

## BAD-NeRF: Bundle Adjusted Deblur Neural Radiance Fields

Peng Wang<sup>1,2</sup> Lingzhe Zhao<sup>2</sup> Ruijie Ma<sup>2</sup> Peidong Liu<sup>2†</sup>  
<sup>1</sup>Zhejiang University <sup>2</sup>Westlake University  
 {wangpeng, zhaolingzhe, maruijie, liupeidong}@westlake.edu.cn

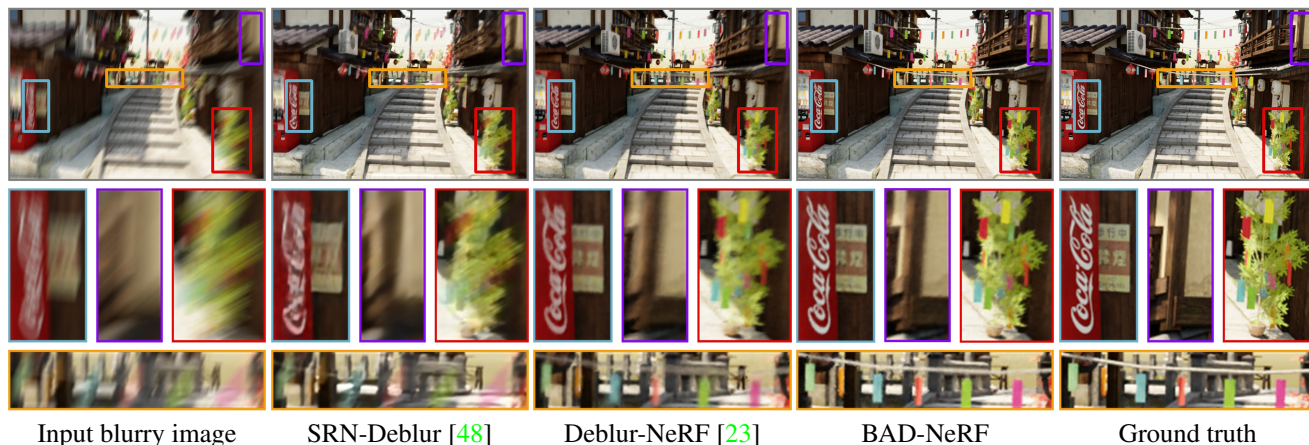


Figure 1. Given a set of severe motion blurred images, our bundle adjusted deblur NeRF (BAD-NeRF) jointly learns the neural radiance fields and recovers the camera motion trajectories within exposure time. It synthesizes novel images of higher quality than prior works.

### Abstract

*Neural Radiance Fields (NeRF) have received considerable attention recently, due to its impressive capability in photo-realistic 3D reconstruction and novel view synthesis, given a set of posed camera images. Earlier work usually assumes the input images are of good quality. However, image degradation (e.g. image motion blur in low-light conditions) can easily happen in real-world scenarios, which would further affect the rendering quality of NeRF. In this paper, we present a novel bundle adjusted deblur Neural Radiance Fields (BAD-NeRF), which can be robust to severe motion blurred images and inaccurate camera poses. Our approach models the physical image formation process of a motion blurred image, and jointly learns the parameters of NeRF and recovers the camera motion trajectories during exposure time. In experiments, we show that by directly modeling the real physical image formation process, BAD-NeRF achieves superior performance over prior works on both synthetic and real datasets. Code and data are available at <https://github.com/WU-CVGL/BAD-NeRF>.*

### 1. Introduction

Acquiring accurate 3D scene geometry and appearance from a set of 2D images has been a long standing problem

in computer vision. As a fundamental block for many vision applications, such as novel view image synthesis and robotic navigation, great progress has been made over the last decades. Classic approaches usually represent the 3D scene explicitly, in the form of 3D point cloud [8, 52], triangular mesh [4, 5, 10] or volumetric grid [31, 45]. Recent advancements in implicit 3D representation by using a deep neural network, such as Neural Radiance Fields (NeRF) [27], have enabled photo-realistic 3D reconstruction and novel view image synthesis, given well posed multi-view images.

NeRF takes a 5D vector (i.e. for spatial location and viewing direction of the sampled 3D point) as input and predicts its radiance and volume density via a multilayer perceptron. The corresponding pixel intensity or depth can then be computed by differentiable volume rendering [19, 25]. While many methods have been proposed to further improve NeRF’s performance, such as rendering efficiency [11, 28], training with inaccurate poses [20] etc., limited work has been proposed to address the issue of training with motion blurred images. Motion blur is one of the most common artifacts that degrades images in practical application scenarios. It usually occurs in low-light conditions where longer exposure times are necessary. Motion blurred images would bring two main challenges to existing NeRF training pipeline: a) NeRF usually assumes the rendered

<sup>†</sup>Corresponding author.

image is sharp (i.e. infinitesimal exposure time), motion blurred image thus violates this assumption; b) accurate camera poses are usually required to train NeRF, however, it is difficult to obtain accurate poses from blurred images only, since each of them usually encodes information of the motion trajectory during exposure time. On the other hand, it is also challenging itself to recover accurate poses (e.g., via COLMAP [41]) from a set of motion blurred images, due to the difficulties of detecting and matching salient keypoints. Combining both factors would thus further degrade NeRF’s performance if it is trained with motion blurred images.

In order to address those challenges, we propose to integrate the real physical image formation process of a motion blurred image into the training of NeRF. We also use a linear motion model in the  $SE(3)$  space to represent the camera motion trajectory within exposure time. During the training stage, both the network weights of NeRF and the camera motion trajectories are estimated jointly. In particular, we represent the motion trajectory of each image with both poses at the start and end of the exposure time respectively. The intermediate camera poses within exposure time can be linearly interpolated in the  $SE(3)$  space. This assumption holds in general since the exposure time is typically small. We can then follow the real physical image formation model of a motion blurred image to synthesize the blurry images. In particular, a sequence of sharp images along the motion trajectory within exposure time can be rendered from NeRF. The corresponding motion blurred image can then be synthesized by averaging those virtual sharp images. Both NeRF and the camera motion trajectories are estimated by minimizing the difference between the synthesized blurred images and the real blurred images. We refer this modified model as BAD-NeRF, i.e. bundle adjusted deblur NeRF.

We evaluate BAD-NeRF with both synthetic and real datasets. The experimental results demonstrate that BAD-NeRF achieves superior performance compared to prior state of the art works (e.g. as shown in Fig. 1), by explicitly modeling the image formation process of the motion blurred image. In summary, our **contributions** are as follows:

- We present a photo-metric bundle adjustment formulation for motion blurred images under the framework of NeRF, which can be potentially integrated with other vision pipelines (e.g. a motion blur aware camera pose tracker [21]) in future.
- We show how this formulation can be used to acquire high quality 3D scene representation from a set of motion blurred images.
- We experimentally validate that our approach is able to deblur severe motion blurred images and synthesize high quality novel view images.

## 2. Related Work

We review two main areas of related works: neural radiance fields, and image deblurring.

