Seeing Motion in the Dark

Chen Chen
UIUC

Qifeng Chen
HKUST

Minh N. Do
UIUC

Vladlen Koltun
Intel Labs

Abstract

Deep learning has recently been applied with impressive results to extreme low-light imaging. Despite the success of single-image processing, extreme low-light video processing is still intractable due to the difficulty of collecting raw video data with corresponding ground truth. Collecting long-exposure ground truth, as was done for single-image processing, is not feasible for dynamic scenes. In this paper, we present deep processing of very dark raw videos: on the order of one lux of illuminance. To support this line of work, we collect a new dataset of raw low-light videos, in which high-resolution raw data is captured at video rate. At this level of darkness, the signal-to-noise ratio is extremely low (negative if measured in dB) and the traditional image processing pipeline generally breaks down. A new method is presented to address this challenging problem. By carefully designing a learning-based pipeline and introducing a new loss function to encourage temporal stability, we train a siamese network on static raw videos, for which ground truth is available, such that the network generalizes to videos of dynamic scenes at test time. Experimental results demonstrate that the presented approach outperforms state-of-the-art models for burst processing, per-frame processing, and blind temporal consistency.

1. Introduction

We are interested in capturing videos of dynamic scenes in the dark: people dancing in the moonlight, an intimate conversation by candlelight, a nocturnal animal foraging. Can such scenes ever be captured effectively, in motion, by widely accessible consumer-grade cameras?

Extreme low-light videography is challenging due to low photon counts. Using high ISO can increase brightness but also amplifies noise. Aperture size is limited in consumer-grade cameras and mobile devices. Flash changes the character of the scene and is problematic for videography. And long exposure times (seconds or tens of seconds) are not feasible for videos of dynamic scenes. This leaves us with computational techniques for low-light video processing.

Researchers have developed many techniques to reduce noise for low-light imaging [40, 38, 13, 33, 16, 11, 20, 45, 46, 3, 8, 29, 22, 47]. These techniques generally assume that images are captured in somewhat dim environments with moderate levels of noise. In addition, these methods are often trained and evaluated using synthetic noise models, which do not reflect the severe quantization, bias, and clipping that arise in extreme low-light conditions.

Recent work proposed end-to-end learning for low-light image processing [9, 41]. The idea is to train a deep network on a dataset of short-exposure raw and long-exposure reference images, such that the network learns the image processing pipeline to maximize low-light imaging performance. However, these datasets contain images of static scenes and do not address video, and the trained networks exhibit temporal instability that is not easily remedied with post-hoc temporal consistency enhancement. Another approach to low-light photography that has seen significant recent progress is burst processing [18, 28, 35, 15]. However, these methods are generally not designed for video capture (e.g., due to the use of 'lucky imaging') and require dense correspondence estimation across the input frames, which can fail due to the massive noise conditions we consider.

In this paper, we tackle deep processing of extreme low-light video, from raw sensor data to sRGB output. This brings challenges beyond those presented by individual low-light images. For example, long-exposure videos of dynamic scenes cannot be obtained, since videos must be acquired at video rate. Thus 'ground-truth' long-exposure video of dark dynamic scenes is not available. It is thus not clear how to train models that produce temporally consistent output in this regime.

To study this problem, we collect a new low-light video dataset and present a systematic approach for this problem. We captured 202 static raw videos for training and evaluation, each of which has corresponding long-exposure ground truth. We also capture real-world low-light videos with hand shake and subject motion. For these videos, long-exposure ground truth is not available, and they are used for perceptual experiments. Using the collected data, we develop a new learning-based pipeline for extreme low-light video processing. The proposed method involves training a deep siamese network [6] with a specially designed loss that encourages temporal stability. We show that the network can be trained on static videos but generalizes

to dynamic scenes. Experimental results demonstrate that our method significantly outperforms state-of-the-art approaches, as measured by reference-based distortion metrics as well as reference-free perceptual studies.

2. Related Work

Single-image denoising. Image denoising has been extensively studied [40, 38, 13, 33, 16, 11, 20]. Most approaches are based on specific image priors such as smoothness, sparsity, low rank, or self-similarity. Learning-based methods further advanced performance in recent years [46, 3, 10, 8, 29, 22, 47, 7]. Lehtinen et al. [26] showed that a denoising network can be trained without clean ground-truth if the noise is unbiased. Some networks can denoise and demosaic images jointly [21, 14] or even replace much of the image processing pipeline [9, 41]. However, as demonstrated in our experiments, frame-by-frame processing can exhibit significant temporal artifacts when applied to video.

Multiple-image denoising. When video or burst images are available, noise can be reduced using spatial and temporal correlations. Liu et al. [28] and Hasinoff et al. [18] propose to merge a burst of images by robust and efficient aligment methods. Godard et al. [15] propose to use recurrent networks for multi-frame denoising, where the burst sequence needs to be pre-warped to the reference frame. Mildenhall et al. [35] propose to align and denoise bursts via learned per-pixel kernels.

In these works, burst denoising involves reference image selection and outputs a single frame. In contrast, video denoising is more challenging since every frame needs to be processed for the output video, which needs to be temporally consistent. State-of-the-art video denoising methods include VBM4D [32] and non-local Bayes [25], which rely on grouping similar patches and jointly filtering them to form the result. When noise is small or moderate, these methods can achieve excellent results. However, these methods do not address the biases present in extreme low-light data due to clipping and quantization.

