AligNeRF: High-Fidelity Neural Radiance Fields via Alignment-Aware Training

Yifan Jiang[1]*, Peter Hedman[2], Ben Mildenhall[2], Dejia Xu[1], Jonathan T. Barron[2],
Zhangyang Wang[1], Tianfan Xue[3]†
[1]University of Texas at Austin, [2]Google Research, [3]The Chinese University of Hong Kong

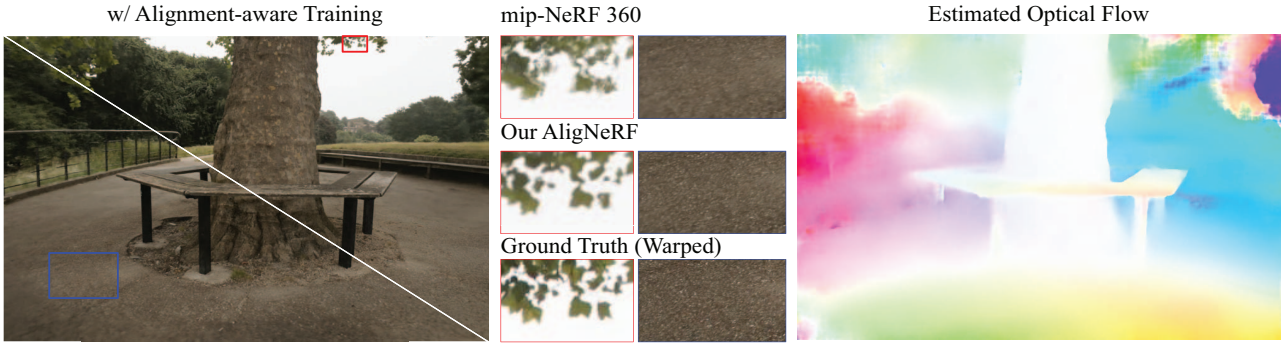

Figure 1. **Left:** Images rendered by mip-NeRF 360 [2] (w/o alignment-aware training) and our AligNeRF (w/ alignment-aware training), with the same amount of training time for both methods. **Middle:** Magnified cropped regions. Here, the ground truth has been aligned with the mip-NeRF 360 rendering, as described in Sec. 4.3. **Right:** Visualization of misalignment. This is the estimated optical flow between the mip-NeRF 360 rendering and the original ground truth images.

Abstract

Neural Radiance Fields (NeRFs) are a powerful representation for modeling a 3D scene as a continuous function. Though NeRF is able to render complex 3D scenes with view-dependent effects, few efforts have been devoted to exploring its limits in a high-resolution setting. Specifically, existing NeRF-based methods face several limitations when reconstructing high-resolution real scenes, including a very large number of parameters, misaligned input data, and overly smooth details. In this work, we conduct the first pilot study on training NeRF with high-resolution data and propose the corresponding solutions: 1) marrying the multilayer perceptron (MLP) with convolutional layers which can encode more neighborhood information while reducing the total number of parameters; 2) a novel training strategy to address misalignment caused by moving objects or small camera calibration errors; and 3) a high-frequency aware loss. Our approach is nearly free without introducing obvious training/testing costs, while experiments on different datasets demonstrate that it can recover more high-frequency details compared with the current state-of-the-art NeRF models. Project page: https://yifanjiang19.github.io/alignerf.

* This work was performed while Yifan Jiang interned at Google.
† This work was performed while Tianfan Xue worked at Google.

1. Introduction

Neural Radiance Field (NeRF [24]) and its variants [1–3, 7, 19, 20], have recently demonstrated impressive performance for learning geometric 3D representations from images. The resulting high-quality scene representation enables an immersive novel view synthesis experience with complex geometry and view-dependent appearance. Since the origin of NeRF, an enormous amount of work has been made to improve its quality and efficiency, enabling reconstruction from data captured "in-the-wild" [15, 20] or a limited number of inputs [7, 11, 26, 32, 44] and generalization across multiple scenes [4, 41, 45].

However, relatively little attention has been paid to high-resolution reconstruction. mip-NeRF [1] addresses excessively blurred or aliased images when rendering at different resolutions, modelling ray samples with 3D conical frustums instead of infinitesimally small 3D points. mip-NeRF 360 [2] further extends this approach to unbounded scenes that contain more complex appearance and geometry. Nevertheless, the highest resolution data used in these two works is only 1280×840 pixels, which is still far away from the resolution of a standard HD monitor (1920×1080), not to mention a modern smartphone camera (4032×3024).

2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) | 979-8-3503-0129-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/CVPR52729.2023.00013

