall working pipeline is shown in Fig.2. which mainly contains steps :

I. Dataset Preparation: For the collected images, the open-source framework COLMAP is used to estimate the poses of the images, and partition the large scene into several sub-blocks according to the estimated poses.

* Corresponding author

II. Mega-NeRF Training: Based on the partitioning results, the original Mega-NeRF is used to train small sub-NeRF for each sub-block.

III. Mega-NeRF++ Optimization: Due to the obviously inconsistent prediction results (shown in Figure 1.) in the overlapping region generated adjacent sub-NeRFs (trained by Mega-NeRF), our Mega-NeRF++ optimizes all sub-models so that the rendering results from different sub-models in the overlapping region are as consistent as possible, while also as close to the ground truth as possible.

In sum, this paper proposes an improved Mega-NeRF, named as Mega-NeRF++, as an updated derivative of Mega-NeRF, an improved loss function is used to fine tune the original Mega-NeRF and all the relevant sub-NeRFs are optimized by using the training data only sampled from overlapping regions. After fine tuning, the predicted rendering results of the adjacent sub-models in the overlapping region are supposed to be more consistent.

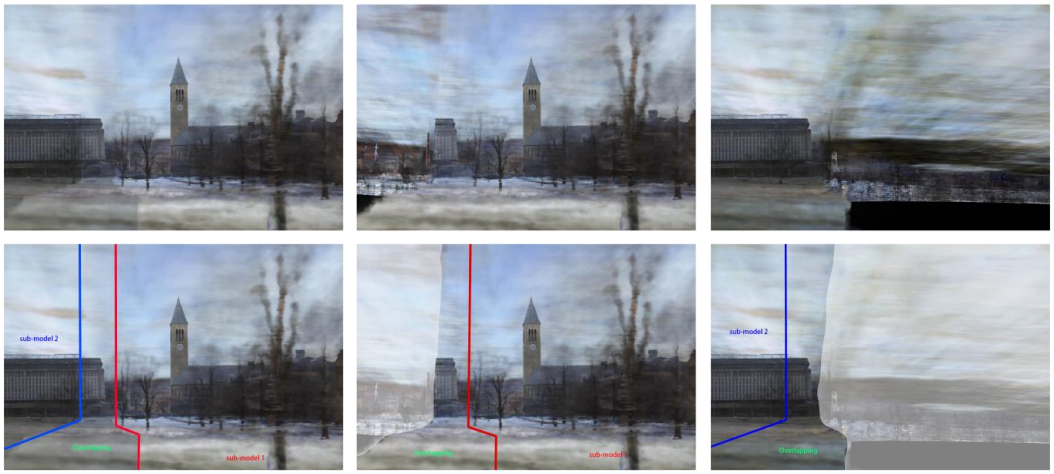

Figure 1. Results of vanilla Mega-NeRF. Final rendering results (left), rendering result only from sub-model1 (middle) and rendering result only from sub-model2 (right). Final rendering results (left) is obtained by merging the rendering result only from sub-model1 (middle) and rendering result only from sub-model2 (right), and it shows the brightness variation between overlapping region and non-overlapping regions on the left and right sides.

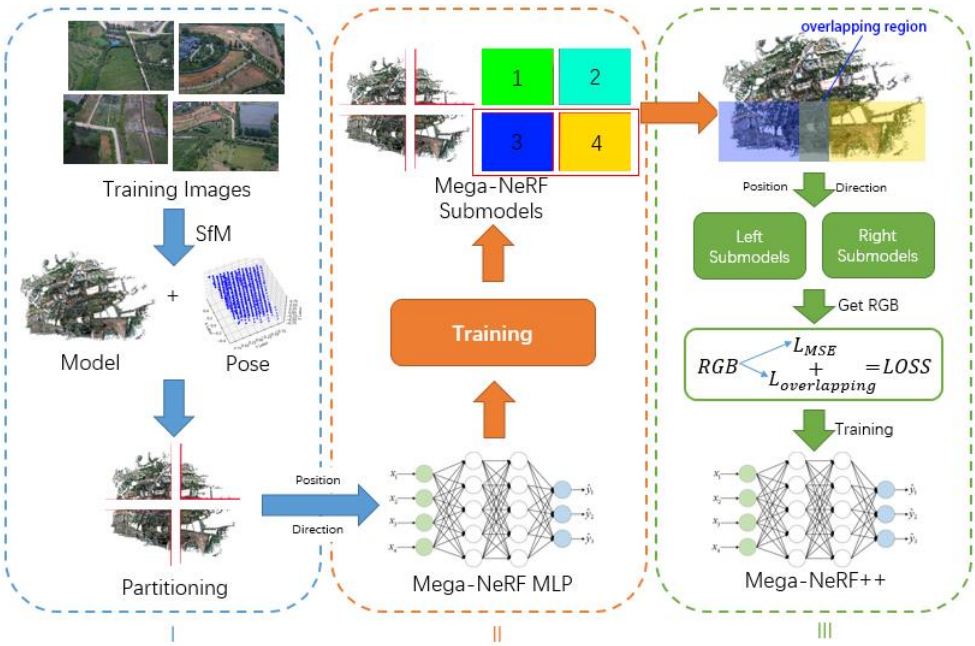

Figure 2. The workflow of our Mega-NeRF++.

