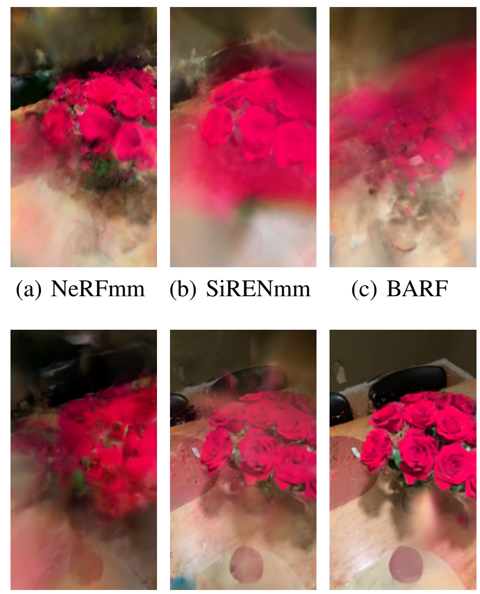
CF-NeRF: Camera Parameter Free Neural Radiance Fields with Incremental Learning

Qingsong Yan[1], Qiang Wang[2,*], Kaiyong Zhao[3], Jie Chen[4], Bo Li[5], Xiaowen Chu[6,5,*], Fei Deng[1,7]

[1]Wuhan University, Wuhan, China, [2]Harbin Institute of Technology (Shenzhen), Shenzhen, China
[3]XGRIDS, Shenzhen, China, [4]Hong Kong Baptist University, Hong Kong SAR, China
[5]The Hong Kong University of Science and Technology, Hong Kong SAR, China
[6]The Hong Kong University of Science and Technology (Guangzhou),Guangzhou, China
[7]Hubei Luojia Laboratory, Wuhan, China
yanqs_whu@whu.edu.cn, qiang.wang@hit.edu.cn, kyzhao@xgrids.com, chenjie@comp.hkbu.edu.hk
bli@cse.ust.hk, xwchu@ust.hk, fdeng@sgg.whu.edu.cn

Abstract

Neural Radiance Fields have demonstrated impressive performance in novel view synthesis. However, NeRF and most of its variants still rely on traditional complex pipelines to provide extrinsic and intrinsic camera parameters, such as COLMAP. Recent works, like NeRFmm, BARF, and L2G-NeRF, directly treat camera parameters as learnable and estimate them through differential volume rendering. However, these methods work for forward-looking scenes with slight motions and fail to tackle the rotation scenario in practice. To overcome this limitation, we propose a novel camera parameter free neural radiance field (CF-NeRF), which incrementally reconstructs 3D representations and recovers the camera parameters inspired by incremental structure from motion. Given a sequence of images, CF-NeRF estimates camera parameters of images one by one and reconstructs the scene through initialization, implicit localization, and implicit optimization. To evaluate our method, we use a challenging real-world dataset, NeRFBuster, which provides 12 scenes under complex trajectories. Results demonstrate that CF-NeRF is robust to rotation and achieves state-of-the-art results without providing prior information and constraints.



(a) NeRFmm (b) SiRENmm (c) BARF (d) GARF (e) L2G-NeRF (f) CF-NeRF

Figure 1: We select a sequence from NeRFBuster (Warburg et al. 2023) and use novel views synthesis to compare the quality of camera parameters from NeRFmm (Wang et al. 2021b), SiRENmm (Guo and Sherwood 2021), BARF (Lin et al. 2021), GARF (Chng et al. 2022), L2G-NeRF (Chen et al. 2023) and our method CF-NeRF.

Introduction

3D reconstruction is a hot topic in computer vision that aims to recover 3D geometry from RGB images. However, traditional methods contain lots of complex procedures, such as feature extraction and matching (Lowe 2004; Yi et al. 2016), sparse reconstruction (Agarwal et al. 2011; Wu 2013; Schonberger and Frahm 2016; Moulon et al. 2016), and dense reconstruction (Yao et al. 2018; Mi, Di, and Xu 2022; Yan et al. 2023). Consequently, traditional methods are not a differential end-to-end reconstruction pipeline and require high-quality results from each sub-module to achieve accurate results. When the quality of results is poor, it is challenging to identify which module is causing the problem.

Recently, Neural Radiance Fields (NeRF) (Mildenhall et al. 2020; Yu et al. 2021a; Müller et al. 2022) have demonstrated a novel way to render highly realistic novel views with impressive quality. Without recovering 3D geometry, NeRF relies on multi-layer perception (MLP) to predict color and sigma for each point in the scene and samples several points along a ray to render a pixel through differential volume rendering. Unlike traditional 3D reconstruction, NeRF simplifies the reconstruction into one step and implicitly represents the 3D scene. Benefiting from the excellent ability of NeRF, it has been further extended to dynamic scenes (Pumarola et al. 2021), large-scale (Turki, Ramanan, and Satyanarayanan 2022), and even surface (Wang et al. 2021a) and material reconstruction (Boss et al. 2021a).

[*]Corresponding author

Despite the remarkable performance of NeRF and its variants in novel view synthesis, they still require camera parameters before training. The most common processing pipeline is first recovering camera parameters using traditional complex methods (Schonberger and Frahm 2016; Moulon et al. 2016), and then training the NeRF through differential volume rendering. In other words, the differentiability of the whole reconstruction pipeline is destroyed and divided into two separate parts, resulting in the NeRF not being end-to-end and the reconstruction quality being unidirectionally dependent on traditional methods.

To unify camera parameter estimation and reconstruction, researchers have tried to recover or optimize camera parameters along with NeRF. The straightforward idea is to treat camera parameters as learnable, as NeRFmm (Wang et al. 2021b) does. BARF (Lin et al. 2021) recovers extrinsic camera parameters and the NeRF model by dynamically adjusting weights of different frequencies of positional encoding. GARF (Chng et al. 2022) replaces ReLU with Gaussian activations to obtain high-accuracy results. NeROIC (Kuang et al. 2022) and NeRFStudio (Tancik et al. 2023) optimize camera parameters and the NeRF simultaneously. However, these methods are only suitable for forward-looking scenes or scenes with initial camera parameters and cannot be directly used in the real world with complex movement.