Low-light image and video enhancement. Methods have been developed that can enhance brightness and contrast of images and videos acquired in moderately dim environments [12, 34, 30, 36, 17, 31]. However, these methods generally assume that image details are preserved in the sRGB camera output. In contrast, in the extreme low-light settings we consider, the associated challenges in the image processing pipeline are not addressed by these models, e.g., noise and color cast.

Noisy image datasets. Image and video denoising datasets have traditionally been created using synthetic noise models, such as Gaussian and Poisson noise, applied to clean images and videos; see Plötz and Roth for review [37]. More recently, datasets were created with real noisy images produced by imaging sensors. These include the RENOIR dataset [4], the Darmstadt Noise Dataset (DND) [37], the Smartphone Image Denoising Dataset (SIDD) [2], and the See in the Dark (SID) dataset [9]. Burst image datasets [28, 18, 15] have also been used for low-light image denoising; however, the bursts are short (less than 10 frames) and scene motion is small. We collect a new dataset of extreme low-light raw videos, with up to 110 frames each. To the best of our knowledge, this is the first public dataset with real-world low-light raw video sequences.

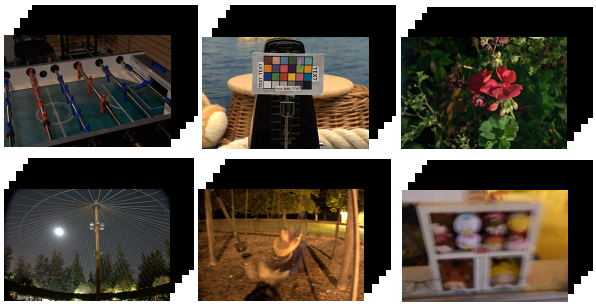


Figure 1. Example videos from the dataset. For each video, the first image is the long exposure reference image. The later frames are the short-exposure images, which are dark in extreme low-light conditions. Note that the last two videos in the second row are from the test set, which contains dynamic scenes. The reference long-exposure images are blurry due to subject and camera motion and thus cannot serve as ground truth for quantitative evaluation.

3. Dark Raw Video Dataset

Raw video processing has been rarely studied due to limited available data. We collected a new Dark Raw Video (DRV) dataset to bridge this gap. We used a Sony RX100 VI camera, which can capture raw image sequences at approximately 16 to 18 frames per second in continuous shooting mode, and the buffer can keep around 110 frames in total. This is equivalent to 5.5-second videos at 20 fps. The resolution for the Bayer image is 3672×5496. The dataset contains both indoor and outdoor scenes.

Another challenge for learning-based raw video processing is the difficulty to collect noise-free videos in dark conditions. Following the SID dataset [9], we capture low-light raw data and corresponding long-exposure images. However, this scheme only works for static scenes. Therefore, we collect two sets of videos: one contains static videos with corresponding long-exposure ground truth, and the other contains dynamic videos without ground truth. We hypothesized that a model trained on static videos can generalize to some extent to dynamic videos. Examples are shown in Fig. 1. Most scenes in the dataset are in the 0.5 to

5 lux range.

To collect a static video, we used a tripod and controlled the camera remotely via a mobile app. The long and short exposure image sequences are perfectly aligned. We have 202 static videos for training and quantitative evaluation. We randomly divide them into approximately 64% for training, 12% for validation, and 24% for testing. Some scenes are in different lighting conditions (e.g., light sources with different color temperatures, illuminance, and positions). Videos for the same scene are distributed within one of the training, testing, and validation sets but not across these sets.

We capture a separate set of dynamic video sequences. The motion is due to scene motion, camera motion, or both. These videos do not have ground-truth long-exposure references. They are used for perceptual experiments.

The exposure differences between the raw low-light input and the long-exposure ground truth in the static set are between factors of 120 and 300. We apply digital gains on the low-light raw frames in preprocessing based on these exposure ratios to match the brightness of the corresponding long exposure images.

Noise analysis. We analyze the noise distribution in the DRV dataset and compare it with a synthetic noise model used in recent work [35]. In the synthetic model, a noisy pixel is assumed to be distributed according to

$$x_p \sim \mathcal{N}(y_p, \sigma_r^2 + \sigma_s y_p). \quad (1)$$

Here x_p is the noisy observation, y_p is the true pixel value, and σ_r and σ_s are parameters for read and shot noise. To simulate synthetic noise for comparison, we use the same sampling strategies for σ_r and σ_s as [35].

The low-light data is linearized by first subtracting the black level and then applied the digital gain mentioned above. After this, the overall intensity of the processed input matches that of the ground truth. (And thus of the synthetic noise model, which is applied to the ground truth for comparison.) The comparison is shown in Fig. 2. This figure shows the distribution of the real data, compared to the distribution of the synthetic noise model. The distributions are estimated via Parzen density estimation.

