NAN: Noise-Aware NeRFs for Burst-Denoising

Naama Pearl, Tali Treibitz
Dept. of Marine Technologies, School of Marine Sciences
University of Haifa, Israel
{npearl@campus,ttreibitz@univ}.haifa.ac.il

Simon Korman
Dept. of Computer Science
University of Haifa, Israel
skorman@cs.haifa.ac.il

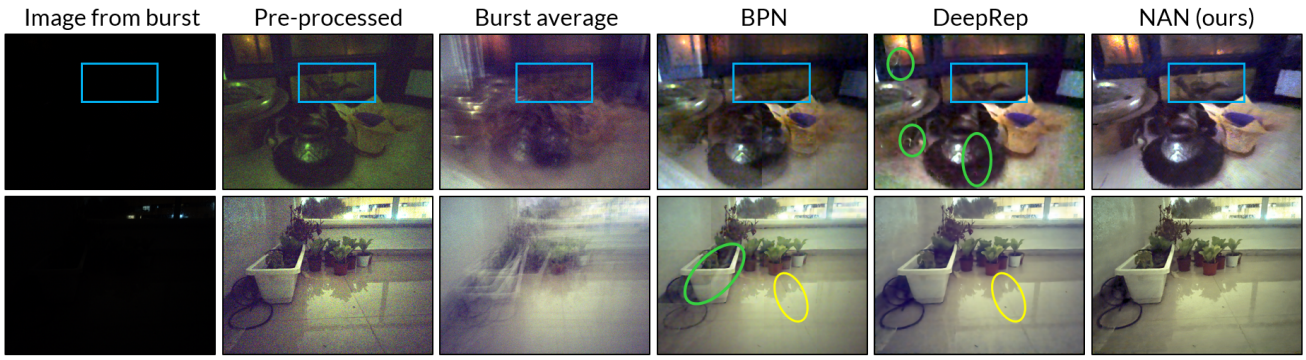


Figure 1. **Burst denoising in challenging real-world low-light scenes.** The dark burst images, scaled before processing, contain high levels of noise and significant camera motion that can be seen in their averages. The results of both BPN [31] and DeepRep [1] are generally more blurry and lack detail compared to those of NAN, the proposed method. Blue rectangles mark a clear example for comparison. Green and yellow ellipses show artifacts and missing detail correspondingly in competitor results. **The reader is encouraged to zoom-in.**

Abstract

Burst denoising is now more relevant than ever, as computational photography helps overcome sensitivity issues inherent in mobile phones and small cameras. A major challenge in burst-denoising is in coping with pixel misalignment, which was so far handled with rather simplistic assumptions of simple motion, or the ability to align in pre-processing. Such assumptions are not realistic in the presence of large motion and high levels of noise.

We show that Neural Radiance Fields (NeRFs), originally suggested for physics-based novel-view rendering, can serve as a powerful framework for burst denoising. NeRFs have an inherent capability of handling noise as they integrate information from multiple images, but they are limited in doing so, mainly since they build on pixel-wise operations which are suitable to ideal imaging conditions.

Our approach, termed NAN[1]*, leverages inter-view and spatial information in NeRFs to better deal with noise. It achieves state-of-the-art results in burst denoising and is especially successful in coping with large movement and occlusions, under very high levels of noise. With the rapid advances in accelerating NeRFs, it could provide a powerful platform for denoising in challenging environments.*

[1]Refer to the project website: noise-aware-nerf.github.io

1. Introduction

Burst denoising has become the de-facto method for low-light imaging, especially in handheld mobile devices that perform onboard processing [5, 9]. It is built on capturing multiple short-exposure (dark and noisy) frames, which are then integrated into a single coherent image. The main challenge of compensating for motion and occlusion, especially under high noise, is evident in limitations of current applications (e.g., mobile phone apps ask to "hold still" while capturing in night mode). Dealing with large parallax could significantly enhance denoising, by allowing the use of longer bursts than (the typical 8) currently common in cameras, and might enable more flexible imaging during movement (e.g., from a vehicle).

Recently, methods based on Neural Radiance Fields (NeRFs) have been demonstrated to be powerful in rendering novel views of scenes, enabling the synthesis of intricate details due to reflections and occlusions, yet all these methods consider clean high-resolution images as inputs. Since NeRFs integrate information from multiple images, they have strong potential to be leveraged in multi-frame image restoration tasks - applications they were not designed for. We show in this paper that they can be very powerful for burst denoising.

NeRFs serve as implicit scene priors and thus can inher-

ently handle noise [2, 14]. However, when using a small number of input images this capability is limited, since current architectures operate separately on each pixel with local computations that are highly prone to noise - implying much room for improvement. We demonstrate that by adding inter-view and spatial awareness to the network we significantly improve its "noise-awareness" and produce SOTA results in burst denoising, especially under large motion and high noise (see Fig. 1). To avoid specific per-scene training, we build on the recently proposed IBRNet [29] that pre-trains on different image sets and is able to produce novel views during inference time in new unseen scenes, using as few as 8 images.

To summarize our contributions: (i) We achieve SOTA results in burst denoising; (ii) We successfully exploit the natural power of radiance fields as scene priors by augmenting them with novel noise-awareness components, both in the spatial and cross-view domains; (iii) We demonstrate the advantages of our approach (NAN) over SOTA burst-denoising methods, which operate in the image plane, while NAN, being NeRF based, explicitly works in 3D space - imperative for dealing with large motion and high noise.

2. Related work

2.1. Neural Radiance Fields

The field of Neural Rendering has seen a surge of interest since the recent appearance of the NeRF [16] representation for image synthesis that has promoted rapid improvements in many aspects of the problem (see [25] for a comprehensive review on the SOTA prior to the introduction of NeRFs and [26] for the current SOTA).

The seminal paper [16] presents a volume rendering approach, in which a neural network in the form of a multilayer perceptron (MLP) encodes an implicit volumetric representation of a 3D scene. It is trained in a supervised manner from a set of posed images, to serve as a 5D radiance field that provides volume density and view-dependent radiance as a function of 3D location and 2D viewing direction.