**Neural Radiance Field.** NeRF demonstrates impressive novel view image synthesis performance and 3D scene representation capability [27]. Many variants of NeRF have been proposed recently. For example, [11, 17, 37, 39, 40, 53, 56] proposed methods to improve the rendering efficiency of NeRF, such that it can render images in real-time. [13, 20, 23, 24, 26] proposed to improve the training performance of NeRF with inaccurate posed images or images captured under challenging conditions. There are also methods that extend NeRF for large scale scene representations [47, 49, 54] and non-rigid object reconstructions (e.g. human body) [1, 9, 34–36, 38, 40, 51]. NeRF is also recently being used for 3D aware image generation models [6, 12, 22, 30, 32, 50].

We will mainly detailed review those methods which are the closest to our work in this section. BARF proposed to optimize camera poses of input images as additional variables together with parameters of NeRF [20]. They propose to gradually apply the positional encoding to better train the network as well as the camera poses. A concurrent work from Jeong et al. [13] also proposed to learn the scene representation and camera parameters (i.e. both extrinsic and intrinsic parameters) jointly. Different from BARF [20] and the work from [13], which estimate the camera pose at a particular timestamp, our work proposes to optimize the camera motion trajectory within exposure time. Deblur-NeRF aims to train NeRF from a set of motion blurred images [23]. They obtain the camera poses from either ground truth or COLMAP [41], and fix them during training stage. Severe motion blurred images challenge the pose estimations (e.g. from COLMAP), which would thus further degrade the training performance of NeRF. Instead of relying heavily on the accurately posed images, our method estimates camera motion trajectories together with NeRF parameters jointly. The resulting pipeline is thus robust to inaccurate initialized camera poses due to severe motion blur.

**Image Deblurring.** Existing techniques to solve motion deblurring problem can be generally classified into two main categories: the first type of approach formulates the problem as an optimization problem, where the latent sharp image and the blur kernel are optimized using gradient descent during inference [2, 7, 15, 18, 33, 43, 55]. Another type of approaches phrases the task as an end-to-end learning problem. Building upon the recent advances of deep convolution neural networks, state-of-the-art results have been obtained for both single image deblurring [16, 29, 48] and video deblurring [46].

The closest work to ours is from Park et al. [33], which jointly recovers the camera poses, dense depth maps,

and latent sharp images from a set of multi-view motion blurred images. They formulate the problem under the classic optimization framework, and aim to maximize both the self-view photo-consistency and cross-view photo-consistencies. Instead of representing the 3D scene as a set of multi-view dense depth maps and latent sharp images, we represent it with NeRF implicitly, which can better preserve the multi-view consistency and enable novel view image synthesis.

### 3. Method

In this section, we present the details of our bundle adjusted deblur NeRF (BAD-NeRF). BAD-NeRF learns both the 3D scene representation and recovers the camera motion trajectories, given a set of motion blurred images as shown in Fig. 2. We follow the real physical image formation process of a motion blurred image to synthesize blurry images from NeRF. Both NeRF and the motion trajectories are estimated by maximizing the photo-metric consistency between the synthesized blurry images and the real blurry images. We will detail each component as follows.

#### 3.1. Neural Radiance Fields

We follow the general architecture of NeRF [27] to represent the 3D scene implicitly using two Multi-layer Perceptrons (MLP). To query the pixel intensity  $\mathbf{I}(\mathbf{x})$  at pixel location  $\mathbf{x}$  for a particular image with pose  $\mathbf{T}_c^w$ , we can shoot a ray into the 3D space. The ray is defined by connecting the camera center and the corresponding pixel. We can then compute the pixel intensity by volume rendering along the ray [25]. This process can be formally described as follows.

Given a particular 3D point  $\mathbf{X}^w$  with depth  $\lambda$  along the ray, we can obtain its coordinate defined in the world coordinate frame as:

$$\mathbf{d}^c = \mathbf{K}^{-1} \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix}, \quad (1)$$

$$\mathbf{X}^w = \mathbf{T}_c^w \cdot \lambda \mathbf{d}^c, \quad (2)$$

where  $\mathbf{x}$  is the pixel location,  $\mathbf{K}$  is the camera intrinsic parameters by assuming a simple pinhole camera model,  $\mathbf{d}^c$  is the ray direction defined in the camera coordinate frame,  $\mathbf{T}_c^w$  is the camera pose defined from camera frame to world frame. We can then query NeRF for its corresponding view-depend color  $c$  and volume density  $\sigma$  using MLPs:

$$(c, \sigma) = \mathbf{F}_\theta(\mathbf{X}^w, \mathbf{R}_c^w \cdot \mathbf{d}^c), \quad (3)$$

where  $\mathbf{F}_\theta$  is the Multi-layer Perceptrons parameterized with learnable parameters  $\theta$ ,  $\mathbf{R}_c^w$  is the rotation matrix which transforms the viewing direction  $\mathbf{d}^c$  from camera coordinate frame to world coordinate frame. Following Mildenhall et

al. [27], which showed that coordinate-based approaches struggle with learning details from low-dimensional inputs, we also use their proposed Fourier embedding  $\gamma(\mathbf{X})$  representation of a 3D point  $\mathbf{X}$  and  $\gamma(\mathbf{d})$  representation of the viewing direction  $\mathbf{d}$ , to map the low-dimensional inputs to high-dimensional space.

Following the definition of volume rendering [25], we can then compute the pixel intensity by sampling 3D points along the ray as follows:

$$\mathbf{I}(\mathbf{x}) = \sum_{i=1}^n T_i (1 - \exp(-\sigma_i \delta_i)) c_i, \quad (4)$$

where  $n$  is the number of sampled 3D points along the ray, both  $c_i$  and  $\sigma_i$  are the predicted color and volume density of the  $i^{th}$  sampled 3D point via  $\mathbf{F}_\theta$ ,  $\delta_i$  is the distance between the  $i^{th}$  and  $(i+1)^{th}$  sampled point,  $T_i$  is the transmittance factor which represents the probability that the ray does not hit any particle until the  $i^{th}$  sampled point.  $T_i$  can be formally defined as:

$$T_i = \exp\left(-\sum_{k=1}^{i-1} \sigma_k \delta_k\right), \quad (5)$$

where  $\sigma_k$  is the predicted volume density for  $k^{th}$  point by  $\mathbf{F}_\theta$ ,  $\delta_k$  is the corresponding distance between neighboring points.

The above derivations show that the rendered pixel intensity  $\mathbf{I}(\mathbf{x})$  is a function of the MLPs with learnable parameters  $\theta$ , as well as the corresponding camera pose  $\mathbf{T}_c^w$ . It can also be derived that  $\mathbf{I}(\mathbf{x})$  is differentiable with respect to both  $\theta$  and  $\mathbf{T}_c^w$ , which lays the foundations for our bundle adjustment formulation with a set of motion blurred images.

