



# NeRF-FF: a plug-in method to mitigate defocus blur for runtime optimized neural radiance fields

Tristan Wirth<sup>1</sup> · Arne Rak<sup>1</sup> · Max von Buelow<sup>1</sup> · Volker Knauthe<sup>1</sup> · Arjan Kuijper<sup>1,2</sup> · Dieter W. Fellner<sup>1,2,3</sup>

Accepted: 18 May 2024 / Published online: 1 July 2024  
© The Author(s) 2024

## Abstract

Neural radiance fields (NeRFs) have revolutionized novel view synthesis, leading to an unprecedented level of realism in rendered images. However, the reconstruction quality of NeRFs suffers significantly from out-of-focus regions in the input images. We propose NeRF-FF, a plug-in strategy that estimates image masks based on Focus Frustums (FFs), i.e., the visible volume in the scene space that is in-focus. NeRF-FF enables a subsequently trained NeRF model to omit out-of-focus image regions during the training process. Existing methods to mitigate the effects of defocus blurred input images often leverage dynamic ray generation. This makes them incompatible with the static ray assumptions employed by runtime-performance-optimized NeRF variants, such as Instant-NGP, leading to high training times. Our experiments show that NeRF-FF outperforms state-of-the-art approaches regarding training time by two orders of magnitude—reducing it to under 1 min on end-consumer hardware—while maintaining comparable visual quality.

**Keywords** Novel View Synthesis · Neural Radiance Fields · Image Restoration

## 1 Introduction

The reconstruction and subsequent novel view synthesis based on 2D input images of complex 3D scenes is a long-standing research problem in computer vision. Recently, the introduction of neural radiance fields (NeRFs) [30] has led to enormous progress regarding the rendering of photorealistic views from multi-view images. NeRFs predict radiance and volume density from a 3D spatial location and viewing direction using a multilayer perceptron (MLP). The outputs of this MLP can be rendered by established volume rendering techniques [15].

Most NeRF variants are restricted to well-defined capturing environments, with noise-free images of a static scene with known camera parameters. Therefore, most NeRF variants produce inferior rendering results when the input images suffer from different types of blur. The most prominent examples are motion blur, where the camera changes position

during exposure time leading to superposition of multiple views, and defocus blur that is caused by a suboptimal choice of camera aperture. There are multiple approaches that address the issue of motion blur [8, 21, 40, 53], defocus blur [59] or both [20, 26, 37] in the context of NeRFs. However, while there is previous work that has shown great progress in accelerating NeRF training and rendering times [4, 10, 31, 48], most of the existing blur-resistant contributions do not leverage the potential of these accelerations. This can be attributed to the fact that most of the blur-resistant strategies rely on dynamically generated rays, while the accelerated NeRF methods leverage explicit density volume representations to prune low density voxels. These strategies are based on the assumption of static rays.

In this paper, we propose NeRF-FF, a novel plug-in method that can be combined with a multitude of NeRF and NeRF-like novel view synthesis strategies, which significantly improves the rendering quality based on blurred input images. NeRF-FF calculates image masks that contain in-focus or only slightly out-of-focus regions of the input images that lead to a beneficial trade-off between image quality and scene coverage. To achieve this, NeRF-FF first identifies regions that are in-focus by applying the discrete LAPLACE operator, which detects sharp edges that are predominantly

✉ Tristan Wirth  
tristan.wirth@gris.tu-darmstadt.de

<sup>1</sup> Technische Universität Darmstadt, Darmstadt, Germany

<sup>2</sup> Fraunhofer IGD, Darmstadt, Germany

<sup>3</sup> Technische Universität Graz, Graz, Austria

prevalent in in-focus regions. These feature points are leveraged to estimate the visible and in-focus volume for each image in the 3D scene space using depth maps from a NeRF, which was trained before. This volume has the shape of a pyramid cropped by a near and far plane. We refer to it as Focus Frustum.

These Focus Frustums are then refined by expanding their near and far plane such that each point in the relevant scene geometry is part of at least a certain percentage of the Focus Frustums. The projection of dense scene points encompassed by the Focus Frustums into the respective image space provides image masks indicating the pixels, a subsequent NeRF model can be trained on, to obtain unblurred novel views. Thus, we provide a trade-off between the reconstruction of the complete scene geometry and usage of optimal, i.e., in-focus or only slightly out-of-focus, views.

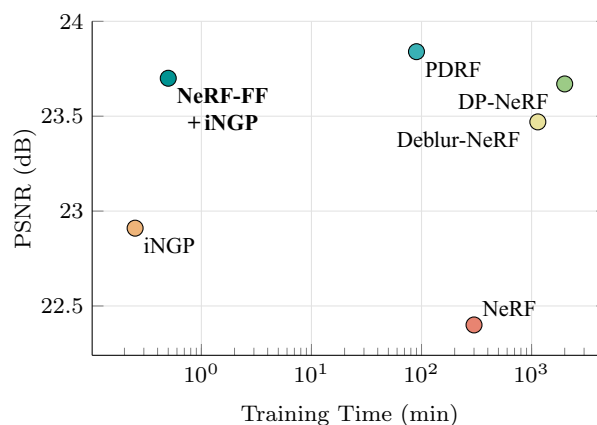
Our experiments show that the resulting image quality of NeRF-FF is comparable to state-of-the-art strategies. At the same time, NeRF-FF can be used in combination with accelerated NeRF variants like Instant-NGP [31], which is not easily integrable with state-of-the-art methods like PDRF [37]. The proposed combination of NeRF-FF and Instant-NGP [31] outperforms the state of the art by two orders of magnitude regarding training time, bringing it down to under 1 min (see Fig. 1).

In summary, this paper offers the following contributions:

- We propose NeRF-FF, a plug-in strategy that estimates image masks based on Focus Frustums (FFs), i.e., the visible volume in scene space that is in-focus, that allow a subsequent NeRF model to omit out-of-focus image regions during training, including a reasonable choice of hyperparameters.
- We provide a quantitative analysis showing that the combination of NeRF-FF with Instant-NGP [31] yields resulting novel views that are comparable to the state of the art in quality while reducing the required computation time by two orders of magnitude. This results in end-to-end runtimes under 1 min on end-consumer hardware.

## 2 Related work

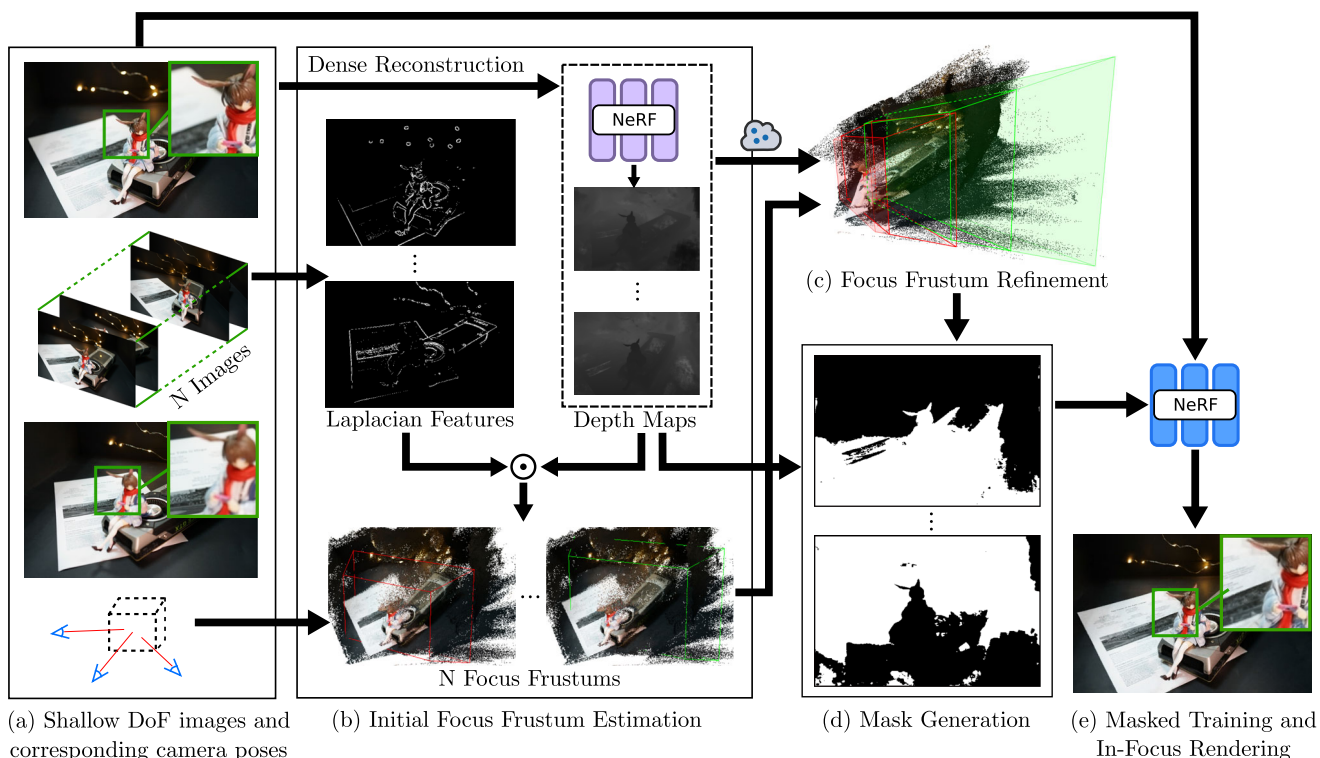
*Neural Radiance Fields (NeRFs).* NeRFs [30] model the radiance of a static 3D scene as implicit representation, which can be used by classical volume rendering techniques [15] to perform photorealistic novel view synthesis—the rendering of an unknown viewing direction based on a set of input images. This approach has gained popularity in recent years, leading to many follow-up works that have extended NeRF capturing capabilities. There are NeRF variants that can handle dynamic scenes [22, 35, 36, 39, 51, 65], make NeRFs



