



HAL
open science

Linear Combinations of Patches are Unreasonably Effective for Single-Image Denoising

Sébastien Herbreteau, Charles Kervrann

► **To cite this version:**

Sébastien Herbreteau, Charles Kervrann. Linear Combinations of Patches are Unreasonably Effective for Single-Image Denoising. 2024. hal-03894346v2

HAL Id: hal-03894346

<https://hal.science/hal-03894346v2>

Preprint submitted on 15 Nov 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Linear Combinations of Patches are Unreasonably Effective for Single-Image Denoising

Sébastien Herbreteau and Charles Kervrann, *Member, IEEE*,

Abstract—In the past decade, deep neural networks have revolutionized image denoising in achieving significant accuracy improvements by learning on datasets composed of noisy/clean image pairs. However, this strategy is extremely dependent on training data quality, which is a well-established weakness. To alleviate the requirement to learn image priors externally, single-image (*a.k.a.*, self-supervised or zero-shot) methods perform denoising solely based on the analysis of the input noisy image without external dictionary or training dataset. This work investigates the effectiveness of linear combinations of patches for denoising under this constraint. Although conceptually very simple, we show that linear combinations of patches are enough to achieve state-of-the-art performance. The proposed parametric approach relies on quadratic risk approximation via multiple pilot images to guide the estimation of the combination weights. Experiments on images corrupted artificially with Gaussian noise as well as on real-world noisy images demonstrate that our method is on par with the very best single-image denoisers, outperforming the recent neural network-based techniques, while being much faster and fully interpretable.

Index Terms—Patch-based single-image denoising, non-local methods, Stein’s unbiased risk estimation, self-supervised learning, statistical aggregation

I. INTRODUCTION

AMONG the inverse problems in imaging, denoising is without doubt the most extensively studied [1]. In its simplest formulation, an image $x \in \mathbb{R}^d$ is perturbed by an additive white Gaussian noise (AWGN) w of variance σ^2 . Denoising then consists in processing the resulting noisy image $y = x + w$ in order to remove the noise component w and recovering the original signal x .

Over the years, a rich variety of strategies, tools and theories have emerged to address this issue at the intersection of statistics, signal processing, optimization and functional analysis. But this field has been recently immensely influenced by the development of machine learning techniques and deep neural networks. Viewing denoising as a simple regression problem, this task ultimately amounts to learning a network to match the corrupted image to its source. In practice, a training phase is necessary beforehand, during which the network is optimized by stochastic gradient descent on an external dataset consisting of clean/noisy image pairs. The power of deep-learning lies in its tremendous generalization capabilities allowing it to be just as effective beyond its training set. This approach has revolutionized denoising, as well as many inverse problems in computer vision. Numerous supervised neural networks

Sébastien Herbreteau and Charles Kervrann are with Centre Inria de l’Université de Rennes and Inserm U1143, Institut Curie, PSL University, France (e-mail: sebastien.herbreteau@inria.fr and charles.kervrann@inria.fr).

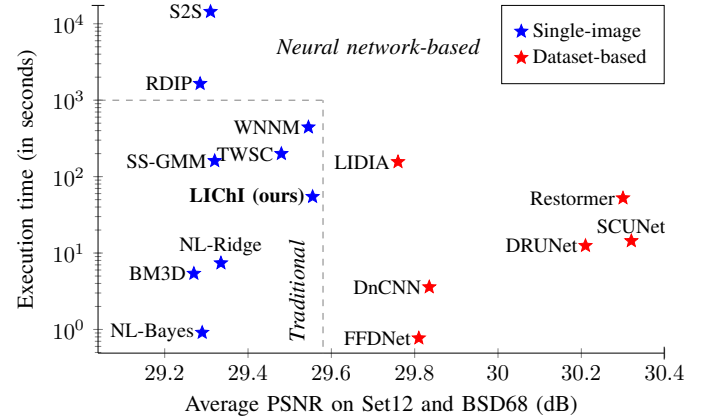


Fig. 1. The execution time on CPU for an image of size 512×512 v.s. the average PSNR results on Set12 and BSD68 [19] for synthetic Gaussian noise with $\sigma = 25$ of the most effective popular methods [2]–[7], [20]–[27]. These results are calculated based on Table I and subsection V-C.

have been proposed since then for image denoising [2]–[16], leading to state-of-the-art performance. Nowadays, the very best methods exploit the transformer architecture that is able to extract non-local features [6]–[8], [15]–[18].

However, these supervised methods, in addition to being cumbersome due to the computationally demanding optimization phase, suffer from their high sensitivity to the quality of the training set. The latter must indeed provide diverse, abundant and representative examples of images; otherwise, mediocre or even totally aberrant results can be obtained afterwards. This makes their use impossible in some cases, especially when noise-free images are missing (although training on datasets composed of noisy/noisy image pairs was studied in [28]–[34]). To address this issue, self-supervised learning - a machine learning technique in which only the input noisy image is used for training - with deep neural networks was investigated as an alternative strategy [26], [27], [33], [35]–[37] but their performance is still limited when compared to their conventional counterparts [20]–[25], [38]–[42].

In dataset-free image denoising, BM3D [22] remains a reference method and is still competitive today even if it was developed fifteen years ago. Leveraging the non-local strategy, its mechanism relies on processing collaboratively groups of similar noisy patches across the image, assuming a locally sparse representation in a transform domain. Since then, a lot of methods based on patch grouping were developed [20], [21], [23], [25], [38], [39]. In a recent paper [20], we proposed a unified view of several non-local two-step methods, BM3D [22] being at the forefront. We showed how these

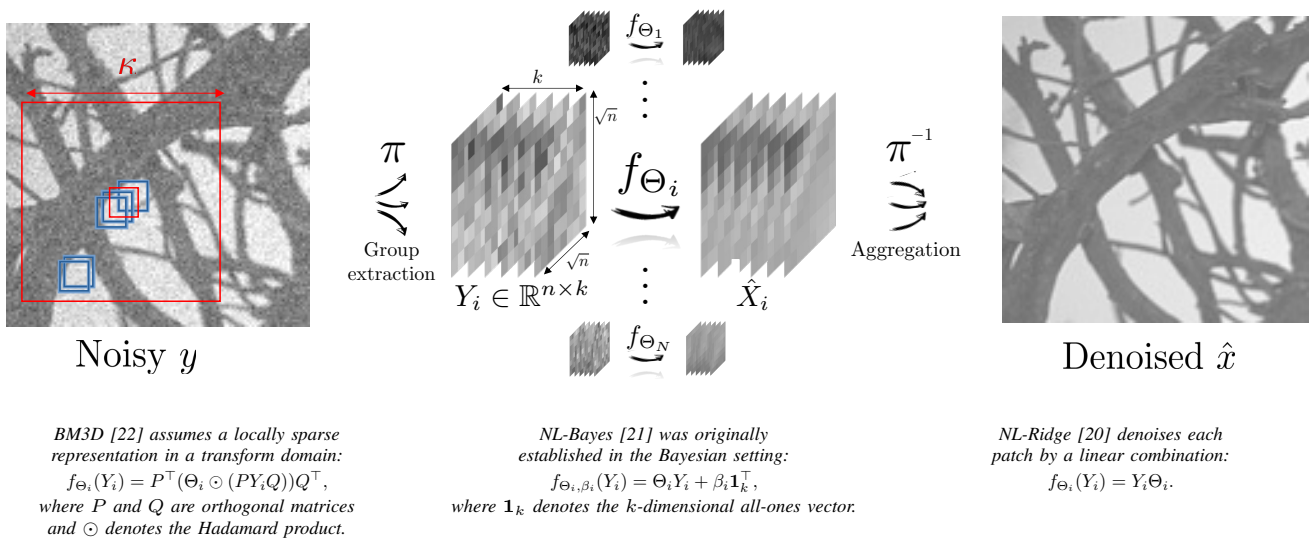


Fig. 2. Illustration of the parametric view of several popular non-local denoisers [20]–[22]. Examples of parameterized functions f_{Θ_i} , unequivocally identifying the denoiser, are given whose parameters Θ_i are eventually selected for each group of patches by “internal adaptation” (see equation (6)).

methods can be reconciled starting from the definition of a family of estimators. Under this paradigm, we inferred a novel algorithm (NL-Ridge [20]) based on ridge regressions, which, despite its apparent simplicity, obtains the best performance when compared to [21], [22]. A natural idea for improving these non-local two-step methods [20]–[22] is to repeat the second step iteratively, taking advantage of the availability of a supposedly better image estimate with each iteration. However, counter-intuitively, it is disappointing in practice as if these methods intrinsically peaked at the second step, with no theoretical justification.

In this paper, our main contributions are the following ones:

- 1) In order to overcome the second stage limitation, we propose to generalize the underlying parametric formulation of non-local denoisers by a novel chaining technique. Compared to other patch-based algorithms, the proposed method does not exhibit a breakdown in reconstruction quality after two iterations. This particularity allows to iterate the process a dozen of times, improving gradually reconstruction quality in small steps [43]. Therefore, the proposed algorithm named LICH (Linear and Iterative Combinations of patchEs for Image denoising) removes a large amount of denoising artifacts, resulting in visually pleasant final images.
- 2) The LICH algorithm achieves state-of-the-art performance among the self-supervised methods (see Fig. 1), while being much faster (factor 8, CPU) than the best performing WNNM [23] method and fully interpretable, requiring no matrix diagonalization. This proves that leveraging only linear combinations of patches (while WNNM is based on a low-rank and SVD decomposition approach) are unreasonably effective for single-image denoising. Indeed, contrary to what might sometimes be believed, deep-learning-based methods are not competitive in self-supervised denoising (see Fig. 1).

- 3) We propose an original technique, grounded in the risk minimization theory and inspired by deep learning theory, for deriving an initial pilot, and study its influence on the final result. To our knowledge, this is the first time such a study is described. We demonstrate that whatever the initial pilot is, similar denoising results are surprisingly obtained.

