Inverting the Imaging Process by Learning an Implicit Camera Model

Xin Huang[1]*, Qi Zhang[2]†, Ying Feng[2], Hongdong Li[3], Qing Wang[1]†

[1] School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China
[2] Tencent AI Lab [3] Australian National University

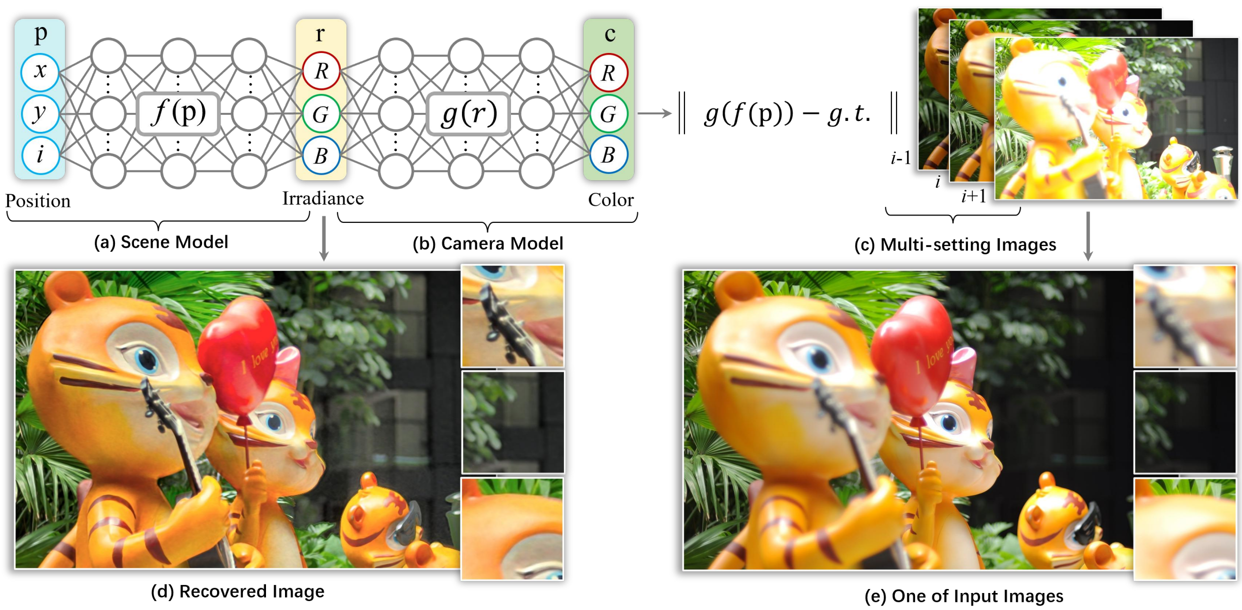

Figure 1. Our method solves inverse imaging tasks by learning an implicit neural camera model. The proposed framework consists of (a) a scene model representing scene contents and (b) a camera model simulating the camera imaging process. Given a pixel position $\mathbf{p}$ (2D pixel coordinate + 1D image index) at an image stack, the scene model maps it to corresponding irradiance value $\mathbf{r}$ (*i.e.*, $\mathbf{r} = f(\mathbf{p})$), and then the camera model maps the irradiance $\mathbf{r}$ to a pixel intensity $\mathbf{c}$ (*i.e.*, $\mathbf{c} = g(\mathbf{r})$). Two models are trained per scene and jointly optimized under the supervision of (c) a set of images captured with different camera settings (multi-focus and multi-exposure). After training, the scene irradiance have been implicitly encoded into the scene model (under an indirect supervision from the multi-setting images). We then remove the camera model and the scene model can render (d) all-in-focus HDR images by taking pixel positions as input.

Abstract

Representing visual signals with implicit coordinate-based neural networks, as an effective replacement of the traditional discrete signal representation, has gained considerable popularity in computer vision and graphics. In contrast to existing implicit neural representations which focus on modelling the scene only, this paper proposes a novel implicit camera model which represents the physical imaging process of a camera as a deep neural network. We demonstrate the power of this new implicit camera model on two inverse imaging tasks: i) generating all-in-focus photos, and ii) HDR imaging. Specifically, we devise an implicit blur generator and an implicit tone mapper to model the aperture and exposure of the camera's imaging process, respectively. Our implicit camera model is jointly learned together with implicit scene models under multi-focus stack and multi-exposure bracket supervision. We have demonstrated the effectiveness of our new model on a large number of test images and videos, producing accurate and visually appealing all-in-focus and high dynamic range images. In principle, our new implicit neural camera model has the potential to benefit a wide array of other inverse imaging tasks.

*Work was done during an internship at Tencent AI Lab.
†Corresponding authors.

1. Introduction

Using deep neural networks to learn an implicit representation of visual signal of a scene has received remarkable success (*e.g.*, NeRF [27]). It has been used to represent visual signals (*e.g.*, images [10,35], videos [5,18], and volume density [27]) with many impressive results. Besides implicit scene modelling (e.g., modelling scene radiance field via an MLP), the physical imaging process of a camera is also important for the image formation process (*i.e.*, from scene radiance field to RGB values of the sensor of a camera [36]).

However, to the best of our knowledge, little in the literature has ever tapped into the issue of finding an implicit representation to model the physical imaging process of a camera. Instead, most existing neural rendering methods assume that each pixel's RGB values are precisely the captured radiance field. In reality, before the light rays hit the imaging sensors, they need to pass through both the aperture and shutter, resulting in possible image blur caused by finite-sized aperture as well as varied dynamic range dictated by exposure time of the shutter.

Moreover, the image signal processor (ISP) inside a digital camera may also alter the obtained image, e.g., luminance change, depth of field (DoF), as well as image noises. The above observation prompts us to address two questions in this paper:

- *Can we learn an implicit camera model to represent the imaging process and control camera parameters?*
- *Can we invert the imaging process from inputs with varying camera settings and recover the raw scene content?*

Recently, learning-based methods simulating the mapping from raw images to sRGB images have been presented [13, 30, 51]. They allow photo-realistic image generation controlled by the shutter or aperture, but inverse problems of raw image restoration are challenging to model. Although a few NeRF-based methods have simply simulated cameras, they still face many issues, e.g., either RawNeRF [26] only models a camera forward mapping for controllable exposures or HDR-NeRF [14] only builds a tone-mapper module with the NeRF on static scenes to inversely recover the high-dynamic-range (HDR) radiance. It is not clear whether a unified coordinate-based MLP module of different implicit camera models can be applied to various implicit neural scene representations for inverting the imaging process in a self-supervised manner, especially for dynamic scenes.

