Fixed Pattern Noise Removal For Multi-View Single-Sensor Infrared Camera

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 Information and Communication Technologies, Spain
{arnaud.barral,axel.davy}@ens-paris-saclay.fr pablo.arias@upf.edu
https://github.com/centreborelli/multiview-fpn

Abstract

Fixed pattern noise (FPN) is a temporally coherent noise present on videos due to the non-uniformities in the response of the imaging sensor. It is a common problem for infrared videos which degrades the quality of the observation and hinders subsequent applications. In this work we introduce a generalization of the FPN removal problem where the input data consists of several different sequences with the same FPN. This is motivated by infrared cameras that capture multiple views with a single sensor via a periodic motion pattern of a mirror or the camera itself, such as those used in surveillance. This multi-view setting allows for a much more accurate estimation of the FPN in comparison with the standard FPN removal problem from a single view. We propose a novel energy minimization approach for multi-view FPN removal, and two optimization algorithms that can be applied both in an off-line and online manner. In addition, we show that the proposed energy can be adapted to the problem of FPN removal from a single view with a rolling window approach, obtaining a significant improvement over the state of the art. We demonstrate the performance of the proposed method with synthetic data and real data from surveillance infrared cameras.

1. Introduction

Noise in imaging devices can be caused by factors external to the device, such as photonic noise, or internal, such as sensor calibration drift and electronic noise. FPN (for Fixed Pattern Noise) is a specific kind of noise that is almost temporally constant and is due to an incorrect sensor calibration. Infrared videos are particularly affected by FPN, due to the nature of IR sensors. In particular the responses of microbolometer IR sensors are heavily impacted by their temperature, and calibrating using temperature sensors might not be accurate enough [2]. In practice, the FPN changes slowly over time. For this reason, most methods for scene-based FPN estimation are online methods that update constantly the FPN estimation (e.g. [9, 13, 17–19, 21, 30]) to complement a first factory calibration.

FPN can negatively affect the performance of various video processing tasks, such as tracking and motion estimation. Often, methods used for these tasks assume that noise is not temporally correlated, and thus can fail due to the presence of FPN. In addition, videos with FPN often contain other types of noise, such as (temporally uncorrelated) photonic noise. Several methods exist for removing photonic noise, but their performance deteriorates in the presence of FPN. Figure 1 illustrates the impact of FPN. It shows on the first row five crops of different views produced each by averaging 150 consecutive frames in order to reduce temporal noise, but not the FPN. The latter is severely reduced on the second row by our method, using in total 16 views with the same sensor. Details that were invisible on the original images are revealed on the processed images.

The following linear model is widely for the FPN:

$$y(t) = a \otimes x(t) + b \tag{1}$$

where $\otimes$ is the element-wise product, $x(t)$ and $y(t)$ are the clean and noisy frames at time t, and a and b are respectively the FPN pixel-wise gain and offset coefficient, modelling respectively multiplicative and additive components of FPN. These components are modeled as white Gaussian noise, with more realistic models considering also structured noise, typically noise constant along rows and columns [10, 12, 28]. Several works omit the multiplicative component a and focus on additive FPN [5, 17, 21, 30].

In this work, we place ourselves in the case where multiple views are available from a single sensor. The views are considered simultaneous (or acquired within a period of time short in comparison with the rate of change of the FPN), so they are assumed to have the same FPN. The motivation for this multi-view FPN removal problem comes from cameras that acquire multiple views with a cyclic motion pattern (*e.g.* surveillance cameras). Such cameras take N frames per motion cycle, thus resulting in N views.

The main difference of the multi-view case over the

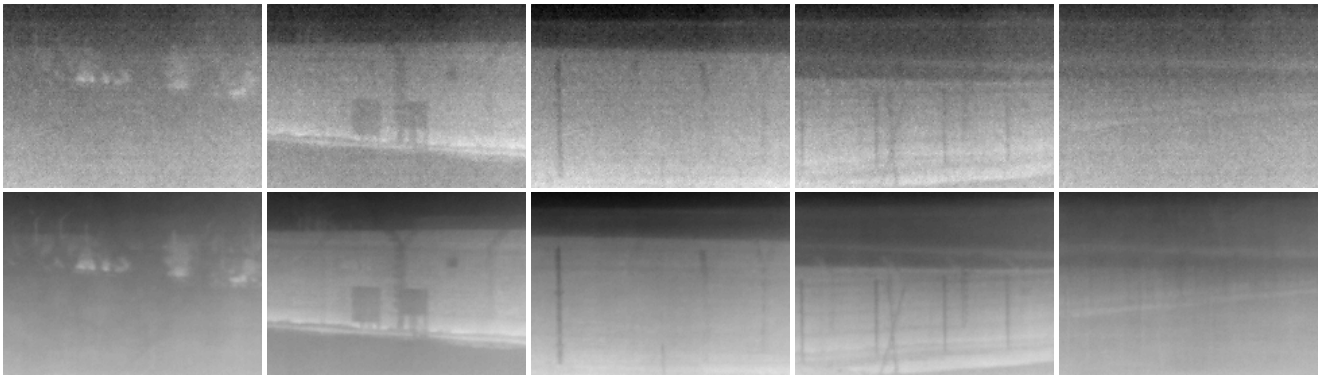


Figure 1. FPN estimation on real infrared data acquired with a multi-view single-sensor microbolometer camera. We display temporal averages to reduce temporal uncorrelated noise and unveil the FPN. Crops from 5 views (first row) and the results obtained with the proposed FPN removal method, estimated using 16 views (second row).

single-view case is that we have access to several views sharing the same FPN. This enables to extract the FPN by exploiting the redundancies between the noise realizations of each view, and the differences between the signal in the acquired view. This is very different to (single-view) scene-based FPN extraction methods where it can be challenging to separate FPN from signal in cases where there is little motion in the scene.