**Fig. 1** NeRF-FF in combination with Instant-NGP (NeRF-FF+iNGP) accelerates training times on blurry input images in comparison with the state of the art by two orders of magnitude, while significantly improving Instant-NGP's novel view synthesis capabilities on these inputs. Results obtained from defocus real dataset by Ma et al. [26] using an RTX 3090

robust against the challenges of in-the-wild image capturing [27, 29, 43, 49, 54, 55], reduce the required input image count [9, 33, 41, 62] or enable training without known camera poses [23, 28, 56]. Other work has addressed the robustness of NeRFs against the influence of different types of noise, like exposure noise [11, 14, 29], illumination changes [6, 27, 67] and aliasing [1, 2]. Some publications attempt to overcome the static nature of NeRFs regarding baked lighting and rigid scene geometry by applying them to generative models [34, 44], enabling dynamic relighting of the captured scenes [32, 46, 54, 64, 67], using them for animatable human avatar generation [13, 38, 47, 68] and through approaches that allow scene editing [5, 12, 19, 24, 34, 60, 63]. To improve the high computation times of the initial NeRF approach, strategies to reduce training [4, 10, 31, 48, 52, 61] and rendering [7, 17, 25, 42] time have been proposed, making NeRFs more accessible to end-consumer hardware.

*NeRFs from blurred images.* In recent years, the challenge of applying NeRFs to blurry input images has been researched extensively. The resulting publications either address out-of-focus [59] or motion blurred [8, 21, 40, 53] images in particular, or blurry images in general [20, 26, 37]. Following the tradition of single image deblurring, Deblur-NeRF [26] estimates a spatially varying blurring kernel by using a deformable sparse kernel module that models the physical blurring process. By taking the translation of the ray origin into account, they extend this 2D kernel estimate to 3D space. DP-NeRF [20] extends this approach by modeling a prior for blurring in order to guarantee blur kernel consistency throughout the image. PDRF [37] takes scene geometry into account by realizing a two-step approach that leverages a 2D blur estimate computed on a coarse set of rays and later incorporates 3D features into it. DoF-NeRF [59] physically



**Fig. 2** The NeRF-FF pipeline. NeRF-FF improves the rendering quality of NeRFs for training data that suffers from a defocus aberration (a). First, NeRF-FF estimates Focus Frustums (b), the view frustum sub-volumes of each training view containing only in-focus image regions. For this, we combine Laplacian features indicating in-focus points with their corresponding depths, which is obtained using an arbitrary dense reconstruction strategy—in our case Instant-NGP. The Focus Frustums

are then refined by iteratively growing their near and far planes until the entire scene geometry is adequately captured (c). Finally, NeRF-FF leverages the refined Focus Frustums to generate a binary mask for each training image that segments in-focus from out-of-focus image regions (d). The masks are used as additional training input data, allowing NeRF models to omit out-of-focus regions during training and, thereby, learn an in-focus representation of the scene (e)

models the aperture of the camera lens as an additional MLP that can be added to a NeRF framework.

These strategies, however, require dynamically generated rays. In contrast to that, highly performance-optimized NeRF variants, like Instant-NGP [31], Plenoxels [10] and Tensor [4], explicitly model the density volume to accelerate the training process by efficient pruning, relying on static rays. Therefore, the above-mentioned approaches for NeRF synthesis on blurred inputs are not compatible with these performance-optimized NeRF variants, leading to training times of multiple hours up to nearly one day on high-end consumer hardware.

### 3 NeRF-FF

Most state-of-the-art approaches jointly estimate a spatially varying blur kernel alongside with the scene geometry and radiance. In contrast to that, NeRF-FF generates masks for the input images, such that a subsequently applied NeRF variant—in our case Instant-NGP [31]—trains on in-focus

regions only. In a first step, NeRF-FF identifies sharp edges, which indicate in-focus regions in the blurry input images, using the discrete LAPLACE operator (Sect. 3.2). Depth maps generated by a preliminary NeRF model (see Sect. 3.1) are sampled at the identified feature points to estimate the focus depth and range, i.e., the depth of field (DoF), using statistical analysis on a per-image basis. This provides an initial estimate for our Focus Frustums, which encompass visible in-focus volumes in scene space (Sect. 3.3). These frustums are iteratively expanded to increase coverage of the scene, leading to a robust training process and a fully reconstructed scene geometry (Sect. 3.4), while keeping the added volumes that contain out-of-focus regions minimal. The masks for subsequent training are computed by reprojecting the geometry inside of each Focus Frustum into its respective 2D image space, leading to a training set that only contains pixels which are mostly influenced by the density of spatial points that are encompassed by the respective Focus Frustum (Sect. 3.5). The pipeline of NeRF-FF is illustrated in Fig. 2.

### 3.1 Initial depth estimates

Defocus blur manifests depending on the distance of surfaces from the camera plane. In order to distinguish in-focus regions from out-of-focus regions, our approach leverages spatial information from the scene space. Therefore, a mapping between points in 2D image space and their corresponding points in scene space is required. We train a preliminary Instant-NGP [31] instance on the blurry input images  $i$  with the color information  $I_i \in [0, 255]^{3 \times W \times H}$ , to obtain per view depth maps  $D_i \in \mathbb{R}^{W \times H}$ . We denote the color information of a pixel that is part of image  $i$  and has position  $(x, y)$  as  $I_i(x, y)$  and call that pixel's depth  $D_i(x, y)$ . The depth maps  $D_i \in \mathbb{R}^{W \times H}$ , that are later used for initial Focus Frustum estimation (Fig. 2b) and mask generation (Fig. 2d), are derived from the rays that are deployed during the volume rendering process of Instant-NGP inference. Even though the reconstructed appearance of the preliminary Instant-NGP model suffers from the blurred image inputs, the estimated scene geometry is adequate for the purpose of generating sufficiently accurate depth maps. Note, that any approach which reliably estimates a dense depth map from the blurry input images can be applied in this stage.

### 3.2 Discrete Laplace operator as DoF indicator

In-focus image regions generally contain sharp edges, be it through textured surfaces or object boundaries, which are greatly reduced by defocus blur [3, 16]. These edges can be detected using the discrete LAPLACE operator  $\Delta$ . Pixels  $I_i(x, y)$ , that exhibit high values after applying the discrete LAPLACE operator on grayscale input images, indicate sharp edges and are therefore used as indicators for in-focus image regions—modeled as a set of indicator pixels  $P_i$  per image  $I_i$  with

$$P_i = \{(x, y) | \Delta(I_i)(x, y) > \tau\}, \quad (1)$$

where  $\Delta(I_i)(x, y)$  denotes the value of a pixel  $I_i(x, y)$  after applying the discrete LAPLACE to the grayscale version of  $I_i$  and  $\tau$  denotes a threshold for the results of the discrete LAPLACE operator that functions as a hyperparameter for NeRF-FF. A suitable value range for  $\tau$  is experimentally identified in Sect. 4.2.