To this end, this paper proposes a novel implicit neural camera model as a general implicit neural representation. Tested on two challenging tasks of inverse imaging, namely all-in-focus and HDR imaging, we have demonstrated the effectiveness of our new implicit neural camera model, as illustrated in Fig. 1.

The key contributions of this paper are:

1. We propose an interesting component, an implicit neural camera model including a *blur generator* module (Sec. 3.2) for the point spread function and a *tone mapper* module (Sec. 3.3) for the camera response function, to model the camera imaging process.
2. We develop a self-supervised framework for image enhancement from visual signals with different focuses and exposures and introduce several regularization terms (Sec. 3.4) to encourage the modules of the implicit neural camera to learn corresponding physical imaging formulation.
3. We showcase implicit image enhancement applications on images and videos fulfilled with the proposed framework, including forwardly controllable generation (changing exposures and focuses) or backwardly inverting restoration (all-in-focus and HDR imaging).

In the experiments, our method outperforms baseline methods in all-in-focus imaging and HDR imaging. Compared with traditional methods, our model can recover all-in-focus HDR images from fewer input images.

2. Related work

Implicit Neural Representation. Coordinate-based MLPs have been widely spread to represent a variety of visual signals, including images [10,35], videos [5,18] and 3D scenes [27]. Dupont *et al.* [10] demonstrate the feasibility of using implicit neural representation for image compression tasks. Kasten *et al.* [18] introduce a coordinate-MLP-based framework that decomposes and maps a video into a set of layered 2D atlases, which enables consistent video editing. However, these methods only focus on the representation of the visual signal and ignore the camera model which is also an important component of the whole implicit representation. Neural radiance fields (NeRF) representation models a radiance field with the weights of a neural network, which can render realistic novel views. [1,3,7,21,23,25,27,50]. Most of the NeRF methods assume input images are of a consistent camera setting. However, without modeling the camera, it's difficult for them to handle the input with varying camera settings (modern cameras always adjust the exposure and focus automatically). Most recently, some NeRF-based methods [14, 17, 43] focus on modifying the defocus blur or exposures of novel views. However, it's hard for them to control both exposures and defocus blur simultaneously. Particularly, Huang *et al.* [14] learn the global tone-mapping process from radiance to image intensity, which enables them to reconstruct the HDR radiance field. However, they only model tone-mapping with NeRF on static scenes. It's challenging for them to deal with the dynamic scenes and the inputs with other varying camera settings.

HDR Imaging. High Dynamic Range (HDR) imaging is a technique that recovers images with a superior dynamic range of luminosity. Debevec and Malik [9] propose the classic method for HDR imaging. They capture a set of images with different exposures and then merge those LDR images into an HDR image by calibrating the camera response function (CRF). However, they may cause ghost artifacts when the images are captured by hand-held cameras or on dynamic scenes. To overcome this, some two-stage approaches have been developed [11, 15, 16, 39, 48]. They first detect and remove the motion regions in the input images, and then merge the processed images into an HDR image. Recently, several methods that do not require optical flow are proposed for HDR imaging of dynamic scenes [29, 42, 46, 47]. They formulate the HDR imaging as an image translation problem from the input LDR images to the HDR images. However, the learning-based HDR imaging methods always need the HDR image as supervision. In contrast, our method is trained per scene in a self-supervised manner only requiring the input LDR images.

Multi-focus Image Fusion. Multi-focus Image Fusion (MFIF) has been studied for over 30 years [54], and various algorithms have been proposed. Li *et al*. [20] propose a matting-based method to fuse the focus information from input images. Liu *et al*. [22] propose the first CNN-based supervised MFIF method, which learns a decision map for the fusion of two source images. Inspired by this work, several works [37, 41, 49] have been conducted to improve the prediction of decision maps. While other methods [19, 55] directly map the source images to fused images via an encoder-decoder architecture. Moreover, GANs have also been applied to MFIF. Guo *et al*. [12] formulate the MFIF as an images-to-image translation problem and utilize the least square GAN objective to improve their method. Supervised methods require a large amount of training data with ground truth, but the all-in-focus images are hard to access, thus some unsupervised MFIF [44, 45] methods have been proposed. Recently, the first GAN-based unsupervised method, MFF-GAN, is proposed by Zhang *et al*. [52]. An adaptive decision block is introduced to evaluate the fusion weight of each pixel based on the repeated blur principle. However, the MFIF methods produce all-in-focus images by fusing the input images, which makes them struggle with the unaligned input images or video frames. Wang *et al*. [40] first propose a deep learning autofocus pipeline that can control the focus and generate all-in-focus images. However, they struggle to dynamic scenes with large camera motions and dynamic objects.

3. Method

In a nutshell, the goal of our method is to invert the imaging process (*e.g.*, all-in-focus imaging and HDR imaging) by learning an implicit camera model. The proposed framework is visualized in Fig. 2. Note that our method is trained per scene, which is similar to NeRF [27]. The input of our method is a set of coordinates (2D pixel coordinate + 1D image index) that denote the pixel locations at an image stack (multi-focus and multi-exposure images). Our model maps these coordinates to their corresponding pixel colors and minimizes the mean squared error between predicted colors and ground truth pixel colors for optimization. After training on the image stack, sharp scene irradiance is encoded in our scene model through indirect supervision from training images. This process resembles self-supervision since ground truth irradiance isn't used for training. During inference, we remove the blur generator and tone-mapper module and render all-in-focus and HDR images by feeding pixel positions into the scene model.

3.1. Neural Scene Representation

Inspired by the neural video representation method [18], we represent scene irradiance with two components: (1) a 2D atlas that records unique scene irradiance and (2) a deformation that matches each 3D pixel coordinate to its corresponding 2D point in the atlas. This deformation resembles the deformed field widely used in neural rendering for dynamic scenes [33]. Specifically, we use an MLP-based deformation network $\mathcal{D}$ to map each 3D pixel position to a 2D coordinate in the atlas. Given a 3D pixel position $\mathbf{p} = (x, y, i)^\top$, where $(x, y)^\top$ is the pixel coordinate and i is the image index, the deformation is given by:

$$\mathbf{q} = \mathcal{D}(\mathbf{p}), \tag{1}$$

where $\mathbf{q} = (u, v)^\top$ denotes the 2D coordinate in the atlas.

