2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Blending Distributed NeRFs with Tri-stage Robust Pose Optimization

Baijun Ye*[1,2], Caiyun Liu*[3], Xiaoyu Ye[2,4], Yuantao Chen[2,4], Yuhai Wang[5], Zike Yan[2], Yongliang Shi[2†], Hao Zhao[2], Guyue Zhou[2]

***Abstract*— Due to the limited model capacity, leveraging distributed Neural Radiance Fields (NeRFs) for modeling extensive urban environments has become a necessity. However, current distributed NeRF registration approaches encounter aliasing artifacts, arising from discrepancies in rendering resolutions and suboptimal pose precision. These factors collectively deteriorate the fidelity of pose estimation within NeRF frameworks, resulting in occlusion artifacts during the NeRF blending stage. In this paper, we present a distributed NeRF system with tri-stage pose optimization. In the first stage, precise poses of images are achieved by bundle adjusting Mip-NeRF 360 with a coarse-to-fine strategy. In the second stage, we incorporate the inverting Mip-NeRF 360, coupled with the truncated dynamic low-pass filter, to enable the achievement of robust and precise poses, termed Frame2Model optimization. On top of this, we obtain a coarse transformation between NeRFs in different coordinate systems. In the third stage, we fine-tune the transformation between NeRFs by Model2Model pose optimization. After obtaining precise transformation parameters, we proceed to implement NeRF blending, showcasing superior performance metrics in both real-world and simulation scenarios. Codes and data will be publicly available at https://github.com/boilcy/Distributed-NeRF.**

I. INTRODUCTION

The field of large-scale scene modeling has garnered considerable scholarly interest, notably in light of the recent advent of NeRF. This emergence is primarily attributed to NeRF's capacity for achieving photorealistic rendering while maintaining a compact model structure. This has brought NeRF to the forefront of attention in drone mapping [22] and navigation [1].

In the domain of large-scale scene representation using NeRF, there exist primarily two predominant methodologies. 1) **Batch learning**, as exemplified by Bungee-NeRF [23], requires substantial computational resources, especially for the exhaustive sampling and fitting of the entire scene. To mitigate the substantial computational requirements associated with batch training on large-scale data, there exist methods [20] [22] of subdividing a vast scene into several smaller scenes. The prerequisite for this approach is that all images to be trained must share a common coordinate system. In GNSS denied environment, we often rely on methods such as Simultaneous localization and mapping (SLAM) [3] or structure-from-motion (SfM) [10] [18] for pose estimation. For large-scale urban scenes, SLAM inevitably accumulates errors over time, making it unreliable for precise pose information. Meanwhile, SfM undoubtedly demands substantial computational resources. 2) **Incremental learning** may lead to insufficient scene representation due to forgetting. Besides, current approaches using explicit encoding methods like grid [28] [17] and octree[25] for real-time performance, which face the challenge of exponentially expanding encoding components as the scene scale increases, leading to substantially increased storage requirements.

We aim to devise a large-scale NeRF system under computational constraints. Our batch learning approach focuses on mitigating the forgetting problem inherent to incremental learning-based NeRF in expansive urban settings. Additionally, it efficiently handles the computational complexities arising from the increased data volumes in these scenarios. However, we encounter a **challenge** in achieving precise registration results for distinct NeRFs acquired by different agents using heterogeneous coordinate systems.

To address these limitations, we put forward a distributed NeRF framework with a tri-stage pose optimization methodology. We use Mip-NeRF 360 [2] as backbone due to its good anti-aliasing effect. In the first stage, drawing inspiration from BARF [9], we enhance Mip-NeRF 360 to implement the bundle-adjusting Mip-NeRF 360. A coarse-to-fine approach is employed for the conjoint optimization of scene representation and pose. In the second stage, taking cues from LATITUDE [27], we leverage the principles of Truncated Dynamic Low-pass Filter (TDLF) to refine the inverting Mip-NeRF 360, termed iMNeRF. This method is similar to blurring images to make the optimization process more robust, thereby enabling pose optimization for Frame2Model. Subsequently, we employ a co-view region retrieval method (detailed in section IV-C) to search the most analogous images across diverse NeRF instances, subsequently identifying their associated poses. Given the associated poses, we employ the iMNeRF to optimize these poses by photometric losses among rendering images and observed images, thereby obtaining reliable Frame2Model transformations. In the third stage, we obtain a rough Model2Model transformation between NeRFs by different Frame2Model transformations. Then, we project the different NeRF models onto a unified coordinate system and further optimize the relative transformations among NeRFs using the rendered images as observation, that is through the Model2Model optimization to obtain the precise transformations among NeRFs. Utilizing the tri-stage pose optimization, we implement the NeRF blending and get better performance. To validate our method, we have concurrently released both real-world and simulation datasets, demonstrating the superiority of our