The remainder of the paper is organized as follows. In Section II, we describe a parametric view of non-local two-step denoisers [20]–[22] and confirm the second stage limitation. In Section III, we introduce a novel chaining technique of the aforementioned denoisers. Our progressive scheme approximates the optimal parameters in a self-supervised manner when considering linear combinations of similar patches. In Section IV, leveraging some techniques inspired from deep-learning theory, we show how to derive an initial pilot, and study its impact on the final result. Finally, in Section V, experimental results on popular datasets, either artificially noisy or real-world data, demonstrate that the resulting algorithm outperforms the self-supervised deep-learning-based techniques and compares favorably with the very best single-image methods [23], [25] while being much faster at execution.

II. A PARAMETRIC VIEW OF TWO-STEP NON-LOCAL METHODS FOR SINGLE-IMAGE DENOISING

In this section, we present the framework of non-local methods and the unified formulation detailed in [20], by taking an even broader view. In this line of work, we propose an extended formulation that constitutes the foundation on which we build upon in the following sections.

A. A unified framework for non-local denoisers

Popularized by BM3D [22], the grouping technique (*a.k.a.* block-matching) has proven to be a key ingredient in achieving

state-of-the-art performance in single-image denoising [20], [21], [23], [38], [39]. This technique consists in gathering noisy patches together according to their similarities in order to denoise them collaboratively. Figure 2 summarizes the whole process composed of three steps. First, groups of k similar noisy square patches $\sqrt{n} \times \sqrt{n}$ are formed. For each overlapping patch i taken as reference, the similarity (e.g. ℓ_2 sense) with its surrounding overlapping patches is computed. The k -nearest neighbors, including the reference patch, are then selected to form a block of similar patches encoded in a matrix $Y_i \in \mathbb{R}^{n \times k}$, where each column represents a flattened patch. Note that the number of groups of patches is strictly equal to the number N of overlapping patches in the noisy input image. Subsequently, the N groups are processed in parallel by applying a local denoising function f . An estimate of the noise-free corresponding block of similar patches $\hat{X}_i = f(Y_i) \in \mathbb{R}^{n \times k}$ is then obtained for each group. Finally, the denoised patches are repositioned to their initial locations in the image and aggregated, or reprojected [44], as pixels may have several estimates. Generally, arithmetic (sometimes weighed) averaging is used to that end.

In this framework, the choice of the local denoising function f remains an open question. Restricting it to be a member of a class of parameterized functions (f_{Θ}), we have proposed in [20] a unified framework to properly select one candidate among the chosen class for each group of patches; Figure 2 gives the underlying parameterized functions for three different denoisers [20]–[22]. Formally, adopting an even broader view, a non-local denoiser ϕ_{Θ} taking as input a noisy image y composed of N overlapping patches of size $\sqrt{n} \times \sqrt{n}$ can itself be viewed as a high-dimensional parametric function:

$$\phi_{\Theta}(y) = \pi^{-1}(F_{\Theta}(\pi(y))) \quad (1)$$

where $\pi : y \mapsto \mathbf{Y}$ is an operator that extracts N blocks of similar patches, viewed as a third-order tensor (i.e. three dimensional array) $\mathbf{Y} = \{Y_i\}_{i=1}^N \in \mathbb{R}^{N \times n \times k}$, π^{-1} is its *pseudo*-inverse (replacing the patches at their initial positions and aggregating them by averaging), F_{Θ} is the function performing the denoising of all blocks of similar patches in a parallel fashion and $\Theta = \{\Theta_i\}_{i=1}^N$. More precisely, $F_{\Theta} : \mathbf{Y} \in \mathbb{R}^{N \times n \times k} \mapsto \mathbb{R}^{N \times n \times k}$ is such that $F_{\Theta}(\mathbf{Y})_{i,\cdot,\cdot} = f_{\Theta_i}(\mathbf{Y}_{i,\cdot,\cdot}) = f_{\Theta_i}(Y_i)$ for all i . In other words, this function processes each group independently through f_{Θ_i} which is exclusively dedicated to the i^{th} block of similar patches Y_i . In the following, we assume that the patch grouping operator π is ideal and forms the patch groups solely based on the similarity of the underlying noise-free patches, and thus independently of the noise realization. This way, $\pi(y)_i$ can be identified as the i^{th} noisy block of similar patches associated to the noise-free one $\pi(x)_i$.

It is worth noting that the number of parameters of ϕ_{Θ} is N times the number of parameters of a single local denoising function f_{Θ_i} . Therefore, the number of parameters grows linearly with the number of patches. As an illustrative example, $\Theta \in \mathbb{R}^{N \times n \times k}$ in the case of BM3D [22] because $\Theta_i \in \mathbb{R}^{n \times k}$ has the same size as a patch group (see Fig. 2). This represents about a hundred million parameters to be found for a 256×256 image with standard patch and group sizes (e.g. $n = 8 \times 8$ and

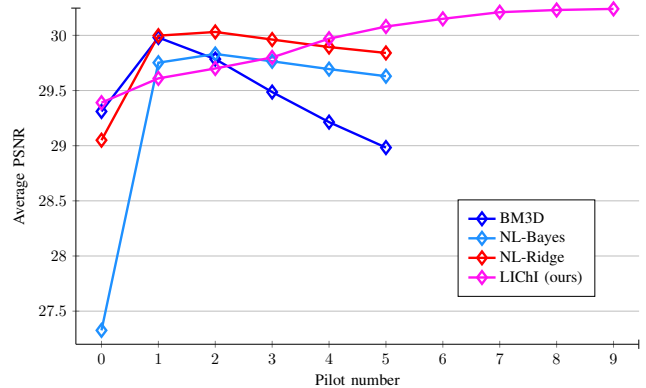


Fig. 3. Evolution of the PSNR for pilot-based single-image denoising methods [20]–[22] on Set12 dataset with noise level $\sigma = 25$. After a PSNR jump between the first and second pilot, obtained with “internal adaptation”, the PSNR decreases for all methods, except ours.

$k = 16$). Fortunately, solutions do exist in practice to reduce this high number of parameters - and thus the computational burden of non-local denoisers - such as the *step trick* (or sub-sampling) which is discussed in Section V-A. In addition to sub-sampling, a heuristic technique commonly used to avoid saturating memory in the case of very large images is simply to process the input image sequentially tile by tile (where a tile is a big patch of size $1,024 \times 1,024$ for example), taking care to avoid edge effects during the final assembly.

B. Parameter optimization

In [20], we showed that several single-image two-step non-local algorithms [20]–[22] could be reconciled by adopting a local minimal risk point of view. The ultimate objective is to determine the parameters $\{\Theta_i\}_{i=1}^N$ by minimizing the global risk defined as:

$$\mathcal{R}_{\Theta}(x) = \mathbb{E} \|\phi_{\Theta}(y) - x\|_2^2, \quad (2)$$

where x is the true image and y is the noisy image. The optimal estimator is $\hat{x} = \phi_{\Theta^*}(y)$ where Θ^* is the minimizer of (2):

$$\Theta^* = \arg \min_{\Theta} \mathcal{R}_{\Theta}(x). \quad (3)$$

Solving (3) directly is difficult due to the intractability of the aggregation operator π^{-1} in (1). Therefore, a suboptimal greedy approach is used and aims at minimizing the risk at the individual patch group level, as originally proposed in [20]. This allows one to decompose the problem into N simpler independent subproblems:

$$\Theta_i^* = \arg \min_{\Theta_i} \underbrace{\mathbb{E} \|f_{\Theta_i}(Y_i) - X_i\|_F^2}_{\mathcal{R}_{\Theta_i}(X_i)}, \quad (4)$$

where $Y_i = \pi(y)_i$ and $X_i = \pi(x)_i$ are the i^{th} noisy and noise-free blocks of similar patches, respectively, and $\|\cdot\|_F$ denotes the Frobenius norm. The problem (4) has a closed-form solution in the case of additive white Gaussian noise as demonstrated in [20] for the underlying parameterized functions of [20]–[22] illustrated in Fig. 2.

C. Principle of internal adaptation

As the true image x is not known, (3) cannot actually be solved. However, assuming that an initial estimate \tilde{x} of the denoised image (*a.k.a.* pilot or oracle estimator [45]) is available, G. Vaksman *et al.* [4] proposed, in the context of deep learning, to substitute \tilde{x} for x in (2). Formally, the idea is to consider the surrogate:

$$\mathcal{R}_{\Theta}(\tilde{x}) = \mathbb{E}\|\phi_{\Theta}(y) - \tilde{x}\|_2^2, \quad (5)$$

where y follows a distribution depending on the noise model; in the case of additive white Gaussian noise of variance σ^2 , $y \sim \mathcal{N}(\tilde{x}, \sigma^2 I_d)$.

Originally, this so-called ‘‘internal adaptation’’ technique was presented as a simple post-processing refinement to boost performance of lightweight networks already trained in a supervised manner [4], [46], [47]. In particular, as argued in [4], the ‘‘internal adaptation’’ trick is useful if the input noisy image y deviates from the general statistics of the training set. Actually, it turns out that this technique is at the core of the second stage of several state-of-the-art single-image two-step denoisers [20]–[22] where each local risk (4) is replaced by the empirical one:

$$R_{\Theta_i}(\tilde{X}_i) = \mathbb{E}\|f_{\Theta_i}(Y_i) - \tilde{X}_i\|_F^2, \quad (6)$$

where $Y_i = \pi(y)_i$ and $\tilde{X}_i = \pi(\tilde{x})_i$.

As long as the pilot \tilde{x} is not too far from the true image x , $\hat{x} = \phi_{\Theta^*}(y)$ obtained through ‘‘internal adaptation’’ by minimizing (5) may be closer to x than the pilot itself (although there is no mathematical guarantee). In practice, all two-step denoisers [20]–[22] always observe a significant boost in performance using this technique compared to the estimator obtained after the first stage. However, counter-intuitively, repeating the process does not bring much improvement and tends on the contrary to severely degrade the image after a few iterations as attested by Figure 3. Therefore, these methods stop directly after a single step of ‘‘internal adaptation’’.