One active line of followups [3, 6, 17, 20, 32] focuses on improving performance, with accelerations of several orders of magnitude in rendering speed, allowing for application in real-time scenarios. Another line of work, including Bundle-Adjusting [10] and Self-Calibrating [7] NeRFs, performs joint learning of camera pose registration and 3D neural representation, allowing NeRFs to be applicable to a much wider range of setups.

One of the other main challenges has been in relaxing the assumption made by [16] regarding the input scene being photometrically static – that the density and radiance of the world are constant. A large collection of works deals with non-static *density*, i.e., the presence of motion in the scene. DNeRF [19], Nerfies [18] and NeRFlow [2] do so by giving a full 4D representation that fully accommodates the motion in the scene and can create novel video clips from a customizable camera capturing motion paths.

The 'NeRF in the Wild' (NeRF-W) work [11] treats motion differently, by explaining it out, in order to reconstruct the static scene that is shared by the input images. It also deals with non-static *radiance* that is due to different per-camera exposure, color correction and tone-mapping, to avoid inaccurate reconstructions. This is done by modeling image-dependent radiance using a per-image latent embedding vector, which is suitable for modeling *global* photometric phenomena, but not independent pixel-level inconsistencies that stem from noise. The approach we present deals directly with such local inconsistencies, by exploiting cross-image and spatial within-image information.

We build on the architecture of IBRNet [29] - a differentiable image-based rendering network, which similarly to pixelNeRF [33] can generalize to new scenes, represented by possibly only a few images, without the need for test-time optimization like other prior works that are trained to model a *specific* scene. To the best of our knowledge, NAN is the first NeRF-based work that explicitly deals with significant photometric noise. Concurrently with our work, 'Nerf in the Dark' [14] demonstrate the power of NeRFs in generating novel clean views using many dozens of extremely low light linear images. In the NAN framework, we make use of the noisy target image and demonstrate efficient denoising using as little as 8 frames.

2.2. Burst Denoising

Recent works [8, 13, 22, 31] have demonstrated that burst denoising can overcome many inherent problems of long exposure photography, including motion blur and non-uniform dynamic ranges across an image that result in dark-and-noisy or overexposed regions.

Deep Burst Denoising [4] uses a recurrent neural network to denoise a burst after stabilizing it using a Lucas-Kanade tracker to find correspondences between successive frames, followed by a rotation-only motion model to estimate a homography between the frames. In Kernel Prediction Networks [13] a CNN is trained to predict per-pixel per-input-image specific 2D denoising kernels, which are used as 3D blending weights to obtain a clean target image from a noisy burst. However, they allow motion of up to 2 pixels and in practice - a vast majority of the blending weights are assigned to the target-image kernels, which means that the use of signal information from the other images is sub-optimal. Basis Prediction Networks [31] enable handling larger motions by using kernels of a much larger spatial extent, which is made possible by representing them as linear combinations of a small set of basis elements.

Very recently, SOTA burst-denoising results were presented by DeepRep [1], which proposes transforming the

MAP estimator used in image restoration tasks into a deep feature space. They tackle the motion issue by aligning each input image to the target image using optical flow [24]. This alignment approach is possibly sufficient for small motion and subtle noise, but is prone to errors in the presence of higher noise, and unlike the NeRF based approach that we adopt - it only considers a pair of frames at a time without reasoning about the scene's underlying 3D structure.

Taking into account the 3D structure enables coping with very high levels of noise. Recent burst denoising works [1, 13, 31] reported results on noise levels of a standard deviation of $\sim$ 0.2 for an image in the range $[0, 1]$, while others [2, 22] test on levels of ~ 0.1. However, higher noise levels like the ones we consider in this work of up to $\sim$ 0.4 are frequent in low-light [30] settings, especially in non-uniformly lit scenes, haze conditions, and underwater.

3. Background

3.1. Problem Setup

In the problem of *burst denoising*, given a burst of N noisy images of a scene $\{I_n\}_{n=1}^{N}$, the goal is to generate a clean version of one of the images by jointly processing the entire burst. In this work, we focus on bursts with large camera motion between frames and do not assume consecutive frames to be closely aligned. Hence we treat the burst as an un-ordered set rather than a sequence.

We follow the noise model used in [1, 13, 31] where the noisy version of a clean linear image I_n^c is given by:

$$I_n(x) \sim \mathcal{N}\Big(I_n^c(x), \sigma_r^2 + \sigma_s^2 I_n^c(x)\Big) \quad , \tag{1}$$

where x is an image coordinate, σ_r is the signal independent read-noise parameter, σ_s is the signal dependent shot-noise parameter and $\mathcal{N}$ represents the Gaussian distribution. For convenience, following [1, 13, 31], we report results on *gain* levels of an example camera (see Appendix A in [12] for a detailed explanation on the connection between sensor gain and noise parameters). These works performed model training in a relatively low-noise region (the blue rectangle in Fig. 2, upper limited by a gain of around 4), with evaluation on gain levels 1, 2, 4 and 8. As we want our method to operate in higher levels of noise, we extend the training range up to gain 20 (purple rectangle in Fig. 2 with a maximum noise level equivalent to a standard deviation of $\sim$ 0.4 for images in [0,1]) and add additional evaluation gain levels in the extended range (black points in 2). For fair comparison, we retrained [1] and [31] on the same noise region.

3.2. IBRNet

The NAN network builds upon the scheme and architecture of IBRNet [29]. Among alternative NeRF-based methods, we chose it as our baseline, first and foremost, since it pre-trains on an entirely separate set of scenes from the test-time scene on which it does not need to optimize, therefore having a relatively short inference time, requiring only a small number of images. In addition, due to its properties, common to all NeRFs, of being permutation invariant (with respect to input image ordering) and being able to handle a variable number of input images (just like [1, 8, 22], but unlike burst denoising methods like [13, 31]).