* Equal contribution. [1]IIIS, Tsinghua University, [2]Institute for AI Industry Research (AIR), Tsinghua University, [3]Peking University, [4]CUHK(SZ), [5]University of Southern California.
† Corresponding author. shiyongliang@air.tsinghua.edu.cn
Sponsored by Tsinghua-Toyota Joint Research Fund (20223930097).

approach. In summary, our contributions are as follows:

- We introduce a distributed NeRF framework for large-scale urban environments, incorporating a tri-stage pose optimization. This is specifically utilized during NeRF blending to address the issue of misalignment caused by inaccuracies in registration.
- We implement bundle-adjusting Mip-NeRF 360, facilitating a conjoint optimization of poses and scene representation. Building upon this, an enhanced Frame2Model pose estimation technique, iMNeRF, is proposed. This not only optimizes registration results but also provides a dependable preliminary Model2Model transformation.
- We release a comprehensive dataset that combines amalgamates both real-world and simulation data. The superior blending and registration results of our methodology are distinctly showcased.

II. Related Work

A. NeRF2NeRF Registration

NeRFs are optimized from accurately posed images. In city reconstruction with UAVs, GPS or RTK-equipped UAVs are commonly employed. However, the initial poses obtained from GPS/RTK often require refinement using SfM [18] techniques for each sub-area to achieve high-quality large-scale reconstructions. This refinement process will result in a coordinate system with arbitrary global positions that are specific to each NeRF submodule.

While extensive research exists on registration methods for explicit representations such as point clouds, there is a notable lack of studies addressing NeRF2NeRF registration for implicit fields. Recent approaches like nerf2nerf [5] and Zero-NeRF [16] rely on extracting surfaces from NeRF representations for pairwise registration. NeRFuser [4] applies an off-the-shelf SfM method on re-rendered images to get transformation between different NeRFs. However, these methods primarily rely on traditional geometry-based methods. The storage of explicit geometric information grows infinitely as the scene expands, while implicit maps can directly store the color and geometric information of the scene through compact neural network representations, achieving registration more concisely. Further, their applicability is mainly limited to object-level or small-scene NeRFs and may not be suitable for large-scale city scene reconstructions. Our method utilizes NeRF to directly optimize transformations between NeRFs, achieving state-of-the-art (SOTA) performance on large-scale city scenes.

B. NeRF for Large-scale Scene

The neural radiance field has shown immense potential in representing large-scale scenes. Some methods have been proposed to leverage this technology in city scene reconstruction. Block-NeRF [20] spatially decomposes the scene into independently trained NeRFs. It employs Inverse Distance Weighting (IDW) and visibility prediction to calculate their contributions to the overall scene representation. Mega-NeRF [22] goes one step further by applying a geometric clustering algorithm to partition training pixels into different NeRF submodules. Switch-NeRF [14] leverages a Sparsely Gated Mixture of Experts (MoE) approach for end-to-end learning-based scene decomposition. However, these approaches require known poses in each scene partition under a common global coordinate system, which is impractical for large-scale scenes where acquiring accurate poses for the entire area at once is unfeasible. Grid-NeRF [24] and GP-nerf [26] use efficient feature grids for scalability but still face network capacity limitations for extremely large scenes. Our method introduces the possibility of high-quality NeRF reconstruction for infinitely large-scale scenes.

III. Formulation

Our goal is to accurately register distributed NeRF models to a global coordinate and make full use of all NeRF models to render images. Pose optimization is divided into three stages: First, local poses are refined while training, e.g. agent i collects image set $\{I_i^{C_k}\}$ for NeRF model $\mathcal{F}_i$, where C_k presents a camera model and k is the camera index in local dataset. Each image has a corresponding imperfect camera pose $\{T_i^{C_k}\}$ obtained from COLMAP [18]. To match the scene images and poses more accurately, bundle adjusting is needed, which can be expressed as eq. (1),

$$\min_{T_i^{C_1},T_i^{C_2},\ldots,T_i^{C_m},\boldsymbol{\Theta}} \sum_{k=1}^{m} \left\| \hat{I}(\mathcal{F}_i(T_i^{C_k};\boldsymbol{\Theta})) - I_i(\mathbf{x}) \right\|_2^2. \quad (1)$$

In the second stage, Frame2Model pose optimization is implemented through a pose gradient updating method. Since different agents define their own local coordinates, a transformation matrix from local to global is required to achieve seamless blending between models. Translation matrix T_{ij} between $\mathcal{F}_i$ and $\mathcal{F}_j$ should satisfy eq. (2),

$$\begin{aligned} T_{ij}^* = \min_{T_{ij}} & \sum_{k=1}^{n_i} \left\| \hat{I}\left(\mathcal{F}_j(T_{ij}^{-1}T_i^{C_k})\right) - I_i^{C_k} \right\|_2^2 \\ & + \sum_{k=1}^{n_j} \left\| \hat{I}\left(\mathcal{F}_i(T_{ij}T_j^{C_k})\right) - I_j^{C_k} \right\|_2^2 \end{aligned} \quad (2)$$

