PDB Unet: A spatio temporal video Fixed Pattern Noise removal network

Arnaud Barral, Pablo Arias, Axel Davy

PDB Unet: A spatio temporal video Fixed Pattern Noise removal network

Arnaud Barral[1], Pablo Arias[2], Axel Davy[1]

[1]Université Paris-Saclay, CNRS, ENS Paris-Saclay, Centre Borelli, France
[2]Universitat Pompeu Fabra, Dept. of Engineering, Spain
{arnaud.barral,axel.davy}@ens-paris-saclay.fr pablo.arias@upf.edu

Abstract. In this paper we propose a novel video non uniformity correction algorithm based on a convolutional neural network. Fixed pattern noise (FPN) is a temporally coherent noise present on videos due to the non-uniformities of the sensors that can exhibit spatial correlation. This is a common problem with infrared video, degrading image quality and hampering subsequent applications. FPN removal has received less attention than other video restoration problems, and until very recently existing neural network approaches were limited to single frame processing. In this work we present a novel network architecture that takes several frames as input and outputs the estimated FPN. We also introduce parallel vertical & horizontal downsampling branches in the network that amplify the receptive field and help capture better the spatial correlation of the signal. We demonstrate the effectiveness of our method with extensive experiments on synthetic FPN comprising white, row and column Gaussian noise. Quantitative and qualitative comparisons against previous methods show that the proposed architecture can better leverage the spatial and the temporal information to remove the FPN, leading to state-of-the-art results.

Keywords: Fixed Pattern Noise (FPN) · Non uniformity Correction NUC · Denoising

1 Introduction

Noise in imaging devices can come from both external factors, for example shot noise, or internal factors, such as the nonuniform response of individual sensors. Fixed Pattern Noise (FPN) is a specific type of noise that remains almost temporally coherent and results from incorrect sensor calibration. Infrared (IR) videos are particularly susceptible to FPN due to the nature of IR sensors. For instance, the responses of microbolometer IR sensors are significantly influenced by their temperature, and calibrations based solely on temperature sensors may lack sufficient accuracy. Even if the FPN is considered fixed, it can actually vary over time, that is why most of the research focus on scene-based FPN estimation methods that operate online, continuously updating the FPN estimation to complement the initial calibration [12, 19, 26–28, 30, 44].

While several works were proposed in the literature for single image FPN denoising with deep neural networks [7, 10, 15, 16, 21, 39, 40], there has been few research into their application to video FPN denoising, with most existing approaches addressing the problem as a single image denoising task. Yet, image and video denoising are not equivalent. The temporal redundancy of the video signal can indeed help the restoration process especially when it comes to FPN since it is then easier to distinguish the fixed noise from the clean video if there is enough motion. In addition, the restored video needs to be temporally consistent, which cannot be achieved with a single frame network.

One of the key ingredient in deep video denoising is how to design the architecture to make full use of the neighboring frames. Existing video restoration methods can be divided into three categories: MISO (multiple input single output), also called sliding window methods, that take as input several neighboring frames and return a single frame usually the central one of the input sequence [35, 36]; MIMO (multiple input multiple output) that take as input several frames and return the estimated restored frames [22, 23], and recurrent networks [20]. MIMO methods compared to MISO and recurrent ones, have mainly two benefits: temporal consistency (except at stack transitions) [6] and computational cost. In the context of FPN denoising, a MIMO network would take several frames as inputs and outputs the estimated FPN, as done in residual learning [13, 43], and thus estimates the same noise several times.

FPN can adversely impact the performance of various video processing tasks, such as tracking and motion estimation. Although there are several methods for eliminating photon noise [2, 4, 5, 33, 42], they are generally not effective against FPN, as they rely on assumptions of spatial and temporal independence of the noise that are not verified in the case of FPN. That is why it is necessary to develop methods to remove FPN as shown in [1].

FPN is generally modelled as follows:

$$y(t) = g \otimes x(t) + o \tag{1}$$

where $\otimes$ denotes the element-wise product, $x(t)$ and $y(t)$ are $W \times H$ images, corresponding to the clean and noisy frames at time t, and g and o are the FPN pixel-wise gain and offset coefficients (also $W \times H$ images), modeling the multiplicative and additive components of the FPN. These components are typically modeled as white Gaussian noise. More realistic models also take into account the spatial correlation of noise, with constant noise along rows and columns [15, 18, 39]. Several works omit the multiplicative component g and focus solely on the additive FPN [1, 3, 20, 26, 30, 44] arguing that g can easily be removed with a first calibration [20].

Our contributions can be summarized as follows:

(i) We explore the application of MIMO networks to the problem of FPN removal. We adopt a residual setting [13, 43] in which the network estimates the FPN noise. We introduce a novel hybrid MIMO framework that takes as input several frames and outputs a single estimated FPN frame which is subtracted from all input frames, exploiting the fact that the FPN is constant in time. We

show the superiority of this hybrid MIMO approach to a naive approach in which the network estimates a multi-frame FPN. To the best of our knowledge it is the first end-to-end trainable network for FPN video denoising.

(ii) We introduce a new architecture with parallel downsampling branches that leverages the temporal correlation of the FPN and does not require alignment, and demonstrate the effectiveness of the proposed method using ablation studies.

(iii) Our proposed method achieves state-of-the-art results compared to other methods.

The next section reviews the related work. Our network is presented in Section 3. In Section 4 we present the results of our methods and compare them with the state of the art.

2 Related work

The task of removing fixed pattern noise (FPN), also referred to as non-uniformity correction (NUC), encompasses two main families of methods: reference-based and scene-based approaches. Reference-based methods mitigate noise based on pre-determined calibration parameters, which are typically derived from using a shutter or a black body at varying temperatures [9]. However, due to the temporal variability of FPN, these calibration parameters necessitate frequent updates. Consequently, the majority of research is focused towards scene-based FPN removal methods.

