Dancing under the stars: video denoising in starlight

Kristina Monakhova
UC Berkeley

Stephan R. Richter
Intel Labs

Laura Waller
UC Berkeley

Vladlen Koltun
Intel Labs

Figure 1. Video denoising in submillilux. (a) One frame from the raw noisy video clip (10 fps) taken between 0.6–0.7 millilux on a clear, moonless night with no external illumination. (b) Result after contrast stretching the video clip. (c) Denoised result using our denoiser.

Abstract

Imaging in low light is extremely challenging due to low photon counts. Using sensitive CMOS cameras, it is currently possible to take videos at night under moonlight (0.05-0.3 lux illumination). In this paper, we demonstrate photorealistic video under starlight (no moon present, <0.001 lux) for the first time. To enable this, we develop a GAN-tuned physics-based noise model to more accurately represent camera noise at the lowest light levels. Using this noise model, we train a video denoiser using a combination of simulated noisy video clips and real noisy still images. We capture a 5-10 fps video dataset with significant motion at approximately 0.6-0.7 millilux with no active illumination. Comparing against alternative methods, we achieve improved video quality at the lowest light levels, demonstrating photorealistic video denoising in starlight for the first time.

1. Introduction

Some animals, such as hawkmoths and carpenter bees, can effectively navigate on the darkest moonless nights by the light of the stars (< 0.001 lux) [29, 47, 55], while our best CMOS cameras generally require at least 3/4 moon illumination (> 0.1 lux) to image moving objects at night [11]. Seeing in the darkest settings (moonless, clear nights) is extremely challenging due to the minuscule amounts of light present in the environment. In such dark settings, photographers can use long exposure times (e.g. 20 seconds or higher) to collect enough light from the scene. This approach works well for still images, but severely limits the temporal resolution and precludes imaging of moving objects. Alternatively, cameras can increase the gain, making each pixel effectively more sensitive to light. This allows shorter exposures, but greatly increases the noise present in each frame. In this setting, motion might be perceptible, but noise overwhelms the images.

Denoising algorithms can be used to improve the image quality in noisy images. Over the years, a number of denoising algorithms have been developed, from classic methods (e.g. BM3D [17], V-BM4D [39]) to deep learning-based approaches [58]. Each of these methods attempt to extract the signal from the noise based on some assumptions about the statistical distributions of the image and noise. While successful for certain denoising tasks, most of these methods are built upon a simplistic noise model (Gaussian or Poisson-Gaussian noise), which does not apply in extremely low-light settings. When high sensor gain is used in low-light images, the noise is often non-Gaussian, non-linear, sensor-specific, and difficult to model or characterize. Without having a good understanding of the structure of the noise in the images, denoising algorithms may fail – mistaking the structured noise for signal.

Recently, several deep learning-based approaches have provided remarkable denoising performance in low light down to 0.1–0.3 lux [13, 14]. Rather than assuming a certain noise model, these methods train a denoiser using clean/noisy image pairs captured by a camera. Such an

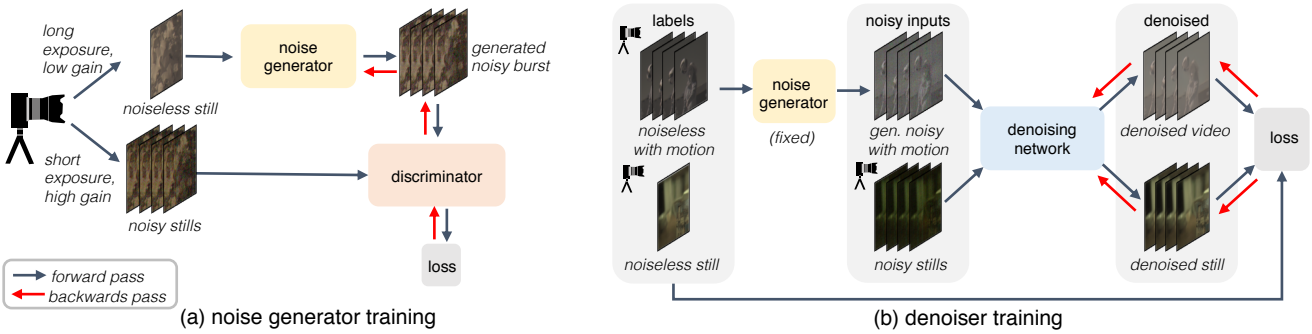

Figure 2. Method overview. (a) First we train our noise generator along with a discriminator, which aims to distinguish between real and synthetic noise. We use a limited dataset of long exposure/low gain and short exposure/high gain non-moving image pairs during this training process. After training, the noise generator can synthesize realistic noise. (b) Next, we train our denoiser using a combination of synthetic clean/noisy video clips produced using our noise generator as well as still clips from our camera. This allows us to train a video denoiser without needing experimental motion-aligned video pairs.

approach automatically accounts for the low-light noise through deep learning, however this comes at the price of needing to capture thousands of training image pairs. Furthermore, the dataset is camera-dependent and must be retaken for each different sensor, since noise can be highly camera-specific. In addition, while it is possible to capture clean/noisy image pairs for non-moving objects by changing the exposure/gain settings, capturing clean/noisy image pairs for moving scenes adds additional complexities (e.g. needing a second camera, aligning motion), making this experimentally impractical [28].

To achieve submillilux video denoising, we propose to use a combination of three things: 1) a very good camera optimized for low-light imaging and set to the highest gain setting (Sec. 4), 2) learning our camera's noise model using a physics-inspired noise generator and easy-to-obtain still noisy images from the camera (Sec. 3), and 3) using this noise model to generate synthetic clean/noisy video pairs to train a video denoiser (Sec. 5). Since our physics-based noise generator is trained using a limited dataset of still clean/noisy bursts, we do not need to acquire experimental motion-aligned clean/noisy video clips, greatly simplifying the experimental setup and decreasing the amount of data we need to collect. After noise generator training, we hold the noise generator fixed and train our video denoising using a combination of still clean/noisy image bursts paired with synthetic video clips (Sec. 5). Figure 2 summarizes this two-stage training approach for our noise generator and denoiser.