Similarly, we use an atlas network $\mathcal{A}$ to represent the 2D atlas. Given the 2D coordinate $\mathbf{q}$, the atlas network $\mathcal{A}$ maps it to the irradiance value at position $\mathbf{q}$. The process is formulated as:

$$\mathbf{r} = \mathcal{A}(\gamma(\mathbf{q})), \tag{2}$$

where $\mathbf{r}$ denotes the irradiance and γ denotes the positional encoding [27].

3.2. Blur Generator

Most classic image deblurring methods model the blur with a 2D convolution between image intensity and a Point Spread Function (PSF) [4] that indicates the degree of blurring in an image. However, according to the physical imaging pipeline, the blur convolution should be applied to the irradiance rather than the image intensity [6]. We therefore extend the model to the linear irradiance domain. In general, the spatially invariant blur is mathematically formulated as:

$$\mathbf{r}' = \mathbf{R} * \mathbf{W}, \tag{3}$$

where $\mathbf{r}'$ represents the blurry irradiance, $\mathbf{R} \in \mathbb{R}^{n\times n}$ is the sharp irradiance patch, and $\mathbf{W} \in \mathbb{R}^{n\times n}$ is the PSF of the

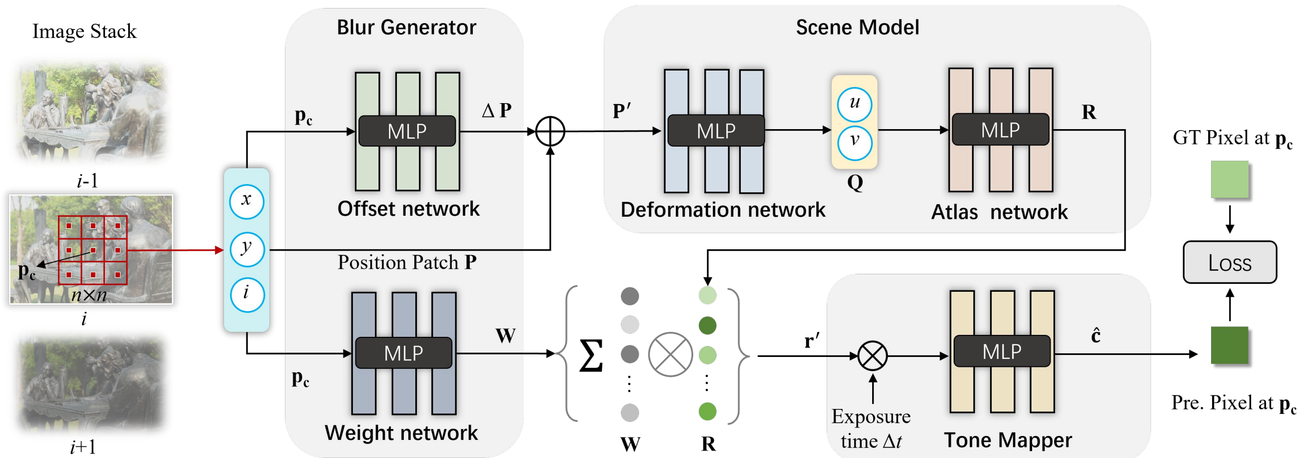

Figure 2. Illustration of our pipeline. $n \times n$ pixel positions $\mathbf{P}$ centered at $\mathbf{p}_c$ in the image stack or video sequence are fed into our model. Taking the $\mathbf{p}_c$ as input, our offset network and weight network outputs an offset patch $\Delta\mathbf{P}$ and a weight patch $\mathbf{W}$ respectively. The new position patch $\mathbf{P}' = \mathbf{P} + \Delta\mathbf{P}$ is then fed into the deformation network which predicts the corresponding 2D coordinate patch $\mathbf{Q}$ in the irradiance atlas. The atlas network maps $\mathbf{Q}$ into irradiance patch $\mathbf{R}$. We compute the blurry irradiance $\mathbf{r}'$ at position $\mathbf{p}_c$ by taking the sum of element-wise multiplication of irradiance patch $\mathbf{R}$ and weight patch $\mathbf{W}$. The blurry irradiance $\mathbf{r}'$ and exposure time Δt are then mapped into pixel intensity $\hat{\mathbf{c}}$ by a *tone mapper* which contains three small MLPs for R, G and B channels. During the inference, we remove the blur generator and tone-mapper and render all-in-focus and HDR images via the deformation network and atlas network.

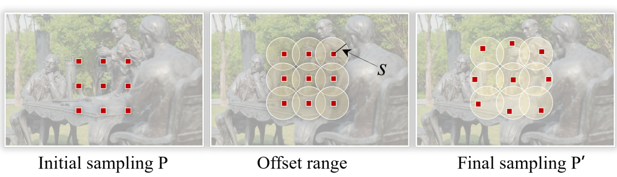

Figure 3. Sampling strategy of our method. For the sampling patch $\mathbf{P}$, 2D offsets for each position are learned to determine the final sampling patch $\mathbf{P}'$, which can flexibly modify the receptive field of the sampling. To limit the offsets, we set the maximum offset range (a circle with radius s), where s is a hyperparameter.

patch which is centered at $\mathbf{p}_c$. The operator $*$ denotes the 2D convolution.

To render the blurry irradiance of point $\mathbf{p}_c$, we once feed $n \times n$ (typically 3×3) positions into the scene model to predict the irradiance patch. However, a larger degree of blur always corresponds to a larger receptive field of the PSF and the 3×3 patch is too small to generate a large blur. One simple way to improve the receptive field of our sampling patches is using larger patches such as 5×5 and 7×7. Unfortunately, it requires a large amount of computation and memory consumption. To solve this problem, we propose a new sampling strategy in which the receptive field of each sampled patch is flexible and optimized by a network. As shown in Fig. 3, given the center position $\mathbf{p}_c = (x, y, i)$ of the initial sampling patch $\mathbf{P}$, we use a lightweight offset network $\mathcal{O}$ to predict an offset patch $\Delta\mathbf{P} = (\Delta x, \Delta y, i)$. Therefore, the final sampling positions are set to $\mathbf{P} + \Delta\mathbf{P}$.

$$\begin{aligned} \Delta\mathbf{P} &= \mathcal{O}(\mathbf{p}_c), \\ \mathbf{P}' &= \mathbf{P} + \Delta\mathbf{P}. \end{aligned} \tag{4}$$

With the learned offsets, the network can automatically modify the receptive field of the sampling to match the large blur and only a little additional computational cost is introduced. Next, We feed the coordinates of sampling patch $\mathbf{P}'$ into the scene model, thus we can obtain the irradiance patch $\mathbf{R}$.

