

# ExBluRF: Efficient Radiance Fields for Extreme Motion Blurred Images

Dongwoo Lee<sup>1</sup>    Jeongtaek Oh<sup>2</sup>    Jaesung Rim<sup>3</sup>    Sunghyun Cho<sup>3,4</sup>    Kyoung Mu Lee<sup>1,2</sup>

<sup>1</sup>Dept. of ECE&ASRI, <sup>2</sup>IPAI, Seoul National University, Korea

<sup>3</sup>GSAI, <sup>4</sup>Dept. of CSE, POSTECH, Korea

{dongwoo.lee,ohjtgood}@snu.ac.kr {jsrim123,s.cho}@postech.ac.kr kyungmu@snu.ac.kr

## Abstract

We present ExBluRF, a novel view synthesis method for extreme motion blurred images based on efficient radiance fields optimization. Our approach consists of two main components: 6-DOF camera trajectory-based motion blur formulation and voxel-based radiance fields. From extremely blurred images, we optimize the sharp radiance fields by jointly estimating the camera trajectories that generate the blurry images. In training, multiple rays along the camera trajectory are accumulated to reconstruct single blurry color, which is equivalent to the physical motion blur operation. We minimize the photo-consistency loss on blurred image space and obtain the sharp radiance fields with camera trajectories that explain the blur of all images. The joint optimization on the blurred image space demands painfully increasing computation and resources proportional to the blur size. Our method solves this problem by replacing the MLP-based framework to low-dimensional 6-DOF camera poses and voxel-based radiance fields. Compared with the existing works, our approach restores much sharper 3D scenes from challenging motion blurred views with the order of  $10\times$  less training time and GPU memory consumption.

## 1. Introduction

Neural Radiance Fields (NeRF) have made great progress on novel view synthesis in recent years. A number of follow-up works pay attention to NeRF's [31] photo-realistic view synthesis performance, and focus on improving training [10, 51] and rendering [13, 40] speed for practical applications. However, enhancing NeRF's rendering quality from degraded multi-view images is yet to be explored extensively.

Camera motion blur is a representative degradation of images taken under low-light conditions and camera shake. Optimizing NeRF from the motion blurred images suf-

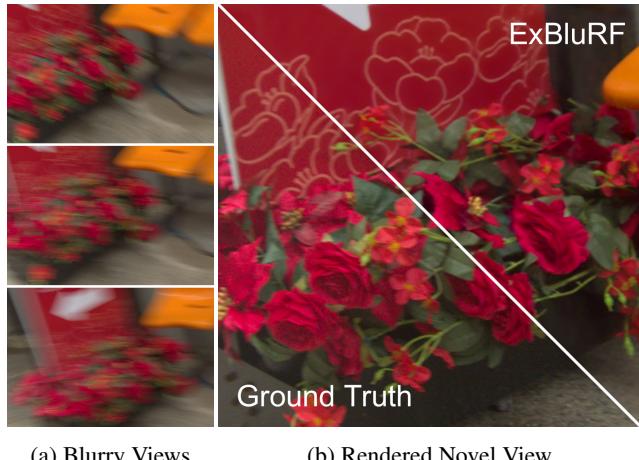


Figure 1: Given a set of extremely blurred multi-view images (a), our method restores sharp radiance fields and renders clearly deblurred novel views (b).

fers from the severe shape-radiance ambiguity [61] and produces inaccurate 3D geometry reconstruction and low-quality view synthesis of the scene.

One naive solution for this problem is to apply deep learning based 2D image deblurring [5, 59] to the input images before optimizing NeRF. However, this straightforward approach has two limitations: 1) Pre-training and fine-tuning strategy of the deep neural network is invalid, since NeRF is per-scene optimized without pairs of sharp images. 2) Independently deblurred multi-view images may yield inconsistent geometry in 3D space, and cannot be recovered with NeRF’s optimization.

The first work that considers the blurry images in NeRF's optimization is DeblurNeRF [28]. DeblurNeRF incorporates 2D pixel-wise blur kernel estimation [2] in front of NeRF's ray marching operation. The end-to-end framework of DeblurNeRF restores sharp 3D scenes from moderate motion blurred images. However, in extreme motion blur cases, the 2D pixel-wise kernel approach is difficult to converge with plausible deblurring, because the blur kernels are spatially varying severely with *6-degree of freedom*

(6-DOF) camera motions. Also, naive implementation using multi-layer perceptron (MLP) networks becomes a bottleneck of training for the extreme motion blur, since the memory consumption and the computation increase proportionally to the blur kernel size as shown in Fig. 2

In this paper, we propose a novel and efficient NeRF framework to synthesize sharp novel views from extreme motion blurred images. Inspired by [24, 36], our framework formulates the latent blur operation of each image by a trajectory of camera motion that generates blur. Like the image capturing of conventional cameras, the colors of rays are accumulated while the origin and the direction of the rays change along the estimated trajectory. These accumulated colors are optimized to minimize the photo-consistency loss with the colors of the input blurry image. Each view’s trajectory is fully constrained by blur patterns of all the pixels on the image, and the latent sharp 3D scene is optimized to satisfy all multi-view blur observations.

In addition to the proposed motion blur formulation, we adopt voxel-based radiance fields for efficient GPU memory consumption and training time. Inherently, the blurry view reconstruction in training time requires explosive computation proportional to the number of sampling along the camera trajectory. We take advantage of volumetric representation approaches [4, 11, 32, 47, 58] that are accelerated by replacing the MLP-based representation to voxel-based representation. Furthermore, our voxel-based radiance fields keep constant memory to the increasing number of sampling on camera trajectory as shown in Fig. 2.

We conduct extensive experiments to validate the proposed approach on both real and synthetic data with extreme motion blur. In particular, we propose ExBlur, which provides multi-view blurred images and sequences of sharp images simultaneously captured by a dual-camera system. The ExBlur dataset enables accurate evaluation of novel view synthesis and optimized camera trajectories on real blurred scenes. In addition, we analyze the efficiency of the proposed method on memory cost and training speed compared to the previous method [28].

To summarize, our contributions are:

- We propose the blur model formulated by 6-DOF camera motion trajectory in NeRF’s volume rendering framework that restores the sharp latent 3D scene without neural networks.
- We adopt the voxel-based radiance fields to realize efficient deblurring optimization in terms of memory consumption and training time.
- We demonstrate the high-quality deblurring and novel view synthesis of the proposed approach by our ExBlur dataset that presents blurry-sharp multi-view images with ground truth (GT) motion trajectory.