979-8-3503-0129-8/23/$31.00 ©2023 IEEE
DOI 10.1109/CVPR52729.2023.00013

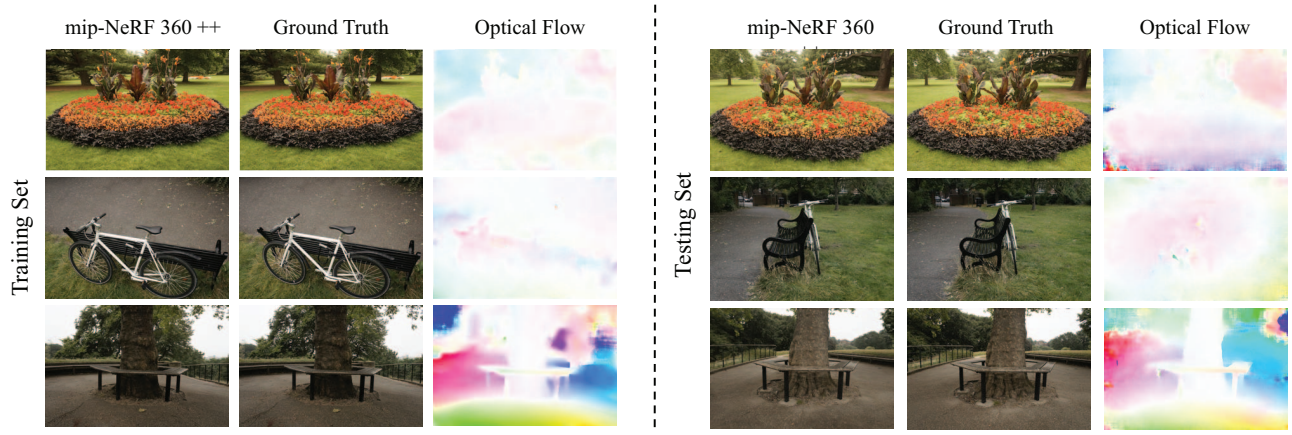

Figure 2. **Analysis of misalignment between rendered and ground truth images**. **mip-NeRF 360++**: Images rendered by a stronger mip-NeRF 360 [2] model (16× larger MLPs than the original). **Ground Truth**: The captured images used for training and testing. **Optical Flow**: Optical flow between the mip-NeRF 360++ and ground truth images, estimated by PWC-Net [38]. Significant misalignment is present in both training and test view renderings.

In this paper, we conduct the first pilot study of training neural radiance fields in the high-fidelity setting, using higher-resolution images as input. This introduces several hurdles. First, the major challenge of using high-resolution training images is that encoding all the high-frequency details requires significantly more parameters, which leads to a much longer training time and higher memory cost, sometimes even making the problem intractable [2, 20, 30].

Second, to learn high-frequency details, NeRF requires accurate camera poses and motionless scenes during capture. However, in practice, camera poses recovered by Structure-from-Motion (SfM) algorithms inevitably contain pixel-level inaccuracies [18]. These inaccuracies are not noticeable when training on downsampled low-resolution images, but cause blurry results when training NeRF with higher-resolution inputs. Moreover, the captured scene may also contain unavoidable motion, like moving clouds and plants. This not only breaks the static-scene assumption but also decreases the accuracy of estimated camera poses. Due to both inaccurate camera poses and scene motion, NeRF's rendered output is often slightly misaligned from the ground truth image, as illustrated in Fig. 2. We investigate this phenomenon in Sec. 4.3, demonstrating that image quality can be significantly improved by iteratively training NeRF and re-aligning the input images with NeRF's estimated geometry. The analysis shows that misalignment results in NeRF learning distorted textures, as it is trained to minimize the difference between rendered frames and ground truth images. Previous work mitigates this issue by jointly optimizing NeRF and camera poses [5, 17, 23, 43], but these methods cannot handle subtle object motion and often introduce non-trivial training overheads, as demonstrated in Sec 4.6.

To tackle these issues, we present AligNeRF, an alignment-aware training strategy that can better preserve high-frequency details. Our solution is two-fold: an approach to efficiently increase the representational power of NeRF, and an effective method to correct for misalignment. To efficiently train NeRF with high-resolution inputs, we marry convolutions with NeRF's MLPs, by sampling a chunk of rays in a local patch and applying ConvNets for post-processing. Although a related idea is discussed in NeRF-SR [40], their setting is based on rendering test images at a higher resolution than the training set. Another line of work combines volumetric rendering with generative modeling [27, 34], where ConvNets are mainly used for efficient upsampling and generative texture synthesis, rather than solving the inverse problem from many input images. In contrast, our approach shows that the inductive prior from a small ConvNet improves NeRF's performance on high-resolution training data, without introducing significant computational costs.

In this new pipeline, we render image patches during training. This allows us to further tackle misalignments between the rendered patch and ground truth that may have been caused by minor pose errors or moving objects. First, we analyze how misalignment affects image quality by leveraging the estimated optical flow between rendered frames and their corresponding ground truth images. We discuss the limitations of previous misalignment-aware losses [22, 48], and propose a novel alignment strategy tailored for our task. Moreover, our patch-based rendering strategy also enables patch-wise loss functions, beyond a simple mean squared error. That motivates us to design a new frequency-aware loss, which further improves the rendering quality with no overheads. As a result, AligNeRF largely outperforms the current best method for high-

resolution 3D reconstruction tasks with few extra costs.

To sum up, our contributions are as follows:

- An analysis demonstrating the performance degradation caused by misalignment in high-resolution training data.
- A novel convolution-assisted architecture that improves the quality of rendered images with minor additional costs.
- A novel patch alignment loss that makes NeRF more robust to camera pose error and subtle object motion, together with a patch-based loss to improve high-frequency details.

2. Preliminaries

The vanilla NeRF [24] method takes hundreds of images with the corresponding camera poses as the training set, working on synthetically rendered objects or real-world forward-facing scenes. Later works extend NeRF to unconstrained photo collections [20], reduce its training inputs to only sparsely sampled views [7, 11, 26, 32], apply it to relighting tasks [36, 49], speed up training/inference time [9, 19, 25, 37], and generalize it to unseen scenes [4, 41, 45]. In contrast, we explore the problem of training on high-resolution input data in this work.

2.1. NeRF

NeRF takes a 3D location $\mathbf{x} = (x, y, z)$ and 2D viewing direction $\mathbf{d} = (\theta, \phi)$ as input and outputs an emitted color $\mathbf{c} = (r, g, b)$ and volume density σ. This continuous 5D scene representation is approximated by an MLP network $F_\Theta : (\mathbf{x}, \mathbf{d}) \rightarrow (\mathbf{c}, \sigma)$. To compute the output color for a pixel, NeRF approximates the volume rendering integral using numerical quadrature [21]. For each camera ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$, where $\mathbf{o}$ is the camera origin and $\mathbf{d}$ is the ray direction, the expected color $\widehat{C}(\mathbf{r})$ of $\mathbf{r}(t)$ with near and far bounds t_n and t_f can be formulated as:

$$\widehat{C}(\mathbf{r}) = \sum_k w_k \mathbf{c}_k \,, \tag{1}$$

$$\text{with } w_k = T_k(1 - \exp(-\tau_k(t_{k+1} - t_k))) \tag{2}$$

$$\text{and } T_k = \exp\left(-\sum_{k'<k} \tau_{k'}\left(t_{k'+1} - t_{k'}\right)\right), \tag{3}$$

where $\{t_k\}_{k=1}^K$ represents a set of sampled 3D points between near and far planes of the camera, and τ indicates the estimated volume density σ. Since neural networks are known to be biased to approximate lower frequency functions [28], NeRF uses a positional encoding function $\gamma(\cdot)$ consisting of sinusoids at multiple frequencies to introduce high-frequency variations into the network input.