This paper proposes a new end-to-end approach called camera parameter free NeRF (CF-NeRF) to address the limitations of existing NeRF-based methods in estimating camera parameters. Figure 1 compares rendered novel views by camera parameters estimated by several methods (Wang et al. 2021b; Guo and Sherwood 2021; Lin et al. 2021; Chng et al. 2022; Chen et al. 2023) and our method CF-NeRF, where CF-NeRF is the only method that successfully reconstructs the 3D scene with rotation. Unlike other methods that simultaneously estimate all camera parameters, CF-NeRF inherits ideas from incremental structure from motion (SfM) and recovers camera parameters one by one. CF-NeRF contains three major components: initialization, implicit localization, and implicit optimization. CF-NeRF uses initialization to recover camera parameters and NeRF by a few images and estimates camera parameters of other images through two steps: the implicit localization provides an initial camera parameter for the newly added image, and the implicit optimization optimizes camera parameters of all images to reduce drift. Our contributions are as follows:

1. We propose a novel end-to-end method, CF-NeRF, that does not need prior information or constraints to recover the intrinsic and extrinsic camera parameters and the NeRF simultaneously.
2. We design an incremental training pipeline for the CF-NeRF, inspired by the incremental SfM, to avoid trapping to local minimal and is suitable for complex trajectories.
3. Experiments of our method achieve state-of-the-art results on the NeRFBuster dataset (Warburg et al. 2023) captured in the real world, proving that the CF-NeRF can estimate accurate camera parameters with the specifically designed training procedure.

Related Work

In this section, we introduce the development of NeRF-related methods with known camera parameters and several camera parameter estimation methods using SfM&SLAM (simultaneous localization and mapping) and the NeRF.

NeRF

NeRF (Mildenhall et al. 2020) uses the MLP to represent the 3D scene implicitly and can be trained through differential volume rendering from a set of images with known camera parameters. However, NeRF suffers from efficiency and needs around 1-2 days to train a scene and several minutes to render a novel view at the testing. Instant-NGP (Müller et al. 2022) builds a multi-resolution hash table to store space-aware feature vectors and reduces the complexity of the MLP network. Meanwhile, (Sun, Sun, and Chen 2022; Fridovich-Keil et al. 2022) try to use the coarse-to-fine strategy and (Yu et al. 2021a; Chen et al. 2022; Garbin et al. 2021) update the network structure to speed up training or testing. Besides, NeRF faces another problem that it cannot work for large-scale, unbounded 3D scenes. NeRF++ (Zhang et al. 2020) and MipNeRF360 (Barron et al. 2021, 2022) utilize different sampling strategies for foreground and background to model unbounded 3D scenes by a finite volume. MegaNeRF (Turki, Ramanan, and Satyanarayanan 2022) and BlockNeRF (Tancik et al. 2022) split a large scene into multiple small regions and assign a network for each part. Moreover, (Martin-Brualla et al. 2021; Pumarola et al. 2021; Attal et al. 2021) extend NeRF to dynamic scenes and (Jain, Tancik, and Abbeel 2021; Yu et al. 2021b; Niemeyer et al. 2022; Kim, Seo, and Han 2022) introduce context or geometry information into NeRF to suit scenes with sparse views. In addition to the advances in novel view synthesis, NeRF has made significant progress in geometric reconstruction (Yariv et al. 2021; Wang, Skorokhodov, and Wonka 2022; Darmon et al. 2022; Long et al. 2023; Fu et al. 2022). UniSURF (Oechsle, Peng, and Geiger 2021) and NeUS (Wang et al. 2021a) estimate the zero-level set of an implicit signed distance function instead of the space density. Furthermore, some work (Zhang et al. 2021; Verbin et al. 2022; Boss et al. 2021a; Kuang et al. 2022; Boss et al. 2021b, 2022) even combines BRDF and NeRF to decompose a scene into shape, reflectance, and illumination. However, all of these methods split the reconstruction into two steps and require traditional methods to provide camera parameters, which significantly limits the application of NeRF.

Camera Parameter Estimation

Traditional SfM (Wu 2013; Moulon, Monasse, and Marlet 2013; Schonberger and Frahm 2016; Moulon et al. 2016) and SLAM (Mur-Artal, Montiel, and Tardos 2015; Engel, Koltun, and Cremers 2017) can estimate camera parameters for given images. However, these methods divide the reconstruction pipeline into several non-differentiable modules that need hand-crafted features (Lowe 2004) or learning-based methods (Yi et al. 2016; Teed and Deng 2020) to establish image correspondences, and then reconstruct a sparse scene and camera parameters through multi-view geometry.

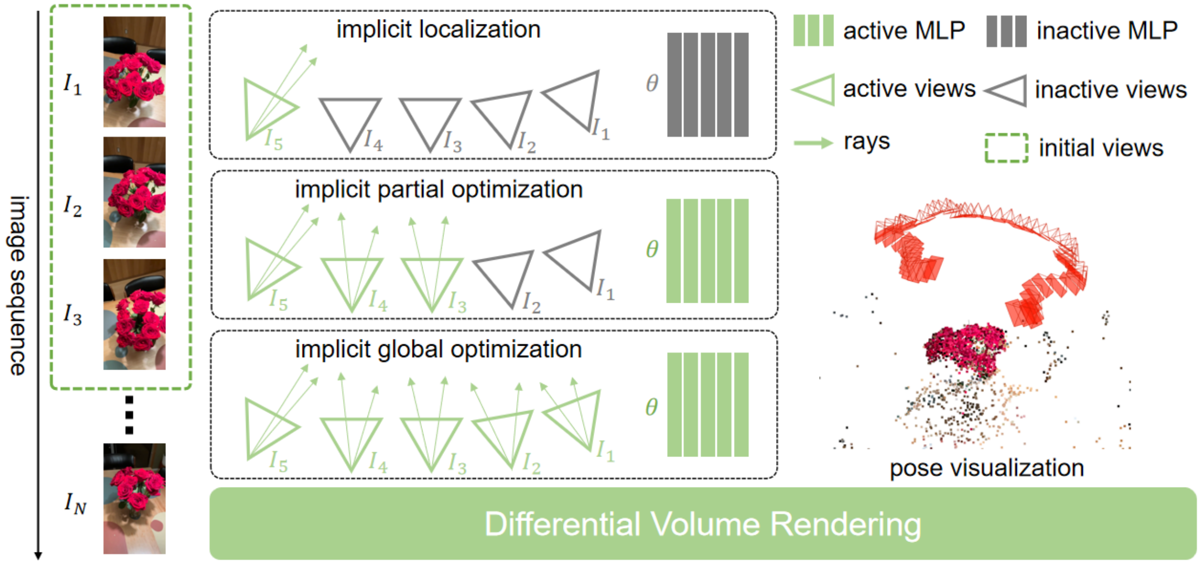


Figure 2: The pipeline of CF-NeRF. CF-NeRF can estimate the weight θ of NeRF $\mathcal{F}$ and the camera parameter δ. After initializing through a few selected images, CF-NeRF recovers δ of the image one by one through implicit localization that only optimizes the newly added image and implicit optimization that refines θ and δ. Implicit optimization can be divided into partial and global optimization depending on the number of images used. We visualize δ reconstructed by CF-NeRF and sparse points from COLMAP (Schonberger and Frahm 2016) to show that CF-NeRF can reconstruct rotation in image sequences.