Scene-based techniques aim to estimate FPN from a single noisy sequence without relying on external information. This estimation is particularly difficult in scenarios where the sequence is static or shows minimal change, as it becomes difficult to distinguish the FPN from the actual scene content. In the case of a completely static scene, the problem is similiar to single-frame denoising, contaminated by spatially correlated noise. Recursive estimation algorithms often mistakenly assimilate temporally constant regions of the scene as part of the FPN and thus remove it.

Within scene-based methods, various sub-categories exist: constants statistics methods that leverages image statistics such as the mean and standard deviation of each pixels [12], and update correction coefficients recursively based on these statistics; temporal high-pass filter (THPF) methods [3, 26, 30, 41, 44], which estimate the noise in each frame by a high pass filter and the FPN as a running average of the estimated noise; and registration methods [11, 27], which utilize motion compensation to estimate the clean signal (which is assumed to be dynamic). Optimization-based methods [1, 19, 28, 31, 32, 37, 38] define an energy function with correction coefficients as variables, the FPN is estimated with online algorithms that apply at every frame one minimization step of the energy.

While most of the aforementioned methods use several images to estimate the FPN, they do not use several neighboring frames. At each iteration of these recursive algorithms, the estimated FPN is updated considering only a single noisy frame and the previous output. The only exception is [1] which proposed a recursive algorithm that uses several images for each estimation of the FPN but

requires images from several views or at least enough motion in the sequence to work correctly.

Recent works [7, 10, 15, 16, 18, 20, 39] have used learning-based methods such as convolutional neural networks (CNNs) to process single noisy images. Most of these methods treat FPN removal akin to single-image denoising, thus overlooking the valuable information in the temporal dimension. The authors of [20] remedy this problem by introducing a recurrent network but, as mentioned above for classical recurrent methods, only one image plus the previous output are taken into account when estimating the noise. The authors of [29] proposed a method to estimate the FPN simultaneously from several images however their network is not end-to-end, but is rather used as a regularizer. Moreover, this method requires very small movements in the video, of the order of a few pixels.

In our work, several neighboring frames are fed into the network which outputs the estimated noise. Using more frames leverages the temporal information and the specificity of the FPN, as it is the same noise on each frame.

Another important point to mention in the literature is the absence of any real consensus on the FPN model to be used, particularly for learning-based methods. Some methods focus on destriping [15, 18] and do not consider the gaussian part of the FPN. Other combine spatial correlated noise, stripes, and gaussian noise in the FPN [10, 20] but do not consider horizontal correlation like in [1] where the authors considered both gaussian and horizontal and vertical spatial correlation in the noise.

3 Proposed method

3.1 Baseline

Let $y_1, ..., y_N$ be N neighboring images that contain the same additive FPN b with variance σ_b:

$$y_n = x_n + b, \quad n = 1, \dots, N. \tag{2}$$

We want to estimate b from those frames. We first designed a baseline CNN, $\mathcal{J}$.

Before describing our architecture, we will present a simple MIMO Unet baseline. This networks takes as input $y_1, ..., y_N$ and returns each restored frames at once, so

$$(\hat{x}_1, ..., \hat{x}_N) = \mathcal{J}^{MIMO}(y_1, ..., y_N). \tag{3}$$

As each frame contains the same noise, this network estimates the same noise several times. While also being ineffective, this may impact the temporal consistency. A network that outputs the estimated common noise and therefore will denoise several frames at once does not have these problems.

Our baseline is a simple Unet that takes as input $y_1, ..., y_N$ and returns $\hat{b}$, the estimated FPN, that we will remove from the input images.

$$\hat{x}_n = y_n - \hat{b} = y_n - \mathcal{J}(y_1, ..., y_N), \quad n = 1, \dots, N. \tag{4}$$

This allows us to denoise all of the images from the input stack while returning only one output. It is a mix of a MISO and a MIMO framework. It combines

the advantages of the MIMO network of denoising several frames at once and forces the network to learn to estimate the common noise to each images and enforce good temporal consistency. Unless otherwise specified, the number of input frames is fixed at five.

We set the number of the depth dimension of the inner features of the Unet to 64 at each scale to limit the number of parameters.

The Unet consists of 4 scales. At each scale in the encoder and decoder we apply a residual block [14] four times with two 3×3 conv kernels and a ReLU activation between them (as opposed to [14], we do not apply the final ReLU after the addition with the skip connection), see Figure 1a. The downsampling is implemented using strided convolution and the upsampling transposed convolution. The bottleneck at the coarsest resolution (the input size divided by 16) consists of another residual block repeated four times. The input and output layers are simply 3×3 convolutions. In the skipped connection between the encoder and the decoder, the features are concatenated and fed to the transposed convolution, which will upsample them and divide the channel dimension by a factor of 2.