### 3.3 Initial Focus Frustum estimation

Based on the focus indicator pixels  $P_i$  and the corresponding depth maps  $D_i$ , we can compute the near and far planes that are parallel to the image plane and in between which the in-focus regions are located. We call the volume that lies between these planes in the respective camera view frustum Focus Frustum. The Focus Frustum represents the visible in-

focus volume in the scene space for the corresponding image. Sharp edges often occur on object boundaries, which exhibit unstable behavior regarding their associated depth, depending on whether the depth of the foreground or background object is considered. These regions are vulnerable to even slight inaccuracies in the geometry estimation of the preliminary NeRF model. Therefore, it is likely that the sets of focus indicators  $P_i$  contain pixels that correspond to out-of-focus regions.

To mitigate the influence of these erroneous focus indicators, we compute the depths of the near and far planes  $n_i, f_i$  of the Focus Frustum  $FF_i$  based on the depth distribution of the focus indicator pixels:

$$n_i = M - \sigma_M(P_i^{<}) \quad (2)$$

$$f_i = M + \sigma_M(P_i^{>}) \quad (3)$$

where  $M$  is the median of the depth values of the focus indicator points in  $P_i$  according to the depth map  $D_i$ ,  $\sigma_\mu(X)$  denotes the standard deviation of the elements in set  $X$  in relation to  $\mu$ , and  $P_i^{<}$  ( $P_i^{>}$ ) is the subset of elements in  $P_i$  with a corresponding depth smaller (greater) than  $M$ . This follows the intuition that the depth of in-focus points in images with defocus blur follows a normal distribution. The probability of a point being in-focus is highest in the center of the distribution and decreases for closer and further points.

Based on these near and far planes, our initial Focus Frustums  $FF_i$  are defined as

$$FF_i = \{\mathbf{p} \in \mathbb{R}^3 | \mathbf{p} \cdot \mathbf{v}_i \in [n_i, f_i] \wedge K_i[R_i|t_i]\mathbf{p} \in I_i\}, \quad (4)$$

where  $\mathbf{v}_i$  denotes the forward facing vector of the camera pose that captured image  $i$  in world coordinates,  $K_i$  denotes the internal and  $[R_i|t_i]$  denotes the external camera parameters of that camera. The first term of Eq. 4 signifies the depth of the corresponding volume lying between the near and far plane defined by  $n_i, f_i$ , while the second term defines the points' localization within the view frustum, i.e., visible volume in scene space, for the camera view corresponding to image  $i$ .

Figure 2b illustrates the process of estimating Focus Frustums based on the focus indicator points  $P_i$ , the depth maps  $D_i$  and the corresponding camera views.

### 3.4 Focus Frustum refinement

Our approach provides image masks that regulate which image regions a subsequent NeRF model is trained on. The Focus Frustums dictate which depth ranges are considered for the generation of these masks by omitting pixels that get their color mainly from spatial positions outside of the Focus Frustum. Therefore, a low coverage of the scene geometry by the Focus Frustums leads to sparse inputs for these regions. Most NeRF variants suffer from significant quality

loss for such sparse inputs. A low coverage of the scene volume can be caused by an insufficient amount of Laplacian features in in-focus image regions, e.g., due to textureless surfaces, or spatial positions that are in-focus in only a few or no images. To mitigate this issue, we iteratively refine the Focus Frustums by expanding their DoF, i.e., the distance between their near and far plane. For a set  $S$  of randomly sampled points in scene space with high optical density in the preliminary NeRF—indicating this scene space is occupied by relevant scene geometry—we examine the amount of Focus Frustums they are encompassed by (see Eq. 4). We refine the depths  $n_i, f_i$  of the near and far plane for each Focus Frustum  $FF_i$ . We notate the updated near and far plane of the resulting Focus Frustums  $RFF_i$  as  $\hat{n}_i, \hat{f}_i$ . The refined near and far planes’ depths are chosen, such that the accumulated change in depth  $R$  (Eq. 5) is minimal, while any spatial position  $\mathbf{p} \in S$  is part of at least a certain fraction  $\varrho$  of the refined Focus Frustums.

$\varrho$  is a hyperparameter of NeRF-FF that trades off scene coverage against reconstruction quality of regions that are depicted in the Focus Frustums. We show the influence of different values of  $\varrho$  in Sect. 4.2.

For the calculation of  $R$ , we only consider Focus Frustums with a corresponding view frustum that encompasses the respective spatial position  $\mathbf{p}$ , i.e., only Focus Frustums are considered that could potentially encompass  $\mathbf{p}$  if an arbitrarily large change of the near or far plane is performed.

$$R = \sum_i ((n_i - \hat{n}_i) + (\hat{f}_i - f_i)) \tag{5}$$

Figure 2c illustrates the refined Focus Frustums obtained in the previous pipeline step. They were expanded to encompass relevant scene geometry that was not sufficiently captured by the initial estimates.

### 3.5 Mask generation

Given the refined Focus Frustums  $RFF_i$ , we calculate corresponding image masks  $Mask_i$  with addressable pixels  $Mask_i(x, y)$  at positions  $(x, y)$  that discriminate positions in image space depicting structures within the Focus Frustum against ones that depict structures on the outside. To generate these masks, we leverage the depth values from the depth maps  $D_i$  (Sect. 3.1). A pixel is activated in an image mask  $Mask_i$  if its corresponding depth value is part of the refined Focus Frustum’s focus range  $[\hat{n}_i, \hat{f}_i]$ :

$$Mask_i(x, y) = \begin{cases} 1, & \text{if } D_i(x, y) \in [\hat{n}_i, \hat{f}_i] \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

Note that since all pixels in the depth map are visible in their corresponding image, it is not necessary to consider the

other boundaries of the Focus Frustum. A subsequent NeRF model is trained on the blurry input images only considering pixels  $I_i(x, y)$  with  $Mask_i(x, y) = 1$ . This is illustrated in Fig. 2e. Due to the general nature of the results of NeRF-FF in the form of these masks, NeRF-FF is compatible with most NeRF variants.

## 4 Experiments

### 4.1 Implementation details

*Training.* NeRF-FF provides image masks for a subsequent training process with a NeRF variant. In our experiments, we evaluate NeRF-FF in combination with Instant-NGP (NeRF-FF+iNGP). Therefore, we employ Instant-NGP [31] as a preliminary and a subsequent NeRF model. Note that current state-of-the-art methods are not easily usable in combination with Instant-NGP because Instant-NGP employs custom CUDA kernels, which are hard to integrate with existing strategies [37]. The preliminary NeRF is trained for 2,000 iterations and the subsequent NeRF for 4,000 iterations on a single NVIDIA RTX3090. We use the standard Instant-NGP training parameters.

*Datasets and evaluation.*

We perform our experiments on the dataset introduced by Ma et al. [26], which provides 5 synthetic and 10 real-world scenes containing images suffering from defocus blur. Furthermore, we present the capabilities of NeRF-FF on the dataset presented by Wu et al. [59], which contains 6 real-world scenes.

For the synthetic scenes of Ma et al. [26], we train one model per scene on the training split. We compare the synthesized novel views for the views in the evaluation splits that are rendered using that model with the respective ground truth image. For our analysis, we consider the median of the results per scene. Each real-world scene of Ma et al. [26] contains a set of in-focus reference images. For each provided reference image, we train a model on any image in the respective scene dataset except the examined reference image, comparing the synthesized image to the reference. For each scene, we report the median of these results in Table 1. The average results are reported in Table 3.

For the real-world scenes of Wu et al. [59], a triplet of images consisting of two out-of-focus and one in-focus image is provided per view. Analogous to the evaluation method of Wu et al., we train a model on the out-of-focus images from  $\frac{8}{9}$  of the views and compare the synthesized images of the remaining  $\frac{1}{9}$  of scenes to their respective in-focus reference image. For each scene, we evaluate different splits until the union of these splits’ evaluation sets contains at least 20 images. Per scene we report the median of these results in Table 2.