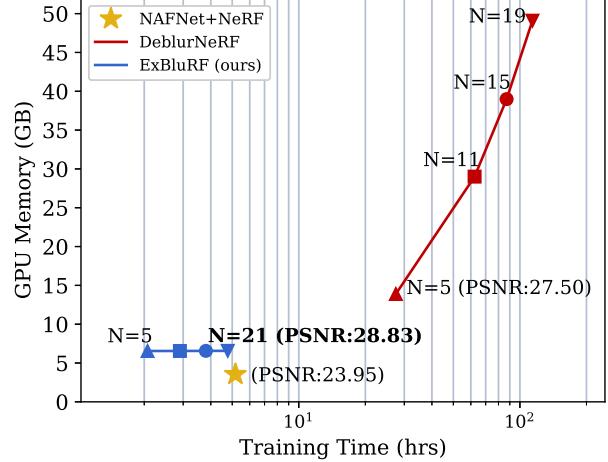


Figure 2: Training time and GPU memory consumption on “Camellia” shown in Fig 1. Our method, ExBluRF, significantly improves efficiency on both the training time and memory cost with better deblurring performance.  $N$  is the number of samples (kernel size) to reconstruct blurry color.

## 2. Related Work

**Image Deblurring.** Deblurring is a long-standing problem in image restoration due to its nature of ill-posedness. Most conventional methods [2, 7, 14, 21] estimate a 2D blur kernel that produces a blurred image by convolving on a latent sharp image. After advance in deep learning and release of large-scale datasets [33, 34, 41, 42, 63] with blurry-sharp image pairs, recent methods [3, 8, 23, 49, 52, 57, 60] are based on supervised learning with convolutional neural networks.

In the novel view synthesis with the NeRF’s per-scene optimization, the supervised learning approaches are not straightforward to be applied. The proposed method follows [24, 36] that jointly estimate multi-view sharp images and the blur kernel formulated by a camera motion trajectory and depth maps. Instead of classical optimization [48] and an auxiliary depth initialization, the optimization of the differentiable volume rendering is employed.

**Neural Radiance Fields.** Neural implicit representation combined with the differentiable volume rendering, NeRF [31] presents the photo-realistic rendering on arbitrary novel views. The NeRF-based approaches are applied to numerous 3D vision applications, where human body [56], face [12, 37, 50], hair [43], large-scale 3D reconstruction [15, 53], and simultaneous localization and mapping (SLAM) [25, 46, 65] are the representative practical applications.

On the other hand, a number of methods focus on ameliorating the weakness of NeRF for practical usage. NeRF with degraded multi-view images is constantly studied by expanding the type of image degradation. Under low-light conditions, noisy and blurry images are acquired inevitably.

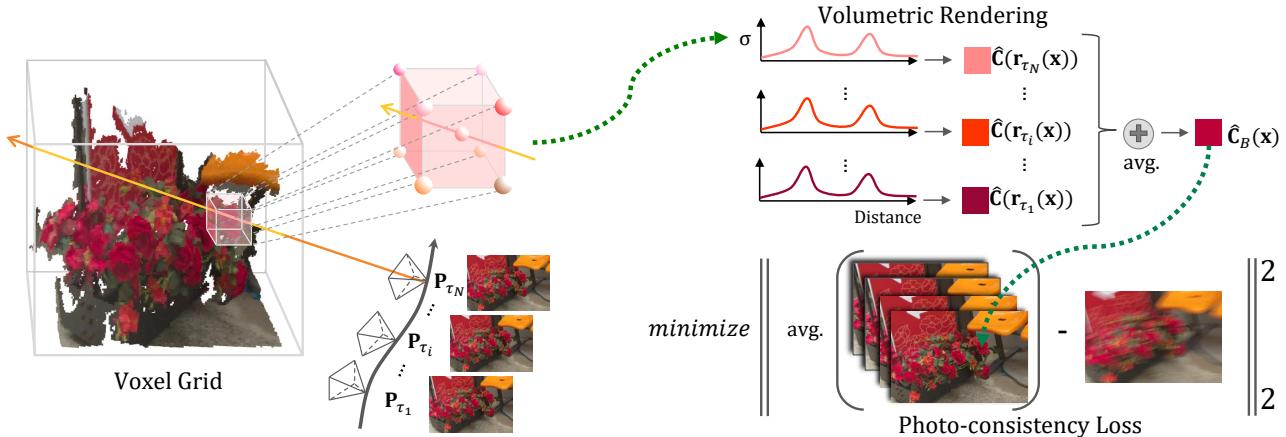


Figure 3: The overview of ExBluRF. We incorporate the physical operation that generates camera motion blur in the volume rendering of radiance fields. The blurry RGB color is reproduced by accumulating the rays along the estimated camera trajectory. By minimizing the photo-consistency loss between the accumulated color and input blurry RGB, we obtain sharp radiance fields and the camera trajectories that explain the motion blur of training views. We adopt voxel-based radiance fields to deal with explosive computation when optimizing extremely motion blurred scenes.

NAN [38] proposes the noise-aware NeRF that deals with burst noised images, and NeRF in the dark [30] more focuses on HDR view synthesis from noisy images. For blurry images, DeblurNeRF [28] introduces the end-to-end volume rendering framework that jointly estimates the pixel-level spatial varying blur kernel and the latent sharp radiance fields. DeblurNeRF yields plausible deblurring and view synthesis results, while it leaves the scalability to the extreme blur as an open problem.

**Voxel-based Radiance Fields.** While NeRF presents a novel aspect of continuous scene modeling with a neural network, the training and rendering time of the neural implicit representation need to be improved for practical usage. Many works [13, 27, 26, 40, 54] propose alternative pipelines to NeRF that can reduce the NeRF’s training and inference time. In particular, voxel-based radiance fields approaches [4, 11, 32, 47, 58] take advantage of NeRF’s differentiable volume rendering, while replacing the MLP-based implicit neural fields to voxel-based representation. Those methods accelerate the volume rendering by directly interpolating the volume densities and colors from voxel grids. Implemented in customized CUDA kernels [35], the voxel-based methods achieve real-time rendering with fast training speed.

### 3. Method

We introduce ExBluRF, efficient radiance fields to synthesize sharp novel views from extremely blurred images, and the overall pipeline is shown in Fig. 3.