In light of these limitations, it is worth exploring to estimate camera parameters during the training process of NeRF. The most direct attempt to utilize NeRF is the visual localization, where iNeRF (Yen-Chen et al. 2021), NeDDF (Ueda et al. 2022), and PNeRFP (Lin et al. 2023) try to estimate the extrinsic camera parameter of a new image by a pre-trained NeRF model. Then, NeRFmm (Wang et al. 2021b) and SiRENmm (Guo and Sherwood 2021) take the NeRF and camera parameters as learnable and prove that it is possible to train the NeRF model from scratch without camera parameters, but they only work for forward-looking scenes. To further enhance accuracy in forward-looking or rotation scenes with initial camera parameters, BARF (Lin et al. 2021) dynamically adjusts the weight of the positional encoding, GARF (Chng et al. 2022) replaces the ReLU activate function with the Gaussian activation function, and L2G-NeRF (Chen et al. 2023) introduces a local-to-global registration. Interestingly, GNeRF (Meng et al. 2021) and VMRF (Zhang et al. 2022) assume there is a prior known distribution of camera parameters to decrease the freedom of camera parameters during training the NeRF model. Meanwhile, other researchers try to add different external restrictions to guide the camera parameter estimation. SCNeRF (Jeong et al. 2021) and Level-S^2fM (Xiao et al. 2023) rely on feature matches to guide camera parameters estimation. NoPe-NeRF (Bian et al. 2023), iMap (Sucar et al. 2021), NeRF-SLAM (Rosinol, Leonard, and Carlone 2022), Nice-SLAM (Zhu et al. 2022), and Nicer-SLAM (Zhu et al. 2023) integrate depth maps from active sensors or CNN networks to tune the NeRF. Additionally, LocalLR (Meuleman et al. 2023) combines depth maps and optical flow to train NeRF.

Regrettably, images acquired from real-world scenarios often exhibit a multitude of challenges. These challenges include rotations and the absence of prior information of camera parameters. Furthermore, the introduction of external constraints can augment the intricacy and unpredictability of the reconstruction process. To solve these problems, we propose CF-NeRF inspired by the traditional incremental SfM, which does not require any prior information or external constraints while reconstructing the 3D scene and camera parameters end-to-end from image sequences, demonstrating the powerful reconstruction capability of the NeRF after using a specific training strategy.

Method

In this section, we provide an overview of the proposed method. Firstly, we introduce the preliminary background of the NeRF and the traditional incremental SfM. Then, we explain the details of CF-NeRF that can recover camera parameters from image sequences.

Preliminary Background

NeRF NeRF can generate realistic images from a set of images $I = (I_1, I_2, ..., I_N)$ from N different places without explicitly reconstructing. However, NeRF needs associated camera parameters δ, including camera rotation $\delta_R = (\delta_{R_1}, \delta_{R_2}, ..., \delta_{R_N})$, camera translation $\delta_T = (\delta_{T_1}, \delta_{T_2}, ..., \delta_{T_N})$, and intrinsic camera parameter δ_K. Given a NeRF model $\mathcal{F}$ and corresponding weight θ , it can estimate color c and density σ through a implicit function $c(x, \vec{d}), \sigma(x) = \mathcal{F}_\theta(x, \vec{d})$ with a point x and a view direction

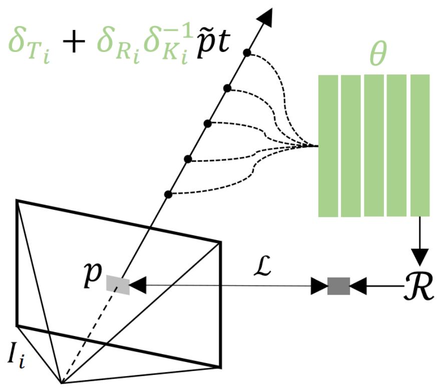


Figure 3: Estimated Parameters. CF-NeRF estimates the weight θ of NeRF model and the camera parameter δ, which include the camera rotation δ_R, the camera translation δ_T, and camera intrinsic parameter δ_K.

$\vec{d}$. To render a pixel p, NeRF needs to sample several points $x_p(t) = o + \vec{d}t$ along a ray shooting from the view position o and generate the color c_p by the volume rendering function $\mathcal{R}$ as Eq. 1 shows, where $\mathcal{T}(t) = exp(-\int_{t_n}^{t} \sigma(x_p(s))ds)$ indicates the accumulated transmittance along the ray. t_n and t_f are the near and far bounds of the ray.

$$c_p = \mathcal{R}(p|\theta) = \int_{t_n}^{t_f} \mathcal{T}(t)\sigma(x_p(t))c(x_p(t), \vec{d})dt \quad (1)$$

Benefiting from the differential property of the volume rendering, NeRF can be trained end to end by minimizing the difference between c_p and observed color $I(p)$ as Eq. 2 shows, where $\mathcal{L}$ is the loss function. To be noted, NeRF only estimates θ and borrows δ from traditional SfM methods. However, NeRFmm (Wang et al. 2021b) prove that it is possible to estimate θ and δ simultaneously under the forward-looking situation.

$$\arg\min_{\theta} \{\sum_{I_i \in I} \sum_{p \in I_i} \mathcal{L}(\mathcal{R}(p|\theta), I_i(p))\} \quad (2)$$

Incremental SfM Given a set of images, the incremental SfM can recover δ one by one in a linear time (Wu 2013) and contains four steps (Schonberger and Frahm 2016):

Initialization The selection of an initial two-view is essential because a suitable initial two-view improves the robustness and quality of the reconstruction. With a given two-view and its matched features, incremental SfM computes the relative pose by multi-view geometry (MVG) and triangulates 3D points to initial the scene.

Image Registration After initialization, incremental SfM adds images to the scene in order. Given a new image, incremental SfM builds the 2D-3D relationship by matching its features with images in the scene and recovers the camera parameter by Perspective-n-Point (PnP).

Triangulation As a newly added image observes additional information that can extend the scale of the scene, incremental SfM triangulates more 3D points based on the new image and matched features.

Bundle Adjustment Adding new images and 3D points without refinement leads to drift. Therefore, it is essential to apply bundle adjustment (BA) by minimizing the reprojection error. In terms of efficiency, incremental SfM proposes partial BA that refines only a subset of images, and global BA that optimizes all images.

CF-NeRF

Fusing the differentiability of NeRF and the reconstruction strategy of SfM, we propose CF-NeRF, which is capable of estimating the camera parameter under complex movement from sequential images. CF-NeRF consists of three modules: initialization, implicit localization, and implicit optimization, as Figure 2 shows. To convenient later introduction, we define the set of images we have completed estimating the camera parameter as E, which starts from $\varnothing$.

