CBARF: Cascaded Bundle-Adjusting Neural Radiance Fields From Imperfect Camera Poses

Hongyu Fu, Xin Yu, Lincheng Li, and Li Zhang

Abstract—**Existing volumetric neural rendering techniques, such as Neural Radiance Fields (NeRF), face limitations in synthesizing high-quality novel views when the camera poses of input images are imperfect. To address this issue, we propose a novel 3D reconstruction framework that enables simultaneous optimization of camera poses, dubbed CBARF (Cascaded Bundle-Adjusting NeRF). In a nutshell, our framework optimizes camera poses in a coarse-to-fine manner and then reconstructs scenes based on the rectified poses. It is observed that the initialization of camera poses has a significant impact on the performance of bundle-adjustment (BA). Therefore, we cascade multiple BA modules at different scales to progressively improve the camera poses. Meanwhile, we develop a neighbor-replacement strategy to further optimize the results of BA in each stage. In this step, we introduce a novel criterion to effectively identify poorly estimated camera poses. Then we replace them with the poses of neighboring cameras, thus further eliminating the impact of inaccurate camera poses. Once camera poses have been optimized, we employ a density voxel grid to generate high-quality 3D reconstructed scenes and images in novel views. Experimental results demonstrate that our CBARF model achieves state-of-the-art performance in both pose optimization and novel view synthesis, especially in the existence of large camera pose noise.**

Index Terms—**3D Reconstruction, novel view synthesis, neural radiance fields, bundle-adjustment, camera pose registration.**

I. Introduction

THREE Dimensional Reconstruction [1], [2], [3], [4], [5], [6] and Novel View Synthesis [7], [8], [9], [10], [11] are essential tasks in computer vision [12]. They aim to reconstruct 3D scenes from the given 2D RGB images and render photo-realistic images in novel views. Inspired by the success of Neural Radiance Fields (NeRF) [11], volumetric neural rendering methods have gained significant popularity in the field of 3D reconstruction in recent years. However, one limitation of existing methods is the requirement for accurate camera poses corresponding to each input image. In other words, when the input camera poses contain noise or are even completely unknown, these methods might fail to reconstruct scenes or generate high-quality novel views.

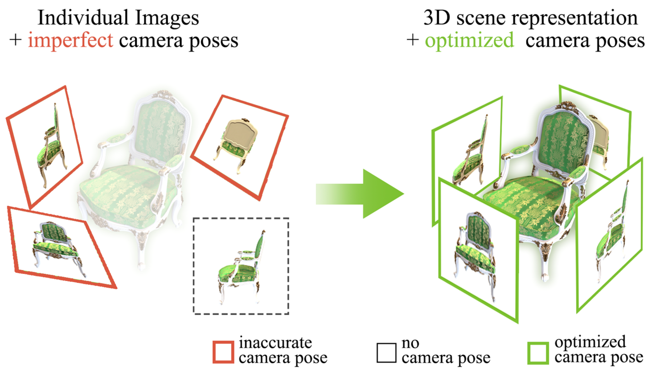


Fig. 1. Learning 3D scene representations relies on accurate camera poses of input images. However, coping with inaccurate or incomplete camera poses imposes a challenge. Our proposed CBARF tackles this problem by effectively reducing large camera pose noise and estimating missing camera poses.

The recent work BARF [13] attempts to solve camera registration and scene reconstruction jointly. BARF can be considered as a variant of photometric Bundle-Adjustment(BA) [14], [15], [16], [17], [18] with view synthesis serving as a proxy objective. BARF can effectively correct camera poses with moderate noise and reconstruct scenes when camera poses lie in restricted 3D space, *e.g.*, sharing similar orientations and lying on a common 2D plane. However, when registering camera poses in a 3D free space, BARF might fail due to the increased optimization difficulty of the joint estimate of camera poses and scene reconstruction. We observe that even when cameras are distributed in a 3D hemispherical space and face an object positioned at the center, BARF cannot handle camera pose noise and produces inferior reconstruction results. Moreover, in some cases, input images do not have the corresponding camera pose information. For instance, when COLMAP [6] might fail to estimate camera poses for some images, these images will not be used for scene reconstruction. This would lead to inferior reconstruction results, such as some parts of scenes are missing.

To address these issues, we propose Cascaded Bundle-Adjusting Neural Radiance Fields (CBARF), a novel approach to reconstructing scenes from inaccurate or partially unknown camera poses (Fig. 1). Our CBARF model adopts a

Manuscript received 6 September 2023; revised 18 January 2024; accepted 3 April 2024. Date of publication 15 April 2024; date of current version 26 August 2024. This work was supported in part by NSFC under Grant 62173005, in part by ARC-Discovery under Grant DP220100800 to XY, and in part by ARC-DECRA under Grant DE230100477 to XY. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Tat-Jen Cham. *(Corresponding author: Li Zhang.)*

Hongyu Fu is with the Department of Electronic Engineering, Tsinghua University, Beijing 100084, China, and also with NetEase Fuxi AI Lab, Hangzhou 310052, China (e-mail: fhy21@mails.tsinghua.edu.cn).

Xin Yu is with the School of Electronic Engineering and Computer Science, University of Queensland, Brisbane 4072, Australia (e-mail: xin.yu@uq.edu.au).

Lincheng Li is with NetEase Fuxi AI Lab, Hangzhou 310052, China (e-mail: lilincheng@corp.netease.com).

Li Zhang is with the Department of Electronic Engineering, Tsinghua University, Beijing 100084, China (e-mail: chinazhangli@mail.tsinghua.edu.cn).

Digital Object Identifier 10.1109/TMM.2024.3388929

coarse-to-fine manner. In each scale, CABRF first updates camera poses by a BA module. Subsequently, we design a novel criterion to identify poorly estimated poses that still have not been rectified after BA optimization. We then introduce a neighbor-replacement strategy to update these inaccurate poses.

Specifically, we find that optimizing camera poses with excessive iterations is computationally costly and does not lead to better performance. Thus, we introduce a compact BA module by modifying the basic BA module to accelerate the pose optimization process. Due to the substantial influence of the initialization state of camera poses on BA, we adopt the preceding pose estimation results as the initialization for the subsequent BA module. As a result, we cascade multiple compact BA modules in series, forming the backbone of our CBARF model. The number of cascades in our model is adaptive to avoid insufficient or excessive optimization rounds.

To further enhance the performance of the cascaded BA, we introduce a neighbor-replacement strategy between each pair of BA modules. This strategy involves replacing inaccurate camera poses with poses from neighboring viewpoints. Due to the absence of ground-truth camera poses, we design a novel criterion to identify potentially inaccurate camera poses based on the quality of rendered images in the corresponding views. In addition, we incorporate non-maxima suppression [19], [20], [21] to enhance the identification of inaccurate poses. The final refined poses are provided into a density voxel grid [22], facilitating the generation of high-quality rendered images for the purpose of result comparison. We conducted a comprehensive evaluation and comparison of our approach on the NeRF-synthetic [11] and BlendedMVS [23] datasets. Our results demonstrate that our approach achieves a new state-of-the-art performance in optimizing camera poses from noisy or insufficient initial estimates.

Overall, the contributions of our work are summarized as follows:

- We propose a robust coarse-to-fine 3D reconstruction framework that effectively optimizes camera poses in the presence of significant noise. Our model exhibits the capability of handling images with noisy camera pose information.
- We demonstrate that the initialization of camera poses is crucial for bundle-adjustment (BA) performance, and we propose the cascaded BA to progressively refine the inaccurate camera poses.
- We propose a neighbor-replacement strategy to improve the optimization process by identifying and replacing inaccurate camera poses with the poses of their neighboring cameras.

II. Related Work

Structure from Motion: Structure from motion (SfM) system [6], [24], [25], [26], [27], [28], [29], such as the COLMAP [6] aims to estimate the camera poses and recover the 3D structure from the given a set of input images. Since SfM only needs RGB images without any pose or depth information as input, it has been widely used to reconstruct sparse point clouds and recover camera poses. Many works on SfM achieve great success such as COLMAP [6] and OpenMVS [24]. However, most SfM systems rely on detecting and matching distinctive key-points. Thus, they may fail to reconstruct scenes, especially in regions with less texture or repetitive patterns.

Neural Radiance Field: With the continuous advancement of deep learning techniques, neural networks have found wide applications in 3D reconstruction and novel view synthesis [30], [31], [32], [33], [34], [35], [36], [37], [38], [39], [40], [41]. NeRF (Neural Radiance Field [11]) is one such technique. NeRF aims to learn scene representation inside an MLP and synthesize novel views directly via differentiable volume rendering. Due to its photo-realistic rendering capabilities, NeRF has gained a lot of attention in various fields, such as high-quality head reconstruction [42], [43]. Many researchers have explored ways to improve its performance and address its weaknesses [13], [22], [44], [45], [46], [47], [48], [49], [50], [51]. Some works aim to predict a continuous neural scene representation from a sparse set of input views. PixelNeRF [44] employs a fully convolutional approach for processing image inputs, enabling the network to be trained across multiple scenes and learn a scene prior. This facilitates generating novel view synthesis conditioned on a limited number of input images. Many researchers have attempted to accelerate the training process of NeRF. Plenoxels [45] achieves comparable performance to NeRF but with a significant speed improvement, being approximately 100 times faster. It utilizes a sparse voxel grid representation, where each voxel is associated with density and spherical harmonic coefficients. DVGO [22] further advances NeRF and 3D scene representation. By using a voxel grid representation rather than an MLP, DVGO can significantly accelerate the rendering process compared to traditional methods. However, those existing volumetric neural rendering methods require a set of images with accurate poses as input. They may fail to synthesize high-quality novel views when camera poses contain some noise.