2.2. mip-NeRF 360

To tackle the aliasing issue when the resolution of rendered views differs from training images, mip-NeRF [1] proposes to sample volumetric frustums from a cone surrounding the ray rather than sampling infinitesimally small 3D points. mip-NeRF 360 [2] extends this integrated positional encoding to unbounded real-world scenes by adopting a "contraction" function that warps arbitrarily far scene content into a bounded domain (similarly to NeRF++ [46]).

To make training more efficient, mip-NeRF 360 adopts an online distillation strategy. The coarse NeRF (also named "Proposal MLP") is no longer supervised by photometric reconstruction loss, but instead learns the distilled knowledge of structure from the fine NeRF (also named "NeRF MLP"). By doing so, the parameter count of the Proposal MLP can be largely reduced, as it does not contribute to the final RGB color. Because the ray samples are better guided by the Proposal MLP, it is also possible to significantly reduce the number of queries to the final NeRF MLP. Thus a fixed computational budget can be reallocated to significantly enlarge this NeRF MLP, improving the rendered image quality for the same total cost as mip-NeRF.

3. Method

In this section, we introduce AligNeRF, an alignment-aware training strategy to address the obstacles discussed in Sec. 1 and analyzed in Sec. 4.3. AligNeRF is an easy-to-plug-in component for any NeRF-like models, including both point-sampling approaches and frustum-based approaches. AligNeRF uses staged training: starting with an initial "normal" pre-training stage, followed by an alignment-aware fine-tuning stage. We choose mip-NeRF 360 as our baseline, as it is the state-of-the-art NeRF method for complex unbounded real-world scenes. Next, we introduce our convolution-augmented architecture, then present our misalignment-aware training procedure and high-frequency loss.

3.1. Marrying Coordinate-Based Representations with Convolutions

Inspired by the recent success of encoding inductive priors in vision transformers [6], our first step is to explore how to effectively encode local inductive priors for coordinate-based representations such as NeRF. Recall that NeRF-like models generally construct a coordinate-to-value mapping function and randomly sample a batch of rays to optimize its parameters, which prevents us from performing any patch-based processing. Thus, our first modification is to switch from random sampling to patch-based sampling, with respect to camera rays (we use 32×32 patches in our experiments).

This patch-based sampling strategy allows us to gather

a small local image region during each iteration and thus make use of 2D local neighborhood information when rendering each pixel. To begin with, we change the number of output channels of the last layer in MLPs from 3 to a larger N, and apply numerical integration along the ray in this feature space rather than in RGB space. This helps gather richer representation in each camera ray. Next, we add a simple 3-layer convolutional network with ReLU activations and 3×3 kernels, following the volumetric rendering function. We adopt "reflectance" padding for the inputs of each convolution and remove the padding values in the outputs, to prevent checkerboard artifacts. At the end of this network, we use a feed-forward perceptron layer to convert the representation from feature space to RGB space. As a result, the rendering process for each pixel does not only rely on the individual ray/cone along that direction, but also depends on its neighboring regions, which help produces better texture detail. Because our CNN is very shallow and does not perform any upsampling, we do not observe any resulting multiview inconsistency in the image outputs.

3.2. Alignment-Aware Loss

Recall that NeRF models a 3D scene using a rendering function $F_\Theta : (\mathbf{x}, \mathbf{d}) \rightarrow (\mathbf{c}, \sigma)$ mapping the coordinates of 3D points to the properties of the scenes. Under this framework, the accuracy of camera poses is crucial for NeRF training, otherwise, rays observing the same 3D point from different viewpoints may not converge to the same location in space. The vanilla NeRF [24] solves this problem by capturing images over a very short time span (to prevent scene motion and lighting changes) and adopts COLMAP [33] to calculate camera parameters. This data preparation pipeline is mostly reliable except 1) There is a gap between the ground truth camera poses and the camera poses from COLMAP, as has been pointed out by previous works [10, 29]; and 2) It is usually hard to avoid images with swaying plants and other nonrigid objects in uncontrolled outdoor scenes, which further hurts the performance of COLMAP. In the high-resolution reconstruction setting, the misalignment issue caused by camera poses and moving objects can be further amplified, as pixel-space misalignment scales linearly with resolution. We explore how much this misalignment make affect the quality of rendered images in Sec. 4.3. To address this issue, we propose an alignment-aware training strategy that can be adopted to refine the quality of rendered images.

Despite the distorted textures, we observe that NeRF still learns the rough structures from the misaligned images, as shown in the second column of Fig. 3. Taking advantage of this, we proposed a loss between aligned ground truth and rendered patches. Let G denote the ground truth patch and R denote the rendered patch. We sample a larger size of ground truth patch G during each iteration and search over every possible subpatch G_i for the best match with the smaller rendered patch R. Since NeRF may render a very blurry patch R that seems equally well-aligned with many possible patches G_i from the ground truth set, we additionally set a regularization term based on Euclidean distance as a penalty for this search space. The final loss function is

$$\mathcal{L}_{PM}(G, R) = \min_{i=1,\ldots,N} (D(G_i, R) + \lambda \cdot D_{coord}(G_i, R)) \tag{4}$$

where $D(G_i, R) = \|G_i - R\|_2^2$ and $D_{coord}(G_i, R) = \sqrt{\Delta x^2 + \Delta y^2}$, and $\Delta x/\Delta y$ are the horizontal/vertical offset between G_i and R. We adopt λ to control the regularization term and empirically set it to 0.01. In all of our experiments, we sample 48×48 patches for G and render 32×32 patches for R within each iteration.