In the third stage, Model2Model pose optimization is performed before NeRF blending, with query camera pose in a specified coordinate, e.g. T_i in coordinate of agent i, we again optimize T_{ij} with previous T_{ij}^* as a initial value:

$$T_{ij}^* = \min_{T_{ij}} \left\| \hat{I}\left(\mathcal{F}_j((T_{ij}^{-1}T_i)\right) - \hat{I}\left(\mathcal{F}_i(T_i)\right) \right\|_2^2. \quad (3)$$

IV. Method

A. System Overview

As shown in Fig.1, the system includes two tasks: NeRF registration and NeRF blending. In the NeRF registration stage, rather than directly addressing equation 2, we adopt a tri-stage pose optimization. Initially, bundle-adjusting Mip-NeRF 360 is introduced to achieve joint optimization of both scene representation and poses of input images. Subsequently, employing the co-view region retrieval approach, we identify similar image pairs like $(I_i^{C_k}, I_j^{C_t})$ in each agent's

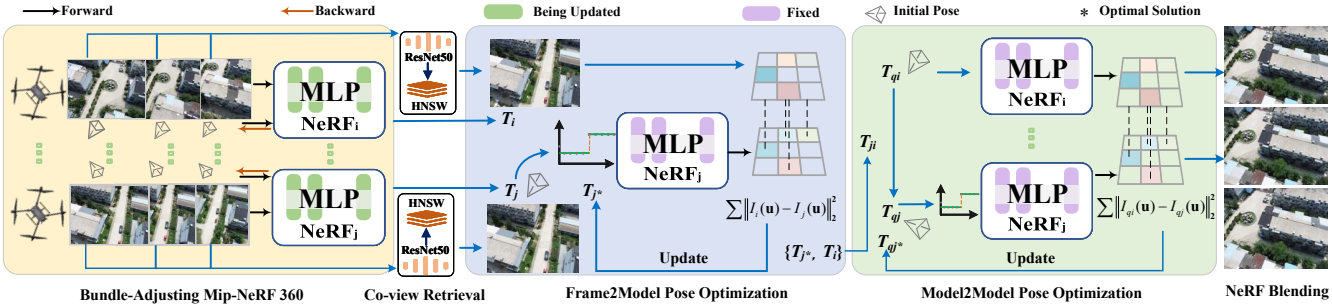

Fig. 1. System Overview. Each agent trains a bundle-adjusting Mip-NeRF 360. Following this, co-view region retrieval provides an initial value for Frame2Model pose optimization. A final Model2Model pose optimization is performed before blending to achieve a seamless fusion of NeRFs.

dataset, along with their local coordinate poses $(T_i^{C_k}, T_j^{C_t})$. Beginning with $T_j^{C_t}$ as the initial guess of $T_j^{C_k}$, the pre-trained NeRF model minimizes the disparity between the rendered image $I_j^{C_k}$ and the $I_i^{C_k}$. A coarse-to-fine strategy ensures avoidance of local optima when performing Frame2Model optimization, and Frame2Model registration results are obtained. Thirdly, coarse relative transformations between NeRFs obtained from Frame2Model registration results are regarded as prior to transform sampled points of different NeRFs to the unified coordinate system. After determining $T_j^{C_k}$, we ascertain the transformation matrix T_{ij}. Initiated with a query camera pose $T_i^{C_q}$, optimization before rendering entails iterating the previous pose optimization with fewer steps and excluding TDLF. After registration, the contribution of each NeRF can be determined using inverse distance weighting (IDW). In the far right of the figure, from top to bottom, the first two images are rendered by two distributed NeRFs from the same pose. The bottom most image is the result of blending.

B. Bundle-Adjusting NeRF for Pose Refinement

As a compact representation of scenes, Mip-NeRF 360 adeptly addresses aliasing issues during rendering at varying resolutions and the ambiguity associated with reconstructing 3D content from 2D images in large-scale unbounded environments. However, even though the camera extrinsic obtained from SfM or RTK are precise, errors are inevitable, which consequently affects the quality of the scene representation. Therefore, given the imperfect camera extrinsic and the corresponding images, our method is to simultaneously optimize the scene representation and camera extrinsic by minimizing a weighted combination of warp loss [9], distortion loss [2] and proposal loss [2]:

$$\Theta^*, T^* = \mathcal{L}_{warp}(\cdot) + \lambda \mathcal{L}_{dist}(\cdot) + \mathcal{L}_{prop}(\cdot), \tag{4}$$

which are respectively responsible for the alignment of pose, scene geometry and training efficiency.