Extended NeRF with Inaccurate Poses: Several works aim to reduce the reliance on highly accurate camera poses. BARF [13] is a type of photometric bundle adjustment (BA) based on view synthesis. BA is a fundamental technique in computer vision and photogrammetry used to refine the parameters of a 3D reconstruction model and the camera poses simultaneously. It aims to minimize the reprojection error between the observed 2D points in multiple images and their corresponding 3D points in the scene. BARF adapts the principles of BA to refine the neural scene representations and register camera frames in the context of training NeRF from imperfect or unknown camera poses. Unlike traditional bundle adjustment methods, BARF can learn scene representations from randomly initialized network weights, which diminishes the reliance on local registration sub-procedures. By incorporating BA techniques into the optimization process, BARF aims to address the limitations of NeRF and enable the learning of accurate 3D scene representations.

L2G-NeRF [52] employs a local-to-global strategy to address neural field reconstruction and camera pose registration jointly. It demonstrates the sensitivity of bundle-adjustment against initialization in neural fields and thus adopts an effective local-to-global registration strategy. GARF [49] introduces a new positional embedding-free neural radiance field architecture with Gaussian activation to solve the joint problem of reconstruction and pose estimation. SPARF [50] exploits multi-view geometry constraints to learn the scene representation and refine the camera poses jointly from sparse viewpoints. NeRF– [51] optimizes camera poses as learnable parameters with NeRF training through a photometric reconstruction on forward-facing datasets. CamP [53] designs a preconditioning matrix to normalize the effects of each camera parameter on the projection of points in a scene and decorrelate the effects of each camera parameter from others. By doing so, CamP can recover camera parameters and challenging scenes. DBARF [54] jointly optimizes GeNeRFs (Generalizable Neural Radiance Fields [55]) and relative camera poses, and demonstrates generalizability across scenes without requiring per-scene fine-tuning. FlowCam [56] employs differentiable rendering to lift frame-to-frame optical flow to 3D scene flow, enabling online joint optimization of camera poses and 3D neural scene representations.

However, these methods still fail to learn scene representations when camera poses contain severe noise in a 3D-free space. Therefore, our proposed CBARF is designed to efficiently reconstruct 3D scenes with inaccurate or insufficient camera poses.

III. Proposed Method

This work addresses the challenge of synthesizing novel views under conditions where certain input camera poses contain significant noise or are even unknown. Thus, we propose CBARF, a novel framework incorporating several simple yet effective strategies, to optimize camera poses and learn 3D scene representations. In this section, we first introduce the cascaded BA in Section III-A. Similar to other gradient descent algorithms, cascaded BA is prone to over-fitting when the initial camera poses have significant errors. We then present the neighbor-replacement strategy in two separate parts, detailing the process of identification and replacement of erroneous camera poses. In Section III-B, we design a novel criterion to detect erroneous camera poses arising from over-fitting during the BA process. Since ground-truth poses are unknown, this method identifies the potentially inaccurate poses based on the quality of their corresponding rendered images. We introduce the procedure for replacing these identified erroneous poses in Section III-C. This technique rectifies inaccurate camera poses by replacing them with poses of neighboring cameras.

A. Cascaded BA

Employing bundle-adjustment methods [13], [14], [15], [16], [17], [18] under conditions where the initial camera poses deviate significantly from ground-truth may result in broken 3D structure and failure pose estimation. The sub-optimal performance of some BA models such as BARF [13] may be attributed to the over-fitting of the neural radiance field network. Specifically, BARF generates novel view images by an MLP and uses the synthesized images as the proxy objective to update camera poses in an alternating manner. However, when some input poses contain large deviations, it may learn an incorrect scene representation, leading to an erroneous optimization and preventing the model from self-correcting. In essence, the BA model heavily relies on accurate initial poses to establish a reliable starting point for reconstruction and optimization.

In some other optimization tasks [57], [58], [59], [60], [61], [62], multi-stage structures are employed for iterative refinement and improved performance. Inspired by this concept, we propose cascaded BA, a multi-stage framework incorporating several compact BA modules connected in series. In cascaded BA, the camera poses estimated from the previous stage are utilized as the initialization for the subsequent stage. In this way, the model coarsely eliminates cumulative errors and results in more accurate pose estimation for the subsequent process. The compact BA module is based on BARF [13]. However, BARF takes much training time to complete the reconstruction. Since high-quality reconstruction is not essential for coarsely optimizing camera poses, we reduce the training iterations of the compact BA module. We also adjust the learning rate to match the different training stages. Consequently, We adopt a coarse-to-fine manner consisting of coarse, recursive, and fine stages (Fig. 2). The number of compact BA modules is adaptive and depends on the characteristics of the datasets. Specifically, we employ a loop detection technique in the recursive stage to assess the current optimization effectiveness and determine the number of compact BA modules. This helps us avoid insufficient or excessive optimization rounds, ensuring an optimal balance for the performance of the model.

Our experiments show that multi-stage BA rectifies inaccurate poses more efficiently than single-stage BA during the same training time. As shown in Fig. 3, the cascaded BA without neighbor-replacement (indicated by the red curve) reduces the amount of camera pose noise than single-stage BA (indicated by the blue curve). The single-stage BA quickly falls into a sub-optimal solution, while the cascaded BA exhibits an improved optimization result. Moreover, the green curve, cascaded BA with neighbor-replacement (Section III-C), demonstrates a more significant reduction in noise.

B. Erroneous Pose Detection

This module aims to overcome the challenge of identifying inaccurate camera poses without ground-truth. It involves generating view synthesis using the current pose estimates and accessing the quality of the rendered images. Inferior rendering quality consistently indicates inaccuracies in the corresponding camera poses. However, we observe that rendering errors caused by inaccurate camera poses are prone to be confused with noise introduced by the model, especially when using conventional image evaluation algorithms [63], [64], [65] for evaluation. Additionally, even minor inaccuracies in camera poses can result in pixel-level displacements in the rendered images. As a result, conventional image evaluation methods predominantly based on pixel comparisons might consequently lead to erroneous assessments. Some other methods aim to identify erroneous camera poses. The rotation averaging [66] estimates the absolute orientations of cameras or views in a way that best agrees with a set of pairwise relative orientations. However, in our task, rotation averaging is not a very applicable method because obtaining accurate relative orientations is challenging. Consequently, we introduce a novel criterion primarily based

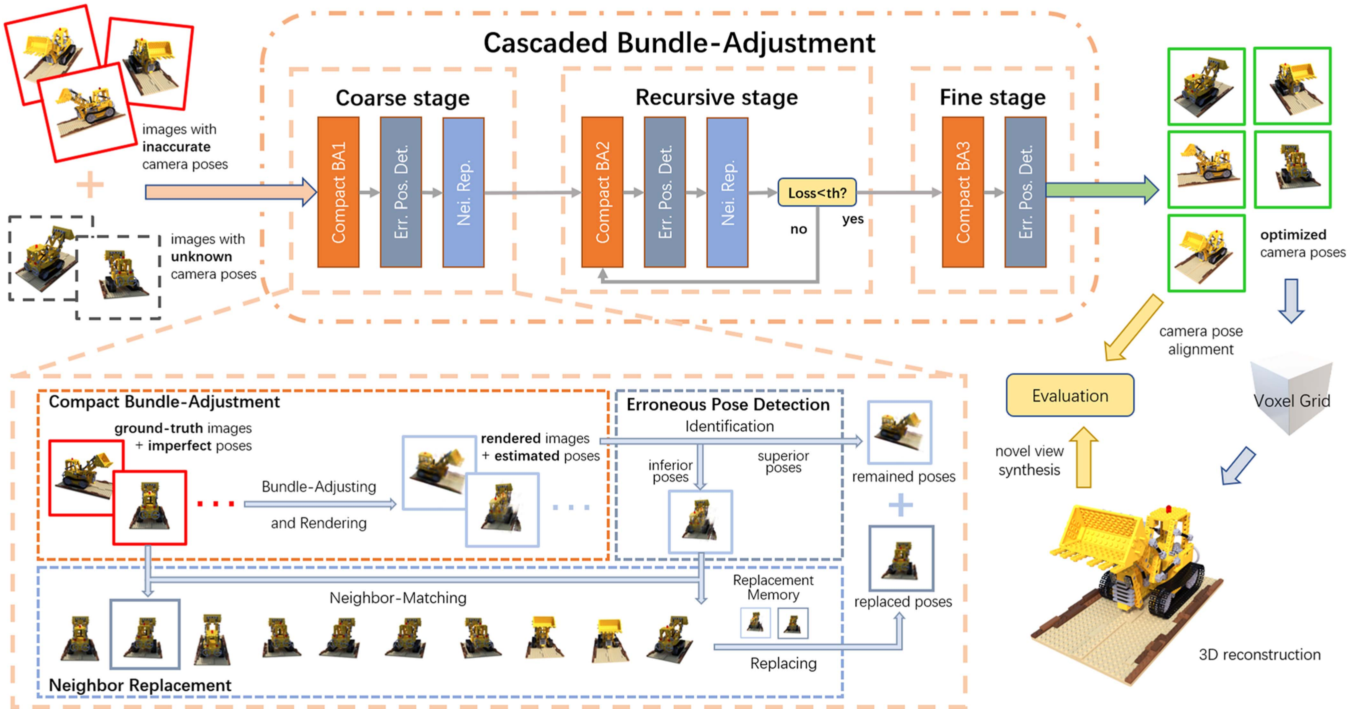


Fig. 2. Overview of our proposed CBARF. At each stage of the cascaded BA, we use a compact BA module to rectify camera poses and generate rendered images. Subsequently, we identify the inferior rendered images to select potentially inaccurate camera poses and update these erroneous poses by their nearest poses found across all views. The updated poses are then fed into the next stage for further optimization. During the recursive stage, we employ a loop detection technique to automatically determine the number of compact BA modules. In the final stage, the optimized poses with reduced noise are checked again to remove any incorrect poses. The remaining poses are then input into a density voxel grid for high-quality reconstruction.

