BARF 🤢: Bundle-Adjusting Neural Radiance Fields

Chen-Hsuan Lin[1] Wei-Chiu Ma[2] Antonio Torralba[2] Simon Lucey[1,3]
[1]Carnegie Mellon University [2]Massachusetts Institute of Technology [3]The University of Adelaide
https://chenhsuanlin.bitbucket.io/bundle-adjusting-NeRF

Abstract

*Neural Radiance Fields (NeRF) [31] have recently gained a surge of interest within the computer vision community for its power to synthesize photorealistic novel views of real-world scenes. One limitation of NeRF, however, is its requirement of accurate camera poses to learn the scene representations. In this paper, we propose Bundle-Adjusting Neural Radiance Fields (BARF) for training NeRF from imperfect (or even unknown) camera poses — the joint problem of learning neural 3D representations and registering camera frames. We establish a theoretical connection to classical image alignment and show that coarse-to-fine registration is also applicable to NeRF. Furthermore, we show that naïvely applying positional encoding in NeRF has a negative impact on registration with a synthesis-based objective. Experiments on synthetic and real-world data show that BARF can effectively optimize the neural scene representations and resolve large camera pose misalignment at the same time. This enables view synthesis and localization of video sequences from unknown camera poses, opening up new avenues for visual localization systems (*e.g.* *SLAM) and potential applications for dense 3D mapping and reconstruction.*

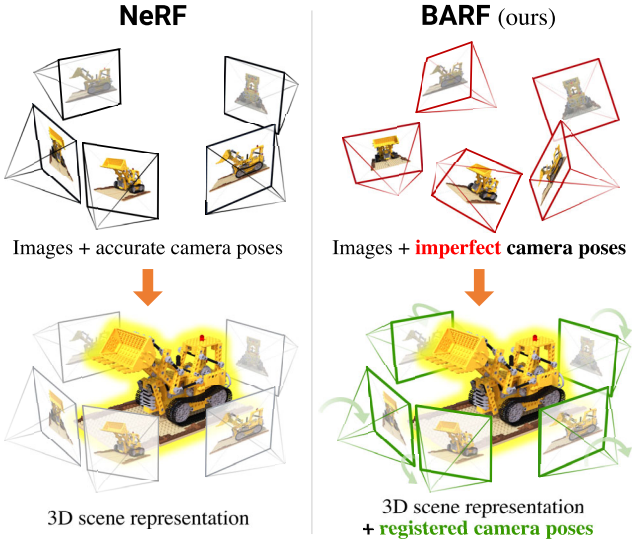


Figure 1: Training NeRF requires accurate camera poses for all images. We present **BARF** for learning 3D scene representations from *imperfect* (or even *unknown*) camera poses by jointly optimizing for registration and reconstruction.

1. Introduction

Humans have strong capabilities of reasoning about 3D geometry through our vision from the slightest ego-motion. When watching movies, we can immediately infer the 3D spatial structures of objects and scenes inside the videos. This is because we have an inherent ability of associating spatial correspondences of the same scene across continuous observations, without having to make sense of the relative camera or ego-motion. Through pure visual perception, not only can we recover a mental 3D representation of *what* we are looking at, but meanwhile we can also recognize *where* we are looking at the scene from.

Simultaneously solving for the 3D scene representation from RGB images (*i.e.* **reconstruction**) and localizing the given camera frames (*i.e.* **registration**) is a long-standing chicken-and-egg problem in computer vision — recovering the 3D structure requires observations with known camera poses, while localizing the cameras requires reliable correspondences from the reconstruction. Classical methods such as structure from motion (S*f*M) [17, 44] or SLAM [13, 32] approach this problem through local registration followed by global geometric bundle adjustment (BA) on both the structure and cameras. S*f*M and SLAM systems, however, are sensitive to the quality of local registration and easily fall into suboptimal solutions. In addition, the sparse nature of output 3D point clouds (often noisy) limits downstream vision tasks that requires dense geometric reasoning.

Closely related to 3D reconstruction from imagery is the problem of view synthesis. Though not primarily purposed for recovering explicit 3D structures, recent advances on photorealistic view synthesis have opted to recover an intermediate dense 3D-aware representation (*e.g.* depth [15, 61], multi-plane images [71, 51, 55], or volume density [27, 31]), followed by neural rendering techniques [14, 29, 47, 54] to

synthesize the target images. In particular, Neural Radiance Fields (NeRF) [31] have demonstrated its remarkable ability for high-fidelity view synthesis. NeRF encodes 3D scenes with a neural network mapping 3D point locations to color and volume density. This allows the scenes to be represented with compact memory footprint without limiting the resolution of synthesized images. The optimization process of the network is constrained to obey the principles of classical volume rendering [23], making the learned representation interpretable as a continuous 3D volume density function.

Despite its notable ability for photorealistic view synthesis and 3D scene representation, a hard prerequisite of NeRF (as well as other view synthesis methods) is accurate camera poses of the given images, which is typically obtained through auxiliary off-the-shelf algorithms. One straightforward way to circumvent this limitation is to additionally optimize the pose parameters with the NeRF model via backpropagation. As discussed later in the paper, however, naïve pose optimization with NeRF is sensitive to initialization. It may lead to suboptimal solutions of the 3D scene representation, degrading the quality of view synthesis.

In this paper, we address the problem of training NeRF representations from imperfect camera poses — the joint problem of *reconstructing* the 3D scene and *registering* the camera poses (Fig. 1). We draw inspiration from the success of classical image alignment methods and establish a theoretical connection, showing that coarse-to-fine registration is also critical to NeRF. Specifically, we show that positional encoding [57] of input 3D points plays a crucial role — as much as it enables fitting to high-frequency functions [53], positional encoding is also more susceptible to suboptimal registration results. To this end, we present Bundle-Adjusting NeRF (BARF), a simple yet effective strategy for coarse-to-fine registration on coordinate-based scene representations. BARF can be regarded as a type of *photometric* BA [8, 2, 26] using view synthesis as the proxy objective. Unlike traditional BA, however, BARF can learn scene representations *from scratch* (*i.e.* from randomly initialized network weights), lifting the reliance of local registration subprocedures and allowing for more generic applications.

In summary, we present the following contributions:

- We establish a theoretical connection between classical image alignment to joint registration and reconstruction with Neural Radiance Fields (NeRF).
- We show that susceptibility to noise from positional encoding affects the basin of attraction for registration, and we present a simple strategy for coarse-to-fine registration on coordinate-based scene representations.
- Our proposed BARF can successfully recover scene representations from imperfect camera poses, allowing for applications such as view synthesis and localization of video sequences from unknown poses.

2. Related Work

Structure from motion (S*f*M) and SLAM. Given a set of input images, S*f*M [37, 38, 48, 49, 1, 62] and SLAM [33, 13, 32, 64] systems aim to recover the 3D structure and the sensor poses simultaneously. These can be classified into (a) *indirect* methods that rely on keypoint detection and matching [6, 32] and (b) *direct* methods that exploit photometric consistency [2, 12]. Modern pipelines following the indirect route have achieved tremendous success [44]; however, they often suffer at textureless regions and repetitive patterns, where distinctive keypoints cannot be reliably detected. Researchers have thus sought to use neural networks to learn discriminative features directly from data [10, 35, 11].

Direct methods, on the other hand, do not rely on such distinctive keypoints — every pixel can contribute to maximizing photometric consistency, leading to improved robustness in sparsely textured environments [59]. They can also be naturally integrated into deep learning frameworks through image reconstruction losses [70, 58, 66]. Our method BARF lies under the broad umbrella of direct methods, as BARF learns 3D scene representations from RGB images while also localizing the respective cameras. However, unlike classical S*f*M and SLAM that represent 3D structures with explicit geometry (*e.g.* point clouds), BARF encodes the scenes as coordinate-based representations with neural networks.

View synthesis. Given a set of posed images, view synthesis attempts to simulate how a scene would look like from novel viewpoints [5, 24, 52, 19]. The task has been closely tied to 3D reconstruction since its introduction [7, 72, 18]. Researchers have investigated blending pixel colors based on depth maps [4] or leveraging proxy geometry to warp and composite the synthesized image [22]. However, since the problem is inherently ill-posed, there are still multiple restrictions and assumptions on the synthesized viewpoints.

State-of-the-art methods have capitalized on neural networks to learn both the scene geometry and statistical priors from data. Various representations have been explored in this direction, *e.g.* depth [15, 61, 42, 43], layered depth [56, 46], multi-plane images [71, 51, 55], volume density [27, 31], and mesh sheets [20]. Unfortunately, these view synthesis methods still require the camera poses to be known *a priori*, largely limiting their applications in practice. In contrast, our method BARF is able to effectively learn 3D representations that encodes the underlying scene geometry from imperfect or even unknown camera poses.

Neural Radiance Fields (NeRF). Recently, Mildenhall *et al.* [31] proposed NeRF to synthesize novel views of static, complex scenes from a set of posed input images. The key idea is to model the continuous radiance field of a scene with a multi-layer perceptron (MLP), followed by differentiable volume rendering to synthesize the images and backpropagate the photometric errors. NeRF has drawn wide attention

across the vision community [68, 34, 40, 36, 65] due to its simplicity and extraordinary performance. It has also been extended on many fronts, *e.g.* reflectance modeling for photorealistic relighting [3, 50] and dynamic scene modeling that integrates the motion of the world [25, 63, 39]. Recent works have also sought to exploit a large corpus of data to pretrain the MLP, enabling the ability to infer the radiance field from a single image [16, 67, 41, 45].

While impressive results have been achieved by the above NeRF-based models, they have a common drawback — the requirement of *posed* images. Our proposed BARF allows us to circumvent such requirement. We show that with a simple coarse-to-fine bundle adjustment technique, we can recover from imperfect camera poses (including *unknown* poses of video sequences) and learn the NeRF representation simultaneously. Concurrent to our work, NeRF-- [60] introduced an empirical, two-stage pipeline to estimate unknown camera poses. Our method BARF, in contrast, is motivated by mathematical insights and can recover the camera poses within a single course of optimization, allowing for direct utilities for various NeRF applications and extensions.

3. Approach

We unfold this paper by motivating with the simpler 2D case of classical image alignment as an example. Then we discuss how the same concept is also applicable to the 3D case, giving inspiration to our proposed BARF.