During training, applied coarse-to-fine optimization mainly emphasise positional encoding. Mip-NeRF 360 constructs an integrated positional encoding (IPE) representation of the volume covered by each conical frustum cast from rendering pixel, instead of constructing positional encoding (PE) features for sampled points along the casted line. While this approach yields effective anti-aliasing performance, pose optimization leveraging this model may also be trapped in a local optimum due to the disproportionate impact of high-frequency components on the gradient. Drawing inspiration from BARF [9], we introduce a coarse-to-fine optimization strategy by adding a weight layer $\omega(\alpha)$, where the weight ω_k of k-th frequency component is:

$$\omega_k(\alpha) = \begin{cases} 0 & \text{if } \alpha < k \\ \frac{1-\cos((\alpha-k)\pi)}{2} & \text{if } 0 \le \alpha - k < 1 \\ 1 & \text{if } \alpha - k \ge 1 \end{cases} \tag{5}$$

Let $\alpha \in [0, L]$ be modulated in relation to the training progression, as illustrated in Fig.2(a). When $\omega_k(\alpha)$ is set to 0, contributions to the gradient from the k-th frequency component and above are effectively neutralized. Starting with $\alpha = 0$, encodings are progressively activated until full positional encoding is achieved at $\alpha = L$. This approach facilitates our model's initial alignment with a smoother signal, paving the way for the subsequent capture of a high-resolution scene representation.

C. Co-view Region Retrieval

Assuming the data collection is guaranteed to have overlapping areas, we present a co-view region retrieval approach that leverages the capabilities of deep learning for feature extraction and efficient similarity search methods. We employ the pre-trained ResNet-50 model, leveraging its established proficiency in extracting intricate feature hierarchies from images [6]. By adapting the ResNet-50 architecture and omitting its final fully connected layer, we seek a high-dimensional representation of images, summarizing their fundamental visual attributes concisely [8]. Consequently, we extract feature vectors from all images of datasets for all distributed NeRFs, and employ these vectors to establish an index within the FAISS library [7]. The Hierarchical Navigable Small World (HNSW) [12] [13] search algorithm is adopted for its for its balanced performance in speed, precision, memory efficiency, and scalability, especially pertinent to the retrieval of high-dimensional vectors. When it comes to similarity searches, the extracted feature vector of a given query image is utilized to probe the FAISS index, identifying the pairs of images with the highest similarity from the dataset based on the L2 distance between feature vectors.

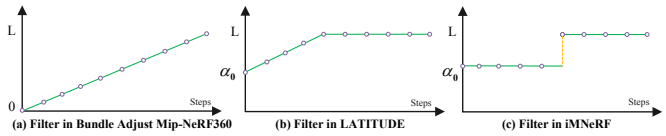

Fig. 2. Different Dynamic Low-pass Filter.

D. Frame2Model Pose Optimization

After finding the co-view region across multiple agents, the subsequent step is computing relative transformations between distributed NeRFs. Given a specific set of image pairs $(I_i^{C_k}, I_j^{C_t})$, we extract the associated poses $T_i^{C_k}$ and $T_j^{C_t}$ which are previously optimized during the training stage of bundle-adjusting Mip-NeRF 360. Taking camera C_k of $(I_i^{C_k}, T_i^{C_k})$ for example, we use iMNeRF to locate this camera in NeRF $\mathcal{F}_j$. $I_i^{C_k}$ serves as an observation and $T_j^{C_t}$ serves as an initial guess of $T_j^{C_k}$.

iMNeRF aims to optimize T_j by minimizing the photometric loss between the observed image I_i and the image $\hat{I}_j^{C_k}$ rendered from our NeRF $\mathcal{F}_j$. To ensure smoother optimization and faster convergence, we perform our optimization on the tangent plane. This can be formulated as follows:

$$\xi^* = \min_{\xi \in \mathfrak{se}(3)} \left\| \hat{I}\left(\mathcal{F}_j(\xi; T_j)\right) - I \right\|_2^2 \tag{6}$$

$$T_j^{C_k} = \exp(\xi^*) T_j \tag{7}$$

Due to our NeRF's outstanding ability to render without image aliasing across various resolutions, our approach remains competent in producing high-quality images, even amidst altitude changes from a drone's top-down perspective. This broadens the potential applications of our model. IPE increases the accuracy of high-frequency position encoding without eliminating the scene's high-frequency information, leading to two pivotal design decisions in our method. First, the coarse-to-fine strategy, supported by [9] and [27], helps avoid local optima. Drawing from LATITUDE [27], we incorporated a TDLF to IPE to suppress invalid outputs at high frequency, such as artifacts. Second, there's no need to sequentially modify high-frequency encoding. Instead of the filter in LATITUDE (Fig.2(b)), we only need to eliminate some high frequencies at the beginning and release all high-frequency information once reaching an iteration threshold. The weight ω_k is:

$$\omega_k(\alpha) = \begin{cases} 0 & \text{if } \alpha < k \\ 1 & \text{if } \alpha \geq k \end{cases} \tag{8}$$

where $\alpha \in \{\alpha_0, L\}$ changes if progress exceeds a threshold $\tau = 0.8$ and the empirical value for α_0 is 0.6 (Fig.2(c)).