In order to overcome the second stage limitation and boost performance beyond the second iteration, we introduce hereunder a generalized expression of (1). Using a progressive optimization scheme, our algorithm, that only performs linear combinations of patches, enables to significantly improve the denoising performance at each iteration, making it as competitive as WNNM [23], the best single-image method to the best of our knowledge (see Fig. 1).

III. LICH1: LINEAR AND ITERATIVE COMBINATIONS OF PATCHES FOR IMAGE DENOISING

In the following, we assume an additive white Gaussian noise model of variance σ^2 .

A. A novel chaining rule for generalization

We propose to study a class of parameterized functions that generalizes (1):

$$\Phi_{\{\Theta_m\}_{m=1}^M}(y) = [\phi_{\Theta_M} \circ \dots \circ \phi_{\Theta_1}](y) \quad (7)$$

where $M \in \mathbb{N}^*$ and \circ denotes the function composition operator. In other words, we consider the M times iterated

version of function (1). In our approach, we focus on group denoising functions of the following straightforward form:

$$f_{\Theta_i}(Y_i) = Y_i \Theta_i \quad (8)$$

as in [20] (see Fig. 2). This choice is motivated by the fact that, despite their apparent simplicity, we proved in [20] that linear combinations of patches are promising for single-image denoising. Note that when fixing $\Theta_m = \{I_k\}_{i=1}^N$ for $m \geq 2$, where I_k denotes the identity matrix of size k , the above class of functions coincides with (1) as f_{I_k} is the identity function $\text{id}_{\mathbb{R}^{n \times k}}$.

B. A progressive scheme for parameter optimization

Following the same approach as for two-step non-local denoisers, our objective is to minimize the quadratic risk:

$$\{\Theta_m^*\}_{m=1}^M = \arg \min_{\{\Theta_m\}_{m=1}^M} \underbrace{\mathbb{E}\|\Phi_{\{\Theta_m\}_{m=1}^M}(y) - x\|_2^2}_{\mathcal{R}_{\{\Theta_m\}_{m=1}^M}(x)}, \quad (9)$$

where x is the true image assumed to be known and $y \sim \mathcal{N}(x, \sigma^2 I)$. The optimal estimator, in the ℓ_2 sense, is then $\hat{x} = \Phi_{\{\Theta_m^*\}_{m=1}^M}(y)$.

Solving (9) is much more challenging than minimizing (2) due to the repeated aggregation/extraction steps implicitly contained in expression (7) via the operation $\pi \circ \pi^{-1}$. Indeed, it is worth noting that $[\pi \circ \pi^{-1}](z) \neq z$ for $z \in \mathbb{R}^{N \times n \times k}$ when patches in z are not consistent (*i.e.* there exists two different patch estimates for the same underlying patch). Therefore, we propose a (suboptimal) progressive approach to approximate the solution of (9) as follows:

$$\begin{cases} \Theta_1^* = \arg \min_{\Theta_1} \mathbb{E}\|\phi_{\Theta_1}(y) - y_1\|_2^2 \\ \Theta_2^* = \arg \min_{\Theta_2} \mathbb{E}\|[\phi_{\Theta_2} \circ \phi_{\Theta_1^*}](y) - y_2\|_2^2 \\ \vdots \\ \Theta_M^* = \arg \min_{\Theta_M} \mathbb{E}\|[\phi_{\Theta_M} \circ \phi_{\Theta_{M-1}^*} \circ \dots \circ \phi_{\Theta_1^*}](y) - y_M\|_2^2 \end{cases} \quad (10)$$

where $y_m = x + \tau_m(y - x)$ with $(\tau_m)_{1 \leq m \leq M}$ a strictly decreasing sequence satisfying $0 \leq \tau_m < 1$ and $\tau_M = 0$ (*i.e.* $y_M = x$). Basically, Θ_m are found iteratively in a way such that composing by a new ϕ_{Θ_m} closes the gap even more with the true image x . Essentially, the proposed scheme amounts to solving M problems of the form:

$$\Theta_m^* = \arg \min_{\Theta_m} \mathbb{E}\|\phi_{\Theta_m}(z_{m-1}) - y_m\|_2^2, \quad (11)$$

where $z_m = [\phi_{\Theta_m^*} \circ \dots \circ \phi_{\Theta_1^*}](y)$ if $m \geq 1$ and $z_0 = y$ (note that, by construction, z_m is expected to be close to y_m).

C. Resolution when the true image is available

In order to solve (11), we adopt a greedy approach by minimizing the quadratic loss at the individual patch group level as performed in (4). The problem is then decomposed into as many independent subproblems as there are patch groups:

$$\Theta_i^{m*} = \arg \min_{\Theta_i^m} \mathbb{E}\|f_{\Theta_i^m}(Z_i^{m-1}) - Y_i^m\|_F^2, \quad (12)$$

Algorithm 1 LICHl: Linear and Iterative Combinations of patchHes for Image denoising

Input: Noisy image y , initial pilot \tilde{x}_1 , noise level σ , group size k , patch size \sqrt{n} , number of iterations M , sequence $(\tau_i)_{1 \leq i \leq M}$.

Output: Denoised image \hat{x} .

$z_0 = y$

for $m = 1$ to M **do**

for each $\sqrt{n} \times \sqrt{n}$ patch in z_{m-1} indexed by i **do**

 Find its k most similar patches in z_{m-1} to form the block of similar patches Z_i^{m-1} .

 Form \tilde{X}_i^m and Y_i^m with the corresponding patches in \tilde{x}_m and y , respectively.

$t_i^{m-1} = 1 - \text{sd}(Y_i^m - Z_i^{m-1})/\sigma$

$\lambda_i^{m-1} = n(t_i^{m-1}\sigma)^2$

$\Xi_i^m = I_k - \lambda_i^{m-1} \left(\tilde{X}_i^{m\top} \tilde{X}_i^m + \lambda_i^{m-1} I_k \right)^{-1}$

$\tilde{X}_i^{m+1} = Z_i^{m-1} \Xi_i^m$

$\Theta_i^m = \left(1 - \frac{\tau_m}{t_i^{m-1}} \right) \Xi_i^m + \frac{\tau_m}{t_i^{m-1}} I_k$

$Z_i^m = Z_i^{m-1} \Theta_i^m$

end for

 Reposition and aggregate patches of each patch group Z_i^m and \tilde{X}_i^{m+1} to form z_m and updated pilot \tilde{x}_{m+1} .

end for

return z_M

where $Y_i^m = \pi(y_m)_i = X_i + \tau_m(Y_i - X_i)$ with $X_i = \pi(x)_i$ and $Z_i^{m-1} = \pi(z_{m-1})_i$.

In its current state, (12) cannot be solved easily as in (4) because the probability distribution of the pixels contained in Z_i^{m-1} is intractable. Indeed, the repeated aggregation/extraction steps from which Z_i^{m-1} is formed make obtaining its law cumbersome. However, it can be approximated by construction as a convex combination of the i^{th} noisy and noise-free blocks of similar patches $Y_i = \pi(y)_i$ and $X_i = \pi(x)_i$, respectively, that is:

$$Z_i^{m-1} \approx X_i + t_i^{m-1}(Y_i - X_i), \quad (13)$$

where $t_i^{m-1} \in (0, 1]$ is estimated for each block of similar patches and is expected to be close to τ_{m-1} when $m \geq 2$. Note that for $m = 1$, this approximation is in fact exact with $t_i^0 = 1$. Denoting $\text{sd}(\cdot)$ the operator that computes the standard deviation of the coefficients of the input random matrix, we have $\text{sd}(Y_i - Z_i^{m-1}) = (1 - t_i^{m-1})\sigma$. The parameter t_i^{m-1} can therefore be estimated as follows:

$$t_i^{m-1} = 1 - \text{sd}(Y_i - Z_i^{m-1})/\sigma. \quad (14)$$

Finally, conceding this small approximation, the minimizer of (12) has the following closed-form solution (see proof in Appendix):

$$\Theta_i^{m*} = I_k - \left(1 - \frac{\tau_m}{t_i^{m-1}} \right) \lambda_i^{m-1} \left(X_i^\top X_i + \lambda_i^{m-1} I_k \right)^{-1}, \quad (15)$$

with $\lambda_i^{m-1} = n(t_i^{m-1}\sigma)^2$.

D. Use of multiple cost-efficient pilots to guide estimation

Solving the initial objective (9) is impossible in practice, whatever the scheme of optimization adopted, as the true image x is missing. In Section II, we have mentioned that substituting a pilot \tilde{x} for x , that is applying ‘‘internal adaptation’’ [4], constitutes the reference method to overcome this issue when $M = 1$. Here, we propose to use M different pilots $\tilde{x}_1, \dots, \tilde{x}_M$. More precisely, pilot \tilde{x}_m is dedicated to the computation of Θ_m^* as follows:

- $X_i = \pi(x)_i$ is replaced by $\tilde{X}_i^m = \pi(\tilde{x}_m)_i$;
- t_i^{m-1} is computed using the sample standard deviation in (14) where $Y_i = \pi(y)_i$ and $Z_i^{m-1} = \pi(z_{m-1})_i$ are the only realizations at our disposal;
- Θ_i^{m*} is computed with (15).