Perfectly clean data would correspond to a delta function at 1. As can be seen in the figure, the synthetic noise model is symmetric about 1. On the other hand, our real low-light data is severely biased, in part due to clipping and quantization. For example, there is a peak at zero because many sensor readings are too weak and are quantized to zero even in the 14-bit raw sensor data. Furthermore, the noise in the DRV data is an order of magnitude stronger than predicted by the synthetic model. (Note that the density is plotted on a logarithmic scale and observe the data at very high noise ratios, such as -10 and 10.) Overall, the average signal-to-noise ratio (SNR) for the synthetic model is 18.59 dB, while in the real data it is -3.24 dB. That is, the SNR in our low-light data, as measured in dB, is negative.

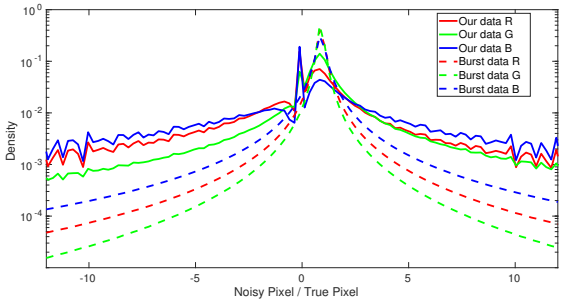


Figure 2. Comparison between real low-light noise in the DRV dataset and a synthetic noise model (used in [35]). The X-axis is the ratio between a noisy raw pixel and the corresponding ground-truth value. The Y-axis is the normalized density of these ratios across all pixels in the dataset, computed via kernel density estimation. Note that the Y-axis is in log-scale. Noise in our low-light data is stronger by an order of magnitude, and exhibits severe non-monotonic bias.

4. Method

Raw image and video processing involves the entire image processing pipeline. The system needs to be designed carefully, which is as important as the machine learning algorithms applied. Brooks et al. validated that a carefully designed raw image processing system can significantly improve results without significant changes in network structure and loss function [7]. We think the following criteria are desired for a low-light video processing system:

a). **Start from raw**. In our extreme low-light dataset, the raw sensor readings are extremely weak. In 8-bit JPEG camera output, most of the signal is destroyed and most pixel values are quantized to zero. We take the 14-bit raw frames as input.

b). **Model image processing pipeline**. A successful model should take care of the image processing pipeline during training [9, 7]. We train the network from the raw data to final sRGB output, which avoid error accumulation caused by multi-step optimization.

c). **Spatial and temporal denoising**. Both spatial and temporal correlations should be utilized to reduce noise.

d). **Generalization**. While ground truth is only available for static sequences in DRV, the trained model must generalize to dynamic videos.

e). **Temporal consistency**. The output video should be temporally stable, without salient flickering artifacts.

In accordance with these requirements, we designed a new learning-based pipeline that uses a deep network to process extreme low-light videos. The training is schematically summarized in Fig. 3. First, the raw Bayer video frames are preprocessed. The preprocessing includes Bayer to raw RGB conversion, black level subtraction, binning

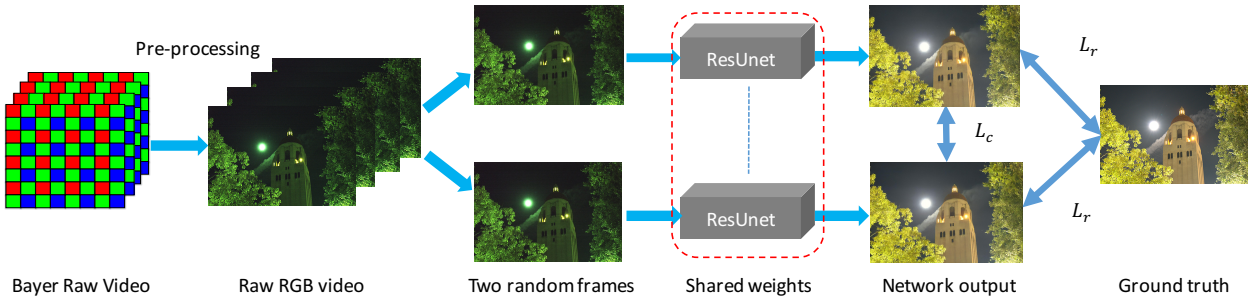


Figure 3. The entire training phase of our method on static videos with ground truth.

and global digital gain. The Bayer data is split into separate RGB channels to form the raw RGB where the green channel is obtained by averaging the two green pixels in each two-by-two block. We trade off resolution for image quality by applying 2×2 binning, which is a commonly used strategy for low-light imaging and applies to all the methods involved in the experiments. The pixel values are linearly scaled based on the exposure value (EV) difference and clipped to match the brightness and dynamic range of the ground truth. In addition, temporal noise is reduced by VBM4D [32] without the need of training data, which works for both static and dynamic videos. Dense correspondence is almost impossible to be estimated accurately in extreme noisy conditions, which is required in some existing methods [27, 18, 15] but not in our system. The result after these preprocessing stages is shown in Fig. 3 as "Raw RGB video". As the Bayer pattern has been destroyed during preprocessing, no demosaicing is applied.