#### 3.1. Preliminary: NeRF

The core idea of neural implicit representation [6, 29, 39] is to replace the conventional 3D scene representation

(e.g. point cloud, voxel, mesh) with the per-scene optimized neural network. The parameters of the network memorize the information for the continuous 3D location  $\mathbf{X} \in \mathbb{R}^3$  on the scene. Along with viewing direction  $\mathbf{d} \in \mathbb{R}^3$ , the neural function of NeRF [31]  $f_{NeRF}$  learns the mapping to the volumetric radiance fields ( $\sigma, \mathbf{c}$ ) as follow:

$$f_{NeRF} : (\mathbf{X}, \mathbf{d}) \mapsto (\sigma, \mathbf{c}), \quad (1)$$

where  $\sigma$  and  $\mathbf{c} \in \mathbb{R}^3$  denote a volume density and RGB color, respectively. To render the RGB color for a pixel coordinate  $\mathbf{x} = (x, y)$  of an arbitrary camera view, NeRF shoots a ray  $\mathbf{r}$  into the 3D space and applies the differentiable volume rendering. Let  $\mathbf{P} = [\mathbf{R}, \mathbf{t}] \in \text{SE}(3)$  and  $\mathbf{K} \in \mathbb{R}^{3 \times 3}$  denote the camera extrinsic and intrinsic parameters, respectively. The 3D location and direction of the  $k^{th}$  sample point on the ray are defined by:

$$\begin{aligned} \mathbf{X}_k &= \mathbf{t} + s_k \cdot \mathbf{d}, \\ \text{and } \mathbf{d} &= \mathbf{R} \cdot \mathbf{K}^{-1}[\mathbf{x}]_+, \end{aligned} \quad (2)$$

where  $[\mathbf{x}]_+$  is a homogeneous coordinate of  $\mathbf{x}$ . Along the ray sample points, the color of the ray  $\hat{\mathbf{C}}(\mathbf{r})$  is computed by the differentiable volume rendering [20] as follows:

$$\begin{aligned} \hat{\mathbf{C}}(\mathbf{r}) &= \sum_{k=1}^N T_k (1 - \exp(-\sigma_k \delta_k)) \mathbf{c}_k, \\ \text{where } T_k &= \exp\left(-\sum_{j=1}^{k-1} \sigma_j \delta_j\right). \end{aligned} \quad (3)$$

$\delta_k$  is the distance between the  $(k-1)^{th}$  and  $k^{th}$  sample points. Note that, NeRF employs the MLP network for the implicit representation  $f_{NeRF}$ .

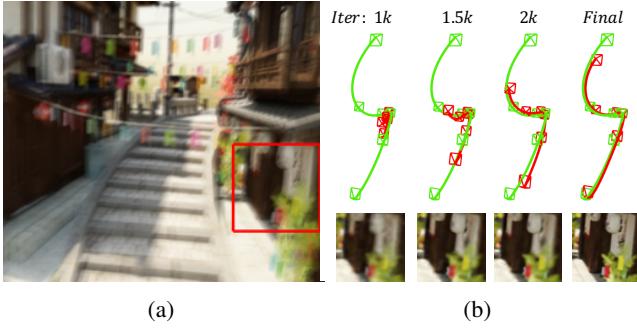


Figure 4: (a) Blurry training view of synthetic data. (b) Visualization of the GT (green) and estimated (red) trajectory of (a) by increasing the training iterations. Our blur model converges well to the GT trajectory without sophisticated initialization.

### 3.2. Motion Blur Formulation

The proposed method trains sharp radiance fields from blurry multi-view images. In particular, we focus on camera motion blur, which originated from the camera shake during exposure time. Physically, the motion blurred RGB image can be interpreted as the accumulated color of rays that hit the pixel while the camera is moving. Let  $\mathbf{r}_\tau(\mathbf{x})$  denotes the ray emitted to the pixel  $\mathbf{x}$  at shutter time  $\tau$ . The blurred RGB  $\hat{\mathbf{C}}_B(\mathbf{x})$  when the shutter is open over  $[\tau_o, \tau_c]$  is derived as follow:

$$\begin{aligned} \hat{\mathbf{C}}_B(\mathbf{x}) &= \int_{\tau_o}^{\tau_c} \hat{\mathbf{C}}(\mathbf{r}_\tau(\mathbf{x})) d\tau \\ &\approx \frac{1}{N} \sum_{i=1}^N \hat{\mathbf{C}}(\mathbf{r}_{\tau_i}(\mathbf{x})), \end{aligned} \quad (4)$$

where the origin of  $\mathbf{r}_\tau(\mathbf{x})$  is the 3D location of the camera  $\mathbf{t}_\tau$  and the direction of  $\mathbf{r}_\tau(\mathbf{x})$  is  $\mathbf{d}_\tau$  defined in Eq. 2. The rays are accumulated continuously while the shutter is open. However, we approximate the integral to a finite sum of  $N$  intermediate sub-frames  $\tau_i = \tau_o + \frac{i-1}{N-1}(\tau_c - \tau_o)$ .

We formulate the blurred RGB  $\hat{\mathbf{C}}_B(\mathbf{x})$  by 6-DOF camera poses of intermediately sub-frames. Therefore, the proposed model jointly optimizes the sub-frame camera poses  $\{\mathbf{P}_{\tau_i}\}_{i=1}^N$  with the radiance fields of the scene. Since the camera moves in an arbitrary but smooth trajectory, we re-parameterize the 6-DOF camera motion trajectory by Bézier curve [45]. Let  $\{\mathbf{p}_{\tau_i}\}_{i=1}^N \in \mathfrak{se}(3)$  denotes Lie algebra of  $\{\mathbf{P}_{\tau_i}\}_{i=1}^N$ , and  $\{\hat{\mathbf{p}}_j\}_{j=0}^M \in \mathfrak{se}(3)$  is the control points of the  $M^{th}$  order Bézier curve in Lie algebra representation. The sub-frame camera pose on Bézier curve is derived as follows:

$$\begin{aligned} \mathbf{p}_{\tau_i} &= \sum_{j=0}^M \binom{M}{j} (1 - \tau'_i)^{M-j} (\tau'_i)^j \cdot \hat{\mathbf{p}}_j, \\ \text{where } \tau'_i &= \frac{\tau_i - \tau_o}{\tau_c - \tau_o}. \end{aligned} \quad (5)$$

$\tau'_i$  is normalized time of sub-frame  $\tau_i$  over  $[0, 1]$ .

In contrast to the prior works [24, 36] that model blur formulation using the camera trajectory in light fields or video, we optimize each training view’s latent camera trajectory independently. Also, the proposed model has no prior assumption to initialize the control points  $\{\hat{\mathbf{p}}_j\}_{j=0}^M$ . However, we found that all the control points initialized with the camera pose estimated using COLMAP [44] well converge to the latent camera trajectory as shown in Fig. 4.

Note that, even though we formulate the blur operation in a low-dimensional 6-DOF pose space, the training of MLP-based radiance fields [31] from extreme camera motion requires heavy computation. We shoot  $N$  rays to synthesize a single blurry pixel with the proposed blur formulation, which increases  $\mathcal{O}(N)$  times more memory consumption and training time compared to training radiance fields on sharp images. The more extreme the camera motion blur, the larger  $N$  is required to keep the latent 3D scene sharp. Therefore, we accelerate the training of the proposed method with memory-efficient voxel-based radiance fields.