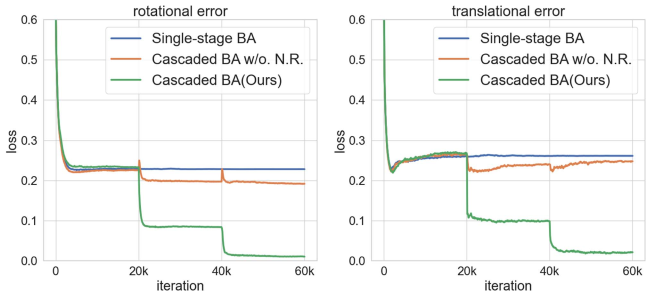


Fig. 3. Comparison between the cascaded BA and the single-stage BA. In this figure, the cascaded BA consists of three compact BA modules, with each module set to 20 k iterations. The single-stage BA is configured with 60,000 iterations, matching the total number of iterations used in the cascaded BA. The green curve represents the cascaded BA with neighbor-replacement at each cascaded node, while the red curve represents the cascaded BA without neighbor-replacement. The blue curve represents the single-stage BA. The single-stage BA quickly falls into a sub-optimal solution, while the cascaded BA exhibits fluctuations at the cascaded nodes, resulting in a better final result.

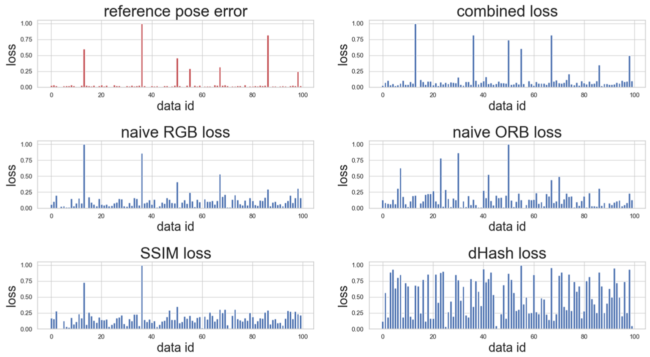


Fig. 4. Comparison between several evaluation methods. The x-axis of each chart represents the identification numbers of different views, while the y-axis represents normalized error values. The first chart (in red) illustrates the distribution of camera pose errors at different views, while the remaining charts display the distribution of rendered image errors calculated by various image evaluation methods. Compared to other traditional loss, our combined loss exhibits a stronger correspondence between rendered image errors and camera pose errors. This demonstrates the superior capability of our method to identify inaccurate camera poses.

on ORB key-point [67] to overcome the disadvantages of conventional evaluation methods. Additionally, we design several supplementary strategies to improve the accuracy and reliability of the criterion. As illustrated in Fig. 4, our combined criterion exhibits superior correlation with camera pose errors when compared to other conventional image evaluation methods.

ORB (Oriented FAST and Rotated BRIEF) [67] is a feature detection and description algorithm used in computer vision and image processing. It is similar to the SIFT (Scale-Invariant Feature Transform) [68] algorithm, but is designed to be faster and more efficient. In our method, the rendered images used for matching exhibit low quality, which could be a potential reason for the failure of other matching algorithms such as SIFT or LOFTR [69]. However, we find that ORB is suitable for finding and matching key-points between rendered images and reference images in our task. Additionally, we propose several supplementary techniques to enhance the accuracy of key-point matching.

- *K-Nearest Neighbors:* In order to effectively identify the matched key-points between reference images and rendered images, we employ the K-Nearest Neighbors (KNN) algorithm [70]. For each key-point in the rendered image, we compute its feature distance with all the key-points in the reference image. Subsequently, we identify the point with the shortest distance as a potential match and compare its feature distance with others. A dependable match generally shows a significantly shorter feature distance compared to non-matching points. Consequently, potential matches without a notably shorter feature distance are disregarded.
- *Bidirectional Check:* We use a bidirectional check [71], [72] between rendered images and reference images to further improve the accuracy of the key-point matching. In detail, for each key-point p_i in the rendered image, we conduct a search to find the best matching point p_m in the reference image. Additionally, we traverse the rendered image to confirm whether p_i is also the best matching point for p_m. We only retain the reliable matches, where two points serve as each other's best matching points.
- *Coordinate Constraints:* In some cases, inaccurate camera poses result in scenes being rendered from an incorrect viewpoint, while still generating visually high-quality images. In the overlapping regions of the scene captured from different viewpoints, there are numerous shared key-points at different pixel positions. Since the matching method relies on feature distances rather than pixel positions, these displaced key-points may be incorrectly matched. This results in overlooking some inferior rendered images caused by perspective errors. To address this issue, we incorporate coordinate constraints for key-point pairs. Specifically, when a pair of matched key-points exhibits a significant coordinate separation, we classify them as unreliable matches and discard them to exclude perspective errors.

To address the issue of insufficient ORB key-points, we supplement the criterion with RGB-MSE. MSE (Mean Squared Error) is a common loss function used in regression problems [73]. It measures the average squared difference between the predicted output and ground-truth. However, directly using RGB-MSE in our task yields unsatisfactory results. We observe that variations in the foreground-to-background ratio can impair the indicative values of MSE across different viewpoints. Therefore, we introduce a compensation factor to address this issue. The revised MSE can be described as

$$MSE_c = \sqrt{\frac{N}{N_f}} \cdot \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2, \tag{1}$$

where N_f represents the number of pixels in the foreground of the reference image, and N represents the total number of pixels. y_i represents the pixel value of ground-truth image while $\hat{y}_i$ represents the rendered image. We use the foreground masks provided by the dataset to differentiate between the foreground and background.

In the final step, we utilize the combined criterion to assign scores for all the rendered images obtained from the current pose estimates. By analyzing these scores, we identify the low-quality images and their corresponding camera poses. The poses are considered to contain a high level of noise and will be further optimized in neighbor-replacement III-C.

C. Neighbor-Replacement

The neighbor-replacement technique involves replacing the camera poses identified as low-quality in Section III-B with their respective neighbors. Specifically, we first denote $\mathbb{Q} = \{q_k | k = 1, 2, \ldots, N\}$ as all views from the input set. Note that we have labeled each q_k as either 'superior' or 'inferior' in Section III-B. We denote them as $q_s \in \mathbb{Q}_s$ and $q_i \in \mathbb{Q}_i$, respectively.

However, we observe that inaccurate camera poses have a detrimental effect on the rendering not only in their corresponding viewpoints but also in neighboring viewpoints. This interference can arise from the model learning incorrect scene representations in the relevant regions. Consequently, our proposed criterion in Section III-B may erroneously identify some accurate camera poses as low-quality due to the broken scene representations caused by neighboring inaccurate poses. To address this issue, we introduce non-maxima suppression [19], [20], [21] after identification. Specifically, when multiple neighboring camera poses are assigned low scores and the ratio between the lowest score and other scores is below a specified threshold (set at 0.7 in our case), it is probable that only the camera pose with the lowest score is inaccurate. Other camera poses are mistakenly labeled as low-scoring poses. However, even in this scenario, the non-lowest-scoring camera poses can still be incorrect and mistakenly excluded. Nevertheless, our CBARF is designed as a multi-stage structure. In our experiments, when an erroneously optimized pose is excluded in a stage, it is likely to be filtered out in the subsequent stages. Thus, we update the classification of 'superior' and 'inferior' in set $\mathbb{Q}$ by discarding misidentified camera poses. We denote the updated classifications as $\widetilde{\mathbb{Q}}_s$ and $\widetilde{\mathbb{Q}}_i$, respectively.

For each inferior view $\widetilde{q}_i$ in set $\widetilde{\mathbb{Q}}_i$, we search for the nearest neighbor view $\widetilde{q}_{simi}$ in the set $\mathbb{Q}$. We adopt the combined criterion introduced in Section III-B to measure and rank the similarity between $\widetilde{q}_i$ and each q_i. We additionally perform matching on the rotated images with their corresponding camera poses equivalently rotating [74]. This approach expands the search space and improves the accuracy of neighbor-matching results. The camera poses in set $\widetilde{\mathbb{Q}}_i$ are then replaced with their nearest neighbor in set $\widetilde{\mathbb{Q}}_s$. As a result, we obtain an updated set of camera pose estimates denoted as $\widetilde{\mathbb{Q}} = \{\widetilde{q}_k | k = 1, 2, \ldots, N\}$. The $\widetilde{\mathbb{Q}}$ is more reliable for further optimization.