We demonstrate the effectiveness of our denoising network on 5-10 fps videos taken on a moonless clear night in 0.6 millilux, showing photorealistic video denoising in submillilux levels of illumination for the first time. We present several challenging scenes with extensive motion, in which subjects dance by only the light of the Milky Way as a meteor shower rains down from above.

2. Related work

Image and video denoising. A variety of techniques for image and video denoising have been proposed and studied throughout the years. Many of the classic denoising methods rely on specific image priors, such as sparsity [20, 42], smoothness [46], or Gaussian mixture models [18, 57]. Others utilize a non-local strategy to collaboratively denoise similar patches across an image [9,17,35,39]. More recently, deep learning-based approaches have been applied to image denoising, in which an image prior is learned from the data rather than explicitly assumed [10, 16, 19, 49, 52]. These methods have shown significant improvements over classic methods in terms of image quality, however they often make simplistic assumptions on the noise statistics, such as i.i.d Gaussian. When trained with these simplistic assumptions, classic techniques such as BM3D often outperform deep learning-based methods on real photographs with real noise [41]. In this regard, several datasets of noisy and clean image pairs from real cameras have been created to benchmark and improve the performance of deep learning-based denoisers on real cameras [3,4,41]. In addition, some work has focused on "unprocessing" online image datasets to better match RAW image distributions in order to generate more synthetic data for training RAW image denoisers [8].

Alternatively, another line of deep learning-based denoising focuses on unsupervised learning, in which no ground truth images are used to train the denoiser. Such methods either assume that the structure of a deep network can act as a prior for the image denoising [51], or assume statistical independence of the noise to train a denoiser using samples drawn from one [6,32,33,43] or multiple [36] noisy image frames. While this line of work is promising since it does not require ground truth data and therefore can be adapted for different camera sensors, these methods are not easily adaptable to the more structured or signal-dependent noise that is present in low-light settings under high gain, such as banding noise.

Low-light photography. A number of methods focus particularly on the challenging case of denoising for low-light and night photography. A popular method for low-light photography is burst denoising, in which multiple images are merged and denoised, as in HDR+ and Google Night Sight [27, 38]. These methods require robust alignment techniques to account for any motion in the scene, which is difficult in the presence of extreme noise. A number of approaches have emerged that attempt to do this burst-alignment step automatically through deep learning [23, 40]. Often, the final goal of burst denoising is to obtain a single clean image from the noisy burst. In our work, we aim to obtain a full denoised video rather than a single clean image.

Recently, a number of deep learning-based methods attempt to address low-light photography by learning to denoise images in the presence of extreme noise. These methods learn a denoiser and image enhancer network for underexposed low-light images by first collecting a training dataset of clean/noisy image pairs [13, 14, 28] and have demonstrated remarkable results for low-light conditions down to 0.1 lx. Our work pushes this limit down by two orders-of-magnitude, demonstrating video denoising below 1 mlx. Furthermore, these methods rely on a camera-specific dataset of ground truth/noisy image pairs for training, which is particularly challenging for video denoising. Our approach only requires a limited dataset of still image pairs for video denoising, eliminating the need for a large experimental dataset of noisy/clean aligned videos.

Noise models. A Gaussian noise model is commonly used for typical imaging systems, however this is not a very realistic representation of real-world sensor noise [41]. Signal-dependent models, such as Poisson-Gaussian [21, 22] or a heteroscedastic Gaussian model [26] are more realistic as they account for the effect of shot noise in cameras. However, there are many more effects, such as clipping [22], fixed pattern noise, and banding noise that these models don't account for [7, 31]. Additional work has focused on characterizing sensor noise in low-light environments by fitting to certain distributions for different noise components [54, 56]. In general, noise modeling for extremely low-light imaging is complicated and it is difficult to accurately characterize and synthesize realistic camera noise, since the noise can be highly structured and sensor-dependent [5, 31].

Recently, rather than characterizing the sensor noise, several methods have attempted to learn to synthesize realistic noise using generative adversarial networks (GANs) [15, 50] and normalizing flow models [2]. However, physics-based statistical methods tend to outperform DNN-based methods [60]. We combine physics-based statistical methods with GAN-based training techniques to learn to approximate the sensor noise in a data-driven manner, without the need for hand-calibrating a noise model.

3. Physics-inspired noise generator

Cameras aim to exactly measure and record the light intensity of a scene, converting photons to voltage readings, which are then converted to bits by an analog to digital converter (ADC). Throughout this process, noise is inadvertently added to the measurement both as a function of photon statistics and the circuitry of the sensor. In well-lit environments, sensor noise is well understood and can be modeled as a combination of two primary components: shot noise, which originates from photon arrival statistics, as well as read noise, which is caused by imperfections in the sensor readout circuitry [26]. In low-light settings, this noise approximation breaks down and does not adequately describe the complex noise statistics of the scene. Previous work has shown that noise in low-light settings can be expressed as a combination of photon shot noise, read noise, row noise, and quantization noise, which can be estimated through a rigorous calibration process [56]. Inspired by this work, we propose a physics-inspired noise generator which consists of several learned statistical noise parameters. Rather than calibrating the noise parameters by hand, we automatically learn the optimal parameters using a GAN that is fed several pairs of calibration clean (long exposure, low gain)/noisy (short exposure, high gain) image pairs. Using this framework, our noise generator is trained to synthesize realistic noise in extremely low light and high gain settings, see Fig. 3.