3.1. Planar Image Alignment (2D)

Let $\mathbf{x}\in\mathbb{R}^2$ be the 2D pixel coordinates and $\mathcal{I}:\mathbb{R}^2\rightarrow\mathbb{R}^3$ be the imaging function. Image alignment aims to find the relative geometric transformation which minimizes the photometric error between two images $\mathcal{I}_1$ and $\mathcal{I}_2$. The problem can be formulated with a synthesis-based objective:

$$\min_{\mathbf{p}}\sum_{\mathbf{x}}\|\mathcal{I}_1(\mathcal{W}(\mathbf{x};\mathbf{p}))-\mathcal{I}_2(\mathbf{x})\|_2^2\ , \tag{1}$$

where $\mathcal{W}:\mathbb{R}^2\rightarrow\mathbb{R}^2$ is the warp function parametrized by $\mathbf{p}\in\mathbb{R}^P$ (with P as the dimensionality). As this is a nonlinear problem, gradient-based optimization is the method of choice: given the current warp state $\mathbf{p}$, warp updates $\Delta\mathbf{p}$ are iteratively solved for and updated to the solution via $\mathbf{p}\leftarrow\mathbf{p}+\Delta\mathbf{p}$. Here, $\Delta\mathbf{p}$ can be written in a generic form of

$$\Delta\mathbf{p}=-\mathbf{A}(\mathbf{x};\mathbf{p})\sum_{\mathbf{x}}\mathbf{J}(\mathbf{x};\mathbf{p})^\top\big(\mathcal{I}_1(\mathcal{W}(\mathbf{x};\mathbf{p}))-\mathcal{I}_2(\mathbf{x})\big)\ , \tag{2}$$

where $\mathbf{J}\in\mathbb{R}^{3\times P}$ is termed the steepest descent image, and $\mathbf{A}$ is a generic transformation which depends on the choice of the optimization algorithm. The seminal Lucas-Kanade algorithm [28] approaches the problem using Gauss-Newton optimization, *i.e.* $\mathbf{A}(\mathbf{x};\mathbf{p})=(\sum_{\mathbf{x}}\mathbf{J}(\mathbf{x};\mathbf{p})^\top\mathbf{J}(\mathbf{x};\mathbf{p}))^{-1}$; alternatively, one could also choose first-order optimizers such as (stochastic) gradient descent which can be more naturally incorporated into modern deep learning frameworks, where $\mathbf{A}$ would correspond to a scalar learning rate.

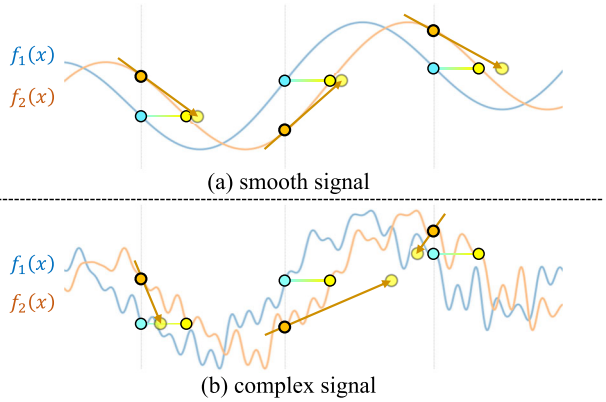


Figure 2: **Predicting alignment from signal differences.** Consider two 1D signals where $f_1(x)=f_2(x+c)$ differs by an offset c. When solving for alignment, smoother signals can predict more coherent displacements than complex signals, which easily results in suboptimal alignment.

The steepest descent image $\mathbf{J}$ can be expanded as

$$\mathbf{J}(\mathbf{x};\mathbf{p})=\frac{\partial\mathcal{I}_1(\mathcal{W}(\mathbf{x};\mathbf{p}))}{\partial\mathcal{W}(\mathbf{x};\mathbf{p})}\frac{\partial\mathcal{W}(\mathbf{x};\mathbf{p})}{\partial\mathbf{p}}\ , \tag{3}$$

where $\frac{\partial\mathcal{W}(\mathbf{x};\mathbf{p})}{\partial\mathbf{p}}\in\mathbb{R}^{2\times P}$ is the warp Jacobian constraining the pixel displacements with respect to the predefined warp. At the heart of gradient-based registration are the image gradients $\frac{\partial\mathcal{I}(\mathbf{x})}{\partial\mathbf{x}}\in\mathbb{R}^{3\times 2}$ modeling a local per-pixel linear relationship between appearance and spatial displacements, which is classically estimated via finite differencing. The overall warp update $\Delta\mathbf{p}$ can be more effectively estimated from pixel value differences if the per-pixel predictions are coherent (Fig. 2), *i.e.* the image signals are smooth. However, as natural images are typically complex signals, gradient-based registration on raw images is susceptible to suboptimal solutions if poorly initialized. Therefore, coarse-to-fine strategies have been practiced by blurring the images at earlier stages of registration, effectively widening the basin of attraction and smoothening the alignment landscape.

Images as neural networks. An alternative formulation of the problem is to learn a coordinate-based image representation with a neural network while also solving for the warp $\mathbf{p}$. Writing the network as $f:\mathbb{R}^2\rightarrow\mathbb{R}^3$ and denoting $\mathbf{\Theta}$ as its parameters, one can instead choose to optimize the objective

$$\min_{\mathbf{p},\mathbf{\Theta}}\sum_{\mathbf{x}}\big(\|f(\mathbf{x};\mathbf{\Theta})-\mathcal{I}_1(\mathbf{x})\|_2^2 + \|f(\mathcal{W}(\mathbf{x};\mathbf{p});\mathbf{\Theta})-\mathcal{I}_2(\mathbf{x})\|_2^2\big)\ , \tag{4}$$

or alternatively, one may choose to solve for warp parameters

$\mathbf{p}_1$ and $\mathbf{p}_2$ respectively for both images $\mathcal{I}_1$ and $\mathcal{I}_2$ through

$$\min_{\mathbf{p}_1,\mathbf{p}_2,\boldsymbol{\Theta}} \sum_{i=1}^{M} \sum_{\mathbf{x}} \left\| f(\mathcal{W}(\mathbf{x};\mathbf{p}_i);\boldsymbol{\Theta}) - \mathcal{I}_i(\mathbf{x}) \right\|_2^2 , \quad (5)$$

where $M = 2$ is the number of images. Albeit similar to (1), the image gradients become the analytical Jacobian of the network $\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}$ instead of numerical estimation. By manipulating the network f, this also enables more principled control of the signal smoothness for alignment without having to rely on heuristic blurring on images, making these forms generalizable to 3D scene representations (Sec. 3.2).

3.2. Neural Radiance Fields (3D)

We discuss the 3D case of recovering the 3D scene representation from Neural Radiance Fields (NeRF) [31] *jointly* with the camera poses. To signify the analogy to Sec. 3.1, we deliberately overload the notations $\mathbf{x}$ as 3D points, $\mathcal{W}$ as camera pose transformations, and f as the network in NeRF.

NeRF encodes a 3D scene as a continuous 3D representation using an MLP $f : \mathbb{R}^3 \to \mathbb{R}^4$ to predict the RGB color $\mathbf{c} \in \mathbb{R}^3$ and volume density $\sigma \in \mathbb{R}$ for each input 3D point $\mathbf{x} \in \mathbb{R}^3$. This can be summarized as $\mathbf{y} = [\mathbf{c}; \sigma]^\top = f(\mathbf{x};\boldsymbol{\Theta})$, where $\boldsymbol{\Theta}$ is the network parameters[1]. NeRF assumes an emission-only model, *i.e.* the rendered color of a pixel is dependent only on the emitted radiance of 3D points along the viewing ray, without considering external lighting factors.

We first formulate the rendering operation of NeRF in the camera view space. Given pixel coordinates $\mathbf{u} \in \mathbb{R}^2$ and denoting its homogeneous coordinates as $\bar{\mathbf{u}} = [\mathbf{u}; 1]^\top \in \mathbb{R}^3$, we can express a 3D point $\mathbf{x}_i$ along the viewing ray at depth z_i as $\mathbf{x}_i = z_i \bar{\mathbf{u}}$. The RGB color $\hat{\mathcal{I}}$ at pixel location $\mathbf{u}$ is extracted by volume rendering via

$$\hat{\mathcal{I}}(\mathbf{u}) = \int_{z_\text{near}}^{z_\text{far}} T(\mathbf{u}, z) \sigma(z\bar{\mathbf{u}}) \mathbf{c}(z\bar{\mathbf{u}}) \mathrm{d}z , \quad (6)$$

where $T(\mathbf{u}, z) = \exp\big(-\int_{z_\text{near}}^{z} \sigma(z'\bar{\mathbf{u}}) \mathrm{d}z'\big)$, and z_near and z_far are bounds on the depth range of interest. We refer our readers to Levoy [23] and Mildenhall *et al.* [31] for a more detailed treatment on volume rendering. In practice, the above integral formulations are approximated numerically via quadrature on discrete N points at depth $\{z_1, \dots, z_N\}$ sampled along the ray. This involves N evaluations of the network f, whose output $\{\mathbf{y}_1, \dots, \mathbf{y}_N\}$ are further composited through volume rendering. We can summarize the ray compositing function as $g : \mathbb{R}^{4N} \to \mathbb{R}^3$ and rewrite $\hat{\mathcal{I}}(\mathbf{u})$ as $\hat{\mathcal{I}}(\mathbf{u}) = g\left(\mathbf{y}_1, \dots, \mathbf{y}_N\right)$. Note that g is differentiable but deterministic, *i.e.* there are no learnable parameters associated.