However, the effectiveness of replacement depends on the accuracy of neighbor-matching. A mismatched neighboring camera pose may be replaced rapidly, leading to eventual optimization failure. Even with the enhancement measures in Section III-B, some matching results may still be incorrect. Hence,

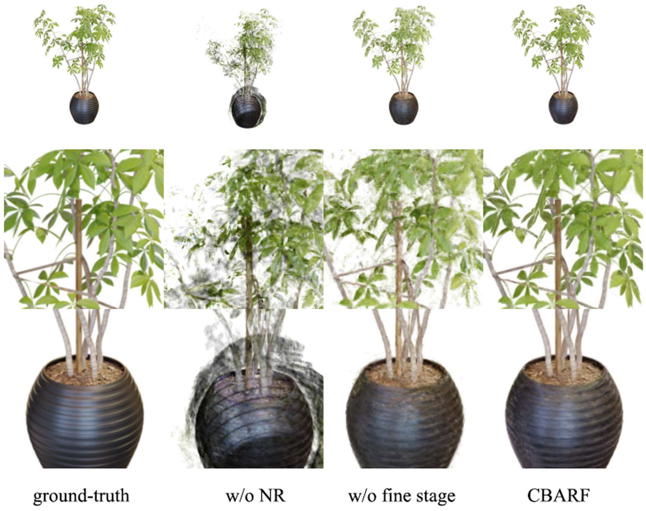


Fig. 5. Qualitative results of CBARF with different module compositions. We test CBARF without the neighbor-replacement or the fine stage, and visualize the image synthesis at estimated poses. CBARF achieves comparable synthetic quality to the ground-truth, indicating successful pose optimization. On the other hand, the absence of the neighbor-replacement or the fine stage resulted in suboptimal registration, leading to synthesis artifacts in rendered images.

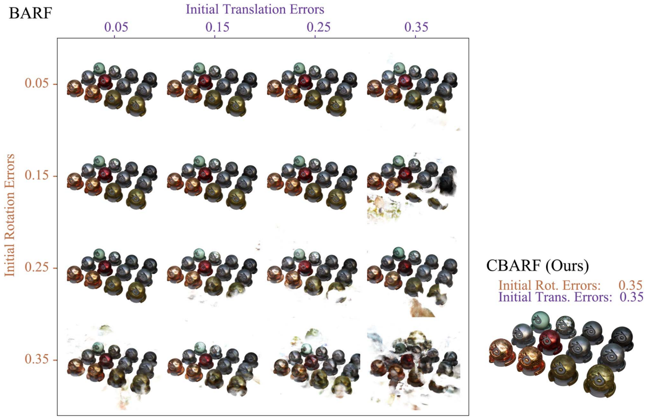

BARF
Initial Translation Errors
0.05
0.15
0.25
0.35
Initial Rotation Errors
0.05
0.15
0.25
0.35
CBARF (Ours)
Initial Rot. Errors: 0.35
Initial Trans. Errors: 0.35

Fig. 6. Visualization results of 3D reconstruction with BARF under different levels of initial camera pose rotation errors and translation errors. Significant degradation is observed when the rotation error reaches 0.25 or the translation error reaches 0.35.

we introduce a replacement-memory technique to prevent redundant erroneous replacements. For each erroneous viewpoint, we keep track of the pose used for replacement and skip poses that have already been utilized. This prevents the optimization process from getting stuck due to inaccurately matched neighboring camera poses.

By replacing the inferior camera poses with more accurate ones, the neighbor-replacement technique improves the initialization effect of each phase in the cascaded BA and significantly enhances the overall performance of the model. As shown in Fig. 3, the cascaded BA with neighbor-replacement (indicated by the green curve) effectively reduces the camera pose noise after each cascaded node, ultimately reaching an extremely low level. Improved camera pose estimation results in higher rendering quality. Fig. 5 illustrates the rendering performance of CBARF with different module compositions. Benefiting from the Neighbor-Replacing module and the coarse-to-fine manner, CBARF achieves comparable rendering performance to the ground-truth.

IV. Experiments

In this section, we first validate the effectiveness of our proposed CBARF in pose registration and view synthesis when dealing with noisy camera poses. Subsequently, we evaluate CBARF's capability of learning scene representations from incomplete camera pose data.

A. Optimizing Noisy Camera Poses

Experimental settings: We conduct evaluations on the NeRF-synthetic dataset [11] with image resolutions of 800×800. The camera poses are described by the Lie group $SE(3)$ [75]. The Lie group can be defined as

$$SE(3) = \left\{ \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \in \mathbb{R}^{4\times4} | \mathbf{R} \in SO(3), \mathbf{t} \in \mathbb{R}^3 \right\}, \quad (2)$$

where the $\mathbf{R}$ represents the rotation matrix and $\mathbf{t}$ represents the translation matrix. $SO(3)$ [76] refers to the special orthogonal group in three dimensions and often be used to describe the rotations. Note that we drop the last row in the Lie group, so the camera poses $\mathbf{P}$ is

$$\mathbf{P} = \left\{ \begin{bmatrix} \mathbf{R}_p & \mathbf{t}_p \end{bmatrix} \in \mathbb{R}^{3\times4} | \mathbf{R}_p \in SO(3), \mathbf{t}_p \in \mathbb{R}^3 \right\}. \quad (3)$$

To simulate inaccurate camera poses, we introduce noise $\mathfrak{n} \in \mathfrak{se}(3)$ by generating 6-dimensional random normal distribution noise based on the Lie algebra [75]. We set the noise coefficient to 0.35 for our method, while it is set to 0.15 in BARF [13]. Increasing the noise coefficient presents more challenges for camera pose optimization. As shown in Fig. 6), the performance of BARF significantly degrades when the rotation error reaches 0.25 or the translation error reaches 0.35. When both rotation and translation errors reach 0.35 (as we set in Section IV-A), BARF struggles to synthesize recognizable images from new viewpoints. In contrast, our method CBARF is robust against the camera pose noise. The noise $\mathfrak{n}$ in our method can be described as

$$\mathfrak{n} = 0.35\mathcal{N}(\mathbf{0}, \mathbf{I}) \in \mathbb{R}^6. \quad (4)$$

It corresponds to an average deviation of 30.4° in rotation and 0.56 in translation. We then transform the noise $\mathfrak{n}$ into the camera transform matrix $\mathbf{N} = \begin{bmatrix} \mathbf{R}_n & \mathbf{t}_n \end{bmatrix}$. We compose it with the reference camera pose $\mathbf{P}$ to get the imperfect camera poses $\widetilde{\mathbf{P}}$ as

$$\widetilde{\mathbf{P}} = \begin{bmatrix} \mathbf{R}_p\mathbf{R}_n & \mathbf{t}_p + \mathbf{t}_n \end{bmatrix}. \quad (5)$$

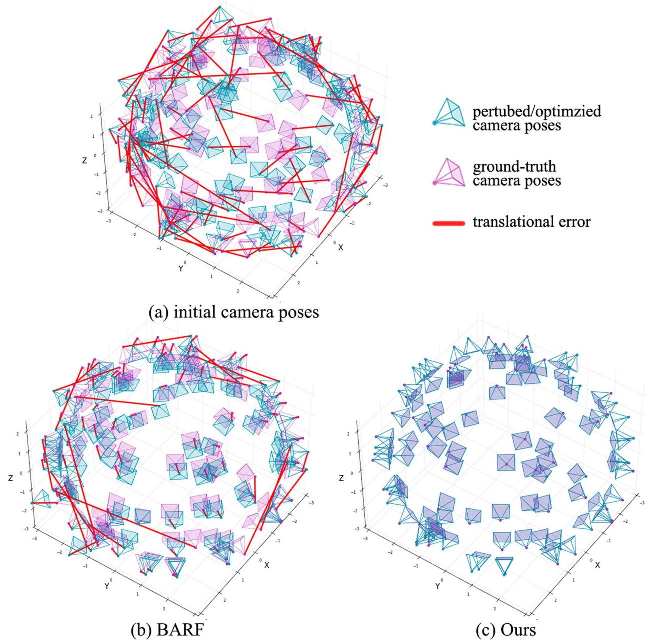
In particular, we optimize the camera poses by training the camera refine parameters $\mathfrak{p} \in \mathfrak{se}(3)$. We then convert it into camera transform matrix $\mathbf{P}_r = \begin{bmatrix} \mathbf{R}_r & \mathbf{t}_r \end{bmatrix}$ to compose with the $\widetilde{\mathbf{P}}$. In this way, we obtain the refined camera poses as

$$\mathbf{Q} = \begin{bmatrix} \mathbf{R}_p \mathbf{R}_n \mathbf{R}_r & \mathbf{t}_p + \mathbf{t}_n + \mathbf{t}_r \end{bmatrix}. \tag{6}$$

Implementation Details: In this paper, we propose a coarse-to-fine structure (Fig. 2) to optimize camera poses and reconstruct 3D scenes. For each compact BA module, we choose the BARF network [13] as our backbone. We follow the architectural settings from the original BARF with some modifications and adopt a coarse-to-fine strategy for each optimization stage. Specifically, in the coarse stage and the recursive stage, the iteration for the compact BA module is set to 20 k, while in the fine stage, it is 100 k. The total iteration is about 160 k, which takes about 7 hours for training in synthetic object. To further improve training efficiency, we reduce the image sizes by half during the pose optimization stage, resulting in resolutions of 400×400. Similar to BARF [13] and NeRF [11], we employ exponential interpolation [77], [78] to calculate a gradually decreasing learning rate. Additionally, we introduce a modulation factor to determine the degree of deviation from the initial value. A larger modulation factor biases the overall learning rate more towards the initial high value. Setting the modulation factor to 1.0 corresponds to using the original exponential learning rate. In the three stages of the cascaded BA, the modulation factors were set to 10.0, 3.0, and 1.0, respectively. In the Voxel Grid Module, DVGO [22] is employed to generate synthetic images with optimized camera poses for evaluation.

Evaluation Metrics: We evaluate the performance of our model in two main aspects: camera pose error for pose optimization and view synthesis quality for scene representation. For camera pose evaluation, we measure the rotation error and translation error separately to assess the accuracy of the optimized camera poses. In terms of view synthesis evaluation, we employ several metrics including PSNR, SSIM and LPIPS [65] to provide quantitative measures of the similarity between the synthesized images and the reference images.