Similar losses are also used in image super-resolution [48] or style transfer [22]. However, those losses are defined on single pixels, while our alignment-aware loss is defined on patches, where alignment vectors $(\Delta x, \Delta y)$ can be more robustly estimated.

3.3. High-frequency aware Loss

Mean squared error (MSE) loss is commonly used to supervise NeRF training, but it is well-known in the image processing literature that MSE often leads to blurry output images [12, 16]. Given our patch sampling strategy, we can adopt a perceptual loss, which better preserves high-frequency details. We first attempted to use L2 loss of a pre-trained VGG features [35]. However, similar to other image restoration tasks [16], we found that perceptual loss produced more high-frequency details but sometimes distorted the actual texture of the object. Instead, we modify the original perceptual loss proposed by Johnson et al. [12], by only using the output of the first block before max-pooling:

$$\mathcal{L}_{\text{HF}}(G_i, R) = \frac{1}{CWH} \|F(G_i) - F(R)\|_2^2, \tag{5}$$

where G_i denotes the ground truth patch after alignment and R denotes the rendered patch, F denotes the first block of a pre-trained VGG-19 model, and C, W, and H are the dimensions of the extracted feature maps. With this change, the proposed loss improves the high-frequency details while still preserving real textures.

In summary, the major difference between AligNeRF and previous work is switching from per-pixel MSE loss to a combination of patch-based MSE loss (accounting for misalignment) and a shallow VGG feature-space loss to improve high-frequency detail:

$$\mathcal{L}_{MSE} \longrightarrow \mathcal{L}_{PM} + w \cdot \mathcal{L}_{HF}, \tag{6}$$

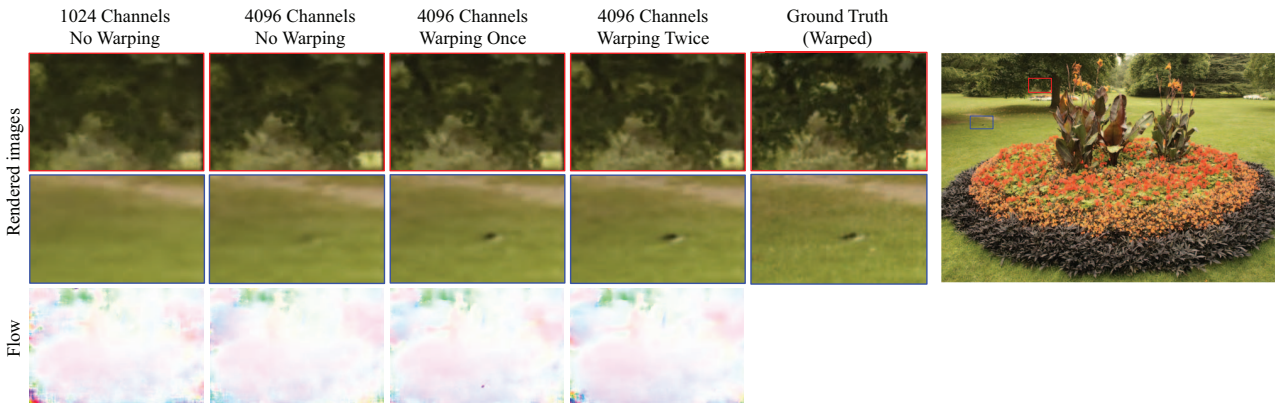

Figure 3. **Misalignment prevents recovery of high-frequency detail.** We randomly select an example test set image. **1024 channels, No Warping**: We train mip-NeRF 360 [2] using the original training set with default parameters (1024 channels for the "NeRF MLP"). **4096 channels, No Warping**: We scale mip-NeRF 360 up with $4\times$ more channels in the "NeRF MLP". **4096 channels, Warping Once**: We train the 4096 channel model using the iterative alignment strategy described in Section 4.3, which uses aligned ground truth images. **4096 channels, Warping Twice**: Similar to "4096 channels, Warping Once", but with two iterations for better alignment. The bottom row shows the flow between rendered frames and ground truth. Note that NeRF can recover much more high-frequency detail with aligned training data (columns 3 and 4).

where w is empirically set to be 0.05. To facilitate comparisons and demonstrate the use of AligNeRF as a simple plug-and-play modification, other regularization losses from mip-NeRF 360 are kept the same by default.

4. Experiments

4.1. Dataset

The main testbed of our experiments are the outdoor scenes from mip-NeRF 360 [2], as this is the most challenging benchmark with unbounded real-world scenes, complex texture details, and subtle scene motion. [2] captured 5 outdoor scenes, including "*bicycle*", "*flowers*", "*treehill*", "*garden*", and "*stump*". However, these captured images are resized to be $4\times$ smaller, resulting in a resolution of approximately 1280×840. To better investigate misalignment we also perform experiments on a higher-resolution version (2560×1680) of these scenes. Finally, we also generate new better-aligned test set to avoid drawing biased conclusions from misaligned data, as described in Sec. 4.3.

4.2. Training Details

Standard Training Setting. Baseline algorithms are optimized following the standard training procedure of mip-NeRF 360: a batch size of 16384, an Adam [13] optimizer with $\beta1 = 0.9$ and $\beta2 = 0.999$, and gradient clipping to a norm of 1e-3. The initial and final learning rates are set to be 2×10^{-3} and 2×10^{-5}, with an annealed log-linear decay strategy. The first 512 iterations are used for a learning rate warm-up phase.