### 3.3. Voxel-based Radiance Fields

Following prior works [4, 11, 58], we adopt voxel-based radiance fields as an efficient alternative to MLP-based implicit representation of NeRF [31]. The advantage of the voxel-based radiance fields is the memory-efficient framework that limits the learnable parameters to the values on its voxel grid. While leveraging the differentiable volume rendering of NeRF, we replace the MLP network  $f_{NeRF}$  to voxel grid  $\mathcal{G}_\sigma : \mathbb{R}^3 \mapsto \mathbb{R}$  and  $\mathcal{G}_{sh} : \mathbb{R}^3 \mapsto \mathbb{R}^{9 \times 3}$ , which are output the volume density and degree 2 of spherical harmonic (SH) coefficients for RGB color, respectively. Let  $\mathbf{X}_k$  and  $\mathbf{d}$  in Eq. 2 denote the 3D location and direction of  $k^{th}$  sample point on the ray, the volume density and color of the point are:

$$\begin{aligned} \sigma_k &= \mathcal{G}_\sigma(\mathbf{X}_k), \\ \mathbf{c}_k &= S(\mathbf{d})^\top \cdot \mathcal{G}_{sh}(\mathbf{X}_k), \end{aligned} \quad (6)$$

where  $\mathcal{G}_\sigma(\mathbf{X}_k)$  and  $\mathcal{G}_{sh}(\mathbf{X}_k)$  are trilinear interpolated from the nearest 8 voxels of  $\mathbf{X}_k$ , and  $S(\mathbf{d}) : \mathbb{R}^3 \mapsto \mathbb{R}^9$  is SH function that maps viewing direction to view-dependent color. In contrast to neural radiance fields that require heavy computation of fully-connected layers, the voxel-based radiance fields directly compute the color of a ray from the interpolated values in voxel grids. By eliminating feed-forward operation, the voxel-based radiance fields achieve real-time volume rendering. In addition, coarse-to-fine sparse voxel reconstruction combined with a pruning strategy enables memory-efficient training of the radiance fields. We leverage these strong points of voxel-based radiance fields to optimization of our blurred view reconstruction.

Method	Real Motion Blur [28]			Synthetic Motion Blur			ExBlur Dataset (Ours)		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Restormer [59]+NeRF	23.22	0.682	0.289	21.15	0.541	0.419	24.33	0.571	0.545
NAFNet [5]+NeRF	24.23	0.709	0.252	20.77	0.526	0.437	26.03	0.612	0.494
DeblurNeRF [28]	25.49	0.763	<b>0.182</b>	23.98	0.687	0.301	28.87	0.709	0.402
BAD-NeRF [55]	22.59	0.633	0.252	21.94	0.590	0.349	27.15	0.647	0.453
<b>ExBluRF (Ours)</b>	<b>25.93</b>	<b>0.775</b>	0.198	<b>27.81</b>	<b>0.823</b>	<b>0.227</b>	<b>30.17</b>	<b>0.757</b>	<b>0.284</b>

Table 1: Quantitative comparison of novel view synthesis.

### 3.4. Loss Functions

We jointly optimize sharp latent radiance fields and camera trajectories of given blurry training views. The radiance fields are parameterized by densities and spherical harmonic coefficients of voxels, and the camera trajectories are 6-DOF poses of Bézier curve’s control points. We compose a blurred color  $\hat{\mathbf{C}}_B(\mathbf{x})$  by Eq. 4 and minimize the photo-consistency loss  $\mathcal{L}_{color}$  with an observed blurry color  $C(\mathbf{x})$  as follow:

$$\mathcal{L}_{color} = \sum_{\mathbf{x} \in \mathcal{N}} \left\| \mathbf{C}(\mathbf{x}) - \hat{\mathbf{C}}_B(\mathbf{x}) \right\|_2^2, \quad (7)$$

where  $\mathcal{N}$  is the set of training view’s pixels in each batch and  $\mathcal{L}_{color}$  is minimized with the mean squared error (MSE). In addition to the color consistency loss, the parameters of voxel grids are regularized by the total variation [1]  $\mathcal{L}_{TV}$  and sparsity priors  $\mathcal{L}_s$  introduced in [16] as follow:

$$\begin{aligned} \mathcal{L}_{TV} &= \sum_{i,k} \sqrt{\partial_x \mathcal{G}_\sigma(\mathbf{v}_{i,k})^2 + \partial_y \mathcal{G}_\sigma(\mathbf{v}_{i,k})^2 + \partial_z \mathcal{G}_\sigma(\mathbf{v}_{i,k})^2}, \\ \mathcal{L}_s &= \sum_k \log(1 + 2\sigma_k^2), \end{aligned}$$

where  $\{\mathbf{v}_{i,k}\}_{i=1}^8$  denotes voxel grids containing  $k^{th}$  sample point along the ray.  $\mathcal{L}_{TV}$  guides voxel grids to have smooth volume density while preserving the object’s boundaries of the scene, and  $\mathcal{L}_s$  leads volume density to sharp distribution along the ray. Note that the same  $\mathcal{L}_{TV}$  is applied to SH coefficients  $\mathcal{G}_{sh}(\mathbf{v}_{i,k})$ . Finally, our combined loss function is set as follows:

$$\mathcal{L} = \mathcal{L}_{color} + \lambda_{TV} \mathcal{L}_{TV} + \lambda_s \mathcal{L}_s, \quad (8)$$

where  $\lambda_{TV}$  and  $\lambda_s$  are set to  $5 \times 10^{-4}$  and  $1 \times 10^{-12}$  in our experiments, respectively.

### 3.5. ExBlur Dataset

To quantitatively evaluate the deblurring performance on real world scenes, we collect a real dataset, named ExBlur dataset, using a dual-camera system similar to [19, 41, 63, 64]. The dual-camera system equally splits

photons into two cameras using a beam-splitter and simultaneously captures a blurry image and sequence of sharp images, as done in [41]. Specifically, one of the cameras captures a single blurry image with a long exposure time, and the other camera captures a sequence of sharp images during the exposure time of the blurry image. Using the system, we captured eight scenes, each consisting of 20 to 40 multi-view blurry images and the corresponding sequences of sharp images.

We utilize the pairs of blurry and sequence of sharp images in two aspects: 1) Accurate evaluation of novel view synthesis on real blurred scenes. 2) Verification of our blur formulation that optimizes latent camera trajectories on real camera shake motion. The real dataset of DeblurNeRF [28] provides real blurry images for optimizing NeRF. However, inaccurate camera poses of blurry images lead to misalignments on test views, resulting in erroneous evaluation. The ExBlur dataset provides accurate camera poses of blurry images by applying structure-from-motion (COLMAP [44]) to the corresponding sharp images. The accurate camera poses confer rigid relative poses to the test views and enable accurate evaluation of novel view synthesis on real blurred scenes. Furthermore, applying structure-from-motion to other views of sharp sequences produces the ground truth camera trajectory that generates blurry images. We evidence that our blur optimization is well converged to the ground trajectories by measuring the trajectory evaluation metrics proposed in [62].