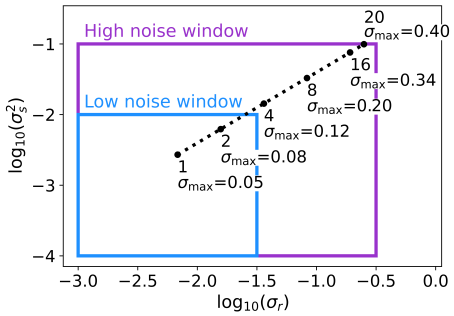


Figure 2. **Noise levels** used for training and testing. The value of $\sigma_{\max}$ indicates the maximum noise level in the image, relative to a maximum image intensity of 1. See text for details.

The general structure of IBRNet [29] is summarized in Figs. 3 and 4. Each pixel in the novel-view 'target' image is processed independently. For each such pixel, a ray r is projected into 3D space in order to estimate the existence and appearance of objects in the real-world along this ray. To achieve that, the ray is sampled (regularly on an inverse-depth scale) at a sequence of 3D positions indexed by $m = 1, \ldots, M$ (Fig. 3). For each such 3D point, the set of pixels imaging it in the other views, along with their local image features, are accumulated, while concatenating the mean and variance vectors as they are important signals for the downstream density and color prediction (Fig. 4 step 7). These representations are processed to predict the point color c_m, and its density ρ_m (Fig. 4).

The density is used to calculate the probability w_m that an object exists in that particular 3D location:

$$w_m = \left(1 - e^{-\rho_m}\right) / \, e^{\left(\sum_{j=1}^{m-1} \rho_j\right)} \, . \tag{2}$$

Eventually, these weight-color pairs, computed for each sample location along the ray, are integrated to produce the final color of the predicted target image pixel $\hat{C}(r)$:

$$\hat{C}(r) = \sum_{m=1}^{M} w_m c_m \quad . \tag{3}$$

This process is repeated in two stages, 'coarse' and 'fine', where the sampling is refined in the second stage around the depth values that most influence the rendering (i.e., at visible surfaces). In training, the novel pose is taken to be one of the available (non-input) viewpoints, allowing for a loss that measures the rendering quality by comparing the original and rendered pixel colors.

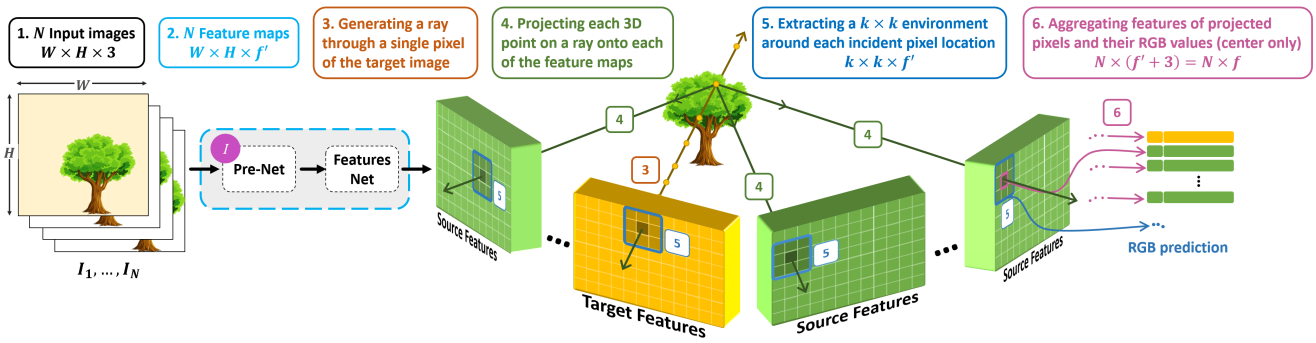


Figure 3. **Our IBRNet-based architecture - part 1: Feature extraction and ray projection**. We train a Pre-Net layer (I) to process the pixels before entering a feature extraction network. Then, for each pixel in the target image (yellow) a ray is projected. Each point sample along the ray is projected onto each of the views. The features from all viewing directions are then aggregated to enter the main network.

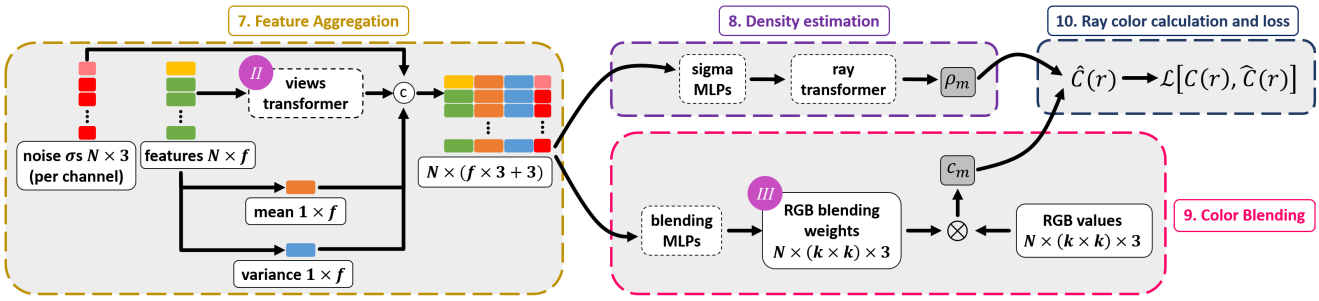


Figure 4. **Our IBRNet-based architecture - part 2: Density and color estimation**. The ray projection features are the input to the feature aggregation stage of IBRNet [29], which we extend with per-pixel noise-level inputs and a views transformer (II) to enrich the relative representation. The aggregated features are inputted into two parallel networks: Density Estimation that compares the features and determines if they stem from the same object point, in the form of an output density ρ_m; Color Blending that calculates the pixel blending weights, where we expand the weights to include a window around the pixel and separate the weights for the RGB channels (III).

4. Method

NeRFs, and in particular IBRNet, as methods for novel-view synthesis were designed with a focus on dealing with the intricate combination of significant camera motion and complex 3D scene geometry. The burst-denoising setup is different in two main aspects: (i) It requires working with degraded noisy images (*e.g.*, captured in low-light conditions), which makes things more challenging; (ii) It does not require generating novel views, but rather fixing an existing view image, which is a valuable source of data that NeRFs do not possess.