3.1. Physics-inspired parameters

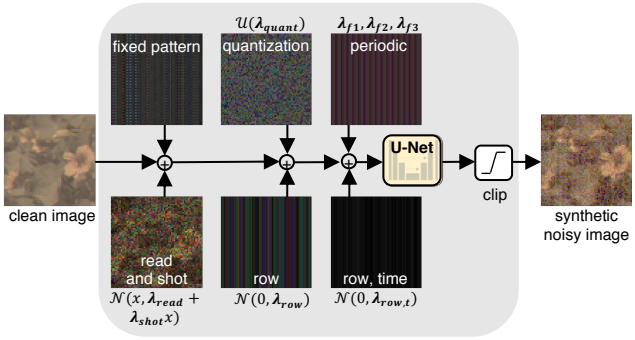

Figure 3. Physics-inspired noise generator. Our noise generator takes in a clean image and produces a synthetic noisy image. During training, our physics-inspired statistical noise parameters are optimized along with a U-Net to produce a synthetic noisy image that is indistinguishable from a real noisy image.

Our noise generator contains several physics-inspired parameters, mainly consisting of variance terms for random distributions, as well as a convolutional neural network (CNN) to capture any additional effects, known or unknown, that are difficult to specifically model. Both components are jointly optimized during training. First, we parameterize the contributions of read and shot noise. Shot noise is a function

of the light intensity hitting the sensor and is often modeled as a Poisson random variable, whereas read noise can be approximated as a zero-mean Gaussian random variable [21, 22]. Together, these are commonly approximated using a single heteroscedastic Gaussian random variable, where the mean is equal to the true signal, x, and the variance is parameterized by the read, λ_{read}, and shot noise, λ_{shot}. We note that a Poisson noise model is more accurate for low photon counts, but a Gaussian model is differentiable with respect to its mean and variance, allowing these parameters to be learned:

$$N_s + N_r \sim \mathcal{N}(\mu = x, \sigma^2 = \lambda_{read} + \lambda_{shot} x) \quad (1)$$

Low-light imaging often suffers from banding noise, which is a camera-dependent noise that results from the camera circuitry and is particularly prominent at high ISO settings. Banding noise often appears as horizontal or vertical lines in the measurement [7, 31]. We model this as a fixed offset added to each column/row, where the fixed offset is drawn from a zero-mean Gaussian random variable with variance λ_{row}, see Fig. 3. Banding noise is generally independent for each frame, however we have noticed that in extreme low light and high gain settings some banding patterns are consistent across a number of frames. To model this, we also include a time-consistent banding pattern noise which is static across each set of frames. As with the original banding noise, this time-consistent noise is modeled as a zero-mean Gaussian random variable with variance $\lambda_{row,t}$.

In addition to this, at extreme gain settings, we notice that the measurements suffer from periodic noise, potentially due to ADC imperfections/effects at these high gain settings. This periodic noise appears as spikes in the frequency domain of the raw noisy measurements, corresponding to adding a 1 or 2 pixel period sinusoidal pattern to the image with a random amplitude (Fig. 3). We parameterize this random amplitude by learned parameters: λ_{f1}, λ_{f2}, λ_{f3}. See Suppl. for further implementation details and discussion.

Next, we add a uniform noise component, to approximate quantization noise in the sensor:

$$N_q \sim \mathcal{U}(\lambda_{quant}). \quad (2)$$

Here λ_{quant} is our parameter for the quantization noise interval. Generally, this noise component is well-defined based on the number of bits used by the camera sensor. However, we find that allowing this noise parameter to vary can improve our noise generator. Finally, we include a fixed pattern noise component, N_f, that stays constant throughout all images. We measure this experimentally using an average of several image sequences. We find that letting this fixed pattern noise be learned can improve the Kullback–Leibler (KL) divergence between the real noise and generated noise, but this parameter is prone to overfitting and we achieve the best denoising performance when leaving N_f fixed and experimentally measured.

Thus, our physics-inspired noise model consists of the following components:

$$N = N_s + N_r + N_{row} + N_{row,t} + N_q + N_f + N_p, \quad (3)$$

where $N_{shot}, N_{read}, N_{row}, N_{row,t}, N_q, N_f,$ and N_p approximate the contributions of shot noise, read noise, row noise, temporal row noise, quantization noise, fixed pattern, and periodic noise.

After initial noise is added to a clean image using the physics-inspired parameters, the intermediate noisy image is passed through a CNN, which aims to improve the initial noise estimate and capture any effects that were not captured by the physics-inspired noise model. We utilize a residual 2D U-Net [45] for this. (See Suppl. for architecture details.) The final output of our noise generator is clipped to $[0, 1]$. Together, we have a total of 8 physics-inspired parameters (λ_{read}, λ_{shot}, $\lambda_{quant}, \lambda_{row}, \lambda_{row_t}, \lambda_{f1}, \lambda_{f2}, \lambda_{f3}$), as well as the parameters of the U-Net. All parameters are optimized during training to produce a realistic synthetic noisy image from a noiseless image. Figure 3 shows our physics-guided noise generator with a sample for each noise component.

3.2. GAN training

We want our noise generator to produce different noise samples at each forward pass. This is incompatible with direct supervision where each clean image would be paired to a ground truth noisy image. Thus, to train our noise generator, we resort to an adversarial setup [24], consisting in this case of our noise generator and a discriminator, which evaluates the realism of synthesized noisy images.

Our discriminator operates on noise patches of size 64x64. For our training objective, we utilize a standard Wasserstain GAN with a gradient penalty framework [25], which is optimized with the following objective function:

$$L = \mathop{\mathbb{E}}_{\tilde{x}\sim\mathbb{P}_g}[D(\tilde{x})] - \mathop{\mathbb{E}}_{x\sim\mathbb{P}_r}[D(x)] + \lambda \mathop{\mathbb{E}}_{\hat{x}\sim\mathbb{P}_{\hat{x}}}[\|(\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2], \quad (4)$$

where $\mathbb{P}_r$ is the real noisy data distribution, $\mathbb{P}_g$ is the model distribution defined by the generator, $\tilde{x} = G(z)$, z is a noiseless image patch, and D is our discriminator. See Suppl. for full training details.