The preprocessed raw RGB frames are fed to a deep network that is trained to perform all subsequent processing needed to obtain the results demonstrated in the ground-truth images. The network takes a single frame as input. For training, two frames from a static sequence in DRV are sampled at random and are fed to the network in siamese mode. Let $\hat{Y}^1$ and $\hat{Y}^2$ denote these two frames and let the ground truth for this sequence be denoted by Y^*. The loss for this training pair is defined as follows:

$$\mathcal{L}(\hat{Y}^1, \hat{Y}^2, Y^*) = \mathcal{L}_r + \mathcal{L}_c, \tag{2}$$

where $\mathcal{L}_r$ is referred to as the the recovery loss and $\mathcal{L}_c$ is called the self-consistency loss. They are defined as follows:

$$\mathcal{L}_r = \sum_l \frac{1}{N^l} \sum_{k=1,2} \|\Phi^l(Y^*) - \Phi^l(\hat{Y}^k)\|_1 \tag{3}$$

$$\mathcal{L}_c = \sum_l \frac{\lambda}{N^l} \|\Phi^l(\hat{Y}^1) - \Phi^l(\hat{Y}^2)\|_1. \tag{4}$$

Here Φ^l denotes the VGG [42] features at the l-th layer and N^l is the number of such features. λ is a regularization parameter and was empirically set to 0.05 for the results.

The recovery loss $\mathcal{L}_r$ encourages the output to be close to the ground truth. However, this alone is not sufficient for temporal consistency. Two outputs may have the same ℓ_1 distance to the ground truth in feature space, but may be far from each other. This corresponds to temporal instability (flickering). To alleviate temporal instability, we use the self-consistency loss, which encourages the two outputs to be close to each other.

The network produces output in sRGB space. We use a ResUnet structure akin to [24] by adding 16 residual blocks [19] to a Unet [39, 9].

Our method easily satisfies the first and second criteria discussed earlier. Noise is reduced using spatial and temporal correlations in preprocessing by VBM4D. Other temporal filters may also work for this purpose although not tried. The trained network can then adapt to the characteristics of the preprocessed input and optimize for fidelity given this input. The siamese network and self-consistency loss, used during training, encourages the network to produce temporally stable output. (As we shall see in the experiments, this temporal stability characteristic carries over into dynamic videos at test time.) Since the network operates on a single frame at test time, it generalizes to dynamic videos.

4.1. Implementation details

We implement our method using Tensorflow [1]. We found that training on complete images rather than patches is important to capture global statistics (e.g., for white balance). We train our model on an Nvidia Tesla V100 GPU with 32 GB of memory. We use the Adam optimizer [23] and the batch size is one. The initial learning rate is 10^{-4} and is reduced to 10^{-5} after 500 epochs. We train the network for 1000 epochs. We use the input, "conv1_2", "conv2_2", "conv3_2", and "conv4_2" layers of the VGG network as features in the loss.

4.2. Discussion of alternative options

Dense correspondence. Some existing methods rely on pre-warping [28, 18, 15] or learning to align [35] to reduce noise temporally. Almost all optical flow methods assume that little to no noise is present in the input frames. However, this assumption breaks down in our setting. Even after

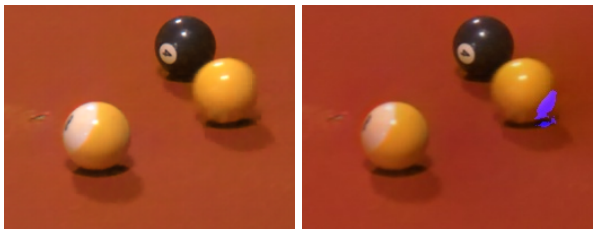
(a) (b) (c)

Figure 4. Optical flow on preprocessed raw RGB images. (a) and (b) are two consecutive frames from a dynamic DRV sequence. In this sequence, the metronome, with a text sign, is ticking, while the rest of the scene is static. (c) Optical flow between the two frames, estimated by PWC-Net [43]. The flow contains significant errors.

preprocessing with VBM4D, there is still substantial noise and artifacts. Fig. 4 shows the optical flow estimated on our data (dynamic sequence) using the state-of-the-art PWC-Net [43]. The flow has significant errors. This suggests that dense flow estimation is problematic in our conditions.

Denoising followed by color conversion. Another option is to learn joint alignment and denoising [35], followed by subsequent pipeline processing (e.g., to map from raw RGB to sRGB). As we will see in controlled experiments, such decoupled processing is suboptimal because the later processing stages do not optimally adapt to the characteristics of the input provided to them. In practice, the later processing stages significantly amplify errors from earlier stages.

Temporal consistency. Another possibility is to enhance temporal consistency in post-processing. Existing methods use the input videos to guide such enhancement [5, 24]. The underlying assumption is that the input video is temporally consistent. This is not true in our case. Due to the extremely low SNR, the input video is temporally unstable. Applying state-of-the-art temporal consistency techniques to our data (e.g., SID [9] for per-frame processing, followed by learned blind temporal consistency [24]) therefore yields severe visible artifacts, as shown in Fig. 5.



(a) (b)

Figure 5. (a) Result by SID [9]. (b) Applying learned blind temporal consistency [24] as post-processing results in visible artifacts. Note the blue patch on the ball.