Let us assume that, for $m \geq 1$, $\Theta_1^*, \dots, \Theta_{m-1}^*$ have already been computed and that a pilot \tilde{x}_m is available. Then, Θ_m^* can be computed using \tilde{x}_m and we propose to update it for the next step with:

$$\tilde{x}_{m+1} = [\phi_{\Xi_m} \circ \phi_{\Theta_{m-1}^*} \circ \dots \circ \phi_{\Theta_1^*}](y) = \phi_{\Xi_m}(z_{m-1}), \quad (16)$$

where parameters Ξ_m must be found. Ideally, we want:

$$\Xi_m^* = \arg \min_{\Xi_m} \mathbb{E} \|\phi_{\Xi_m}(z_{m-1}) - x\|_2^2, \quad (17)$$

for which the solution is given, according to (15) with $\tau_m = 0$, by:

$$\Xi_m^* = \left\{ I_k - \lambda_i^{m-1} \left(X_i^\top X_i + \lambda_i^{m-1} I_k \right)^{-1} \right\}_{i=1}^N, \quad (18)$$

with $\lambda_i^{m-1} = n(t_i^{m-1}\sigma)^2$.

Nevertheless, as x is unknown, $X_i = \pi(x)_i$ is replaced by the previous pilot, that is $\tilde{X}_i^m = \pi(\tilde{x}_m)_i$, and sample standard deviation is used for the computation of t_i^{m-1} in (14). This way, provided that an initial pilot \tilde{x}_1 is available, all set of matrices from Θ_1^* to Θ_M^* can be computed iteratively with updated pilots at each step to finally get

$$\hat{x} = \Phi_{\{\Theta_m^*\}_{m=1}^M}(y) = z_M = \tilde{x}_M, \quad (19)$$

as the final estimate for x . As for the choice of the initial pilot \tilde{x}_1 , the reader is referred to Section IV in this regard. Figure 3 shows how the M pilots improve with each iteration.

We want to emphasize that using the M proposed pilots instead of a single one for each step does not increase the computational complexity. Indeed, the cost for computing the Ξ_m^* can be immediately recycled to compute the Θ_m^* at the same time, by noticing how close they are in expression. The whole procedure is summarized in Algorithm 1 where patch grouping is performed on z_m at step m .

E. Weighted average reprojection

At step m , after processing all groups of similar patches thanks to functions F_{Θ_m} and F_{Ξ_m} , all processed patches are repositioned at their right location in the image and aggregated by averaging via π^{-1} operator. If arithmetic averaging is possible for the aggregation of the pixels belonging to the same position in the image, a weighted-average reprojection is recommended in [44]. As suggested in [44], each pixel

Algorithm 2 Pilot computation

Input: Noisy image y , noise level σ , group size k , patch size \sqrt{n} .

Output: Pilot estimation \tilde{x} .

for each $\sqrt{n} \times \sqrt{n}$ patch in y indexed by i **do**

Find its k most similar patches in y to form the block of similar patches Y_i .

Compute combination weights Θ_i with (21), (26), (27) or (28).

$\tilde{X}_i = Y_i \Theta_i$.

end for

Reposition and aggregate patches of each patch group \tilde{X}_i to form the pilot image \tilde{x} .

return \tilde{x}

belonging to column j of Z_i^m (*i.e.* belonging to the j^{th} patch of Z_i^m) is assigned a weight inversely proportional to the squared ℓ_2 norm of the combination weights calculated for its processing, that is proportional to $1/\|\Theta_i^m e_j\|_2^2$ or $1/\|\Xi_i^m e_j\|_2^2$, depending on the combinations weights used, where e_j is the j^{th} canonical basis vector of \mathbb{R}^k . Those weights are such that the sum of all weights associated to a same pixel equals one.

IV. BUILDING AN INITIAL PILOT

In Algorithm 1, an initial pilot \tilde{x}_1 is necessary to start. If, in theory, any denoiser can be used to that end, we show in this section how to build one of the form $\tilde{x}_1 = \phi_{\Theta}(y)$ where linear combinations of patches are once again leveraged for local denoising (8). The denoisers that we consider in this section are then described by Algorithm 2, all differing in the estimation of the parameters $\Theta = \{\Theta_i\}_{i=1}^N$ corresponding to the combination weights. At the end, most of them yield the same denoising performance.

A. Stein's unbiased risk estimate (SURE)

Considering the same risk minimization problem as (2) for the optimization of $\Theta = \{\Theta_i\}_{i=1}^N$ brings us back to the study of the N independent subproblems of the form (4). However, this time we aim to minimize each local risk by getting rid of any surrogate for the ground truth blocks of similar patches X_i . Stein's unbiased risk estimate (SURE) is probably the most traditional choice as it only depends on the noisy image. Indeed, this popular estimate in image denoising [48]–[50] provides an approximation of the risk $R_{\Theta_i}(X_i)$ that solely depends on the observation Y_i . In the case of linear combinations of patches (8), the computation of SURE yields (see [20]):

$$\text{SURE}_{\Theta_i}(Y_i) = -kn\sigma^2 + \|Y_i \Theta_i - Y_i\|_F^2 + 2n\sigma^2 \text{tr}(\Theta_i), \quad (20)$$

where $\text{tr}(\cdot)$ denotes the trace operator. Substituting this estimate for the risk $R_{\Theta_i}(X_i)$ and minimizing $\text{SURE}_{\Theta_i}(Y_i)$ with regards to Θ_i , we get:

$$\Theta_i^{\text{SURE}} = I_k - n\sigma^2 (Y_i^\top Y_i)^{-1}. \quad (21)$$

Note that Θ_i^{SURE} is close to the parameters Θ_i^* minimizing the risk as long as the variance of SURE is low. A rule of thumbs

used in [48] states that the number of parameters must not be “too large” compared to the number of data in order for the variance of SURE to remain small. In our case, this suggests that $n > k$. In other words, a small amount of large patches are necessary for applying this technique. Finally, a possible pilot for x is $\tilde{x}_1 = \phi_{\Theta^{\text{SURE}}}(y)$ with $\Theta^{\text{SURE}} = \{\Theta_i^{\text{SURE}}\}_{i=1}^N$.

B. Noisier2Noise

Although somewhat counter-intuitive, N. Moran *et al.* [29] showed that training a dataset-based neural network to recover the original noisy image from a noisier version (synthetically generated by adding extra noise) constitutes an efficient strategy to learn denoising weights without access to any clean training examples. Transposing this idea to single-image denoising, this amounts in our case to considering the minimization of the following risk:

$$\mathcal{R}_{\Theta}(y) = \mathbb{E}\|\phi_{\Theta}(z) - y\|_2^2, \quad (22)$$

where y is the only noisy observation at our disposal and z is a noisier random vector; in the case of additive white Gaussian noise of variance $(\alpha\sigma)^2$, where $\alpha > 0$ is an hyperparameter controlling the amount of extra noise, $z \sim \mathcal{N}(y, (\alpha\sigma)^2 I)$. Formally, minimizing (22) is no more difficult than minimizing (2) and the same greedy approximation used in (4) can be applied to solve the N independent local subproblems:

$$\hat{\Theta}_{\alpha,i} = \arg \min_{\Theta_i} \mathbb{E}\|f_{\Theta_i}(Z_i) - Y_i\|_F^2, \quad (23)$$

where $Z_i = \pi(z)_i$ and $Y_i = \pi(y)_i$. As showed in [20], the problem (23) amounts to solving a multivariate ridge regression for which the closed-form solution can be found:

$$\hat{\Theta}_{\alpha,i} = I_k - n(\alpha\sigma)^2 (Y_i^\top Y_i + n(\alpha\sigma)^2 I_k)^{-1}. \quad (24)$$

To get an estimate of the noise-free image x , Noisier2Noise [29] suggests to compute:

$$\mathbb{E}(x|z) \approx \frac{(1 + \alpha^2)\phi_{\hat{\Theta}_{\alpha}}(z) - z}{\alpha^2}, \quad (25)$$

where $\hat{\Theta}_{\alpha}$ is the minimizer of (22) approximated by $\{\hat{\Theta}_{\alpha,i}\}_{i=1}^N$. In Appendix, we show that this quantity is equal to $\phi_{\Theta_{\alpha}^{\text{Nr2N}}}(y)$ on average with $\Theta_{\alpha}^{\text{Nr2N}} = \{\Theta_{\alpha,i}^{\text{Nr2N}}\}_{i=1}^N$ where:

$$\Theta_{\alpha,i}^{\text{Nr2N}} = I_k - n(1 + \alpha^2)\sigma^2 (Y_i^\top Y_i + n(\alpha\sigma)^2 I_k)^{-1}. \quad (26)$$

The choice of the hyperparameter α remains an open question. N. Moran *et al.* [29] recommend to set $\alpha = 0.5$ to handle a variety of noise levels in their experiments. Interestingly, for $\alpha \rightarrow 0$, parameters $\Theta_{\alpha}^{\text{Nr2N}}$ converge to Θ^{SURE} ; a practical advantage of $\Theta_{\alpha}^{\text{Nr2N}}$ over Θ^{SURE} is that the matrices $Y_i^\top Y_i + n(\alpha\sigma)^2 I_k$ in (26) are symmetric positive-definite and therefore invertible, contrary to $Y_i^\top Y_i$ in (21) which is only positive semi-definite and positive-definite almost surely in the case of ideal additive white Gaussian noise when $n \geq k$. For real-world noisy images, estimation through combination weights $\Theta_{\alpha}^{\text{Nr2N}}$ is recommended over Θ^{SURE} as, in some cases, matrices $Y_i^\top Y_i$ may not be invertible. By the way, the combination weights (26) can be efficiently computed based on the Cholesky factorization [51].

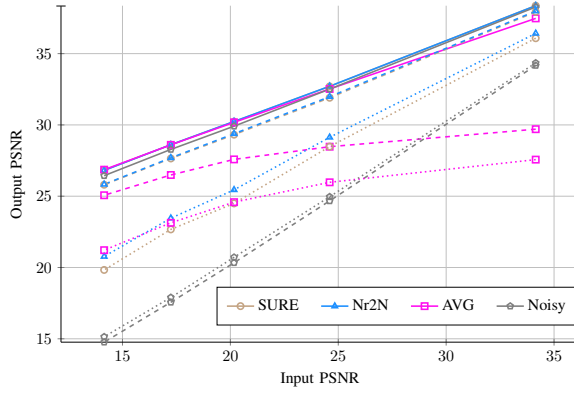


Fig. 4. Average PSNR (in dB) results on patch groups (dotted line), after aggregation (dashed line) and when taken as input for Algorithm 1 (solid line) for Set12 dataset depending on combination weights used and noise level. Patch and group sizes are chosen as indicated by Table III.