The PSF depends on a set of factors, such as aperture size, focal length, object depth, etc. It's complicated to consider all these factors, especially the depth which is difficult to obtain. To simplify the model, each weight patch is optimized independently according to its 2D center coordinate (x, y) and the image index i, where the index i is used to embed the unique blur pattern of each training image. Thus, for an irradiance patch centered at position $\mathbf{p}_c = (x, y, i)$, we feed the $\mathbf{p}_c$ into a weight network $\mathcal{W}$ to predict the blending weight patch W. The process is expressed as:

$$\mathrm{W} = \mathcal{W}(\mathbf{p}_c). \tag{5}$$

We further formulate the blur convolution as the sum of element-wise multiplication of weight patch and irradiance patch. Eq.(3) thus is rewritten as:

$$\mathbf{r}' = \sum_{\mathbf{x} \in \mathbf{P}'} \mathbf{r}(\mathbf{x}) w(\mathbf{x}), \tag{6}$$

where $\mathbf{P}'$ is the final $n \times n$ sampled position. $\mathbf{r}(\mathbf{x})$ and $w(\mathbf{x})$ denote the irradiance and weight of position $\mathbf{x}$, respectively.

3.3. Tone Mapper

To simplify the global tone-mapping process, we take the ISO gain and aperture size as implicit factors. Consequently, the tone-mapping function f (also called CRF, Camera Response Function) is defined as:

$$\mathbf{c} = f(\mathbf{r}\Delta t), \tag{7}$$

where $\mathbf{r}$ is the irradiance captured by a camera, $\mathbf{c}$ denotes the pixel intensity, and Δt denotes the exposure time (shutter speed). We assume the exposure times are known because these can be obtained from the camera EXIF tags. Even if the exposure time is unavailable, we can jointly optimize the exposure time by taking it as a latent code. We also assume that the lighting change is insignificant and can be ignored.

Similar to HDR-NeRF [14], we transform the global tone-mapping into the logarithm irradiance domain for better optimization. Specifically, we take logarithms to both sides of Eq. (7) (base 2 is convenient, as we usually measure the exposure with exposure values (EVs)). Consequently, Eq. (7) is rewritten as:

$$\log f^{-1}(\mathbf{c}) = \log \mathbf{r} + \log \Delta t, \tag{8}$$

We further use a tone-mapping network $\mathcal{T}$ to implicitly represent $(\log f^{-1})^{-1}$. According Eq. (6) and Eq. (8), our global tone-mapping is defined as:

$$\hat{\mathbf{c}} = \mathcal{T}\left(\log \mathbf{r}' + \log \Delta t\right). \tag{9}$$

where $\mathbf{r}'$ denotes the blurry irradiance (see Eq. (6)) and $\hat{\mathbf{c}}$ denotes the predicted blurry color. Generally, each color is consisted of red, green, and blue channels, so three small MLPs are used to model the *tone mapper*.

3.4. Optimization

To adapt to various scenarios, we design several loss terms for our neural camera, including color reconstruction loss, flow loss, white balance loss and gradient loss, to encourage our implicit neural camera to learn the imaging process correctly.

Color reconstruction loss. The color reconstruction loss is the main loss in our model. The predicted blurry LDR color is supervised by the input ground truth color. We minimize their mean squared error for optimization. Formally, the loss is given by:

$$\mathcal{L}_c = \|\hat{\mathbf{c}} - \mathbf{c}\|_2^2, \tag{10}$$

where $\hat{\mathbf{c}}$ is our predicted color, and $\mathbf{c}$ is the ground truth color.

Flow loss. Ideally, for each point in the scene, its corresponding pixels that are captured under different camera settings should be mapped consistently into the same position in the atlas. We find the deformation network can fulfill this expectation in some special cases, for example when input images are aligned or the motions of the camera and objects are small. To improve our results on challenging scenes, we use an off-the-shelf optical flow estimation method [38] to predict the optical flow of each point and design an explicit constraint. Specifically, our flow loss is defined as:

$$\mathcal{L}_f = \|\mathbf{q}_\mathbf{p} - \mathbf{q}_{\mathbf{p}^*}\|_2^2, \tag{11}$$

where $\mathbf{q}_\mathbf{p}$ is the coordinate of position $\mathbf{p}$ in the irradiance atlas (Eq. (1)), $\mathbf{p}^*$ is the corresponding position of $\mathbf{p}$ in the adjacent image. The estimated optical flow is not always accurate due to the different focuses and exposures of input images. Therefore, during the training phase, the flow loss weight gradually decays to 0 over the course of optimization.

White balance loss. Theoretically, the irradiance recovered from multi-exposure images is relative to the true value with an unknown scale factor. The white balance of the irradiance is changed with the factor of each channel, thus the unconstrained scale factors estimated by our network lead to a random white balance in recovered irradiance. To tackle this problem, we introduce a loss to regularize the white balance. Formally, the white balance loss is given by:

$$\mathcal{L}_w = \|\mathcal{T}(0) - \mathbf{c}_0\|_2^2, \tag{12}$$

where $\mathbf{c}_0$ is a hyperparameter which is generally set as the midway of the color value such as 0.5 [14]. The white balance loss encourages the unit irradiance to be mapped into $\mathbf{c}_0$, which allows us to regularize the white balance of the recovered irradiance.

Gradient loss. CRF is a monotonically non-decreasing function [9]. Therefore, we add an explicit regularization to ensure the gradient of each point at the learned CRF is non-negative. We define the gradient loss as:

$$\mathcal{L}_g = \mathrm{ReLU}\left(-\frac{\mathrm{d}\,\mathcal{T}(\mathbf{r})}{\mathrm{d}\mathbf{r}}\right), \tag{13}$$

where $\mathbf{r}$ is the input irradiance of *tone mapper* $\mathcal{T}$ and ReLU denotes the ReLU (rectified linear unit) activation function.