2. Related works

Novel view synthesis and NeRF. Novel view synthesis typically refers to the task of generating a new target image based on some collected images whose relevant information is known (such as, exterior and interior orientation parameters). Before the emergence of NeRF, traditional algorithms had already made strides in novel view synthesis to a certain extent. Chen et al. (1993) proposed the concept of interpolation, presenting a method for predicting intermediate image from multiple images stored at adjacent view points. Struct from Motion (SfM) and Multi-View Stereo (MVS) construct the mesh model according to the images obtained in corresponding scene, and generate the image of any view point via back-projection (Schonberger et al., 2016). With the development of machine learning, the emergence

of NeRF proposed a method for implicit reconstruction of 3D scenes using MLP. The original NeRF employed MLPs to predict the color and volume density at multiple sampling points on the ray, and use the discrete form of volume rendering integral formula to obtain the color of the corresponding pixel (Mildenhall et al., 2020). NeRF’s variants have improved the performance of origin NeRF in many aspects: NeRF++ begins by examining the mechanisms underlying the effective resolution of the shape-radiance ambiguity in the original NeRF. Additionally, it employs the inverted sphere parametrization method to enhance NeRF's rendering performance in unbounded scenes (Zhang et al., 2020). Mip-NeRF utilizes a conical frustums-based rendering sampling approach and integrates Positional Encoding to mitigate aliasing in NeRF's original rendering process, thereby enhancing rendering quality (Martin-Brualla et al., 2021). NeRF-W utilizes appearance embedding vectors to enable the NeRF model to accommodate varying lighting conditions in image data, and incorporates an additional MLP for predicting transient objects to address potential transient occlusion in multi-source images (Martin-Brualla et al., 2021). BlockNeRF (Tancik et al., 2022) and Mega-NeRF (Turki et al., 2022) both employ the strategy of partitioning complex large-scale scenes and utilizing individual NeRF models to learn 3D information of each sub-block, thereby successfully extending the application of NeRF to large-scale scenes.

Unbounded scenes. To lift NeRF out of desktop-sized scene and address the issue of unbounded application, Zhang et al. (2020) introduced the NeRF++ and Barron et al. (2022) proposed the Mip-NeRF 360 to tackle the challenge. NeRF++ divides the entire unbounded scene into foreground and background, separately training an individual MLP for the foreground and background, and utilizes the corresponding MLP to predict the color and volume density of sampling points within the foreground and background (Zhang et al., 2020). Mip-NeRF 360 employs a non-linear scene parameterization approach, effectively confining the unbounded scene within a bounded space. Furthermore, Mip-NeRF 360 introduces online distillation and a novel distortion-based regularization, enhancing model training and rendering efficiency while also mitigating the blurring artifacts encountered during the rendering of unbounded scenes (Barron et al., 2022).

Lighting differences and transient occlusions. In practice, the collected image data often exhibit various lighting conditions and transient occlusions, both of them are likely to impact NeRF's ability to capture the 3D characteristics of static objects within the scene. NeRF in the wild (NeRF-W) addresses this challenge by employing an additional MLP to analyze all transient objects present in the images, thus facilitating the separation of static objects from transient ones across the entire image dataset. Additionally, NeRF-W assigns an appearance embedding vector to each training image, leveraging it as an input parameter for the MLP, enabling the trained NeRF model to effectively mitigate differences in illumination among different images (Martin-Brualla et al., 2021).

Fast training and rendering. The computational overhead associated with NeRF training and rendering escalates sharply with scene complexity and image resolution, prompting a critical need to mitigate time costs while upholding NeRF performance standards. Instant-NGP introduces a multiresolution hash encoding approach, enabling NeRF implementation with a reduced network size without sacrificing accuracy. This innovation effectively compresses training time from hours to seconds (Muller et al., 2022). 3D Gaussian Splatting utilizes 3D Gaussian spheres to model the entire scene, offering a representation that overcomes noise and rendering mode limitations while significantly curtailing rendering time (Kerbl et al., 2023). Mega-NeRF (Turki et al., 2022), which is most relative to us, employs various strategies to expedite both training and rendering. During training, Mega-NeRF utilizes Spatial Data Pruning to eliminate extraneous rays from training images, and Guided Sampling to skip empty spaces and sample fewer points near the surface of objects, thereby markedly reducing the number of sampling points per ray. Furthermore, Mega-NeRF optimizes rendering efficiency by reusing results from previously rendered images, thereby accelerating the rendering of subsequent frames and enhancing the efficiency of continuous rendering sequences.