It is worth noting that the same weight expressions as (26) can also be obtained within the Recorruped-to-Recorruped paradigm [30], which was originally applied in a dataset-based deep learning context, providing an unbiased estimate of a different type of risk which is close to (4).

C. Two additional extreme pilots

In the case where the noisy patches within a group Y_i are originally strictly identical (perfect patch group), the optimal weights are the ones computing an arithmetic averaging:

$$\Theta_i^{\text{AVG}} = \mathbf{1}_k \mathbf{1}_k^\top / k, \quad (27)$$

where $\mathbf{1}_k$ denotes the k -dimensional all-ones vector. Under the optimistic assumption that each patch group formed is perfect, the pilot $\tilde{x}_1 = \phi_{\Theta^{\text{AVG}}}(y)$ with $\Theta^{\text{AVG}} = \{\Theta_i^{\text{AVG}}\}_{i=1}^N$ is then optimal.

On the contrary, when the patch groups formed are highly dissimilar, collaborative denoising cannot be beneficial and the resulting “do-nothing” weights are:

$$\Theta_i^{\text{Noisy}} = I_k, \quad (28)$$

where I_k is the identity matrix of size k . Under this pessimistic assumption, the pilot $\tilde{x}_1 = \phi_{\Theta^{\text{Noisy}}}(y) = y$ is optimal. This amounts to considering the original noisy image itself as an initial pilot in Algorithm 1.

D. Comparison of the pilots

To study the performance of the proposed pilots, we examined the outputs at three different levels: *i*) the individual patch group level; *ii*) the global level after the aggregation stage; *iii*) the output of Algorithm 1. Figure 4 displays the average PSNR results obtained for these three levels computed for different noise levels on Set12 dataset. Although the studied pilots have very different behaviors at the patch group level, they tend to give similar results when used as inputs for Algorithm 1. The “do-nothing” weights (28), however, perform slightly worse than the others, especially at high noise levels, while the averaging ones (27) are disappointing for low noise levels. As for SURE (21) and Noisier2Noise (26) weights,

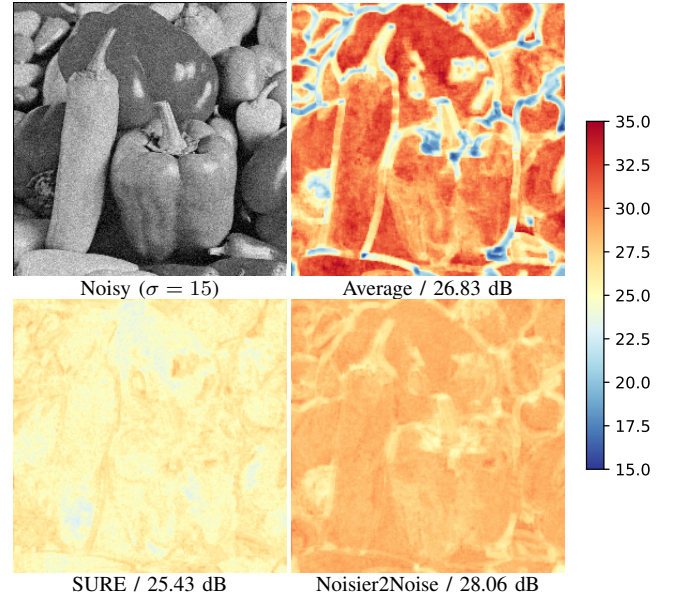


Fig. 5. Colormap of the PSNR (in dB) of the denoised blocks of similar patches ($n = 7 \times 7$ and $k = 18$) associated with each overlapping patch of the noisy image. The average PSNR on blocks of similar patches is also indicated.

they give almost identical results in the end even if the Noisier2Noise weights are much more efficient on the blocks of similar patches. By the way, this highlights a non-intuitive phenomenon which was already observed in [46]: efficiency at the patch scale is a sufficient but not necessary condition to be efficient after the aggregation stage. This confirms that aggregation is not a basic post-processing step but plays a crucial role in image denoising.

For illustration, Fig. 5 provides a visual comparison of the performance of the different combination weights Θ_i^{Name} where $\text{Name} = \{\text{SURE}, \text{Nr2N}, \text{AVG}, \text{Noisy}\}$, depending on the location of the reference patch for intermediate noise level. Unsurprisingly, combination weights (27) are extremely effective on the smooth parts of the image because they are theoretically optimal when applied on groups of patches being originally identical. However, when the patch groups are less homogeneous, which occurs when the reference patch is a rare patch, averaging over inherently dissimilar patches severely affects denoising. On the contrary, SURE (21) and Noisier2Noise (26) weights seem to be more versatile and less sensitive to the homogeneity of the blocks of similar patches, yielding comparable reconstruction errors regardless of the rarity of the reference patch.

E. The crucial role of the aggregation stage

To get a better understanding of the role of the aggregation stage, let us define $\psi_{\Theta}(y)$ the estimator that skips this operation:

$$\psi_{\Theta}(y) = \chi(F_{\Theta}(\pi(y))), \quad (29)$$

where operator χ replaces each patch at their initial location and selects a single estimate among those available for a given pixel. The single estimate is arbitrarily chosen at random from the most central pixels of the denoised reference patches to

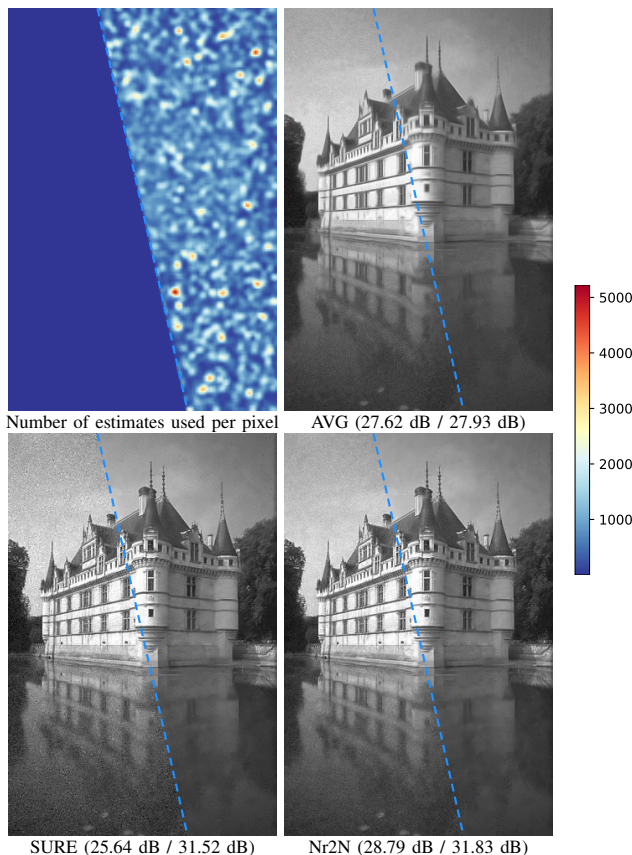


Fig. 6. Image denoising of *Castle* image from BSD68 dataset ($\sigma = 15$) by Algorithm 2 for three different combination weights: (21), (26) and (27). Left: a single estimate per pixel (no aggregation), right: aggregation by averaging all estimates per pixel.

avoid considering poor quality estimates. In particular, when the patch size \sqrt{n} is an odd number, the chosen estimates are the denoised central pixels of the reference patches. Figure 6 illustrates the gap of performance between $\psi_{\Theta}(y)$ and $\phi_{\Theta}(y)$ for combination weights (21), (26) and (27). Skipping the aggregation step results in a much poorer estimation, especially for weights (21) and (26). As a matter of fact, non-local methods have the particularity of producing a large number of estimates per noisy pixel, up to a few thousand (see Fig. 6), because a noisy pixel can appear in many blocks of similar patches and even several times in one. To study the benefit of exploiting those multiple estimates, a bias-variance decomposition can be leveraged:

$$\underbrace{\mathbb{E}\|\tilde{x} - x\|_2^2/d}_{\text{MSE}} = \underbrace{\mathbb{E}\|\tilde{x} - x\|_2^2/d}_{\text{squared-bias}} + \underbrace{\mathbb{E}\|\tilde{x} - \mathbb{E}(\tilde{x})\|_2^2/d}_{\text{variance}} \quad (30)$$

where \tilde{x} is the estimator for the true image x . Figure 7 highlights the bias-variance tradeoff for estimators $\psi_{\Theta}(y)$ and $\phi_{\Theta}(y)$ and combination weights (21), (26) and (27). We can notice that the squared-bias part of the MSE in (30) is practically unchanged whether aggregation is applied or not. However, a drop in variance is noticeable. This is particularly impressive for SURE estimator (21), which significantly reduces its variance and so the MSE thanks to aggregation, closing the gap with the Noisier2Noise estimator (26) as they share almost the same squared-bias. However, for averaging

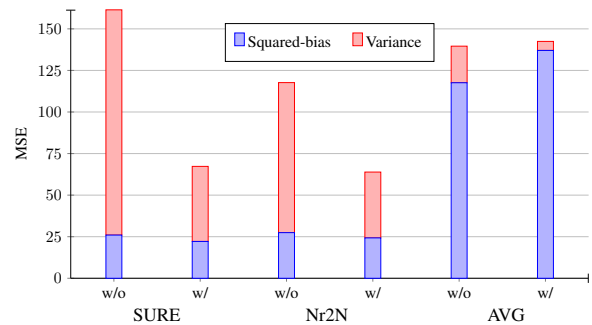


Fig. 7. Bias-variance tradeoff between estimators (1) and (29), i.e. with (w/) and without (w/o) aggregation, for three different types of combination weights. Results obtained with noisy image *Cameraman* from Set12 at noise level $\sigma = 20$ with patch size $n = 9 \times 9$ and group size $k = 18$, estimated via Monte-Carlo simulation using 100 different realizations of the noise.

estimator (27), the variance represents a small part in the MSE decomposition (30) and so aggregation is not beneficial.