4.1. A Simple Baseline

As a baseline for our method we simply train the original IBRNet described above on a dataset of noisy images, and denote the resulting network IBRNet-*N*. For doing so we created a new dataset which we term LLFF-N, by taking (clean) scene images from the LLFF [15] and NeRF [16] datasets, which were used by IBRNet [29], and adding varying levels of noise according to the model in Eq. (1). Since this model refers to *linear* images, we first "linearize" the images by applying inverse gamma correction and inverse random white balancing as in [1, 22, 31] before we add the noise. These pseudo-linear images are the input to the tested methods, where one of the scene images is chosen at random as the target to be denoised (against which loss is calculated) and the output linear denoised image is re-processed by applying the respective gamma correction and white balancing. Although the noise in LLFF-N is simulated, its inter-frame motion is real and hence realistic in terms of occlusions and projections, in contrast with the random image-plane translations used in [13, 31].

Examining the denoising performance of IBRNet and of IBRNet-*N*, its version retrained-with-noise on the LLFF-N dataset, we observe promising initial results. Nevertheless, it is clear there is room for improvement. In the following, we detail our main proposed adaptations over the baseline model, which extend IBRNet to make it significantly more capable of handling image noise, outperforming SOTA burst denoising methods (see Sec. 5).

In particular, the quality of a radiance field depends critically on its ability to estimate both the density ρ_m and the color c_m associated with each sampled 3D position along the projected rays. Note that the density and color are learned jointly, in an end-to-end manner in which the ability to estimate one clearly affects the ability to estimate the other. We indeed observe a degradation in both as the base-

line is not truly suited to handle noise.

In Sec. 4.2, we focus on improving the density prediction of the network by a specific change in the feature extraction sub-network (purple *I* in Fig. 3, Sec. 4.2.1) and by modifying the main feature aggregation stage (purple *II* in Fig. 4, Sec. 4.2.2). In Sec. 4.3, we describe the enhancement of the color predictions of the network by using specific spatial considerations that are suitable for color blending in noisy regimes (purple *III* in Fig. 4).

4.2. Noise-Aware Density Estimation

Fig. 5 depicts the ray weights w_m (directly related to the density through Eq. (2)) for several target image points. For sharp rendering, we expect the weight function to be close to a δ function around the true depth, as is the case in clean images (Fig. 5 left). This ideal situation degrades with the addition of noise. In the IBRNet-N case (Fig. 5 middle) the distribution is very wide and less deterministic regarding the object's location. The spread of the distribution can be explained by the difficulty to accurately predict the true depth, but perhaps also as something that the learning resorts to in the optimization, since spread-out density results in photometric blurring which reduces noise (at the cost of losing details). The following suggested adaptations are shown to improve the density distribution predictions.

4.2.1 Feature Extraction

We observed that the feature extraction network used in [29] does not generalize well to handle noisy inputs, since we were able to improve performance by naïvely preprocessing the noisy images individually using even a simple Gaussian low-pass filter. We found the original entry-point convolutional layer incapable of performing simple noise filtering, due to its large 7×7 kernel size and immediate spatial resolution reduction by a factor of two.

Motivated by these observations, we suggest a simple addition to the truncated ResNet34 encoder-decoder network used in IBRNet [29], in the form of an additional (trainable) single convolution layer with 3×3 kernel size, 3 output channels and without an activation function, at the entry point of the network (see purple *I* in Fig. 3). Our new 'Pre-Net' layer, whose weights we initialize to Gaussian per-channel filters, preserves the spatial and channel dimension of the input image and reduces the input noise efficiently. It impacts the entire denoising performance, as demonstrated in our detailed ablation study (Sec. 5).

Note that the RGB values of the source images are used twice in the architecture (in addition to being the input of the feature embedding): first, they are concatenated to the corresponding feature vectors (see Fig. 3 step 6) and second, as the input to the final blending (see Fig. 4 step 9). In experimentation, we found that better performance is achieved when passing the 'Pre-Net'-filtered values in the first case

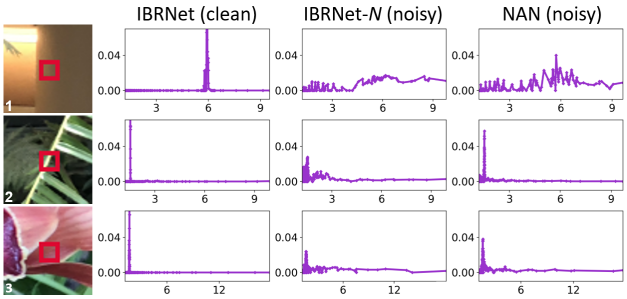


Figure 5. Ray density weight (w_m) plot for several object points. In clean images (**left**) the distributions are narrow, resembling a δ function that represents the true surface depth. In IBRNet-*N* (**center**), the δ shape is replaced by widely spread densities. NAN (**right**) better handles the noise, evident in more density accumulating around the original δ location, resulting in better details in the restored image. In smooth regions, such as in row 1, there is an advantage in a wider distribution that enables improved denoising.

and the original noisy pixels values in the second. Hence, overall, this solution is more accurate and adaptive compared to the alternative of simple input noise filtering.

4.2.2 Feature Aggregation

Recall that the pixel feature embedding vectors are the main input of the density and color estimation parts of the network (Fig. 4). On their entry, they are each expanded by concatenation to include their cross-view mean and variance feature vectors of the specific pixel (step 7). We examine this extended feature, by separately considering the original and extended statistics (mean and variance) parts. Their relative contribution to the final predictions can be ablated by zeroing their part in the vector at inference.

The comparison in the case of the original IBRNet on clean images (Fig. 6 left column) clearly shows that the feature component is practically ignored by the network (the clean target image is omitted to avoid a trivial solution). This can be explained by the sterile conditions of lack of noise. In this case, the variance of the features is indicative towards differentiating between (i) non-surface (i.e., free-space or occluded) 3D points, for which variance is typically high and (ii) visible surface points on which the rays intersect, for which variance is very low (no noise) and the mean well represents the color information (no noise).