3.2 Proposed architecture

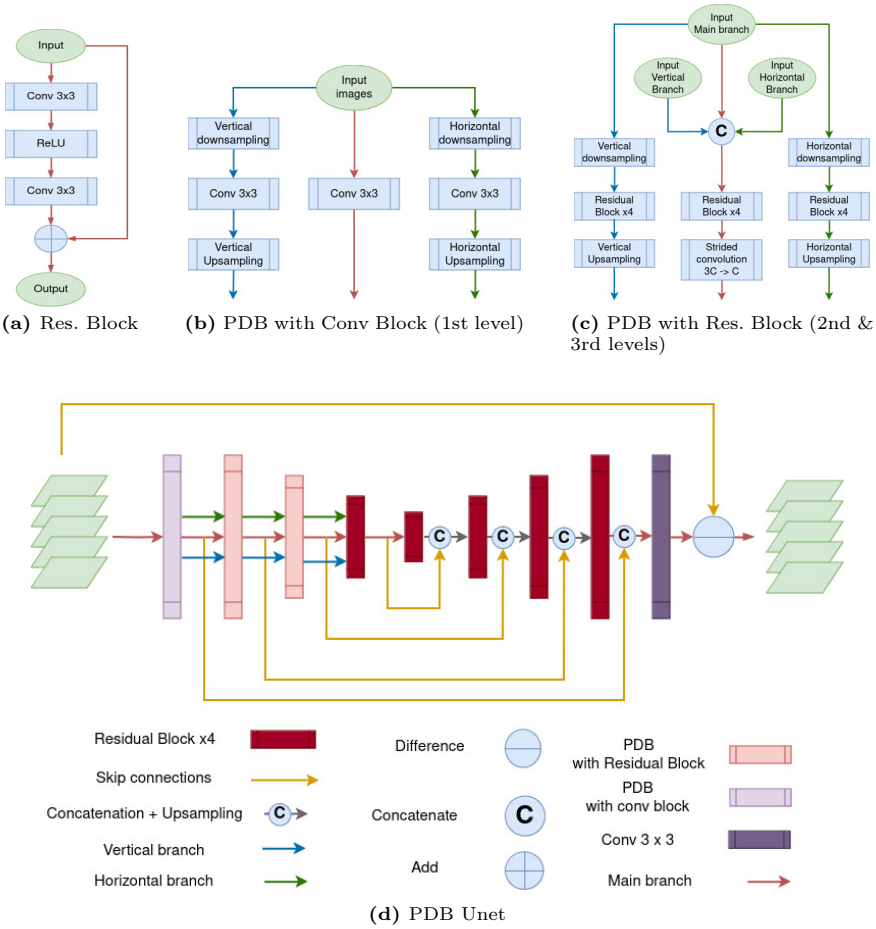
The proposed architecture is based on a modification of the baseline Unet by the addition of parallel vertical and horizontal downsampling branches (described in the next section) in the encoder path, which is called PDB Unet for parallel downsampling branches Unet. An overview can be seen in Figure 1d. The input is a stack of N noisy frames and the output is a single FPN frame $\hat{b}$, which is subtracted from the noisy inputs.

3.3 Parallel downsampling branches (PDB)

In this section we introduce parallel (vertical or horizontal) downsampling branches.

Given an $W_\ell \times H_\ell \times C$ input feature map φ_ℓ from the main branch at some level ℓ of the Unet we add two branches parallel to the main one: a vertically downsampled branch and a horizontally downsampled. In both cases the downsampling is by a factor of 2. See Figure 1c. In each branch, we downsample the image to size $W_\ell/2 \times H$ or $W_\ell \times H/2$, apply four residual blocks and then upsample back to the size of the main branch using bilinear upsampling. Then we apply a strided convolution to each branch for the downsampling (not shown in the diagram). If there are parallel branches at the previous layer $\ell - 1$ of the Unet as in Figure 1c, the outputs of the parallel branches at layer $\ell - 1$ are concatenated and become the input of the main branches at layer ℓ of the Unet. The output of the main branch at layer $\ell - 1$ is used as the input to the new parallel branches at layer ℓ. Figure 1b is the first PDB block which starts from the concatenated input images.

In the main branch, the three input branches are merged via concatenation along the channel dimension. The resulting $W_\ell \times H_\ell \times 3C_\ell$ feature map is processed by 4 residual blocks. Then we apply a residual block repeated four times



(a) Res. Block **(b)** PDB with Conv Block (1st level) **(c)** PDB with Res. Block (2nd & 3rd levels)

(d) PDB Unet

Fig. 1: Diagram of the proposed network built upon a Unet network. The network takes five frames as inputs and outputs a single FPN frame, which is subtracted from the input frames. See text for more details.

to the concatenated features and a strided convolution that will downsample the features and divide the channel dimension by a factor of 3.

The idea behind these branches is to take advantage of the temporal and spatial correlation of the stripes present in the FPN. In practise, the proposed architecture do improve the results but this is regardless of the noise type as long as the noise is fixed, see part 4.2.

3.4 Noise model

For the noise model, we will use the same as the one used in [1] where both structured component, row and column noise, and unstructured component, gaussian

FPN are used.

$$b(h,v) = b_w(h,v) + b_r(h) + b_c(v) \tag{5}$$

where (h,v) is a position on the image plane, $b_w(h,v) \sim \mathcal{N}(0,\sigma_{b_u})$ models the unstructured noise, $b_r(v) \sim \mathcal{N}(0,\sigma_{b_r})$ is constant along rows, and $b_c(h) \sim \mathcal{N}(0,\sigma_{b_c})$ is constant along columns. Note that we will consider the same variance for all type of noise.

3.5 Training

For training our networks we used the REDS training set [24]. The dataset was temporally downsampled by a factor of 3 and converted to grayscale. This dataset is often used for training video restoration networks. We used cropped patches of size $128\times128\times N$. We corrupted the clean videos with horizontal, vertical stripes and Gaussian noise as synthetic FPN as described above. All types of noise have the same standard deviation chosen uniformly between $[10.2, 25.5]$ in part 4.3, otherwise fixed to 10. The proposed architecture was trained for 10 epochs with 100000 iterations for each epochs, using the ADAM optimizer [17] with a minibatch size of 50. The learning rate is initialized at 0.0001 and decreases by a factor of 10 in the fifth and eighth epochs. We used the $L1$-loss as the loss function between the estimated stack and the ground truth:

$$\mathcal{L}(y_1,...,y_N) = \sum_{n=1}^{n=N} \|x_n - (y_n - \mathcal{J}(y_1,...,y_N))\|_1. \tag{6}$$

All experiments were done using the PyTorch package [25].

For the testing, we used the Set8 [36] and the FLIR [8] dataset. Set8 is often used for testing denoising networks. The FLIR dataset [8] was used in [20] by the authors to test their network and compare it with others, that is why we also decided to use it to compare our own method with [10, 15, 20]. We retrained our network using their noise model which is vertical stripe and gaussian noise as FPN.

4 Experiments

4.1 Parameters for the networks

Influence of the number of images on performance. We start by proving the effectiveness of using several frames to estimate the FPN. We used our baseline network, which is a Unet presented in Section 3.1. We changed the number N of input images to test its influence on the output. Quantitative results are shown in Table 1. The PSNR increased with the number of images used at the input of the network. If going from one image to five, leads to a gain of more than 2dB, the gain is less than 1 dB if the number of frames used is increased again by four to nine images.