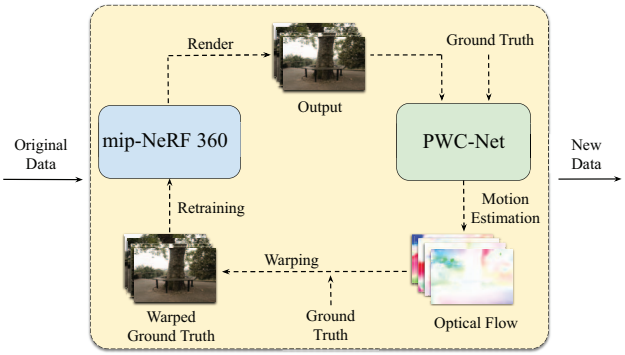

Figure 4. **Illustration of one iteration in our iterative alignment strategy**. Note that we were unable to use RAFT [39], as it runs out of memory on our high-resolution images. **This strategy only serves for analysis and does not represent the proposed algorithm in this work.**

Alignment-aware Training Setting. We split the training process of AligNeRF into two stages: the pre-training stage (MLPs only) and the fine-tuning stage (MLPs together with ConvNets). In the first stage, we generally follow the standard training procedure but with **60%** as many iterations. This is followed by another stage using the alignment-aware training strategy described in Sections 3.1-3.3, where we sample a 32×32 patch in each batch instead of individual pixels and set the batch size to 32 patches. Since the total number of rays is $2\times$ larger than the pre-training stage, we only use **20%** as many iterations for the second stage to equalize the total number of rays seen during training. This makes the total training cost of both stages approxi-

Table 1. Analysis of misalignment. All results are reported on the "*flowers*" scene (2560×1680) in the "outdoor" dataset [2]. The "warping" column indicates whether we use the re-generated data and how many iterations it takes. c represents the channels of the "NeRF MLP", which contributes to the RGB color. Training time contains both the NeRF training time and data generation time. We report metrics on both the standard and warped test sets respectively, split by "/".

Methods	Warping	PSNR ↑	SSIM ↑	LPIPS ↓	Time	#Params
mip-NeRF 360 ($c = 4096$)	None	21.23/22.06	0.560/0.613	0.384/0.365	27.52	**139.2M**
	Once	21.33/22.46	0.570/0.637	0.368/0.346	90.04	**139.2M**
	Twice	21.33/22.68	0.581/0.661	0.344/0.321	**153.1**	**139.2M**
mip-NeRF 360 ($c = 1024$)	None	20.83/21.63	0.516/0.566	0.436/0.419	6.88	9.9M
	Once	20.89/21.85	0.523/0.580	0.427/0.409	23.24	9.9M
	Twice	20.80/21.86	0.519/0.582	0.426/0.408	**39.36**	9.9M
Ours ($c = 1024$)	None	20.89/21.86	0.521/0.580	0.425/0.380	7.12	10.4M

mately equal to the standard training. Moreover, we report benchmarks for two settings: the standard 250k iterations used in mip-NeRF 360 [2] and a $4\times$ longer version trained for 1000k steps. This is because using $2\times$ larger resolution training images requires $4\times$ more iterations for the same number of total epochs. All experiments are conducted on a TPU v2 accelerator with 32 cores.

4.3. Misalignment Analysis

We begin by analyzing the causes of quality degradation when scaling NeRF up to higher resolution. In particular, we show how training image misalignment affect the quality of images rendered by NeRF. To illustrate this, we perform an ablation study where we correct for misalignment using motion estimation techniques. Inspired by [50], we use motion estimation to align the training views with the geometry reconstructed by NeRF.

4.3.1 Re-generating Training/Testing Data with Iterative Alignment

Here we correct for misalignment in the dataset by using optical flow to align the input images with the geometry estimated by NeRF. To achieve this, we use a high-quality but expensive motion estimator (PWC-Net [38]) to calculate the optical flow between images rendered by NeRF and their corresponding ground truth views. However, we observe that our case partially violates the assumption of general optical flow estimators, which usually expect two sharp images as paired inputs. In practice though, the rendered images produced by NeRF are mostly blurry due to misalignment. As shown in the second column in Fig. 3, the current best NeRF variant (mip-NeRF 360 [2]) fails to generate sharp details, which hurts the optical flow result estimated by PWC-Net.

To address this issue, we propose an iterative alignment strategy, shown in Fig. 4. That is, at every alignment iteration we: 1) Train mip-NeRF 360 using the original ground truth images, and render output images for each training view. 2) Next, we calculate the flow from the rendered views to ground truth views using PWC-Net. Although the flow field might contain some minor inaccuracies due to the blurry NeRF images, PWC-Net can generally produce reasonable results based on global structures and shapes. 3) Finally, we warp the ground truth images using these estimated optical flows, building a new training set which is better aligned with the geometry estimated by NeRF. At each alignment iteration, mip-NeRF 360 can leverage better aligned data to produce sharper images. These sharper images then further improve the accuracy of estimated optical flow. And more accurate optical flow enables us to generate better aligned training images for the next round of training. We observe that it generally takes 2-3 iterations to reach the best quality.

Using this alignment strategy, we are also to also generate a new set of better aligned test images, which helps us avoid drawing biased conclusions from the original misaligned test set. Note that we use the same aligned test set for all methods, which we generated using the highest quality mip-NeRF 360 model with 4096 channels. In the following experiments we report results on both two test sets.