We propose an energy minimization method to tackle the multi-sequence FPN removal problem, with an energy consisting in data attachment terms and total variation (TV) spatial regularization. The proposed energy can be applied to N sequences or N single images, and it can be even applied to the standard single sequence setting by using a rolling window with the N latest frames.

To minimize the energy we discuss off-line methods which minimize a static energy, and on-line methods, which perform one (or a small amount) of iterations per frame. Off-line methods are suitable for calibration (i.e. estimate a FPN pattern which is assumed to be constant), whereas on-line methods are suitable for tracking the (slow) temporal drift of FPN. This results in a flexible framework which we can apply to different setups:

Multi-view offline FPN removal: several sequences (taken from different scenes) that have the same FPN. This is useful as a calibration step assuming that the FPN is constant in time. We study an extreme case of this setup, in which the used sequences have a single frame.

Multi-view online FPN removal: The input here is videos from different views with the same sensor assumed to be synchronized. In this setting, we use the online optimization so as to track temporal variations of the FPN.

Online single-sequence rolling window: The input is a single video, which is processed with an online optimization with a rolling window approach.

Our contributions can be summarized as follows:

(i) We introduce a generalized setting of the classical FPN removal problem, where we assume N input sequences with T frames, showing different realistic use cases for some particular cases of the problem.

(ii) A new variational FPNR algorithm that can be used in different settings, as explained above. We compare different optimization schemes, both on-line and off-line.

(iii) For the standard single-video FPN removal problem, the proposed rolling window method achieves state-of-the-art results.

The next section reviews the related work. In Section 3, we present the formalization. The results from our method in different setups are presented in Section 4.

2. Related work

The problem of removing the FPN is called FPN removal (FPNR) or non-uniformity correction (NUC). FPNR methods can be divided into two main families: reference-based and scene-based method. The reference-based methods remove noise according to fixed calibration parameters, estimated for example with a shutter, a black body at two different temperatures [6]. However, the FPN changes slightly over time, which requires regularly updating the calibration parameters. Because of this, most of the research is focused on the other scene-based FPN removal.

Scene-based methods seek to estimate the FPN from a single noisy sequence, without additional external information. Scene-based FPN removal is a challenging problem, specially if the sequence is static or changes slowly. In such cases it is difficult to separate the FPN from the scene content. In the extreme case of a constant scene, the problem becomes equivalent to single frame denoising, with the added difficulty of the structured row and column noise. Recursive estimation algorithms often remove temporally constant regions of the scene content as part of the

FPN. From scene-based methods, we can distinguish several subfamilies: those that work from image statistics [9] and update correction coefficient recursively according to these image statistics. Temporal high pass filters methods [5,17,21,29,30] apply a simple spatial denoiser to each frame. The residual contains most of the FPN, plus temporally varying noise and high frequency details from the clean signal. The main idea is then to extract the FPN as the temporal low-frequency content of this residual. Registration methods [8, 18] use the accurate motion estimation between frames to obtain a reasonable estimate of the true scene. Optimization-based methods [13, 19, 22, 23, 25, 26] define an energy with correction coefficients as variable that are minimized, generally with a gradient descent.

In this work we consider a generalization of the scene-based FPN problem, where we have at our disposal M synchronized sequences. To the best of our knowledge, no other previous work consider this problem before. Our energy-based formulation is most closely related to [13, 19, 22,23,25,26]. These are online approaches in which the energy being minimized considers only the current frame. In contrast, our energy involves multiple images (either from the different views, or from different frames in our single-view method). This enables a significantly improved precision and reduces the amount of images needed for precise estimation.

Recent approaches [7, 10, 12, 28] are learning-based methods that use convolutional neural networks (CNN) to process a single noisy image. These approaches treat the FPN removal as a single image denoising, thus disregarding the valuable information in the temporal dimension. Another problem of learning-based methods is poor generalization. CNNs are known to perform sub-optimally outside of their training domain. FPN removal is a low level problem heavily dependent on the specifics of the sensor used. For this type of problem it is difficult to obtain real training data. Instead CNNs are trained on synthetic datasets and might not generalize well to real raw imagery. Our approach can be applied to synthetic and real data with little parameter tuning, as opposed to deep networks which require retraining when the input data distribution changes.

3. Proposed method

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

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

Where $\{x_n\}_n$ is the set of N clean images and b is the additive FPN with standard deviation σ_b. The images are defined over a discrete rectangular domain $\Omega = \{1, 2, ..., W\} \times \{1, 2, ..., H\}$, and we denote by $\mathbb{R}^\Omega$ the spatial domain of the images.

Total variation (TV) is a regularization term widely used in image processing and computer vision [3]. Given y a noisy image with additive Gaussian noise the discrete version of TV can be defined as

$$\mathrm{TV}(y) = \sum_{p \in \Omega} |\nabla^+ y_p|, \tag{3}$$

where ∇^+ is the discrete forward difference gradient and if we note $p = (i, j)$, then we have

$$(\nabla^+ y)_{i,j} = (y_{i+1,j} - y_{i,j}, y_{i,j+1} - y_{i,j})$$

and $|\cdot|$ is the Euclidean norm in $\mathbb{R}^2$.