#### 3.2. Motion Blur Image Formation Model

The physical image formation process refers to a digital camera collecting photons during the exposure time and converting them into measurable electric charges. The mathematical modeling of this process involves integrating over a set of virtual sharp images:

$$\mathbf{B}(\mathbf{x}) = \phi \int_0^\tau \mathbf{I}_t(\mathbf{x}) dt, \quad (6)$$

where  $\mathbf{B}(\mathbf{x}) \in \mathbb{R}^{W \times H \times 3}$  is the captured image,  $W$  and  $H$  are the width and height of the image respectively,  $\mathbf{x} \in \mathbb{R}^2$  represents the pixel location,  $\phi$  is a normalization factor,  $\tau$  is the camera exposure time,  $\mathbf{I}_t(\mathbf{x}) \in \mathbb{R}^{W \times H \times 3}$  is the virtual sharp image captured at timestamp  $t$  within the exposure time. A blurred image  $\mathbf{B}(\mathbf{x})$  caused by camera motion during the exposure time, is formed by different virtual images  $\mathbf{I}_t(\mathbf{x})$  for each  $t$ . The model can be discretely approximated as

$$\mathbf{B}(\mathbf{x}) \approx \frac{1}{n} \sum_{i=0}^{n-1} \mathbf{I}_i(\mathbf{x}), \quad (7)$$

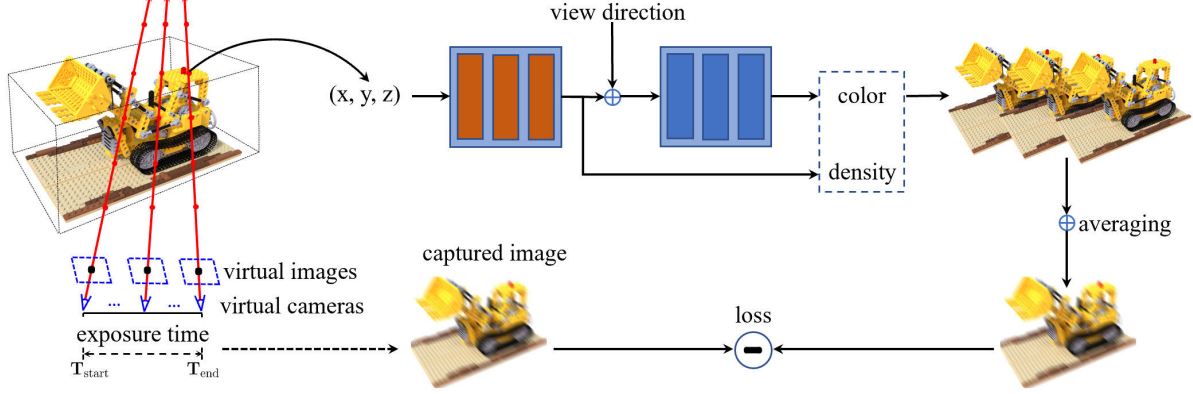


Figure 2. **The pipeline of BAD-NeRF.** Given a set of motion blurred images, we train NeRF to learn the 3D scene representation. Different from prior works, which usually model the camera pose at a fixed timestamp, we represent the motion trajectory of each image with both poses at the start and end of the exposure time respectively. Intermediate virtual camera poses can then be linearly interpolated in  $\mathbf{SE3}$  space. We then follow the standard neural rendering procedures of NeRF to synthesize those virtual sharp images within exposure time. The blurry image can then be synthesized by averaging those virtual images, which obeys the real physical image formation process of a motion blurred image. The whole network and both the start and end poses are jointly estimated by minimizing the photo-metric loss between the synthesized and real blurry images.

where  $n$  is the number of discrete samples.

The degree of motion blur in an image thus depends on the camera motion during the exposure time. For example, a fast-moving camera causes little relative motion for shorter exposure time, whereas a slow-moving camera leads to a motion blurred image for long exposure time (e.g. in low light conditions). It can be further derived that  $\mathbf{B}(\mathbf{x})$  is differentiable with respect to each of virtual sharp images  $\mathbf{I}_i(\mathbf{x})$ .

### 3.3. Camera Motion Trajectory Modeling

As derived in Eq. (7), we need to model the corresponding poses of each latent sharp image within exposure time, so that we can render them from NeRF (i.e.  $\mathbf{F}_\theta$ ). We approximate the camera motion with a linear model during exposure time which is usually small (e.g.  $\leq 200$  ms). Specifically, two camera poses are parameterized, one at the beginning of the exposure  $\mathbf{T}_{\text{start}} \in \mathbf{SE}(3)$  and one at the end  $\mathbf{T}_{\text{end}} \in \mathbf{SE}(3)$ . Between these two poses, we linearly interpolate poses in the Lie-algebra of  $\mathbf{SE}(3)$ . The virtual camera pose at time  $t \in [0, \tau]$  can thus be represented as

$$\mathbf{T}_t = \mathbf{T}_{\text{start}} \cdot \exp\left(\frac{t}{\tau} \cdot \log(\mathbf{T}_{\text{start}}^{-1} \cdot \mathbf{T}_{\text{end}})\right), \quad (8)$$

where  $\tau$  is the exposure time.  $\frac{t}{\tau}$  can be further derived as  $\frac{i}{n-1}$  for the  $i^{\text{th}}$  sampled virtual sharp image (i.e. with pose as  $\mathbf{T}_i$ ), when there are  $n$  images being sampled in total. It can be derived that  $\mathbf{T}_i$  is differentiable with respect to both  $\mathbf{T}_{\text{start}}$  and  $\mathbf{T}_{\text{end}}$ . For more details on the interpolation and derivations of the related Jacobian, please refer to prior work from Liu et al. [21]. The goal of BAD-NeRF is now to estimate both  $\mathbf{T}_{\text{start}}$  and  $\mathbf{T}_{\text{end}}$  for each frame, as well as the learnable parameters of  $\mathbf{F}_\theta$ .

Besides the linear interpolation approach, we also explore the trajectory representation with a higher order spline (i.e. cubic B-Spline), which can represent more complex camera motions. Since the exposure time is usually relatively short, we find that a linear interpolation can already deliver satisfying performance from the experimental results. More details on the cubic B-Spline formulation can be found in our supplementary material.

### 3.4. Loss Function

Given a set of  $K$  motion blurred images, we can then estimate the learnable parameters  $\theta$  of NeRF as well as the camera motion trajectories for each image (i.e.  $\mathbf{T}_{\text{start}}$  and  $\mathbf{T}_{\text{end}}$ ) by minimizing the photo-metric loss:

$$\mathcal{L} = \sum_{k=0}^{K-1} \|\mathbf{B}_k(\mathbf{x}) - \mathbf{B}_k^{gt}(\mathbf{x})\|, \quad (9)$$

where  $\mathbf{B}_k(\mathbf{x})$  is the  $k^{\text{th}}$  blurry image synthesized from NeRF by following the above image formation model,  $\mathbf{B}_k^{gt}(\mathbf{x})$  is the corresponding real captured blurry image.