## 4. Experiments

**Datasets.** We evaluate our method with three datasets, the real and synthetic datasets from DeblurNeRF [28] and the proposed ExBlur dataset. The real dataset of DeblurNeRF [28] consists of 10 scenes that are captured from a hand-held camera. Each scene has 20 to 40 blurry training views and 4 to 5 sharp test views. The camera poses are estimated from blurry training views and sharp test views using COLMAP [44]. For the synthetic dataset, 5 scenes are collected using Blender [9], where each scene consists of 29 blurry training views and 5 sharp test views. The GT camera poses are exported while rendering using Blender. In the synthetic dataset of DeblurNeRF [28], the synthetic blurry

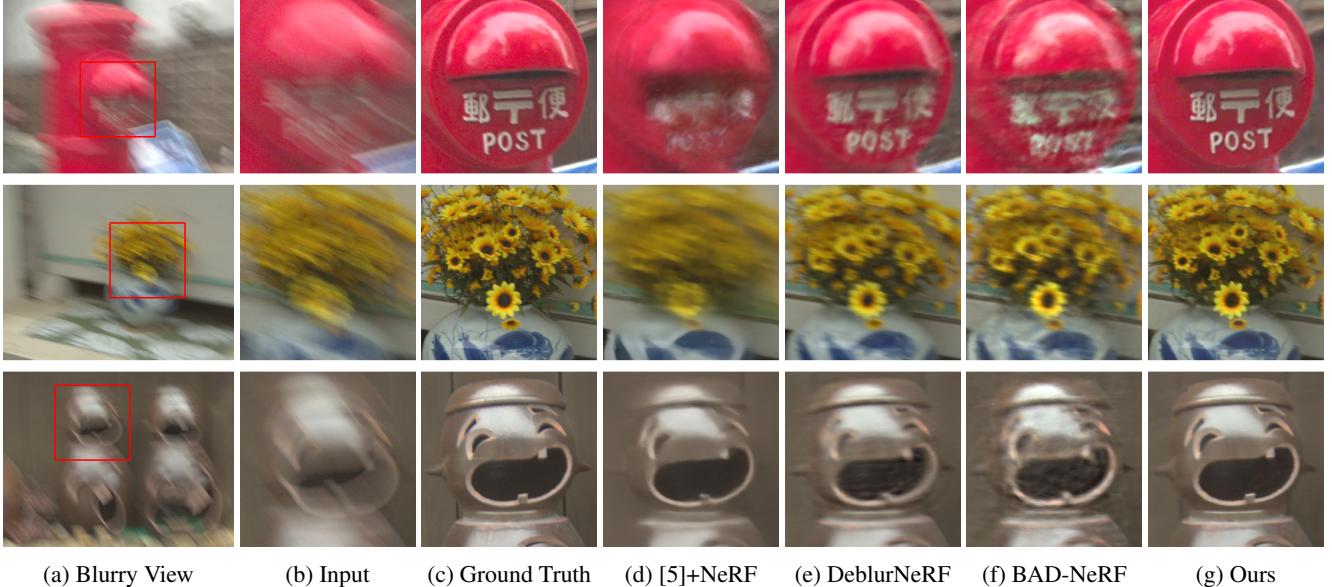


Figure 5: Qualitative comparison of deblurring on ExBlur dataset.

views are generated only with linear motion. However, for our experiments, we generate more challenging motion blur by random camera motion trajectories in 6-DOF. The image resolution is  $600 \times 400$  for both the real and synthetic datasets of DeblurNeRF.

For the ExBlur dataset, we collected 8 diverse outdoor scenes with challenging camera motion. Each scene has 20 to 40 blurry views for training and 4 to 6 test views for evaluation of novel view synthesis. All blurry images of the ExBlur dataset are paired with sequences of sharp images. We leverage the sharp images for evaluation of the deblurring performance and accuracy of our estimated camera trajectories. The resolution for the multi-view images of ExBlur dataset is  $800 \times 540$ .

**Implementation Details.** To model complex camera motion with the differentiable curve, Bézier curve of order  $M = 7$  is applied and the number of sampling on the trajectory is  $N = 21$  for the proposed method. We build our voxel-based radiance fields with a custom CUDA [35] kernel that extends the implementation of [11] to backpropagate gradients to 6-DOF camera trajectory for our blur model. For the parameters of camera motion trajectory, we utilize Adam [22] optimizer with learning rate  $5 \times 10^{-4}$  and RMSProp [17] optimizer is applied for voxel grids. To optimize radiance fields from blurry images, we train ExBluRF for 200k iterations with a batch size of 25k rays on a single NVIDIA RTX Quadro RTX 8000. In addition, the pruning and coarse-to-fine strategies are applied for voxel reconstruction, where voxel resolutions of  $xy$ -dimensions are upsampled every 40k iteration.

#### 4.1. Results

We compare the deblurring and novel view synthesis performance with DeblurNeRF [28], BAD-NeRF [55] and 2D image deblurring methods [59, 5] combined with NeRF [28]. DeblurNeRF jointly optimizes neural radiance fields and 2D pixel-wise blur kernel estimation in an end-to-end manner. BAD-NeRF [55] is our concurrent work that models the camera motion blur using the 6-DOF linear camera trajectory. For our experiments, we follow the default configuration of DeblurNeRF. To compare with image deblurring methods, we select the two state-of-the-art image deblurring methods that are Restormer [59] and NAFNet [5]. Blurry training views are independently deblurred, and the deblurred images are used to input of vanilla NeRF.

**Evaluation on Deblurring.** The rendered novel view images are evaluated by PSNR, SSIM [18], LPIPS [18] as metrics. The three metrics are vulnerable to misalignment, therefore we use camera poses from sharp image pairs for the ExBlur and synthetic datasets for quantitative evaluation. Since the real dataset of DeblurNeRF [28] only provides camera poses estimated from blurry images, we re-align the poses of test views by rendered images from radiance fields of each method for quantitative comparison. After optimization of each method, we render deblurred images of training views with initial poses. Then, we apply image registration of COLMAP [44] to produce aligned poses of test views with deblurred training views. Note that, the indirect alignment using deblurred results depends on each method’s deblurring performance, so the evaluation could not be accurate.