We can draw a parallel with a popular machine-learning ensemble meta-algorithm: bootstrap aggregating, also called bagging [52]. Bagging consists of fitting several (“weak” in some sense) models to sampled versions of the original training dataset (bootstrap) and combining them by averaging the outputs of the models during the testing phase (aggregation). This procedure is known to improve model performance, as it decreases the variance of the model, without increasing the squared-bias. In our case, the bootstrap samples can be materialized by the numerous noisy blocks of similar patches Y_i on which (weak) models $f_{\Theta_i}(\cdot)$ are trained in an unsupervised manner. Combining pixel estimates by aggregation enables to significantly reduce the variance while keeping the squared-bias unchanged.

V. EXPERIMENTAL RESULTS

In this section, we compare the performance of the proposed single-image method, referred to as LICH (Algorithm 1), with state-of-the-art methods, including related neural network-based methods [2]–[7], [16], [26], [27], [33], [35]–[37] applied to standard gray images artificially corrupted with additive white Gaussian noise with zero mean and variance σ^2 and on real-world noisy images. We used the implementations provided by the authors as well as the corresponding trained weights for supervised networks. Performance of LICH and other methods are assessed in terms of PSNR values. The code can be downloaded at: <https://github.com/sherbret/LICH/>.

A. Setting of algorithm parameters

In all our experiments, the patch size n , the group size k and the strictly decreasing sequence $(\tau_m)_{1 \leq m \leq M}$ in Algorithm 1 are empirically chosen as follows: $n = 6 \times 6$, $k = 64$ and $\tau_m = 0.75 \times (1 - m/M)$. The number of iterations M depends on the noise level σ ; the higher the noise level, the more iterations of linear combinations of patches are necessary. Moreover, the optimal value of M is also influenced by the quality of the initial pilot, itself depending on patch and group sizes according to Algorithm 2. In Table III, we report, for each

TABLE I

THE PSNR (dB) RESULTS OF DIFFERENT METHODS APPLIED TO THREE DATASETS CORRUPTED WITH SYNTHETIC WHITE GAUSSIAN NOISE. THE BEST METHOD AMONG EACH CATEGORY (DATASET-BASED OR SINGLE-IMAGE) IS EMPHASIZED IN RED; SECOND BEST IS IN BLUE.

Methods		Set12					BSD68					Urban100				
Noise level σ		5	15	25	35	50	5	15	25	35	50	5	15	25	35	50
Dataset-based Neural network	DnCNN [2]	37.74	32.86	30.44	28.82	27.18	37.71	31.73	29.23	27.69	26.23	37.52	32.68	29.97	28.11	26.28
	FFDNet [3]	38.11	32.75	30.43	28.92	27.32	37.80	31.63	29.19	27.73	26.29	38.12	32.43	29.92	28.27	26.52
	LIDIA [4]	-	32.85	30.41	-	27.19	-	31.62	29.11	-	26.17	-	32.80	30.12	-	26.51
	DRUNet [5]	38.64	33.25	30.94	29.45	27.90	38.07	31.91	29.48	28.00	26.59	38.91	33.44	31.11	29.61	27.96
	Restormer [6]	38.70	33.42	31.08	29.57	28.01	38.11	31.96	29.52	28.05	26.62	39.06	33.79	31.46	30.00	28.33
	SCUNet [7]	-	33.43	31.09	-	28.04	-	31.99	29.55	-	26.67	-	33.88	31.58	-	28.56
	Stoformer [16]	-	33.85	31.53	-	28.46	-	32.57	30.06	-	27.07	-	34.24	31.92	-	28.72
Single-image Neural network	DIP [35]	-	30.12	27.54	-	24.67	-	28.83	26.59	-	24.13	-	-	-	-	-
	N2S [33]	-	31.01	28.64	-	25.30	-	29.46	27.72	-	24.77	-	-	-	-	-
	N2F [37]	-	30.21	28.17	-	25.09	-	29.82	27.79	-	25.05	-	-	-	-	-
	ZS-N2N [36]	-	30.05	27.26	-	23.56	-	29.61	26.86	-	23.46	-	-	-	-	-
	S2S [27]	-	32.07	30.02	-	26.49	-	30.62	28.60	-	25.70	-	-	-	-	-
	RDIP [26]	-	32.26	29.79	-	26.60	-	31.21	28.78	-	25.81	-	-	-	-	-
	Single-image Traditional Multi-step 2-step	BM3D [22]	38.02	32.37	29.97	28.40	26.72	37.55	31.07	28.57	27.08	25.62	38.30	32.35	29.70	27.97
NL-Bayes [21]		38.12	32.25	29.88	28.30	26.45	37.62	31.16	28.70	27.18	25.58	38.33	31.96	29.34	27.61	25.56
NL-Ridge [20]		38.19	32.46	30.00	28.41	26.73	37.67	31.20	28.67	27.14	25.67	38.56	32.53	29.90	28.13	26.29
Single-image Traditional Multi-step 1-step	SS-GMM [24]	38.02	32.41	29.88	28.24	26.53	37.17	31.28	28.76	27.19	25.71	-	-	-	-	-
	TWSC [25]	38.17	32.61	30.21	28.63	26.95	37.67	31.28	28.75	27.24	25.76	-	-	-	-	-
	WNNM [23]	38.36	32.70	30.26	28.69	27.05	37.80	31.37	28.83	27.30	25.87	38.77	32.97	30.39	28.70	26.83
	LICHl (ours)	38.36	32.71	30.24	28.61	26.81	37.80	31.41	28.87	27.31	25.72	38.77	33.00	30.37	28.59	26.56

TABLE II

DENOISING RESULTS ON RAW DATA ON DARMSTADT NOISE DATASET (DND). BEST METHOD AMONG EACH CATEGORY IS IN BOLD.

Methods	Single-image						Dataset-based		
	BM3D [22]	NL-Ridge [20]	KSVD [40]	PEWA [41]	NCSR [38]	WNNM [23]	LICHl	DnCNN [2]	N ³ Net [13]
PSNR (in dB)	47.15	47.01	46.87	46.72	47.07	47.05	47.35	47.37	47.56

noise range, the recommended patch size n and group size k in algorithm 2 with Noisier2Noise weights (with $\alpha = 0.5$), as this is the most relevant choice based on the experiments in Section IV, as well as the associated number of iterations M .

TABLE III

RECOMMENDED PATCH SIZE n AND GROUP SIZE k FOR ALGORITHM 2 AND CORRESPONDING NUMBER OF ITERATIONS M IN ALGORITHM 1.

σ	n	k	M
$0 < \sigma \leq 10$	9×9	16	6
$10 < \sigma \leq 30$	11×11	16	9
$30 < \sigma \leq 50$	13×13	16	11

For the sake of computational efficiency, the search for similar patches, computed in the ℓ_2 sense, across the image is restricted to a small local window $\kappa \times \kappa$ centered around each reference patch (in our experiments $\kappa = 65$). Considering iteratively each overlapping patch of the image as reference patch is also computationally demanding, therefore only one patch out of δ , both horizontally and vertically, is considered as a reference patch. The number of reference patches and thus the time spent searching for similar patches is then divided by δ^2 . This common technique [20], [22], [23] is sometimes referred in the literature as the *step trick*. In our experiments, we take $\delta = 3$. Finally, to further speed up the algorithm, the search for the location of patches similar to the reference ones is only performed every third iteration because, in practice, the calculated locations vary little from one iteration to the next.

B. Results on artificially noisy and real-world noisy images

We tested the denoising performance of our method on three well-known grayscale datasets, namely Set12, BSD68 [19] and Urban100 [54], that were artificially corrupted by additive white Gaussian noise (AWGN). PSNR results are reported in Table I (the missing figures are due either to the unavailability of a specific model for the noise levels concerned in the case of dataset-based methods, or to prohibitive execution times on large datasets [54] for single-image methods). For the sake of a fair comparison, algorithms are divided into two categories: single-image methods, meaning that these methods (either traditional or neural network-based) only have access to the input noisy image, and dataset-based methods that require a training phase beforehand on an external dataset. Note that only the single-image extension was considered for Noise2Self [33] and the time-consuming ‘‘internal adaptation’’ option was not used for LIDIA [4]. Results show that, although simpler conceptually, LICHl is as efficient as WNNM [23], the best single-image denoiser, to the best of our knowledge. Note that contrary to our method, WNNM is based on a low-rank and SVD decomposition approach and does not use pilot images. Such results demonstrate that linear combinations of noisy patches provide state-of-the-art performance in single-image denoising. It is worth noting that LICHl excels for low noise levels ($\sigma \leq 15$) on all datasets; however, for very high noise levels ($\sigma \geq 50$), our method seems to lose some of its effectiveness and is surpassed by [23], [25]. Finally,