To optimize the learnable parameter  $\theta$ ,  $\mathbf{T}_{\text{start}}$  and  $\mathbf{T}_{\text{end}}$  for each image, we need to have the corresponding Jacobians:

$$\frac{\partial \mathcal{L}}{\partial \theta} = \sum_{k=0}^{K-1} \frac{\partial \mathcal{L}}{\partial \mathbf{B}_k(\mathbf{x})} \cdot \frac{1}{n} \sum_{i=0}^{n-1} \frac{\partial \mathbf{B}_k(\mathbf{x})}{\partial \mathbf{I}_i(\mathbf{x})} \frac{\partial \mathbf{I}_i(\mathbf{x})}{\partial \theta}, \quad (10)$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{T}_{\text{start}}} = \sum_{k=0}^{K-1} \frac{\partial \mathcal{L}}{\partial \mathbf{B}_k(\mathbf{x})} \cdot \frac{1}{n} \sum_{i=0}^{n-1} \frac{\partial \mathbf{B}_k(\mathbf{x})}{\partial \mathbf{I}_i(\mathbf{x})} \frac{\partial \mathbf{I}_i(\mathbf{x})}{\partial \mathbf{T}_{\text{start}}}, \quad (11)$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{T}_{\text{end}}} = \sum_{k=0}^{K-1} \frac{\partial \mathcal{L}}{\partial \mathbf{B}_k(\mathbf{x})} \cdot \frac{1}{n} \sum_{i=0}^{n-1} \frac{\partial \mathbf{B}_k(\mathbf{x})}{\partial \mathbf{I}_i(\mathbf{x})} \frac{\partial \mathbf{I}_i(\mathbf{x})}{\partial \mathbf{T}_{\text{end}}}. \quad (12)$$



We parameterize both  $\mathbf{T}_{\text{start}}$  and  $\mathbf{T}_{\text{end}}$  with their corresponding Lie algebras of  $\mathbf{SE}(3)$ , which can be represented by a 6D vector respectively.

## 4. Experiments

### 4.1. Experimental details

**Benchmark datasets.** To evaluate the performance of our network, we use both the synthetic datasets and real datasets from prior works, i.e. the datasets from Deblur-NeRF [23] and MBA-VO [21]. The synthetic dataset from Deblur-NeRF [23] is synthesized by using Blender [3]. The datasets are generated from 5 virtual scenes, assuming the camera motion is in constant velocity within exposure time. Both the anchor camera poses and camera velocities (i.e. in 6 DoFs) within exposure time are randomly sampled, i.e. they are not sampled along a continuous motion trajectory. The blurry image is then generated by averaging the sampled virtual images within the exposure time for each camera. To synthesize more realistic blurry images, we increase the number of virtual images to 51 from 10 and keep the other settings fixed. They also capture a real blurry dataset by deliberately shaking a handheld camera.

To investigate the performance of our method more thoroughly, we also evaluate our method on a dataset for motion blur aware visual odometry benchmark (i.e. MBA-VO [21]). The dataset contains both synthetic and real blurry images. The synthetic images are synthesized from an Unreal game engine and the real images are captured by a handheld camera within indoor environment. Different from the synthetic dataset from Deblur-NeRF [23], the synthetic images from MBA-VO [21] are generated based on real motion trajectories (i.e. not constant velocity) from the ETH3D dataset [42].

**Baseline methods and evaluation metrics.** We evaluate our method against several state-of-the-art learning-based deblurring methods, i.e. SRNDeblurNet [48], PVD [44], MPR [57], Deblur-NeRF [23] as well as a classic multi-view image deblurring method from Park et al. [33]. For deblurring evaluation, we synthesize images corresponding to the middle virtual camera poses (i.e. the one at the middle of the exposure time) from the trained NeRF, and evaluate its performance against the other methods. Since SRNDeblurNet [48], PVD [44] and MPR [57] are primarily designed for single image deblurring, they are not generically suitable for novel view image synthesis. We thus firstly apply them to each blurry image to get the restored images, and then input those restored images to NeRF [27] for training. For novel view image synthesis evaluation, we render novel-view images with their corresponding poses from those trained NeRF. All the methods use the estimated poses from COLMAP [41] for NeRF training. To investigate the sensitivity of Deblur-NeRF [23] to the accuracy of the cam-

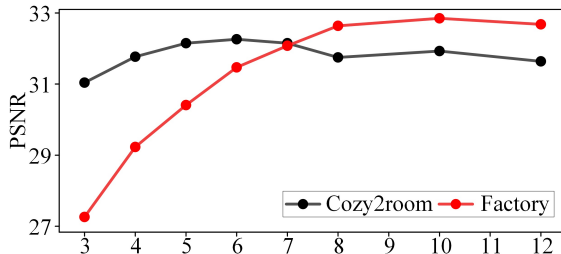


Figure 3. **The effect of the number of interpolated virtual cameras.** The results demonstrate that the performance saturates as the number increases.

	<i>Deblur-NeRF</i>			<i>MBA-VO</i>		
	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
Direct optimization	29.99	.8737	.0996	28.86	.8454	.2044
Cubic B-Spline	30.89	.8941	<b>.0884</b>	<b>29.93</b>	<b>.8643</b>	<b>.1922</b>
Linear Interpolation	<b>30.94</b>	<b>.8946</b>	.0916	29.67	.8620	.1982

Table 1. **Ablation studies on the effect of trajectory representations.** The results demonstrate that spline-based methods perform better than that of directly optimizing  $N$  poses. It also demonstrates that linear interpolation achieves comparably performance as that of cubic B-Spline, due to the short time interval within camera exposure.

era poses, we also train it with the ground-truth poses provided by Blender during dataset generation.

The quality of the rendered image is evaluated with the commonly used metrics, i.e. the PSNR, SSIM and LPIPS [58] metrics. We also evaluate the absolute trajectory error (ATE) against the method from Park et al. [33] and BARF [20] on the evaluation of the estimated camera poses. The ATE metric is commonly used for visual odometry evaluations [21].

**Implementation and training details.** We implement our method with PyTorch. We adopt the MLP network (i.e.  $\mathbf{F}_\theta$ ) structure of the original NeRF from Mildenhall et al. [27] without any modification. Both the network parameters and the camera poses (i.e.  $\mathbf{T}_{\text{start}}$  and  $\mathbf{T}_{\text{end}}$ ) are optimized with the two separate Adam optimizer [14]. The learning rate of the NeRF optimizer and pose optimizer exponentially decays from  $5 \times 10^{-4}$  to  $5 \times 10^{-5}$  and  $1 \times 10^{-3}$  to  $1 \times 10^{-5}$ . We set number of interpolated poses between  $\mathbf{T}_{\text{start}}$  and  $\mathbf{T}_{\text{end}}$  ( $n$  in Eq. 7) as 7. A total number of 128 points are sampled along each ray. We train our model for 200K iterations on an NVIDIA RTX 3090 GPU. We use COLMAP [41] to initialize the camera poses for our method.