Large-scale NeRF. The original NeRF and subsequent advancements have predominantly focused on training and rendering small-scale scenes, yielding notable achievements. However, due to the constraints posed by limited computing resources and the substantial time investment required for training, the feasibility of training a NeRF for large-scale scenes on conventional consumer computers is severely limited. Consequently, efforts have been directed towards adapting NeRF for large-scale scene training with limited computing resources and time. Notable approaches in this context include Urban Radiance Fields, CityNeRF, BlockNeRF, and Mega-NeRF: Urban Radiance Fields (Rematas et al., 2021) integrates LiDAR scanning data with RGB imagery, introducing a range of LiDAR-based losses to facilitate accurate surface estimations of solid structures and volumetric structures. CityNeRF (Xiangli et al., 2021) adopts a progressive learning strategy to address the challenges associated with applying NeRF solely to single-scale image data, which enables the incorporation of multi-scale data with varying levels of detail and spatial coverage. Tancik et al. (2022) proposed BlockNeRF that implements a blocking approach, dividing complex scenes into sub-blocks and training a sub-model for each block using parallel training methods. Subsequently, relevant sub-NeRF rendering results are seamlessly integrated when generating new views. Similarly, Mega-NeRF employs the same strategy (Turki et al., 2022), dividing large-scale scenes into sub-blocks to manage computational complexity. Additionally, Mega-NeRF utilizes the inverted sphere parameterization method to address rendering challenges in unbounded scenes (Zhang et al., 2020). Furthermore, the introduction of appearance embedding vectors in Mega-NeRF helps mitigate issues stemming from inconsistent imaging conditions in outdoor images (Martin-Brualla et al., 2021).

3. Methodology

3.1 Mega-NeRF Revisiting

In general**,** to train large-scale NeRF with an ordinary consumer machine, Mega-NeRF employs the solution of dividing into several sub-block in training, it sets a centroid $\boldsymbol{N} = (X_n, Y_n, Z_n)$ for each sub-block and trains an individual NeRF model to learn the 3D scene information for each sub-block (Turki et al., 2022). Each sub-model within Mega-NeRF adopts a network architecture akin to the original NeRF, with a MLP (responsible for volume desity prediction) consisting of 8 layers of 256 channels and a fully connected ReLU layer of 128 channels and another small MLP (responsible for color prediction), as shown in Fig.4. Additionally, it incorporates an appearance embedding vector $l^{(a)}$, as introduced by Martin-Brualla et al. (2021) in NeRF-W, as an additional input to the MLP for predicting the color. Employing the Appearance embedding vector enables Mega-NeRF to effectively accommodate datasets featuring

images under diverse lighting conditions. This capability is essential for Mega-NeRF to accurately render extensive outdoor scenes characterized by varying lighting conditions. Therefore, for any given 3D position $\boldsymbol{X} = (x, y, z)$, the viewing direction $\boldsymbol{D}$ and the corresponding appearance embedding vector $l^{(a)}$, Mega-NeRF use the trained NeRF of the sub-block that is closest to the target point to predict the corresponding color $\boldsymbol{c} = (r, g, b)$ and its volume density σ. Mega-NeRF first predicts the volume density σ from 3D position $\boldsymbol{X}$ and outputs an intermediate vector l' (Eq.1). The direction $\boldsymbol{D}$, appearance embedding vector $l^{(a)}$ and intermediate vector l' are then fed into another MLP to predict color $\boldsymbol{c}$ (Eq.2). Finally, for a ray $\boldsymbol{r}(t) = \boldsymbol{o} + t\boldsymbol{d}$, Mega-NeRF only needs to predict the color and volume density along the sampling points on the ray, and integrates according to the volume fraction formula to obtain the color $\boldsymbol{C}$ (Eq. 3):

$$\sigma_i, l_i' = f^{\boldsymbol{N}}(\boldsymbol{X_i}), where\ \boldsymbol{N} = argmin\|\boldsymbol{N} - \boldsymbol{X_i}\|^2 \quad (1)$$

$$\boldsymbol{c_i} = f^{\boldsymbol{N}}\left(l_i', \boldsymbol{D_i}, l_i^{(a)}\right) \quad (2)$$

$$\boldsymbol{C} = \sum_{i=1}^{n} w_i \boldsymbol{c_i}, where\ w_i = T_i\left(1 - e^{-\Delta_i \sigma_i}\right) \quad (3)$$

$$T_i = exp\left(-\sum_{j=1}^{i} \Delta_j \sigma_j\right), \Delta_i = t_i - t_{i-1} \quad (4)$$

Moreover, in the process of scene partitioning by Mega-NeRF, there exists an overlapping region between adjacent sub-blocks. Consequently, when a sampled point falls within the overlapping region of two or more sub-blocks, Mega-NeRF employs multiple sub-models to individually predict the RGB and volume density of the point. A weighting mechanism is determined based on the reciprocal distance between the point and the center of each sub-block. Ultimately, a weighted average method is employed to derive both the color $\boldsymbol{c}$ and the volume density σ at this sampled point (Turki et al., 2022).