Under a 6-DoF camera pose parametrized by $\mathbf{p} \in \mathbb{R}^6$, a 3D point $\mathbf{x}$ in the camera view space can be transformed to the 3D world coordinates through a 3D rigid transformation $\mathcal{W} : \mathbb{R}^3 \to \mathbb{R}^3$. Therefore, the synthesized RGB value at pixel $\mathbf{u}$ becomes a function of the camera pose $\mathbf{p}$ as

$$\hat{\mathcal{I}}(\mathbf{u};\mathbf{p}) = g\Big(f(\mathcal{W}(z_1\bar{\mathbf{u}};\mathbf{p});\boldsymbol{\Theta}), \dots, f(\mathcal{W}(z_N\bar{\mathbf{u}};\mathbf{p});\boldsymbol{\Theta})\Big). \quad (7)$$

Given M images $\{\mathcal{I}_i\}_{i=1}^M$, our goal is to optimize NeRF *and* the camera poses $\{\mathbf{p}_i\}_{i=1}^M$ over the synthesis-based objective

$$\min_{\mathbf{p}_1,\dots,\mathbf{p}_M,\boldsymbol{\Theta}} \sum_{i=1}^{M} \sum_{\mathbf{u}} \left\| \hat{\mathcal{I}}(\mathbf{u};\mathbf{p}_i,\boldsymbol{\Theta}) - \mathcal{I}_i(\mathbf{u}) \right\|_2^2 , \quad (8)$$

where $\hat{\mathcal{I}}$ also depends on the network parameters $\boldsymbol{\Theta}$.

One may notice the analogy between the synthesis-based objectives of 2D image alignment (5) and NeRF (8). Similarly, we can also derive the "steepest descent image" as

$$\mathbf{J}(\mathbf{u};\mathbf{p}) = \sum_{i=1}^{N} \frac{\partial g(\mathbf{y}_1, \dots, \mathbf{y}_N)}{\partial \mathbf{y}_i} \frac{\partial \mathbf{y}_i(\mathbf{p})}{\partial \mathbf{x}_i(\mathbf{p})} \frac{\partial \mathcal{W}(z_i\bar{\mathbf{u}};\mathbf{p})}{\partial \mathbf{p}} , \quad (9)$$

which is formed via backpropagation in practice. The linearization (9) is also analogous to the 2D case of (3), where the Jacobian of the network $\frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}$ linearly relates the change of color $\mathbf{c}$ and volume density σ with 3D spatial displacements. To solve for effective camera pose updates $\Delta\mathbf{p}$ through backpropagation, it is also desirable to control the smoothness of f for predicting coherent geometric displacements from the sampled 3D points $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$.

3.3. On Positional Encoding and Registration

The key of enabling NeRF to synthesize views with high fidelity is positional encoding [57], a deterministic mapping of input 3D coordinates $\mathbf{x}$ to higher dimensions of different sinusoidal frequency bases[2]. We denote $\gamma : \mathbb{R}^3 \to \mathbb{R}^{3+6L}$ as the positional encoding with L frequency bases, defined as

$$\gamma(\mathbf{x}) = \big[\mathbf{x}, \gamma_0(\mathbf{x}), \gamma_1(\mathbf{x}), \dots, \gamma_{L-1}(\mathbf{x})\big] \in \mathbb{R}^{3+6L} , \quad (10)$$

where the k-th frequency encoding $\gamma_k(\mathbf{x})$ is

$$\gamma_k(\mathbf{x}) = \big[\cos(2^k\pi\mathbf{x}), \sin(2^k\pi\mathbf{x})\big] \in \mathbb{R}^6 , \quad (11)$$

with the sinusoidal functions operating coordinate-wise. The special case of $L = 0$ makes γ an identity mapping function. The network f is thus a composition of $f(\mathbf{x}) = f' \circ \gamma(\mathbf{x})$, where f' is the subsequent learnable MLP. Positional encoding allows coordinate-based neural networks, which are typically bandwidth limited, to represent signals of higher frequency with faster convergence behaviors [53].

The Jacobian of the k-th positional encoding γ_k is

$$\frac{\partial \gamma_k(\mathbf{x})}{\partial \mathbf{x}} = 2^k\pi \cdot \big[-\sin(2^k\pi\mathbf{x}), \cos(2^k\pi\mathbf{x})\big] , \quad (12)$$

[1] In practice, f is also conditioned on the viewing direction [31] for modeling view-dependent effects, which we omit here for simplicity.

[2] Although we focus on 3D input coordinates here, positional encoding is also directly applicable to 2D image coordinates in Sec. 3.1 as well.

which amplifies the gradient signals from the MLP f' by $2^k\pi$ with its direction changing at the same frequency. This makes it difficult to predict effective updates $\Delta\mathbf{p}$, since gradient signals from the sampled 3D points are incoherent (in terms of both direction and magnitude) and can easily cancel out each other. Therefore, naïvely applying positional encoding can become a double-edged sword to NeRF for the task of joint registration and reconstruction.

3.4. Bundle-Adjusting Neural Radiance Fields

We describe our proposed BARF, a simple yet effective strategy for coarse-to-fine registration for NeRF. The key idea is to apply a smooth mask on the encoding at different frequency bands (from low to high) over the course of optimization, which acts like a dynamic low-pass filter. Inspired by recent work of learning coarse-to-fine deformation flow fields [36], we weigh the k-th frequency component of γ as

$$\gamma_k(\mathbf{x};\alpha) = w_k(\alpha)\cdot\big[\cos(2^k\pi\mathbf{x}),\sin(2^k\pi\mathbf{x})\big]\ , \tag{13}$$

where the weight w_k is defined as

$$w_k(\alpha) = \begin{cases} 0 & \text{if } \alpha < k \\ \dfrac{1-\cos((\alpha-k)\pi)}{2} & \text{if } 0\leq\alpha-k<1 \\ 1 & \text{if } \alpha-k\geq 1 \end{cases} \tag{14}$$

and $\alpha\in[0,L]$ is a controllable parameter proportional to the optimization progress. The Jacobian of γ_k thus becomes

$$\frac{\partial\gamma_k(\mathbf{x};\alpha)}{\partial\mathbf{x}} = w_k(\alpha)\cdot 2^k\pi\cdot\big[-\sin(2^k\pi\mathbf{x}),\cos(2^k\pi\mathbf{x})\big]\ . \tag{15}$$

When $w_k(\alpha)=0$, the contribution to the gradient from the k-th (and higher) frequency component is nullified.

Starting from the raw 3D input $\mathbf{x}$ ($\alpha=0$), we gradually activate the encodings of higher frequency bands until full positional encoding is enabled ($\alpha=L$), equivalent to the original NeRF model. This allows BARF to discover the correct registration with an initially smooth signal and later shift focus to learning a high-fidelity scene representation.

4. Experiments

We validate the effectiveness of our proposed BARF with a simple experiment of 2D planar image alignment, and show how the same coarse-to-fine registration strategy can be generalized to NeRF [31] for learning 3D scene representations.

4.1. Planar Image Alignment (2D)

We choose a representative image from ImageNet [9], shown in Fig. 3. Given $M=5$ patches from the image generated with homography perturbations (Fig. 3(a)), we aim to find the homography warp parameters $\mathbf{p}\in\mathbb{R}^8$ for each patch (Fig. 3(b)) while *also* learning the neural representation of the entire image with a network f by optimizing (5). We initialize all M patches with a center crop (Fig. 3(c)), and we anchor the warp of the first patch as identity so the recovered image would be implicitly aligned to the raw image. We parametrize homography warps with the $\mathfrak{sl}(3)$ Lie algebra.

Experimental settings. We investigate how positional encoding impacts this problem by comparing networks with naïve (full) positional encoding and without any encoding. We use a simple ReLU MLP for f with four 256-dimensional hidden units, and we use the Adam optimizer [21] to optimize both the network weights and the warp parameters for 5000 iterations with a learning rate of 0.001. For BARF, we linearly adjust α for the first 2000 iterations and activate all frequency bands ($L=8$) for the remaining iterations.

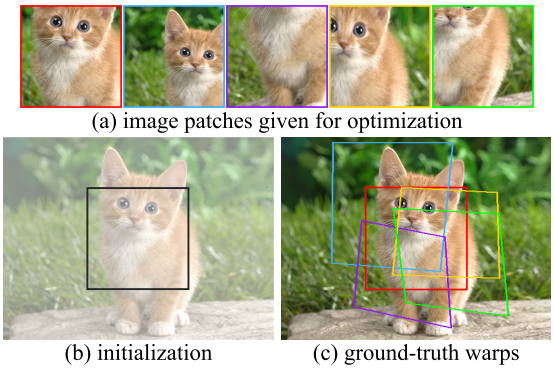
Results. We visualize the registration results in Fig. 4. Alignment with full positional encoding results in suboptimal registration with ghostly artifacts in the recovered image representation. On the other hand, alignment without positional encoding achieves decent registration results, but cannot recover the image with sufficient fidelity. BARF discovers the precise geometric warps with the image representation optimized with high fidelity, quantitatively reflected in Table 1. The image alignment experiment demonstrates the general advantage of BARF for coordinate-based representations.

4.2. NeRF (3D): Synthetic Objects

We investigate the problem of learning 3D scene representations with Neural Radiance Fields (NeRF) [31] from imperfect camera poses. We experiment with the 8 synthetic object-centric scenes provided by Mildenhall *et al.* [31], which consists of $M=100$ rendered images with ground-truth camera poses for each scene for training.

Experimental settings. We parametrize the camera poses $\mathbf{p}$ with the $\mathfrak{se}(3)$ Lie algebra and assume known intrinsics. For each scene, we synthetically perturb the camera poses with additive noise $\delta\mathbf{p}\sim\mathcal{N}(\mathbf{0},0.15\mathbf{I})$, which corresponds to a standard deviation of $14.9°$ in rotation and 0.26 in translational magnitude (Fig. 5(a)). We optimize the objective in (8) jointly for the scene representation and the camera poses. We evaluate BARF mainly against the original NeRF model with naïve (full) positional encoding; for completeness, we also compare with the same model without positional encoding.