4. Camera selection and data collection

To meet our objective of producing photorealistic videos at submillilux illumination levels, we need to carefully choose an appropriate camera sensor and lens. Generally, larger pixel sizes are preferable for low-light imaging, so that each pixel can collect more photons. In addition, near infrared (NIR) sensitivity is useful for nighttime imaging

because there are more detectable photons in NIR than at RGB wavelengths at night [37, 53].

We choose to use a Canon LI3030SAI Sensor, which is a 2160x1280 sensor with 19μm pixels, 16 channel analog output, and increased quantum efficiency in NIR. This camera has a Bayer pattern consisting of red, green, blue (RGB), and NIR channels (800-950nm). Each RGB channel has an additional transmittance peak overlapping with the NIR channel to increase light throughput at night. During daylight, the NIR channel can be subtracted from each RGB channel to produce a color image, however at night when NIR is dominant, subtracting out the NIR channel will remove a large portion of the signal resulting in muffled colors. We pair this sensor with a ZEISS Otus 28mm f/1.4 ZF.2 lens, which we choose due to its large aperture and wide field-of-view.

We capture 3 sets of datasets from our camera: bursts of paired clean (low gain, long exposure)/noisy (high gain, short exposure) static scenes, clean videos of moving objects, and noisy videos of moving objects in submillilux conditions. All images/videos are captured in RAW format. The paired dataset of static scenes is used to train our noise generator. Both the paired dataset and the clean videos of moving objects are used to train the denoiser. The final dataset is reserved for testing the performance of our denoiser in the most challenging setting. Our submillilux dataset can be used as a challenge for future denoising algorithms [1].

Paired clean/noisy bursts of static scenes. We collect 10 clips of grayscale and color targets, each with one clean image along with a burst of 100-900 noisy images, resulting in a dataset with 2558 noisy images. We use this dataset exclusively for the noise generator training. In addition to this, we acquire a more complex dataset of 67 clean/noisy image pairs consisting of 16 noisy bursts for each clean image. This second dataset contains indoor and outdoor scenes with various lighting conditions. We use this dataset both for our noise generator and denoiser training.

Unpaired clean RGB+NIR videos. With our trained noise generator, we can generate unlimited amounts of clean/noisy pairs from clean videos. Given an absence of open-sourced RGB+NIR RAW datasets, we collect our own dataset of noiseless video clips (unpaired). We collect 10 video sequences which we break into 166 video clips for training and 10 for testing. The videos are taken at different frame rates of both indoor and outdoor scenes. We capture these images at a low-gain setting during the daytime.

To augment our dataset, we utilize 329 video clips from the MOT video challenge [34], which we then unprocess [8] to resemble RAW video clips. While these video clips have significant motion, they have a different distribution of colors than the raw data from our camera. Due to this, we utilize the MOT videos during our initial pre-training step, then refine our denoiser using only the video clips from our camera.

Submillilux RGB+NIR videos. To test our method in the lowest light settings, we collected videos in a remote location on a clear, moonless night (outside of most of the night-glow from cities). Throughout our experiments, no outside light sources were used to illuminate the scene. The illuminace, measured by a PR-810 Prichard Photometer, ranged within 0.6-0.7 mlx, which is within the range expected for a clear moonless night. Videos were taken with exposures ranging within 0.1-0.2ms per image, corresponding to 10-5fps. All videos were taken with the largest lens aperture to maximize the amount of light hitting the sensor, and at the highest gain settings for the camera.

5. Video Denoising

Now that we can generate clean/video pairs, our next step is to train a denoiser that will generalize well to real noisy video clips from our camera. Inspired by burst denoising, in which a burst of multiple noisy frames are used together to denoise a central frame, we choose a network architecture that can operate on multiple frames at a time. This is beneficial because denoising a burst of images can improve PSNR over single-image denoising, especially in photon-starved regimes. In addition, video denoising can help us maintain temporal consistency across frames and reduce flickering throughout the denoised video.

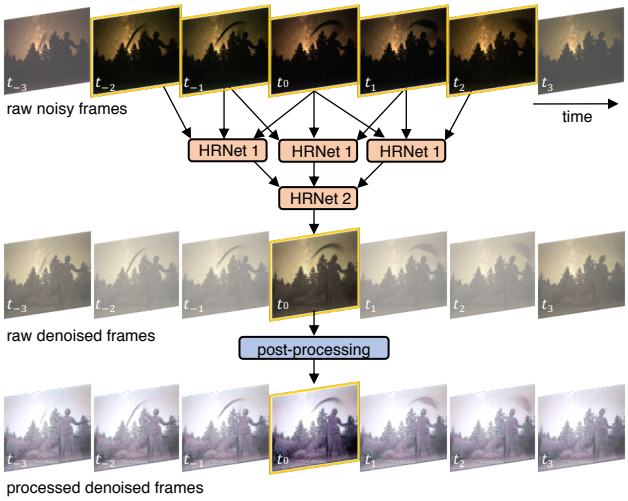

Figure 4. Denoising network. Our denoising network has a similar overall structure to FastDVDnet [49], sequentially taking in 5 noisy RAW images to produce 1 denoised RAW image. After denoising, off the shelf post-processing (e.g. white-balancing, histogram equalization) is applied to produce the final denoised video.