Parameter CF-NeRF estimates camera parameter δ, which includes δ_R, δ_T, and δ_K, and the weight θ of NeRF, as Figure 3 shows. During the differential volume rendering, we calculate the ray $\vec{r}_p(t) = \delta_{T_i} + \delta_{R_i}\delta_{K_i}^{-1}\tilde{p}t$ of pixel p in image $I_i \in I$, where $\tilde{p}$ is the homogeneous expression of p. Following NeRFmm (Wang et al. 2021b), we use the axis-angle to represent δ_R and assume all images have the same camera intrinsic parameter without distortion so that δ_K only contains the focal length. We initialize δ_R and δ_T to zero, and set δ_K to $53°$ by a common field of view. The activation function determines how to initialize θ. NeRF using ReLU are initialized according to NeRF (Mildenhall et al. 2020), while NeRF using sine are initialized according to SIREN (Sitzmann et al. 2020).

Initialization Similar to incremental SfM, CF-NeRF requires initialize θ, δ_{R_1}, δ_{T_1}, and δ_K before adding images to E. We select the first N_{init} images I_{init} from I to optimise these parameters by Eq. 3 with ξ_{init} iterations. Since the rotation between adjacent images is not large and NeRF is hard to estimate rotation (Lin et al. 2021), we do not estimate the rotation in the initialization to reduce the freedom. After initialization, we add I_1 to E and keep θ, δ_{R_1}, δ_{T_1}, and δ_K but discard other camera parameters. Note that, unlike the initialization in the previous section, the initialization here is data-specific, similar to the warm-up procedure.

$$\arg\min_{\theta, \delta_T, \delta_K} \{\sum_{I_i \in I_{init}} \sum_{p \in I_i} \mathcal{L}(\mathcal{R}(p|\theta, \delta_{T_i}, \delta_K), I_i(p))\} \quad (3)$$

Implicit Localization After initialization, CF-NeRF estimates the camera parameter of the remaining images one by one and determines δ_{R_n} and δ_{T_n} for each new image I_n by localization. Specifically, we first initialize δ_{R_n} and δ_{T_n} by $\delta_{R_{n-1}}$ and $\delta_{T_{n-1}}$, and then optimize them by minimizing Eq. 4 with fixed θ through ξ_{loc} iterations. The localization is similar to iNeRF (Yen-Chen et al. 2021), but CF-NeRF does not have a pre-trained $\mathcal{F}$.

$$\arg\min_{\delta_{R_n}, \delta_{T_n}} \{\sum_{p \in I_n} \mathcal{L}((p|\delta_{R_n}, \delta_{T_n}), I_n(p))\} \quad (4)$$

(a) GT (b) NeRFmm (c) SiRENmm (d) BARF (e) GARF (f) L2G-NeRF (g) CF-NeRF

Figure 4: We select two sequences from NeRFBuster (Warburg et al. 2023) and render novel views to evaluate camera parameters. Our method CF-NeRF generates high-quality images, while results of NeRFmm (Wang et al. 2021b), SiRENmm (Guo and Sherwood 2021), BARF (Lin et al. 2021), GARF (Chng et al. 2022) and L2G-NeRF (Chen et al. 2023) contain lots of noise.

Implicit Optimization Although implicit localization can roughly determine δ_{R_n} and δ_{T_n}, it faces two problems: the observation from I_n is not added to NeRF, and the localization does not take the multi-view consistency into account to reduce drift. Incremental SfM solves these problems using two separate steps: triangulation and BA, while CF-NeRF benefits from the volume rendering and deals with these problems together. However, it is time-consuming to optimize all images in E every time a new image is added. Therefore, CF-NeRF splits optimization into implicit partial optimization and implicit global optimization.

Each time localizing a new image I_n, CF-NeRF performs implicit partial optimization. We select I_n and previous $N_{part}-1$ images to construct the partial image set I_{part}, then optimizes them with ξ_{part} iterations, as Eq.5 shows.

$$\underset{\theta,\delta_R,\delta_T}{\arg\min}\{\sum_{I_i\in I_{part}}\sum_{p\in I_i}\mathcal{L}(\mathcal{R}(p|\theta,\delta_{R_i},\delta_{T_i}),I_i(p))\} \quad (5)$$

When the number of images in E can be evenly divided by N_{glob}, CF-NeRF employs implicit global optimization for θ and all images in E to enhance the overall accuracy and reduce drifts with ξ_{glob} iterations, as Eq. 6 shows.

$$\underset{\theta,\delta_R,\delta_T,\delta_K}{\arg\min}\{\sum_{I_i\in I_E}\sum_{p\in I_i}\mathcal{L}(\mathcal{R}(p|\theta,\delta_{R_i}\delta_{T_i},\delta_K),I_i(p))\} \quad (6)$$

Coarse-to-Fine CF-NeRF uses a coarse-to-fine strategy to improve robustness. CF-NeRF first constructs a Gaussian pyramid with depth d_G, then recovers all parameters at a low-resolution image through the incremental pipeline. Finally, CF-NeRF directly performs implicit global optimization with a higher resolution in each scale of the Gaussian pyramid with ξ_G iterations.

Loss Function To improve robustness, we employ the Smooth-L1 loss function, as Eq. 7 shows, where gt represents the ground truth, pr is the estimated value, and β is the set to 1.0 by default.

$$\mathcal{L}(pr,gt)=\begin{cases}0.5*(gt-pr)^2/\beta & if|gt-pr|<\beta\\ |gt-pr|-0.5*\beta & otherwise\end{cases} \quad (7)$$

Experiments

Dataset

We evaluate our method using a real-world dataset NeRFBuster (Warburg et al. 2023), mainly rotating around an object. We sample around 50 frames for each scene and resize all images to 480×270 with ground truth (GT) camera parameters from COLMAP (Schonberger and Frahm 2016).