### 4.2. Ablation study

**Number of virtual poses.** We evaluate the effect of the number of interpolated virtual cameras (i.e. for virtual image synthesis in Eq. (7)) within exposure time. We choose two sequences from the synthetic dataset of Deblur-NeRF [23] for experiment, i.e. the *cozy2room* sequence and the *factory* sequence, which represent sequences with low-level and high-level motion blur respectively. The experiments

	Cozy2room			Factory			Pool			Tanabata			Trolley			Average		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Park [33]	23.82	.7221	.2020	21.02	.5090	.4193	27.98	.7258	.2305	17.91	.4637	.4030	19.96	.5610	.3222	22.14	.5963	.3154
MPR [57]	29.90	.8862	.0915	25.07	.6994	.2409	33.28	<b>.8938</b>	.1290	22.60	.7203	.2507	26.24	.8356	.1762	27.42	.8071	.1777
PVD [44]	28.06	.8443	.1515	24.57	.6877	.3150	30.38	.8393	.1977	22.54	.6872	.3351	24.44	.7746	.2600	26.00	.7666	.2519
SRNDeblur [48]	29.47	.8759	.0950	26.54	.7604	.2404	32.94	.8847	.1045	23.20	.7274	.2438	25.36	.8119	.1618	27.50	.8121	.1691
DeblurNeRF [23]	25.96	.7979	.1024	23.21	.6487	.2618	31.21	.8518	.1382	22.46	.6946	.2455	24.94	.7923	.1766	25.56	.7571	.1849
DeblurNeRF* [23]	30.26	.8933	.0791	26.40	.7991	.2191	32.30	.8755	.1345	24.56	.7749	.2166	26.24	.8254	.1671	27.95	.8336	.1633
BAD-NeRF (ours)	<b>32.15</b>	<b>.9170</b>	<b>.0547</b>	<b>32.08</b>	<b>.9105</b>	<b>.1218</b>	<b>33.36</b>	.8912	<b>.0802</b>	<b>27.88</b>	<b>.8642</b>	<b>.1179</b>	<b>29.25</b>	<b>.8892</b>	<b>.0833</b>	<b>30.94</b>	<b>.8946</b>	<b>.0916</b>

Table 2. **Quantitative deblurring comparisons on the synthetic dataset of Deblur-NeRF [23]**. Note that DeblurNeRF\* is trained with the ground-truth poses, while the other one is trained with the estimated poses by COLMAP [41]. The experimental results demonstrate that our method achieves the best performance over prior methods. It also demonstrates that the DeblurNeRF is sensitive to the accuracy of the provided camera poses.

	Cozy2room			Factory			Pool			Tanabata			Trolley			Average		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
NeRF+Park	23.44	.7024	.2634	20.83	.5041	.4133	28.69	.7512	.2865	19.29	.5317	.4342	20.73	.6012	.3804	22.60	.6181	.3556
NeRF+MPR	27.17	.8334	.1196	23.78	.6375	.2499	31.15	.8402	.1837	21.24	.6914	.2801	26.14	.8154	.1979	25.90	.7636	.2062
NeRF+PVD	26.26	.7977	.1764	23.88	.6450	.3074	29.02	.7792	.2287	21.03	.6566	.3406	23.96	.7502	.2772	24.83	.7257	.2661
NeRF+SRNDeblur	27.27	.8321	.1261	26.19	.7494	.2274	31.09	.8375	.1770	21.46	.6943	.2839	25.01	.7883	.2077	26.20	.7803	.2044
Deblur-NeRF	26.05	.8084	.1072	25.17	.7253	.2447	30.97	.8447	.1554	21.77	.7172	.2515	24.45	.7785	.2088	25.68	.7748	.1935
Deblur-NeRF*	29.88	.8901	.0747	26.06	.8023	.2106	30.94	.8399	.1694	22.56	.7639	.2285	25.78	.8122	.1797	27.04	.8217	.1726
BAD-NeRF (ours)	<b>30.97</b>	<b>.9014</b>	<b>.0552</b>	<b>31.65</b>	<b>.9037</b>	<b>.1228</b>	<b>31.72</b>	<b>.8580</b>	<b>.1153</b>	<b>23.82</b>	<b>.8311</b>	<b>.1378</b>	<b>28.25</b>	<b>.8727</b>	<b>.0914</b>	<b>29.28</b>	<b>.8734</b>	<b>.1045</b>

Table 3. **Quantitative novel view synthesis comparisons on the synthetic dataset of Deblur-NeRF [23]**. The experimental results demonstrate that our method delivers state-of-the-art performance compared to prior works.

are conducted by training our network with a varying number of interpolated virtual cameras. The PSNR metrics computed and plotted in Fig. 3. The experimental results demonstrate that the number of virtual cameras does not affect much for images with low-level motion blur. As expected, more virtual cameras are required for images with high-level motion blur. By compromising the image rendering quality and the training complexity (i.e. the larger the number of virtual images, the more computational resource is required for training), we choose 7 virtual images for our experiments.

**Trajectory representations.** To evaluate the effect of different trajectory representations, we conduct three experiments: the first is based on optimizing  $N$  (i.e.  $N = 7$ ) camera poses directly, the second is based on optimizing  $\mathbf{T}_{\text{start}}$  and  $\mathbf{T}_{\text{end}}$  to represent a linear trajectory, and the last is based on a higher order spline (i.e. cubic B-Spline) which jointly optimizes 4 control knots  $\mathbf{T}_1$ ,  $\mathbf{T}_2$ ,  $\mathbf{T}_3$  and  $\mathbf{T}_4$  to represent more complex camera motions. For more detailed formulation of cubic B-Spline, please refer to our supplementary material. Since directly optimizing poses would lose the ordering information, we compute the metrics (e.g. PSNR) for all 7 poses and choose the best one for comparison. The average quantitative results on the datasets of *Deblur-NeRF* [23] (Cozy2room, Factory, Pool, Tanabata and Trolley) and *MBA-VO* [21] (ArchViz-low and ArchViz-high) are shown in Table 1. It demonstrates that directly optimizing  $N$  camera poses performs worse than spline-based methods. It also demonstrates that linear interpolation performs comparably as that of cubic B-Spline interpolation. In particular, linear interpolation achieves slightly better performance on the datasets of *Deblur-NeRF*, and cubic B-Spline performs slightly better on that of *MBA-VO*,

	ArchViz-low			ArchViz-high		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Park [33]	21.08	.6032	.3524	21.10	.5963	.4243
MPR [57]	29.60	.8757	.2103	25.04	.7711	.3576
PVD [44]	27.82	.8318	.2475	25.59	.7792	.3441
SRNDeblur [48]	30.15	.8814	.1703	27.07	.8190	.2796
Deblur-NeRF [23]	28.06	.8491	.2036	25.76	.7832	.3277
Deblur-NeRF* [23]	29.65	.8744	.1764	26.44	.8010	.3172
BAD-NeRF (ours)	<b>31.27</b>	<b>.9005</b>	<b>.1503</b>	<b>28.07</b>	<b>.8234</b>	<b>.2460</b>

Table 4. **Quantitative deblurring comparisons on the synthetic dataset of MBA-VO [21]**. The experimental results demonstrate that our method achieves the best performance even with a camera that is not moving at a constant velocity within exposure time.

compared to linear motion model. It is due to the relatively short time interval for camera exposure. Linear interpolation is already sufficient to represent the camera motion trajectory accurately within such short time interval.