Implementation details. We follow the architectural settings from the original NeRF [31] with some modifications. We train a single MLP with 128 hidden units in each layer and without additional hierarchical sampling for simplicity. We resize the images to 400×400 pixels and randomly sample 1024 pixel rays at each optimization step. We choose $N=128$ sample for numerical integration along each ray, and we use the softplus activation on the volume density output σ for improved stability. We use the Adam optimizer and train all models for 200K iterations, with a learning rate of 5×10^{-4} exponentially decaying to 1×10^{-4} for the network f and 1×10^{-3} decaying to 1×10^{-5} for the poses $\mathbf{p}$. For



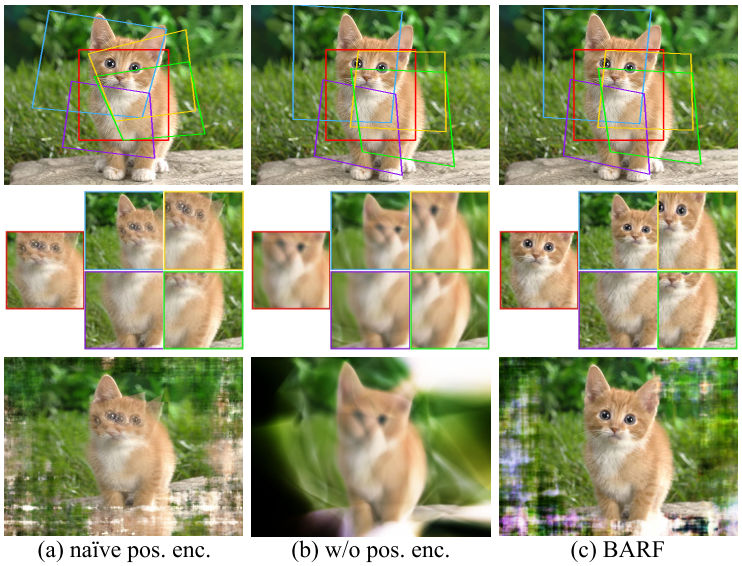
(a) image patches given for optimization

(b) initialization (c) ground-truth warps

Figure 3: Given image patches color-coded in (a), we aim to recover the alignment *and* the neural representation of the entire image, with the patches initialized to center crops shown in (b) and the corresponding ground-truth warps shown in (c).

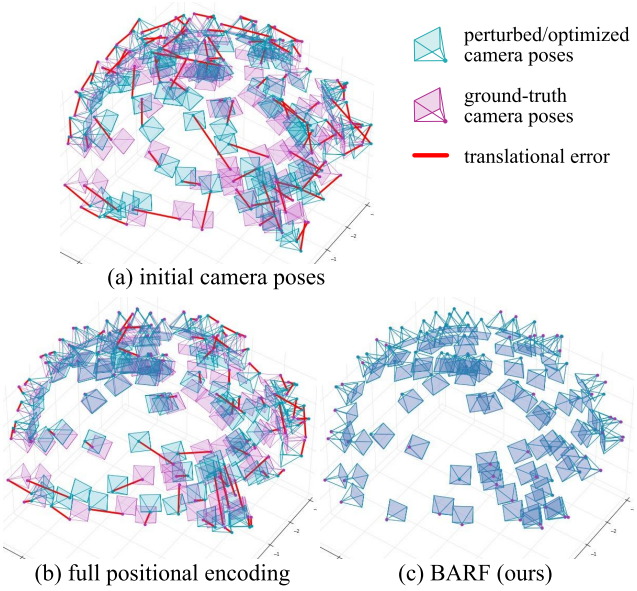
positional encoding	$\mathfrak{sl}(3)$ error	patch PSNR
naïve (full)	0.2949	23.41
without	0.0641	24.72
BARF (coarse-to-fine)	0.0096	35.30

Table 1: **Quantitative results** of planar image alignment. BARF optimizes for more accurate alignment and patch reconstruction compared to the baselines.



(a) naïve pos. enc. (b) w/o pos. enc. (c) BARF

Figure 4: **Qualitative results** of the planar image alignment experiment. We visualize the optimized warps (top row), the patch reconstructions in corresponding colors (middle row), and recovered image representation from f (bottom row). BARF is able to recover accurate alignment and high-fidelity image reconstruction, while baselines result in suboptimal alignment with naïve positional encoding and blurry reconstruction without any encoding. Best viewed in color.



(a) initial camera poses

(b) full positional encoding (c) BARF (ours)

Figure 5: Visual comparison of the initial and optimized camera poses (Procrustes aligned) for the *chair* scene. BARF successfully realigns all the camera frames while NeRF naïve positional encoding gets stuck at suboptimal solutions.

BARF, we linearly adjust α from iteration 20K to 100K and activate all frequency bands (up to $L = 10$) subsequently.

Evaluation criteria. We measure the performance in two aspects: pose error for registration and view synthesis quality for the scene representation. Since both the scene and camera poses are variable up to a 3D similarity transformation, we evaluate the quality of registration by pre-aligning the optimized poses to the ground truth with Procrustes analysis on the camera locations. For evaluating view synthesis, we run an additional step of test-time photometric optimization on the trained models [26, 65] to factor out the pose error that may contaminate the view synthesis quality. We report the average rotation and translation errors for pose and PSNR, SSIM and LPIPS [69] for view synthesis.

Results. We visualize the results in Fig. 6 and report the quantitative results in Table 2. BARF takes the best of both worlds of recovering the neural scene representation with the camera pose successfully registered, while naïve NeRF with full positional encoding finds suboptimal solutions. Fig. 5 shows that BARF can achieve near-perfect registration for the synthetic scenes. Although the NeRF model without positional encoding can also successfully recover alignment, the learned scene representations (and thus the synthesized images) lack the reconstruction fidelity. As a reference, we also compare the view synthesis quality against standard NeRF models trained under ground-truth poses, showing that BARF can achieve comparable view synthesis quality in all metrics, albeit initialized from imperfect camera poses.

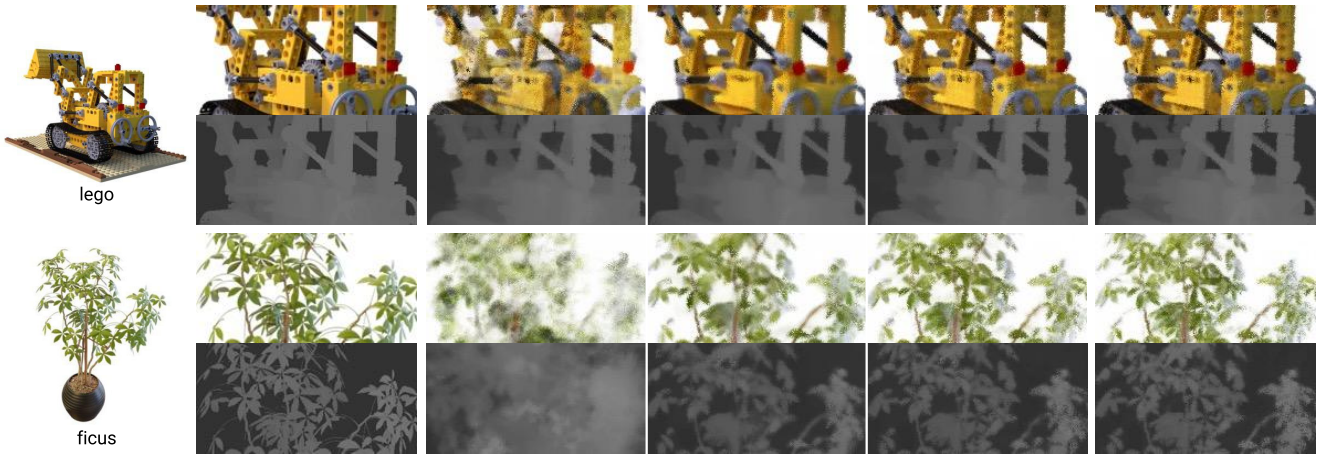


Figure 6: **Qualitative results** of NeRF on synthetic scenes. We visualize the image synthesis (top) and the expected depth through ray compositing (bottom). BARF achieves comparable synthesis quality to the reference NeRF (trained under perfect camera poses), while full positional encoding results in suboptimal registration, leading to synthesis artifacts.

Scene	Camera pose registration						View synthesis quality											
	Rotation (°) ↓			Translation ↓			PSNR ↑				SSIM ↑				LPIPS ↓			
	full pos.enc.	w/o pos.enc.	BARF	full pos.enc.	w/o pos.enc.	BARF	full pos.enc.	w/o pos.enc.	BARF	ref. NeRF	full pos.enc.	w/o pos.enc.	BARF	ref. NeRF	full pos.enc.	w/o pos.enc.	BARF	ref. NeRF
Chair	7.186	0.110	**0.096**	16.638	0.555	**0.428**	19.02	30.22	**31.16**	31.91	0.804	0.942	**0.954**	0.961	0.223	0.065	**0.044**	0.036
Drums	3.208	0.057	**0.043**	7.322	0.255	**0.225**	20.83	23.56	**23.91**	23.96	0.840	0.893	**0.900**	0.902	0.166	0.116	**0.099**	0.095
Ficus	9.368	0.095	**0.085**	10.135	**0.430**	0.474	19.75	25.58	**26.26**	26.68	0.836	0.922	**0.934**	0.941	0.182	0.070	**0.058**	0.051
Hotdog	3.290	**0.225**	0.248	6.344	**1.122**	1.308	28.15	34.00	**34.54**	34.91	0.923	0.967	**0.970**	0.973	0.083	0.040	**0.032**	0.029
Lego	3.252	0.108	**0.082**	4.841	0.391	**0.291**	24.23	26.35	**28.33**	29.28	0.876	0.880	**0.927**	0.942	0.102	0.112	**0.050**	0.037
Materials	6.971	0.845	**0.844**	15.188	**2.678**	2.692	16.51	26.86	**27.84**	28.48	0.747	0.926	**0.936**	0.944	0.294	0.068	**0.058**	0.049
Mic	10.554	0.081	**0.071**	22.724	0.356	**0.301**	15.10	30.93	**31.18**	31.98	0.788	0.968	**0.969**	0.971	0.334	0.050	**0.048**	0.044
Ship	5.506	0.095	**0.075**	7.232	0.354	**0.326**	22.12	26.78	**27.50**	28.00	0.755	0.833	**0.849**	0.858	0.255	0.175	**0.132**	0.118
Mean	6.167	0.202	**0.193**	11.303	0.768	**0.756**	22.12	26.78	**27.50**	29.40	0.821	0.917	**0.930**	0.936	0.205	0.087	**0.065**	0.057

Table 2: **Quantitative results** of NeRF on synthetic scenes. BARF successfully optimizes for camera registration (with less than $0.2°$ rotation error) while still consistently achieving high-quality view synthesis that is comparable to the reference NeRF models (trained under perfect camera poses). Translation errors are scaled by 100.

4.3. NeRF (3D): Real-World Scenes

We investigate the challenging problem of learning neural 3D representations with NeRF on real-world scenes, where the camera poses are *unknown*. We consider the LLFF dataset [30], which consists of 8 forward-facing scenes with RGB images sequentially captured by hand-held cameras.