As will be seen next, things change drastically with the presence of noise. In the middle column of Fig. 6, we perform the same comparison on the inference of IBRNet-N on noisy inputs (gain 16). Here the variance and the mean cannot capture the density and color of the 3D point (as non-agreement due to noise or due to non-surface ray intersection can be confused), hence the original features are seen to be of greater significance to the downstream predictions.

With this observation regarding the importance of the individual features in the noisy regime as well as the poor con-

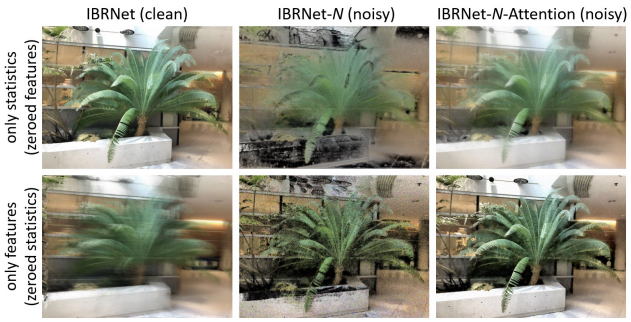


Figure 6. **Feature aggregation by the NAN attention module (Sec. 4.2.2)**: A demonstration of the importance and quality of individual features vs. joint statistics (mean, variance) in the clean vs. noisy regime, with and without our attention module. The ***columns*** compare denoising performance of IBRNet on clean images [**left**] and IBRNet-N on noisy images without **[middle]** and with [**right**] our attention module. The ***rows*** compare the influence of the features statistics (mean, variance) [**top**] to that of the individual features [**bottom**]. Please see text (in Sec 4.2.2) for a detailed discussion and interpretation of these results.

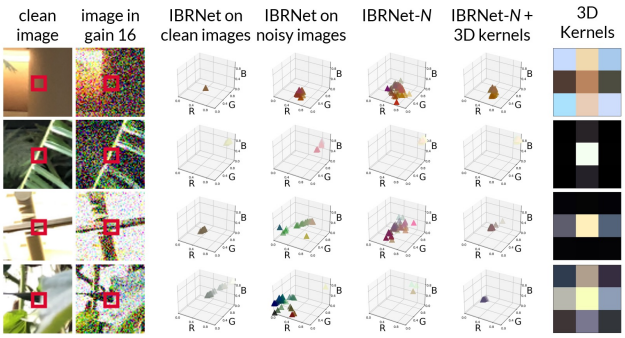


Figure 7. **Influence of the NAN 3D-blending kernels** (Sec. 4.3 and step *III* in Fig. 4). **[Left]** Target pixels (clean and noisy with gain 16). **[Center]** Scatter plots in RGB-space of final predicted colors c_m along the ray through the pixel (we show only colors that are more influential in the final integration along the ray - the ones at locations with above-average density ρ_m). These are shown for IBRNet on clean and noisy images, IBRNet-N on noisy images, and IBRNet-N with added 3D-blending kernels. **[Right]** The learned 3D-kernels associated with the 'true' pixel depth (we take the maximum-density sample), shown with RGB color-coding, scaled for clarity. See text for interpretation (Sec. 4.3).

tribution of the mean and variance as summarizing statistics, we added a standard single 5-head self-attention transformer [28] in order to capture the more intricate relations and statistics of the cross-view features (see *II* in Fig. 4). Indeed, the right column of Fig. 6, shows (especially considering the features-only bottom row) that the trained transformer provides a significant representation improvement to the vectors, resulting in a synthesized (denoised) image with lower noise and better preservation of detail. We validate the contribution of the transformer in our ablation study.

In addition, we add the estimated noise variances per-pixel per-color channel as additional features, concatenated to the global features (see Fig. 4). This helps the network to generalize over different noise levels without over-smoothing the results in lower noise levels [1, 31].

4.3. Noise-Aware Color Blending

In IBRNet [29] (and NeRF architectures in general), the output predicted color c_m is obtained as a linear combination of the N corresponding multi-view RGB-values, where the N blending weights are predicted by an MLP at the end of the network (step 9 in Fig. 4). While this strategy is standard practice in image-based rendering, it is clearly insufficient for noise filtering, since only cross-view (single pixel) information is gathered, without any spatial extent.

We suggest replacing the length-N blending vector, by a per-pixel per-view $k \times k$ spatial kernel of blending weights that is applied on the $k \times k$ spatial neighborhood of each projected pixel. Furthermore, we learn separate weights per color-channel. Thus, instead of outputting N blending scalars, we output N vectors, each of dimension $k^2 \times 3$ (Fig. 4 *III*). The separation per color channel is important especially in the case of signal-dependent noise, that can vary across the color channels. Prior to applying the blending, we normalize the weights per channel to sum to 1 by a softmax operation. In practice, we find the choice of $k = 3$ to give a good balance between quality gain and increase in network complexity, and we add a simple bilateral filtering post-process, explained in Sec. 5. This approach of using 3D kernels for blending is closely related to the per-pixel 3D kernel prediction suggested by [13]. We demonstrate that this idea works well also in the context of NeRFs.

An intuition regarding the importance of the blending kernels is visualized in Fig. 7. We look at the final predicted colors along the ray projected through several chosen example pixels (one per row). The RGB-space scatter-plots in the middle show the predicted dominant colors to be integrated over the ray. The added value of our 3D-blending kernels is clearly visible in the clean batch of colors, in comparison to the IBRNet-N reference. In addition, our learned 3D-blending kernels (visualized on the right) can be seen to (i) have true spatial extent for an improved spatial support for denoising; (ii) have actual color values, which means that the channel separation was indeed exploited by the model.