Number of images	1	2	3	5	7	9
PSNR / SSIM	32.15/.91	33.16/.932	33.67/.94	34.54/.951	34.86/.955	35.26/.958

Table 1: Performance vs. number of images. Average PSNR and SSIM results on the Set8 dataset are reported. Several baseline models were trained with varying number of images at the input. Simulated additive FPN, spatially structured, both row and column, and spatially independent with a standard deviation of $\sigma = 10$ was added.

How to make use of neighboring frames. Our approach outputs a single noise frame b, thus it can be considered a MISO network. At the same time, due to the nature of FPN, the estimated FPN frame is used to produce the N denoised output frames, thus from this point of view it can also be considered a MIMO network.

In the following experiment we will compare our approach with a MISO and a truly MIMO approach. With a MISO network such as FastDVDNet [36] several input images are fed into the network to output a single image, usually the central one. In that case the denoised image is the following

$$\hat{x}_{N/2}^{MISO} = y_{N/2} - \hat{b}^{MISO} = y_{N/2} - \mathcal{J}^{MISO}(y_1, ..., y_N). \tag{7}$$

The corresponding loss function is the following

$$\mathcal{L}^{MISO}(y_1, ..., y_N) = \left\| x_{N/2} - \left(y_{N/2} - \mathcal{J}^{MISO}(y_1, ..., y_N)\right) \right\|_1. \tag{8}$$

Note that the loss is the only difference between the MISO approach and the proposed loss in Eq. (6).

For a MIMO network such as VRT [22] several input images are fed into the network to output the restored images. For our FPN estimation tasks, the network will output an FPN estimate for each frame:

$$\hat{x}_n^{MIMO} = y_n - \hat{b}_n^{MIMO} = y_n - \mathcal{J}^{MIMO}(y_1, ..., y_N)_n, \quad n = 1, \ldots, N. \tag{9}$$

The corresponding loss function in the MIMO framework is the following:

$$\mathcal{L}_{MIMO}(y_1, ..., y_N) = \sum_{n=1}^{n=N} \left\| x_n - \left(y_n - \mathcal{J}^{MIMO}(y_1, ..., y_N)_n\right) \right\|_1. \tag{10}$$

We trained our PDB Unet with a MISO, a MIMO and our framework, which is multiple input and one output, that is the estimated noise common to each frame. For each network, we used $N = 5$. Table 2 shows the average PSNR and SSIM on the Set 8. Each network was trained according to their framework described above. Our framework performs better than the MIMO framework and is similar to the MISO on. Moreover, compared with the MISO framework, ours is capable of denoising five images at once. Our framework estimates a single FPN frame for the stack of frames processed by the network. Thus within that stack, the output is temporally consistent.

Network type	Our framework	MIMO	MISO
PSNR / SSIM	36.00 / 0.966	35.86 / 0.966	35.98 / 0.966

Table 2: Influence of the use of the neighboring frames. Average PSNR and SSIM results on the Set8 dataset are reported. The PDB Unet network was tested with several framework, MISO, MIMO and our framework. Simulated additive FPN, spatially structured, both row and column, and spatially independent with a standard deviation of $\sigma = 10$ was added.

Downsampling factor Since our PDB architecture relies on downsampling, we tried several downsampling factors to see which one gives the best results. We test our PDB Unet with three downsampling factors, 2, 4 and 8. We also tried with a downsampling factor of 1 which means no downsampling. Quantitative results are shown in Table 3. Using a downsampling factor greater than 2 produces worse results. It would seem that the higher the factor, the lower the SSIM and the PSNR as reported in the table. Surprisingly having no downsampling in parallel branches is just as good as downsample with a factor of 2.

Downsampling factor	1	2	4	8
PSNR / SSIM	36.03 / 0.967	36.02 / 0.967	35.90 / 0.966	34.90 / 0.957

Table 3: Influence of the downsampling factor. Average PSNR and SSIM results on the Set8 dataset are reported. The PDB Unet network was tested with several downsampling factors. Simulated additive FPN, spatially structured, both row and column, and spatially independent with a standard deviation of $\sigma = 10$ was added.

4.2 Ablation study

Effectiveness of the parallel downsampling branches. To validate the effectiveness of the proposed parallel downsampling branches, we compare the PDB Unet with the baseline model. For a fair comparison, we trained the baseline model with more parameters by increasing the number of features in the coarser levels. With that we can compare our PDB Unet with a baseline that has more parameters to be sure that the effectiveness of our proposed architecture is real and not caused by an increase in the number of parameters. The quantitative results on the Set 8 is shown on Table 4. Baseline+ refers to our baseline model with more parameters by increasing the number of features in the coarser levels. PDB Unet refers to our baseline model with parallel downsampling branches. PDB Unet+ is the PDB Unet network where we added more parameters, by increasing the number of features in the coarser levels in the same way as for the Baseline+ model. While the Baseline+ allows a gain of 1dB over the baseline, the Unet PDB architecture manages to achieve a PSNR and SSIM 0.5dB higher than Baseline+ while having three times fewer parameters. This proves the effectiveness of the proposed architecture.