5. Experiments

5.1. Experimental setup

Real camera processing pipelines are often commercial secrets. Following existing work [9], we use a simplified pipeline Rawpy (a Python wrapper for LibRaw) as the reference traditional non-learning-based pipeline. The long-exposure raw images are processed by Rawpy to form the sRGB ground truth. We use the metadata of the ground truth raw data for all Rawpy processing. This benefits the traditional pipeline, as the white balance estimation is worse in low-light conditions. For our method and SID [9], we use the preprocessed low-light raw frames as input and learn to convert the colors without the need for metadata. As one of our baselines, we train the kernel prediction network (KPN) [35] for spatial and temporal denoising with default settings using the author-provided code; the denoised results are followed by Rawpy to produce the sRGB output. Both VBM4D [32] and KPN use 8 frames for temporal denoising.

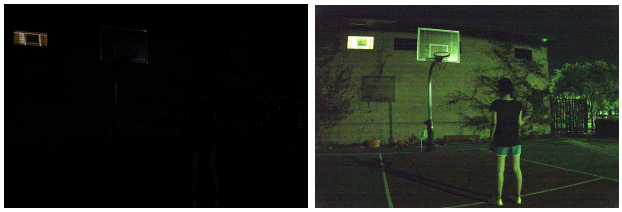
5.2. Image quality evaluation

We evaluate different methods on the static test videos. The 5th frame of the output video is compared with the ground truth using Peak Signal to Noise Ratio (PSNR) and the Structural Similarity Index (SSIM) [44]. The average results over the entire test set are listed in Table 1. Our method significantly outperforms the baselines. The ablations confirm the benefits of temporal preprocessing and the siamese network.

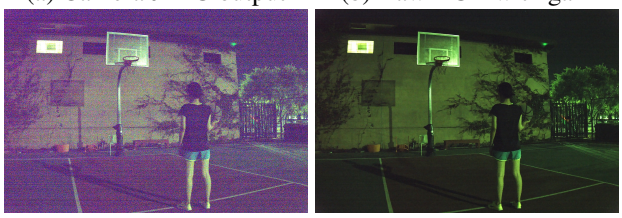
Table 1. Quantitative evaluation of image quality on the static video test set.

	PSNR (dB)	SSIM
Input+Rawpy	12.94	0.165
VBM4D [32]+Rawpy	14.77	0.315
KPN [35]+Rawpy	18.81	0.540
SID [9] w/o VBM4D	27.32	0.790
SID [9]	27.69	0.803
Ours	**28.26**	**0.815**
Ours w/o siamese	27.66	0.805
Ours w/o VBM4D [32]	27.62	0.803
Ours w/o both	27.26	0.793

An example is shown in Fig. 6. The camera output is almost completely black when the exposure time is fixed to 1/30 seconds in this dark environment. Applying digital gain on the raw RGB image, as shown in Fig. 6(b), reveals the content but also amplifies the noise and bias. As shown in Fig. 6(c), the traditional image processing pipeline results in color shift, where red and blue channels are boosted by white balance. VBM4D [32] can remove the noise to an extent, but cannot correct the color shift. As shown in Fig. 6(f), KPN learns to remove the noise in the raw color



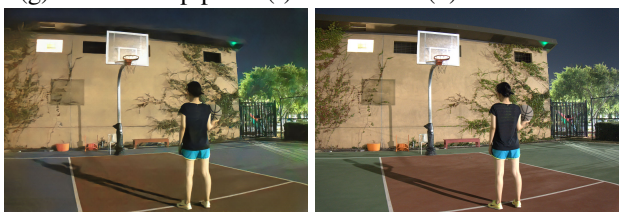
(a) Camera JPEG output (b) Raw RGB with gain



(c) Traditional pipe on (b) (d) VBM4D on (b)

(e) Traditional pipe on (d) (f) KPN denoising on (b)

(g) Traditional pipe on (f) (h) SID



(i) Ours (j) Ground truth

Figure 6. An example from a night-time sequence captured with a Sony RX100 VI camera with aperture f/4, ISO 2000, and 1/30 second exposure. This is a static DRV sequence, so ground truth is available for reference. Zoom in for details.

space. However, as shown in Fig. 6(g), when the traditional pipeline converts the raw color to sRGB and increases image contrast, it significantly amplifies the residual errors. This suggests that independent denoising followed by a traditional pipeline is sub-optimal. The denoising stage leaves some residual noise that can be boosted by later processing. Our method is trained end-to-end to avoid such error accumulation. In comparison with our results (Fig. 6(i)), SID [9] has strong artifacts in the sky and on the wall (Fig. 6(h)).

Since SID is the strongest baseline, we further compare with it in Fig. 8. While SID has strong denoising ability, it sometimes exhibits discoloration artifacts. For example, the letters in "Voice" and "Daily Post" in Fig. 8(a) (top) shift to a yellow tint. In Fig. 8(a) (bottom), the green box and orange book lose their color in the center. Such artifacts happen less frequently on the SID dataset [9], captured by a more expensive camera with longer exposure (up to 1/10 seconds), but occur more prominently in the DRV dataset. Although both SID and our method use the same input, our results have consistently higher quality.

5.3. Video quality evaluation

We further evaluate the video quality of SID and our method. Adopting the methodology of [24], we measure temporal error on every pair of consecutive frames using PSNR, SSIM, and mean absolute error (MAE) on the static test videos. We use the terms temporal PSNR (TPSNR), temporal SSIM (TSSIM), and temporal MAE (TMAE) to distinguish the temporal variants from single-image metrics. The results are shown in Table 2, where the images are scaled to [0, 1] for TMAE calculation. The table demonstrates that our method has much lower temporal errors than SID. We found that larger λ leads to smaller temporal errors, but at the cost of lower spatial accuracy, as illustrated in Fig. 7. $\lambda = 0.05$ was used as default for our results.