Experimental settings. We parametrize the camera poses $\mathbf{p}$ with $\mathfrak{se}(3)$ following Sec. 4.2 but initialize all cameras with the *identity* transformation, *i.e.* $\mathbf{p}_i = \mathbf{0}\ \ \forall i$. We assume known camera intrinsics (provided by the dataset). We compare against the original NeRF model with naïve positional encoding, and we use the same evaluation criteria described in Sec. 4.2. However, we note that the camera poses provided in LLFF are also estimations from S*f*M packages [44]; therefore, the pose evaluation is at most an indication of how well BARF agrees with classical geometric pose estimation.

Implementation details. We follow the same architectural settings from the original NeRF [31] and resize the images to 480×640 pixels. We train all models for 200K iterations and randomly sample 2048 pixel rays at each optimization step, with a learning rate of 1×10^{-3} for the network f decaying to 1×10^{-4}, and 3×10^{-3} for the pose $\mathbf{p}$ decaying to 1×10^{-5}. We linearly adjust α for BARF from iteration 20K to 100K and activate all bands (up to $L=10$) subsequently.

Results. The quantitative results (Table 3) show that the recovered camera poses from BARF highly agrees with those estimated from off-the-shelf S*f*M methods (visualized in Fig. 8), demonstrating the ability of BARF to localize from scratch. Furthermore, BARF can successfully recover the 3D scene representation with high fidelity (Fig. 7). In contrast, NeRF with naïve positional encoding diverge to incorrect camera poses, which in turn results in poor view synthesis. This highlights the effectiveness of BARF utilizing a coarse-to-fine strategy for joint registration and reconstruction.

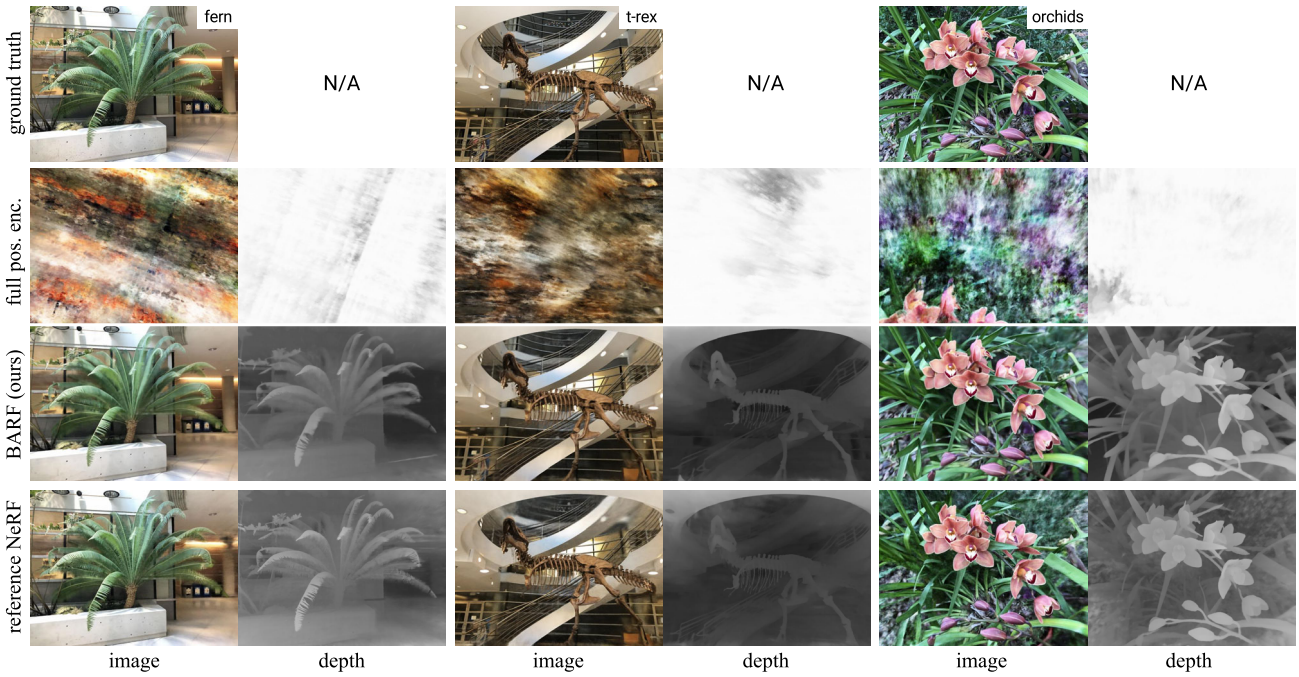
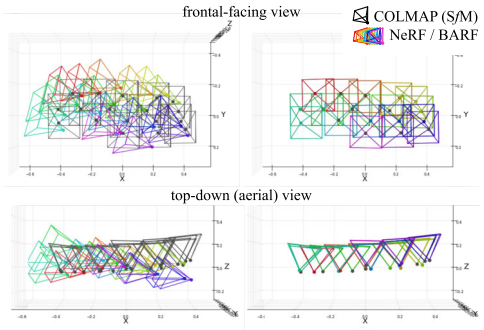


Figure 7: **Qualitative results** of NeRF on real-world scenes from *unknown* camera poses. Compared to a reference NeRF model trained with camera poses provided from S*f*M [44], BARF can effectively optimize for the poses jointly with the scene representation. NeRF models with full positional encoding diverge to incorrect localization and hence poor synthesis quality.

Scene	Camera pose registration				View synthesis quality								
	Rotation (°) ↓		Translation ↓		PSNR ↑			SSIM ↑			LPIPS ↓		
	full pos.enc.	BARF	full pos.enc.	BARF	full pos.enc.	BARF	ref. NeRF	full pos.enc.	BARF	ref. NeRF	full pos.enc.	BARF	ref. NeRF
Fern	74.452	**0.191**	30.167	**0.192**	9.81	**23.79**	23.72	0.187	**0.710**	0.733	0.853	**0.311**	0.262
Flower	2.525	**0.251**	2.635	**0.224**	17.08	**23.37**	23.24	0.344	**0.698**	0.668	0.490	**0.211**	0.244
Fortress	75.094	**0.479**	33.231	**0.364**	12.15	**29.08**	25.97	0.270	**0.823**	0.786	0.807	**0.132**	0.185
Horns	58.764	**0.304**	32.664	**0.222**	8.89	**22.78**	20.35	0.158	**0.727**	0.624	0.805	**0.298**	0.421
Leaves	88.091	**1.272**	13.540	**0.249**	9.64	**18.78**	15.33	0.067	**0.537**	0.306	0.782	**0.353**	0.526
Orchids	37.104	**0.627**	20.312	**0.404**	9.42	**19.45**	17.34	0.085	**0.574**	0.518	0.806	**0.291**	0.307
Room	173.811	**0.320**	66.922	**0.270**	10.78	**31.95**	32.42	0.278	**0.940**	0.948	0.871	**0.099**	0.080
T-rex	166.231	**1.138**	53.309	**0.720**	10.48	**22.55**	22.12	0.158	**0.767**	0.739	0.885	**0.206**	0.244
Mean	84.509	**0.573**	31.598	**0.331**	11.03	**23.97**	22.56	0.193	**0.722**	0.665	0.787	**0.238**	0.283

Table 3: **Quantitative results** of NeRF on the LLFF forward-facing scenes from *unknown* camera poses. BARF can optimize for accurate camera poses (with an average $< 0.6°$ rotation error) and high-fidelity scene representations, enabling novel view synthesis whose quality is comparable to reference NeRF model trained under S*f*M poses. Translation errors are scaled by 100.



(a) full pos. enc. (b) BARF (ours)

Figure 8: Visualization of optimized camera poses from the *fern* scene (Procrustes aligned). Results from BARF highly agrees with S*f*M, whereas the baseline poses are suboptimal.

5. Conclusion

We present Bundle-Adjusting Neural Radiance Fields (BARF), a simple yet effective strategy for training NeRF from imperfect camera poses. By establishing a theoretical connection to classical image alignment, we demonstrate that coarse-to-fine registration is necessary for joint registration and reconstruction with coordinate-based scene representations. Our experiments show that BARF can effectively learn the 3D scene representations from scratch and resolve large camera pose misalignment at the same time.

Despite the intriguing results at the current stage, BARF has similar limitations to the original NeRF formulation [31] (*e.g.* slow optimization and rendering, rigidity assumption, sensitivity to dense 3D sampling), as well as reliance on heuristic coarse-to-fine scheduling strategies. Nevertheless, since BARF keeps a close formulation to NeRF, many of the latest advances on improving NeRF are potentially transferable to BARF as well. We believe BARF opens up exciting avenues for rethinking visual localization for S*f*M/SLAM systems and self-supervised dense 3D reconstruction frameworks using view synthesis as a proxy objective.

Acknowledgements. We thank Chaoyang Wang, Mengtian Li, Yen-Chen Lin, Tongzhou Wang, Sivabalan Manivasagam, and Shenlong Wang for helpful discussions and feedback on the paper. This work was supported by the CMU Argo AI Center for Autonomous Vehicle Research.

A. Visualizing the Basin of Attraction

The planar image alignment setting allows us to analyze how positional encoding affects the basin of attraction. We use the same image in Fig. 3 and consider the simpler case of aligning two image patches differing by an offset. We use a translational warp $\mathbf{p}\in\mathbb{R}^2$ on a square box whose size is $1/3$ of the raw image height and initialized to the raw center. We aim to register the center box to a single target patch of the same size shifted by some offset, shown in Fig. 9(a). We optimize the image neural network f with the objective in (4), where $\mathcal{I}_1$ is the center patch and $\mathcal{I}_2$ is the target patch, and investigate the convergence behavior of translational alignment as a function of target offsets. We search over the entire pixel grid to as far as where the target patch has no overlapping region with the initial center box.

We visualize the results in Fig. 9. Naïve positional encoding results in a more nonlinear alignment landscape and a smaller basin of attraction, while not using positional encoding sacrifices the reconstruction quality due to the limited representability of the network f. In contrast, BARF can widen the basin of attraction while reconstructing the image representation with high fidelity. This also justifies the importance of coarse-to-fine registration for NeRF in the 3D case. Please also refer to the supplementary videos for more visualizations of the basin of attraction.