4.3.2 Qualitative Analysis of Misalignment Issue

In Fig. 3 we compare the intermediate visual examples from our iterative alignment strategy. First, we train a mip-NeRF 360 model with default parameters (1024 channels). This results in blurry images and the estimated optical flow contains artifacts in the distorted regions (first column). Next, we increase the mip-NeRF 360 parameters by $4\times$, which only marginally improves the visual quality of the results. We also apply our iterative alignment strategy to improve the results of this stronger model. By comparing the third and fourth columns, we see that the model trained with re-generated data recovers much sharper details compared to the ones trained on misaligned data (first two columns). This observation implies that the current best NeRF models are strongly affected by misaligned training examples.

4.3.3 Quantitative Analysis of Misalignment Issue

We next conduct a quantitative evaluation of these models, using three common metrics (PSNR, SSIM [42], and LPIPS [47]). As shown in Table. 1, mip-NeRF 360 with bigger MLPs (4096 channels) is consistently improved by better-aligned training data, with its PSNR score increasing by a large margin (**+0.62dB** on the warped test set). When it comes to the smaller model (1024 channels), the iterative alignment strategy still brings some improvement (**+0.23dB** PSNR), although it is smaller due to underparameterization.

Although this alignment strategy produces good results, it is very time-consuming, as it requires retraining mip-

Table 2. Quantitative comparison between ours and state-of-the-art methods on the "outdoor" dataset [2] at high-resolution (2560 × 1680).

Methods	Iterations	Standard Test Set			Warped Test Set			Time (hrs)	#Params
		PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓		
NeRF [24]	1×	21.44	0.474	0.665	-	-	-	4.16	1.5M
mip-NeRF [1]	1×	21.36	0.484	0.553	-	-	-	3.17	0.7M
mip-NeRF [1] w/ bigger MLPs	1×	21.90	0.566	0.447	22.44	0.605	0.433	22.71	9.0M
mip-NeRF 360 [2]	1×	23.71	0.644	0.368	24.58	0.693	0.349	6.88	9.9M
Ours	1×	23.84	0.649	0.365	24.77	0.70	0.340	7.12	10.4M
NeRF [24]	4×	21.60	0.483	0.631	-	-	-	16.64	1.5M
mip-NeRF [1]	4×	21.64	0.511	0.523	-	-	-	12.68	0.7M
mip-NeRF 360 [2]	4×	23.88	0.665	0.339	24.83	0.718	0.320	27.56	9.9M
Ours	4×	24.16	0.678	0.327	25.22	0.734	0.299	28.48	10.4M

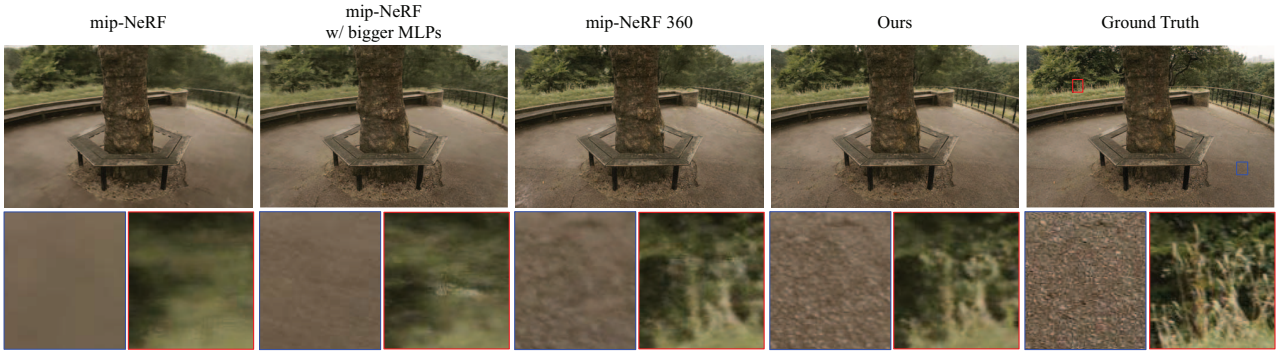

Figure 5. Qualitative comparison on a high-resolution version (2560 × 1680) of the "*treehill*" scene.

NeRF 360 from scratch and re-rendering the entire training dataset multiple times. In contrast, the proposed solution in Section 3 improves the baseline by **+0.23dB** PSNR on the warped test set, with little additional cost.

4.4. Comparing with Previous Methods

We first evaluate our method and previous works on the high-resolution (2560 × 1680) "outdoor" scenes collected by [2]. For a fair comparison, we apply the proposed AligNeRF techniques to mip-NeRF 360, and take care to not increase training time with our staged training (pre-training + fine-tuning). By default, mip-NeRF 360 is trained for 250k iterations. However, since this experiment uses higher resolution images, we also look at results where we increase the training time by 4× to keep the same number of training epochs. As shown in Table 2, NeRF [24] and mip-NeRF [1] exhibit poor performance, as they are not designed for 360 degree unbounded scenes. Increasing the parameters of mip-NeRF brings a small improvement, but makes the training time longer. mip-NeRF 360 [2] serves as a strong baseline, reaching 23.88dB and 24.83dB PSNR on the standard and warped test sets, respectively. Our proposed method outperforms the baseline methods in both groups, without introducing significant training overhead. We also include visual examples in Fig. 5 and supplemental material, where our method produces sharper and clearer textures than all other approaches.

Table 3. Quantitative comparison on the low resolution version (1280 × 840) of the "outdoor" dataset [2].