Total variation can be used as a regularization term for AWGN denoising with the following model

$$\min_{\hat{x}} \mathrm{TV}(\hat{x}) + \frac{\lambda}{2}\|y - \hat{x}\|_2^2 \tag{4}$$

where y is a noisy version of an image x that we would like to estimate and λ is a trade-off parameter between the TV regularization and the data attachment term. This energy is called the Rudin-Osher-Fatemi (ROF) model [20]. From a Bayesian perspective, the solution $\hat{x}$ can be interpreted as the Maximum A Posteriori (MAP) estimation problem with a prior induced by the TV regularizer (see e.g. [14, 16]).

A naive extension of the ROF model to our multi-image case can be written as follows

$$\min_{\hat{x}_1,\dots,\hat{x}_N} \sum_{n=1}^{N} \mathrm{TV}(\hat{x}_n) + \frac{\lambda}{2}\|\hat{x}_n - y_n\|_2^2. \tag{5}$$

Since we assume a common noise b, we have $\hat{x}_i = y_i - b$ and thus we re-parameterize the energy as

$$E(b) = \sum_{n=1}^{N} \mathrm{TV}(y_n - b) + \frac{\lambda_b}{2}\|b\|_2^2, \tag{6}$$

where $\lambda_b = N\lambda$.

This convex energy minimizing problem can be solved by several ways. In this paper, we will explore the use of the Adam optimizer [11] with automatic differentiation of PyTorch and the Chambolle-Pock algorithm [4].

3.1. Chambolle-Pock primal-dual algorithm

Chambolle-Pock [4] is a first-order primal-dual optimization algorithm for convex problems of the form:

$$\min_{b \in \mathbb{R}^\Omega} F(Kb) + G(b). \tag{7}$$

In cases in which F is non-differentiable, it can be beneficial to solve the following equivalent min-max problem:

$$\min_{b} \max_{\mathbf{q}, \|\mathbf{q}\|_\infty \leq 1} \langle Kb, \mathbf{q} \rangle + G(b) - F^*(\mathbf{q})$$

where F^* is the convex conjugate of F. The min-max problem requires an additional dual variable $\mathbf{q}$. Primal-dual

methods estimate $\mathbf{q}$ alongside b. For our energy E in (6), F corresponds to the sum of the total variation terms, G to the data attachment, and the linear operator $K : \mathbb{R}^{\Omega} \to \mathbb{R}^{2N\Omega}$ concatenates N copies of the discrete gradient of b, for each one of the N TV terms: $K(b) = (\nabla^+ b, ..., \nabla^+ b)$.

Algorithm 1 details the Chambolle-Pock algorithm [4], which it alternates minimization steps on the primal variable with (line 4) and maximization steps on the dual variable (line 3) with time-varying step sizes σ^m and τ^m, followed by a so-called extragradient step (line 6 with step θ). The remaining line in the main loop update the time-varying step sizes. The initial time step sizes σ^0 and τ^0 are parameters of the algorithm. In Algorithm 1, we have defined

$$\mathbf{g} = (\nabla^+ y_1, \nabla^+ y_2, ..., \nabla^+ y_N) \in \mathbb{R}^{2N\Omega}, \qquad (8)$$

the concatenation of the discrete gradients of the N input noisy images. Lastly, $K^* = \sum_{n=1}^{N} \text{div}^-$ is the adjoint of K, where div^- is a discrete divergence operator. We refer the reader to the supplementary material for more details.

3.2. Adam optimizer

The Adam optimizer was introduced in the context of deep learning [11], where it is widely used. It is an adaptive gradient descent method with momentum. We apply it to our primal problem. We use automatic differentiation for simplicity of implementation, but the gradient can be easily computed in closed-form.

3.3. Online optimization

So far we have considered N different images $y_1, ..., y_N$ affected by the same FPN. In this section we consider the case in which we have N synchronized sequences (which we denote $y_n(t)$, as a function of the frame index t) and a temporally varying FPN $b(t)$:

$$y_n(t) = x_n(t) + b(t), \quad n = 1, ..., N.$$

We estimate $b(t)$ by minimizing for each frame t the energy

$$E_t(b(t)) = \sum_{n=1}^{N} \text{TV}(y_n(t) - b(t)) + \frac{\lambda_b}{2}\|b(t)\|_2^2. \qquad (9)$$

This defines a temporal sequence of energies. We could minimize each one of them by running an iterative solver until convergence at each frame. But due to the temporal consistency in the sequences $y_n(t)$, the energy changes smoothly with time t. Thus, we perform a single iteration per frame hoping that the resulting iterates will track the temporal changes in the energy. In Algorithm 1, this implies that the iteration index m becomes the frame index t, thus $\mathbf{g}(t) = (\nabla^+ y_1(t), \ldots, \nabla^+ y_N(t))$ is now dependent on t (and changes in each iteration). In doing so we are not guaranteed to have an optimal solution for any frame, but we significantly reduce the computational cost. Similar online minimization algorithms have been proposed previously for FPN removal from a single sequence [19, 22, 26, 27].