Table 2. Temporal errors on the static video test set for SID and our method.

	TPSNR (dB)	TSSIM	TMAE ($\times 10^2$)
SID [9] w/o VBM4D	33.72	0.939	1.56
SID [9]	37.05	0.961	1.05
Ours	**38.31**	**0.974**	**0.89**
Ours w/o siamese	37.76	0.969	0.98
Ours w/o VBM4D	34.64	0.953	1.38
Ours w/o both	34.55	0.952	1.40

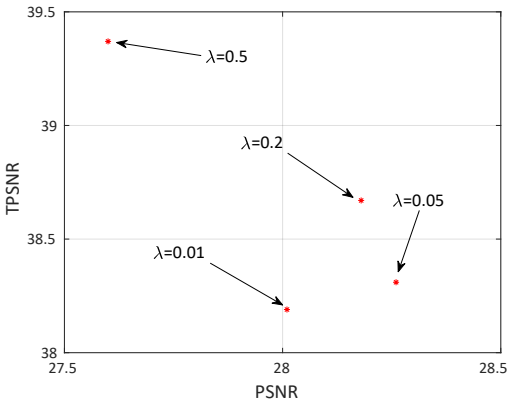
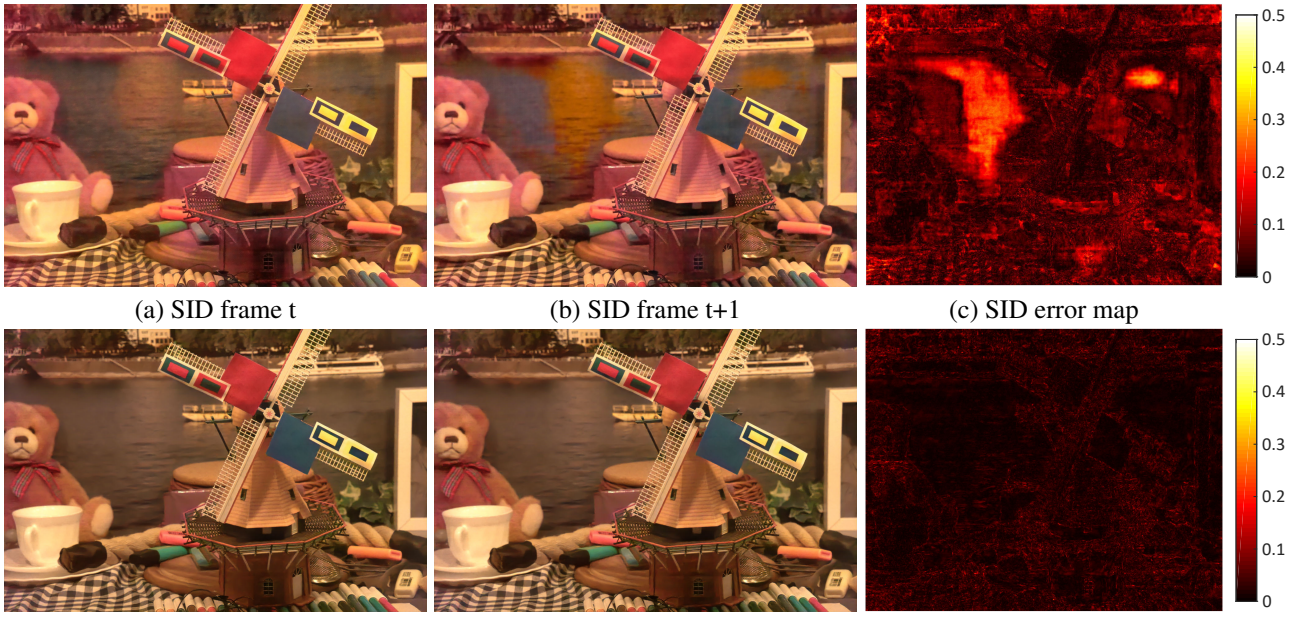


Figure 7. The parameter λ trades off spatial and temporal accuracy.

(a) SID (b) Ours (c) Ground truth

Figure 8. Image quality comparison with SID [9] on two examples. Note the discoloration artifacts in the demarcated regions.



(a) SID frame t (b) SID frame t+1 (c) SID error map

(d) Our frame t (e) Our frame t+1 (f) Our error map

Figure 9. The visual results of two consecutive frames on a static video. The error maps show per-pixel error, measured by Euclidean distance in [0,1] sRGB space. Brighter means larger errors.

Although the input frames are static, they contains strong flickering artifacts due to random noise. We further visualize the temporal errors in Fig. 9, which shows two consecutive frames from a static DRV sequence, processed by SID and by our method. The SID results exhibit temporal instability in the form of discoloration. Our method is much more stable.