### 4.3. Results

**Quantitative evaluation results.** For the subsequent experimental results, we use linear interpolation by default, unless explicitly stated. We evaluate the deblurring and novel view image synthesis performance with both the synthetic images from Deblur-NeRF [23] and MBA-VO [21]. We also evaluate the accuracy of the refined camera poses of our method against that from Park [33] and BARF [20] in terms of the ATE metric. Both Table 2 and Table 3 present the experimental results in deblurring and novel view image synthesis respectively, with the dataset from Deblur-NeRF [23]. It reveals that single image based deblurring methods, e.g. MPR [57], PVD [44] and SRNDeblur [48] fail to outperform our method, due to the limited information possessed by a single blurry image and the deblurring network struggles to restore the sharp image. The results also reveal



Figure 4. **Qualitative results of different methods with synthetic datasets.** BAD-NeRF achieves the best performance under inaccurate poses on various scenes and different levels of blur. Since DeblurNeRF does not explicitly model occlusions, it fails to render sharp edges (i.e. result in the second column) where large depth change exists.

that classic multi-view image deblurring method, i.e. the work from Park et al. [33], cannot outperform our method, thanks to the powerful representation capability of deep neural networks for our method. In the experiments, we also investigate the effect of the input camera poses accuracy to DeblurNeRF [23]. We conduct two experiments, i.e. the DeblurNeRF network is trained with ground truth poses and with that computed by COLMAP [41]. Since the images are motion blurred, the poses estimated by COLMAP are not accurate. The results reveal that the DeblurNeRF network trained with poses from COLMAP [41] performs poorly compared to that of ground truth poses. It further demonstrates that DeblurNeRF is sensitive to the accuracy of the camera poses, since they do not optimize them during training.

To better evaluate the performance of our network, we also conduct experiments with the dataset from MBA-VO [21]. The images from MBA-VO [21] are generated by using real camera motion trajectories from ETH3D dataset [42]. The motion is not at constant velocity compared to that of the dataset from DeblurNeRF [23]. The experimental results presented in Table 4 demonstrate that our network

	Cozy2room	Factory	Pool	Tanabata	Trolley
COLMAP-blur [41]	.128±.090	.148±.093	.057±.026	.103±.090	.045±.042
BARF [20]	.291±.111	.145±.088	.083±.036	.203±.091	.244±.074
BAD-NeRF (ours)	.050±.025	.033±.012	.020±.007	.016±.008	.007±.004

Table 5. **Pose estimation performance of BAD-NeRF on various blur sequences.** The results are in the absolute trajectory error metric (ATE). The COLMAP-blur represents the result of COLMAP with blurry images.

also outperforms other methods, even with a camera that is not moving in constant velocity within exposure time.

To evaluate the performance of camera pose estimation, we also tried to compare our method against the work from Park et al. [33]. However, we found that the method from Park et al. [33] hardly converges and we did not list their metrics. We therefore only present the comparisons of our method against that of COLMAP [41] and BARF [20]. The experiments are conducted on the datasets from DeblurNeRF [23]. The estimated camera poses are aligned with the ground truth poses before the absolute trajectory error metric is computed. The experimental results presented in Table 5 demonstrate that our method can recover the camera poses more accurately.



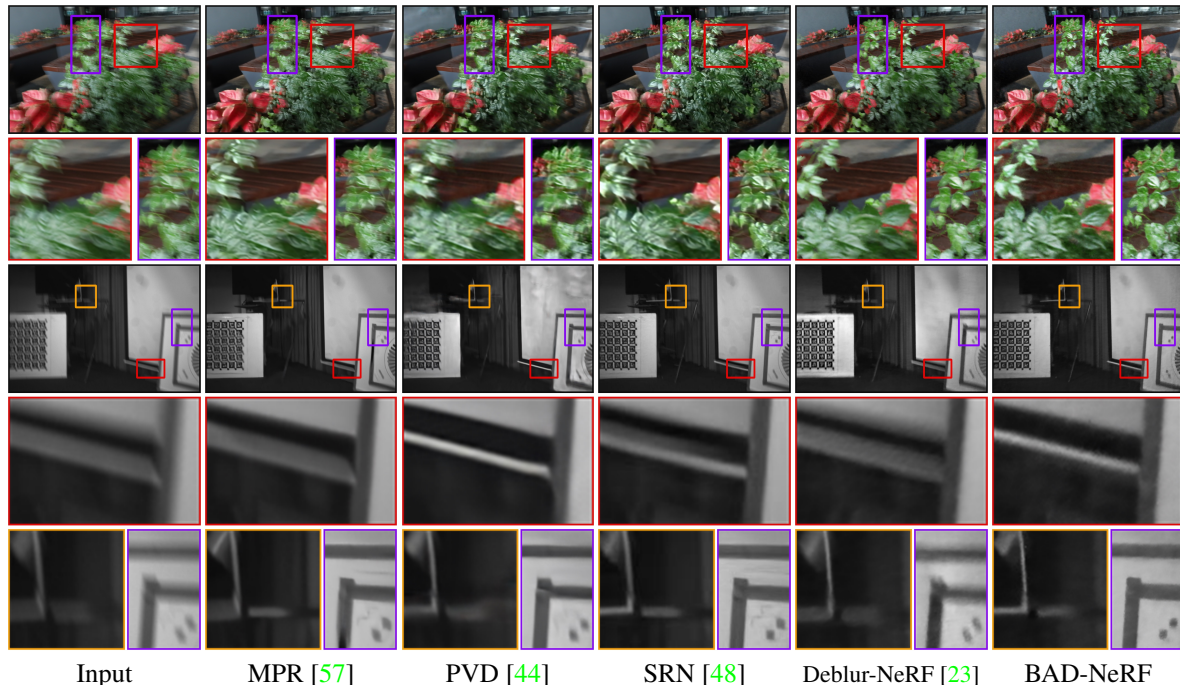


Figure 5. **Qualitative results of different methods with the real datasets.** The experimental results demonstrate that our method achieves superior performance over prior methods on the real dataset as well. Best viewed in high resolution.

**Qualitative evaluation results.** We also evaluate the qualitative performance of our method against the other methods. The experiments are conducted on both synthetic and real-world datasets. The experimental results presented in both Fig. 4 and Fig. 5 demonstrate that our method also outperforms other methods as in the quantitative evaluation section. In particular, single image deblurring networks (i.e. MPR [57], PVD [44] and SRNDeblur [48]) indeed can achieve impressive performance on some images, which are not severely blurred. However, they fail to restore the sharp image and bring in unpleasing artifacts for severely blurred images. On the contrary, our method always delivers consistent performance regardless of the level of motion blur. The reason is that our method takes advantage of multi-view images to recover a consistent 3D representation of the scene. It learns to fuse information from other views to improve the deblurring performance. Our method also delivers better results compared to DeblurNeRF [23]. One reason is that DeblurNeRF does not optimize the camera poses/motion trajectories within exposure time. They can deliver impressive results if the accurate poses are known. However, the performance would degrade if inaccurate poses are provided. Unfortunately, it is usually not trivial to recover accurate camera poses from motion blurred images, especially when the images are severely blurred. Thanks to the explicit modeling of the camera motion trajectory within exposure time, our method does not have such limitations. As shown in Table 5, our method can accurately estimate the camera poses together

with the learning of the network. Another reason is caused by the formulation of DeblurNeRF. The motion blur aware image formation model of DeblurNeRF does not model occlusions. They synthesize a blurry image by convolving the rendered image with a learned point spread function. It thus cannot accurately model the image formation process for pixels/3D points lying around an edge, which has a large change of depth. In contrast, our method follows the real physical image formation process of a motion blurred image and does not have such an issue. Fig. 4 clearly shows that DeblurNeRF fails to render sharp images around the edge of the floating box even trained with ground truth poses, caused by the occlusion problem.