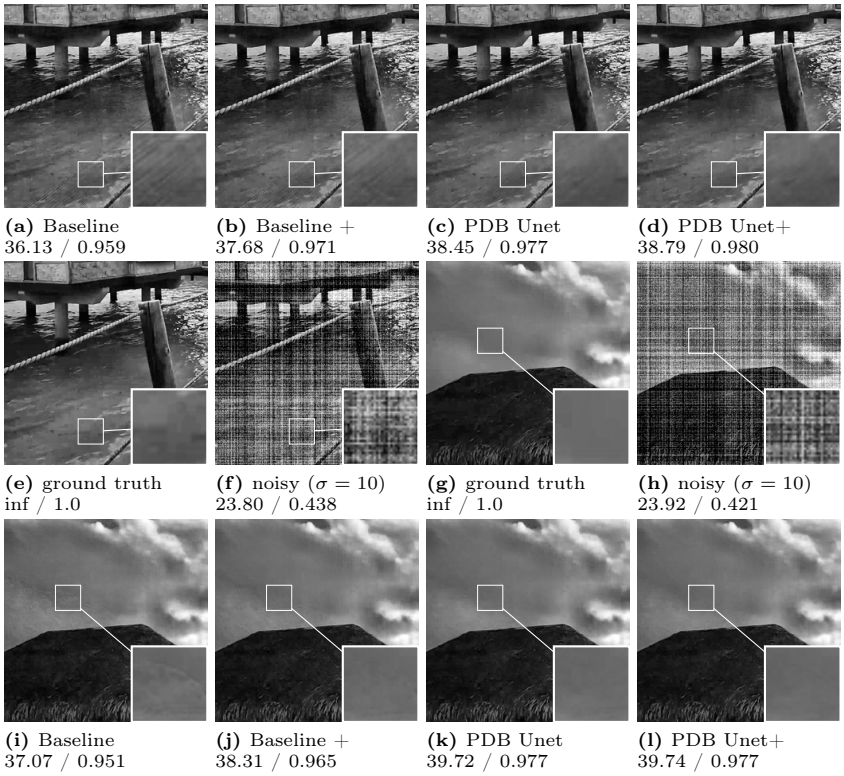


Fig. 2: Visual comparison of several crops of an image from the hypersmooth sequence of the Set8 dataset. Simulated additive, spatially structured and spatially independent noise with a standard deviation of $\sigma = 10$ was added to the frames. The Contrast Limited Adaptive Histogram Equalization has been applied to better account for artifacts. PSNR / SSIM on the cropped images are reported.

These quantitative differences can also be seen in Figure 2. The baseline and the baseline+ output images with visible artefacts like lines on the water. These artefacts are not visible on the images denoised by the PDB Unet and the PDB Unet+. The PDB Unet makes better use of the video's spatiotemporal information than the baseline.

Impact of the PDB on different kind of noise To understand the benefits of our proposed architecture, we performed ablation studies on different noise models. In Table 5, we removed the horizontal branch from the PDB Unet to obtain the PDB Unet vertical, and the vertical branch in PDB Unet horizontal. We compared them with our Baseline Unet that does not have parallel branches. All networks had the same training. At each iteration the noise model alternates between either gaussian FPN (i.e. without striped noise), gaussian plus vertical stripe FPN and finally gaussian plus horizontal stripe FPN. The goal of this

Network type	Baseline	PDB Unet	Baseline+	PDB Unet+
PSNR / SSIM	34.54 / 0.951	36.01 / 0.967	35.51 / 0.962	36.52 / 0.971
Parameters	2.2M	10.6M	33.3M	87.6M
Runtime (s) per frame	0.273	0.938	0.497	1.689
GMACs per frame	49	263	132	607

Table 4: Average PSNR and SSIM results on the Set8 dataset are reported. The baseline and the PDB Unet compared. Simulated additive FPN, spatially structured, both row and column, and spatially independent with a standard deviation of $\sigma_b = 10$ was added. MACs and runtime are computed on 480×640 images. Ptflops [34] was used to compute MACs.

experiment is to test if the PDB Unet horizontal removes better horizontal stripe FPN compared to the baseline and the PDB Unet vertical. Quantitative results on Table 5 show that is not the case. The networks with branches produce better results regardless of the FPN model compared to the baseline model with an equivalent number of parameters. It seems that the improvement seems to be due less to horizontal and vertical downsampling and more to parallel branches as it gives the network more channels to filter correlated noise.

In a second experiment we trained two MISO models one with and one without PDB on AGWN (additive white gaussian noise). Even if the PDB Unet achieves a better PSNR and SSIM compared to the baseline MISO showed in Table 6, the gap is much smaller compared to fixed noise, see Table 4. This seems to show that our proposed architecture works better on FPN.

Noise type \ Network	PDB Unet vertical	PDB Unet horizontal	Baseline
Gaussian FPN	38.73/0.972	38.71/0.972	37.86/0.968
Gaussian and vertical stripe FPN	37.20/0.969	37.19/0.965	36.49/0.964
Gaussian and horizontal stripe FPN	36.55/0.965	36.61/0.966	36.02/0.960
Parameters	5.6M	5.6M	7.0M

Table 5: Ablation study for the PDB. Average PSNR and SSIM results on the Set8 dataset are reported. Simulated additive FPN, spatially structured, both row and column, and spatially independent with a standard deviation of $\sigma_b = 10$ was added.

4.3 Comparison

In this part, we compare our PDB Unet with other methods from the literature. We choose to compare against RCNN-NUC [20] which is a recent recurrent network that achieved state-of-the-art results. As no official code has been published, we have used the results given in their article and tried to reproduce the same configuration they used for their experiments in order to make a comparison as fair as possible. We also reported the results of [10, 15] that were present in their paper. We trained our network according to their noise model, gaussian

Network	Baseline	PDB Unet
PSNR / SSIM	35.15 / 0.940	35.29 / 0.943
Parameters	10.6M	10.7M
Runtime (s) per frame	0.353	0.938
GMACs per frame	89	263

Table 6: Our proposed PDB Unet compared to the baseline on additive white gaussian noise. Average PSNR and SSIM results on the Set8 dataset are reported. No FPN was added to the images, only AWGN with a $\sigma = 10$ was added.

and vertical stripes FPN with the same standard deviation chosen uniformly between $[10.2, 25.5]$. We also compared with Multi-view FPNR [1], a state-of-the art recursive optimization-based method that uses several images to estimate the FPN. We used the official implementation provided by the authors and fixed the number of images to 16 as they do in their article. For comparison we used our PDB Unet and a bigger version, PDB Unet+ that takes 9 frames as input and has more parameters.