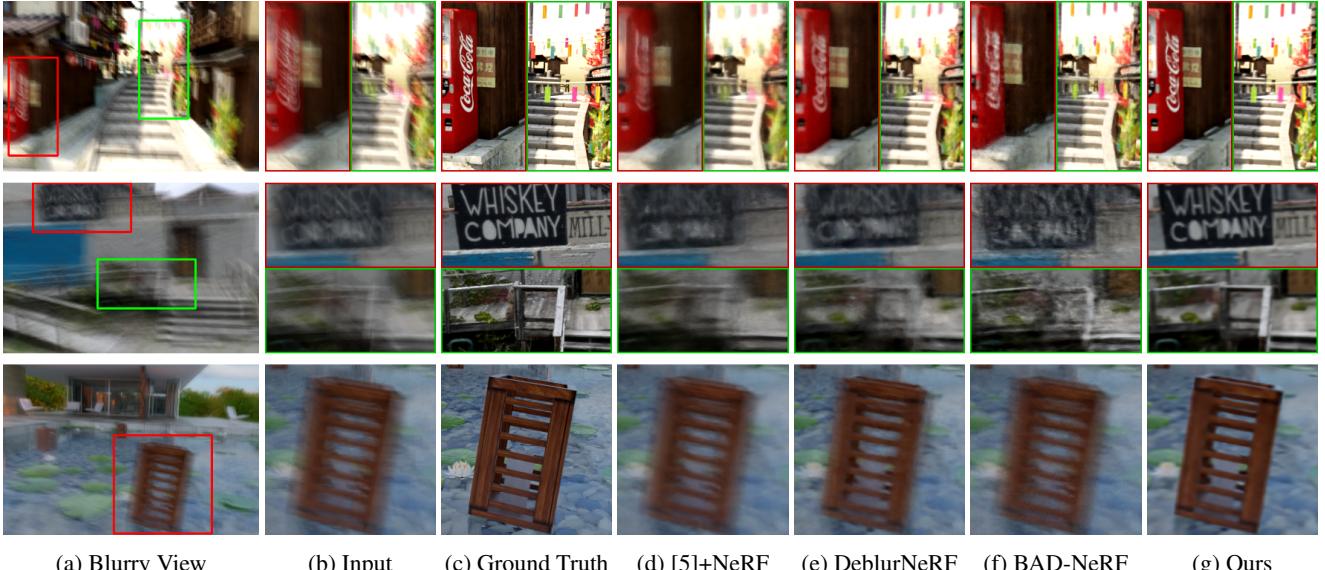


Figure 6: Qualitative comparison of deblurring on synthetic dataset.

Method	Blur Model	Radiance Fields	Memory cost	$N = 5$	$N = 11$	$N = 15$	$N = 19$
DeblurNeRF [28]	2D kernel	Neural	$\mathcal{O}(BNPL) + \theta_{NeRF} + \theta_{kernel}$	13.89 GB	29.01 GB	38.99 GB	49.09 GB
ExbluRF (Ours)	Camera trajectory	Voxel	$\mathcal{O}(BN) + \phi_{voxel}$	6.54 GB	6.55 GB	6.56 GB	6.57 GB

Table 2: Comparison of memory cost of DeblurNeRF [28] and our ExBluRF.

The quantitative results are shown in Tab. 1, and qualitative results are shown in Fig. 5 and Fig. 6. Tab. 1 shows that our method consistently outperforms previous approaches on novel view synthesis. For Real Motion Blur dataset [28] that is captured on relatively small motion blur, our method is slightly better than others. If the camera motion is small, DeblurNeRF with 2D pixel-wise kernel estimation converges to a sharp latent 3D scene as well. Without considering blur operation in volume rendering, view-dependent deblurring errors of Restormer and NAFNet generate hardly deblurred surfaces when training NeRF, as shown in Fig. 5. On the other hand, the proposed method, DeblurNeRF reconstruct 3D consistent scenes, where our approach renders sharper latent 3D scenes. Note that BAD-NeRF formulates the camera motion in the 6-DOF linear trajectory, and fails to restore the sharp radiance fields from the real-world blurry scenes with the hand-shake camera motion.

In the case of extreme camera motion, a large kernel size is required for DeblurNeRF, otherwise, the latent 3D scene becomes a little blurry not to produce a discontinuity in blurred view reconstruction while training. The kernel size on DeblurNeRF, the number of sampling on a trajectory in our method, is the key and inevitable factor to deblur extreme motion. However, DeblurNeRF’s naive MLP-based blur kernel estimation network and neural radiance fields demand painfully increasing training time and computation resources.

	Notation	Meaning
Common	$N$	Number of sampling for blur operation
	$B$	Number of ray batch
	$P$	Number of sample points on a ray
DeblurNeRF	$\theta_{kernel}$	MLP parameters for kernel estimation
	$\theta_{NeRF}$	MLP parameters for NeRF
	$L$	The number of MLP layers in NeRF
ExBluRF (Ours)	$\phi_{voxel}$	Voxel grid parameters

Table 3: Notations used for the analysis of memory cost.

## 4.2. Analysis and Ablation

**Analysis on Memory Efficiency.** We examine the efficiency of our model in terms of memory cost as shown in Tab. 2. In Tab. 3, we categorize the key factors that influence memory consumption. We observe that our method keeps constant memory usage regardless of the number of sampling on a trajectory. On the other hand, DeblurNeRF [28]’s memory consumption grows proportional to the kernel size. Both approaches compute total  $B \times N$  rays per iteration. For our model, each ray marching process sequentially evaluates sample points on the ray from the origin to the far. This only takes  $O(1)$  memory space for each ray. In contrast, DeblurNeRF [28] synchronously feed-forwards  $P$  points for each ray. Moreover, DeblurNeRF [28]’s network saves activation value from  $L$  layers. This results in  $O(PL)$  memory cost required for each ray marching. Consequently, our approach can enjoy a large number of sampling which is a crucial factor of high performance with little memory limi-

tation. Note that our model has  $\phi_{voxel}$  as base model memory which might be larger than DeblurNeRF [28]’s model size  $\theta_{kernel}$  and  $\theta_{NeRF}$ , while this does not scale by  $N$ .

Blur Model	Radiance Fields	$N$ Sampling	PSNR	GPU Memory	Training time
-	Neural	-	22.96	3.55 GB	5.17 hrs
2D kernel	Neural	5	27.76	13.89 GB	27.48 hrs
2D kernel	Voxel	21	27.79	6.19 GB	5.72 hrs
Cam. Traj.	Voxel	21	28.83	4.31 GB	4.41 hrs

Table 4: Ablation test on the main components in our method. We report “Camellia” scene of ExBlur dataset.