Implementation

CF-NeRF is implemented using PyTorch. Similar to NeRFmm (Wang et al. 2021b), CF-NeRF does not have hierarchical sampling and uses the coarse network, which has eight layers and the dimension of the hidden layers is set to 128. Moreover, we use the sine activation function instead of the ReLU, as SiRENmm (Guo and Sherwood 2021) is more robust than NeRFmm. We utilize the Adam optimizer to optimize all learnable parameters. Specifically, we set the learning rate of θ to 0.001, which undergoes a decay

		aloe	art	car	century	flowers	garbage	picnic	pikachu	pipe	plant	roses	table
$\Delta R\downarrow$	NeRFmm	159.973	177.591	129.580	119.626	106.920	150.823	154.778	113.700	164.821	165.030	102.275	115.299
	SiRENmm	155.151	177.364	127.267	89.0172	103.874	82.9375	44.3671	25.3603	159.757	114.076	132.538	93.2612
	BARF	158.669	59.1868	133.453	101.601	88.6842	88.7832	69.7201	41.0302	64.5250	143.198	133.757	111.288
	GARF	125.980	171.917	153.559	105.187	106.060	84.4992	49.8503	32.7285	126.960	156.606	118.975	164.656
	L2G-NeRF	124.237	24.3753	55.7291	131.968	96.3498	110.478	146.312	116.891	70.9131	70.7562	95.9982	116.497
	CF-NeRF	**12.1226**	**19.2496**	**17.5570**	**9.6811**	**8.2556**	**9.7658**	**12.6501**	**11.3067**	**19.9926**	**4.8968**	**5.1229**	**4.5837**
$\Delta T\downarrow$	NeRFmm	11.5935	15.0762	23.9514	24.5934	12.8753	16.3842	12.9675	25.6841	19.3563	23.6613	8.0367	13.3849
	SiRENmm	11.4912	14.8720	27.7235	28.8582	15.4841	13.3099	8.3607	15.8052	20.4572	31.2943	9.0498	13.8350
	BARF	9.6196	17.3299	36.0351	26.0549	15.3166	17.1936	9.6679	18.7184	18.8936	31.8629	8.5266	17.5453
	GARF	13.9100	17.1527	26.5184	26.0096	14.1459	16.1054	11.7021	17.4975	18.8366	32.3170	11.2654	15.0367
	L2G-NeRF	14.4012	13.4240	20.5634	23.2650	7.4559	17.7167	12.9408	32.4048	11.4012	18.5012	10.7110	12.8061
	CF-NeRF	**3.3788**	**2.2821**	**6.5452**	**2.7383**	**2.7026**	**4.0535**	**1.2833**	**4.0586**	**9.4491**	**3.4346**	**1.1945**	**1.2127**
PSNR$\uparrow$	NeRFmm	20.6912	16.8220	17.7504	16.0326	18.4377	17.7229	18.4300	25.3819	20.0978	21.1558	17.7735	14.1651
	SiRENmm	22.7462	20.3890	22.0268	18.0252	19.4640	17.2283	21.5628	27.8706	20.9538	23.4980	16.7480	18.8135
	BARF	22.4366	21.1947	16.7665	15.3436	17.8350	15.9065	19.1846	23.0386	19.9728	25.5135	13.6741	13.8227
	GARF	19.0241	19.3556	15.4460	14.4117	16.2955	15.3383	15.4035	20.9663	18.5371	20.5600	13.1274	12.6677
	L2G-NeRF	21.3398	19.8099	17.3255	16.6476	18.0016	13.6077	18.5268	22.4939	18.1787	19.0160	17.2614	15.5658
	CF-NeRF	**26.9367**	**26.5293**	**22.4654**	**21.7072**	**21.6950**	**22.4736**	**22.5475**	**32.3661**	**22.2719**	**25.7312**	**24.3918**	**26.8491**
LPIPS$\downarrow$	NeRFmm	0.5560	0.4954	0.5991	0.5793	0.5778	0.5661	0.6113	0.3683	0.5614	0.4927	0.5371	0.6073
	SiRENmm	0.4508	0.4034	0.4450	0.4785	0.5048	0.5193	0.5227	0.2883	0.5170	0.3256	0.5333	0.4659
	BARF	0.3328	0.3511	0.5361	0.5394	0.5552	0.5480	0.5358	0.3440	0.5198	0.3217	0.6138	0.5913
	GARF	0.5257	0.4055	0.5984	0.5845	0.6158	0.5931	0.6086	0.3987	0.5688	0.4345	0.6189	0.6356
	L2G-NeRF	0.4620	0.4186	0.5409	0.5116	0.5466	0.6016	0.5530	0.4051	0.4741	0.3840	0.4788	0.5309
	CF-NeRF	**0.1939**	**0.2316**	**0.3983**	**0.3627**	**0.3983**	**0.3859**	**0.4686**	**0.1679**	**0.4453**	**0.2594**	**0.2831**	**0.3011**

Table 1: We conduct experiments on the NeRFBuster (Warburg et al. 2023), which is captured in the real world with complex trajectories. CF-NeRF achieves state-of-the-art results compared to NeRFmm (Wang et al. 2021b), SiRENmm (Guo and Sherwood 2021), BARF (Lin et al. 2021), GARF (Chng et al. 2022), L2G-NeRF (Chen et al. 2023).

of 0.9954 every 200 epochs. Similarly, the learning rate of δ is set to 0.001 and undergoes a decay of 0.9000 every 2000 epochs. Here, we describe how to set the hyper-parameters in CF-NeRF. We set N_{init} and N_{part} to 3 to meet the minimum requirements that can filter outliers based on MVG. To balance drift and efficiency, we set N_{glob} to 5. Considering the input image resolution, we set d_G to 3 to reconstruct all parameters by coarse-to-fine strategy. The most important parameter in CF-NeRF is iteration, which is the epoch number for each image. During initialization, we set ξ_{init} to 3000 to guarantee that θ and δ can be correctly initialized with fewer images. Subsequently, during the incremental training, we maintain a consistent value of ξ, setting $\xi = \xi_{loc} = \xi_{part} = \xi_{glob} = \xi_G$ to 900, thus reconstructing the scene from images one by one. Throughout all our experiments, we use the NVIDIA RTX3090.

Evaluation

To demonstrate the performance of the proposed method, we conduct a comprehensive comparison between CF-NeRF and several state-of-the-art models, including NeRFmm (Wang et al. 2021b) SiRENmm (Guo and Sherwood 2021), BARF (Lin et al. 2021), GARF (Chng et al. 2022), and L2G-NeRF (Chen et al. 2023). We use all images for camera parameter estimation without employing a train/test split. To evaluate the quality of the camera parameters, we calculate the average translation error ΔT and the average rotation error ΔR by aligning the estimated camera parameters δ_R and δ_T with COLMAP using a similarity transformation Sim(3) (Lin et al. 2021). It is worth noting that δ_T represents a relative translation error rather than an absolute measurement, as COLMAP can not reconstruct an absolute scale of the scene. We further evaluate the estimated camera parameters through a novel view synthesis by PSNR and LPIPS. To ensure a fair comparison and avoid the influence of varying network backbones across different methods, we uniformly use the NerfAcc (Li, Tancik, and Kanazawa 2022), where we select one image for testing in every eight images and the remaining is for training.

Results

We performed qualitative and quantitative evaluations of these methods on 12 scenes of the NeRFBuster (Warburg et al. 2023) dataset. Notably, BARF (Lin et al. 2021), GARF (Chng et al. 2022), and L2G-NeRF (Chen et al. 2023) require manual setting the focal length. In contrast, NeRFmm (Wang et al. 2021b), SiRENmm (Guo and Sherwood 2021), and CF-NeRF have the ability to estimate the focal length.