Quantitative results are reported in Table 7. Both of our networks, PDB Unet and PDB Unet+, outperform all single image FPNR networks up to more than 2dB for strong noise. RCNN-NUC [20] which is recurrent and so not single frame, produces better results for smaller noise level compared to our PDB Unet. The testing set being a sequence of 4000 frames, RCNN-NUC and Multi-view FPNR have a better receptive fields since they are recurrent compare to our network that only sees 5 frames at a time or 9 frames for PDB Unet+. The bigger version of PDB Unet achieves a higher PSNR for small noise by a small margin and by a high margin for stronger noise. Multi-view FPNR produces better results than single image networks and sometimes even RCNN-NUC. Visual results, see Figure 3, show that our methods, both PDB Unet and DPB Unet+, produce less artefacts compared to Multi-view FPNR even through our PDB Unet has a lower PSNR.

Network \ σ	10.2	15.3	20.4	25.5
DLS-NUC [15]	34.75	33.24	31.47	30.53
CNN-FPNR [10]	35.61	34.36	32.25	31.33
RCNN-NUC [20]	36.58	35.45	33.35	32.13
Multi-view FPNR [1]	36.50	34.95	33.91	33.43
PDB Unet (ours)	35.76	34.63	33.72	32.96
PDB Unet+ (ours)	36.82	35.79	34.94	34.22

Table 7: Average PSNR results on the FLIR dataset are reported. Simulated additive FPN, with vertical stripes and Gaussian FPN of varying std. dev. was added (no horizontal striped FPN). We used the same noise model as in [20] for comparison. Best results are shown in red, second best in blue.

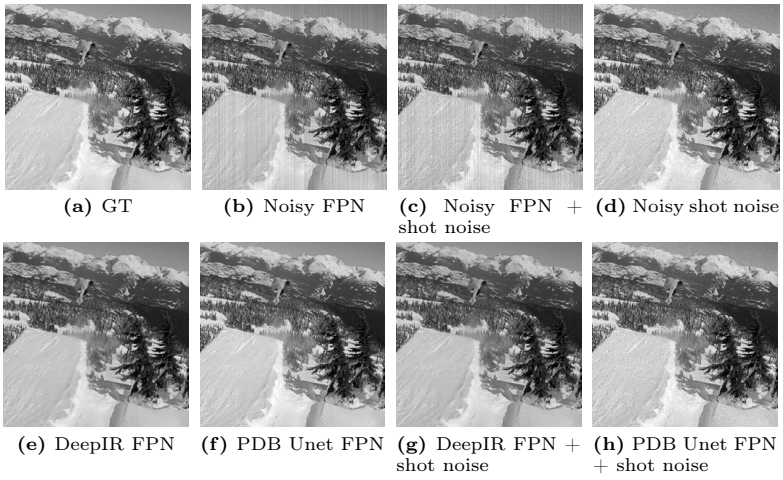
(a) GT **(b)** Noisy **(c)** PDB Unet **(d)** PDB Unet+ **(e)** Multi-view [1]

Fig. 3: Visual comparison on the FLIR dataset [8]. Simulated additive gaussian and vertical stripe FPN with a standard deviation of $\sigma = 25.5$ were added to the frames.

We also compared our method with the DeepIR method [29]. It is a state-of-the-art multi-image optimization-based method that removes FPN and shot noise and uses a neural network as a regularizer. This method does not use the same noise model as in RCNN [20], that is why we cannot add it to Table 7. We compared our method against DeepIR on their FPN model. We used the official implementation provided by the authors to test this method. Our network was trained on REDS with their noise model. For the testing sequences, we used the first images of each sequence of the Set8 Dataset and applied the simulation of the DeepIR method, which translates images by a few pixels, to have a comparison as fair as possible. We tested with the FPN model of DeepIR and with their full noise model that adds shot noise, which is not fixed, to the FPN. With FPN only, our method outperforms DeepIR 8. However DeepIR was made to remove FPN and shot noise simultaneously. In this configuration, our method that only removes fixed noise, cannot remove shot noise. We still achieve a higher PSNR but a lower SSIM which indicates that our method can better remove FPN compared to DeepIR. We manually selected parameters for this method and kept the best ones. We set the number of images used, the step size and the prior weight respectively to 20, 10^{-3} and 10^{-4}. Our method produces sharper results that contain more details compared to DeepIR results which are more blurred, see Figure 4.

Network \ Noise model	DeepIR FPN	DeepIR full noise model	Shot noise
DeepIR [29]	33.53 / 0.963	33.34 / 0.961	
PDB Unet (ours)	42.01 / 0.995	34.84 / 0.907	40.63 / 0.984

Table 8: Average PSNR results on the set8 dataset are reported, with simulated multiplicative vertical stripes FPN (no horizontal stripes FPN). We used the same noise model as in [29] for fair comparison. DeepIR full noise model also includes shot noise. In the column "Shot noise" we compared the output of the PDB Unet tested on the full noise model with the noisy images that only have shot noise to show that our method preserves varying noise and only removes FPN.



(a) GT **(b)** Noisy FPN **(c)** Noisy FPN + shot noise **(d)** Noisy shot noise

(e) DeepIR FPN **(f)** PDB Unet FPN **(g)** DeepIR FPN + shot noise **(h)** PDB Unet FPN + shot noise

Fig. 4: Visual comparison of an image from the snowboard sequence of the Set8 dataset. Simulated FPN according to DeepIR [29] model was added to the frames.

5 Conclusion

In this paper, we introduced a novel video non uniformity correction method based on a convolutional neural network. We showed the effectiveness of using several images to estimate the FPN. We also introduced parallel downsampling branches that leverages the spatial and temporal information. Extensive experiments and ablation studies show the efficiency of the proposed method.

We demonstrate the effectiveness of the proposed approach with synthetic FPN. Our method provides state-of-the-art results and outperforms previous single-image denoising works by a significant margin . Our evaluation is limited to synthetic noise, as there are currently no real standard FPN benchmarks. Future research should address the application of exiting methods to real FPN.