**Ablation Study.** We analyze the effectiveness of the proposed blur model and voxel-based radiance fields in Tab. 4. The baseline method that consists of an MLP-based 2D kernel estimator and neural radiance fields refers to DeblurNeRF [28] with a default configuration. When replacing the neural radiance fields to voxel-based radiance fields with increased kernel size, the computational cost is significantly reduced. However, the deblurring performance is hardly improved for the extremely motion blurred scene. The 2D pixel-wise kernel estimation is difficult to leverage the prior knowledge that the camera is moving on a single trajectory and rays are sequentially accumulated.

	Synthetic		ExBlur	
	Factory	Tanabata	Postbox	Sunflowers
COLMAP <sub>init</sub> [44]	0.0480	0.0422	0.0146	0.0211
ExBluRF (Ours)	0.0141	0.0102	0.0019	0.0040

Table 5: Evaluation of absolute trajectory error on ExBlur and synthetic datasets. We report 2 scenes for each dataset.

**Convergence of camera trajectory.** To validate estimated camera trajectories from our method, we evaluate absolute trajectory error (ATE) [62] using GT trajectories of the ExBlur and synthetic datasets. Tab. 5 shows that our approach minimizes ATE without any additional supervision on the trajectory. Even though each training view is extremely blurred, and difficult to restore the camera trajectory separately, a single 3D scene is optimized to explain all blurs of training views. This strong constraint encourages our method to overcome the ill-posedness of deblurring problem and reconstruct sharp radiance fields with accurate camera trajectories.

**Sensitivity to order of Bézier curve.** We investigate how our camera trajectory model is affected by order of Bézier curve on the “Pool” scene of the synthetic dataset. Tab. 6 shows PSNR and ATE metrics on translation and rotation by gradually increasing the number of control points of Bézier curve. The performance significantly increases when the order grows from 1 to 3. The 1<sup>th</sup> order Bézier curve refers to linear motion in 6-DOF pose space. For challenging camera motions, the linear motion is difficult to fit the trajectory that generates blur, therefore high-order modeling of motion trajectory is required. The result of 9<sup>th</sup>

Order of Bézier curve	PSNR↑	ATE <sub>pos</sub> ↓	ATE <sub>rot</sub> ↓
1	25.30	0.0942	0.367
3	29.65	0.0491	0.225
5	30.06	0.0486	0.152
7	30.05	0.0454	0.163
9	29.92	0.0483	0.243

Table 6: Sensitivity to order of Bézier curve. We uniformly increase the order by {1, 3, 5, 7, 9}. The first-order curve means linear motion in 6-DOF pose space.

$N$ of sampling on trajectory	PSNR↑	SSIM↑	LPIPS↓
5	28.62	0.717	0.360
11	30.03	0.763	0.271
21	30.11	0.767	0.254
31	30.23	0.768	0.230

Table 7: Effectiveness of the number of sampling on the camera trajectory for deblurring performance.

order shows the possibility of overfitting when increasing the curve’s order unnecessarily.

**Effectiveness of the number of samplings.** Tab. 7 shows the deblurring performance of our method on the “Postbox” scene of the ExBlur dataset according to the number of sampling points on a trajectory. As discussed in Sec. 4.1, the number of sampling on a trajectory is crucial to the deblurring performance as more sampling points lead to sharper novel view synthesis. In particular, the performance of 5 samplings, used in DeblurNeRF [28], significantly degraded the performance compared to the other variants. The result again shows the limitation of DeblurNeRF [28], which cannot handle a large number of sampling points. On the other hand, ExBluRF can deal with a large number of sampling points without concern about memory and computation costs.

## 5. Conclusion

In this paper, we proposed ExBluRF, efficient radiance fields and blur formulation that restore a sharp 3D scene from extreme motion blurred images. We formulate motion blur with a 6-DOF camera trajectory and validate that the proposed blur model is well converged to the ground truth trajectory and produces sharp 3D radiance fields. This blur model is explicitly parameterized in 6-DOF camera poses and combined with computation-efficient voxel-based radiance fields. We collect the ExBlur dataset that provides extremely blurred multi-view images with ground truth sharp pairs to quantitatively evaluate the deblurring performance on real world scenes. ExBluRF outperforms existing deblurring approaches both on real and synthetic datasets, while resolving explosive computation cost of the deblurring operation on radiance fields optimization.

## 6. Acknowledgment

This work was supported in part by the IITP grants [No.2021-0-01343, Artificial Intelligence Graduate School Program (Seoul National University), No. 2021-0-02068, and No.2023-0-00156], the NRF grant [No. 2021M3A9E4080782] funded by the Korea government (MSIT), and the SNU-Naver Hyperscale AI Center.

## References

- [1] Amir Beck and Marc Teboulle. Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. *IEEE transactions on image processing*, 18(11):2419–2434, 2009.
- [2] Patrizio Campisi and Karen Egiazarian. *Blind image deconvolution: theory and applications*. CRC press, 2017.
- [3] Ayan Chakrabarti. A neural approach to blind motion deblurring. In *ECCV*, 2016.
- [4] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *ECCV*, 2022.
- [5] Liangyu Chen, Xiaojie Chu, Xiangyu Zhang, and Jian Sun. Simple baselines for image restoration. In *ECCV*, 2022.
- [6] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *CVPR*, 2019.
- [7] Sunghyun Cho and Seungyong Lee. Fast motion deblurring. In *ACM SIGGRAPH Asia*, pages 1–8. ACM, 2009.
- [8] Sung-Jin Cho, Seo-Won Ji, Jun-Pyo Hong, Seung-Won Jung, and Sung-Jea Ko. Rethinking coarse-to-fine approach in single image deblurring. In *ICCV*, 2021.
- [9] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018.
- [10] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *CVPR*, 2022.
- [11] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *CVPR*, 2022.
- [12] Guy Gafni, Justus Thies, Michael Zollhofer, and Matthias Nießner. Dynamic neural radiance fields for monocular 4d facial avatar reconstruction. In *CVPR*, 2021.
- [13] Stephan J Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. Fastnerf: High-fidelity neural rendering at 200fps. In *ICCV*, 2021.
- [14] Ankit Gupta, Neel Joshi, C Lawrence Zitnick, Michael Cohen, and Brian Curless. Single image deblurring using motion density functions. In *ECCV*, 2010.
- [15] Zekun Hao, Arun Mallya, Serge Belongie, and Ming-Yu Liu. Gancraft: Unsupervised 3d neural rendering of minecraft worlds. In *ICCV*, 2021.
- [16] Peter Hedman, Pratul P Srinivasan, Ben Mildenhall, Jonathan T Barron, and Paul Debevec. Baking neural radiance fields for real-time view synthesis. In *ICCV*, 2021.
- [17] Geoffrey Hinton. Rmsprop.
- [18] Alain Hore and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In *ICPR*, 2010.
- [19] Jucheol Won Jaesung Rim, Haeyun Lee and Sunghyun Cho. Real-world blur dataset for learning and benchmarking de-blurring algorithms. In *ECCV*, 2020.
- [20] James T Kajiya and Brian P Von Herzen. Ray tracing volume densities. *ACM SIGGRAPH*, 18(3):165–174, 1984.
- [21] Tae Hyun Kim and Kyoung Mu Lee. Segmentation-free dynamic scene deblurring. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014.
- [22] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.
- [23] Orest Kupyn, Tetiana Martyniuk, Junru Wu, and Zhangyang Wang. Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better. In *ICCV*, 2019.
- [24] Dongwoo Lee, Haesol Park, In Kyu Park, and Kyoung Mu Lee. Joint blind motion deblurring and depth estimation of light field. In *ECCV*, 2018.
- [25] Kejie Li, Yansong Tang, Victor Adrian Prisacariu, and Philip HS Torr. Bnv-fusion: dense 3d reconstruction using bi-level neural volume fusion. In *CVPR*, 2022.
- [26] David B Lindell, Julien NP Martel, and Gordon Wetzstein. Autoint: Automatic integration for fast neural volume rendering. In *CVPR*, 2021.
- [27] Stephen Lombardi, Tomas Simon, Gabriel Schwartz, Michael Zollhoefer, Yaser Sheikh, and Jason Saragih. Mixture of volumetric primitives for efficient neural rendering. *ACM Transactions on Graphics*, 40(4):1–13, 2021.
- [28] 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 *CVPR*, 2022.
- [29] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *CVPR*, 2019.
- [30] 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 *CVPR*, 2022.
- [31] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.
- [32] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics*, 41(4):1–15, 2022.
- [33] Seungjun Nah, Sungyong Baik, Seokil Hong, Gyeongsik Moon, Sanghyun Son, Radu Timofte, and Kyoung Mu Lee. Ntire 2019 challenge on video deblurring and super-resolution: Dataset and study. In *CVPR Workshops*, 2019.
- [34] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *CVPR*, 2017.
- [35] NVIDIA, Péter Vingelmann, and Frank H.P. Fitzek. Cuda, release: 10.2.89, 2020.
- [36] Haesol Park and Kyoung Mu Lee. Joint estimation of camera pose, depth, deblurring, and super-resolution from a blurred image sequence. In *ICCV*, 2017.