B. Additional NeRF Details & Results

We provide more details and results from our NeRF experiments in this section (for real-world scenes in particular).

B.1. Evaluation Details

As mentioned in the main paper, the optimized solutions of the 3D scenes and camera poses are up to a 3D similarity transformation. Therefore, we evaluate the quality of registration by pre-aligning the optimized poses to the reference poses, which are the ground truth poses for the synthetic objects (Sec. 4.2) and pose estimation computed from S*f*M packages [44] for the real-world scenes (Sec. 4.3).

We use Procrustes analysis on the camera locations for aligning the coordinate systems. The algorithm details are described in Alg. 1. We write the reference poses $\{[\mathbf{R}_i,\mathbf{t}_i]\}_{i=1}^M$ and the optimized poses $\{[\widehat{\mathbf{R}}_i,\hat{\mathbf{t}}_i]\}_{i=1}^M$ in the form of camera extrinsic matrices, and the aligned poses can be written as $\{[\widehat{\mathbf{R}}'_i,\hat{\mathbf{t}}'_i]\}_{i=1}^M=\textsc{PreAlign}(\{[\mathbf{R}_i,\mathbf{t}_i]\}_{i=1}^M,\{[\widehat{\mathbf{R}}_i,\hat{\mathbf{t}}_i]\}_{i=1}^M)$. After the cameras are Procrustes-aligned, we apply the relative rotation (solved for via the Procrustes analysis process) to account for rotational differences. We measure the rotation error between the S*f*M poses and the aligned poses from NeRF/BARF by the angular distance as

$$\Delta\theta_i=\cos^{-1}\frac{\text{trace}\big(\mathbf{R}_i\widehat{\mathbf{R}}_i^{\prime\top}\big)-1}{2}\ ,\quad i=\{1,\dots,M\}\ , \tag{16}$$

where $\langle\cdot,\cdot\rangle$ is the quaternion inner product. For additional clarity, we provide a more detailed visualization of the optimized camera poses in Fig. 10 (for the LLFF dataset).

Algorithm 1: Pre-align camera poses for evaluation

Function PreAlign($\{[\mathbf{R}_i,\mathbf{t}_i]\}_{i=1}^M,\{[\widehat{\mathbf{R}}_i,\hat{\mathbf{t}}_i]\}_{i=1}^M$)**:**
- **Input :** reference poses $\{[\mathbf{R}_i,\mathbf{t}_i]\}_{i=1}^M$, optimized poses $\{[\widehat{\mathbf{R}}_i,\hat{\mathbf{t}}_i]\}_{i=1}^M$
- **Output :** optimized poses $\{[\widehat{\mathbf{R}}'_i,\hat{\mathbf{t}}'_i]\}_{i=1}^M$ aligned to the reference poses

for $i=\{1,\dots,M\}$ **do**
- $\mathbf{o}_i=-\mathbf{R}_i^\top\mathbf{t}_i$
- $\hat{\mathbf{o}}_i=-\widehat{\mathbf{R}}_i^\top\hat{\mathbf{t}}_i$

end

$s,\hat{s},\mathbf{t},\hat{\mathbf{t}},\mathbf{R}=\textsc{Procrustes}(\{\mathbf{o}_i\}_{i=1}^M,\{\hat{\mathbf{o}}_i\}_{i=1}^M)$

for $i=\{1,\dots,M\}$ **do**
- $\hat{\mathbf{o}}'_i=s\mathbf{R}\left(\frac{1}{\hat{s}}(\hat{\mathbf{o}}_i-\hat{\mathbf{t}})\right)+\mathbf{t}$
- $\widehat{\mathbf{R}}'_i=\widehat{\mathbf{R}}_i\mathbf{R}^\top$
- $\hat{\mathbf{t}}'_i=-\widehat{\mathbf{R}}_i^{\prime\top}\hat{\mathbf{o}}'_i$

end

return $\{[\widehat{\mathbf{R}}'_i,\hat{\mathbf{t}}'_i]\}_{i=1}^M$

end

Function Procrustes($\{\mathbf{o}_i\}_{i=1}^M,\{\hat{\mathbf{o}}_i\}_{i=1}^M$)**:**
- **Input :** reference camera centers $\{\mathbf{o}_i\}_{i=1}^M$, optimized camera centers $\{\hat{\mathbf{o}}_i\}_{i=1}^M$
- **Output :** scale s, $\hat{s}$, translation $\mathbf{t}$, $\hat{\mathbf{t}}$, rotation $\mathbf{R}$

$\mathbf{t}=\frac{1}{M}\sum_{i=1}^M\mathbf{o}_i\ \in\mathbb{R}^3$

$\hat{\mathbf{t}}=\frac{1}{M}\sum_{i=1}^M\hat{\mathbf{o}}_i\ \in\mathbb{R}^3$

$s=\sqrt{\frac{1}{M}\sum_{i=1}^M\|\mathbf{o}_i-\mathbf{t}\|_2^2}\ \in\mathbb{R}$

$\hat{s}=\sqrt{\frac{1}{M}\sum_{i=1}^M\|\hat{\mathbf{o}}_i-\hat{\mathbf{t}}\|_2^2}\ \in\mathbb{R}$

$\mathbf{X}=\frac{1}{s}\left([\mathbf{o}_1,\dots,\mathbf{o}_M]-\mathbf{t}\mathbf{1}_M^\top\right)\ \in\mathbb{R}^{3\times M}$

$\widehat{\mathbf{X}}=\frac{1}{\hat{s}}\left([\hat{\mathbf{o}}_1,\dots,\hat{\mathbf{o}}_M]-\hat{\mathbf{t}}\mathbf{1}_M^\top\right)\ \in\mathbb{R}^{3\times M}$

$\mathbf{U},\mathbf{S},\mathbf{V}^\top=\text{SVD}(\mathbf{X}\widehat{\mathbf{X}}^\top)$

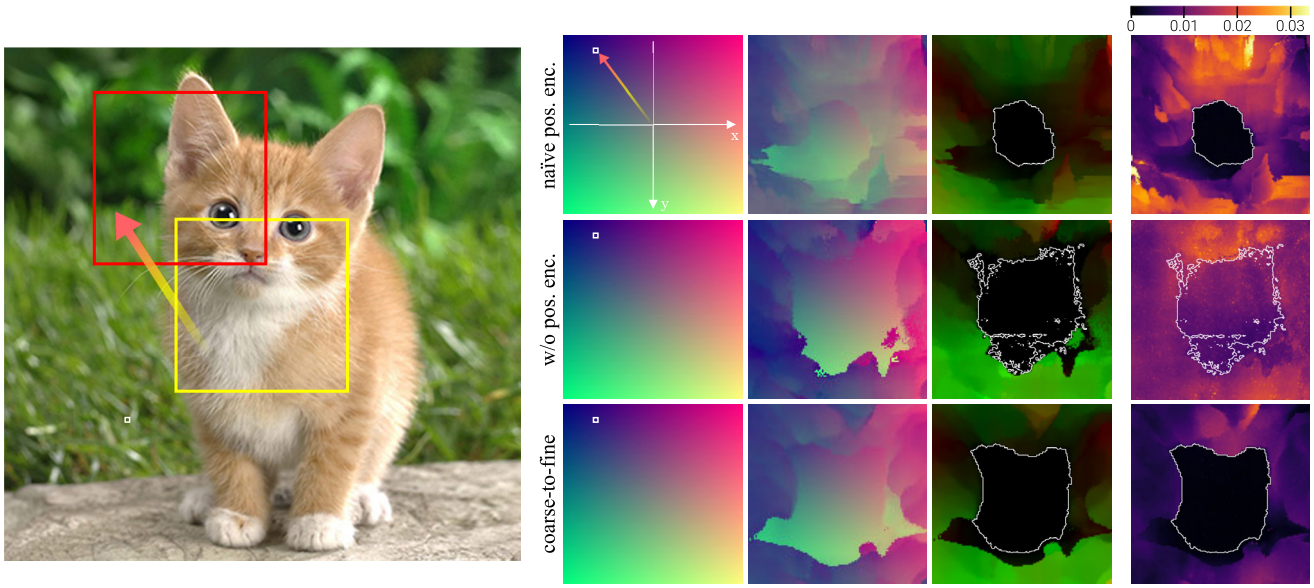
$\mathbf{R}=\mathbf{U}\mathbf{V}^\top\ \in\mathbb{R}^{3\times3}$

if $\det(\mathbf{R})=-1$ **then**
- multiply last row of $\mathbf{R}$ by -1

end

return s, $\hat{s}$, $\mathbf{t}$, $\hat{\mathbf{t}}$, $\mathbf{R}$

end



(a) source and target patch (b) target warp (c) optimized warp (d) warp error (e) image recon. error

Figure 9: Visualization of the **basin of attraction**. (a) We aim to align a center box (yellow) to a target patch (red) at *every* possible location within the raw image. For each target patch, we jointly optimize f and the translational warp $\mathbf{p}$ to analyze the final warp error and the image reconstruction loss. (b) The target offsets forms a color-coded map, where green indicates horizontal offsets and red indicates vertical offsets. The above example corresponds to the highlighted pixel. (c) The optimized warp parameters and (d) the warp error for every target patch location, where the white contours highlight the offset error threshold of 0.5 pixels. BARF effectively widens the basin of attraction (range of successful alignment) with a smoother landscape compared to naïve positional encoding. (e) Without positional encoding, f has limited capacity of representing the image details, resulting in nonzero image errors despite the registration being successful as well.

To evaluate the quality of novel view synthesis while being minimally affected by camera misalignment, we transform the test views (provided by Mildenhall *et al*. [30]) to the coordinate system of the optimized poses by applying the scale/rotation/translation from the Procrustes analysis, as in Alg. 1. The camera trajectories from the baseline NeRF with naïve full positional encoding exhibits large rotational and translational differences compared to S*f*M poses in general. For this reason, the view synthesis results from the baseline NeRF, whose corresponding test views are also determined using Procrustes analysis, are far from plausible. Unfortunately, there is no other systematic way of determining what the corresponding views held out from the S*f*M poses would be in the learned coordinate system. Nevertheless, we provide additional qualitative results in Fig. 11, where the novel views are selected from a *training* view closest to the average pose and sampling translational perturbations. Please also see the supplementary video for more details.