Table 1 shows the results of qualitative experiments. Our method obtains the highest accuracy camera parameters, while all other methods fail outright. It is important to understand that ΔR and ΔT are calculated by aligning the camera positions with Sim(3) and that a slight difference in camera position can lead to huge errors. The rotation error ΔR of our method CF-NeRF is roughly around $10°$, while the other methods are around $100°$. Moreover, the translation error δ_T

	G, ξ, N_{glob}	aloe	art	car	century	flowers	garbage	picnic	pikachu	pipe	plant	roses	table
$\Delta R\downarrow$	$F, 600, 10$	17.8029	24.3389	17.1692	11.5924	11.6163	9.1240	14.6452	13.0037	19.0749	5.3354	5.9091	6.4731
	$F, 900, 10$	14.8730	22.8142	17.8879	11.1201	10.4707	**8.6973**	**11.4209**	12.0625	18.4305	4.7303	5.8538	6.8481
	$C, 900, 5$	12.4862	**19.1647**	17.4755	9.7177	8.4555	9.6460	12.3162	10.9802	19.9855	5.1579	5.5133	5.2821
	$F, 900, 5$	**12.1226**	19.2496	17.5570	**9.6811**	**8.2556**	9.7658	12.6501	**11.3067**	19.9926	**4.8968**	**5.1229**	**4.5837**
$\Delta T\downarrow$	$F, 600, 10$	4.5457	5.9307	7.5697	2.9652	3.6234	4.5340	2.6677	5.3105	9.3544	4.4384	1.3324	1.9535
	$F, 900, 10$	3.9111	6.1190	7.3752	3.6834	3.3956	4.3080	3.4918	**3.2682**	**8.4666**	3.6109	1.3013	2.3973
	$C, 900, 5$	3.4681	2.2770	6.6250	2.8224	2.7405	4.1085	1.2886	4.2462	9.6998	3.5309	1.2182	1.2232
	$F, 900, 5$	**3.3788**	**2.2821**	**6.5452**	**2.7383**	**2.7026**	**4.0535**	**1.2833**	4.0586	9.4491	**3.4346**	**1.1945**	**1.2127**

Table 2: Ablation experiments. We compare the accuracy of camera parameters of CF-NeRF under different hyper-parameter settings, including the iteration ξ, the global optimization frequency N_{glob} and the coarse-to-fine strategy, where C means the coarse stage and F means the fine stage.

of CF-NeRF is approximately about 4, while all other methods are around 15. Although NeRFmm, SiRENmm, BARF, GARF, and L2G-NeRF claim high accuracy on forward-looking scenes from scratch, they are unsuitable for scenes with rotation and are prone to be trapped in a local minimum. In contrast, CF-NeRF recovers the camera parameters sequentially and can effectively handle image sequences with complex trajectories. Furthermore, SiRENmm outperforms NeRFmm in camera parameter estimation, which is why CF-NeRF uses the sine activate function.

Table 1 also shows the quality of the novel view synthesis, which serves as an additional evaluation criterion for the quality of camera parameters. CF-NeRF achieves state-of-the-art results on PSNR and LPIPS. Interestingly, the reconstruction results of other methods appear reasonable compared to their poor camera parameters, mainly due to the high over-fitting ability of NeRF and partial camera parameters are correctly reconstructed. We further visualize the rendering results of three scenes from different methods in Figure 1 and Figure 4. CF-NeRF can generate high-quality results, while other methods have lots of noise in their results due to their inability to provide accurate camera parameters.

Ablation Experiments

We conduct several ablation experiments on the iteration ξ, the global optimization frequency N_{glob}, and the coarse-to-fine strategy to validate the influence of hyper-parameters in CF-NeRF, and results are presented in Table 2.

The iteration ξ The iteration ξ is the most important hyper-parameter in our method, determining how many times to optimize the camera parameter for each image. We compare two configurations: $F, \xi = 600, N_{glob} = 10$ and $F, \xi = 900, N_{glob} = 10$. Table 2 reveals that increasing ξ improves the final results for almost all scenes. This observation aligns with NeRF (Mildenhall et al. 2020) and iNeRF (Yen-Chen et al. 2021), where NeRF requires a large number of iterations to converge, and iNeRF enhances the quality of camera parameters through more iterations.

The global optimization frequency N_{glob} To mitigate drift while maintaining efficiency, CF-NeRF employs the implicit global optimization when every N_{glob} image is added E. We conduct two experiments $F, \xi = 900, N_{glob} = 10$ and $F, \xi = 900, N_{glob} = 5$ to find out the influence of N_{glob}. As highlighted in Table 2, reducing N_{glob} yields improved final results, which can be attributed to the fact that global optimization ensures global consistency to avoid NeRF trap into a local minimum.

The coarse-to-fine strategy CF-NeRF adopts a coarse-to-fine strategy to avoid directly estimating camera parameters on high-resolution images, where the fine stage refines initial results from the coarse stage. We conduct two experiments $C, \xi = 900, N_{glob} = 5$ and $F, \xi = 900, N_{glob} = 5$. Results in Table 2 demonstrate that the fine stage outperforms the coarse stage across almost all scenes. The coarse-to-fine strategy facilitates the training process of CF-NeRF, as the pixel gradient is smoother at the coarse stage and has less RGB information to learn.

Limitation

Although CF-NeRF achieves state-of-the-art results in camera parameter estimation, surpassing other NeRF-based methods, there are still some gaps between CF-NeRF and COLMAP (Schonberger and Frahm 2016), and the accuracy can be further improved through the adjustment of the sample space (Wang et al. 2023) or the utilization of a more robust function (Sabour et al. 2023).

Conclusion

This paper presents CF-NeRF, a novel end-to-end method that does not require prior camera parameters to deal with image sequences with complex trajectories. Following the pipeline of incremental SfM, CF-NeRF contains three major sub-modules: initialization, implicit localization, and implicit optimization. Experiments on the NeRFBuster dataset demonstrate that CF-NeRF achieves state-of-the-art results, while NeRFmm, SiRENmm, BARF, GARF, and L2G-NeRF only work for forward-looking scenes and get trapped in the local minimum on the NeRFBuster dataset. More importantly, CF-NeRF highlights the unlimited potential of NeRF and differential volume rendering, showing that NeRF has impressive reconstruction capabilities and can also be used to estimate camera parameters in complex trajectories.

Acknowledgments

The research was supported in part by a RGC RIF grant under the contract R6021-20, RGC CRF grants under the contracts C7004-22G and C1029-22G, and RGC GRF grants under the contracts 16209120, 16200221, and 16207922. This research was also supported by the National Natural Science Foundation of China (No. 62302126), and the Shenzhen Science and Technology Program (No. RCBS20221008093125065, No. JCYJ20220818102414030).