Total loss. Finally, the total loss function is the weighted combination of the loss terms from Eqs. (10) to (13):

$$\mathcal{L}_{total} = \lambda_c \mathcal{L}_c + \lambda_f \mathcal{L}_f + \lambda_w \mathcal{L}_w + \lambda_g \mathcal{L}_g, \tag{14}$$

where $\lambda_c, \lambda_f, \lambda_w, \lambda_g$ are the weights of our loss terms.

4. Experiments

4.1. Implementation Details

We employ positional encoding in atlas network $\mathcal{A}$, with the number of frequencies 7. In *blur generator* module, the size n of the sampling path is set to 3, and the maximum offset s is set to 5 pixels. The weights of loss terms are empirically set as $\lambda_c = 1$, $\lambda_f = 100$, $\lambda_w = 1$ and $\lambda_g = 100$. We use Adam optimizer with a learning rate of 1×10^{-4} over the course of optimization. In each iteration, the batch size

Figure 4. Example results of our method compared with two-stage methods on the MFME synthetic dataset. "PM" denotes the HDR imaging method in Photomatix [31] (a) Our input images with different focuses and exposures. (b-e) All-in-focus and HDR images produced by three two-stage methods and our method. The red and green insets show the zoom-in views of the images. All HDR images are tone-mapped for display.

Table 1. Quantitative comparisons with two-stage methods on MFME synthetic dataset. Metrics are averaged over the synthetic scenes. PSNR-μ, SSIM-μ and LPIPS are computed in the global tone-mapping domain. PSNR-L, SSIM-L and HDR-VDP-2 are computed in the HDR domain. "PM" denotes the HDR imaging method in Photomatix [31].

	FusionD +PM	U2Fusio +PM	MFFGAN +PM	Ours
PSNR-μ	17.39	29.19	27.88	**31.25**
SSIM-μ	0.596	0.893	0.879	**0.895**
LPIPS	0.310	0.132	0.116	**0.104**
PSNR-L	28.24	35.51	36.43	**37.79**
SSIM-L	0.697	0.960	0.953	**0.963**
HDR-VDP-2	45.40	55.27	54.47	**58.11**

of point positions is set to $30,000$, and each model is optimized for around $150,000$ iterations. All experiments are conducted on a single V100 GPU. We train our model on an image stack, which takes about 1 hours to finish. When training on video sequences, it takes about $5 \sim 8$ hours to finish according the number of frames.

4.2. Datasets and Metrics

Datasets. We evaluate our method on three datasets: a multi-focus and multi-exposure (MFME) dataset, a multi-focus (MF) dataset, and a multi-exposure (ME) dataset. The MFME dataset consists of 4 real-world scenes and 4 synthetic scenes. Each scene contains 9 images of 3 different focuses and 3 different exposures. The real-world images are captured by a digital camera with a tripod, and the synthetic images are rendered in Blender [2]. We also render all-in-focus HDR images for the synthetic scenes, which allows for evaluating our method quantitatively. The MF dataset contains 8 real-world scenes. Two images focusing on the foreground and background respectively are captured for each scene. The resolution of the above images is 600×900 pixels. The ME dataset contains 5 real-world dynamic scenes from the HDR imaging dataset [34]. Three images with different exposures are captured for each scene. Note that the multi-focus images in the MFME dataset and MF dataset are unaligned since these images are captured by modifying the distance between the camera's lens and the image sensor.

Metrics. PSNR (higher is better), SSIM (higher is better) and LPIPS (lower is better) [53] are utilized as measurements for evaluation. Specifically, all HDR images are tone-mapped by μ-law with $\mu = 5000$, which is a simple classic global tone-mapping operator wildly used for HDR image evaluation [16,32,46]. We compute PSNR-μ, SSIM-μ and LPIPS between predicted all-in-focus HDR images and ground truth images in the global tone-mapping domain. We also compute PSNR-L and SSIM-L for comparison in the original HDR domain. Moreover, another visual metric named HDR-VDP-2 (higher is better) [24] is also computed, which is specifically designed for the evaluation of HDR images.

4.3. Evaluation

Baselines. To validate our method, we compare it with several methods on all-in-focus image restoration and HDR imaging tasks: (1) For all-in-focus image restoration, three

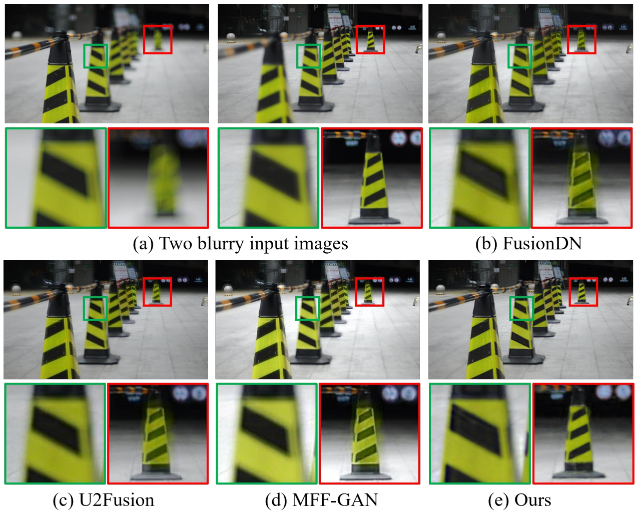

Figure 5. Example results of our method compared with MFIF methods on the MF dataset. (a) Two input images. One is near-focused and the other is far-focused. (b-e) All-in-focus results by MFIF methods and our method. The red and green insets show the zoom-in views of the images, **better viewed on screen with zoom in.**

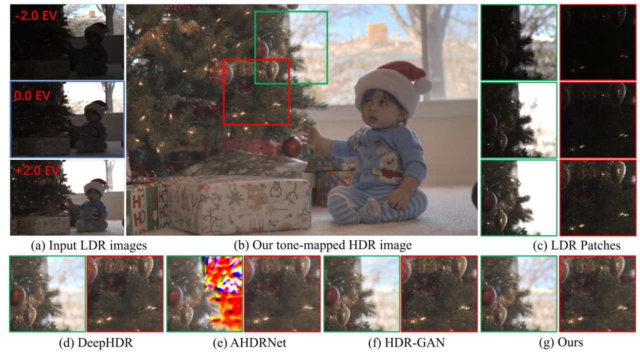

Figure 6. Example results of our method compared with HDR imaging methods on the ME dataset. (a) Three input images with different exposures. Exposure values (EVs) are shown in the upper left. The image highlighted with a blue box denotes the reference image. (b) Our recovered HDR image for the reference image. (c) zoom-in insets cropped from the input LDR images. (d-g) zoom-in insets cropped from the HDR images predicted by HDR imaging methods and our method. All HDR images are tone-mapped for display.