**Table 1** Quantitative results for real-world scenes from the dataset by Ma et al. [26]

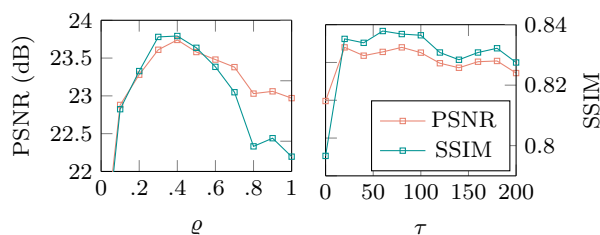
	Cake			Caps			Cisco			Cupcake			Coral		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Naive NeRF	24.42	0.7210	0.2250	22.73	0.6312	0.2801	20.72	0.7217	0.1256	21.88	0.6809	0.2155	19.81	0.5658	0.2689
iNGP	24.56	0.8032	0.2311	23.53	<b>0.7801</b>	0.2627	20.99	<b>0.8429</b>	.1317	22.33	<b>0.7791</b>	0.2283	<b>20.58</b>	<b>0.7572</b>	0.2808
Deblur-NeRF	<b>26.27</b>	<b>0.7800</b>	<b>0.1282</b>	23.87	0.7128	0.1612	<b>20.83</b>	0.7270	<b>0.0868</b>	22.26	0.7219	<b>0.1160</b>	19.85	0.5999	<b>0.1214</b>
DP-NeRF	26.16	0.7781	<b>0.1267</b>	23.95	0.7122	<b>0.1430</b>	<b>20.73</b>	0.7260	<b>0.0840</b>	22.80	0.7409	<b>0.0960</b>	20.11	0.6107	<b>0.1178</b>
PDRF	<b>27.01</b>	0.7756	0.1408	<b>24.09</b>	0.6812	0.1749	20.62	0.7086	0.1128	<b>23.04</b>	0.7445	0.1205	19.60	0.5894	0.1381
NeRF-FF	25.41	<b>0.8337</b>	0.1617	<b>25.18</b>	<b>0.8264</b>	<b>0.1516</b>	<b>20.73</b>	<b>0.8348</b>	0.1004	<b>23.07</b>	<b>0.8121</b>	0.1553	<b>20.53</b>	<b>0.8184</b>	0.1682
	Daisy			Sausage			Seal			Tools					
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Naive NeRF	25.02	0.7581	0.2315	22.74	0.6203	0.2621	17.79	0.4830	0.2789	22.79	0.6267	0.2680	26.08	0.8523	0.1547
iNGP	25.57	<b>0.8347</b>	0.2113	22.69	<b>0.7607</b>	0.2867	<b>20.49</b>	<b>0.7873</b>	0.2424	23.23	0.7204	0.2727	25.13	0.8799	0.1869
Deblur-NeRF	26.21	0.7987	<b>0.1271</b>	23.52	0.6870	0.1208	18.01	0.4998	0.1796	<b>26.04</b>	0.7773	<b>0.1048</b>	27.81	0.8949	<b>0.0610</b>
DP-NeRF	<b>26.75</b>	0.8136	<b>0.1035</b>	<b>23.79</b>	0.6971	<b>0.1075</b>	18.35	0.5443	<b>0.1473</b>	25.95	0.7779	<b>0.1026</b>	28.07	0.8980	<b>0.0539</b>
PDRF	<b>26.27</b>	0.7980	0.1506	<b>24.39</b>	0.7611	<b>0.1162</b>	18.90	0.5675	<b>0.1690</b>	<b>26.29</b>	<b>0.8154</b>	0.1146	<b>28.18</b>	<b>0.9139</b>	0.0815
NeRF-FF	26.04	<b>0.8547</b>	0.1567	22.58	<b>0.8098</b>	0.1394	<b>21.17</b>	<b>0.8231</b>	0.1831	25.74	<b>0.8349</b>	0.1491	<b>26.52</b>	<b>0.9012</b>	0.1201

Best result is indicated by **bold** and second-best result by *bold-italic* text

**Table 2** Quantitative results for real-world scenes from the dataset by Wu et al. [59]

	Amiya			Camera			Desk			Kendo		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Naive NeRF	26.92	0.9092	0.1633	25.59	0.8862	0.1574	<b>29.62</b>	0.9374	0.1099	18.46	0.6643	0.2906
iNGP	<b>28.44</b>	0.9091	<b>0.1164</b>	26.94	0.8804	0.1487	28.75	0.9317	<b>0.0660</b>	<b>22.36</b>	<b>0.8223</b>	<b>0.2217</b>
DoF-NeRF	28.31	<b>0.9289</b>	0.1370	<b>27.71</b>	<b>0.9134</b>	<b>0.1259</b>	<b>30.55</b>	<b>0.9426</b>	0.1047	19.31	0.6970	0.2773
NeRF-FF (Ours)	<b>29.74</b>	<b>0.9296</b>	<b>0.0654</b>	<b>29.58</b>	<b>0.9316</b>	<b>0.0471</b>	29.19	<b>0.9396</b>	<b>0.0487</b>	<b>23.68</b>	<b>0.8796</b>	<b>0.0913</b>
Plant	Shelf			Average								
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓			
Naive NeRF	28.27	0.8961	0.1581	31.55	<b>0.9515</b>	0.0811	26.74	0.8741	0.1601			
iNGP	29.50	0.9032	0.1383	30.76	0.9369	<b>0.0690</b>	27.79	<b>0.8973</b>	<b>0.1267</b>			
DoF-NeRF	<b>30.32</b>	<b>0.9290</b>	<b>0.1178</b>	<b>32.00</b>	<b>0.9565</b>	0.0723	<b>28.03</b>	0.8946	0.1392			
NeRF-FF (Ours)	<b>30.94</b>	<b>0.9364</b>	<b>0.0536</b>	<b>31.78</b>	0.9481	<b>0.0380</b>	<b>29.15</b>	<b>0.9275</b>	<b>0.0573</b>			

Best result is indicated by **bold** and second-best result by *bold-italic* text



**Fig. 3** Comparison of PSNR and SSIM of NeRF-FF + iNGP for different values of the hyperparameters  $\rho$  and  $\tau$

## 4.2 Hyperparameter optimization and ablation

Our approach relies on a set of hyperparameters ( $\tau$ ,  $\rho$ ). The threshold  $\tau$  determines which values, that result from the application of the discrete LAPLACE operator, are considered indicators for in-focus regions (see Sect. 3.2).  $\rho$  indicates the minimum fraction of Focus Frustums each randomly sampled scene point should be encompassed by after the FF refinement step (Sect. 3.4). This hyperparameter defines the trade-off between the sharpness of the image regions that are used for training and coverage of the scene.

In this section, we identify reasonable values for these parameters by comparing resulting images regarding their visual quality measured by the peak signal-to-noise ratio (PSNR) and the structural similarity index (SSIM) [66]. The experiments are conducted on the blurred scenes from the real dataset by Ma et al. [26].

To identify a suitable value for  $\rho$ , we consider values  $\rho \in [0, 0.1, \dots, 1.0]$ . Note that  $\rho = 0$  corresponds to omission of the Focus Frustum Refinement step and  $\rho = 1$  is equal to the training of the subsequent NeRF model without applying NeRF-FF beforehand. For this experiment, we set  $\tau = 80$ , which constitutes an educated guess that turns out to be optimal for the dataset at hand. Figure 3 shows the results of that evaluation, which indicate that  $\rho = 0.4$  results in best performance for PSNR and SSIM, while significantly lower or higher values for  $\rho$  lead to dramatically decreased performance. Therefore, we chose  $\rho = 0.4$  for our subsequent experiments.

Figure 3 also shows the results for different values of  $\tau$ , which determines the threshold for the results of the discrete Laplace operator, employed for identifying DoF indicator pixels. The experiment is conducted with  $\rho = 0.4$ , since we have established it as an optimal parameter. The results indicate that low values of  $\tau$  lead to bad performance. In these cases, even weak signals are considered indicators for in-focus regions, leading to feature points in regions that are essentially out-of-focus but exhibit strong texture. On the other hand, high values of  $\tau$  lead to slowly decreasing performance due to the fact that even some sharp edges in in-focus regions are not considered relevant. The decline for higher values of  $\tau$ , however, is small in comparison with low values

of  $\tau$ . This is most likely attributed to the fact that the indicator points are only used for the initial Focus Frustum estimates, which are then further refined. Smaller initial Focus Frustums, which occur for high values of  $\tau$ , are then alleviated by the subsequent Focus Frustum refinement. Following the results of this experiment, the value of  $\tau = 80$  is identified as the optimal choice for subsequent experiments, as it maximizes PSNR and SSIM values.