5.1. Denoising Network

For our denoising network, we build off of FastDVDNet [49], which is a state-of-the-art video denoiser that implicitly handles motion estimation within the network. We modify this network by replacing the

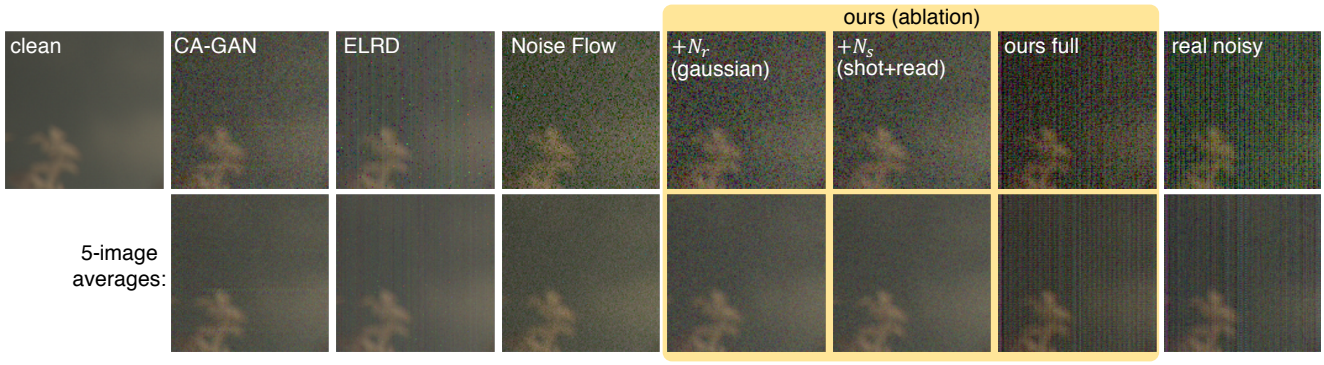

Figure 5. Noise model comparison. We show example image patches with our noise model vs. alternative noise models, as well as the mean of this image patch over 5 samples. Our synthetic noise appears more similar to the real noise than alternative methods and closely matches the average noise as well.

U-Net denoising blocks with an HRNet from [48], which we find leads to better temporal consistency across our denoised video than the original U-Net architecture. Our denoiser operates on RAW video sequences, Fig. 4, and off-the-shelf post-processing is used produce the final output. See Suppl. for our full denoiser network architecture and our evaluation against the original FastDVDNet architecture.

5.2. Training

We train the denoiser on a combination of synthetic noisy video clips and real still images from camera. First, we pre-train our network for 500 epochs using a combination of real paired stills, synthetic noisy clips from our camera, and synthetic noisy clips from the MOT dataset to help prevent overfitting. After pretraining, we refine the model for 817 epochs on our real still images and synthetic clips from our camera. All images are cropped to 512×512 patches throughout training. We use a combination a perceptual loss (LPIPS) [59] with an $\mathcal{L}_1$ loss for our training objective, choosing only the first 3 RAW channels for the LPIPS loss, which requires a 3-channel image. We gamma correct our ground truth images with gamma $= (1/2.2)$, thereby training the denoising network to output a gamma-corrected image. We found that this outperformed applying the gamma correction after denoising. For both pre-training and refinement, we utilize the Adam optimizer [30] with learning rate 1e-4 and all default parameters.

5.3. Post-processing

Our denoiser is trained on the raw images from the camera. In doing so, this system can work with a number of different post-processing pipelines. To display our final images, we apply the following post-processing steps: demosaicing via bilinear interpolation, white balancing, and histogram equalization. We note that our denoised images are already in a gamma-corrected space. We display the RGB channels of the video clips, omitting the NIR channel in our visualization. We expect that manual post-processing in Adobe Lightroom or a comparable platform could further improve the contrast and perceptual quality of our images.

6. Evaluation

First, we evaluate our noise generator performance against several existing noise models for low-light imaging, as well as perform an ablation analysis on the components of our noise model. Next, we compare our noise generator + video denoiser pipeline against several existing denoising schemes. We quantitatively compare on a held-out dataset of still noisy/clean image pairs. Finally, we qualitatively compare on our submillilux dataset of noisy videos, which do not contain ground-truth labels for a quantitative comparison.

6.1. Noise generator

After training our noise generator, we assess its performance on a held-out dataset consisting of 832 128×128 video patches. Each patch has 4 color channels and 5 temporal channels. We calculate the KL divergence between our synthetic noise and the real noisy clips after subtracting the clean image. We compare against a non-deep low-light noise model (ELD) [56], as well as two deep-learning-based noise models, CA-GAN [12] and Noise Flow [2]. ELD is hand-calibrated using dark frames and grayscale frames to fit to several distributions for different noise sources. Following this calibration scheme, we found that our noise distributions are very different from those described in [56], likely due to our extremely high gain settings, predominant fixed pattern noise, and periodic components, leading to poor performance of this model. CA-GAN is a camera-aware noise model that takes in a clean image, an estimated shot and read noise image, as well as an example real noisy image from the camera in order to synthesize a signal-dependent noisy image. We use this model off-the-shelf, but find that it generalizes poorly to our camera and noise. Similarly, Noise Flow is designed to work with multiple gain settings and lighting conditions, but does not generalize to our extreme low light, high gain setting.

We summarize our findings in Table 1, and show an example synthetic noisy patch from each method in Fig. 5. We can clearly see that both Noise Flow and CA-GAN miss the significant banding noise (column offsets) present in our real noisy clips. ELD captures the banding noise pattern well, but misses other components of the noise and does match the real noisy clips visually or in terms of KL divergence.

Noise model	KLD
ELD [56]	1.361
Noise Flow [2]	0.386
CA-GAN model [12]	0.513
Ours (ablation)	
N_r (Gaussian)	0.400
$N_s + N_r$ (shot + read noise)	0.400
$N_s + N_r + N_q$	0.122
$N_s + N_r + N_q + N_{row} + N_{row_t}$	0.118
$N_s + N_r + N_q + N_{row} + N_{row,t} + N_p$	0.113
$N_s + N_r + N_q + N_{row} + N_{row,t} + N_p + N_f$	0.138
$N_s + N_r + N_q + N_{row} + N_{row,t} + N_p + N_{f*}$	0.084
Full model:	**0.069**

Table 1. We compare our noise generator to prior work, representative of different approaches to modeling noise distributions. Our method significantly outperforms all baselines. We also present an ablation of components modeled by our noise generators. See Figure 5 for a visual comparison.