E. Model2Model Registration

Once the $T_j^{C_k}$ is derived from Frame2Model process, the relative transformation can be formulated as $T_{ji} = T_j^{C_k}(T_i)^{-1}$. A parallel process is employed to locate camera C_t of (I_j, T_j) within NeRF $\mathcal{F}_i$, yielding the transformation $T_{ij} = T_i^{C_t}(T_j)^{-1}$.

This method is reiterated across several image pairs displaying relatively high similarity. For each iteration, we compute $|T_{ji}T_{ij} - E|$, where E is the identity matrix. The transformation T_{ji} that offers the minimum value for this metric is selected as the preliminary estimation of relative poses between the distinct NeRFs.

However, for achieving high-fidelity rendering, a more refined transformation matrix is necessary when considering the query camera C_q, as noise in the observational data including inaccurate pose or blurry captures can compromise the accuracy of relative pose after the Frame2Model pose optimization.

Therefore, we conduct a final optimization before blending. Given the pre-trained model of NeRF $\mathcal{F}_i$, we utilize it to generate a set of synthesized images $\{I_i^{C_q}\}$ by querying it with sampled camera pose $\{T_i^{C_q}\}$, while $\{I_j^{C_q}\}$ is rendered by pose $\{T_j^{C_q}\}$ that is computed by $\{T_i^{C_q}\}$ and previous T_{ji}. Again we perform iMNeRF to solve T_{ji}^*. Upon obtaining the ultimate T_{ji}^*, the rendered images align closely, serving as a cornerstone for facilitating the blending of large-scale scenes.

F. NeRF Blending

We use the modified inverse distance weighting (IDW) method based on [4] to carry out NeRF blending. A given query camera will generate samples in multiple models and IDW considers combining the 3D points sampled by rays casted in different coordinate systems, rendering in the same frame in the context of accurate registration result. Although Mip-NeRF 360 performs sampling based on Gaussian, we can interpret the sampling results as the midpoints of intervals and subsequently aggregate them along the same ray. After merged samples $\left\{\left(\bar{t}_k, \bar{\delta}_k\right)\right\}_k$ (k is the point index on merged ray) are obtained from outputs of NeRF $\mathcal{F}_i$ and $\mathcal{F}_j$, distance from sample points to each NeRF origin are known. We selectively adjust contribution by weight w_k from NeRF points-wise based on the distance ratio.

$$w_{i,k} = \begin{cases} 1 & \text{if } \frac{d_{i,k}}{d_{j,k}} < 0.5 \\ \frac{d_{j,k}^5}{d_{i,k}^5 + d_{j,k}^5} & \text{if } 0.5 \leq \frac{d_{i,k}}{d_{j,k}} < 2 \\ 0 & \text{if } \frac{d_{i,k}}{d_{j,k}} \geq 2 \end{cases} \tag{9}$$

$w_{j,k}$ is determined in the same manner.

V. EXPERIMENTS AND ANALYSIS

A. Implementation Details

We implement bundle-adjusting Mip-NeRF 360, iMNeRF, and NeRF Blending based on JAX. The real scene is around $120 \times 60\ m^2$. We train the scene for 250,000 iterations. The MLPs configuration of Mip-NeRF 360 configuration is slightly changed, including a proposal MLP with 4 hidden layers and 256 units, a MLP with 8 layers and 1024 hidden units. We resize the images to 1216 × 912 pixels and randomly cast rays during the training steps. Adam optimizer is adopted with an initial learning rate of 2×10^{-3} decaying exponentially to 2×10^{-5} for scene reconstruction and 2×10^{-2} to 6×10^{-3} for pose optimize. In order to implement

a coarse-to-fine strategy, we add a deterministic layer after IPE, which applies the weights on the encoded feature. Experiments are conducted on a single NVIDIA RTX3090 GPU with 24GB of memory.