5.4. Perceptual evaluation

We conduct a perceptual experiment that compares the results of SID and our approach on dynamic videos. In blind randomized A/B tests, we display corresponding video pairs and ask workers to indicate which of the two videos has better quality. Order is randomized both within and across pairs. 34 workers participated in the experiment, ranking results on 10 dynamic video sequences. Fig. 10 shows the

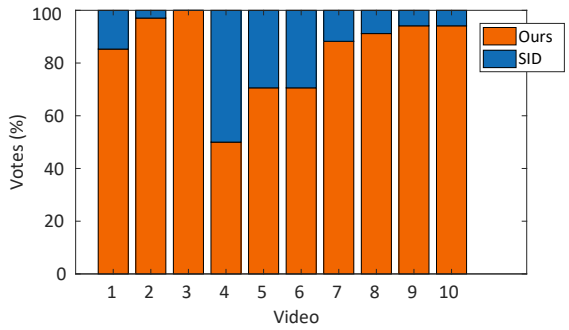


Figure 10. Perceptual experiment. Results of blind randomized A/B tests on 10 dynamic videos. The figure shows preferred percentage for each video.

results. Overall, the workers rate videos produced by our method as superior in quality in 84.12% of the comparisons. The result is statistically significant with $p < 10^{-3}$.

5.5. Extreme imaging

Finally, we demonstrate our method qualitatively in Fig. 11. Videos from an iPhone X and the Sony RX100 VI camera video mode are used for reference. In this mock birthday party video, illumination was provided by a single candle. This is a sub-lux setting. The iPhone video was captured using the auto mode. For the Sony video, we fixed the exposure time to 1/30 seconds while keeping the maximum aperture and ISO. The raw image sequences for SID and our method were captured with ISO 2000 in continuous shooting mode.

Light intensity is inversely proportional to the square of the distance from the source. We thus see in Fig. 11(a,b) that in the iPhone and Sony sequences only the birthday lady can be (dimly) made out in the image. Both SID and our method reveal the entire scene. However, the SID result suffers from both spatial and temporal artifacts, while our result is cleaner and more stable. This is video #10 in the perceptual experiment (Fig. 10), for which 94.1% of the comparisons are in favor of our result. Readers are encouraged to watch the supplementary video.

6. Conclusion

We presented a new dataset and a new method for learning extreme low-light video processing. We proposed a siamese network that preserves color while significantly suppressing spatial and temporal artifacts. The model was trained on static videos only but was shown to generalize to dynamic videos. Quantitative and qualitative results demonstrate that our method achieves superior performance over a range of baselines, particularly in the more extreme low-light scenarios. While the improvement is significant, certain failure modes remain. For example, our method (as well as the baselines) completely failed on moon-light videos (approximately 0.01-0.03 lux) using the same camera. Furthermore, we did not preserve high dynamic range due to the preprocessing to match the ground truth. The area around the candles in Fig. 11 is over-exposed. Exciting work remains in further pushing the boundaries in computational low-light imaging.

(a) iPhone X video frame

(b) Sony camera video frame

(c) SID result

(d) Our result

Figure 11. Video of a dynamic scene lit with a single candle. The illuminance is 0.73 lux at the birthday lady's ear.

References

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: a system for large-scale machine learning. In *OSDI*, 2016. 4

[2] Abdelrahman Abdelhamed, Stephen Lin, and Michael S Brown. A high-quality denoising dataset for smartphone cameras. In *CVPR*, 2018. 2

[3] Forest Agostinelli, Michael R. Anderson, and Honglak Lee. Adaptive multi-column deep neural networks with application to robust image denoising. In *NIPS*, 2013. 1, 2

[4] Josue Anaya and Adrian Barbu. Renoir-a dataset for real low-light image noise reduction. *arXiv:1409.8230*, 2014. 2

[5] Nicolas Bonneel, James Tompkin, Kalyan Sunkavalli, Deqing Sun, Sylvain Paris, and Hanspeter Pfister. Blind video temporal consistency. *ACM TOG*, 34(6), 2015. 5

[6] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a "siamese" time delay neural network. In *NIPS*, 1994. 1

[7] Tim Brooks, Ben Mildenhall, Tianfan Xue, Jiawen Chen, Dillon Sharlet, and Jonathan T Barron. Unprocessing images for learned raw denoising. In *CVPR*, 2019. 2, 3

[8] Harold C Burger, Christian J Schuler, and Stefan Harmeling. Image denoising: Can plain neural networks compete with bm3d? In *CVPR*, 2012. 1, 2

[9] Chen Chen, Qifeng Chen, Jia Xu, and Vladlen Koltun. Learning to see in the dark. In *CVPR*, 2018. 1, 2, 3, 4, 5, 6, 7

[10] Yunjin Chen and Thomas Pock. Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. *IEEE TPAMI*, 39(6):1256–1272, 2017. 2

[11] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE TIP*, 16(8):2080–2095, 2007. 1, 2

[12] Xuan Dong, Guan Wang, Yi Pang, Weixin Li, Jiangtao Wen, Wei Meng, and Yao Lu. Fast efficient algorithm for enhancement of low lighting video. In *ICME*, 2011. 2

[13] Michael Elad and Michal Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE TIP*, 15(12):3736–3745, 2006. 1, 2

[14] Michaël Gharbi, Gaurav Chaurasia, Sylvain Paris, and Frédo Durand. Deep joint demosaicking and denoising. *ACM TOG*, 35(6), 2016. 2

[15] Clément Godard, Kevin Matzen, and Matt Uyttendaele. Deep burst denoising. In *ECCV*, 2018. 1, 2, 4