Ablation of noise parameters. We ablate different noise components of our generator in Table 1 and show a qualitative comparison in Figure 5. As before, we calculate the KL divergence between synthetic and real noisy patches, and we find that each component of our noise model improves the KL divergence. Specifically, shot, read, quantization, and row noise, were all used ELD [56], but were hand-calibrated. Here, we automatically calibrate the different noise components through our GAN training, resulting in better performance than the hand-calibrated model. In addition, our model takes into account the noise behavior over time by including components that are constant over multiple image patches (temporal row noise, fixed pattern noise). As seen in the 5-images averages on the bottom of Figure 5, our noise model closely matches the average noise, which is important for video denoising. Adding a periodic noise component makes our noise better match the Fourier spectrum of the real noise, which has several prominent peaks in Fourier space (see Suppl. for details). Adding our measured fixed pattern, N_f, improves the behavior over time, and learning the fixed pattern, N_{f*} further improves the KL divergence, but at the price of risking overfitting since this is a pixelwise addition of an image to the synthetic noise. In our final model, we use a measured fixed pattern and a learned U-Net which can account for noise that we do not specifically model, such as chromatic effects, or enhance our Gaussian noise approximations to better match the true noise distributions. Our final noise model produces synthetic noise that closely matches the real noise for a single noise instance, over time, and in Fourier space.

6.2. Full pipeline: video denoising

Next, we evaluate our video denoiser trained using combination real and synthetic noisy samples against existing denoisers. First, we quantitatively compare against several alternative methods using our dataset of 21 still clean/noisy bursts (since we do not have ground truth for our noisy video clips). We split up our comparison into two categories: single-image denoising methods and video denoising methods, which take in multiple clips at a time.

Method	PSNR	SSIM	LPIPS
Single Image Methods:			
Noise2Self [6]	20.11	0.210	0.545
Unprocesing [8]	12.86	0.249	0.355
L2SID (pretrained) [14]	13.6	0.512	0.338
L2SID (retrained) [14]	26.9	0.892	0.198
Video Methods:			
V-BM4D [39]	16.2	0.322	0.419
pretrained PaCNet [52]	13.65	0.512	0.338
pretrained FastDVDnet [49]	23.8	0.618	0.282
ours	**27.7**	**0.931**	**0.078**

Table 2. Performance on still images from test set.

For single-image denoising, we compare against two pretrained deep denoising methods: Unprocessing [8], which operates on RAW images and is trained using different read and shot noise levels, as well as L2SID [14] which takes in a raw noisy image and jointly denoises and processes the image. Both of these methods do not perform well on our dataset, due to the extreme noise in our raw measurements. We also retrained L2SID [14] using our still image pairs, resulting in better performance single-images, but significant flickering over time for video (see Suppl. for example). Noise2Self, a self-supervised approach, does poorly with our noise due to its highly structured content (e.g. correlated lines for the row offsets), and results in denoised images with prominent line artifacts. These results are summarized in Table 2, with full images shown in the Suppl.

For video denoising, we feed in 5 noisy clips to each denoiser, then compare against a single still ground-truth image. We compare our method against V-BM4D (a classic video denoising method), as well as two pre-trained state of the art deep video denoisers, FastDVDnet [49] and PaCNet [52]. Both of these models use additive Gaussian noise, so as expected, they do not perform well for our real noisy clips. FastDVDnet, which is designed to operate at multiple noise levels, outperforms PaCNet [52], which is designed for a specific Gaussian noise level. Our denoising method, which is based on a modified FastDVDnet and trained using our noise generator, achieves the best

Figure 6. Results on noisy video clips taken at 10 fps in 0.0006 lx. The input sequence (left), V-BM4D, pretrained FastDVDnet, and our results are shown. Our method maintains more details throughout the clip and does not contain the prevalent streaking artifacts that are present in V-BM4D and the pretrained FastDVDnet. See supplement for full video clips. (Digital zoom recommended.)

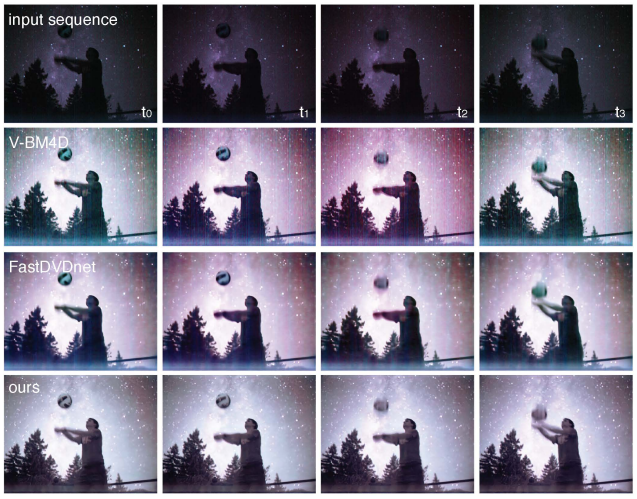

Figure 7. Results on noisy video clip, showing performance over time for a video taken at 10 fps in 0.6 mlx. Our method achieves better temporal consistency than V-BM4D or the pretrained FastDVDnet, and also has fewer artifacts within each frame. See supplement for full video clips. (Digital zoom recommended.)

performance. This demonstrates the importance of having a realistic noise model during denoiser training.