B. Dataset

We evaluate our method using our Distributed Urban Minimum Altitude Dataset (DUMAD), comprising both real-world and virtual scenes. The virtual scene is generated using the AirSim [19] simulation built on Unreal Engine. To simulate distributed scenarios, we deploy three drones in both scenes, each with distinct starting origins resulting in disparate coordinate systems. Further, To enhance data realism, we incorporate two authentic city scene models within Unreal Engine, faithfully replicating urban environments resembling New York and San Francisco. Owing to space constraints in this paper, the visualization of the simulation experiments will be viewed in the attached supplementary video.

The real-world data is collected using a DJI M300RTK drone with traditional oblique photography settings at different times. We employ COLMAP [18] for pose refinement, dividing the entire scene into four areas. As far as we know, we are the first to release a dataset tailored for training distributed NeRF models in large-scale city scenes.

C. Bundle-Adjusting for NeRF Representation

In the real world, there are inherent errors in pose estimation, which are not present in simulation environments. Therefore, comparing our bundle-adjusting Mip-NeRF 360 with the original Mip-NeRF 360 in a simulation setting is not meaningful. To provide a meaningful comparison, we benchmark against NGP-based [15] NeRFacto [21] (NeRFacto also operates in a joint optimization mode of scene representation and pose), which serves as the backbone of NeRFuser and currently the most widely used NeRF method in real-world scenarios. As shown in Table I, our method outperforms in terms of quantitative results. This ensures that we can provide more accurate and reliable prior information for the subsequent registration stage.

TABLE I
PERFORMANCE OF NERF.

Scene	Methods	PSNR↑	SSIM↑	LPIPS↓
Real	NeRFacto	21.95	0.699	0.302
	Mip-NeRF 360	26.25	0.801	0.224
	Ours	**27.61**	**0.855**	**0.176**

D. Frame2Model Registration

In this section, current SOTA PE-based LATITUDE [27] and hash-encoding-based iNGP [11] are compared with our method on varying levels of initial translation error in altitude on both real-world and simulation datasets.

As shown in Table II, errors initially occur during the drone's ascent and descent, due to changes in the captured scene's resolution. A noticeable decline in LATITUDE's performance can be observed. This is attributable to the fact

Fig. 3. Frame2Model registration results: the LATITUDE[27] experiences registration failure due to aliasing, resulting in confusion in the merged observation and rendering images. While the iNGP[11] shows notable accuracy improvements, noticeable floating artifacts in the highlighted area are observed, potentially inducing registration errors. Ours effectively avoids aliasing, ensuring superior rendering quality and registration accuracy.

that LATITUDE's network architecture, derived from Mega-NeRF [22], cannot handle aliasing issues under different resolutions, leading to diminished performance. The previous centimeter-level registration achievement was due to the errors occurring while the drone was in level flight, with consistent imaging resolution for the scene.

For iNGP, its explicit encoding approach contributes to excellent overall image quality. However, some floaters affect the registration results. As can be observed, the method achieves better results with minor errors. Yet, the storage of model in this paper is 2.5GB, and it will exponentially increase with the scale of the scene.

Our method, termed iMNeRF, overall exhibits superior performance, especially in real-world data scenarios. This is mainly attributed to our refinement of the joint optimization of scene and pose in Mip-NeRF 360, which enhances the representation of the scene itself. Incorporating the TDLF in our method enhances robustness by suppressing high-frequency artifacts, particularly in managing perspectives from both higher and lower altitudes, outperforming other methods. Furthermore, the storage size of our model is only 103MB.

Fig.3 shows the registration process of these methods when the initial error is 10 meters. LATITUDE is incapable of addressing the aliasing issues resulting from resolution variations. Conversely, the presence of floaters in iNGP adversely impacts the quality of localization. Furthermore, solely relying on entire frequency domain information for pose optimization can inadvertently lead to local optima. In contrast, our approach excels at addressing challenges arising from image aliasing and complex signal interferences. As observed in the figure, initially, we suppressed a portion of the high-frequency information, leading to a reduction in image quality. However, utilizing both high and low-frequency information toward the end restored the image quality to a satisfactory level.

E. NeRF Blending with Model2Model Registration

In a comprehensive evaluation of both real-world and simulation datasets, we compared with the blending results

TABLE II

TRANSLATION AND ROTATION ERRORS OF NERF-BASED REGISTRATION RESULTS

Scene	Method	4		-4		10		-10	
		Rotation(°)	Translation(m)	Rotation(°)	Translation(m)	Rotation(°)	Translation(m)	Rotation(°)	Translation(m)
Sim	LATITUDE	0.9720	0.1200	7.9650	0.0210	12.602	0.3550	2.6820	0.4630
	iNGP	**0**	**0.0040**	**0**	0.0768	0.0470	4.2940	0.0720	8.3580
	Ours(iMNeRF)	0.0005	0.0291	0.0005	**0.0118**	**0.0005**	**0.0189**	**0.0005**	**0.0214**
Real	LATITUDE	2.0710	2.4040	1.3260	0.7950	3.8430	6.9930	4.1260	5.2150
	iNGP	0.0167	0.5270	**0**	**0.0040**	0.0860	5.3230	0.0160	2.9510
	Ours(iMNeRF)	**0.0005**	**0.0111**	0.0005	0.0058	**0.0007**	**0.0063**	**0**	**0.0151**