[16] Shuhang Gu, Lei Zhang, Wangmeng Zuo, and Xiangchu Feng. Weighted nuclear norm minimization with application to image denoising. In *CVPR*, 2014. 1, 2

[17] Xiaojie Guo, Yu Li, and Haibin Ling. Lime: Low-light image enhancement via illumination map estimation. *IEEE TIP*, 26(2):982–993, 2017. 2

[18] 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 TOG*, 35(6), 2016. 1, 2, 4

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 4

[20] Felix Heide, Markus Steinberger, Yun-Ta Tsai, Mushfiqur Rouf, Dawid Pajak, Dikpal Reddy, Orazio Gallo, Jing Liu, Wolfgang Heidrich, Karen Egiazarian, et al. Flexisp: A flexible camera image processing framework. *ACM TOG*, 33(6), 2014. 1, 2

[21] Keigo Hirakawa and Thomas W Parks. Joint demosaicing and denoising. *IEEE TIP*, 15(8):2146–2157, 2006. 2

[22] Viren Jain and H. Sebastian Seung. Natural image denoising with convolutional networks. In *NIPS*, 2008. 1, 2

[23] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 4

[24] Wei-Sheng Lai, Jia-Bin Huang, Oliver Wang, Eli Shechtman, Ersin Yumer, and Ming-Hsuan Yang. Learning blind video temporal consistency. In *ECCV*, 2018. 4, 5, 6

[25] Marc Lebrun, Antoni Buades, and Jean-Michel Morel. A nonlocal bayesian image denoising algorithm. *SIAM Journal on Imaging Sciences*, 6(3):1665–1688, 2013. 2

[26] Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. Noise2noise: Learning image restoration without clean data. In *ICML*, 2018. 2

[27] Sifei Liu, Jin-shan Pan, and Ming-Hsuan Yang. Learning recursive filters for low-level vision via a hybrid neural network. In *ECCV*, 2016. 4

[28] Ziwei Liu, Lu Yuan, Xiaoou Tang, Matt Uyttendaele, and Jian Sun. Fast burst images denoising. *ACM TOG*, 33(6), 2014. 1, 2, 4

[29] Kin Gwn Lore, Adedotun Akintayo, and Soumik Sarkar. Llnet: A deep autoencoder approach to natural low-light image enhancement. *Pattern Recognition*, 61, 2017. 1, 2

[30] Artur Łoza, David R Bull, Paul R Hill, and Alin M Achim. Automatic contrast enhancement of low-light images based on local statistics of wavelet coefficients. *Digital Signal Processing*, 23(6):1856–1866, 2013. 2

[31] Feifan Lv, Feng Lu, Jianhua Wu, and Chongsoon Lim. Mbllen: Low-light image/video enhancement using cnns. In *BMVC*, 2018. 2

[32] Matteo Maggioni, Giacomo Boracchi, Alessandro Foi, and Karen Egiazarian. Video denoising, deblocking, and enhancement through separable 4-d nonlocal spatiotemporal transforms. *IEEE TIP*, 21(9):3952–3966, 2012. 2, 4, 5

[33] Julien Mairal, Francis Bach, Jean Ponce, Guillermo Sapiro, and Andrew Zisserman. Non-local sparse models for image restoration. In *ICCV*, 2009. 1, 2

[34] Henrik Malm, Magnus Oskarsson, Eric Warrant, Petrik Clarberg, Jon Hasselgren, and Calle Lejdfors. Adaptive enhancement and noise reduction in very low light-level video. In *ICCV*, 2007. 2

[35] Ben Mildenhall, Jonathan T Barron, Jiawen Chen, Dillon Sharlet, Ren Ng, and Robert Carroll. Burst denoising with kernel prediction networks. In *CVPR*, 2018. 1, 2, 3, 4, 5

[36] Seonhee Park, Soohwan Yu, Byeongho Moon, Seungyong Ko, and Joonki Paik. Low-light image enhancement using variational optimization-based retinex model. *IEEE Transactions on Consumer Electronics*, 63(2):178–184, 2017. 2

[37] Tobias Plötz and Stefan Roth. Benchmarking denoising algorithms with real photographs. *CVPR*, 2017. 2

[38] Javier Portilla, Vasily Strela, Martin J Wainwright, and Eero P Simoncelli. Image denoising using scale mixtures of gaussians in the wavelet domain. *IEEE TIP*, 12(11):1338–1351, 2003. 1, 2

[39] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 4

[40] 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. 1, 2

[41] Eli Schwartz, Raja Giryes, and Alex M Bronstein. DeepISP: Learning end-to-end image processing pipeline. *IEEE TIP*, 28(2):912–923, 2019. 1, 2

[42] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 4

[43] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *CVPR*, 2018. 5

[44] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 13(4):600–612, 2004. 5

[45] Bihan Wen, Yanjun Li, Luke Pfister, and Yoram Bresler. Joint adaptive sparsity and low-rankness on the fly: an online tensor reconstruction scheme for video denoising. In *ICCV*, 2017. 1

[46] Junyuan Xie, Linli Xu, and Enhong Chen. Image denoising and inpainting with deep neural networks. In *NIPS*, 2012. 1, 2

[47] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE TIP*, 2017. 1, 2