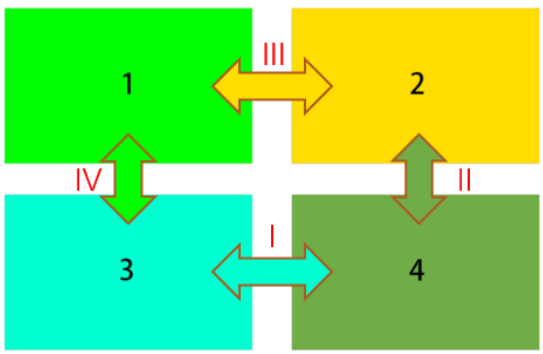

Figure 3. Dual Sub-model Optimization Strategy. Any two adjacent sub-models will be optimized considering the consistency of adjacent overlapping region. In the case shown in the figure, any sub-model will be optimized twice. For example, sub-model 1 will be jointly trained with both sub-model 2 and sub-model 3.

3.2 Extra fine-tuning of the proposed Mega-NeRF++

Due to the independent parallel train strategy of Mega-NeRF (Turki et al., 2022), the overlapping information between any two adjacent sub-model are not considered. Therefore, when training Mega-NeRF in this paper, we perform extra training after the independent parallel training, and a new loss function is used to fine tune each NeRF of sub-block.

Fine-tuning. In terms of sub-NeRF optimization, we propose **Dual Sub-model Optimization** strategies: essentially, this strategy can be considered as an incremental fine-tuning solution, as Fig. 3 illustrates. Firstly, a pair of adjacent sub-models, whose training are individually started with the original Mega-NeRF method for a certain number of epochs, is selected. Subsequently, individual training based on original photometric information (identical to Mega-NeRF) and joint training based on photometric consistency in overlapping regions (employing consistency training loss, see Eq. 7 and 8) are alternated applied until pre-set iterations are achieved. Secondly, for all other sub-model pairs with overlapping relationships, we will optimize based on their performance in the overlapping region until any sub-model pair with overlapping relationships has been optimized once. Finally, when rendering images with overlapping regions, this strategy uses the rendering results from original Mega-NeRF and the optimized model. For overlapping regions, we use the rendering results of the optimized model, while for non-overlapping regions, we continue to use the rendering results of the original Mega-NeRF. In the subsequent experimental parts, we conduct a corresponding efficacy analysis of this strategies (Tab. 3).

Hybrid Rendering. When rendering images containing overlapping regions, we employ a hybrid approach utilizing rendering results from both the original Mega-NeRF and the corresponding optimized sub-models. Non-overlapping sections of the image uses the rendering results from original Mega-NeRF, while the overlapping regions uses the rendering results from the corresponding optimized sub-models.

Consistency Training Loss. Compared to the loss of original Mega-NeRF, we proposed an improved loss. In addition to computing the Mean Squared Error (MSE) loss between the predicted RGB and ground truth, we introduce a novel regularization based on the ambiguous rendering result between any two adjacent sub-models. Our refined loss not only considers the discrepancy between predicted RGB and ground truth but also incorporates the reciprocal influence exerted by adjacent sub-models. When calculating loss for any two adjacent sub-model A and B, whose corresponding model weights (i.e., NeRF parameters) are $f^{\boldsymbol{A}}$ and $f^{\boldsymbol{B}}$, we first acquire the predicted RGB of sub-model A and B for the shared ray respectively (Eq. 5 and Eq. 6), and then estimate the corresponding loss of sub-model A and B (Eq. 7 and Eq. 8):

$$\sigma_A, \boldsymbol{c}_A = f^{\boldsymbol{A}}\left(\boldsymbol{X}, \boldsymbol{D}, l^{(a)}\right) \quad \sigma_B, \boldsymbol{c}_B = f^{\boldsymbol{B}}\left(\boldsymbol{X}, \boldsymbol{D}, l^{(a)}\right) \quad (5)$$

$$\boldsymbol{C}_A = \sum w(\sigma_A)\boldsymbol{c}_A, \boldsymbol{C}_B = \sum w(\sigma_B)\boldsymbol{c}_B \quad (6)$$

$$Loss_A = \frac{1}{n}\sum_{i=1}^{n}\left[L_{MSE}(\boldsymbol{C}_A - \boldsymbol{C}_{True}) + f_{loss}(\boldsymbol{C}_A - \boldsymbol{C}_B)\right] \quad (7)$$

$$Loss_B = \frac{1}{n}\sum_{i=1}^{n}\left[L_{MSE}(\boldsymbol{C}_B - \boldsymbol{C}_{True}) + f_{loss}(\boldsymbol{C}_B - \boldsymbol{C}_A)\right] \quad (8)$$

where
$\boldsymbol{C}_A$, $\boldsymbol{C}_B$ = color of a shared ray
$\boldsymbol{C}_{True}$ = ground truth
σ_A, σ_B = volume density of shared point
$\boldsymbol{c}_A$, $\boldsymbol{c}_B$ = predicted color of a shared point
$Loss_A$, $Loss_B$ = loss of model A and B
$f_{loss}(x) = \mu \cdot L_{MSE}(x)$, μ is a constant

Then each NeRF model is optimized based on $Loss_A$ and $Loss_B$. The following section of this paper shows more details about the experiments and compare the rendering results of the Mega-NeRF++ model with the original Mega-NeRF model.