Methods	PSNR ↑	SSIM ↑	LPIPS ↓	Time (hrs)	#Params
NeRF [24]	21.46	0.458	0.515	4.16	1.5M
mip-NeRF [1]	21.69	0.471	0.505	3.17	0.7M
NeRF++ [46]	22.76	0.548	0.427	9.45	2.4M
Deep Blending [8]	21.54	0.524	0.364	-	-
Point-Based [14]	21.66	0.612	0.302	-	-
Instant-NGP [25]	22.90	0.566	0.371	0.17	51.8M
Stable View Synthesis [31]	23.01	0.662	0.253	-	-
mip-NeRF [1] w/bigger MLPs	22.98	0.625	0.348	22.71	9.0M
NeRF++ [46] w/bigger MLPs	23.80	0.642	0.338	19.88	9.0M
mip-NeRF 360 [2]	24.36	0.689	0.280	6.89	9.9M
Ours	24.55	0.703	0.263	7.12	10.4

In Table 3, we analyze how our method works on lower-resolution data by running on the "outdoor" scenes at the same resolution (1280 × 840) used by [2]. The scores for prior approaches are mostly taken directly from [2]. However, we also include the recent instant-NGP [25] method for comparison. We reached out to the authors and tuned instant-NGP [25] for large scenes by increasing the size of the hash grid (2^{21}), training batches (2^{20}) and the bounding box of the scene (32). Although Stable View Synthesis (SVS) [31] reaches the best LPIPS score, their visual results are of lower quality than other methods, as demonstrated in [2]. Among these approaches, our method demonstrates the best performance among the three metrics, even though the misalignment issue is much less severe on low-resolution images.

Table 4. Ablation study for the proposed components. All results are reported on the "*bicycle*" scene (2560 × 1680) from the "outdoor" dataset [2] with 4× longer training. We report metrics on both the standard and warped test sets respectively, split by "/".

Methods	PSNR ↑	SSIM ↑	LPIPS ↓	Time	#Params
Baseline	24.17/24.96	0.669/0.715	0.348/0.333	27.56	9.9M
+ Convolution	24.41/25.20	0.679/0.724	0.339/0.317	28.2	10.4M
+ Alignment	24.51/25.45	0.679/0.729	0.337/0.312	28.40	10.4M
+ High-frequency loss	24.55/25.51	0.683/0.734	0.331/0.306	28.48	10.4M

4.5. Ablation Study

In Table 4, we conduct an ablation study of our method on the "*bicycle*" scene (with 2560 × 1680 resolution) from the "outdoor" dataset [2]. We first train a baseline mip-NeRF 360 model for 1000k iterations. Next, we train another mip-NeRF 360 model for 600k iterations, and apply our convolution-augmented architecture on top of it. We further fine-tune this new architecture with 200k iterations, using our patch-wise sampling strategy. This makes the convolution augmented model reach much higher quality (**+0.24dB/0.24dB** on standard/warped test set), without significantly increasing the training time. Meanwhile, if we additionally add the alignment-aware loss together to this convolution-augmented architecture, it further improves quality. Especially, we observe that the gain on the warped test set (**+0.25dB** PSNR) is larger than the standard test set (**+0.10dB** PSNR) after applying the proposed alignment-aware training, demonstrating its particular improvement in misaligned parts of the scene. Finally, the proposed high-frequency loss improves the latest results by a small margin. In contrast to previous perceptual losses [12] which tend to improve the LPIPS score at the expense of other metrics, our loss improves all three metrics.

4.6. Evaluating Pose-free NeRFs on Non-still Scenes

One may wonder if the nonalignment issue can be fixed by jointly optimizing camera poses, e.g., using bundle-adjustment NeRF [17]. While this strategy indeed help the static scenes, we argue that the the subtle movement of objects may even hurt its performance, as their pose optimization requires object to be static for feature matching. To demonstrate it, we conduct a toy experiment to measure the performance of pose-free NeRF-like [17] models on non-still scenes when subtle movement is included.

To simulate this phenomenon, we adopt Blender software to render a new set of "*lego*" examples with 100 training views and 200 testing views, where its arm is randomly set to be two different conditions with 0.5 probability. Thus we are able to manually create misalignment. We show the visual examples in the first row of Fig. 6, where the difference of two conditions indicates the misaligned region. We render a higher resolution (1200 × 1200) to fit our high-fidelity setting.

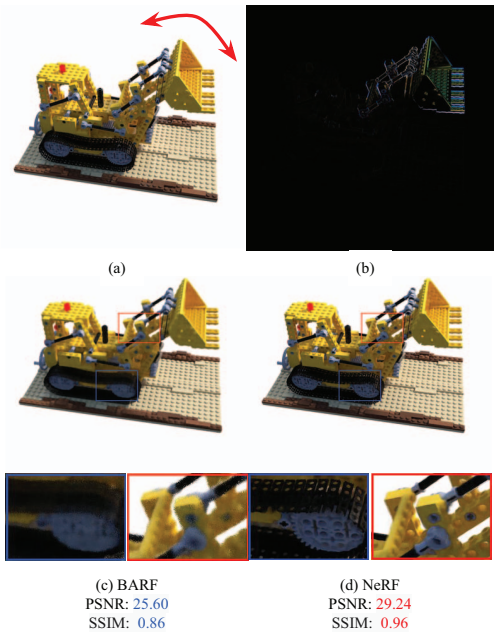

Figure 6. Visual comparison of BARF [17] and NeRF [24] on moving scene of lego. (a) Illustration of "*lego*" with moving arm. (b) Visualization of the subtle motion, calculated by the **difference** of views with two conditions. (c) Visual examples rendered by BARF. (d) Visual examples rendered by NeRF.

We evaluate BARF [17] model on this new dataset, where its initial poses are perturbed from ground truth poses. Meanwhile, we train the vanilla NeRF [24] for comparison. As shown in Fig. 6, NeRF produce sharp details on the aligned region and blurry textures on the misaligned region, where BARF renders blurry results on both two regions. The quantitative and qualitative experiments demonstrate that jointly optimizing NeRF and camera poses may face severe issues when moving objects are included.

5. Conclusion

In this work, we conducted a pilot study on training neural radiance fields on high-resolution data. We presented AligNeRF, an effective alignment-aware training strategy that improves NeRF's performance. Additionally, we also quantitatively and qualitatively analyze the performance degradation brought by misaligned data, by re-generating aligned data using motion estimation. This analysis further helps us to understand the current bottleneck of scaling NeRF to higher resolutions. we still observe that NeRF can be further improved by vastly increasing the number of parameters and by further increasing the training time. We will investigate how to close this gap in the future.