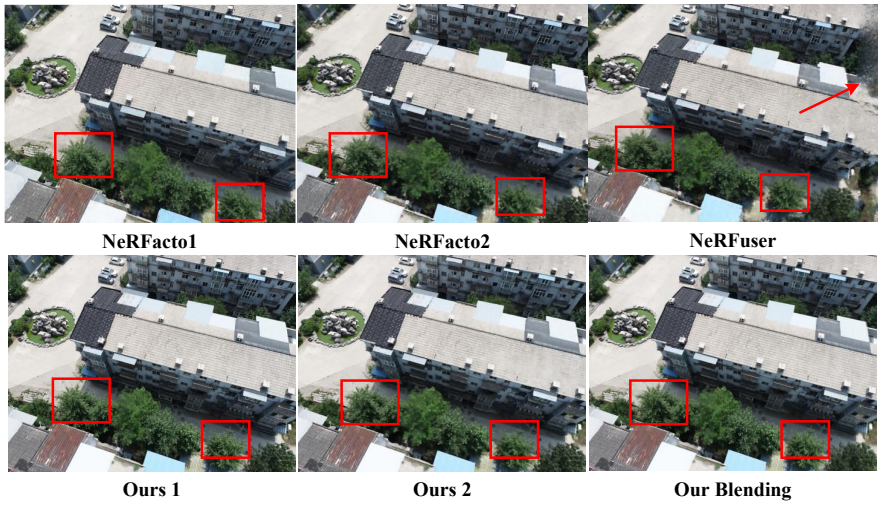

Fig. 4. Blending Results in Real World. The first two columns represent the rendering results of distributed NeRFs trained by different methods. Due to data loss at the edges of the UAV trajectory, aliasing occurs in the rendering as indicated. When comparing renderings from the two blended NeRFs, our method significantly enhances areas where individual NeRF renderings perform poorly.

of NeRFuser. Specifically, our approach takes posed images in distinct NeRFs as input and produces NeRF rendering and blending results as output under the same setting as NeRFuser. Quantitative results, as detailed in Tables III and Table IV, highlight a significant observation: for the same city-scale scenes, the individual NeRF reconstruction results by NeRFuser consistently underperform compared to ours.

TABLE III

NERF BLENDING RESULTS OF NERFUSER [4]

Scene	Methods	PSNR↑	SSIM↑	LPIPS↓
Sim	NeRFacto1	23.94	0.789	0.132
	NeRFacto2	23.77	0.792	0.134
	NeRFuser	19.58	0.634	0.254
Real	NeRFacto1	21.95	0.699	0.302
	NeRFacto2	20.50	0.605	0.328
	NeRFuser	18.26	0.401	0.399

TABLE IV

NERF BLENDING RESULTS OF OURS.

Scene	Methods	PSNR↑	SSIM↑	LPIPS↓
Sim	Ours 1	26.43	0.825	0.306
	Ours 2	24.19	0.780	0.339
	Our Blending	**25.58**	**0.812**	**0.317**
Real	Ours 1	27.61	0.855	0.176
	Ours 2	24.12	0.678	0.355
	Our Blending	**26.52**	**0.820**	**0.195**

Utilizing Model2Model pose optimization for blending, our approach excels in integrating the most desirable features while adeptly addressing inherent limitations. In contrast, using NeRFuser for blending visibly reduces scene representation quality, indicating limitations in leveraging distributed agents' full potential. Thus, our method significantly outperforms in large-scale scene blending.

The Fig.4 show the qualitative result of NeRF blending. NeRFuser, which employs NeRFacto for training a distributed NeRF, exhibits registration inaccuracies. Thus images synthesized fail to align seamlessly within a unified coordinate system, leading to suboptimal blending. In contrast, our framework benefits from the excellent registration result after tri-stage optimization, achieving a better blending result. It can be observed that the our blended image effectively incorporates the strengths and compensates for the weaknesses of the two distributed images, resulting in improved performance.

VI. CONCLUSIONS

In this paper, we propose a robust tri-stage pose optimization technique within a distributed NeRF system. This approach effectively addresses the challenges observed in previous NeRF registration stages. Through the strategic implementation of the bundle-adjusting Mip-NeRF 360, our system offers precise pose estimation for the images themselves. Further enhanced by the incorporation of the truncated dynamic low-pass filter, our iMNeRF achieves dependable and accurate Frame2Model registration. Building

on the initial relative transformation after Frame2Model optimization, Model2Model registration is then executed. As a result, our system not only corrects occlusion artifacts during the NeRF blending process but also demonstrates significant performance enhancements in both real-world and simulation environments. Looking ahead, our future efforts will focus on air-ground collaboration to achieve more generalizable distributed scenes reconstruction.