## 4.3 Novel view synthesis from blurry inputs

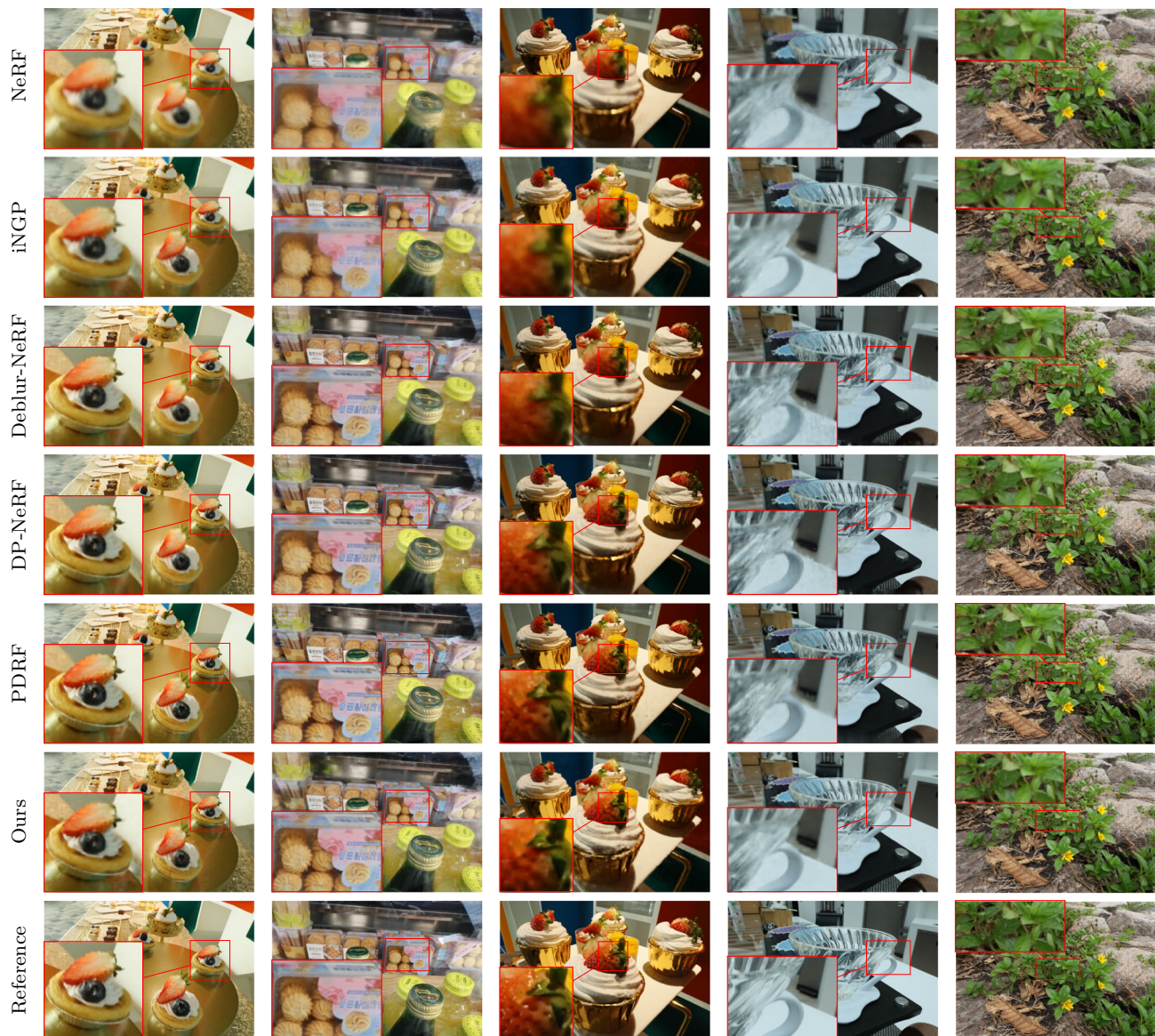
*Evaluation Methodology.* In this section, we summarize the results of our proposed approach in combination with Instant-NGP (NeRF-FF + iNGP) on the synthetic and real-world dataset of Ma et al. [26] and the dataset proposed by Wu et al. [59]. Similar to previous works, we conduct our quantitative analysis of the examined approaches by comparing the rendered images to their respective reference image regarding their PSNR, SSIM, and LPIPS. *Real-World Comparison.*

Table 3 and Figs. 4 and 5 show that our model produces results that are comparable to SOTA methods. We compare our approach to Deblur-NeRF [26], DP-NeRF [20] and PDRF [37] that leverage 3D scene information for the estimation of a blur kernel while simultaneously incorporating dynamic ray generation. Ma et al. [26] show that these methods significantly outperform the process of deblurring single images beforehand by using strategies like KPAC [45]. Our results show that NeRF-FF + iNGP leads to images with PSNR values that are on par with state-of-the-art work. Regarding SSIM, our approach consistently outperforms these methods by a fair margin. The results indicate that Instant-NGP outperforms these methods regarding SSIM as well, suggesting that these improvements in structural similarity can be attributed to the usage of that NeRF variant. However, our approach trails the visual quality of the reference deblurring strategies regarding its LPIPS score by a slight margin. While this indicates inferior quality regarding human perception, we consider this difference in rendering quality negligible in comparison with the achieved acceler-

**Table 3** Average results for synthetic and real-world scenes from the dataset by Ma et al. [26]

	Synthetic			Real-World		
	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
Naive NeRF	25.93	0.7791	0.2303	22.40	0.6661	0.2310
iNGP	25.21	0.8240	0.2164	22.91	<b>0.7945</b>	0.2334
Deblur-NeRF	28.37	<b>0.8527</b>	<b>0.1188</b>	23.47	0.7199	<b>0.1207</b>
DP-NeRF	<b>29.33</b>	<b>0.8713</b>	<b>0.0987</b>	23.67	0.7299	<b>0.1082</b>
PDRF	<b>29.92</b>	0.8483	0.1057	<b>23.84</b>	0.7355	0.1319
<b>NeRF-FF</b>	26.00	0.8474	0.1689	<b>23.70</b>	<b>0.8349</b>	0.1485

Best result is indicated by **bold** and second-best result by *bold-italic* text



**Fig. 4** Qualitative comparison of NeRF-FF + iNGP (ours) with the current state of the art on the real-world defocus dataset of Ma et al. [26]. The depicted scenes are *cake*, *caps*, *cupcake*, *cups* and *daisy* (left to right)

ation. Our approach only requires 45 s of training time on average, which constitutes an acceleration of two orders of magnitude in comparison with state-of-the-art methods. This training time encompasses both the execution of NeRF-FF, including the preliminary NeRF training, as well as the training of the Instant-NGP model.

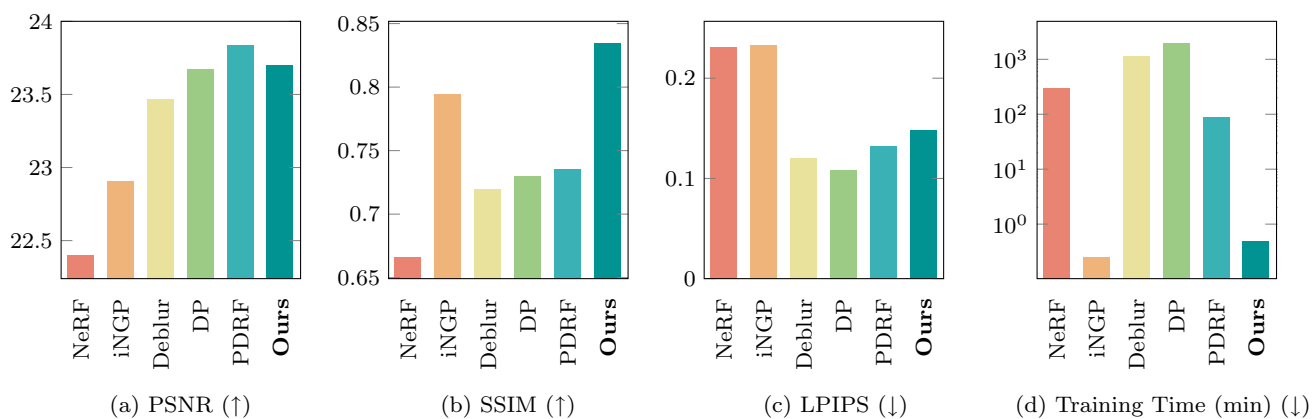
The results of our experiment on the dataset by Wu et al. [59] are illustrated in Table 2 and Fig. 6. These results indicate that we consistently produce images of better quality than DoF-NeRF [59], which physically models defocus blur by estimating the camera optics—especially the lens—as a multilayer perceptron. *Synthetic Comparison.* Table 3 shows the results of NeRF-FF+iNGP on the synthetic dataset of Ma et al. [26]. On this dataset, our approach underperforms in comparison with the state-of-the-art approaches. An analysis

of the intermediary pipeline stages shows that this is caused by a low-quality dense reconstruction of the synthetic scenes leading to depth maps of inferior quality. This is caused by fully blurred images in the dataset. We discuss this issue further in Sect. 5.