Acknowledgement

We thank Deqing Sun and Charles Herrmann for providing the codebase and pretrained model for PWC-Net, as well as for constructive discussions.

References

[1] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021.

[2] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. *arXiv preprint arXiv:2111.12077*, 2021.

[3] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14124–14133, 2021.

[4] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14124–14133, 2021.

[5] Shin-Fang Chng, Sameera Ramasinghe, Jamie Sherrah, and Simon Lucey. Garf: Gaussian activated radiance fields for high fidelity reconstruction and pose estimation. *arXiv e-prints*, pages arXiv–2204, 2022.

[6] Zihang Dai, Hanxiao Liu, Quoc V Le, and Mingxing Tan. Coatnet: Marrying convolution and attention for all data sizes. *Advances in Neural Information Processing Systems*, 34:3965–3977, 2021.

[7] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. *arXiv preprint arXiv:2107.02791*, 2021.

[8] Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. Deep blending for free-viewpoint image-based rendering. *ACM Transactions on Graphics (TOG)*, 37(6):1–15, 2018.

[9] Peter Hedman, Pratul P Srinivasan, Ben Mildenhall, Jonathan T Barron, and Paul Debevec. Baking neural radiance fields for real-time view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5875–5884, 2021.

[10] Chima Jude Iheaturu, Emmanuel Gbenga Ayodele, and Chukwuma John Okolie. An assessment of the accuracy of structure-from-motion (sfm) photogrammetry for 3d terrain mapping. *Geomatics, Landmanagement and Landscape*, 2020.

[11] Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5885–5894, 2021.

[12] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

[13] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[14] Georgios Kopanas, Julien Philip, Thomas Leimkühler, and George Drettakis. Point-based neural rendering with per-view optimization. In *Computer Graphics Forum*, volume 40, pages 29–43. Wiley Online Library, 2021.

[15] Zhengfei Kuang, Kyle Olszewski, Menglei Chai, Zeng Huang, Panos Achlioptas, and Sergey Tulyakov. Neroic: Neural rendering of objects from online image collections. *arXiv preprint arXiv:2201.02533*, 2022.

[16] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017.

[17] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5741–5751, 2021.

[18] Philipp Lindenberger, Paul-Edouard Sarlin, Viktor Larsson, and Marc Pollefeys. Pixel-Perfect Structure-from-Motion with Featuremetric Refinement. In *ICCV*, 2021.

[19] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *Advances in Neural Information Processing Systems*, 33:15651–15663, 2020.

[20] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7210–7219, 2021.

[21] Nelson Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):99–108, 1995.

[22] Roey Mechrez, Itamar Talmi, and Lihi Zelnik-Manor. The contextual loss for image transformation with non-aligned data. In *Proceedings of the European conference on computer vision (ECCV)*, pages 768–783, 2018.

[23] Quan Meng, Anpei Chen, Haimin Luo, Minye Wu, Hao Su, Lan Xu, Xuming He, and Jingyi Yu. Gnerf: Gan-based neural radiance field without posed camera. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6351–6361, 2021.

[24] 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*, pages 405–421. Springer, 2020.

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

[26] Michael Niemeyer, Jonathan T Barron, Ben Mildenhall, Mehdi SM Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. *arXiv preprint arXiv:2112.00724*, 2021.

[27] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11453–11464, 2021.

[28] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International Conference on Machine Learning*, pages 5301–5310. PMLR, 2019.

[29] Vincent Raoult, Sarah Reid-Anderson, Andreas Ferri, and Jane E Williamson. How reliable is structure from motion (sfm) over time and between observers? a case study using coral reef bommies. *Remote Sensing*, 9(7):740, 2017.

[30] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14335–14345, 2021.

[31] Gernot Riegler and Vladlen Koltun. Stable view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12216–12225, 2021.

[32] Barbara Roessle, Jonathan T Barron, Ben Mildenhall, Pratul P Srinivasan, and Matthias Nießner. Dense depth priors for neural radiance fields from sparse input views. *arXiv preprint arXiv:2112.03288*, 2021.

[33] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[34] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. *Advances in Neural Information Processing Systems*, 33:20154–20166, 2020.

[35] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[36] 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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7495–7504, 2021.

[37] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. *arXiv preprint arXiv:2111.11215*, 2021.

[38] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8934–8943, 2018.

[39] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, pages 402–419. Springer, 2020.

[40] Chen Wang, Xian Wu, Yuan-Chen Guo, Song-Hai Zhang, Yu-Wing Tai, and Shi-Min Hu. Nerf-sr: High-quality neural radiance fields using super-sampling. *arXiv preprint arXiv:2112.01759*, 2021.

[41] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2021.

[42] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[43] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. Nerf–: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*, 2021.

[44] Dejia Xu, Yifan Jiang, Peihao Wang, Zhiwen Fan, Humphrey Shi, and Zhangyang Wang. Sinnerf: Training neural radiance fields on complex scenes from a single image. *arXiv preprint arXiv:2204.00928*, 2022.

[45] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2021.

[46] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020.

[47] 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.

[48] Xuaner Zhang, Qifeng Chen, Ren Ng, and Vladlen Koltun. Zoom to learn, learn to zoom. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3762–3770, 2019.

[49] Xiuming Zhang, Pratul P Srinivasan, Boyang Deng, Paul Debevec, William T Freeman, and Jonathan T Barron. Nerfactor: Neural factorization of shape and reflectance under an unknown illumination. *ACM Transactions on Graphics (TOG)*, 40(6):1–18, 2021.

[50] Qian-Yi Zhou and Vladlen Koltun. Color map optimization for 3d reconstruction with consumer depth cameras. *ACM Transactions on Graphics (TOG)*, 33(4), 2014.