REFERENCES

[1] Michal Adamkiewicz, Timothy Chen, Adam Caccavale, Rachel Gardner, Preston Culbertson, Jeannette Bohg, and Mac Schwager. Vision-only robot navigation in a neural radiance world. *IEEE Robotics and Automation Letters*, 7(2):4606–4613, 2022.

[2] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5460–5469, 2022.

[3] Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Transactions on Robotics*, 37(6):1874–1890, 2021.

[4] Jiading Fang, Shengjie Lin, Igor Vasiljevic, Vitor Guizilini, Rares Ambrus, Adrien Gaidon, Gregory Shakhnarovich, and Matthew R. Walter. Nerfuser: Large-scale scene representation by nerf fusion, 2023.

[5] Lily Goli, Daniel Rebain, Animesh Garg Sara Sabour, and Andrea Tagliasacchi. nerf2nerf: Pairwise registration of neural radiance fields. *ICRA*, 2023.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[7] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019.

[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[9] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5721–5731, 2021.

[10] Philipp Lindenberger, Paul-Edouard Sarlin, Viktor Larsson, and Marc Pollefeys. Pixel-perfect structure-from-motion with featuremetric refinement. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5987–5997, 2021.

[11] Sainan Liu, Shan Lin, Jingpei Lu, Shreya Saha, Alexey Supikov, and Michael Yip. Baa-ngp: Bundle-adjusting accelerated neural graphics primitives. *arXiv preprint arXiv:2306.04166*, 2023.

[12] Yu A. Malkov and D. A. Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(4):824–836, 2020.

[13] Yury Malkov, Alexander Ponomarenko, Andrey Logvinov, and Vladimir Krylov. Approximate nearest neighbor algorithm based on navigable small world graphs. *Information Systems*, 45:61–68, 2014.

[14] Zhenxing Mi and Dan Xu. Switch-nerf: Learning scene decomposition with mixture of experts for large-scale neural radiance fields. In *International Conference on Learning Representations (ICLR)*, 2023.

[15] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, July 2022.

[16] Casey Peat, Oliver Batchelor, Richard Green, and James Atlas. Zero nerf: Registration with zero overlap, 2022.

[17] Antoni Rosinol, John J Leonard, and Luca Carlone. Nerf-slam: Real-time dense monocular slam with neural radiance fields. *arXiv preprint arXiv:2210.13641*, 2022.

[18] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016.

[19] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and Service Robotics*, 2017.

[20] Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P Srinivasan, Jonathan T Barron, and Henrik Kretzschmar. Block-nerf: Scalable large scene neural view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8248–8258, 2022.

[21] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, et al. Nerfstudio: A modular framework for neural radiance field development. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–12, 2023.

[22] Haithem Turki, Deva Ramanan, and Mahadev Satyanarayanan. Mega-nerf: Scalable construction of large-scale nerfs for virtual fly-throughs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12922–12931, 2022.

[23] Yuanbo Xiangli, Linning Xu, Xingang Pan, Nanxuan Zhao, Anyi Rao, Christian Theobalt, Bo Dai, and Dahua Lin. Bungeenerf: Progressive neural radiance field for extreme multi-scale scene rendering. In *The European Conference on Computer Vision (ECCV)*, 2022.

[24] Linning Xu, Yuanbo Xiangli, Sida Peng, Xingang Pan, Nanxuan Zhao, Christian Theobalt, Bo Dai, and Dahua Lin. Grid-guided neural radiance fields for large urban scenes, 2023.

[25] Xingrui Yang, Hai Li, Hongjia Zhai, Yuhang Ming, Yuqian Liu, and Guofeng Zhang. Vox-fusion: Dense tracking and mapping with voxel-based neural implicit representation. In *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 499–507, 2022.

[26] Yuqi Zhang, Guanying Chen, and Shuguang Cui. Efficient large-scale scene representation with a hybrid of high-resolution grid and plane features, 2023.

[27] Zhenxin Zhu, Yuantao Chen, Zirui Wu, Chao Hou, Yongliang Shi, Chuxuan Li, Pengfei Li, Hao Zhao, and Guyue Zhou. Latitude: Robotic global localization with truncated dynamic low-pass filter in city-scale nerf. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8326–8332. IEEE, 2023.

[28] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R Oswald, and Marc Pollefeys. Nice-slam: Neural implicit scalable encoding for slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12786–12796, 2022.