- [37] 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 *ICCV*, 2021.
- [38] Naama Pearl, Tali Treibitz, and Simon Korman. Nan: Noise-aware nerfs for burst-denoising. In *CVPR*, 2022.
- [39] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *ECCV*, 2020.
- [40] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In *ICCV*, 2021.
- [41] Jaesung Rim, Geonung Kim, Jungeon Kim, Junyong Lee, Seungyong Lee, and Sunghyun Cho. Realistic blur synthesis for learning image deblurring. In *ECCV*, 2022.
- [42] Jaesung Rim, Haeyun Lee, Juchel Won, and Sunghyun Cho. Real-world blur dataset for learning and benchmarking deblurring algorithms. In *ECCV*, 2020.
- [43] Radu Alexandru Rosu, Shunsuke Saito, Ziyan Wang, Chengelei Wu, Sven Behnke, and Giljoo Nam. Neural strands: Learning hair geometry and appearance from multi-view images. In *ECCV*, 2022.
- [44] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016.
- [45] Pratul P Srinivasan, Ren Ng, and Ravi Ramamoorthi. Light field blind motion deblurring. In *CVPR*, 2017.
- [46] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew J Davison. imap: Implicit mapping and positioning in real-time. In *ICCV*, 2021.
- [47] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *CVPR*, 2022.
- [48] Deqing Sun, Stefan Roth, and Michael J Black. Secrets of optical flow estimation and their principles. In *CVPR*, 2010.
- [49] Jian Sun, Wenfei Cao, Zongben Xu, and Jean Ponce. Learning a convolutional neural network for non-uniform motion blur removal. In *CVPR*, 2015.
- [50] Jingxiang Sun, Xuan Wang, Yong Zhang, Xiaoyu Li, Qi Zhang, Yebin Liu, and Jue Wang. Fenerf: Face editing in neural radiance fields. In *CVPR*, 2022.
- [51] Matthew Tancik, Ben Mildenhall, Terrance Wang, Divi Schmidt, Pratul P Srinivasan, Jonathan T Barron, and Ren Ng. Learned initializations for optimizing coordinate-based neural representations. In *CVPR*, 2021.
- [52] Xin Tao, Hongyun Gao, Xiaoyong Shen, Jue Wang, and Jiaya Jia. Scale-recurrent network for deep image deblurring. In *CVPR*, 2018.
- [53] Haithem Turki, Deva Ramanan, and Mahadev Satyanarayanan. Mega-nerf: Scalable construction of large-scale nerfs for virtual fly-throughs. In *CVPR*, 2022.
- [54] Huan Wang, Jian Ren, Zeng Huang, Kyle Olszewski, Menglei Chai, Yun Fu, and Sergey Tulyakov. R2l: Distilling neural radiance field to neural light field for efficient novel view synthesis. In *ECCV*, 2022.
- [55] Peng Wang, Lingzhe Zhao, Ruijie Ma, and Peidong Liu. Bad-nerf: Bundle adjusted deblur neural radiance fields. In *CVPR*, June 2023.
- [56] Chung-Yi Weng, Brian Curless, Pratul P Srinivasan, Jonathan T Barron, and Ira Kemelmacher-Shlizerman. Humanerf: Free-viewpoint rendering of moving people from monocular video. In *CVPR*, 2022.
- [57] Patrick Wieschollek, Michael Hirsch, Bernhard Scholkopf, and Hendrik Lensch. Learning blind motion deblurring. In *ICCV*, 2017.
- [58] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. In *ICCV*, 2021.
- [59] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *CVPR*, 2022.
- [60] Kaihao Zhang, Wenhan Luo, Yiran Zhong, Lin Ma, Bjorn Stenger, Wei Liu, and Hongdong Li. Deblurring by realistic blurring. In *CVPR*, 2020.
- [61] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020.
- [62] Zichao Zhang and Davide Scaramuzza. A tutorial on quantitative trajectory evaluation for visual (-inertial) odometry. In *IROS*, 2018.
- [63] Zhihang Zhong, Ye Gao, Yinqiang Zheng, and Bo Zheng. Efficient spatio-temporal recurrent neural network for video deblurring. In *ECCV*, 2020.
- [64] Zhihang Zhong, Ye Gao, Yinqiang Zheng, Bo Zheng, and Imari Sato. Real-world video deblurring: A benchmark dataset and an efficient recurrent neural network. *International Journal of Computer Vision*, 2022.
- [65] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R. Oswald, and Marc Pollefeys. Nice-slam: Neural implicit scalable encoding for slam. In *CVPR*, 2022.