Algorithm 1: Chambolle-Pock FPN Estimation

Input $y_1, ..., y_N$**:** noisy $H \times W$ images
Output b**:** estimated FPN

1 $b^0, \bar{b}^0, \mathbf{q}^0 \coloneqq$ `zeros` `# init with zeros`
2 **for** m **from** 1 **to** M **do**
3 $\quad \mathbf{q}_{n,p}^{m+1} = \dfrac{\mathbf{q}_{n,p}^m + \sigma^m K\bar{b}^m - \sigma^m \mathbf{g}_{n,p}}{\max\{|\mathbf{q}_{n,p}^m + \sigma^m K\bar{b}^m - \sigma^m \mathbf{g}_{n,p}|_2, 1\}}$
4 $\quad b^{m+1} = \frac{1}{1+\tau^m \lambda_b}(b^m - \tau^m K^\star \mathbf{q}^{m+1})$
5 $\quad \theta = \frac{1}{(1+2\gamma\tau^m)^{\frac{1}{2}}};\ \tau^{m+1} = \theta\tau^m;\ \sigma^{m+1} = \frac{\sigma^m}{\tau^m}$
6 $\quad \bar{b}^{m+1} = b^{m+1} + \theta(b^{m+1} - b^m)$

3.4. Use cases

We discuss 3 use-cases for our FPN removal framework.

Multi-view offline FPN estimation. In this setting, we assume that the FPN is constant, or has a negligible temporal drift. The FPN calibration process can be done once in a while, by taking N still images $y_1, ..., y_N$. The N images should be as different as possible, for example obtained by pointing the camera at different scenes, or by extracting N distant frames from a long video where the scene is not fixed. Since the images have the same FPN, which is estimated by minimizing energy E in (6).

Multi-view online FPN estimation. For this use-case, we assume the sensor captures images cycling over N different views. For example a mirror (or the camera itself) is moved in a pre-determined cyclic pattern. We thus have N synchronized sequences with the same FPN, and can estimate its slow temporal variation using online minimization described in Section 3.3 for the sequence (9) of energies E_t.

Single-view online FPN estimation. This is the standard FPN removal problem, in which we only have a single noisy video $y(t)$. In this case, we propose a rolling window approach that considers the previous N frames to define the following sequence of energies:

$$E_t^{\text{RW}}(b(t)) = \sum_{s=t-N+1}^{t} \text{TV}(y(s) - b(t)) + \frac{\lambda_b}{2}\|b(t)\|_2^2. \quad (10)$$

Note that if $N = 1$, then we end up a sequence of ROF problems as in Eq. (4).

4. Experiments

In this section we evaluate the proposed energy for FPN estimation in the three uses cases described before. We show results on synthetic and real data. We use PSNR for quantitative evaluation.

Figure 2. Online multi-view FPN removal on real microbolomer camera. Crops from 2 different views corresponding to averages (50 frames) to reduce temporal noise, and the corresponding results with our estimated FPN removed (adam) using 16 views (second row), bilateral-THPF (third row), SL-THPF (fourth row).

Noise modelling. Real FPN usually features both a structured component and an unstructured component [2]. We will consider the structured component as row and column noise as follows:

$$b(i,j) = b_w(i,j) + b_r(j) + b_c(i) \tag{11}$$

where (i,j) is a position on the image plane, $b_w(i,j) \sim \mathcal{N}(0, \sigma_{b_u})$ models the unstructured noise, $b_r(j) \sim \mathcal{N}(0, \sigma_{b_r})$ is constant along rows, and $b_c(i) \sim \mathcal{N}(0, \sigma_{b_c})$ is constant along columns. Note that we consider the same variance for the row and column. For the experiments, we set $\sigma_{b_c} = \sigma_{b_u} = 5$.

Discussion about parameters. The main parameter of the method is λ_b which controls the trade-off between data attachment and regularization in the energy. Thus it controls the denoising strength, and should be a function of the noise level (smaller λ_b will produce a stronger denoising). For the offline method, the number of input images N is also an important parameter. Larger values of N lead to better results (albeit at a higher computation cost: for both optimizers, the cost per-iteration scales linearly with N).

The step sizes of the optimization algorithm (η for Adam, σ_0, τ_0 for Chambolle-Pock) are additional parameters which determine the convergence speed and how close to the global minimum the result is. Larger step sizes lead to faster convergence, but if they are too large the algorithm to diverge [4]. Other parameters of Adam were left with their default values.

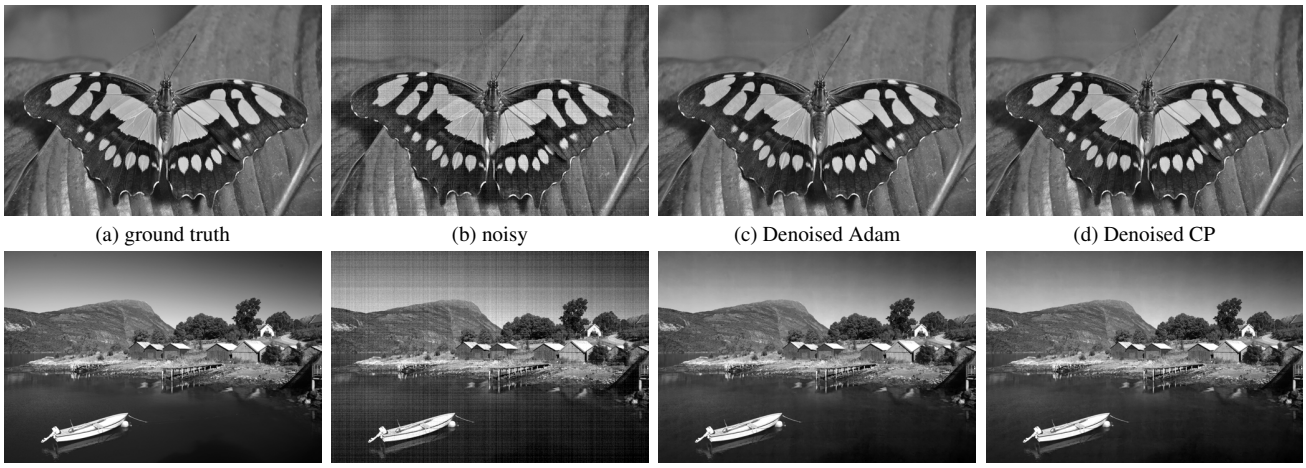
4.1. Multi-view offline FPN estimation

We start with the multi-view offline FPN estimation via the minimization of energy E in (6). This is a static energy and thus it fits the setting in which optimizers have been shown to converge. The input data is a set of N still images assumed to have the same FPN.