The reference camera poses are only used during the evaluation process to calculate the errors and are not used as supervision during the optimization process. As a result, the optimized camera poses may have a global offset from the ground-truth values. The global offset can be thought of as the combination of overall rotation and translation. It does not affect the learning of scene representation, but it can introduce a misalignment between the coordinate system of the optimized camera poses and the ground-truth. This misalignment may lead to incorrect evaluations of pose optimization and scene reconstruction quality. Thus, we use Procrustes analysis [79] to align the optimized poses with the ground-truth before evaluation. Procrustes analysis is a common technique used to align two sets of data points by minimizing the difference between them. By calculating the global offset through Procrustes analysis, we can align the optimized camera poses to the ground-truth and accurately calculate the rotation and translation errors of the optimized camera poses. Additionally, we can also align the camera poses in the test set for view synthesis evaluation.



(a) initial camera poses

(b) BARF

(c) Ours

Fig. 7. Visualization of the pose optimization result in NeRF- synthetic dataset. Each figure shows the translation errors between ground-truth camera poses and the perturbed or optimized camera poses for the materials scene. The initial noise coefficient is set to 0.35. BARF encounters overfitting before completing the optimization, while CBARF successfully optimizes all camera poses. The camera poses are aligned by Procrustes Analysis.

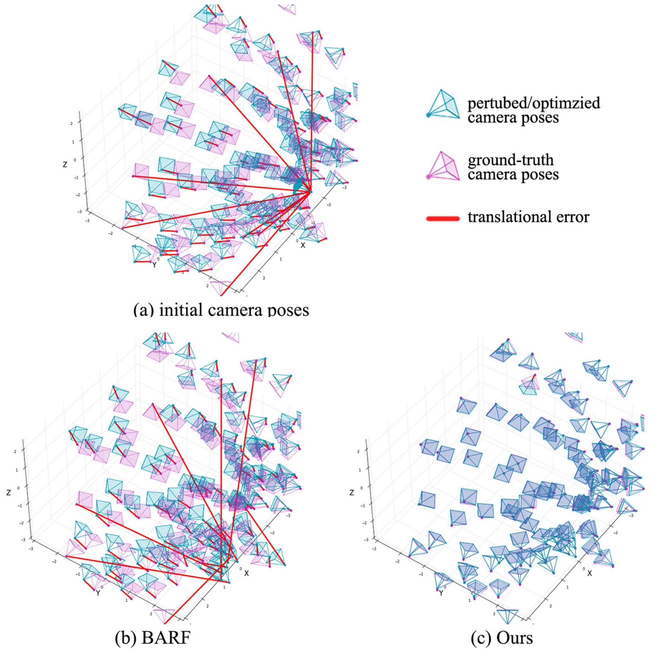
Results: We incorporate the camera poses with added noise as inputs to models and then perform optimization and reconstruction. We calculate both pose error and rendering quality. The visualization of the pose optimization result is shown in Fig. 7. The results of rendering are visualized in Fig. 10 and the quantitative metrics are reported in Table. I. The initial camera poses exhibit an average deviation of 30.4° in rotation and 56.2 (scaled by 100) in translation. As depicted in Table. I, CBARF consistently outperforms the baseline method BARF in terms of camera pose optimization. Additionally, our method demonstrates higher accuracy in estimating camera poses compared to the reference model COLMAP [6]. The rendering results of our method, as illustrated in Fig. 10, exhibit comparable quality to the ground-truth images.

While our CBARF employs a cascaded structure, the model size is comparable with the state-of-the-art (about 7 MB for each stage). The total number of training iterations is around 260 k, which is slightly more than 200 k used in BARF. Our experiments show that CBARF takes about 10 hours for training on synthetic objects, while training BARF takes about 8 hours. Moreover, our CBARF provides two fast modes (fast and fast+), effectively enhancing training speed through a reduction of training iterations. As shown in Fig. 9, in the fast mode, the training time for each scene is reduced to 7.5 hours (less than 8 hours for BARF), while our image quality still remains high-quality and is significantly better than BARF. In the fast+

TABLE I
QUANTITATIVE RESULTS ON SYNTHETIC OBJECT SCENES

Scene	Camera pose optimization				View synthesis quality								
	Rotation (°)↓		Translation↓		PSNR↑			SSIM↑			LPIPS↓		
	BARF	CBARF	BARF	CBARF	BARF	CBARF	ref.	BARF	CBARF	ref.	BARF	CBARF	ref.
Chair	5.208	**0.099**	15.24	**0.479**	17.08	**27.94**	34.09	0.801	**0.927**	0.976	0.181	**0.038**	0.027
Drums	5.748	**0.042**	19.21	**0.148**	12.69	**25.29**	25.42	0.714	**0.928**	0.929	0.287	**0.080**	0.079
Ficus	5.316	**0.083**	12.49	**0.444**	17.05	**30.54**	32.58	0.821	**0.969**	0.977	0.142	**0.028**	0.025
Hotdog	4.931	**0.248**	14.30	**1.305**	15.97	**23.44**	36.76	0.827	**0.891**	0.980	0.223	**0.078**	0.033
Lego	7.053	**0.073**	21.86	**0.261**	12.13	**31.35**	34.71	0.680	**0.962**	0.976	0.317	**0.033**	0.027
Materials	11.85	**0.047**	28.68	**0.179**	11.09	**28.90**	29.58	0.669	**0.947**	0.950	0.311	**0.061**	0.059
Mic	6.568	**0.063**	17.26	**0.252**	13.45	**30.56**	33.11	0.827	**0.976**	0.982	0.172	**0.020**	0.018
Ship	10.61	**1.099**	25.22	**0.899**	10.43	**28.01**	29.04	0.622	**0.870**	0.877	0.406	**0.163**	0.161
Mean	7.161	**0.219**	19.28	**0.496**	13.74	**28.25**	31.91	0.745	**0.934**	0.956	0.255	**0.062**	0.054

CBARF achieves superior performance from noisy camera poses compared to the baseline methods. Moreover, CBARF maintains comparable view synthesis quality to the reference images rendered at the ground-truth camera poses. Translation errors in this table are scaled by 100.
The bold values represent the best results obtained among all methods (excluding reference values).



(a) initial camera poses

(b) BARF

(c) Ours

Fig. 8. Visualization of the pose optimization result in BlendedMVS dataset. Each figure shows the translation errors between ground-truth camera poses and the perturbed or optimized camera poses for the bear scene. Before training, 10% of the camera pose data is discarded and reinitialized. BARF is ineffective in optimizing these missing camera poses, while CBARF can successfully address this challenge.

mode, our training time is even half of the time of the normal mode, approximately 5 hours. Although PSNR scores decrease slightly under two fast modes, CBARF is capable of generating high-quality reconstruction results that are visually similar to those of the version trained with 260 k iterations.

B. Optimizing Incomplete Camera Poses

Experimental settings: In this work, we conduct evaluations on the BlendedMVS dataset [23]. The camera poses are estimated by COLMAP [6]. To simulate the scenario where the camera pose estimation fails or is unavailable for certain images, we randomly drop 10% of the camera poses and use the remaining images as the test set F. The other images with camera poses are grouped as T. We then assign an initial camera pose $\mathbf{P}_{ini}$ to the images in F. During the training phase, we use both BARF and CBARF to learn scene representation from T and jointly optimize the camera poses of both groups, resulting in T' and F'. Since there are no ground-truth camera poses for evaluation, we use optimized poses in F' to assess rendering quality to estimate the camera pose error. In the testing phase, we compare the rendering results generated using the camera poses estimated by different models. This allows for a meaningful comparison of the rendering quality between different approaches.

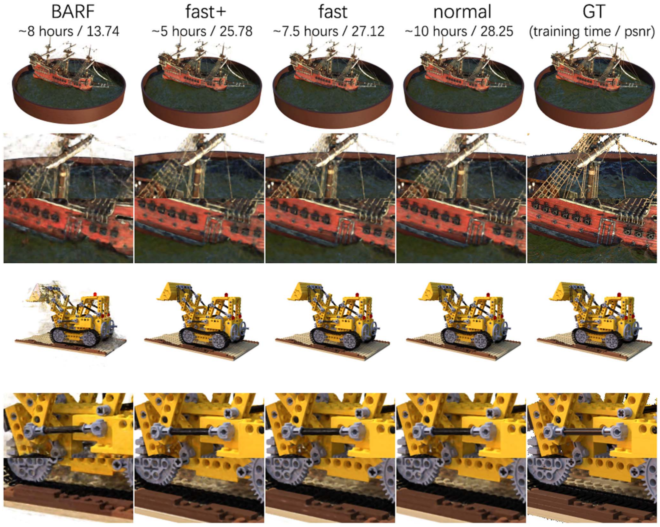


Fig. 9. Visualization results of CBARF under three modes (fast+, fast, normal). In fast mode, the training time for each scene is around 7.5 hours, which is less than 8 hours for BARF, and the image quality is significantly better than that of BARF. In fast+ mode, the training time is even half, approximately 5 hours, compared to that of the normal mode. In these three modes, the visual quality of their reconstructed results is similar.