B.2. Real-World Scenes (LLFF Dataset)

Dataset. The LLFF dataset [30] consists of 8 forward-facing scenes with RGB images sequentially captured by hand-held cameras. In the original NeRF paper [30], the test views were selected by holding out every 8th frame from the video sequence and training with the remaining frames. Unlike Mildenhall *et al*. [30], however, we hold out the last 10% of the frames for evaluation and train with the first 90% frames. This train/test split does not assume that the held-out views are interpolations of the training views, which allows a more practical simulation of predicting future viewpoints from previous observations. The statistics of the train/test split for each scene is provided in Table 4.

	Fern	Flower	Fortress	Horns	Leaves	Orchids	Room	T-rex
split	18/2	31/3	38/4	56/6	24/2	23/2	37/4	50/5
total	20	34	42	62	26	25	41	55

Table 4: Dataset statistics of the train/test splits for the real-world scene (LLFF) experiments, where we hold out the last 10% frames from each sequences.

Full comparison. We provide a more complete evaluation of the LLFF experiment in Table 5, where we also include the baseline without any positional encoding. Note that we consider the same schedule for all scenes in the dataset (adjusting the positional encoding from iterations 20K to 100K); due to the per-scene optimization nature, however,

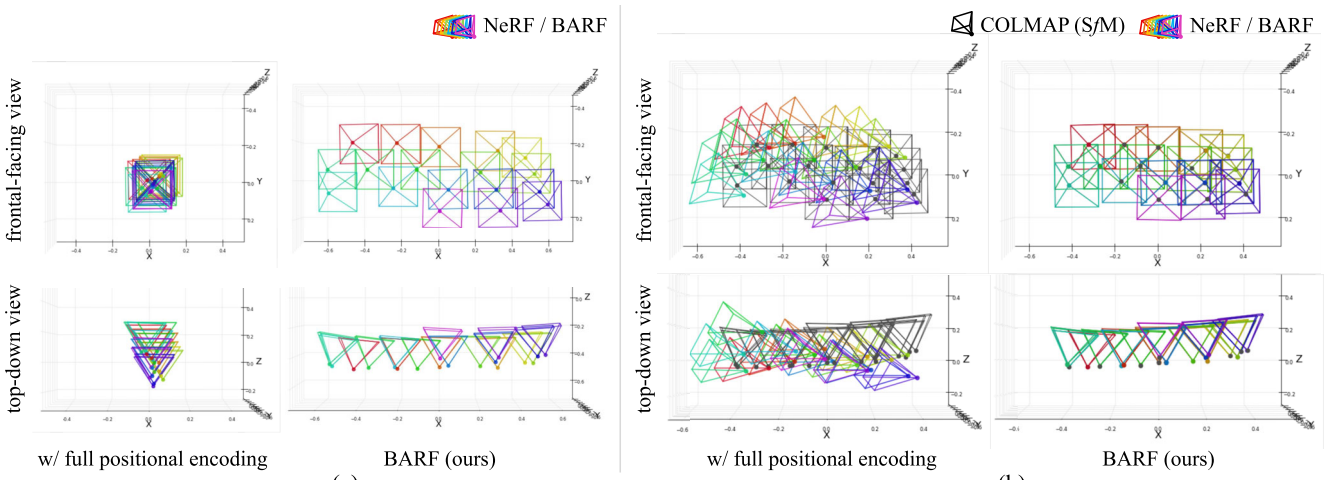


Figure 10: Visualization of the **optimized camera poses** for the *fern* scene. The poses for both the baseline NeRF (with full positional encoding) and BARF are initialized to the identity transform for all frames. (a) The camera poses of the baseline NeRF get stuck in a suboptimal solution that does not accurately reflect the actual viewpoints, whereas BARF can effectively optimize for the underlying poses. (b) We compare the optimized poses to those computed from S*f*M [44] (colored in black), where we align the pose trajectories using Procrustes analysis. The camera poses optimized by BARF highly agree with those from S*f*M, whereas those from the baseline NeRF cannot be well-aligned with Procrustes analysis. Therefore, there is no systematic way of finding a reasonable set of corresponding held-out views with respect to the optimized coordinate system.

Scene	Camera pose registration						View synthesis quality											
	Rotation (°) ↓			Translation ↓			PSNR ↑				SSIM ↑				LPIPS ↓			
	full pos.enc.	w/o pos.enc.	BARF	full pos.enc.	w/o pos.enc.	BARF	full pos.enc.	w/o pos.enc.	BARF	ref. NeRF	full pos.enc.	w/o pos.enc.	BARF	ref. NeRF	full pos.enc.	w/o pos.enc.	BARF	ref. NeRF
Fern	74.452	0.194	**0.191**	30.167	0.194	**0.192**	9.81	23.73	**23.79**	23.72	0.187	0.709	**0.710**	0.733	0.853	0.371	**0.311**	0.262
Flower	2.525	0.883	**0.251**	2.635	0.297	**0.224**	17.08	**24.66**	23.37	23.24	0.344	**0.739**	0.698	0.668	0.490	**0.200**	0.211	0.244
Fortress	75.094	**0.320**	0.479	33.231	**0.289**	0.364	12.15	28.35	**29.08**	25.97	0.270	0.774	**0.823**	0.786	0.807	0.206	**0.132**	0.185
Horns	58.764	**0.182**	0.304	32.664	**0.170**	0.222	8.89	22.27	**22.78**	20.35	0.158	0.724	**0.727**	0.624	0.805	0.312	**0.298**	0.421
Leaves	88.091	2.938	**1.272**	13.540	0.468	**0.249**	9.64	**19.08**	18.78	15.33	0.067	**0.566**	0.537	0.306	0.782	0.375	**0.353**	0.526
Orchids	37.104	**0.550**	0.627	20.312	**0.396**	0.404	9.42	19.27	**19.45**	17.34	0.085	0.566	**0.574**	0.518	0.806	0.313	**0.291**	0.307
Room	173.811	0.384	**0.320**	66.922	0.311	**0.270**	10.78	30.71	**31.95**	32.42	0.278	0.928	**0.940**	0.948	0.871	0.135	**0.099**	0.080
T-rex	166.231	**0.138**	1.138	53.309	**0.261**	0.720	10.48	22.48	**22.55**	22.12	0.158	**0.783**	0.767	0.739	0.885	**0.197**	0.206	0.244
Mean	84.509	0.699	**0.573**	31.598	**0.298**	0.331	11.03	23.82	**23.97**	22.56	0.193	**0.724**	0.722	0.665	0.787	0.264	**0.238**	0.283

Table 5: Full quantitative comparison of NeRF on the LLFF forward-facing scenes from *unknown* camera poses. BARF and our baseline without positional encoding are competitive in different metrics. An optimal coarse-to-fine schedule for BARF could be theoretically found per scene that are at least as good as the baseline methods; exhaustively or adaptively search for such optimal schedule is currently out of scope of this paper. Translation errors are scaled by 100.

the optimal coarse-to-fine scheduling for each scene would actually be data-dependent. Despite this, the coarse-to-fine scheduling considered here already allows BARF to achieve an averaged similar or better performance on real-world scenes. An exhaustive analysis of searching for the best scheduling is currently out of scope of this paper.

In the main LLFF experiments, we sample 3D points along each ray linearly in the inverse depth (disparity) space, where the lower and upper bounds are the image plane and infinity respectively (*i.e.* $1/z_{\text{near}} = 1$ and $1/z_{\text{far}} = 0$). To analyze the effect of depth parametrization on the performance of real-world scenes, we run an additional set of the same experiments by sampling the 3D points in the regular (metric) depth space, bounded by $z_{\text{near}} = 1$ and $z_{\text{far}} = 20$.

We report the quantitative results in Table 6. The baseline NeRF with full positional encoding still performs poorly in all metrics. Although the baseline without positional encoding may be slightly better than BARF in this setup, all methods being compared here exhibit better performance when the 3D points are sampled in the inverse depth space. We present empirical results as a supplement and leave a complete analysis of depth parametrization to future work.

Figure 11: Additional novel view synthesis results from the real-world scene experiment (LLFF dataset). Instead of visualizing the held-out views computed by Procrustes analysis, we show qualitative results at new viewpoints by sampling camera pose perturbations around the viewpoint from the training set (closest to the average pose). Note that for this set of qualitative results, we do not have ground-truth RGB images to compare against. BARF can optimize for scene representations of much higher quality. Please refer to the supplementary video for more details.

Scene	Camera pose registration						View synthesis quality											
	Rotation (°) ↓			Translation ↓			PSNR ↑				SSIM ↑				LPIPS ↓			
	full pos.enc.	w/o pos.enc.	BARF	full pos.enc.	w/o pos.enc.	BARF	full pos.enc.	w/o pos.enc.	BARF	ref. NeRF	full pos.enc.	w/o pos.enc.	BARF	ref. NeRF	full pos.enc.	w/o pos.enc.	BARF	ref. NeRF
Fern	164.243	**0.391**	0.448	18.265	**0.260**	0.283	9.17	23.39	**23.55**	22.76	0.148	**0.700**	**0.700**	0.655	1.041	0.362	**0.335**	0.397
Flower	7.462	**0.177**	3.282	1.959	**0.211**	0.724	18.81	**23.63**	22.99	23.37	0.408	**0.710**	0.651	0.654	0.657	**0.224**	0.227	0.272
Fortress	172.581	**0.502**	0.576	46.673	**0.466**	0.468	11.17	26.75	**26.92**	25.67	0.222	0.684	**0.716**	0.662	1.122	0.348	**0.270**	0.403
Horns	34.840	**0.248**	0.266	18.207	**0.223**	0.228	8.95	21.52	**21.79**	20.37	0.174	**0.714**	0.701	0.599	1.028	0.325	**0.310**	0.464
Leaves	4.708	**1.194**	1.832	1.105	**0.261**	0.367	11.66	**18.36**	17.68	16.34	0.104	**0.516**	0.473	0.353	0.822	0.407	**0.356**	0.534
Orchids	172.600	0.531	**0.443**	37.887	**0.413**	**0.413**	8.22	**18.84**	18.57	16.97	0.062	**0.536**	0.513	0.402	1.086	**0.357**	0.373	0.564
Room	160.757	0.456	**0.207**	51.988	0.454	**0.203**	8.09	30.90	**31.99**	32.10	0.127	0.924	**0.938**	0.935	1.215	0.139	**0.104**	0.109
T-rex	175.893	**0.334**	5.586	61.026	**0.328**	3.085	8.30	**22.74**	21.24	22.42	0.123	**0.794**	0.731	0.770	1.174	**0.187**	0.225	0.205
Mean	111.635	**0.479**	1.580	29.639	**0.327**	0.721	10.54	**23.26**	23.09	22.50	0.171	**0.698**	0.678	0.629	1.018	0.294	**0.275**	0.368

Table 6: Quantitative results of NeRF on the LLFF forward-facing scenes from *unknown* camera poses, sampling the 3D points in the regular depth space (instead of the inverse depth space). Translation errors are scaled by 100.