Synthetic visual results are shown in figure 3 and quantitative results in table 1. We use images from the DIV2K dataset [1], converted to grayscale and with additive FPN as modeled above. Both optimization algorithms produce good results. Using several images from different views help these methods separate the FPN from the signal (see figure 3), and the quality of the results increases with the number of available views N. The results produced by the Chambole-Pock algorithm are on par with Adam, which is reasonable since the energy is convex and has a unique minimizer. A visual inspection of the results shows that both methods produce similar results.

The optimal value of λ_b depends on the noise variance. All our experiments were done with fixed variances for the three noise components. Best results are achieved with $\lambda_b \approx 5 \cdot 10^{-2}$ for Adam and CP. For the step-size, values around $5 \cdot 10^{-1}$ seem to guarantee convergence relatively fast for Adam.

To validate the synthetic experiments, figure 1 shows the result of Adam on real data taken with a microbolometer camera with a cyclic motion pattern of $N = 16$ views captured with the same sensor. The original data (not shown) features photonic noise, electronical noise and FPN with small slow temporal variations. For each view, an image with reduced temporal noise is estimated by averaging 150 frames of the same view. The first row of the figure shows the averaged images for 5 views. After removing the temporally uncorrelated noise the FPN becomes apparent. We estimate it by minimizing our off-line multi-view energy using the $N = 16$ temporal averages as the $(y_n)_n$ in (6). The images with the FPN removed are shown in the second row.



(a) ground truth (b) noisy (c) Denoised Adam (d) Denoised CP

(e) ground truth (f) noisy (g) Denoised Adam (h) Denoised CP

Figure 3. Visual comparison of images from the DIV2K dataset. Simulated additive (no temporal independent noise), spatially structured and spatially independent noise with a standard deviation of $\sigma_{b_c} = \sigma_{b_u} = 5$ was added to the frames. (a)(e), (b)(f), (c)(g), (d)(h) are respectively the ground truth clean images, the noisy image with simulated FPN, the image denoised by our method with Adam, the image denoised by our method with CP. The results were obtained with $N = 16$ for both, $\lambda_b = 5 \times 10^{-2}$ and step size equals to 0.5 for Adam, $\lambda_b = 5 \times 10^{-2}$, $\tau = 0.02$, $\sigma = 0.5$, for Chambolle Pock.

Table 1. Multi-image FPN estimation. Average PSNR results obtained with simulated additive FPN (no temporal independent noise), spatially structured and spatially independent with a standard deviation of $\sigma_{b_c} = \sigma_{b_u} = 5$. The number of iteration is set to 500. Images are from DIV2K. N is set to 16 except specified otherwise.

	$N \rightarrow$	16	16	16	8	32
Method	$\lambda_b \rightarrow$	.05	.005	.0005	.005	.005
Noisy		29.4	29.4	29.4	29.4	29.4
Adam $\eta = .05$		38.9	39.0	39.0	37.6	40.2
Adam $\eta = .5$		40.5	38.9	38.1	35.2	43.1
Adam $\eta = 5$		40.4	38.1	34.9	35.0	42.3
CP $\sigma_0 = .1, \tau_0 = .1$		40.3	38.8	38.4	35.6	38.5
CP $\sigma_0 = .5, \tau_0 = .02$		40.5	40.2	40.2	37.8	43.9

4.2. Multi-view online FPN estimation

In this setting, we assume that the FPN might have slow variations and that we have frequent acquisitions from multiple views with the same sensor. Thus, we have N temporal sequences with the same slowly varying FPN.

For the synthetic experiments, we use N sequences from the REDS dataset [15] converted to grayscale to simulate the different views. Table 2 shows the performances of our method with Adam or CP against our own implementation of two baseline methods from the literature SL-THPF (Spatial Low-pass and Temporal High-pass Temporal High Pass Filtering) [17] and bilateral THPF (Bilateral Temporal High Pass Filtering) [30]. Note that these methods were developed for a single-view video. We adapt them to the multi-view case by running them on each of the N views, but keeping a single estimate of the FPN which is updated at each time t. At time t, we run the update rule of the method for each view and obtain N updates, which we average to update the FPN estimate. The proposed methods show a huge improvement with respect to the baselines, with Adam obtaining better results than CP.

For the experiment on real data, we use the moving microbolometer camera with $N = 16$ different views, each with 150 frames. We apply the multi-view online method (9) on the raw data which contains the temporally varying noise in addition to the FPN. In this case, the algorithm should ideally remove only the FPN leaving the temporally varying noise untouched. To evaluate visually the performance of the algorithm we show temporal averages of the outputs sequences on the last 50 frames (we use the first 100 frames as warm-up due to the slow convergence of the baseline methods). The first column of figure 2 shows the selected views, while the second, third and fourth columns present respectively the results with our method (using Adam), and the adapted bilateral THPF and SL-THPF. Looking at the results they seem to still contain a portion of FPN that is not removed, while our method gives significantly smoother results.