Results: Due to the lack of ground-truth camera poses, it is not possible to calculate the error of camera poses. We can only calculate the rendered image quality to evaluate the performance of different models. The results on the BlendedMVS dataset are

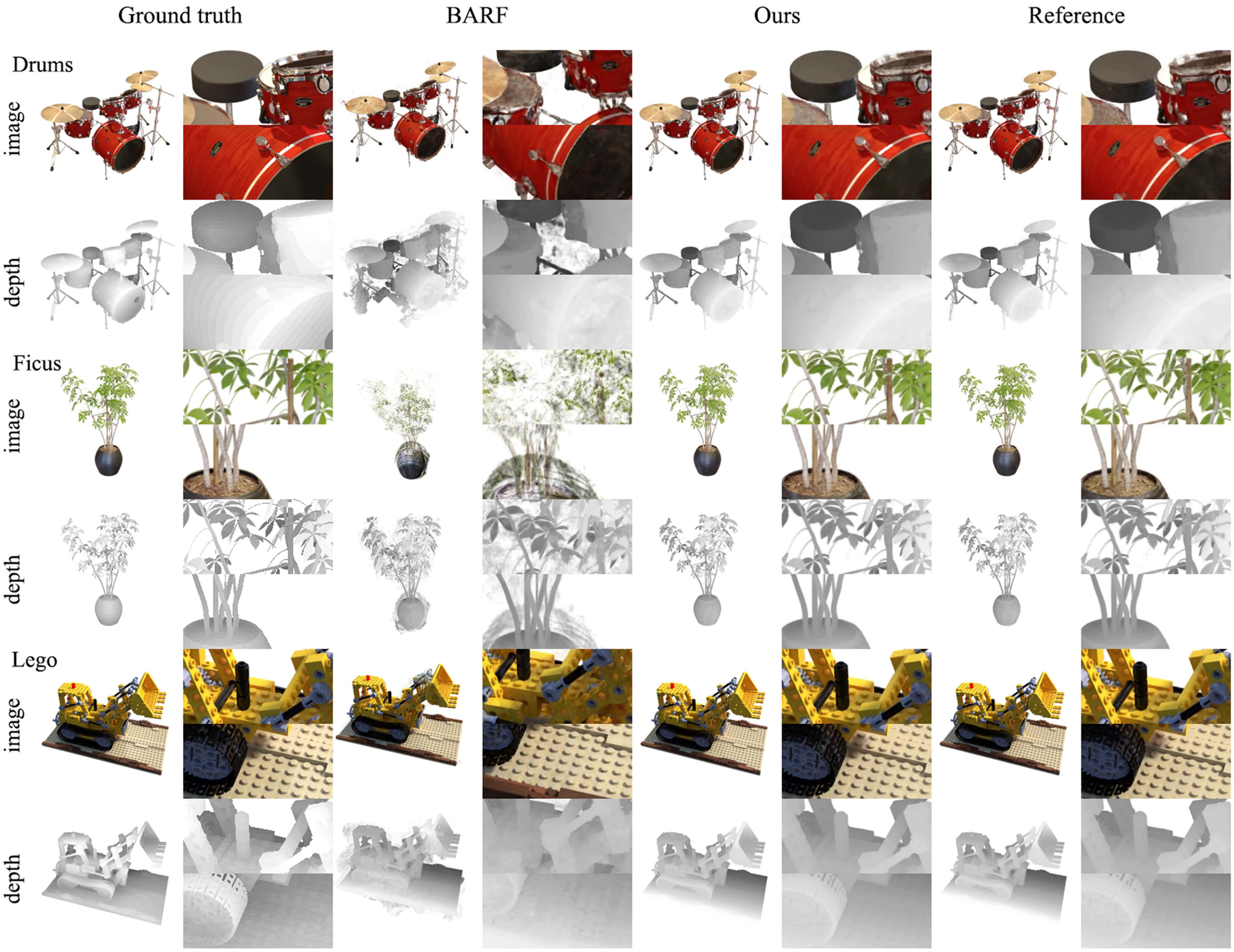


Fig. 10. Qualitative results of rendering on NeRF-synthetic scenes. For each scene, the top row displays the synthesized images, while the bottom row shows the estimated depth. To facilitate comparison, the reference images rendered at perfect camera poses are included on the rightmost column. CBARF achieves high-quality rendering results comparable to the reference rendered images, whereas BARF produces blurry and incorrect renderings due to unsuccessful camera pose optimization.

visualized in Fig. 11, and the quantitative rendering quality is reported in Table. II. The visualization of the pose optimization result is shown in Fig. 8. We use the quality of rendered images as an indirect measure to evaluate the pose optimization capability of different models, as higher rendering quality highly probably indicates more accurate camera pose estimation.

V. Discussion

Our method addresses neural field reconstruction and camera pose registration jointly. As discussed in Section II, there are other 3D reconstruction methods involving pose optimization similar to ours. We also conduct experiments with GARF and L2G-NeRF. However, GARF focuses on handling forward-facing datasets, meaning it can only optimize camera pose correction in roughly 2D space. Therefore, GARF is unable to process the inward-facing data and thus fails to generate appealing results. While L2G-NeRF significantly enhances the capability of reducing camera pose translation errors, surpassing BARF, it diverges within a few iterations when the initial camera poses contain significant rotation errors. In Fig. 12, we present visual results of L2G-NeRF with an average camera pose deviation of 30.4° in rotation and 0.56 in translation.

Despite the success of our methods, there are still areas for improvement. CBARF has similar limitations to the reference model BARF [13], including rigidity assumption and dependence on initial pose information. Some studies [60], [80], [81], [82] discuss pose estimation and tracking, while certain methods [83], [84] draw inspiration from these pose estimation concepts. These methods pre-train a pose estimator, enabling them to perform joint registration and reconstruction on unposed images. Specifically, GNeRF [83] adopts Generative Adversarial Networks (GANs) to extend the NeRF model. It first acquires coarse camera poses and radiance fields through adversarial

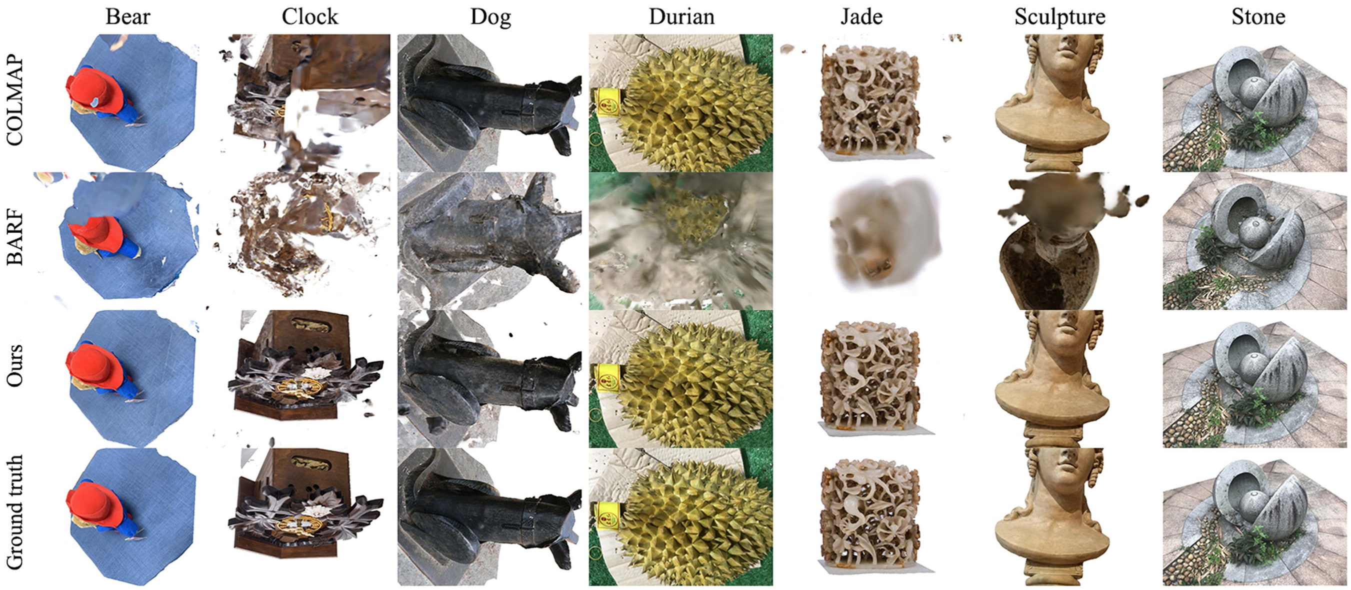
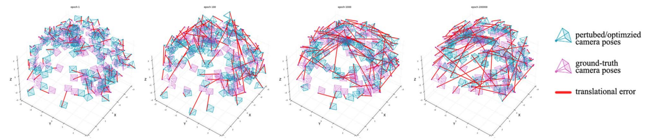


Fig. 11. Rendering results of inward-facing scenes using incomplete BlendedMVS datasets, with 10% of the input images lacking camera pose information. We employ various models to estimate the missing camera poses and generate rendered images at these poses. A closer resemblance between the rendered image and the GT image indicates a more accurate estimation of the missing camera pose. BARF struggles to generate recognizable rendered images, while CBARF exhibits comparable view synthesis quality to the ground-truth images. In most scenes, CBARF also outperforms the reference model COLMAP.

TABLE II
QUANTITATIVE EVALUATION OF THE RENDERED IMAGES OBTAINED AFTER OPTIMIZING THE UNKNOWN CAMERA POSES

Scene	View synthesis quality								
	PSNR↑			SSIM↑			LPIPS↓		
	BARF	CBARF	COLMAP	BARF	CBARF	COLMAP	BARF	CBARF	COLMAP
Bear	10.64	**23.85**	21.68	0.525	**0.718**	0.698	0.583	**0.318**	0.339
Clock	8.75	**16.68**	9.74	0.443	**0.672**	0.539	0.593	**0.415**	0.542
Dog	10.11	**19.43**	19.25	0.377	**0.680**	0.669	0.597	**0.341**	0.351
Durian	10.33	25.09	**25.87**	0.302	0.789	**0.809**	0.752	0.292	**0.279**
Jade	11.57	**22.31**	17.54	0.681	**0.812**	0.684	0.424	**0.234**	0.375
Sculture	11.06	**29.11**	26.86	0.691	**0.934**	0.915	0.400	**0.110**	0.126
Stone	12.63	**26.89**	26.70	0.210	**0.792**	0.777	0.602	**0.230**	0.235
Mean	11.38	**24.23**	21.75	0.503	**0.793**	0.749	0.522	**0.250**	0.294

The rendering quality of images at the camera poses estimated by CBARF surpasses that of BARF and COLMAP. This indicates that CBARF successfully optimizes camera poses from incomplete datasets. Moreover, the camera poses estimated from CBARF are more accurate than those estimated by other methods.
The bold values represent the best results obtained among all methods (excluding reference values).