References

Agarwal, S.; Furukawa, Y.; Snavely, N.; Simon, I.; Curless, B.; Seitz, S. M.; and Szeliski, R. 2011. Building rome in a day. *Communications of the ACM*, 54: 105–112.

Attal, B.; Laidlaw, E.; Gokaslan, A.; Kim, C.; Richardt, C.; Tompkin, J.; and O'Toole, M. 2021. Törf: Time-of-flight radiance fields for dynamic scene view synthesis. *NeurIPS*.

Barron, J. T.; Mildenhall, B.; Tancik, M.; Hedman, P.; Martin-Brualla, R.; and Srinivasan, P. P. 2021. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *ICCV*, 5855–5864.

Barron, J. T.; Mildenhall, B.; Verbin, D.; Srinivasan, P. P.; and Hedman, P. 2022. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *CVPR*, 5470–5479.

Bian, W.; Wang, Z.; Li, K.; Bian, J.-W.; and Prisacariu, V. A. 2023. Nope-nerf: Optimising neural radiance field with no pose prior. In *CVPR*, 4160–4169.

Boss, M.; Braun, R.; Jampani, V.; Barron, J. T.; Liu, C.; and Lensch, H. 2021a. Nerd: Neural reflectance decomposition from image collections. In *ICCV*, 12684–12694.

Boss, M.; Engelhardt, A.; Kar, A.; Li, Y.; Sun, D.; Barron, J.; Lensch, H.; and Jampani, V. 2022. Samurai: Shape and material from unconstrained real-world arbitrary image collections. *NeurIPS*, 26389–26403.

Boss, M.; Jampani, V.; Braun, R.; Liu, C.; Barron, J.; and Lensch, H. 2021b. Neural-pil: Neural pre-integrated lighting for reflectance decomposition. *NeurIPS*, 10691–10704.

Chen, A.; Xu, Z.; Geiger, A.; Yu, J.; and Su, H. 2022. Tensorf: Tensorial radiance fields. In *ECCV*, 333–350.

Chen, Y.; Chen, X.; Wang, X.; Zhang, Q.; Guo, Y.; Shan, Y.; and Wang, F. 2023. Local-to-global registration for bundle-adjusting neural radiance fields. In *CVPR*, 8264–8273.

Chng, S.-F.; Ramasinghe, S.; Sherrah, J.; and Lucey, S. 2022. Gaussian activated neural radiance fields for high fidelity reconstruction and pose estimation. In *ECCV*.

Darmon, F.; Bascle, B.; Devaux, J.-C.; Monasse, P.; and Aubry, M. 2022. Improving neural implicit surfaces geometry with patch warping. In *CVPR*, 6260–6269.

Engel, J.; Koltun, V.; and Cremers, D. 2017. Direct sparse odometry. *TPAMI*, 40: 611–625.

Fridovich-Keil, S.; Yu, A.; Tancik, M.; Chen, Q.; Recht, B.; and Kanazawa, A. 2022. Plenoxels: Radiance fields without neural networks. In *CVPR*, 5501–5510.

Fu, Q.; Xu, Q.; Ong, Y. S.; and Tao, W. 2022. Geo-neus: Geometry-consistent neural implicit surfaces learning for multi-view reconstruction. *NeurIPS*, 3403–3416.

Garbin, S. J.; Kowalski, M.; Johnson, M.; Shotton, J.; and Valentin, J. 2021. Fastnerf: High-fidelity neural rendering at 200fps. In *ICCV*, 14346–14355.

Guo, J.; and Sherwood, A. 2021. imporved-nerfmm. github.com/ventusff/improved-nerfmm. Accessed: 2023-07-01.

Jain, A.; Tancik, M.; and Abbeel, P. 2021. Putting nerf on a diet: Semantically consistent few-shot view synthesis. In *ICCV*, 5885–5894.

Jeong, Y.; Ahn, S.; Choy, C.; Anandkumar, A.; Cho, M.; and Park, J. 2021. Self-calibrating neural radiance fields. In *ICCV*, 5846–5854.

Kim, M.; Seo, S.; and Han, B. 2022. Infonerf: Ray entropy minimization for few-shot neural volume rendering. In *CVPR*, 12912–12921.

Kuang, Z.; Olszewski, K.; Chai, M.; Huang, Z.; Achlioptas, P.; and Tulyakov, S. 2022. NeROIC: neural rendering of objects from online image collections. *ACM TOG*, 1–12.

Li, R.; Tancik, M.; and Kanazawa, A. 2022. NerfAcc: A General NeRF Acceleration Toolbox. *arXiv preprint arXiv:2210.04847*.

Lin, C.-H.; Ma, W.-C.; Torralba, A.; and Lucey, S. 2021. Barf: Bundle-adjusting neural radiance fields. In *ICCV*.

Lin, Y.; Müller, T.; Tremblay, J.; Wen, B.; Tyree, S.; Evans, A.; Vela, P. A.; and Birchfield, S. 2023. Parallel inversion of neural radiance fields for robust pose estimation. In *ICRA*.

Long, X.; Lin, C.; Liu, L.; Liu, Y.; Wang, P.; Theobalt, C.; Komura, T.; and Wang, W. 2023. Neuraludf: Learning unsigned distance fields for multi-view reconstruction of surfaces with arbitrary topologies. In *CVPR*, 20834–20843.

Lowe, D. G. 2004. Distinctive image features from scale-invariant keypoints. *IJCV*, 91–110.

Martin-Brualla, R.; Radwan, N.; Sajjadi, M. S.; Barron, J. T.; Dosovitskiy, A.; and Duckworth, D. 2021. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *CVPR*, 7210–7219.

Meng, Q.; Chen, A.; Luo, H.; Wu, M.; Su, H.; Xu, L.; He, X.; and Yu, J. 2021. Gnerf: Gan-based neural radiance field without posed camera. In *ICCV*, 6351–6361.

Meuleman, A.; Liu, Y.-L.; Gao, C.; Huang, J.-B.; Kim, C.; Kim, M. H.; and Kopf, J. 2023. Progressively optimized local radiance fields for robust view synthesis. In *CVPR*.

Mi, Z.; Di, C.; and Xu, D. 2022. Generalized binary search network for highly-efficient multi-view stereo. In *CVPR*.

Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; and Ng, R. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *ECCV*.

Moulon, P.; Monasse, P.; and Marlet, R. 2013. Global fusion of relative motions for robust, accurate and scalable structure from motion. In *ICCV*, 3248–3255.

Moulon, P.; Monasse, P.; Perrot, R.; and Marlet, R. 2016. OpenMVG: Open multiple view geometry. In *International Workshop on RRPR*, 60–74.

Müller, T.; Evans, A.; Schied, C.; and Keller, A. 2022. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics*, 41: 1–15.