4.3. Single-view online FPN estimation

In this setting, a standard single view video is used to estimate the FPN (which can have a smooth temporal drift) using the rolling window energy (10). For the experiments we evaluate our results on several sequences of the REDS

Table 2. Average PSNR results of the last frames of several sequences of the REDS dataset, as described above, obtained with simulated additive FPN (no temporal independent noise), spatially structured and spatially independent with a standard deviation of $\sigma_{b_c} = \sigma_{b_u} = 5$. N is set to 16. For Adam the step size is equal to 0.5, for CP τ_0 and σ_0 are respectively equal to 0.02 and 0.5.

FPN	Noisy	SLTH*	bilateral*	Adam	CP
Constant	29.5	36.8	35.5	43.4	42.6
Varying	29.4	36.2	35.0	42.7	40.6

dataset converted to grayscale and temporally downsampled by a factor of 3. In addition we use a window of size $N = 16$.

The single-view sensor setting is the standard setting in which existing FPN removal algorithms operate, thus we can compare the performance of the proposed method against other approaches in the literature. We include results of our own implementations of SL-THPF [17] and bilateral THPF [30] (note that these are not the multi-view adaptations of the previous section). We also included results from a single image deep neural network SNRWDNN [7]. We used the authors' implementation and re-trained it with our noise model. For more details and additional results, please refer to the supplementary material. Whereas our algorithm uses N images for each FPN update (the frames in the rolling window), all existing online FPN removal methods, including [17] and [30], use a single frame per update.

Table 3 shows the PSNR obtained at the last frame for 5 test sequences. Our methods achieve several dBs over both THPF methods. Results from SNRWDNN are worse than the others as it is a single image network and only the PSNR of the last frame is reported. Figure 4 features plots of the evolution of the PSNR with the frame index of the compared methods for one sequence of REDS. The first plot considers the case of constant FPN. In the second plot the FPN varies slowly, as a linear interpolation between two realizations of FPN at frames $t = 1$ and $t = 150$. In both cases, the proposed rolling window method with Adam and CP has significantly improved performance compared to its competitors (with CP achieving the best results). In addition the convergence is faster, surpassing the single frame CNN SNRWDNN after 10 to 20 frames, and reaching a steady state after around 50 frames. Instead the competitors do not seem fully converged after 150 frames on the first plot. Plus, on the second plot, the competitors converge to lower performance, indicating troubles to estimate the changing FPN. On the first plot, we include variants of our algorithm with several values of N. As before, there is a small performance increase with higher values of N.

The difference of PSNR can be seen in the output images, shown in figure 5. Most of high frequencies of the FPN is removed by all methods, but the images denoised by the SL-THPF and the bilateral THPF algorithm still show considerable low frequency noise.

The third plot in figure 4 corresponds to the more realistic case where in addition to the FPN, the sequence is contaminated with temporally uncorrelated additive white Gaussian noise. We first remove the FPN using different FPN removal methods, and then remove the temporally uncorrelated AWGN with a video denoising network (FastDVDNet [24]) trained for blind AWGN denoising. The baseline curve labeled "FastDVDNet" shows the performance obtained by applying the AWGN video denoiser without removing the FPN. We can see that when removing the FPN before FastDVDNet is applied, the denoising performance is greatly improved. This plot shows that the proposed FPN removal methods can work properly in presence of temporal noise, which makes them a suitable pre-processing for further video restoration tasks.

Table 3. PSNR results of the last frame of sequences from the REDS dataset obtained with simulated additive FPN (no temporal independent noise), spatially structured and spatially independent with a standard deviation of $\sigma_{b_c} = \sigma_{b_u} = 5$. N is set to 16. For Adam the step size is equal to 0.5, for CP τ_0 and σ_0 are respectively equal to 0.02 and 0.5.

Method \ Seq	001	002	003	004	005
Noisy	29.5	29.4	29.5	29.4	29.3
SL-THPF	36.1	36.2	35.7	36.5	36.4
Bilateral THPF	35.2	35.2	34.7	35.2	35.2
SNRWDNN	33.8	34.1	33.2	34.7	33.7
Ours (Adam)	38.9	39.7	38.2	42.0	40.2
Ours (CP)	40.3	41.0	39.5	42.3	40.9

5. Conclusion

In this paper, introduced a multi-view generalization of the FPN estimation problem and proposed a novel energy minimization approach to remove FPN that exploits different views having the same FPN. The proposed energy is convex and we compare two optimization methods Adam and the algorithm Chambolle-Pock, which is better suited for convex energies with non-differentiable regularization terms (such as TV).