Next, we qualitatively compare our performance on our unlabeled dataset of submillilux video clips. We show the performance of our method as compared to V-BM4D and the pretrained FastDVDnet in Figure 6, with additional video comparisons available in the Suppl. Our method had fewer horizontal streak artifacts than the other methods, maintains more details such as stars, and has better overall image quality. We can clearly see the Milky Way in the denoised videos, and our method is robust to fast-moving objects in the background (e.g. we capture a shooting star in Fig. 6). When viewing the adjacent frames in the video clip, Fig. 7, our method has less flickering than V-BM4D and the pretrained FastDVDnet, which both have significant flickering between frames, likely due to the significant noise present in the raw data.

Finally, we perform a perceptual experiment with blind randomized A/B tests between our method, V-BM4D, FastDVDNet, and L2SID using 10 clips from our video dataset. Throughout 300 comparisons with 10 workers, our method is rated as having superior image quality than the alternative methods over 95% of the time (details in Suppl.).

7. Conclusion and Discussion

We have demonstrated photorealistic video denoising at submillilux levels of illumination for the first time. We achieved this through a combination of excellent camera hardware (a low-light optimized RGB+NIR camera), a physics-inspired noise generator used to generate realistic noisy video clips, and a video denoiser trained using a combination of real still images and synthetic noisy video clips. Our work showcases the power of deep-learning-based denoising for extremely low-light settings. We hope that this work leads to future scientific discoveries in extremely low light levels (e.g. studying nocturnal animal behavior in moonless conditions or under a forest canopy), and will help push the limits of robot vision in extremely dark settings. Potential misuse of this work includes night-time surveillance or use in conjunction with weapons systems.

Our approach has limitations. First, our noise generator is limited to producing noise that mimics a single gain setting (in this case, the highest gain). Future work could expand the noise generator model to work with multiple camera gains/ISOs. Next, our denoised night videos have muffled colors due to the dominance of NIR over RGB at night. Work in style transfer and recoloring could further improve the visual appearance of the denoised video clips by enhancing embedded color cues or synthesizing realistic-looking colors. Finally, the performance of the denoiser may be improved in the future through class-aware denoising [44] and joint denoising/segmentation.

References

[1] Dancing under the stars: video denoising in starlight dataset. http://kristinamonakhova.com/starlight_denoising/#dataset. Accessed: 2022-03-21.

[2] Abdelrahman Abdelhamed, Marcus A Brubaker, and Michael S Brown. Noise flow: Noise modeling with conditional normalizing flows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3165–3173, 2019.

[3] Abdelrahman Abdelhamed, Stephen Lin, and Michael S Brown. A high-quality denoising dataset for smartphone cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1692–1700, 2018.

[4] Josue Anaya and Adrian Barbu. Renoir–a dataset for real low-light image noise reduction. *Journal of Visual Communication and Image Representation*, 51:144–154, 2018.

[5] European Machine Vision Association. Emva standard 1288: Standard for characterization of image sensors and cameras. 2012.

[6] Joshua Batson and Loic Royer. Noise2self: Blind denoising by self-supervision. In *International Conference on Machine Learning*, pages 524–533. PMLR, 2019.

[7] Assim Boukhayma. Low-noise cmos image sensors. In *Ultra Low Noise CMOS Image Sensors*, pages 13–34. Springer, 2018.

[8] Tim Brooks, Ben Mildenhall, Tianfan Xue, Jiawen Chen, Dillon Sharlet, and Jonathan T Barron. Unprocessing images for learned raw denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11036–11045, 2019.

[9] Antoni Buades, Bartomeu Coll, and J-M Morel. A non-local algorithm for image denoising. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 60–65. IEEE, 2005.

[10] Harold C Burger, Christian J Schuler, and Stefan Harmeling. Image denoising: Can plain neural networks compete with bm3d? In *2012 IEEE conference on computer vision and pattern recognition*, pages 2392–2399. IEEE, 2012.

[11] Edmund Butt. Night on earth. Plimsoll Productions, 2020.

[12] Ke-Chi Chang, Ren Wang, Hung-Jin Lin, Yu-Lun Liu, Chia-Ping Chen, Yu-Lin Chang, and Hwann-Tzong Chen. Learning camera-aware noise models. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2020.

[13] Chen Chen, Qifeng Chen, Minh N Do, and Vladlen Koltun. Seeing motion in the dark. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3185–3194, 2019.

[14] Chen Chen, Qifeng Chen, Jia Xu, and Vladlen Koltun. Learning to see in the dark. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3291–3300, 2018.

[15] Jingwen Chen, Jiawei Chen, Hongyang Chao, and Ming Yang. Image blind denoising with generative adversarial network based noise modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3155–3164, 2018.

[16] Michele Claus and Jan van Gemert. Videnn: Deep blind video denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[17] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on image processing*, 16(8):2080–2095, 2007.

[18] Weisheng Dong, Guangming Shi, Yi Ma, and Xin Li. Image restoration via simultaneous sparse coding: Where structured sparsity meets gaussian scale mixture. *International Journal of Computer Vision*, 114(2):217–232, 2015.

[19] Thibaud Ehret, Axel Davy, Jean-Michel Morel, Gabriele Facciolo, and Pablo Arias. Model-blind video denoising via frame-to-frame training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11369–11378, 2019.

[20] Michael Elad and Michal Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image processing*, 15(12):3736–3745, 2006.

[21] Alessandro Foi. Clipped noisy images: Heteroskedastic modeling and practical denoising. *Signal Processing*, 89(12):2609–2629, 2009.

[22] Alessandro Foi, Mejdi Trimeche, Vladimir Katkovnik, and Karen Egiazarian. Practical poissonian-gaussian noise modeling and fitting for single-image raw-data. *IEEE Transactions on Image Processing*, 17(10):1737–1754, 2008.

[23] Clément Godard, Kevin Matzen, and Matt Uyttendaele. Deep burst denoising. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 538–554, 2018.