#### 4.4 Compatibility with other technologies

Table 4 illustrates the results of NeRF-FF when applied in combination with different volumetric novel view synthesis strategies. For these experiments, we employed NeRF-FF in combination with *nerfacto*—a NeRF variant—and *splatfacto*, which is based on 3D Gaussian Splatting [18]. These strategies are both implemented in Nerfstudio [50].





**Fig. 5** Comparison of average PSNR, SSIM, LPIPS and training times between NeRF+iNGP and state-of-the-art methods on the real defocus dataset by Ma et al. [26]. The results indicate that NeRF+iNGP

achieves comparable visual quality while decreasing the required training time by at least two orders of magnitude. For ↑ (↓), higher (lower) values indicate better results



**Fig. 6** Qualitative comparison of DoF-NeRF [59] and NeRF-FF+iNGP (ours) on the real-world dataset provided by Wu et al. [59] on the scenes kendo (top), camera (middle) and amiya (bottom)

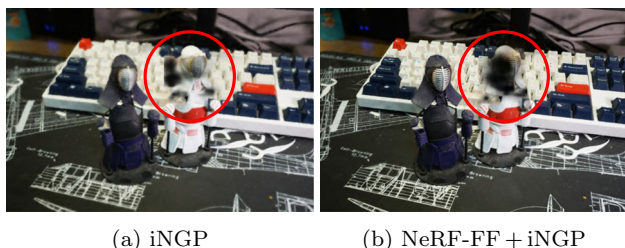
The applied evaluation strategy is analogous to the one presented in Sect. 4.1. The results show that applying the masks generated by NeRF-FF to the input images of *nerfacto* and *splatfacto* produces higher-quality rendering results, manifesting in improvements in the observed metrics PSNR, SSIM and LPIPS. The impact of NeRF-FF on the training duration is negligible. These results further underline that NeRF-FF is compatible with a multitude of other training time optimized NeRF or NeRF-like strategies.

### 5 Limitations and future work

**Dense Reconstruction Quality.** Low quality of the preliminary NeRF’s dense reconstruction subsequently reduces the quality of image masks and therefore the overall result of NeRF-FF and a subsequent NeRF model significantly. This inferior performance is caused by mismatches between in-focus indicators and the respective spatial position of the scene geometry that mainly contributes to this pixel’s rendered color. These mismatches lead to erroneous depth estimates of the in-focus indicators, negatively influencing the initial Focus Frustum estimates. An analysis of

**Table 4** Average results for real-world scenes from the dataset by Wu et al. [59] for other runtime-optimized methods in combination with NeRF-FF. For each scene, *nerfacto* [50] is trained for 6,000 iterations, *splatfacto* [18, 50] for 20,000 iterations

	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	Training (s)
nerfacto	28.33	0.9177	0.0867	94.15
nerfacto+NeRF-FF	28.73	0.9348	0.0485	105.9
splatfacto	28.78	0.9250	0.1017	199.6
splatfacto+NeRF-FF	29.83	0.9417	0.0525	183.3



**Fig. 7** Floater artifacts manifesting in the volumetric scene representation of Instant-NGP (a). This results in aberrated depth maps, impurifying the generated in-focus masks. As a result, the floater remains in the trained model augmented by NeRF-FF (b)

the occurring failure cases during our experiments shows that low-quality dense reconstructions often occur when the inputs contain images without in-focus regions which leads to floater artifacts—scene volume with high estimated optical density with no counterpart in the ground truth scene—in the reconstructed scene. Therefore, NeRF-FF is not suited for datasets containing fully blurred images. Future work could enhance the robustness of NeRF-FF by automatically pruning these images from the training dataset.

**Floater artifacts.** Some results of NeRF-FF+iNGP suffer from floater artifacts in the subsequent reconstruction of the scene based on the masked images (Fig. 7). These floater artifacts are also prevalent in unmasked Instant-NGP models and reduce image quality depending on the camera position. Recently, approaches to mitigate the influence of floater artifacts either by changing the applied training process [2] or by removing them after the training process [12, 57, 58] have been discussed. Integrating these strategies into the training process could further improve the visual quality of the results.

## 6 Conclusion

In this paper, we propose NeRF-FF, a plug-in method that enables the processing of partially defocus blurred input images for a multitude of NeRF variants. Our method leverages the discrete LAPLACE operator to detect in-focus regions in images to estimate in-focus volumes—Focus Frustums (FF)—on a per-image base. By iteratively expanding these Focus Frustums, our approach reaches full scene coverage

while maintaining high visual quality. NeRF-FF is compatible with accelerated NeRF variants like Instant-NGP, offering qualitative results comparable to SoTA methods like Deblur-NeRF [26], DP-NeRF [20] and PDRF [37] while reducing training times below 1 min on end-consumer hardware. This corresponds to a relative speed-up of two orders of magnitude.

**Supplementary Information** The online version contains supplementary material available at <https://doi.org/10.1007/s00371-024-03507-y>.

**Acknowledgements** We thank the anonymous reviewers for their guidance and encouraging feedback.

**Author Contributions** T.W. contributed to conceptualization, investigation, methodology, writing—original draft, and writing—review and editing. A.R. was involved in investigation, methodology, experimental design, writing—original draft, and writing—review and editing, and provided software. M.B. contributed to Laplacian operator and writing—review and editing. V.K., A.K., and D.W.F. were involved in guidance and feedback during the research process and writing—review and editing.