The method require little parameter tuning, and thus can be easily adapted to different kinds of data. We demonstrate the effectiveness of the proposed approach on synthetic and real data from microbolometer IR multi-view cameras for the three use cases: an off-line multi-view calibration for constant FPN, an online multi-view FPN removal, and also a single-view FPN removal where the proposed approach

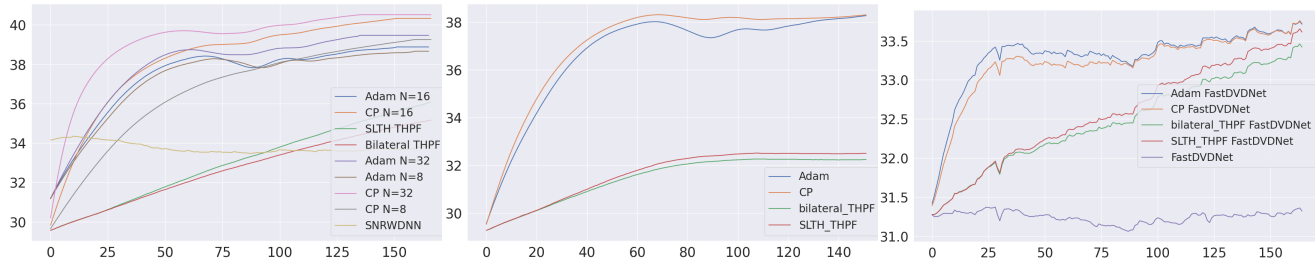
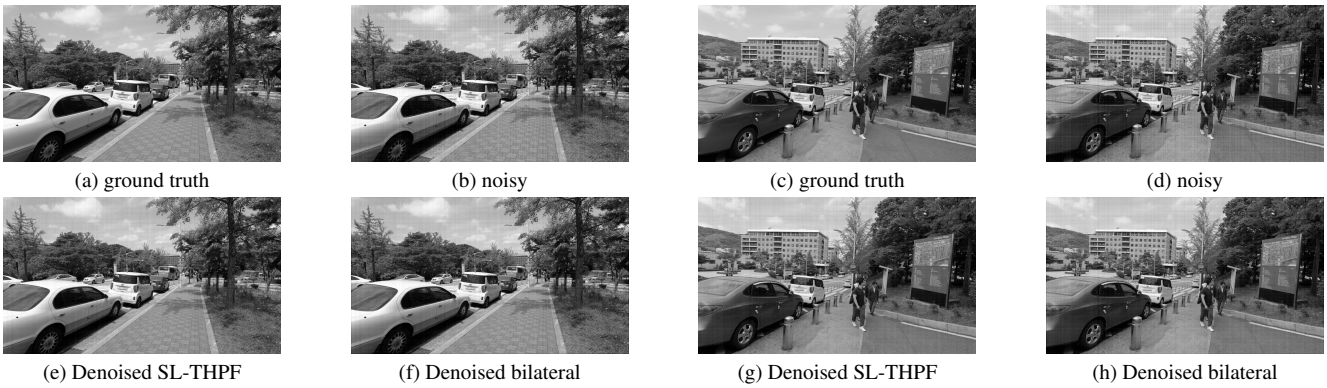


Figure 4. Single-view online FPN estimation. Graphs showing the evolution of the PSNR (vertical axis) per frame (horiz. axis) for the sequence 001 of the REDS dataset. We compare the results of two baseline methods from the literature of video FPN removal, bilateral THPF [30] and SL-THPF [17] with our rolling window approach (10) with $N = 16$ (unless specified otherwise) minimized with online versions of Adam and Chambole-Pock (CP). We also include the results of a single frame FPN removal network SNRWDNN applied on every frame. For Adam the step size is equal to 0.5 and $\lambda_b = 10^{-2}$. For CP, τ_0 and σ_0 are respectively equal to 0.02 and 0.5 and $\lambda_b = 10^{-2}$. *Left:* Constant FPN has been added to the sequence. We show results obtained with different number of frames N in the rollowing window. *Center:* The FPN changes slowly over time, linearly interpolating two realizations of noise at $t = 0$ and $t = 150$. Both temporal high pass filtering methods (THPF) fail to track the change in FPN. *Right:* In addition to the FPN (which is constant) we add additive white Gaussian noise (temporally uncorrelated) of $\sigma = 10$. To denoise this sequence, we first apply FPN removal methods and then apply a FastDVDNet network [24] trained grayscale images with AWGN. We report the PSNR obtained by FastDVDNet (without prior FPN removal) to highlight the need of FPN removal.



(a) ground truth (b) noisy (c) ground truth (d) noisy

(e) Denoised SL-THPF (f) Denoised bilateral (g) Denoised SL-THPF (h) Denoised bilateral

(i) Denoised Adam (j) Denoised CP (k) Denoised Adam (l) Denoised CP

Figure 5. Visual comparison on two sequences of the REDS dataset, contaminated with simulated additive FPN with two components: spatially structured and spatially independent with a standard deviations of $\sigma_{b_c} = \sigma_{b_u} = 5$ (no temporal independent noise was added). (a), (b), (e), (f), (i), (j) are respectively the ground truth clean image, the noisy image, the image denoised by SL-THPF, the image denoised by bilateral THPF, the image denoised by our method with Adam, the image denoised by our method with CP. Images (c), (d), (g), (h), (k), (l) are the same for another sequence from the REDS dataset.

outperforms previous works by a significant margin.