SOTA methods for MFIF, MFF-GAN [52], U2Fusion [44] and FusionDN [45], are selected as comparison methods. (2) For HDR imaging, we compare our method with three currently SOTA ghost-free HDR imaging methods, including HDR-GAN [29], AHDRNet [46] and DeepHDR [42]. (3) For all-in-focus and HDR image generation, we design two-stage comparison methods by combining MFIF methods with HDR imaging methods. The all-in-focus images are firstly recovered images by the aforementioned MFIF methods from the multi-focus images with consistent exposure. Taking all-in-focus images with different exposures as input, the final HDR images are reconstructed using the HDR imaging method in Photomatix [31] since we find the results by Photomatix is better than the standard static HDR imaging method by Debevec and Malik [8].

All-in-focus and HDR imaging. Figure 4 presents qualitative comparisons with two-stage methods for all-in-focus HDR image restoration. Note that the two-stage methods require all 9 images as input, while our method are only trained 3 images of different exposures and focuses, as shown in Fig. 4 (a). Compared to these two-stage methods, our method can recover image details from blurry LDR images (*e.g.*, the dog in Fig. 4). All comparison methods produce ghosting near object boundaries (*e.g.*, the boundary of the stool in Fig. 4), while our method produces sharp boundaries with a better visual experience. We also noticed that the results by FusionDN + PM show serious artifacts, likely due to color distortion in FusionDN's outputs from which Photomatix struggles to recover HDR images. Table 1 shows quantitative comparisons on the MFME synthetic dataset. Our method outperforms state-of-the-art techniques on all metrics. Although U2Fusion + PM and our method have comparable SSIM values, our method performs better on object boundaries as discussed above. Results on LPIPS and HDR-VDP-2 metrics also validate that our method recovers all-in-focus HDR images with higher visual quality.

Only all-in-focus. Comparisons with the MFIF methods for all-in-focus image restoration are visualized in Fig. 5. One can see that three MFIF methods produce ghost artifacts near the boundaries of objects. Generally, the focused and defocused boundary (FDB) is an important area where many algorithms do not perform well [54]. In the patches near the FDB, both the focused area and the defocused area exist, which makes it difficult for these methods to produce plausible weights for image fusion. Compared with them, our method achieves sharp results without color distortion because our implicit camera learns the PSF rather than fusing the input images directly, which also indicates that our method has a better performance for recovering all-in-focus images from multi-focus images.

Only HDR imaging. Comparisons with the HDR imaging methods for HDR image restoration are visualized in Fig. 6. In this challenging scene, the sitting baby has small motions. DeepHDR struggles with complex textures and produces blurry results, as shown in the red insets. AHDRNet yields serious artifacts in over-exposed areas, and that is perhaps because AHDRNet is trained on the scenes without severe exposure deviation. HDR-GAN achieves acceptable results, but struggles to reconstruct over-exposed textures such as the twigs shown in the green insets. DeepHDR also has similar limitations. Compared with the above methods, our method recovers HDR results with clear textures and details in both under-exposed and over-exposed areas.

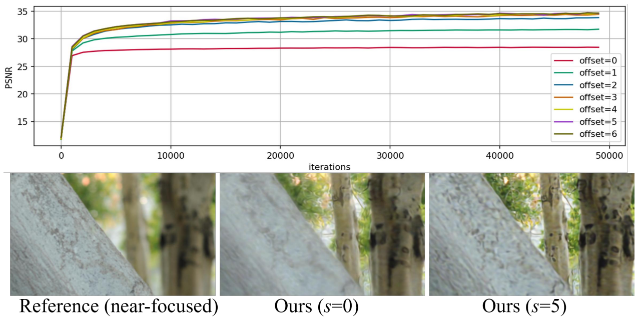

Figure 7. Results comparison of our method with different maximum offsets s on an MF scene. Top row shows the PSNR during the training phase. Bottom row shows one of the input images and our all-in-focus results.

Moreover, our method also achieves impressive results on moving objects, which can be referred from the face of the sitting baby, and that validates our method can deal with scenes with small motions.

4.4. Ablation Study

Maximum offset. In our *blur generator*, we introduce a maximum offset s to control the receptive field of our sampling (see Sec. 3.2). To assess the impact of the maximum offset s, we train and test the framework with different s values. The PSNR of the training phase is visualized in Fig. 7. It's noticed that the PSNR improves with the value of maximum offset and the increase is gradually reduced, so we generally set $s = 5$ in our experiments. Moreover, we can see that the recovered results with a larger offset (s=5) are sharper than the one with an offset set to 0, which demonstrates the effectiveness of our sampling strategy.

Losses. The ablations of gradient loss and white balance loss are shown in Fig. 8. The model without gradient loss produces distorted colors due to an incorrect CRF in the green channel. The model without white balance loss produces results with random white balance, which is unacceptable. In contrast, the CRFs of the model with full loss terms are smooth and similar across all RGB channels. This matches the ground truth that each channel is tone-mapped with the same CRF in this synthetic scene. These ablations demonstrate that our loss terms encourage the camera model to correctly fit the global tone-mapping process.

4.5. Limitations

Our method has a few limitations. Our method is trained per scene, which takes lots of time for optimization. However, the optimization time can't restrict our method to practical application, we can implement our model with the strategy of Instant NGP [28] that takes only 2.5 minutes to approximate an RGB image of resolution 20000×23466.