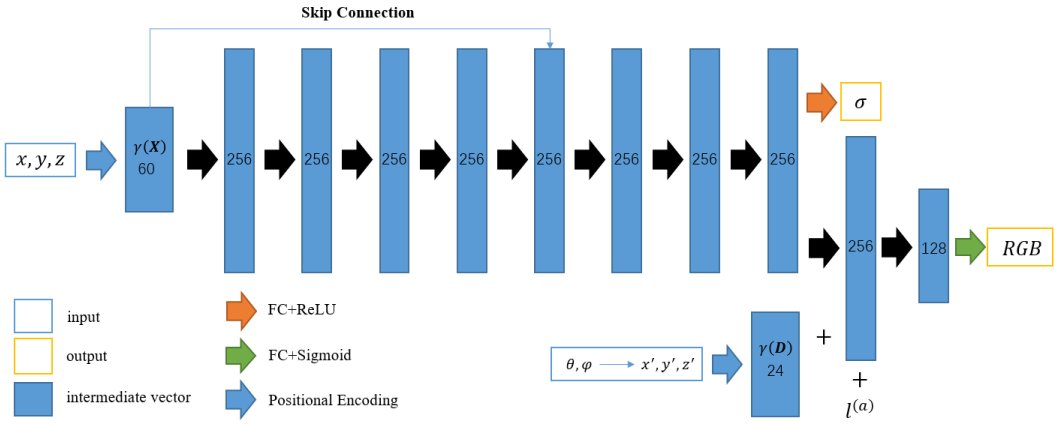

Figure 4. Vanilla Mega-NeRF architecture.

4. Experiment

To investigate the performance of our Mega-NeRF++, we primarily conduct the evaluation from two key perspectives: 1) The capability of the fine-tuned sub-model for generating higher-quality rendered images, with a particular attention on its rendering performance in the overlapping regions of adjacent sub-models. 2) The computational resources and time required for model optimization.

4.1 Experimental details

Datasets and Preprocess. For a more intuitive comparison between Mega-NeRF++ and the original Mega-NeRF, several datasets previously employed in Mega-NeRF are leveraged, including Mill 19 dataset (Turki et al., 2022) which consists of two scenes (buildings and rubble) and Quad 6k image dataset (Crandall et al., 2011) captured from a large-scale scene for SfM. Akin to Mega-NeRF, before utilizing these images for model training, COLMAP (Schonberger et al., 2016) is employed to estimate pose information for images. Subsequently, based on the pose information derived from previous method, we partition the entire scene into several sub-blocks, facilitating the subsequent parallel training of multiple sub-models.

Model architecture and parameter settings. Similar to Mega-NeRF (Turki et al., 2022), Mega-NeRF++ involves partitioning testing scenario into 8 sub-blocks (note that more sub-block can be divided for larger scenario) and sequentially training each sub-model. Each sub-model contains two MLPs. The first one is trained for predicting volume density, containing 8 layers of 256 hidden channels and a fully connected layer whose activation function is Rectified Linear Unit (ReLU). Another smaller MLP is responsible for predicting RGB, containing a fully connected layer utilizing Sigmoid activation function. Hierarchical sampling is used for both foreground NeRF and background NeRF during training, sampling 256 coarse points and 512 fine points per rays for foreground NeRF, while also sampling 128 coarse points and 256 fine points per rays for background NeRF. 1024 rays are sampled per batch for training any foreground NeRF or background NeRF. Adam optimizer is used for training and the initial learning rate is 5×10^{-4}, gradually decaying to 5×10^{-5}. Each sub-model will be trained 500000 iterations (for model construction) and another additional 10000 iterations (for appearance matching).

Training details. We follow the training strategies outlined in section 3.2. Over all the 500,000 training iterations, the starting 100,000 iterations will focus solely on training the sub-models individually, disregarding the consistency of adjacent sub-models within the overlapping regions. For the remaining optimization iterations, we adopt a regimen of alternating between individual and joint training. During individual training, only rays sampled within a single sub-module are utilized for training. Joint training incorporates rays sampled from the overlapping regions of adjacent sub-modules. The consistency of predictions across the two sub-models for the shared ray serves as new constraint to optimize the parameters of the corresponding adjacent sub-models. Throughout the experiment, the training regimen alternates every 10,000 iterations between individual and joint training. Consequently, out of the total 500,000 training iterations, we first individually train all sub-models for 100,000 iterations. Then, in the remaining training iterations, 200,000 iterations of training for all sub-models utilizing individual and joint training.