(a) initial camera poses (b) 100 iterations (c) 1k iterations (d) 200k iterations

Fig. 12. Visual result of L2G-NeRF with an initial camera pose deviation of 30.4° in rotation and 0.56 in translation. Under the condition of the initial camera pose with significant rotation errors, L2G-NeRF diverges within a few iterations.

training and subsequently refines them jointly. IR-NeRF [84] develops GNeRF with pose regularization to refine the pose estimator with unposed real images. It constructs a scene codebook, encoding scene features and implicitly capturing scene-specific camera pose distribution as priors. It also introduces implicit regularization to enhance the robustness of pose estimation for real images.

However, these methods still rely on prior knowledge of camera distribution, and their accuracy in camera pose estimation is limited (Table III). In some scenarios, they may even fail to estimate camera poses and reconstruct scenes. We present visualizations of unsuccessful reconstruction results for GNeRF [83] in Fig. 13. Thus, when camera pose priors exist and there is a high demand for optimization precision, CBARF becomes a preferable choice. Moreover, these pose-free methods involve more complicated procedures, involving approximate camera pose estimation, coarse NeRF learning, and collaborative refinement of both camera pose and NeRF. As a result, GNeRF requires approximately 32 hours for training.

TABLE III
QUANTITATIVE COMPARISON OF CAMERA POSES ACCURACY BETWEEN GNERF, IR-NERF, AND CBARF (NOISE COEFFICIENT IS 0.35) ON THE NERF-SYNTHETIC DATASET

Scene	Camera pose optimization					
	Rotation (°)↓			Translation↓		
	GNeRF	IR-NeRF	CBARF	GNeRF	IR-NeRF	CBARF
Chair	0.363	0.251	**0.099**	0.018	0.013	**0.005**
Drums	0.204	0.185	**0.042**	0.010	0.008	**0.001**
Ficus	-	-	**0.083**	-	-	**0.004**
Hotdog	2.349	1.932	**0.248**	0.122	0.098	**0.013**
Lego	0.430	0.371	**0.073**	0.023	0.015	**0.003**
Materials	-	-	**0.047**	-	-	**0.002**
Mic	1.865	1.598	**0.063**	0.031	0.019	**0.003**
Ship	3.721	3.253	**1.099**	0.176	0.125	**0.009**
Mean	1.489	1.265	**0.219**	0.063	0.046	**0.005**

We report rotation errors and translation errors of estimated camera poses after training. GNeRF and IR-NeRF fail to reconstruct ficus and materials scenes.
The bold values represent the best results obtained among all methods (excluding-reference values).

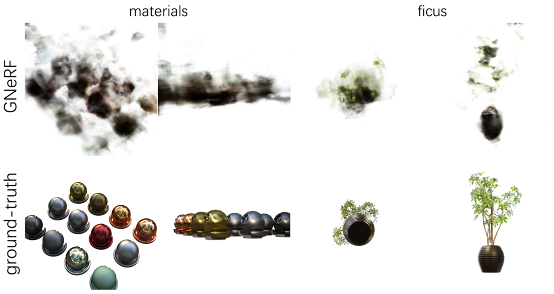


Fig. 13. 3D reconstruction results of GNeRF from unposed images. GNeRF fails in certain scenes, such as ficus and materials, because GNeRF may not initialize coarse poses when the images of a scene are highly similar.

VI. CONCLUSION

In this paper, we propose CBARF (Cascaded Bundle-Adjusting Neural Radiance Fields), a novel 3D reconstruction model aiming to effectively optimize imperfect camera poses. We demonstrate the significance of camera pose initialization for the performance of bundle-adjustment (BA). Consequently, we introduce the cascaded BA to progressively refine the camera poses. Then our proposed neighbor-replacement strategy effectively rectifies erroneous poses that cannot be automatically optimized in the BA process. We also design a novel criterion to identify such poorly estimated poses without relying on ground-truth. Our experiments demonstrate the superiority of our CBARF model in both camera pose optimization and novel view synthesis. Future research can explore extensions of the CBARF model for more complex scenes and reduced reliance on initial pose information. We believe CBARF opens up new possibilities for 3D reconstruction framework with unknown camera poses.

ACKNOWLEDGMENT

The authors would like to thank all anonymous reviewers and ACs for their constructive suggestions.

REFERENCES

[1] A. Geiger, J. Ziegler, and C. Stiller, “Stereoscan: Dense 3D reconstruction in real-time,” in *Proc. IEEE Intell. Veh. Symp.*, 2011, pp. 963–968.

[2] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd, “Real time localization and 3d reconstruction,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2006, pp. 363–370.

[3] Z. Kang, J. Yang, Z. Yang, and S. Cheng, “A review of techniques for 3d reconstruction of indoor environments,” *ISPRS Int. J. Geo- Inf.*, vol. 9, no. 5, 2020, Art. no. 330.

[4] S. Izadi et al., “KinectFusion: Real-time 3D reconstruction and interaction using a moving depth camera,” in *Proc. 24th Annu. ACM Symp. User Interface Softw. Technol.*, 2011, pp. 559–568.

[5] M. Pollefeys et al., “Detailed real-time urban 3D reconstruction from video,” *Int. J. Comput. Vis.*, vol. 78, pp. 143–167, 2008.

[6] J. L. Schonberger and J.-M. Frahm, “Structure-from-motion revisited,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4104–4113.

[7] S. Avidan and A. Shashua, “Novel view synthesis in tensor space,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 1997, pp. 1034–1040.

[8] E. Park, J. Yang, E. Yumer, D. Ceylan, and A. C. Berg, “Transformation-grounded image generation network for novel 3d view synthesis,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3500–3509.

[9] G. Riegler and V. Koltun, “Free view synthesis,” in *Proc. 16th Euro. Conf. Comput. Vis.*, 2020, pp. 623–640.

[10] T. Zhou, R. Tucker, J. Flynn, G. Fyffe, and N. Snavely, “Stereo magnification: Learning view synthesis using multiplane images,” in *Proc. SIGGRAPH*, 2018.

[11] B. Mildenhall et al., “Nerf: Representing scenes as neural radiance fields for view synthesis,” *Commun. ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[12] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese, “3D-R2N2: A unified approach for single and multi-view 3D object reconstruction,” in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 628–644.

[13] C.-H. Lin, W.-C. Ma, A. Torralba, and S. Lucey, “Barf: Bundle-adjusting neural radiance fields,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 5741–5751.

[14] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon, “Bundle adjustment–a modern synthesis,” *Vis. Algorithms: Theory Pract.*, vol. 5, no. 6, pp. 298–372, 1999.

[15] C. Wu, S. Agarwal, B. Curless, and S. M. Seitz, “Multicore bundle adjustment,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* 2011, pp. 3057–3064.

[16] S. Agarwal, N. Snavely, S. M. Seitz, and R. Szeliski, “Bundle adjustment in the large,” in *Proc. 11th Eur. Conf. Comput. Vis.*, 2010, pp. 29–42.

[17] C. Zach, “Robust bundle adjustment revisited,” in *Proc. 13th Eur. Conf. Comput. Vis.*, 2014, pp. 772–787.

[18] C. Engels, H. Stewénius, and D. Nistér, “Bundle adjustment rules,” *Photogrammetric Comput. Vis.*, vol. 2, no. 32, 2006.

[19] A. Neubeck and L. Van Gool, “Efficient non-maximum suppression,” *Image Vis. Comput.*, vol. 24, no. 5, pp. 565–571, 2006.

[20] J. Hosang, R. Benenson, and B. Schiele, “Learning non-maximum suppression,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4507–4515.

[21] J. Hosang, R. Benenson, and B. Schiele, “A convnet for non-maximum suppression,” in *Proc. Pattern Recognition: 38th German Conf.*, 2016, pp. 192–204.

[22] C. Sun, M. Sun, and H.-T. Chen, “Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 5459–5469.

[23] Y. Yao et al., “Blendedmvs: A large-scale dataset for generalized multi-view stereo networks,” in *Proc. Comput. Vis. Pattern Recognit.*, 2020, pp. 1790–1799.

[24] D. Cernea, “OpenMVS: Multi-view stereo reconstruction library,” 2020. [Online]. Available: https://cdcseacave.github.io/openMVS

[25] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[26] O. Özyeşil, V. Voroninski, R. Basri, and A. Singer, “A survey of structure from motion*,” *Acta Numerica*, vol. 26, pp. 305–364, 2017.

[27] S. Ullman, “The interpretation of structure from motion,” *Proc. Roy. Soc. London. Ser. B. Biol. Sci.*, vol. 203, no. 1153, pp. 405–426, 1979.

[28] R. A. Andersen and D. C. Bradley, “Perception of three-dimensional structure from motion,” *Trends Cogn. Sci.*, vol. 2, no. 6, pp. 222–228, 1998.

[29] S. Y. Bao and S. Savarese, “Semantic structure from motion,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 2025–2032.

[30] C. Liu, D. Kong, S. Wang, J. Li, and B. Yin, “Dlgan: Depth-preserving latent generative adversarial network for 3d reconstruction,” *IEEE Trans. Multimedia*, vol. 23, pp. 2843–2856, 2021.