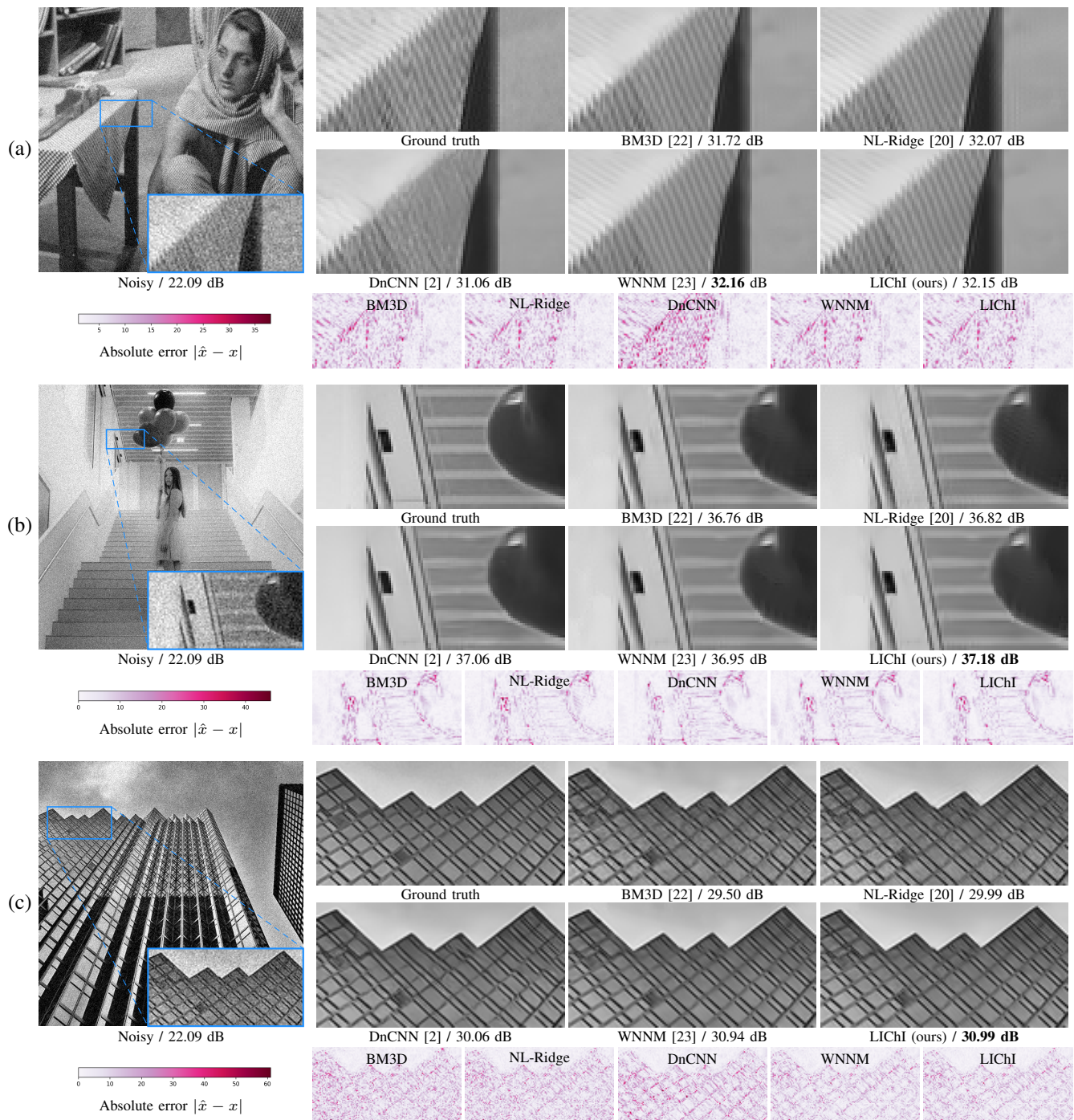


Fig. 8. Qualitative comparison of image denoising results with synthetic white Gaussian noise ($\sigma = 20$). Zoom-in regions are indicated for each method and absolute errors to ground truth are shown for better visualization. From top to bottom: *Barbara* from Set12, *Img009* from Urban100 and *Img019* from Urban100. Best viewed by zooming on a computer screen.

it is interesting to notice that, on Urban100 [54] dataset which contains abundant structural patterns and textures, some supervised neural networks [2]–[4] are outperformed by the best single-image methods.

Figure 8 provides visual comparisons with several popular algorithms. LiChI compares favorably with the very best single-image methods, but also with the famous supervised network DnCNN [2] which is here used as a baseline for

dataset-based methods (better-performing alternatives include [5]–[7], see Table I). In particular, DnCNN [2], contrary to our method, is unable to recover properly the stripes on *Barbara* image (see Fig. 8a). Moreover, the benefit of iterating linear combinations, compared to the one-pass version represented by NL-Ridge [20] is clearly visible. Indeed, many eye-catching artifacts (e.g., see Fig. 8b and 8c), especially around the edges, are removed and the resulting denoised image is much more

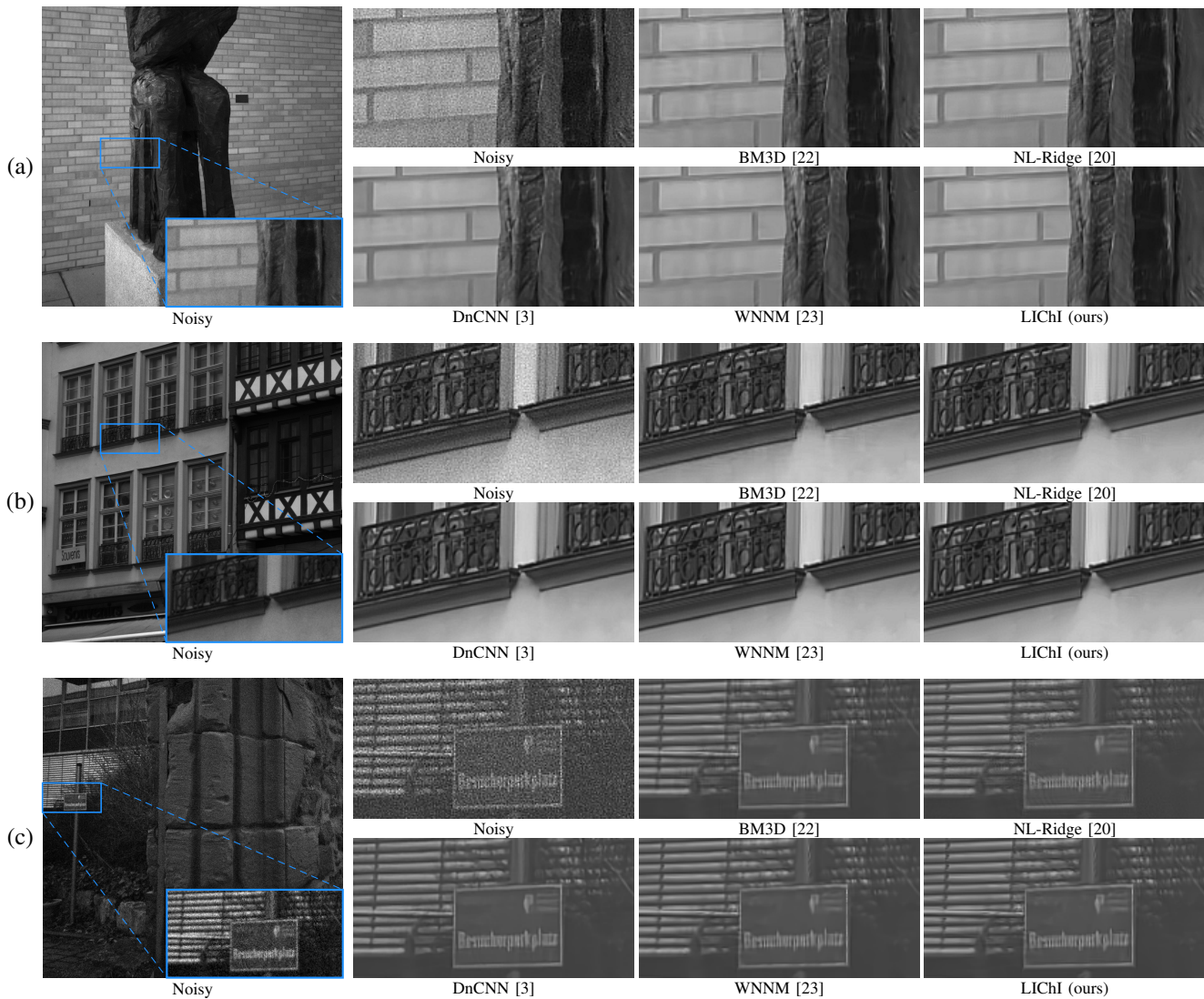


Fig. 9. Qualitative comparison of image denoising results on real-world noisy images from Darmstadt Noise Dataset [53]. Zoom-in regions, enhanced with a 3×3 sharpen filter for better visualization, are indicated for each method. From top to bottom: *Img0003*, *Img0037* and *Img0048*. Best viewed by zooming.

pleasant and natural.

In order to demonstrate the applicability of our method on real-world data, we tested the performance on Darmstadt Noise Dataset [53]. This popular dataset is composed of 50 real-noisy photographs, for which ground-truth images are not disclosed to avoid any bias in the evaluation (denoising results can only be evaluated online at <https://noise.visinf.tu-darmstadt.de/>). The real noise on the photographs of this dataset can be modeled as a Poisson-Gaussian noise, which is further approximated with a heteroscedastic Gaussian noise whose variance is intensity-dependent:

$$y = \mathcal{N}(x, \text{diag}(ax + b)) \quad (31)$$

where $(a, b) \in \mathbb{R}_*^+ \times \mathbb{R}_*^+$ are the noise parameters and diag is the vector-to-matrix diagonal operator. For each noisy image, the authors [53] calculated the adequate noise parameters (a, b) based on this model and made them available to the user. Moreover, they evaluated their own network $N^3\text{Net}$ [13] on this dataset. As no training set is provided for this challenge [53], $N^3\text{Net}$ was trained on academic image databases cor-

rupted by synthetic heteroscedastic Gaussian noise with noise parameters (a, b) similar to those previously calculated in [53]. Same training protocol was used for DnCNN [2]. Both DnCNN [2] and $N^3\text{Net}$ [13] will serve as comparison baselines for dataset-based methods. To apply denoisers dedicated to Gaussian noise removal such as LICHl, a variance-stabilizing transformation (VST) is necessary beforehand. We used the generalized Anscombe transform [55] to that end.