Acknowledgments This work was partly funded by AID-DGA (Agence de l'Innovation de Défense à la Direction Générale de l'Armement—Ministère des Armées), and was performed using HPC resources from GENCI-IDRIS (grants 2023-AD011011801R3, 2023-AD011012453R2, 2023-AD011012458R2) and from the "Mésocentre" computing center of CentraleSupélec and ENS Paris-Saclay supported by CNRS and Région Île-de-France (`http://mesocentre.centralesupelec.fr/`). Centre Borelli is also with Université Paris Cité, SSA and INSERM.

References

1. Barral, A., Arias, P., Davy, A.: Fixed pattern noise removal for multi-view single-sensor infrared camera. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 1669–1678 (January 2024)

2. Cao, Y., Wu, X., Qi, S., Liu, X., Wu, Z., Zuo, W.: Pseudo-isp: Learning pseudo in-camera signal processing pipeline from a color image denoiser (2021), https://arxiv.org/abs/2103.10234
3. Cheng, K., Zhou, H.X., Rong, S., Qin, H., Lai, R., Zhao, D., Zeng, Q.: Temporal high-pass filter nonuniformity correction algorithm based on guided filter for irfpa. p. 96752S (10 2015). https://doi.org/10.1117/12.2202781
4. Dewil, V., Anger, J., Davy, A., Ehret, T., Facciolo, G., Arias, P.: Self-supervised training for blind multi-frame video denoising. In: 2021 IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 2723–2733 (2021). https://doi.org/10.1109/WACV48630.2021.00277
5. Dewil, V., Barral, A., Facciolo, G., Arias, P.: Self-supervision versus synthetic datasets: which is the lesser evil in the context of video denoising? In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). pp. 4896–4906 (2022). https://doi.org/10.1109/CVPRW56347.2022.00537
6. Dewil, Valéry and Zheng, Zhe and Barral, Arnaud and Raad, Lara and Nicolas, Nao and Cassagne, Ioanis, and Morel, Jean-Michel and Facciolo, Gabriele and Galerne, Bruno and Arias, Pablo: Adapting mimo video restoration networks to low latency constraints. arXiv preprint arXiv:2408.12439 (2024)
7. Fayyaz, Z., Platnick, D., Fayyaz, H., Farsad, N.: Deep unfolding for iterative stripe noise removal. In: 2022 International Joint Conference on Neural Networks (IJCNN). pp. 1–7 (2022). https://doi.org/10.1109/IJCNN55064.2022.9892708
8. FLIR, T.: Free teledyne flir thermal dataset for algorithm training (2018)
9. Friedenberg, A., Goldblatt, I.: Nonuniformity two-point linear correction errors in infrared focal plane arrays. Optical Engineering **37**(4), 1251 – 1253 (1998). https://doi.org/10.1117/1.601890, https://doi.org/10.1117/1.601890
10. Guan, J., Lai, R., Xiong, A., Liu, Z., Gu, L.: Fixed pattern noise reduction for infrared images based on cascade residual attention cnn. Neurocomputing **377**, 301–313 (2020). https://doi.org/https://doi.org/10.1016/j.neucom.2019.10.054, https://www.sciencedirect.com/science/article/pii/S0925231219314341
11. Hardie, R., Hayat, M., Armstrong, E., Yasuda, B.: Scene-based nonuniformity correction with video sequences and registration. Applied optics **39**, 1241–50 (04 2000). https://doi.org/10.1364/AO.39.001241
12. Harris, J., Chiang, Y.M.: Nonuniformity correction of infrared image sequences using the constant-statistics constraint. IEEE Transactions on Image Processing **8**(8), 1148–1151 (1999). https://doi.org/10.1109/83.777098
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778 (2016). https://doi.org/10.1109/CVPR.2016.90
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
15. He, Z., Cao, Y., Dong, Y., Yang, J., Cao, Y., Tisse, C.L.: Single-image-based nonuniformity correction of uncooled long-wave infrared detectors: a deep-learning approach. Appl. Opt. **57**(18), D155–D164 (Jun 2018). https://doi.org/10.1364/AO.57.00D155, http://www.osapublishing.org/ao/abstract.cfm?URI=ao-57-18-D155
16. Juntao, G., Lai, R., Xiong, A.: Wavelet deep neural network for stripe noise removal. IEEE Access **PP**, 1–1 (04 2019). https://doi.org/10.1109/ACCESS.2019.2908720

17. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization (2017)
18. Kuang, X., Sui, X., Chen, Q., Gu, G.: Single infrared image stripe noise removal using deep convolutional networks. IEEE Photonics Journal **9**, 1–13 (08 2017). https://doi.org/10.1109/JPHOT.2017.2717948
19. Lai, R., Guan, J., Yang, Y., Xiong, A.: Spatiotemporal adaptive nonuniformity correction based on btv regularization. IEEE Access **7**, 753–762 (2019). https://doi.org/10.1109/ACCESS.2018.2885803
20. Li, F., Zhao, Y., Luo, H., Lv, C.: Spatio-temporal deep recurrent convolutional neural network for infrared focal plane arrays non-uniformity correction. Infrared Physics & Technology **140**, 105390 (2024). https://doi.org/https://doi.org/10.1016/j.infrared.2024.105390, https://www.sciencedirect.com/science/article/pii/S1350449524002743
21. Li, Y., Liu, N., Xu, J.: Infrared scene-based non-uniformity correction based on deep learning model. Optik **227**, 165899 (2021). https://doi.org/https://doi.org/10.1016/j.ijleo.2020.165899, https://www.sciencedirect.com/science/article/pii/S0030402620317162
22. Liang, J., Cao, J., Fan, Y., Zhang, K., Ranjan, R., Li, Y., Timofte, R., Van Gool, L.: Vrt: A video restoration transformer. arXiv preprint arXiv:2201.12288 (2022)
23. Liang, J., Fan, Y., Xiang, X., Ranjan, R., Ilg, E., Green, S., Cao, J., Zhang, K., Timofte, R., Van Gool, L.: Recurrent video restoration transformer with guided deformable attention. arXiv preprint arXiv:2206.02146 (2022)
24. Nah, S., Baik, S., Hong, S., Moon, G., Son, S., Timofte, R., Lee, K.M.: Ntire 2019 challenge on video deblurring and super-resolution: Dataset and study. In: CVPR Workshops (June 2019)
25. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems 32, pp. 8024–8035. Curran Associates, Inc. (2019), http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf
26. Qian, W., Chen, Q., Gu, G.: Space low-pass and temporal high-pass nonuniformity correction algorithm. Optical Review **17**, 24–29 (02 2010). https://doi.org/10.1007/s10043-010-0005-8
27. Ratliff, B., Hayat, M., Hardie, R.: An algebraic algorithm for nonuniformity correction in focal-plane arrays. Journal of the Optical Society of America. A, Optics, image science, and vision **19**, 1737–47 (10 2002). https://doi.org/10.1364/JOSAA.19.001737
28. Rossi, A., Diani, M., Corsini, G.: Bilateral filter-based adaptive nonuniformity correction for infrared focal-plane array systems. Optical Engineering - OPT ENG **49** (05 2010). https://doi.org/10.1117/1.3425660
29. Saragadam, V., Dave, A., Veeraraghavan, A., Baraniuk, R.G.: Thermal image processing via physics-inspired deep networks. In: 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW). pp. 4040–4048 (2021). https://doi.org/10.1109/ICCVW54120.2021.00451
30. Scribner, D.A., Sarkady, K.A., Caulfield, J.T., Kruer, M.R., Katz, G., Gridley, C.J., Herman, C.: Nonuniformity correction for staring IR focal plane arrays using scene-based techniques. In: Dereniak, E.L., Sampson, R.E. (eds.) Infrared Detectors and Focal Plane Arrays. vol. 1308, pp. 224 – 233. International Society for Optics and Photonics, SPIE (1990). https://doi.org/10.1117/12.21730, https://doi.org/10.1117/12.21730

31. Scribner, D.A., Sarkady, K.A., Kruer, M.R., Caulfield, J.T., Hunt, J.D., Herman, C.: Adaptive nonuniformity correction for IR focal-plane arrays using neural networks. In: Jayadev, T.S.J. (ed.) Infrared Sensors: Detectors, Electronics, and Signal Processing. vol. 1541, pp. 100 – 109. International Society for Optics and Photonics, SPIE (1991). https://doi.org/10.1117/12.49324, https://doi.org/10.1117/12.49324
32. Sheng-Hui, R., Hui-Xin, Z., Han-Lin, Q., Rui, L., Kun, Q.: Guided filter and adaptive learning rate based non-uniformity correction algorithm for infrared focal plane array. Infrared Physics & Technology **76**, 691–697 (2016). https://doi.org/https://doi.org/10.1016/j.infrared.2016.04.037, https://www.sciencedirect.com/science/article/pii/S1350449515300529
33. Sheth, D.Y., Mohan, S., Vincent, J., Manzorro, R., Crozier, P.A., Khapra, M.M., Simoncelli, E.P., Fernandez-Granda, C.: Unsupervised deep video denoising. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (October 2021)
34. Sovrasov, V.: ptflops: a flops counting tool for neural networks in pytorch framework (2018-2024), https://github.com/sovrasov/flops-counter.pytorch
35. Tassano, M., Delon, J., Veit, T.: Dvdnet: A fast network for deep video denoising. In: 2019 IEEE International Conference on Image Processing (ICIP). IEEE (Sep 2019). https://doi.org/10.1109/icip.2019.8803136, http://dx.doi.org/10.1109/ICIP.2019.8803136
36. Tassano, M., Delon, J., Veit, T.: Fastdvdnet: Towards real-time deep video denoising without flow estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2020)
37. Torres, S., Hayat, M.: Kalman filtering for adaptive nonuniformity correction in infrared focal-plane arrays. Journal of the Optical Society of America. A, Optics, image science, and vision **20**, 470–80 (04 2003). https://doi.org/10.1364/JOSAA.20.000470
38. Vera, E., Meza, P., Torres, S.: Total variation approach for adaptive nonuniformity correction in focal-plane arrays. Optics letters **36**, 172–4 (01 2011). https://doi.org/10.1364/OL.36.000172
39. Xiao, P., Guo, Y., Zhuang, P.: Removing stripe noise from infrared cloud images via deep convolutional networks. IEEE Photonics Journal **10**, 1–1 (07 2018). https://doi.org/10.1109/JPHOT.2018.2854303
40. Xu, K., Zhao, Y., Li, F., Xiang, W.: Single infrared image stripe removal via deep multi-scale dense connection convolutional neural network. Infrared Physics & Technology **121**, 104008 (2022). https://doi.org/https://doi.org/10.1016/j.infrared.2021.104008, https://www.sciencedirect.com/science/article/pii/S1350449521003807
41. Yuan, Y., Song, Q., Guo, X., Wang, Y.: A new temporal high-pass adaptive filter nonuniformity correction based on rolling guidance filter. p. 112 (01 2020). https://doi.org/10.1117/12.2539305
42. Zamir, S.W., Arora, A., Khan, S., Hayat, M., Khan, F.S., Yang, M.H., Shao, L.: Cycleisp: Real image restoration via improved data synthesis. In: CVPR (2020)
43. Zhang, K., Zuo, W., Chen, Y., Meng, D., Zhang, L.: Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. IEEE Transactions on Image Processing **26**(7), 3142–3155 (2017). https://doi.org/10.1109/TIP.2017.2662206
44. Zuo, C., Chen, Q., Gu, G., Qian, W.: New temporal high-pass filter nonuniformity correction based on bilateral filter. Optical Review **18**, 197–202 (03 2011). https://doi.org/10.1007/s10043-011-0042-y