## 5. Conclusion

In this paper, we propose a photometric bundle adjustment formulation for motion blurred images by using NeRF to implicitly represent the 3D scene. Our method jointly learns the 3D representation and optimizes the camera poses with blurry images and inaccurate initial poses. Extensive experimental evaluations with both real and synthetic datasets are conducted. The experimental results demonstrate that our method can effectively deblur images, render novel view images and recover the camera motion trajectories accurately within exposure time.

**Acknowledgements.** This work was supported in part by NSFC under Grant 62202389, in part by a grant from the Westlake University-Muyuan Joint Research Institute, and in part by the Westlake Education Foundation.



## References

- [1] Shahrukh Athar, Zexiang Xu, Adobe Research, Kalyan Sunkavalli, Eli Shechtman, and Zhixin Shu. RigNeRF: Fully Controllable Neural 3D Portraits. In *Computer Vision and Pattern Recognition (CVPR)*, pages 20364–20373, 2022. [2](#)
- [2] Sunghyun Cho and Seungyong Lee. Fast Motion Deblurring. *ACM Transactions on Graphics (TOG)*, 28(5), 2009. [2](#)
- [3] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. [5](#)
- [4] N. Cornelis, B. Leibe, K. Cornelis, and L. J. Van Gool. 3D urban scene modeling integrating recognition and reconstruction. *International Journal of Computer Vision (IJCV)*, 78(2-3):121–141, July 2008. [1](#)
- [5] Amaël Delaunoy and Marc Pollefeys. Photometric Bundle Adjustment for Dense Multi-view 3D Modeling. In *Computer Vision and Pattern Recognition (CVPR)*, 2014. [1](#)
- [6] Yu Deng, Jiaolong Yang, Jianfeng Xiang, and Xin Tong. GRAM: Generative Radiance Manifolds for 3D-Aware Image Generation. In *Computer Vision and Pattern Recognition (CVPR)*, volume i, 2022. [2](#)
- [7] Rob Fergus, Barun Singh, Aaron Hertzmann, Sam T. Roweis, and William T. Freeman. Removing camera shake from a single photograph. In *ACM Transactions on Graphics (TOG)*, 2006. [2](#)
- [8] Yasutaka Furukawa and Jean Ponce. Accurate, Dense, and Robust Multi-View Stereopsis. *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 32(8):1362–1376, 2010. [1](#)
- [9] Guy Gafni, Justus Thies, Michael Zollhofer, and Matthias Niesner. Dynamic Neural Radiance Fields for Monocular 4D Facial Avatar Reconstruction. In *Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2021. [2](#)
- [10] David Gallup, Marc Pollefeys, and Jan-Michael Frahm. 3D Reconstruction Using an n-Layer Heightmap. In *DAGM German Conference on Pattern Recognition (DAGM GCPR)*, 2010. [1](#)
- [11] Stephan J. Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. FastNeRF: High-Fidelity Neural Rendering at 200FPS. In *International Conference on Computer Vision (ICCV)*, 2021. [1](#), [2](#)
- [12] Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. StyleNeRF: A Style-Based 3D Aware Generator for High Resolution Image Synthesis. In *International Conference on Learning Representations (ICLR)*, pages 1–20, 2022. [2](#)
- [13] Yoonwoo Jeong, Seokjun Ahn, Christopher Choy, Animesh Anandkumar, Minsu Cho, and Jaesik Park. Self-Calibrating Neural Radiance Fields. *International Conference on Computer Vision (ICCV)*, 2021. [2](#)
- [14] Diederik P Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980*, 2014. [5](#)
- [15] Dilip Krishnan and Rob Fergus. Fast image deconvolution using Hyper-Laplacian priors. In *Neural Information Processing Systems (NIPS)*, 2009. [2](#)
- [16] Orest Kupyn, Tetiana Martyniuk, Junru Wu, and Zhangyang Wang. DeblurGAN-v2: Deblurring (Orders-of-Magnitude) Faster and Better. In *International Conference on Computer Vision (ICCV)*, 2019. [2](#)
- [17] Christoph Lassner and Michael Zollhofer. Pulsar: Efficient sphere-based neural rendering. In *Computer Vision and Pattern Recognition (CVPR)*, 2021. [2](#)
- [18] Anat Levin, Yair Weiss, Fredo Durand, and William T. Freeman. Understanding and evaluating blind deconvolution algorithm. In *Computer Vision and Pattern Recognition (CVPR)*, 2009. [2](#)
- [19] Marc Levoy. Efficient Ray Tracing of Volume Data. *ACM Transactions on Graphics (TOG)*, 9(3):245–261, 1990. [1](#)
- [20] Lin, Chen-Hsuan and Ma, Wei-Chiu and Torralba, Antonio and Lucey, Simon. BARF: Bundle-Adjusting Neural Radiance Fields. In *International Conference on Computer Vision (ICCV)*, 2021. [1](#), [2](#), [5](#), [6](#), [7](#)
- [21] Peidong Liu, Xingxing Zuo, Viktor Larsson, and Marc Pollefeys. MBA-VO: Motion Blur Aware Visual Odometry. In *International Conference on Computer Vision (ICCV)*, pages 5550–5559, 2021. [2](#), [4](#), [5](#), [6](#), [7](#)
- [22] Steven Liu, Xiuming Zhang, Zhoutong Zhang, Richard Zhang, Jun-Yan Zhu, and Bryan Russell. Editing Conditional Radiance Fields. In *International Conference on Computer Vision (ICCV)*, 2021. [2](#)
- [23] Li Ma, Xiaoyu Li, Jing Liao, Qi Zhang, Xuan Wang, Jue Wang, and Pedro V Sander. Deblur-NeRF: Neural Radiance Fields from Blurry Images. In *Computer Vision and Pattern Recognition (CVPR)*, pages 12861–12870, 2022. [1](#), [2](#), [5](#), [6](#), [7](#), [8](#)
- [24] Ricardo Martin-Brualla, Noha Radwan, Mehdi S.M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. In *Computer Vision and Pattern Recognition (CVPR)*, pages 7206–7215, 2021. [2](#)
- [25] Nelson Max. Optical Models for Direct Volume Rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):99–108, 1995. [1](#), [3](#)
- [26] Ben Mildenhall, Peter Hedman, Ricardo Martin-Brualla, Pratul P Srinivasan, and Jonathan T Barron. NeRF in the Dark: High Dynamic Range View Synthesis from Noisy Raw Images. In *Computer Vision and Pattern Recognition (CVPR)*, pages 16190–16199, 2022. [2](#)
- [27] 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 (ECCV)*. [1](#), [2](#), [3](#), [5](#)
- [28] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. *ACM Transactions on Graphics (TOG)*, 41(4):102:1–102:15, July 2022. [1](#)
- [29] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep Multi-Scale Convolutional Neural Network for Dynamic Scene Deblurring. In *Computer Vision and Pattern Recognition (CVPR)*, July 2017. [2](#)
- [30] Michael Niemeyer and Andreas Geiger. GIRAFFE: Representing Scenes as Compositional Generative Neural Feature Fields. In *Computer Vision and Pattern Recognition (CVPR)*, pages 11448–11459, 2021. [2](#)