5. Results

Implementation Details. **(i) Generation of the LLFF-N** dataset is detailed in Sec. 4.1. For training, we used 35 scenes from LLFF [15] and for testing - the test set from IBRNet [29] which consists of 5 scenes from [15] and 3 scenes from [16]. **(ii) Loss:** We use l_1 loss between the original (clean) pixel values and the predicted ones over a batch of random image projection rays $\mathcal{R}$. Choosing l_1 over the l_2 loss originally used in IBRNet was motivated by the

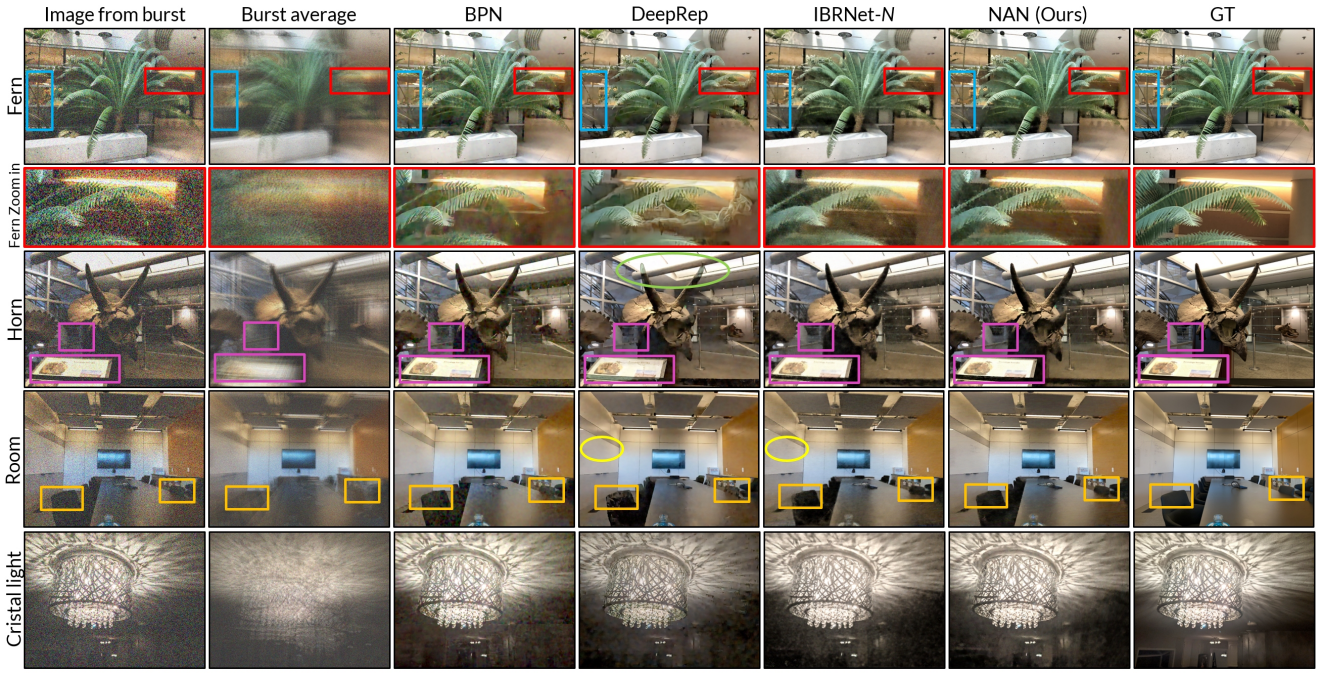


Figure 8. Comparison of results on the LLFF-N dataset (rows 1-3), gain 20, and a single scene (last row) with gain 16 from the training set of [29] (not used in our training). We show a single noisy view, the burst average, results of BPN [31], DeepRep [1], IBRNet-N, NAN, and the clean image. Interesting areas for comparison are marked by color rectangles. Color ellipses show artifacts (green) and missing detail (yellow) in competitor results. **The reader is encouraged to zoom-in.**

findings in [35] regarding l_1 being generally more suitable for image restoration tasks, and indeed it improved results by an average of 0.4 dB in our case.

(iii) Training and evaluation: We train the NAN network on LLFF-N for 255k iterations using the same training scheme as in [29], taking around 2 days on a GeForce RTX 3090 GPU. We fix the resulting model and do not fine-tune it on the novel test scenes (which could possibly improve results at the cost of runtime, as suggested in [29]). The batch size is 512 rays, which includes processing of all views and points on each ray. Since the architecture of [31] is limited to a fixed burst size of 8 images, we decrease the burst sizes that were used in [29] and fix them to 8 for all methods. Note that the loss function is applied on linear space images, but the evaluation - on the reprocessed ones. We train our network (and retrain [1, 31]) on the extended noise-region (purple rectangle in Fig. 2) and test on several gain values in this range (black points in Fig. 2).

(iv) Post-processing: We found that a simple bilateral filter [27] nicely complements our method, as a fast post-processing stage at inference time only. It decreases noise levels in homogeneous regions, while not affecting textured ones. We also found this choice to give a good efficiency trade-off, instead of enlarging our blending kernels, which would cause an increase in runtime and memory footprint, both in training and inference.

Results on LLFF-*N*. As an evaluation set we use the 8 real scenes that were used in [29], with a total of 43 bursts. We compare our results to the baseline IBRNet-*N*, and to two recent SOTA burst denoising methods, BPN [31] and DeepRep [1] on a range of gain levels. Poses were extracted from the noisy images using COLMAP [23]. Several examples results are shown in Fig. 8, with color rectangles pointing to interesting areas. Fig. 9 [top] summarizes the quantitative results in terms of PSNR, SSIM and LPIPS [34]. In PSNR, interestingly, there is a "scissor-like" behavior between BPN and DeepRep, where BPN/DeepRep is better in lower/higher noise. In contrast, NAN consistently outperform the others. In terms of SSIM and LPIPS we improve on BPN, especially in higher noise levels, and are competitive with DeepRep. While DeepRep's SSIM scores are better, we notice that its results contain many qualitative inconsistencies due to artifacts, over-smoothing and loss of detail, as can be seen in crop areas in Fig. 8.

Novel-View Results. Additionally, we test novel-view generation from noisy images by retraining the network to conduct inference without receiving the target image. Quantitative results in Fig. 9 [bottom] show consistent improvements across noise-levels and qualitative results are presented in the NAN website.