Evaluation metrics. For quantitative evaluation, we assess the performance of the Mega-NeRF++ method using metrics such as PSNR, SSIM (Wang et al., 2004), and the VGG implementation of LPIPS (Zhang et al., 2018). Additionally, we evaluated the time cost and computational resources for training on one NVIDIA GeForce RTX 4090 GPU. Tancik et al. (2022) proposed appearance matching approach in BlockNeRF to cope with inconsistencies between views, so that we also compare Mega-NeRF++ to appearance matching approach in the following experiments.

4.2 Ablation Studies

Diagnostics. To demonstrate the effectiveness of the proposed fine-tuning strategy and hybrid rendering solution, we conduct ablation studies including: the proposed Mega-NeRF++ that uses both our fine-tuning strategy and hybrid rendering solution to several ablations, Mega-NeRF++_no_merge that does not applied the hybrid rendering while the fine-tuned NeRFs are used to render images for both overlapping and non-overlapping area,

and the original Mega-NeRF. The qualitative and quantitative comparison are shown in Fig. 5 and Tab. 1, respectively. We can find that our Mega-NeRF++ can basically outperform the other two variants of Mega-NeRF++_no_merge and original Mega-NeRF, Fig. 5 clearly shows that our method is always higher than the other two in terms of PSNR, which is in general consistent to Tab. 1. This proves that our joint training based on overlapping regions of adjacent sub-models and the hybrid rendering results from original Mega-NeRF is able to significantly improving novel view synthesis performance.

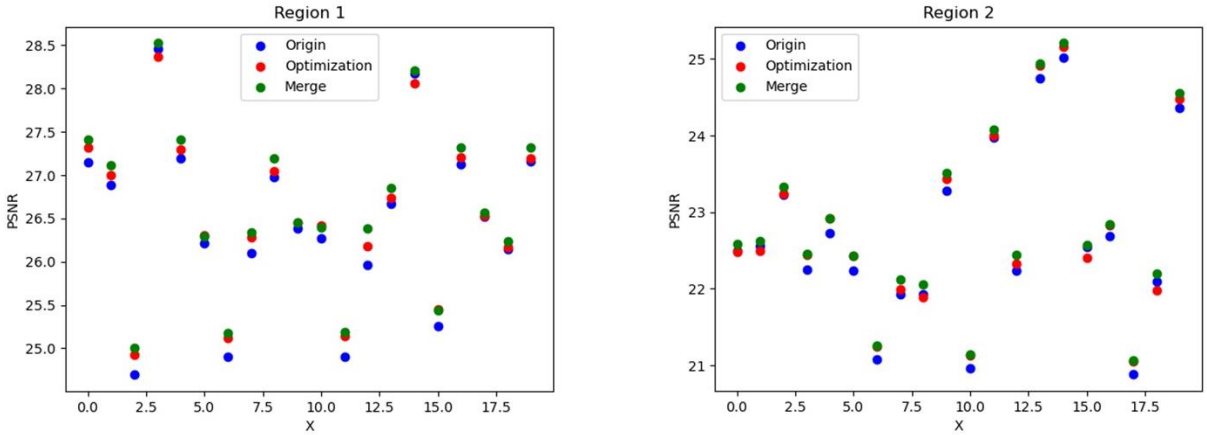

Figure 5. Ablation study of PSNR value. PSNR in region 1 (left) and PSNR in region 2 (right). We conduct rendering for 20 images within region 1 and region 2, and assess the corresponding their PSNR values.