Mur-Artal, R.; Montiel, J. M. M.; and Tardos, J. D. 2015. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 1147–1163.

Niemeyer, M.; Barron, J. T.; Mildenhall, B.; Sajjadi, M. S.; Geiger, A.; and Radwan, N. 2022. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *CVPR*, 5480–5490.

Oechsle, M.; Peng, S.; and Geiger, A. 2021. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *ICCV*, 5589–5599.

Pumarola, A.; Corona, E.; Pons-Moll, G.; and Moreno-Noguer, F. 2021. D-nerf: Neural radiance fields for dynamic scenes. In *CVPR*, 10318–10327.

Rosinol, A.; Leonard, J. J.; and Carlone, L. 2022. NeRF-SLAM: Real-Time Dense Monocular SLAM with Neural Radiance Fields. *arXiv preprint arXiv:2210.13641*.

Sabour, S.; Vora, S.; Duckworth, D.; Krasin, I.; Fleet, D. J.; and Tagliasacchi, A. 2023. RobustNeRF: Ignoring Distractors with Robust Losses. In *CVPR*, 20626–20636.

Schonberger, J. L.; and Frahm, J.-M. 2016. Structure-from-motion revisited. In *ICCV*, 4104–4113.

Sitzmann, V.; Martel, J.; Bergman, A.; Lindell, D.; and Wetzstein, G. 2020. Implicit neural representations with periodic activation functions. *NeurIPS*, 33: 7462–7473.

Sucar, E.; Liu, S.; Ortiz, J.; and Davison, A. J. 2021. iMAP: Implicit mapping and positioning in real-time. In *ICCV*.

Sun, C.; Sun, M.; and Chen, H.-T. 2022. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *CVPR*, 5459–5469.

Tancik, M.; Casser, V.; Yan, X.; Pradhan, S.; Mildenhall, B.; Srinivasan, P. P.; Barron, J. T.; and Kretzschmar, H. 2022. Block-nerf: Scalable large scene neural view synthesis. In *CVPR*, 8248–8258.

Tancik, M.; Weber, E.; Ng, E.; Li, R.; Yi, B.; Wang, T.; Kristoffersen, A.; Austin, J.; Salahi, K.; Ahuja, A.; et al. 2023. Nerfstudio: A modular framework for neural radiance field development. In *ACM SIGGRAPH*, 1–12.

Teed, Z.; and Deng, J. 2020. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 402–419.

Turki, H.; Ramanan, D.; and Satyanarayanan, M. 2022. Mega-nerf: Scalable construction of large-scale nerfs for virtual fly-throughs. In *CVPR*, 12922–12931.

Ueda, I.; Fukuhara, Y.; Kataoka, H.; Aizawa, H.; Shishido, H.; and Kitahara, I. 2022. Neural Density-Distance Fields. In *ECCV*, 53–68.

Verbin, D.; Hedman, P.; Mildenhall, B.; Zickler, T.; Barron, J. T.; and Srinivasan, P. P. 2022. Ref-nerf: Structured view-dependent appearance for neural radiance fields. In *CVPR*.

Wang, P.; Liu, L.; Liu, Y.; Theobalt, C.; Komura, T.; and Wang, W. 2021a. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*.

Wang, P.; Liu, Y.; Chen, Z.; Liu, L.; Liu, Z.; Komura, T.; Theobalt, C.; and Wang, W. 2023. F2-NeRF: Fast Neural Radiance Field Training with Free Camera Trajectories. In *CVPR*, 4150–4159.

Wang, Y.; Skorokhodov, I.; and Wonka, P. 2022. HF-NeuS: Improved Surface Reconstruction Using High-Frequency Details. In *NeurIPS*.

Wang, Z.; Wu, S.; Xie, W.; Chen, M.; and Prisacariu, V. A. 2021b. NeRF–: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*.

Warburg, F.; Weber, E.; Tancik, M.; Holynski, A.; and Kanazawa, A. 2023. Nerfbusters: Removing Ghostly Artifacts from Casually Captured NeRFs. In *ICCV*.

Wu, C. 2013. Towards linear-time incremental structure from motion. In *3DV*, 127–134. IEEE.

Xiao, Y.; Xue, N.; Wu, T.; and Xia, G.-S. 2023. Level-S^2fM: Structure From Motion on Neural Level Set of Implicit Surfaces. In *CVPR*, 17205–17214.

Yan, Q.; Wang, Q.; Zhao, K.; Li, B.; Chu, X.; and Deng, F. 2023. Rethinking Disparity: A Depth Range Free Multi-View Stereo Based on Disparity. In *AAAI*, 3091–3099.

Yao, Y.; Luo, Z.; Li, S.; Fang, T.; and Quan, L. 2018. Mvsnet: Depth inference for unstructured multi-view stereo. In *ECCV*, 767–783.

Yariv, L.; Gu, J.; Kasten, Y.; and Lipman, Y. 2021. Volume rendering of neural implicit surfaces. *NeurIPS*, 4805–4815.

Yen-Chen, L.; Florence, P.; Barron, J. T.; Rodriguez, A.; Isola, P.; and Lin, T.-Y. 2021. inerf: Inverting neural radiance fields for pose estimation. In *IROS*, 1323–1330.

Yi, K. M.; Trulls, E.; Lepetit, V.; and Fua, P. 2016. Lift: Learned invariant feature transform. In *ECCV*, 467–483.

Yu, A.; Li, R.; Tancik, M.; Li, H.; Ng, R.; and Kanazawa, A. 2021a. Plenoctrees for real-time rendering of neural radiance fields. In *CVPR*, 5752–5761.

Yu, A.; Ye, V.; Tancik, M.; and Kanazawa, A. 2021b. pixelnerf: Neural radiance fields from one or few images. In *CVPR*, 4578–4587.

Zhang, J.; Zhan, F.; Wu, R.; Yu, Y.; Zhang, W.; Song, B.; Zhang, X.; and Lu, S. 2022. Vmrf: View matching neural radiance fields. In *ACM MM*, 6579–6587.

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

Zhang, X.; Srinivasan, P. P.; Deng, B.; Debevec, P.; Freeman, W. T.; and Barron, J. T. 2021. Nerfactor: Neural factorization of shape and reflectance under an unknown illumination. *ACM Transactions on Graphics*, 1–18.

Zhu, Z.; Peng, S.; Larsson, V.; Cui, Z.; Oswald, M. R.; Geiger, A.; and Pollefeys, M. 2023. Nicer-slam: Neural implicit scene encoding for rgb slam. *arXiv preprint arXiv:2302.03594*.

Zhu, Z.; Peng, S.; Larsson, V.; Xu, W.; Bao, H.; Cui, Z.; Oswald, M. R.; and Pollefeys, M. 2022. Nice-slam: Neural implicit scalable encoding for slam. In *CVPR*, 12786–12796.