References

[1] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, and Richard Szeliski. Building rome in a day. *ACM Communications*, 2011. 2

[2] Hatem Alismail, Brett Browning, and Simon Lucey. Photometric bundle adjustment for vision-based slam. In *Asian Conference on Computer Vision*, pages 324–341. Springer, 2016. 2

[3] Mark Boss, Raphael Braun, Varun Jampani, Jonathan T Barron, Ce Liu, and Hendrik Lensch. Nerd: Neural reflectance decomposition from image collections. *arXiv*, 2020. 3

[4] Gaurav Chaurasia, Sylvain Duchene, Olga Sorkine-Hornung, and George Drettakis. Depth synthesis and local warps for plausible image-based navigation. *TOG*, 2013. 2

[5] Shenchang Eric Chen and Lance Williams. View interpolation for image synthesis. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, 1993. 2

[6] Andrew J Davison, Ian D Reid, Nicholas D Molton, and Olivier Stasse. Monoslam: Real-time single camera slam. *TPAMI*, 2007. 2

[7] Paul E Debevec, Camillo J Taylor, and Jitendra Malik. Modeling and rendering architecture from photographs: A hybrid geometry-and image-based approach. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 1996. 2

[8] Amaël Delaunoy and Marc Pollefeys. Photometric bundle adjustment for dense multi-view 3d modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1486–1493, 2014. 2

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 5

[10] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *CVPR*, 2018. 2

[11] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-net: A trainable cnn for joint detection and description of local features. *arXiv*, 2019. 2

[12] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *TPAMI*, 2017. 2

[13] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *European conference on computer vision*, pages 834–849. Springer, 2014. 1, 2

[14] SM Ali Eslami, Danilo Jimenez Rezende, Frederic Besse, Fabio Viola, Ari S Morcos, Marta Garnelo, Avraham Ruderman, Andrei A Rusu, Ivo Danihelka, Karol Gregor, et al. Neural scene representation and rendering. *Science*, 360(6394):1204–1210, 2018. 1

[15] John Flynn, Michael Broxton, Paul Debevec, Matthew DuVall, Graham Fyffe, Ryan Overbeck, Noah Snavely, and Richard Tucker. Deepview: View synthesis with learned gradient descent. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2367–2376, 2019. 1, 2

[16] Chen Gao, Yichang Shih, Wei-Sheng Lai, Chia-Kai Liang, and Jia-Bin Huang. Portrait neural radiance fields from a single image. *arXiv*, 2020. 3

[17] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004. 1

[18] Peter Hedman, Suhib Alsisan, Richard Szeliski, and Johannes Kopf. Casual 3d photography. *TOG*, 2017. 2

[19] Benno Heigl, Reinhard Koch, Marc Pollefeys, Joachim Denzler, and Luc Van Gool. Plenoptic modeling and rendering from image sequences taken by a hand-held camera. 1999. 2

[20] Ronghang Hu and Deepak Pathak. Worldsheet: Wrapping the world in a 3d sheet for view synthesis from a single image. *arXiv*, 2020. 2

[21] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. 5

[22] Johannes Kopf, Michael F Cohen, and Richard Szeliski. First-person hyper-lapse videos. *TOG*, 2014. 2

[23] Marc Levoy. Efficient ray tracing of volume data. *ACM Transactions on Graphics (TOG)*, 9(3):245–261, 1990. 2, 4

[24] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 1996. 2

[25] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. *arXiv*, 2020. 3

[26] Chen-Hsuan Lin, Oliver Wang, Bryan C Russell, Eli Shechtman, Vladimir G Kim, Matthew Fisher, and Simon Lucey. Photometric mesh optimization for video-aligned 3d object reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 6

[27] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *arXiv preprint arXiv:1906.07751*, 2019. 1, 2

[28] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'81, pages 674–679, 1981. 3

[29] Moustafa Meshry, Dan B Goldman, Sameh Khamis, Hugues Hoppe, Rohit Pandey, Noah Snavely, and Ricardo Martin-Brualla. Neural rerendering in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6878–6887, 2019. 1

[30] Ben Mildenhall, Pratul P Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)*, 38(4):1–14, 2019. 7, 10

[31] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision*, 2020. 1, 2, 4, 5, 7, 8

[32] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015. 1, 2

[33] Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. Dtam: Dense tracking and mapping in real-time. In *ICCV*, 2011. 2

[34] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. *arXiv*, 2020. 3

[35] Yuki Ono, Eduard Trulls, Pascal Fua, and Kwang Moo Yi. Lf-net: Learning local features from images. *arXiv*, 2018. 2

[36] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo-Martin Brualla. Deformable neural radiance fields. *arXiv preprint arXiv:2011.12948*, 2020. 3, 5

[37] Marc Pollefeys, Reinhard Koch, and Luc Van Gool. Self-calibration and metric reconstruction inspite of varying and unknown intrinsic camera parameters. *International Journal of Computer Vision*, 32(1):7–25, 1999. 2

[38] Marc Pollefeys, Luc Van Gool, Maarten Vergauwen, Frank Verbiest, Kurt Cornelis, Jan Tops, and Reinhard Koch. Visual modeling with a hand-held camera. *International Journal of Computer Vision*, 59(3):207–232, 2004. 2

[39] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. *arXiv*, 2020. 3

[40] Daniel Rebain, Wei Jiang, Soroosh Yazdani, Ke Li, Kwang Moo Yi, and Andrea Tagliasacchi. Derf: Decomposed radiance fields. *arXiv*, 2020. 3

[41] Konstantinos Rematas, Ricardo Martin-Brualla, and Vittorio Ferrari. Sharf: Shape-conditioned radiance fields from a single view. *arXiv*, 2021. 3

[42] Gernot Riegler and Vladlen Koltun. Free view synthesis. In *ECCV*, 2020. 2

[43] Gernot Riegler and Vladlen Koltun. Stable view synthesis. *arXiv*, 2020. 2

[44] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. 1, 2, 7, 8, 9, 11

[45] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. *arXiv*, 2020. 3

[46] Meng-Li Shih, Shih-Yang Su, Johannes Kopf, and Jia-Bin Huang. 3d photography using context-aware layered depth inpainting. In *CVPR*, 2020. 2

[47] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *Advances in Neural Information Processing Systems*, 2016. 1

[48] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. In *SIGGRAPH*. 2006. 2

[49] Noah Snavely, Steven M Seitz, and Richard Szeliski. Modeling the world from internet photo collections. *IJCV*, 2008. 2

[50] Pratul P Srinivasan, Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T Barron. Nerv: Neural reflectance and visibility fields for relighting and view synthesis. *arXiv*, 2020. 3

[51] Pratul P Srinivasan, Richard Tucker, Jonathan T Barron, Ravi Ramamoorthi, Ren Ng, and Noah Snavely. Pushing the boundaries of view extrapolation with multiplane images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 175–184, 2019. 1, 2

[52] Richard Szeliski and Polina Golland. Stereo matching with transparency and matting. In *ICCV*, 1998. 2

[53] Matthew Tancik, Pratul P Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In *Advances in Neural Information Processing Systems*, 2020. 2, 4

[54] Ayush Tewari, Ohad Fried, Justus Thies, Vincent Sitzmann, Stephen Lombardi, Kalyan Sunkavalli, Ricardo Martin-Brualla, Tomas Simon, Jason Saragih, Matthias Nießner, et al. State of the art on neural rendering. In *Computer Graphics Forum*, volume 39, pages 701–727. Wiley Online Library, 2020. 1

[55] Richard Tucker and Noah Snavely. Single-view view synthesis with multiplane images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 551–560, 2020. 1, 2

[56] Shubham Tulsiani, Richard Tucker, and Noah Snavely. Layer-structured 3d scene inference via view synthesis. In *ECCV*, 2018. 2

[57] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30:5998–6008, 2017. 2, 4

[58] Chaoyang Wang, José Miguel Buenaposada, Rui Zhu, and Simon Lucey. Learning depth from monocular videos using direct methods. In *CVPR*, 2018. 2

[59] Rui Wang, Martin Schworer, and Daniel Cremers. Stereo dso: Large-scale direct sparse visual odometry with stereo cameras. In *ICCV*, 2017. 2

[60] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. Nerf −−: Neural radiance fields without known camera parameters. *arXiv*, 2021. 3

[61] Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. Synsin: End-to-end view synthesis from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7467–7477, 2020. 1, 2

[62] Changchang Wu et al. Visualsfm: A visual structure from motion system. 2

[63] Wenqi Xian, Jia-Bin Huang, Johannes Kopf, and Changil Kim. Space-time neural irradiance fields for free-viewpoint video. *arXiv*, 2020. 3

[64] Anqi Joyce Yang, Can Cui, Ioan Andrei Bârsan, Raquel Urtasun, and Shenlong Wang. Asynchronous multi-view SLAM. In *ICRA*, 2021. 2

[65] Lin Yen-Chen, Pete Florence, Jonathan T Barron, Alberto Rodriguez, Phillip Isola, and Tsung-Yi Lin. inerf: Inverting neural radiance fields for pose estimation. *arXiv preprint arXiv:2012.05877*, 2020. 3, 6

[66] Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *CVPR*, 2018. 2

[67] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *CVPR*, 2021. 3

[68] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv*, 2020. 3

[69] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 6

[70] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, 2017. 2

[71] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. In *SIGGRAPH*, 2018. 1, 2

[72] C Lawrence Zitnick, Sing Bing Kang, Matthew Uyttendaele, Simon Winder, and Richard Szeliski. High-quality video view interpolation using a layered representation. *TOG*, 2004. 2