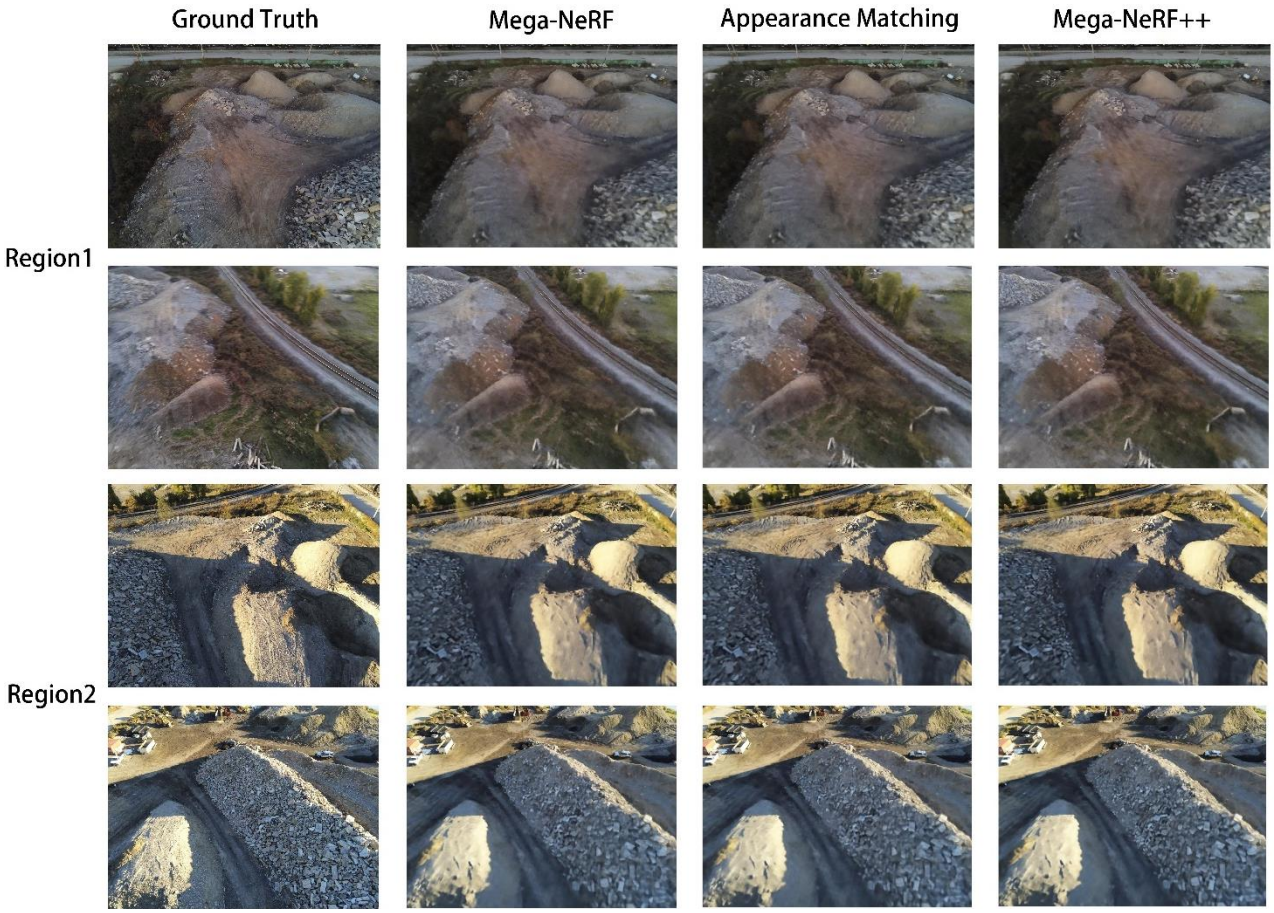

Figure 6. Rendering results. In every scene, we utilize each method to render images in both overlapping and non-overlapping regions, depending on how the scene was partitioned during training with Mega-NeRF++. Mega-NeRF++ effectively enhances the image rendering quality in overlapping regions while preserving the rendering capability in non-overlapping regions.

	Mega-NeRF++			Mega-NeRF++_no_merge			Mega-NeRF		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
Region1	**26.638**	**0.639**	0.513	26.560	0.635	0.513	26.457	0.633	**0.511**
Region2	**22.818**	**0.538**	0.533	22.744	0.532	0.538	22. 661	0. 531	**0.531**

Table 1. Ablation studies. We compare Mega-NeRF++ to its multiple ablations. Mega-NeRF++_no_merge doesn’t combine the rendering result of original Mega-NeRF and our optimized NeRFs.

	Mega-NeRF++			Mega-NeRF			Appearance matching		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
Region1	**26.638**	**0.639**	0.513	26.457	0.633	**0.511**	26.479	0.634	**0.511**
Region2	**22.818**	**0.538**	0.533	22.661	0.531	**0.531**	22.638	0.530	**0.531**

Table 2. Optimized model evaluation. Each method is employed on each scene for novel view synthesis. PSNR, SSIM and LPIPS are calculated separately for each method on each scene. Mega-NeRF++ consistently outperforms other methods, consistently yielding superior rendering results with the highest PSNR and SSIM.

4.3 Comparison with other methods

We present the evaluation results of all methods in Table. 2. Additionally, we present partial rendering results obtained from each method when rendering from a new perspective in each scene in Figure. 6. Mega-NeRF++ demonstrates superior performance compared to Mega-NeRF and other methods, achieving better rendering results while maintaining comparable training and rendering times to Mega-NeRF.

Discussion. Although the current training strategy has demonstrated effective optimization, our training strategy requires two models to be trained simultaneously in each training iteration, which significantly increases the time cost and computational resources required for model training. Additionally, there still remain several untested and unverified training strategies due to the limitations on time consumption and computational resource. Examples include simultaneously training all sub-models of the entire scene and optimizing a single sub-model while considering all prediction results from other overlapping sub-models, allowing for synchronous adjustment of all sub-models. Additionally, this paper currently employs only Mean Squared Error (MSE) losses to calculate losses in the overlapping regions between adjacent sub-models during training, with a weight value of μ=1. Therefore, we will try to solve the existing problems in the following work and try more training strategies.