Acknowledgments This work was partly funded by AID-DGA (l'Agence de l'Innovation de Defense a la Direction Generale de l'Armement—Minitsere des Armees), 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] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017. 5

[2] Musaed Alhussein, Syed Irtaza Haider, et al. Simulation and analysis of uncooled microbolometer for serial readout architecture. *Journal of Sensors*, 2016, 2016. 1, 5

[3] Vicent Caselles, Antonin Chambolle, Matteo Novaga, et al. Total variation in imaging. *Handbook of mathematical methods in imaging*, 1(2):3, 2015. 3

[4] Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40, 05 2011. 3, 4, 5

[5] Kuanhong Cheng, Hui-Xin Zhou, Shenghui Rong, Hanlin Qin, Rui Lai, Dong Zhao, and Qingjie Zeng. Temporal high-pass filter nonuniformity correction algorithm based on guided filter for irfpa. page 96752S, 10 2015. 1, 3

[6] Abraham Friedenberg and Isaac Goldblatt. Nonuniformity two-point linear correction errors in infrared focal plane arrays. *Optical Engineering*, 37(4):1251 – 1253, 1998. 2

[7] Juntao Guan, Rui Lai, and Ai Xiong. Wavelet deep neural network for stripe noise removal. *IEEE Access*, 7:44544–44554, 2019. 3, 7

[8] Russell Hardie, Majeed Hayat, Esther Armstrong, and Brian Yasuda. Scene-based nonuniformity correction with video sequences and registration. *Applied optics*, 39:1241–50, 04 2000. 3

[9] J.G. Harris and Yu-Ming Chiang. Nonuniformity correction of infrared image sequences using the constant-statistics constraint. *IEEE Transactions on Image Processing*, 8(8):1148–1151, 1999. 1, 3

[10] Zewei He, Yanpeng Cao, Yafei Dong, Jiangxin Yang, Yanlong Cao, and Christel-Löic Tisse. Single-image-based nonuniformity correction of uncooled long-wave infrared detectors: a deep-learning approach. *Appl. Opt.*, 57(18):D155–D164, Jun 2018. 1, 3

[11] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. 3, 4

[12] Xiaodong Kuang, Xiubao Sui, Qian Chen, and Guohua Gu. Single infrared image stripe noise removal using deep convolutional networks. *IEEE Photonics Journal*, 9:1–13, 08 2017. 1, 3

[13] Rui Lai, Juntao Guan, Yintang Yang, and Ai Xiong. Spatiotemporal adaptive nonuniformity correction based on btv regularization. *IEEE Access*, 7:753–762, 2019. 1, 3

[14] Cécile Louchet and Lionel Moisan. Posterior expectation of the total variation model: Properties and experiments. *SIAM Journal on Imaging Sciences*, 6(4):2640–2684, 2013. 3

[15] Seungjun Nah, Sungyong Baik, Seokil Hong, Gyeongsik Moon, Sanghyun Son, Radu Timofte, and Kyoung Mu Lee. Ntire 2019 challenge on video deblurring and super-resolution: Dataset and study. In *CVPR Workshops*, June 2019. 6

[16] Mila Nikolova. Model distortions in bayesian map reconstruction. *Inverse Problems and Imaging*, 1(2):399, 2007. 3

[17] Weixian Qian, Qian Chen, and Guohua Gu. Space low-pass and temporal high-pass nonuniformity correction algorithm. *Optical Review*, 17:24–29, 02 2010. 1, 3, 6, 7, 8

[18] Bradley Ratliff, Majeed Hayat, and Russell Hardie. 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. 1, 3

[19] Alessandro Rossi, M. Diani, and Giovanni Corsini. Bilateral filter-based adaptive nonuniformity correction for infrared focal-plane array systems. *Optical Engineering - OPT ENG*, 49, 05 2010. 1, 3, 4

[20] 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. 3

[21] Dean A. Scribner, Kenneth A. Sarkady, John T. Caulfield, Melvin R. Kruer, G. Katz, C. J. Gridley, and Charles Herman. Nonuniformity correction for staring IR focal plane arrays using scene-based techniques. In Eustace L. Dereniak and Robert E. Sampson, editors, *Infrared Detectors and Focal Plane Arrays*, volume 1308, pages 224 – 233. International Society for Optics and Photonics, SPIE, 1990. 1, 3

[22] Dean A. Scribner, Kenneth A. Sarkady, Melvin R. Kruer, John T. Caulfield, J. D. Hunt, and Charles Herman. Adaptive nonuniformity correction for IR focal-plane arrays using neural networks. In T. S. Jay Jayadev, editor, *Infrared Sensors: Detectors, Electronics, and Signal Processing*, volume 1541, pages 100 – 109. International Society for Optics and Photonics, SPIE, 1991. 3, 4

[23] Rong Sheng-Hui, Zhou Hui-Xin, Qin Han-Lin, Lai Rui, and Qian Kun. Guided filter and adaptive learning rate based non-uniformity correction algorithm for infrared focal plane array. *Infrared Physics & Technology*, 76:691–697, 2016. 3

[24] 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. 7, 8

[25] Sergio Torres and Majeed Hayat. 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. 3

[26] Esteban Vera, Pablo Meza, and Sergio Torres. Total variation approach for adaptive nonuniformity correction in focal-plane arrays. *Optics letters*, 36:172–4, 01 2011. 3, 4

[27] Esteban Vera and Sergio Torres. Fast adaptive nonuniformity correction for infrared focal-plane array detectors. *EURASIP J. Adv. Sig. Proc.*, 2005:1994–2004, 08 2005. 4

[28] Pengfei Xiao, Yecai Guo, and Peixian Zhuang. Removing stripe noise from infrared cloud images via deep convolutional networks. *IEEE Photonics Journal*, 10:1–1, 07 2018. 1, 3

[29] Yuan Yuan, Qiong Song, Xin Guo, and Yuehuan Wang. A new temporal high-pass adaptive filter nonuniformity correction based on rolling guidance filter. page 112, 01 2020. 3

[30] Chao Zuo, Qian Chen, Guohua Gu, and Weixian Qian. New temporal high-pass filter nonuniformity correction based on bilateral filter. *Optical Review*, 18:197–202, 03 2011. 1, 3, 6, 7, 8