**Funding** Open Access funding enabled and organized by Projekt DEAL.

**Data availability** No datasets were generated or analyzed during the current study.

## Declarations

**Conflict of interest** The authors declare no conflict of interest.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Barron, J.T., Mildenhall, B., Tancik, M., Hedman, P., Martin-Brualla, R., Srinivasan, P.P.: Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In: *IEEE/CVF ICCV Conference Proceedings*, pp. 5855–5864 (2021)
2. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In: *IEEE/CVF CVPR Conference Proceedings*, pp. 5470–5479 (2022)
3. von Buelow, M., Tausch, R., Schurig, M., Knauth, V., Wirth, T., Guthe, S., Santos, P., Fellner, D.W.: Depth-of-field segmentation for near-lossless image compression and 3d reconstruction. *J. Comput. Cult. Herit.* (2022). <https://doi.org/10.1145/3500924>
4. Chen, A., Xu, Z., Geiger, A., Yu, J., Su, H.: Tensorf: Tensorial radiance fields. In: *European Conference on Computer Vision*, pp. 333–350. Springer (2022)
5. Chen, J.K., Lyu, J., Wang, Y.X.: Neuraleditor: Editing neural radiance fields via manipulating point clouds. In: *IEEE/CVF CVPR Conference Proceedings*, pp. 12,439–12,448 (2023)
6. Chen, X., Zhang, Q., Li, X., Chen, Y., Feng, Y., Wang, X., Wang, J.: Hallucinated neural radiance fields in the wild. In: *IEEE/CVF CVPR Conference Proceedings*, pp. 12–943–12952 (2022)
7. Chen, Z., Funkhouser, T., Hedman, P., Tagliasacchi, A.: Mobilenerf: Exploiting the polygon rasterization pipeline for efficient neural field rendering on mobile architectures. In: *IEEE/CVF CVPR Conference Proceedings*, pp. 16569–16578 (2023)
8. Dai, P., Zhang, Y., Yu, X., Lyu, X., Qi, X.: Hybrid neural rendering for large-scale scenes with motion blur. In: *IEEE/CVF CVPR Conference Proceedings*, pp. 154–164 (2023)
9. Deng, K., Liu, A., Zhu, J.Y., Ramanan, D.: Depth-supervised nerf: Fewer views and faster training for free. In: *IEEE/CVF CVPR Conference Proceedings*, pp. 12882–12891 (2022)
10. Fridovich-Keil, S., Yu, A., Tancik, M., Chen, Q., Recht, B., Kanazawa, A.: Plenoxels: Radiance fields without neural networks. In: *IEEE/CVF CVPR Conference Proceedings*, pp. 5501–5510 (2022)
11. Huang, X., Zhang, Q., Feng, Y., Li, H., Wang, X., Wang, Q.: Hdrnerf: High dynamic range neural radiance fields. In: *IEEE/CVF CVPR Conference Proceedings*, pp. 18398–18408 (2022)
12. Jambon, C., Kerbl, B., Kopanas, G., Diolatzis, S., Drettakis, G., Leimkühler, T.: Nerfshop: Interactive editing of neural radiance fields. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* **6**(1) (2023)
13. Jiang, S., Jiang, H., Wang, Z., Luo, H., Chen, W., Xu, L.: Humanerf: Generating human radiance fields with explicit priors. In: *IEEE/CVF CVPR Conference Proceedings*, pp. 12543–12554 (2023)
14. Jun-Seong, K., Yu-Ji, K., Ye-Bin, M., Oh, T.H.: Hdr-plenoxels: Self-calibrating high dynamic range radiance fields. In: *European Conference on Computer Vision*, pp. 384–401. Springer (2022)
15. Kajiya, J.T., Von Herzen, B.P.: Ray tracing volume densities. *ACM SIGGRAPH Comput. Graph.* **18**(3), 165–174 (1984)
16. Karaali, A., Jung, C.R.: Edge-based defocus blur estimation with adaptive scale selection. *IEEE Trans. Image Process.* **27**(3), 1126–1137 (2018). <https://doi.org/10.1109/TIP.2017.2771563>
17. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.* **42**(4), 1–14 (2023)
18. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.* **42**(4), 1–12 (2023)
19. Lazova, V., Guzov, V., Olszewski, K., Tulyakov, S., Pons-Moll, G.: Control-nerf: Editable feature volumes for scene rendering and manipulation. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 4340–4350 (2023)
20. Lee, D., Lee, M., Shin, C., Lee, S.: Dp-nerf: Deblurred neural radiance field with physical scene priors. In: *IEEE/CVF CVPR Conference Proceedings*, pp. 12386–12396 (2023)
21. Lee, D., Oh, J., Rim, J., Cho, S., Lee, K.M.: Exblurf: Efficient radiance fields for extreme motion blurred images. In: *IEEE/CVF ICCV Conference Proceedings*, pp. 17639–17648 (2023)
22. Li, T., Slavcheva, M., Zollhoefer, M., Green, S., Lassner, C., Kim, C., Schmidt, T., Lovegrove, S., Goesele, M., Newcombe, R., et al.: Neural 3d video synthesis from multi-view video. In: *IEEE/CVF CVPR Conference Proceedings*, pp. 5521–5531 (2022)
23. Lin, C.H., Ma, W.C., Torralba, A., Lucey, S.: Barf: Bundle-adjusting neural radiance fields. In: *IEEE/CVF ICCV Conference Proceedings*, pp. 5741–5751 (2021)
24. Liu, S., Zhang, X., Zhang, Z., Zhang, R., Zhu, J.Y., Russell, B.: Editing conditional radiance fields. In: *IEEE/CVF ICCV Conference Proceedings*, pp. 5773–5783 (2021)
25. Lombardi, S., Simon, T., Schwartz, G., Zollhoefer, M., Sheikh, Y., Saragih, J.: Mixture of volumetric primitives for efficient neural rendering. *ACM Trans. Graph.* **40**(4), 1–13 (2021)
26. Ma, L., Li, X., Liao, J., Zhang, Q., Wang, X., Wang, J., Sander, P.V.: Deblur-nerf: Neural radiance fields from blurry images. In: *IEEE/CVF CVPR Conference Proceedings*, pp. 12861–12870 (2022)
27. Martin-Brualla, R., Radwan, N., Sajjadi, M.S., Barron, J.T., Dosovitskiy, A., Duckworth, D.: Nerf in the wild: Neural radiance fields for unconstrained photo collections. In: *IEEE/CVF CVPR Conference Proceedings*, pp. 7210–7219 (2021)
28. Meng, Q., Chen, A., Luo, H., Wu, M., Su, H., Xu, L., He, X., Yu, J.: Gnerf: Gan-based neural radiance field without posed camera. In: *IEEE/CVF ICCV Conference Proceedings*, pp. 6351–6361 (2021)
29. Mildenhall, B., Hedman, P., Martin-Brualla, R., Srinivasan, P.P., Barron, J.T.: Nerf in the dark: High dynamic range view synthesis from noisy raw images. In: *IEEE/CVF CVPR Conference Proceedings*, pp. 16190–16199 (2022)
30. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: *ECCV* (2020)
31. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.* **41**(4), 1–15 (2022)
32. Munkberg, J., Hasselgren, J., Shen, T., Gao, J., Chen, W., Evans, A., Müller, T., Fidler, S.: Extracting triangular 3d models, materials, and lighting from images. In: *IEEE/CVF CVPR Conference Proceedings*, pp. 8280–8290 (2022)
33. Niemeyer, M., Barron, J.T., Mildenhall, B., Sajjadi, M.S., Geiger, A., Radwan, N.: Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In: *IEEE/CVF CVPR Conference Proceedings*, pp. 5480–5490 (2022)
34. Niemeyer, M., Geiger, A.: Giraffe: Representing scenes as compositional generative neural feature fields. In: *IEEE/CVF CVPR Conference Proceedings*, pp. 11453–11464 (2021)
35. Park, K., Sinha, U., Barron, J.T., Bouaziz, S., Goldman, D.B., Seitz, S.M., Martin-Brualla, R.: Nerfies: Deformable neural radiance fields. In: *IEEE/CVF ICCV Conference Proceedings*, pp. 5865–5874 (2021)
36. Park, K., Sinha, U., Hedman, P., Barron, J.T., Bouaziz, S., Goldman, D.B., Martin-Brualla, R., Seitz, S.M.: Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *arXiv preprint arXiv:2106.13228* (2021)
37. Peng, C., Chellappa, R.: Pdrf: progressively deblurring radiance field for fast scene reconstruction from blurry images. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, pp. 2029–2037 (2023)
38. Peng, S., Dong, J., Wang, Q., Zhang, S., Shuai, Q., Zhou, X., Bao, H.: Animatable neural radiance fields for modeling dynamic human