5. Conclusion

This paper presents an improved method, Mega-NeRF++, for boosting the original large-scale Mega-NeRF based on the consistency of overlapping regions between adjacent sub-models. By taking the consistency of rendering results across adjacent sub-models within overlapping regions as a novel constraint, we incorporate discrepancies between predicted rendering result and ground truth, as well as discrepancies between predicted rendering result from adjacent sub-models for the shared ray during training, establishing a more effective large-scale NerF fine-tuning approach. This method successfully minimizes deviations between Mega-NeRF++ predicted rendering results and ground truth, while mitigating color inconsistency errors that may arise during rendering in overlapping regions of adjacent sub-models. Furthermore, ours Mega-NeRF++ consistently achieves the best rendering results compared to Mega-NeRF and other SOTA methods employed for novel view synthesis.

In the future, we would like to focus on exploring and selecting optimal training strategies, refining the selection of loss functions for overlapping regions, and determining appropriate weight μ for the two types of losses (loss between model-predicted values and ground truth, and loss in overlapping regions). Furthermore, we will leverage established model fast training strategies to optimize model, while concurrently minimizing the time cost and computational resources overhead associated with our proposed methodology.

Acknowledgements

This work was jointly supported Natural Science Foundation of Hubei Province, China (2022CFB727) and National Natural Sci ence Foundation of China (42301507).

References

Mildenhall B., Srinivasan P.P., Tancik M., Barron J.T., Ramamoorthi R., Ng R., 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In: Proceedings of the European Conference on Computer Vision (ECCV), doi.org/10.48550/arXiv.2003.08934.

Zhang K., Riegler G., Snavely N., Koltun V., 2020. NeRF++: Analyzing and Improving Neural Radiance Fields. Computer Vision and Pattern Recognition (cs.CV), doi.org/10.48550/arXiv.2010.07492

Kerbl B., Kopanas G., Leimkuhler T., Drettakis G., 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. ACM Transactions on Graphics 42.4, pp. 1-14.

Barron J.T., Mildenhall B., Verbin D., Srinivasan P.P., Hedman P., 2022. Mip-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5470-5479.

Barron J.T., Mildenhall B., Tancik M., Hedman P., Martin-Brualla R., Srinivasan P.P., 2021. Mip-NeRF: A Multiscale Representation for Anti-Aliasing Neural Radiance Fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 5855-5864.

Jia Z., Wang B., Chen C.H., 2023. Drone-NeRF: Efficient NeRF Based 3D Scene Reconstruction for Large-Scale Drone Survey. Image and Vision Computing, 143, 104920.

Chen S.C., Williams L., 1993. View Interpolation for Image Synthesis. In: Seminal Graphics Papers: Pushing the Boundaries, Volume 2, pp. 423-432, doi.org/10.1145/166117.166153

Muller T., Evans A., Schied C., Keller A., 2022. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. ACM transactions on graphics (TOG), 41(4), pp. 1-15. https://doi.org/10.48550/arXiv.2201.05989

Turki H., Ramanan D., Satyanarayanan M., 2022. Mega-NeRF: Scalable Construction of Large-Scale NeRFs for Virtual Fly-Throughs. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 12922-12931

Martin-Brualla R., Radwan N., Sajjadi M.S.M., Barron J.T., Dosovitskiy A., Duckworth D., 2021. NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7210-7219.

Schönberger J.L., Frahm J.M., 2016. Structure-from-Motion Revisited. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4104-4113

Tancik M., Casser V., Yan X.C., Pradhan S., Mildenhall B.P., Srinivasan P., Barron J.T., Kretzschmar H., 2022. Block-NeRF: Scalable Large Scene Neural View Synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8248-8258.

Xu N.L., Qin R.J., Huang D.B., Remondino F., 2023. Enabling Neural Radiance Fields (NeRF) for Large-scale Aerial Images – A Multi-tiling Approach and the Geometry Assessment of NeRF. arXiv preprint arXiv:2310.00530. https://arxiv.org/ftp/arxiv/papers/2310/2310.00530.pdf

Crandall D., Owens A., Snavely N., Huttenlocher D., 2011. Discrete-continuous optimization for large-scale structure from motion. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3001-3008.

Schonberger J.L., Zheng E.L., Pollefeys M., Frahm J.M., 2016. Pixelwise view selection for unstructured multi-view stereo. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 501-518.

Wang Z., Bovik A.C., Sheikh H.R., Simoncelli E.P., 2004. Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing, 13(4), pp. 600-612.

Zhang R., Isola P., Efros A.A., Shechtman E., Wang O., 2018. The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp. 586-595.