[24] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014.

[25] Ishaan Gulrajani, Faruk Ahmed, Martín Arjovsky, Vincent Dumoulin, and Aaron C. Courville. Improved training of wasserstein gans. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5767–5777, 2017.

[26] Samuel W Hasinoff. Photon, poisson noise., 2014.

[27] 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):1–12, 2016.

[28] Haiyang Jiang and Yinqiang Zheng. Learning to see moving objects in the dark. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7324–7333, 2019.

[29] Almut Kelber, Anna Balkenius, and Eric J Warrant. Scotopic colour vision in nocturnal hawkmoths. *Nature*, 419(6910):922–925, 2002.

[30] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[31] Mikhail Konnik and James Welsh. High-level numerical simulations of noise in ccd and cmos photosensors: review and tutorial. *arXiv preprint arXiv:1412.4031*, 2014.

[32] Alexander Krull, Tim-Oliver Buchholz, and Florian Jug. Noise2void-learning denoising from single noisy images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2129–2137, 2019.

[33] Alexander Krull, Tomáš Vičar, Mangal Prakash, Manan Lalit, and Florian Jug. Probabilistic noise2void: Unsupervised content-aware denoising. *Frontiers in Computer Science*, 2:5, 2020.

[34] Laura Leal-Taixé, Anton Milan, Ian Reid, Stefan Roth, and Konrad Schindler. Motchallenge 2015: Towards a benchmark for multi-target tracking. *arXiv preprint arXiv:1504.01942*, 2015.

[35] Marc Lebrun, Antoni Buades, and Jean-Michel Morel. A nonlocal bayesian image denoising algorithm. *SIAM Journal on Imaging Sciences*, 6(3):1665–1688, 2013.

[36] Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. Noise2noise: Learning image restoration without clean data. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 2971–2980. PMLR, 2018.

[37] Ch Leinert, S Bowyer, LK Haikala, MS Hanner, MG Hauser, A-Ch Levasseur-Regourd, I Mann, K Mattila, WT Reach, W Schlosser, et al. The 1997 reference of diffuse night sky brightness. *Astronomy and Astrophysics Supplement Series*, 127(1):1–99, 1998.

[38] 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):1–16, 2019.

[39] Matteo Maggioni, Giacomo Boracchi, Alessandro Foi, and Karen Egiazarian. Video denoising, deblocking, and enhancement through separable 4-d nonlocal spatiotemporal transforms. *IEEE Transactions on image processing*, 21(9):3952–3966, 2012.

[40] Ben Mildenhall, Jonathan T Barron, Jiawen Chen, Dillon Sharlet, Ren Ng, and Robert Carroll. Burst denoising with kernel prediction networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2502–2510, 2018.

[41] Tobias Plotz and Stefan Roth. Benchmarking denoising algorithms with real photographs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1586–1595, 2017.

[42] Javier Portilla, Vasily Strela, Martin J Wainwright, and Eero P Simoncelli. Image denoising using scale mixtures of gaussians in the wavelet domain. *IEEE Transactions on Image processing*, 12(11):1338–1351, 2003.

[43] Mangal Prakash, Manan Lalit, Pavel Tomancak, Alexander Krul, and Florian Jug. Fully unsupervised probabilistic noise2void. In *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, pages 154–158. IEEE, 2020.

[44] Tal Remez, Or Litany, Raja Giryes, and Alex M Bronstein. Class-aware fully convolutional gaussian and poisson denoising. *IEEE Transactions on Image Processing*, 27(11):5707–5722, 2018.

[45] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[46] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 60(1-4):259–268, 1992.

[47] Hema Somanathan, Renee M Borges, Eric J Warrant, and Almut Kelber. Visual ecology of indian carpenter bees i: light intensities and flight activity. *Journal of Comparative Physiology A*, 194(1):97–107, 2008.

[48] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5693–5703, 2019.

[49] Matias Tassano, Julie Delon, and Thomas Veit. Fastdvdnet: Towards real-time deep video denoising without flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1354–1363, 2020.

[50] Linh Duy Tran, Son Minh Nguyen, and Masayuki Arai. Gan-based noise model for denoising real images. In *Proceedings of the Asian Conference on Computer Vision*, 2020.

[51] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9446–9454, 2018.

[52] Gregory Vaksman, Michael Elad, and Peyman Milanfar. Patch craft: Video denoising by deep modeling and patch matching. *arXiv preprint arXiv:2103.13767*, 2021.

[53] Richard H Vollmerhausen and Tana Maurer. Night illumination in the visible, nir, and swir spectral bands. In *Infrared Imaging Systems: Design, Analysis, Modeling, and Testing XIV*, volume 5076, pages 60–69. International Society for Optics and Photonics, 2003.

[54] Wei Wang, Xin Chen, Cheng Yang, Xiang Li, Xuemei Hu, and Tao Yue. Enhancing low light videos by exploring high sensitivity camera noise. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4111–4119, 2019.

[55] Eric Warrant. Vision in the dimmest habitats on earth. *Journal of Comparative Physiology A*, 190(10):765–789, 2004.

[56] Kaixuan Wei, Ying Fu, Jiaolong Yang, and Hua Huang. A physics-based noise formation model for extreme low-light raw denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2758–2767, 2020.

[57] Guoshen Yu, Guillermo Sapiro, and Stéphane Mallat. Solving inverse problems with piecewise linear estimators: From gaussian mixture models to structured sparsity. *IEEE Transactions on Image Processing*, 21(5):2481–2499, 2011.

[58] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE transactions on image processing*, 26(7):3142–3155, 2017.

[59] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[60] Yi Zhang, Hongwei Qin, Xiaogang Wang, and Hongsheng Li. Rethinking noise synthesis and modeling in raw denoising. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4593–4601, 2021.