[31] C. Yan et al., “3D room layout estimation from a single RGB image,” *IEEE Trans. Multimedia*, vol. 22, no. 11, pp. 3014–3024, Nov. 2020.

[32] P. Hu, E. S. Ho, and A. Munteanu, “3DBodyNet: Fast reconstruction of 3D animatable human body shape from a single commodity depth camera,” *IEEE Trans. Multimedia*, vol. 24, pp. 2139–2149, 2022.

[33] D. S. Alexiadis, D. Zarpalas, and P. Daras, “Real-time, full 3-D reconstruction of moving foreground objects from multiple consumer depth cameras,” *IEEE Trans. Multimedia*, vol. 15, no. 2, pp. 339–358, Feb. 2013.

[34] R. Liu, Y. Cheng, S. Huang, C. Li, and X. Cheng, “Transformer-based high-fidelity facial displacement completion for detailed 3d face reconstruction,” *IEEE Trans. Multimedia*, vol. 26, pp. 799–810, 2024.

[35] X. Zuo et al., “Sparsefusion: Dynamic human avatar modeling from sparse rgbd images,” *IEEE Trans. Multimedia*, vol. 23, pp. 1617–1629, 2021.

[36] Z. Liu et al., “Deep view synthesis via self-consistent generative network,” *IEEE Trans. Multimedia*, vol. 24, pp. 451–465, 2022.

[37] H. Zhang et al., “3D human pose and shape reconstruction from videos via confidence-aware temporal feature aggregation,” *IEEE Trans. Multimedia*, vol. 25, pp. 3868–3880, 2023.

[38] V. Sadbhawna et al., “Context region identification based quality assessment of 3d synthesized views,” *IEEE Trans. Multimedia*, vol. 25, pp. 6183–6193, 2023.

[39] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, “Occupancy networks: Learning 3d reconstruction in function space,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4460–4470.

[40] X.-F. Han, H. Laga, and M. Bennamoun, “Image-based 3d object reconstruction: State-of-the-art and trends in the deep learning era,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 5, pp. 1578–1604, May 2021.

[41] J. Zhang et al., “3D reconstruction for motion blurred images using deep learning-based intelligent systems,” *Comput., Mater. Continua*, vol. 66, no. 2, pp. 2087–2104, 2021.

[42] X. Wang, Y. Guo, Z. Yang, and J. Zhang, “Prior-guided multi-view 3D head reconstruction,” *IEEE Trans. Multimedia*, vol. 24, pp. 4028–4040, 2022.

[43] S. Shen et al., “SD-NeRF: Towards lifelike talking head animation via spatially-adaptive dual-driven nerfs,” *IEEE Trans. Multimedia*, vol. 26, pp. 3221–3234, 2024.

[44] A. Yu, V. Ye, M. Tancik, and A. Kanazawa, “pixelNeRF: Neural radiance fields from one or few images,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 4578–4587.

[45] S. Fridovich-Keil et al., “Plenoxels: Radiance fields without neural networks,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 5501–5510.

[46] R. Martin-Brualla et al., “Nerf in the wild: Neural radiance fields for unconstrained photo collections,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 7210–7219.

[47] J. T. Barron et al., “Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 5855–5864.

[48] A. Pumarola, E. Corona, G. Pons-Moll, and F. Moreno-Noguer, “D-NeRF: Neural radiance fields for dynamic scenes,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 10318–10327.

[49] S.-F. Chng, S. Ramasinghe, J. Sherrah, and S. Lucey, “Gaussian activated neural radiance fields for high fidelity reconstruction and pose estimation,” in *Proc. 17th Euro. Conf. Comput. Vis.*, 2022, pp. 264–280.

[50] P. Truong, M.-J. Rakotosaona, F. Manhardt, and F. Tombari, “Sparf: Neural radiance fields from sparse and noisy poses,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 4190–4200.

[51] Z. Wang, S. Wu, W. Xie, M. Chen, and V. A. Prisacariu, “NeRF–: Neural radiance fields without known camera parameters,” 2021, *arXiv:2102.07064*.

[52] Y. Chen et al., “Local-to-global registration for bundle-adjusting neural radiance fields,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 8264–8273.

[53] K. Park, P. Henzler, B. Mildenhall, J. T. Barron, and R. Martin-Brualla, “Camp: Camera preconditioning for neural radiance fields,” *ACM Trans. Graph.*, vol. 42, no. 6, pp. 1–11, 2023.

[54] Y. Chen and G. H. Lee, “DBARF: Deep bundle-adjusting generalizable neural radiance fields,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 24–34.

[55] D. Wang, X. Cui, S. Salcudean, and Z. J. Wang, “Generalizable neural radiance fields for novel view synthesis with transformer,” 2022, *arXiv:2206.05375*.

[56] C. Smith, Y. Du, A. Tewari, and V. Sitzmann, “Flowcam: Training generalizable 3D radiance fields without camera poses via pixel-aligned scene flow,” in *Proc. NeurIPS*, 2023.

[57] P. Dollár, P. Welinder, and P. Perona, “Cascaded pose regression,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 1078–1085.

[58] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, “What is the best multi-stage architecture for object recognition,” in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, 2009, pp. 2146–2153.

[59] S. W. Zamir et al., “Multi-stage progressive image restoration,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 14821–14831.

[60] X. Yu, F. Porikli, B. Fernando, and R. Hartley, “Hallucinating unaligned face images by multiscale transformative discriminative networks,” *Int. J. Comput. Vis.*, vol. 128, no. 2, pp. 500–526, 2020.

[61] Y. Zheng, X. Yu, M. Liu, and S. Zhang, “Residual multiscale based single image deraining,” in *Proc. Brit. Mach. Vis. Conf.*, K. Sidorov and Y. Hicks, Eds., Sep. 2019, pp. 27.1–27.12, doi: 10.5244/C.33.27.

[62] X. Yu, F. Xu, S. Zhang, and L. Zhang, “Efficient patch-wise non-uniform deblurring for a single image,” *IEEE Trans. Multimedia*, vol. 16, no. 6, pp. 1510–1524, Oct. 2014.

[63] Q. Huynh-Thu and M. Ghanbari, “The scope of PSNR in image/video quality assessment,” *IEEE Trans. Image Process.*, vol. 17, no. 8, pp. 1251–1260, 2008.

[64] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[65] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, “The unreasonable effectiveness of deep features as a perceptual metric,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 586–595.

[66] Y. Chen, J. Zhao, and L. Kneip, “Hybrid rotation averaging: A fast and robust rotation averaging approach,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 10358–10367.

[67] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “ORB: An efficient alternative to SIFT or SURF,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 2564–2571.

[68] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[69] J. Sun, Z. Shen, Y. Wang, H. Bao, and X. Zhou, “LoFTR: Detector-free local feature matching with transformers,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 8922–8931.

[70] T. Cover and P. Hart, “Nearest neighbor pattern classification,” *IEEE Trans. Inform. Theory*, vol. 13, no. 1, pp. 21–27, Jan. 1967.

[71] S. Hao, Y. Cheng, X. Ma, and J. Zhao, “A fast detection algorithm of cooperative object corners based on SIFT partition bidirectional matching,” in *Proc. IEEE 6th Int. Symp. Comput. Intell. Des.*, 2013, pp. 59–62.

[72] A. Wu, W. Chen, Y. Bian, and S. Xue, “Image matching algorithm based on topology consistency of bidirectional optimal matching point pairs,” *Sensors Mater.*, vol. 34, no. 2, pp. 493–514, 2022.

[73] Z. Wang, E. P. Simoncelli, and A. C. Bovik, “Mean squared error: Love it or leave it? a new look at signal fidelity measures,” *IEEE Signal Process. Mag.*, vol. 26, no. 1, pp. 98–117, Jan. 2009.

[74] C. Shorten and T. M. Khoshgoftaar, “A survey on image data augmentation for deep learning,” *J. Big Data*, vol. 6, no. 1, 2019, Art. no. 60.

[75] A. Mishra et al., “Lie groups in computer vision and image processing: A survey,” *J. Math. Imag. Vis.*, vol. 47, no. 3, pp. 209–252, 2013.

[76] C. Triola, “Special orthogonal groups and rotations,” 2009. [Online]. Available: https://api.semanticscholar.org/CorpusID:125837960

[77] L. N. Smith, “Cyclical learning rates for training neural networks,” in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2017, pp. 464–472.

[78] N. R. Carlson, F. N. Fritsch, and F. Y. Kuo, “Fast and accurate exponential interpolation,” *SIAM J. Numer. Anal.*, vol. 32, no. 3, pp. 674–694, 1995.

[79] J. C. Gower, “Procrustes methods in the statistical analysis of shape,” *J. Roy. Stat. Soc.: Ser. B. (Methodological)*, vol. 64, no. 4, pp. 643–682, 2002.

[80] C. Han, X. Yu, C. Gao, N. Sang, and Y. Yang, “Single image based 3d human pose estimation via uncertainty learning,” *Pattern Recognit.*, vol. 132, 2022, Art. no. 108934.

[81] X. Yu, S. Zhang, X. Zhao, and L. Zhang, “Removing blur kernel noise via a hybrid p norm,” *J. Electron. Imag.*, vol. 24, no. 1, pp. 013011–013011, 2015.

[82] J. Liu et al., “Leaping from 2D detection to efficient 6DoF object pose estimation,” in *Proc. Comput. Vis.*, 2020, pp. 707–714.

[83] Q. Meng et al., “GNeRF: GAN-based neural radiance field without posed camera,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 6331–6341.

[84] J. Zhang et al., “Pose-free neural radiance fields via implicit pose regularization,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 3534–3543.