- bodies. In: IEEE/CVF ICCV Conference Proceedings, pp. 14314–14323 (2021)
39. Pumarola, A., Corona, E., Pons-Moll, G., Moreno-Noguer, F.: D-nerf: Neural radiance fields for dynamic scenes. In: IEEE/CVF CVPR Conference Proceedings, pp. 10318–10327 (2021)
  40. Qi, Y., Zhu, L., Zhang, Y., Li, J.: E2nerf: Event enhanced neural radiance fields from blurry images. In: IEEE/CVF ICCV Conference Proceedings, pp. 13254–13264 (2023)
  41. Rebain, D., Matthews, M., Yi, K.M., Lagun, D., Tagliasacchi, A.: Lolnerf: Learn from one look. In: IEEE/CVF CVPR Conference Proceedings, pp. 1558–1567 (2022)
  42. Reiser, C., Peng, S., Liao, Y., Geiger, A.: Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In: IEEE/CVF ICCV Conference Proceedings, pp. 14335–14345 (2021)
  43. Rematas, K., Liu, A., Srinivasan, P.P., Barron, J.T., Tagliasacchi, A., Funkhouser, T., Ferrari, V.: Urban radiance fields. In: IEEE/CVF CVPR Conference Proceedings, pp. 12932–12942 (2022)
  44. Schwarz, K., Liao, Y., Niemeyer, M., Geiger, A.: Graf: Generative radiance fields for 3d-aware image synthesis. *Adv. Neural. Inf. Process. Syst.* **33**, 20154–20166 (2020)
  45. Son, H., Lee, J., Cho, S., Lee, S.: Single image defocus deblurring using kernel-sharing parallel atrous convolutions. In: IEEE/CVF ICCV Conference Proceedings, pp. 2642–2650 (2021)
  46. Srinivasan, P.P., Deng, B., Zhang, X., Tancik, M., Mildenhall, B., Barron, J.T.: Nerv: Neural reflectance and visibility fields for relighting and view synthesis. In: IEEE/CVF CVPR Conference Proceedings, pp. 7495–7504 (2021)
  47. Su, S.-Y., Yu, F., Zollhöfer, M., Rhodin, H.: A-nerf: Articulated neural radiance fields for learning human shape, appearance, and pose. *Adv. Neural. Inf. Process. Syst.* **34**, 12278–12291 (2021)
  48. Sun, C., Sun, M., Chen, H.T.: Direct voxel grid optimization: Superfast convergence for radiance fields reconstruction. In: IEEE/CVF CVPR Conference Proceedings, pp. 5459–5469 (2022)
  49. Tancik, M., Casser, V., Yan, X., Pradhan, S., Mildenhall, B., Srinivasan, P.P., Barron, J.T., Kretschmar, H.: Block-nerf: Scalable large scene neural view synthesis. In: IEEE/CVF CVPR Conference Proceedings, pp. 8248–8258 (2022)
  50. Tancik, M., Weber, E., Ng, E., Li, R., Yi, B., Wang, T., Kristoffersen, A., Austin, J., Salahi, K., Ahuja, A., et al.: Nerfstudio: A modular framework for neural radiance field development. In: ACM SIGGRAPH 2023 Conference Proceedings, pp. 1–12 (2023)
  51. Tretschk, E., Tewari, A., Golyanik, V., Zollhöfer, M., Lassner, C., Theobalt, C.: Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video. In: IEEE/CVF ICCV Conference Proceedings, pp. 12959–12970 (2021)
  52. Wang, L., Zhang, J., Liu, X., Zhao, F., Zhang, Y., Zhang, Y., Wu, M., Yu, J., Xu, L.: Fourier plenoctrees for dynamic radiance field rendering in real-time. In: IEEE/CVF CVPR conference proceedings, pp. 13524–13534 (2022)
  53. Wang, P., Zhao, L., Ma, R., Liu, P.: Bad-nerf: Bundle adjusted deblur neural radiance fields. In: IEEE/CVF CVPR Conference Proceedings, pp. 4170–4179 (2023)
  54. Wang, Z., Shen, T., Gao, J., Huang, S., Munkberg, J., Hasselgren, J., Gojcic, Z., Chen, W., Fidler, S.: Neural fields meet explicit geometric representations for inverse rendering of urban scenes. In: The IEEE Conference on IEEE/CVF CVPR Conference Proceedings (CVPR) (2023)
  55. Wang, Z., Shen, T., Gao, J., Huang, S., Munkberg, J., Hasselgren, J., Gojcic, Z., Chen, W., Fidler, S.: Neural fields meet explicit geometric representations for inverse rendering of urban scenes. In: IEEE/CVF CVPR Conference Proceedings, pp. 8370–8380 (2023)
  56. Wang, Z., Wu, S., Xie, W., Chen, M., Prisacariu, V.A.: Nerf-: Neural radiance fields without known camera parameters. arXiv preprint [arXiv:2102.07064](https://arxiv.org/abs/2102.07064) (2021)
  57. Warburg, F., Weber, E., Tancik, M., Holynski, A., Kanazawa, A.: Nerfbusters: Removing ghostly artifacts from casually captured nerfs. arXiv preprint [arXiv:2304.10532](https://arxiv.org/abs/2304.10532) (2023)
  58. Wirth, T., Rak, A., Knauthe, V., Fellner, D.W.: A post processing technique to automatically remove floater artifacts in neural radiance fields. In: Computer Graphics Forum, p. e14977. Wiley Online Library (2023)
  59. Wu, Z., Li, X., Peng, J., Lu, H., Cao, Z., Zhong, W.: Dof-nerf: Depth-of-field meets neural radiance fields. In: Proceedings of the 30th ACM International Conference on Multimedia, pp. 1718–1729 (2022)
  60. Yang, B., Bao, C., Zeng, J., Bao, H., Zhang, Y., Cui, Z., Zhang, G.: Neumesh: Learning disentangled neural mesh-based implicit field for geometry and texture editing. In: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVI, pp. 597–614. Springer (2022)
  61. Yu, A., Li, R., Tancik, M., Li, H., Ng, R., Kanazawa, A.: Plenoctrees for real-time rendering of neural radiance fields. In: IEEE/CVF ICCV Conference Proceedings, pp. 5752–5761 (2021)
  62. Yu, A., Ye, V., Tancik, M., Kanazawa, A.: pixelnerf: Neural radiance fields from one or few images. In: IEEE/CVF CVPR Conference Proceedings, pp. 4578–4587 (2021)
  63. Yuan, Y.J., Sun, Y.T., Lai, Y.K., Ma, Y., Jia, R., Gao, L.: Nerf-editing: geometry editing of neural radiance fields. In: IEEE/CVF CVPR Conference Proceedings, pp. 18353–18364 (2022)
  64. Zeng, C., Chen, G., Dong, Y., Peers, P., Wu, H., Tong, X.: Relighting neural radiance fields with shadow and highlight hints. In: ACM SIGGRAPH 2023 Conference Proceedings, pp. 1–11 (2023)
  65. Zhang, J., Liu, X., Ye, X., Zhao, F., Zhang, Y., Wu, M., Zhang, Y., Xu, L., Yu, J.: Editable free-viewpoint video using a layered neural representation. *ACM Trans. Graph.* **40**(4), 1–18 (2021)
  66. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE Conference on IEEE/CVF CVPR Conference Proceedings, pp. 586–595 (2018)
  67. Zhang, X., Srinivasan, P.P., Deng, B., Debevec, P., Freeman, W.T., Barron, J.T.: Nerfactor: Neural factorization of shape and reflectance under an unknown illumination. *ACM Trans. Graph.* **40**(6), 1–18 (2021)
  68. Zhao, F., Yang, W., Zhang, J., Lin, P., Zhang, Y., Yu, J., Xu, L.: Humannerf: Efficiently generated human radiance field from sparse inputs. In: IEEE/CVF CVPR Conference Proceedings, pp. 7743–7753 (2022)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Tristan Wirth** has received his MSc in Computer Science at TU Darmstadt in 2020. He was selected for the Software Campus Program 2021 sponsored by the German Federal Ministry of Education and Research. As a PhD candidate at the Interactive Graphics Systems Group at TU Darmstadt, he follows his research interests in the areas of neural volume rendering, i.e., neural radiance fields and 3D Gaussian splatting, in the context of scene composition and augmented reality.



**Arne Rak** received his MSc in Computer Science at TU Darmstadt in 2021. After researching GPU accelerated flood simulations in the Numerical Methods in Mechanical Engineering Group of TU Darmstadt, he joined the Interactive Graphics Systems Group in 2022, continuing his research on various accelerated Mechanical Engineering tasks.



**Max von Buelow** defended his PhD thesis on performance visualization of graphics hardware at the Technical University of Darmstadt in 2024.



**Volker Knauthe** received his MSc in Computer Science at TU Darmstadt in 2019. He was selected for the Software Campus Program 2020 sponsored by the German Federal Ministry of Education and Research. As a PhD candidate at the Interactive Graphics Systems Group at TU Darmstadt, he follows his research interests in the areas of transparent structures and vision systems for material technology.



**Arjan Kuijper** holds a chair in Mathematical and Applied Visual Computing at TU Darmstadt and is a member of the management of Fraunhofer IGD, responsible for scientific dissemination. He obtained a MSc in applied mathematics from Twente University and a PhD from Utrecht University, both in the Netherlands. He was assistant research professor at the IT University of Copenhagen, Denmark, and senior researcher at RICAM in Linz, Austria. He obtained his habilitation degree from TU Graz, Austria. He is author of over 400 peer-reviewed publications, associate editor for CVIU, PR, and TVCJ, President of the International Association for Pattern Recognition (IAPR). He serves as reviewer for many journals and conferences, and as program committee member and organizer of conferences. His research interests cover all aspects of mathematics-based methods for computer vision, biometry, graphics, imaging, pattern recognition, interaction, and visualization.



**Dieter W. Fellner** is professor of computer science at TU Darmstadt, Germany, and Director of the Fraunhofer Institute of Computer Graphics (IGD) at the same location. He is also professor at TU Graz, Austria, and CEO of Fraunhofer Austria. His research activities over the last years covered efficient rendering and visualization algorithms, generative and reconstructive modeling, virtual and augmented reality, graphical aspects of internet-based multimedia information systems, cultural heritage and digital libraries as well as visual healthcare technologies. He is a member of the Academia Europaea and the National Academy of Science and Engineering (acatech) and a Fellow of the German Computer Society (GI) and of the EUROGRAPHICS Association.