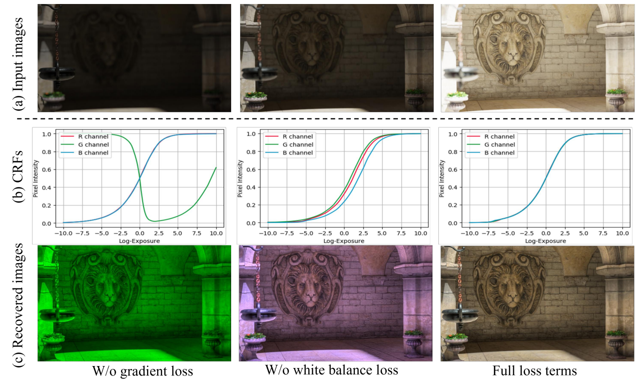

Figure 8. Results of our method with different loss terms on a synthetic MFME scene. (a) Three input blurry LDR images. (b) The learned CRFs by our *tone mapper* module with different loss terms. (c) The all-in-focus HDR results by our method with different loss terms. **Better viewed on screen with zoom in.**

In addition, the noise module is also an important component of the physical imaging formulation, especially when scenes are captured in low light. The noise model isn't simulated in our model, so the noise is recovered as part of an image. Finally, the results of our camera model are affected by the performance of the scene model. For example, the layered neural atlases model fails on dynamic scenes with complex geometry, self-occlusions, or extreme deformations with a single atlas layer (as shown in the supplementary).

5. Conclusion

In this paper, we propose an interesting component for implicit neural representations, an implicit camera model, to simulate the physical imaging process. In particular, our camera model contains an implicit *blur generator* module and an implicit *tone mapper* module, to estimate the point spread function and camera response function respectively. It is jointly optimized with scene models to invert the imaging process under the supervision of visual signals with different focuses and exposures. To disentangle the camera imaging functions and combine various captured scenarios better, a set of regularization terms are introduced to leverage the geometry and camera knowledge to achieve image tasks, including HDR imaging and all-in-focus. Experiments on various tasks confirm the superiority of the proposed self-supervised implicit camera model. With simple modifications, the framework of our camera model can be adapted to solve other inverse imaging tasks. Our code and models will be released publicly to facilitate reproducible research.

Acknowledgements. The work was supported by NSFC under Grant 62031023.

References

[1] Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *ICCV*, pages 5855–5864, October 2021. 2

[2] Blender. Blender project. https://www.blender.org/features/, 2022. 6

[3] Mark Boss, Raphael Braun, Varun Jampani, Jonathan T Barron, Ce Liu, and Hendrik Lensch. NeRD: Neural reflectance decomposition from image collections. In *ICCV*, pages 12684–12694, 2021. 2

[4] Patrizio Campisi and Karen Egiazarian. *Blind image deconvolution: theory and applications*. CRC press, 2017. 3

[5] Hao Chen, Bo He, Hanyu Wang, Yixuan Ren, Ser Nam Lim, and Abhinav Shrivastava. Nerv: Neural representations for videos. *NeurIPS*, 34, 2021. 2

[6] Xiaogang Chen, Feng Li, Jie Yang, and Jingyi Yu. A theoretical analysis of camera response functions in image deblurring. In *ECCV*, pages 333–346. Springer, 2012. 3

[7] Xingyu Chen, Qi Zhang, Xiaoyu Li, Yue Chen, Feng Ying, Xuan Wang, and Jue Wang. Hallucinated neural radiance fields in the wild. In *CVPR*, 2022. 2

[8] Paul E. Debevec and Jitendra Malik. Recovering high dynamic range radiance maps from photographs. In *SIGGRAPH*, page 369–378, 1997. 7

[9] Paul E Debevec and Jitendra Malik. Recovering high dynamic range radiance maps from photographs. In *ACM SIGGRAPH 2008 classes*, pages 1–10. 2008. 3, 5

[10] Emilien Dupont, Adam Goliński, Milad Alizadeh, Yee Whye Teh, and Arnaud Doucet. Coin: Compression with implicit neural representations. *arXiv preprint arXiv:2103.03123*, 2021. 2

[11] Thorsten Grosch et al. Fast and robust high dynamic range image generation with camera and object movement. *Vision, Modeling and Visualization, RWTH Aachen*, 277284, 2006. 3

[12] Xiaopeng Guo, Rencan Nie, Jinde Cao, Dongming Zhou, Liye Mei, and Kangjian He. Fusegan: Learning to fuse multi-focus image via conditional generative adversarial network. *IEEE TMM*, 21(8):1982–1996, 2019. 3

[13] Yuanming Hu, Hao He, Chenxi Xu, Baoyuan Wang, and Stephen Lin. Exposure: A white-box photo post-processing framework. *ACM TOG*, 37(2):1–17, 2018. 2

[14] Xin Huang, Qi Zhang, Feng Ying, Hongdong Li, Xuan Wang, and Qing Wang. Hdr-nerf: High dynamic range neural radiance fields. In *CVPR*, 2022. 2, 5

[15] Katrien Jacobs, Celine Loscos, and Greg Ward. Automatic high-dynamic range image generation for dynamic scenes. *IEEE Computer Graphics and Applications*, 28(2):84–93, 2008. 3

[16] Nima Khademi Kalantari, Ravi Ramamoorthi, et al. Deep high dynamic range imaging of dynamic scenes. *ACM TOG*, 36(4):144–1, 2017. 3, 6

[17] Takuhiro Kaneko. Ar-nerf: Unsupervised learning of depth and defocus effects from natural images with aperture rendering neural radiance fields. In *CVPR*, pages 18387–18397, 2022. 2

[18] Yoni Kasten, Dolev Ofri, Oliver Wang, and Tali Dekel. Layered neural atlases for consistent video editing. *ACM TOG*, 40(6):1–12, 2021. 2, 3

[19] Huaguang Li, Rencan Nie, Jinde Cao, Xiaopeng Guo, Dongming Zhou, and Kangjian He. Multi-focus image fusion using u-shaped networks with a hybrid objective. *IEEE Sensors Journal*, 19(21):9755–9765, 2019. 3

[20] Shutao Li, Xudong Kang, Jianwen Hu, and Bin Yang. Image matting for fusion of multi-focus images in dynamic scenes. *Information Fusion*, 14(2):147–162, 2013. 3

[21] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *CVPR*, pages 6498–6508, 2021. 2

[22] Yu Liu, Xun Chen, Hu Peng, and Zengfu Wang. Multi-focus image fusion with a deep convolutional neural network. *Information Fusion*, 36:191–207, 2017. 3

[23] Li Ma, Xiaoyu Li, Jing Liao, Qi Zhang, Xuan Wang, Jue Wang, and Pedro V Sander. Deblur-nerf: Neural radiance fields from blurry images. In *CVPR*, 2022. 2