Real-World Results. Fig. 1 depicts real-world results in two challenging low-light scenes, both photographed with a Google Pixel 4 phone. Both image sets contain 8 frames that were saved in RAW format. On both image sets, NAN outperforms BPN [31] and DeepRep [1], with less artifacts

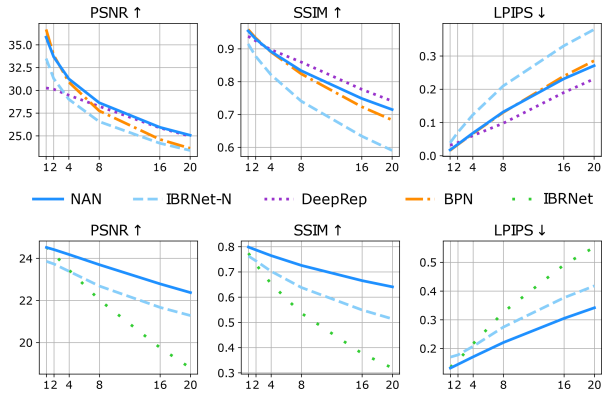


Figure 9. **Comparison to SOTA burst denoising methods on LLFF-N data**. Metrics for evaluation are PSNR, SSIM and LPIPS [34] on gain levels of $1, 2, 4, 8, 16, 20$ (x-axis). **Top: Burst-Image Denoising** (using a noisy input target image); **Bottom: Novel-View Generation** (without input target view).

and sharper results with a finer level of detail. Additional results are presented in the NAN web-page. The input to the algorithms are the linear images scaled to the range of $[0, 1]$ to match the network training, with noise parameters extracted from the EXIF file and scaled using the brightness scale of the images. Results are post-processed for display (see interesting details marked on the images). The BPN [31] results have visible square artifacts stemming from the original implementation that is limited in resolution.

Robustness to Motion. For each point of each target image in LLFF-N we calculated the average 2D displacement from each of the other frames, using EpicFlow [21]. In Fig. 10 we compared average MSE results, for gain levels 1, 8 and 20, as a function of the calculated average displacement. The distribution of disparities (bottom) shows presence of large motion that needs to be dealt with. It is noticeable that NAN outperforms the competitors in the challenging regime of high noise and large motion and its performance is the least affected by motion magnitude.

Ablation Study. We ablate the performance of NAN on the LLFF-N data with gain 16, with results in Table 1 demonstrating the efficacy of our proposed additions. We also calculated the error of the predicted depth map with respect to the depth map generated by [29] from the clean images. We see a steady improvement from each of the components, with the 3D-blending being most significant - demonstrating that incorporating spatial awareness within the network improves 3D understanding, which might be a key factor in burst-denoising. We provide further ablations, regarding loss functions and kernel sizes in the NAN web-page.

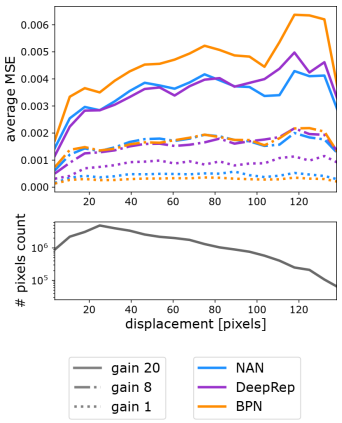
6. Discussion

We demonstrated that the NeRF solution can serve as a powerful burst denoising paradigm for challenging sets containing large motion, parallax effects and high noise. Implicit consideration of the underlying scene geometry regularizes the displacement and imposes an excellent prior. It enables exploiting spatial and inter-frame information to infer the clean scene appearance. We show denoising of challenging motion in the order of 100 pixels, including occlusions, under severe noise, higher than previously tested.

Limitations of NAN lead to our future plans: Currently we calculate the poses in pre-processing, while recent methods have incorporated pose estimation into the NeRF itself. We plan doing so, and hypothesize that it will enable coping with even higher noise levels, that currently prohibit separate pose estimation. The runtime of NAN is its main drawback - a common issue with NeRFs. Nevertheless, recent advancements have shown impressive results in speeding up and parallelizing radiance field calculations, which we are sure our framework can benefit from. We believe that with these future developments, burst denoising using NeRFs can result in a very compelling framework for challenging burst-denoising inputs.

Acknowledgements. The research was funded by Israel Science Foundation grant #680/18 and the Israeli Ministry of Science and Technology grant #3 – 15621, the Leona M. and Harry B. Helmsley Charitable Trust, the Maurice Hatter Foundation. We thank the Interuniversity Institute for Marine Sciences of Eilat for making their facilities available to us; and Deborah Steinberger-Levy, Opher Bar Nathan, and Amit Peleg for help with experiments.

Figure 10. Denoising error as a function of pixel displacement in gains 1, 8 and 20. We calculated the average displacement for each object point using optical-flow [21]. [Bottom] Histogram of pixel displacements in our test set. Pixel count is in log space. [Top] We calculated the average MSE reconstruction error per displacement. See text for interpretation.



I	*II*	*III*	*IV*	PSNR ↑	SSIM ↑	LPIPS ↓	depth ↓
-	-	-	-	24.562	0.661	0.306	11.510
+	-	-	-	23.895	0.608	0.348	10.741
-	+	-	-	24.558	0.656	0.305	10.714
-	-	+	-	24.880	0.692	0.278	9.226
-	+	+	-	25.122	0.693	0.273	8.621
+	-	+	-	25.415	0.711	0.256	9.731
+	+	-	-	25.384	0.694	0.263	11.892
+	+	+	-	25.688	0.717	0.251	9.213
+	+	+	+	25.853	0.730	0.238	9.028

Table 1. **Ablation study**. Trained on the high noise range rectangle, evaluated on gain 16. *I*- pre-net, *II* - feature aggregation, *III* - 3D-blending, *IV* - noise parameters. Depth error is average MSE.