- [31] M. Nießner, M. Zollhöfer, S. Izadi, and M. Stamminger. Real-time 3D Reconstruction at Scale using Voxel Hashing. In *ACM Transactions on Graphics (TOG)*, 2013. 1
- [32] Roy Or-El, Xuan Luo, Mengyi Shan, Eli Shechtman, Jeong Joon Park, and Ira Kemelmacher-Shlizerman. StyleSDF: High-Resolution 3D-Consistent Image and Geometry Generation. In *Computer Vision and Pattern Recognition (CVPR)*, pages 13503–13513, 2022. 2
- [33] Haesol Park and Kyoung Mu Lee. Joint Estimation of Camera Pose, Depth, Deblurring, and Super-Resolution from a Blurred Image Sequence. In *International Conference on Computer Vision (ICCV)*, pages 4613–4621, 2017. 2, 5, 6, 7
- [34] Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable Neural Radiance Fields. In *International Conference on Computer Vision (ICCV)*, 2020. 2
- [35] Sida Peng, Junting Dong, Qianqian Wang, Shangzhan Zhang, Qing Shuai, Xiaowei Zhou, and Hujun Bao. Animatable neural radiance fields for modeling dynamic human bodies. In *International Conference on Computer Vision (ICCV)*, 2021. 2
- [36] Sida Peng, Yuanqing Zhang, Yinghao Xu, Qianqian Wang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Neural Body: Implicit Neural Representations with Structured Latent Codes for Novel View Synthesis of Dynamic Humans. In *Computer Vision and Pattern Recognition (CVPR)*, pages 9050–9059, 2021. 2
- [37] Martin Píala and Ronald Clark. TerminiNeRF: Ray Termination Prediction for Efficient Neural Rendering. In *International Conference on 3D Vision (3DV)*, 2021. 2
- [38] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-NeRF: Neural Radiance Fields for Dynamic Scenes. In *Computer Vision and Pattern Recognition (CVPR)*, pages 10313–10322, 2021. 2
- [39] Daniel Rebain, Wei Jiang, Soroosh Yazdani, Ke Li, Kwang Moo Yi, and Andrea Tagliasacchi. Derf: Decomposed radiance fields. In *Computer Vision and Pattern Recognition (CVPR)*, pages 14148–14156, 2021. 2
- [40] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. KiloNeRF: Speeding up Neural Radiance Fields with Thousands of Tiny MLPs. In *International Conference on Computer Vision (ICCV)*, 2021. 2
- [41] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion Revisited. In *Computer Vision and Pattern Recognition (CVPR)*, pages 4104–4113, 2016. 2, 5, 6, 7
- [42] Thomas Schops, Torsten Sattler, and Marc Pollefeys. Bad slam: Bundle adjusted direct rgb-d slam. In *Computer Vision and Pattern Recognition (CVPR)*, pages 134–144, 2019. 5, 7
- [43] Q Shan, Jiaya Jia, and Agarwala A. High-quality Motion Deblurring from a Single Image. *ACM Transactions on Graphics (TOG)*, 27(3), 2008. 2
- [44] Hyeonseok Son, Junyong Lee, Jonghyeop Lee, Sunghyun Cho, and Seungyong Lee. Recurrent Video Deblurring with Blur-Invariant Motion Estimation and Pixel Volumes. *ACM Transactions on Graphics (TOG)*, 40(5):1–18, 2021. 5, 6, 7, 8
- [45] F. Steinbruecker, J. Sturm, and D. Cremers. Volumetric 3D Mapping in Real-Time on a CPU. In *International Conference on Robotics and Automation (ICRA)*, 2014. 1
- [46] Shuo Chen Su, Mauricio Delbracio, and Jue Wang. Deep Video Deblurring for Hand-held Cameras. In *Computer Vision and Pattern Recognition (CVPR)*, 2017. 2
- [47] Matthew Tancik, Vincent Casser, Xinchun Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P. Srinivasan, Jonathan T. Barron, and Henrik Kretzschmar. Block-NeRF: Scalable Large Scene Neural View Synthesis. In *Computer Vision and Pattern Recognition (CVPR)*, 2022. 2
- [48] Xin Tao, Hongyun Gao, Xiaoyong Shen, Jue Wang, and Jiaya Jia. Scale-recurrent network for deep image deblurring. In *Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 2, 5, 6, 7, 8
- [49] Haithem Turki, Deva Ramanan, and Mahadev Satyanarayanan. Mega-NeRF: Scalable Construction of Large-Scale NeRFs for Virtual Fly-Throughs. In *Computer Vision and Pattern Recognition (CVPR)*, pages 12922–12931, 2022. 2
- [50] Can Wang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. CLIP-NeRF: Text-and-Image Driven Manipulation of Neural Radiance Fields. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1–10, 2022. 2
- [51] Chung-Yi Weng, Brian Curless, Pratul P. Srinivasan, Jonathan T. Barron, and Ira Kemelmacher-Shlizerman. HumanNeRF: Free-viewpoint Rendering of Moving People from Monocular Video. In *Computer Vision and Pattern Recognition (CVPR)*, pages 16210–16220, 2022. 2
- [52] Thomas Whelan, Stefan Leutenegger, Renato F. Salas-Moreno, Ben Glocker, and Andrew J. Davison. ElasticFusion: Dense SLAM Without a Pose Graph. In *Robotics: Science and Systems (RSS)*, 2015. 1
- [53] Suttisak Wizadwongsa, Pakkapon Phongthawee, Jiraphon Yenphraphai, and Supasorn Suwajanakorn. NeX: Real-time View Synthesis with Neural Basis Expansion. In *Computer Vision and Pattern Recognition (CVPR)*, pages 8530–8539, 2021. 2
- [54] Yuanbo Xiangli, Linning Xu, Xingang Pan, Nanxuan Zhao, Anyi Rao, Christian Theobalt, Bo Dai, and Dahua Lin. CityNeRF: Building NeRF at City Scale. In *European Conference on Computer Vision (ECCV)*, 2022. 2
- [55] Li Xu and Jiaya Jia. Two-Phase Kernel Estimation for robust motion deblurring. In *European Conference on Computer Vision (ECCV)*, 2010. 2
- [56] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. PlenOctrees for Real-time Rendering of Neural Radiance Fields. In *International Conference on Computer Vision (ICCV)*, 2021. 2
- [57] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage Progressive Image Restoration. In *Computer Vision and Pattern Recognition (CVPR)*, pages 14821–14831, 2021. 5, 6, 7, 8
- [58] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *Computer Vision and Pattern Recognition (CVPR)*, pages 586–595, 2018. 5