[24] Rafał Mantiuk, Kil Joong Kim, Allan G Rempel, and Wolfgang Heidrich. Hdr-vdp-2: A calibrated visual metric for visibility and quality predictions in all luminance conditions. *ACM TOG*, 30(4):1–14, 2011. 6

[25] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. NeRF in the wild: Neural radiance fields for unconstrained photo collections. In *CVPR*, pages 7210–7219, 2021. 2

[26] Ben Mildenhall, Peter Hedman, Ricardo Martin-Brualla, Pratul Srinivasan, and Jonathan T Barron. Nerf in the dark: High dynamic range view synthesis from noisy raw images. In *CVPR*, 2022. 2

[27] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, pages 405–421. Springer, 2020. 2, 3

[28] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *arXiv preprint arXiv:2201.05989*, 2022. 8

[29] Yuzhen Niu, Jianbin Wu, Wenxi Liu, Wenzhong Guo, and Rynson WH Lau. Hdr-gan: Hdr image reconstruction from multi-exposed ldr images with large motions. *IEEE TIP*, 30:3885–3896, 2021. 3, 7

[30] Hao Ouyang, Zifan Shi, Chenyang Lei, Ka Lung Law, and Qifeng Chen. Neural camera simulators. In *CVPR*, pages 7700–7709, 2021. 2

[31] Photomatrix. Photo editing software for hdr & real estate photography. https://www.hdrsoft.com/, 2021. 6, 7

[32] K Ram Prabhakar, Susmit Agrawal, Durgesh Kumar Singh, Balraj Ashwath, and R Venkatesh Babu. Towards practical and efficient high-resolution HDR deghosting with CNN. In *ECCV*, pages 497–513. Springer, 2020. 6

[33] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *CVPR*, pages 10318–10327, 2021. 3

[34] Pradeep Sen, Nima Khademi Kalantari, Maziar Yaesoubi, Soheil Darabi, Dan B Goldman, and Eli Shechtman. Robust patch-based hdr reconstruction of dynamic scenes. *ACM TOG*, 31(6):203–1, 2012. 6

[35] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *NeurIPS*, 33:7462–7473, 2020. 2

[36] J.M. Sturge, V. Walworth, and A. Shepp. *Imaging Processes and Materials*. Van Nostrand Reinhold, 1989. 2

[37] Han Tang, Bin Xiao, Weisheng Li, and Guoyin Wang. Pixel convolutional neural network for multi-focus image fusion. *Information Sciences*, 433:125–141, 2018. 3

[38] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, pages 402–419. Springer, 2020. 5

[39] Okan Tarhan Tursun, Ahmet Oğuz Akyüz, Aykut Erdem, and Erkut Erdem. The state of the art in HDR deghosting: a survey and evaluation. In *Comput. Graph. Forum*, volume 34, pages 683–707. Wiley Online Library, 2015. 3

[40] Chengyu Wang, Qian Huang, Ming Cheng, Zhan Ma, and David J Brady. Deep learning for camera autofocus. *IEEE Transactions on Computational Imaging*, 7:258–271, 2021. 3

[41] Chang Wang, Zongya Zhao, Qiongqiong Ren, Yongtao Xu, and Yi Yu. A novel multi-focus image fusion by combining simplified very deep convolutional networks and patch-based sequential reconstruction strategy. *Applied Soft Computing*, 91:106253, 2020. 3

[42] Shangzhe Wu, Jiarui Xu, Yu-Wing Tai, and Chi-Keung Tang. Deep high dynamic range imaging with large foreground motions. In *ECCV*, pages 117–132, 2018. 3, 7

[43] Zijin Wu, Xingyi Li, Juewen Peng, Hao Lu, Zhiguo Cao, and Weicai Zhong. Dof-nerf: Depth-of-field meets neural radiance fields. In *ACM MM*, pages 1718–1729, 2022. 2

[44] Han Xu, Jiayi Ma, Junjun Jiang, Xiaojie Guo, and Haibin Ling. U2fusion: A unified unsupervised image fusion network. *IEEE TPAMI*, 44(1):502–518, 2020. 3, 7

[45] Han Xu, Jiayi Ma, Zhuliang Le, Junjun Jiang, and Xiaojie Guo. Fusiondn: A unified densely connected network for image fusion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12484–12491, 2020. 3, 7

[46] Qingsen Yan, Dong Gong, Qinfeng Shi, Anton van den Hengel, Chunhua Shen, Ian Reid, and Yanning Zhang. Attention-guided network for ghost-free high dynamic range imaging. In *CVPR*, pages 1751–1760, 2019. 3, 6, 7

[47] Qingsen Yan, Lei Zhang, Yu Liu, Yu Zhu, Jinqiu Sun, Qinfeng Shi, and Yanning Zhang. Deep hdr imaging via a non-local network. *IEEE TIP*, 29:4308–4322, 2020. 3

[48] Qingsen Yan, Yu Zhu, and Yanning Zhang. Robust artifact-free high dynamic range imaging of dynamic scenes. *Multimedia Tools and Applications*, 78(9):11487–11505, 2019. 3

[49] Yong Yang, Zhipeng Nie, Shuying Huang, Pan Lin, and Jiahua Wu. Multilevel features convolutional neural network for multifocus image fusion. *IEEE Transactions on Computational Imaging*, 5(2):262–273, 2019. 3

[50] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelNeRF: Neural radiance fields from one or few images. In *CVPR*, pages 4578–4587, 2021. 2

[51] Ke Yu, Zexian Li, Yue Peng, Chen Change Loy, and Jinwei Gu. Reconfigisp: Reconfigurable camera image processing pipeline. In *ICCV*, pages 4248–4257, 2021. 2

[52] Hao Zhang, Zhuliang Le, Zhenfeng Shao, Han Xu, and Jiayi Ma. Mff-gan: An unsupervised generative adversarial network with adaptive and gradient joint constraints for multi-focus image fusion. *Information Fusion*, 66:40–53, 2021. 3, 7

[53] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, pages 586–595, 2018. 6

[54] Xingchen Zhang. Deep learning-based multi-focus image fusion: A survey and a comparative study. *IEEE TPAMI*, 2021. 3, 7

[55] Yu Zhang, Yu Liu, Peng Sun, Han Yan, Xiaolin Zhao, and Li Zhang. Ifcnn: A general image fusion framework based on convolutional neural network. *Information Fusion*, 54:99–118, 2020. 3