References

[1] Goutam Bhat, Martin Danelljan, Fisher Yu, Luc Van Gool, and Radu Timofte. Deep reparametrization of multi-frame super-resolution and denoising. In *Proc. IEEE CVPR*, pages 2460–2470, 2021. 1, 2, 3, 4, 6, 7

[2] Yilun Du, Yinan Zhang, Hong-Xing Yu, Joshua B Tenenbaum, and Jiajun Wu. Neural radiance flow for 4D view synthesis and video processing. In *Proc. IEEE ICCV*, pages 14324–14334, 2021. 2, 3

[3] Stephan J Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. Fastnerf: High-fidelity neural rendering at 200fps. In *Proc. IEEE ICCV*, 2021. 2

[4] Clément Godard, Kevin Matzen, and Matt Uyttendaele. Deep burst denoising. In *ECCV*, pages 538–554, 2018. 2

[5] Samuel W Hasinoff, Dillon Sharlet, Ryan Geiss, Andrew Adams, Jonathan T Barron, Florian Kainz, Jiawen Chen, and Marc Levoy. Burst photography for high dynamic range and low-light imaging on mobile cameras. *ACM Transactions on Graphics (ToG)*, 35(6), 2016. 1

[6] Peter Hedman, Pratul P Srinivasan, Ben Mildenhall, Jonathan T Barron, and Paul Debevec. Baking neural radiance fields for real-time view synthesis. In *Proc. IEEE ICCV*, 2021. 2

[7] Yoonwoo Jeong, Seokjun Ahn, Christopher Choy, Anima Anandkumar, Minsu Cho, and Jaesik Park. Self-calibrating neural radiance fields. In *Proc. IEEE ICCV*, pages 5846–5854, 2021. 2

[8] Filippos Kokkinos and Stamatis Lefkimmiatis. Iterative residual cnns for burst photography applications. In *Proc. IEEE CVPR*, pages 5929–5938, 2019. 2, 3

[9] Orly Liba, Kiran Murthy, Yun-Ta Tsai, Tim Brooks, Tianfan Xue, Nikhil Karnad, Qiurui He, Jonathan T Barron, Dillon Sharlet, Ryan Geiss, et al. Handheld mobile photography in very low light. *ACM Transactions on Graphics (TOG)*, 38(6), 2019. 1

[10] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. In *Proc. IEEE ICCV*, 2021. 2

[11] 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 *Proc. IEEE CVPR*, pages 7210–7219, 2021. 2

[12] Ben Mildenhall, Jonathan T Barron, Jiawen Chen, Dillon Sharlet, Ren Ng, and Robert Carroll. Burst denoising with kernel prediction networks. *arXiv preprint arXiv:1712.02327*, 2017. 3

[13] Ben Mildenhall, Jonathan T Barron, Jiawen Chen, Dillon Sharlet, Ren Ng, and Robert Carroll. Burst denoising with kernel prediction networks. In *Proc. IEEE CVPR*, 2018. 2, 3, 4, 6

[14] 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. *Proc. IEEE CVPR*, 2022. 2

[15] Ben Mildenhall, Pratul P Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)*, 38(4), 2019. 4, 6

[16] 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 *Proc. ECCV*, pages 405–421. Springer, 2020. 2, 4, 6

[17] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *arXiv:2201.05989*, Jan. 2022. 2

[18] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *Proc. IEEE ICCV*, pages 5865–5874, 2021. 2

[19] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proc. IEEE ICCV*, pages 10318–10327, 2021. 2

[20] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In *Proc. IEEE ICCV*, 2021. 2

[21] Jerome Revaud, Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. Epicflow: Edge-preserving interpolation of correspondences for optical flow. In *Proc. IEEE CVPR*, pages 1164–1172, 2015. 8

[22] Xuejian Rong, Denis Demandolx, Kevin Matzen, Priyam Chatterjee, and Yingli Tian. Burst denoising via temporally shifted wavelet transforms. In *ECCV*, pages 240–256. Springer, 2020. 2, 3, 4

[23] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proc. IEEE CVPR*, 2016. 7

[24] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proc. IEEE CVPR*, pages 8934–8943, 2018. 3

[25] Ayush Tewari, Ohad Fried, Justus Thies, Vincent Sitzmann, Stephen Lombardi, Kalyan Sunkavalli, Ricardo Martin-Brualla, Tomas Simon, Jason Saragih, Matthias Nießner, et al. State of the art on neural rendering. In *Computer Graphics Forum*, volume 39, pages 701–727. Wiley Online Library, 2020. 2

[26] Ayush Tewari, O Fried, J Thies, V Sitzmann, S Lombardi, Z Xu, T Simon, M Nießner, E Tretschk, L Liu, et al. Advances in neural rendering. In *ACM SIGGRAPH 2021*, pages 1–320. ACM, 2021. 2

[27] Carlo Tomasi and Roberto Manduchi. Bilateral filtering for gray and color images. In *Proc. IEEE CVPR*, pages 839–846, 1998. 7

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 6

[29] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *Proc. IEEE CVPR*, 2021. 2, 3, 4, 5, 6, 7, 8

[30] Ruixing Wang, Xiaogang Xu, Chi-Wing Fu, Jiangbo Lu, Bei Yu, and Jiaya Jia. Seeing dynamic scene in the dark: A high-quality video dataset with mechatronic alignment. In *Proc. IEEE ICCV*, pages 9700–9709, 2021. 3

[31] Zhihao Xia, Federico Perazzi, Michaël Gharbi, Kalyan Sunkavalli, and Ayan Chakrabarti. Basis prediction networks for effective burst denoising with large kernels. In *Proc. IEEE CVPR*, pages 11844–11853, 2020. 1, 2, 3, 4, 6, 7, 8

[32] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. PlenOctrees for real-time rendering of neural radiance fields. In *Proc. IEEE ICCV*, 2021. 2

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

[34] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proc. IEEE CVPR*, pages 586–595, 2018. 7, 8

[35] Hang Zhao, Orazio Gallo, Iuri Frosio, and Jan Kautz. Loss functions for image restoration with neural networks. *IEEE Transactions on computational imaging*, 2016. 7