PSNR results are reported in Table II. It turns out that LICHl obtained the best score, surpassing BM3D [22] which was so far the best single-image method on this dataset, further closing the gap with dataset-based methods. This good performance is partly explained by the fact that noise in real-world photographs is generally low, which benefits our method (see Table I). In terms of visual comparison (see Fig. 9), same conclusions can be drawn as for Gaussian noise. Indeed, our method produces much less eye-catching artifacts than [20], [22] and is visually similar to the eye-pleasing results of a supervised neural network such as DnCNN [2].

C. Complexity

We would like to emphasize that although LICH is an iterative algorithm, it is relatively fast compared to its traditional and neural network-based single-image counterparts. In Fig. 1, we have reported the running time of different denoising algorithms. It is provided for informational purposes only, since the implementation, the language used, and the machine on which the code is run strongly influence the execution time. The gap in terms of running time between dataset-based and single-image methods is explained by the fact that these latter solve optimization problems “on the fly”. In comparison, dataset-based methods are trained in advance and this training time, sometimes counting in days on a GPU, is not taking into account in Fig. 1. Nevertheless, it is worth noting that traditional single-image methods [20]–[25] are much less computationally demanding than their state-of-the-art neural network-based counterparts [26], [27], which use time-consuming gradient descent algorithms for optimization. Finally, note that LICH has been entirely implemented in Python with Pytorch [56], enabling it to run on GPU and thus achieve even higher speeds (about 10 times faster with our implementation).

VI. CONCLUSION

In some critical application fields such as biomedical imaging, it is of paramount importance to reduce image noise without creating hallucination artifacts generally induced by the training datasets. It follows that interpretable zero-shot image denoising methods are generally more appreciated by practitioners. Here, we have presented a single-image denoising method based solely on linear combinations of patches that surprisingly achieves state-of-the-art performance. The proposed algorithm is built by extending the unified parametric view of non-local two-step denoisers via a novel chaining rule. Optimization relies on multiple pilot images to guide the estimation of the combination weights. Our experimental results show that LICH preserves much better structural patterns and textures and generates much less visual artifacts, in particular around the edges, than single-iterated denoisers, such as BM3D. The proposed algorithm is on par with WNNM, the best single-image denoiser to the best of our knowledge, in terms of both quantitative measurement and visual perception quality, while being much simpler conceptually and faster at execution.

APPENDIX

In what follows, $X, Y \in \mathbb{R}^{n \times k}$ with $Y_{i,j} \sim \mathcal{N}(X_{i,j}, \sigma^2)$ independent along each row and $f_\Theta : Y \mapsto Y\Theta$ with $\Theta \in \mathbb{R}^{k \times k}$.

Lemma 1. Let $A \in \mathbb{R}^{n \times k}$, $\lambda \in \mathbb{R}^+$ and $\mu \in \mathbb{R}$. If $A^\top A$ is invertible or $\lambda \neq 0$:

$$\begin{aligned} \arg \min_{\Theta \in \mathbb{R}^{k \times k}} \|A\Theta - A\|_F^2 + \lambda \|\Theta\|_F^2 + 2\mu \text{tr}(\Theta) \\ = I_k - (\lambda + \mu)(A^\top A + \lambda I_k)^{-1}. \end{aligned}$$

Proof. Let $H : \Theta \in \mathbb{R}^{k \times k} \mapsto \|A\Theta - A\|_F^2 + \lambda \|\Theta\|_F^2 + 2\mu \text{tr}(\Theta)$ and $h_j : \theta \in \mathbb{R}^k \mapsto \|A\theta - A_{\cdot,j}\|_2^2 + \lambda \|\theta\|_2^2 + 2\mu\theta_j$ such that

$$H(\Theta) = \sum_{j=1}^k h_j(\Theta_{\cdot,j}).$$

Note that Hess $h_j(\theta) = 2(A^\top A + \lambda I_k)$ is symmetric positive definite, hence h_j is strictly convex and so h_j has a unique global minimum. Then, as $A^\top A + \lambda I_k$ is *a fortiori* invertible, we have, by canceling the gradient:

$$\begin{aligned} \nabla h_j(\theta) &= 2A^\top(A\theta - A_{\cdot,j}) + 2\lambda\theta + 2\mu e_j = 0 \\ \Leftrightarrow \theta &= (A^\top A + \lambda I_k)^{-1}(A^\top A_{\cdot,j} - \mu e_j), \end{aligned}$$

and finally, by concatenating all optimal columns, we get:

$$\begin{aligned} \arg \min_{\Theta} H(\Theta) &= (A^\top A + \lambda I_k)^{-1}(A^\top A - \mu I_k) \\ &= I_k - (\lambda + \mu)(A^\top A + \lambda I_k)^{-1}. \end{aligned}$$

□

Proposition 1. Let $(\tau_1, \tau_2) \in \mathbb{R}^2$ with $\tau_1 \neq 0$.

$$\begin{aligned} \arg \min_{\Theta} \mathbb{E} \|f_\Theta(X + \tau_1(Y - X)) - (X + \tau_2(Y - X))\|_F^2 \\ = I_k - (1 - \frac{\tau_2}{\tau_1})n(\tau_1\sigma)^2 (X^\top X + n(\tau_1\sigma)^2 I_k)^{-1}. \end{aligned}$$

Proof. Let $W = Y - X$ and $\Theta' = (\tau_1\Theta - \tau_2 I_k)$.

By development of the squared Frobenius norm and by linearity of expectation:

$$\begin{aligned} \mathbb{E} \|f_\Theta(X + \tau_1 W) - (X + \tau_2 W)\|_F^2 \\ = \mathbb{E} (\|X\Theta - X\|_F^2 + 2\langle X\Theta - X, W\Theta' \rangle_F + \|W\Theta'\|_F^2) \\ = \|X\Theta - X\|_F^2 + \mathbb{E} \|W\Theta'\|_F^2, \end{aligned}$$

with

$$\begin{aligned} \mathbb{E} \|W\Theta'\|_F^2 &= \mathbb{E} \left(\sum_{i=1}^n \sum_{j=1}^k \left(\sum_{l=1}^k W_{i,l} \Theta'_{l,j} \right)^2 \right) \\ &= \sum_{i=1}^n \sum_{j=1}^k \sum_{l=1}^k \sigma^2 \Theta'^2_{l,j} = n\sigma^2 \|\Theta'\|_F^2 \\ &= n\sigma^2 (\tau_1^2 \|\Theta\|_F^2 - 2\tau_1\tau_2 \text{tr}(\Theta) + k\tau_2^2). \end{aligned}$$

The use of the Lemma 1 allows to conclude. □

Proposition 2 (Noisier2Noise). Let $\alpha > 0$ and y, z two vectors with $z \sim \mathcal{N}(y, (\alpha^2\sigma^2)I)$. We have:

$$\mathbb{E} \left[\frac{(1 + \alpha^2)\phi_{\hat{\Theta}_\alpha}(z) - z}{\alpha^2} \right] = \phi_{\Theta_\alpha^{\text{Nr2N}}}(y)$$

with $\hat{\Theta}_\alpha$ and $\Theta_\alpha^{\text{Nr2N}}$ defined in Section IV-B.

Proof. First of all, notice that:

$$\frac{(1 + \alpha^2)\hat{\Theta}_{\alpha,i} - I_k}{\alpha^2} = \Theta_{\alpha,i}^{\text{Nr2N}}.$$

Therefore,

$$\frac{(1 + \alpha^2)\phi_{\hat{\Theta}_\alpha}(z) - z}{\alpha^2} = \phi_{\frac{1+\alpha^2}{\alpha^2}\hat{\Theta}_\alpha - \frac{1}{\alpha^2}I}(z) = \phi_{\Theta_\alpha^{\text{Nr2N}}}(z)$$

with $\mathbf{I} = \{I_k\}_{k=1}^N$. And finally, by linearity of expectation,

$$\mathbb{E} \left[\frac{(1 + \alpha^2) \phi_{\Theta_\alpha}(z) - z}{\alpha^2} \right] = \mathbb{E} \left[\phi_{\Theta_\alpha^{N/2N}}(z) \right] = \phi_{\Theta_\alpha^{N/2N}}(y).$$

□

ACKNOWLEDGMENT

This work was supported by Bpifrance agency (funding) through the LiChIE contract. Computations were performed on the Inria Rennes computing grid facilities partly funded by France-BioImaging infrastructure (French National Research Agency - ANR-10-INBS-04-07, “Investments for the future”).

We would like to thank R. Fraisse (Airbus) for fruitful discussions.

REFERENCES

- [1] M. Elad, B. Kowar, and G. Vaksman, “Image denoising: The deep learning revolution and beyond—a survey paper,” *SIAM Journal on Imaging Sciences*, vol. 16, no. 3, pp. 1594–1654, 2023.
- [2] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, “Beyond a Gaussian denoiser: residual learning of deep CNN for image denoising,” *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.
- [3] K. Zhang, W. Zuo, and L. Zhang, “FFDNet: Toward a fast and flexible solution for CNN-based image denoising,” *IEEE Transactions on Image Processing*, vol. 27, no. 9, pp. 4608–4622, 2018.
- [4] G. Vaksman, M. Elad, and P. Milanfar, “LIDIA: Lightweight learned image denoising with instance adaptation,” in *Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2020, pp. 2220–2229.
- [5] K. Zhang, Y. Li, W. Zuo, L. Zhang, L. Van Gool, and R. Timofte, “Plug-and-Play image restoration with deep denoiser prior,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 10, pp. 6360–6376, 2022.
- [6] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, and M. Yang, “Restormer: Efficient transformer for high-resolution image restoration,” in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 5718–5729.
- [7] K. Zhang, Y. Li, J. Liang, J. Cao, Y. Zhang, H. Tang, D.-P. Fan, R. Timofte, and L. V. Gool, “Practical blind image denoising via Swin-Conv-UNet and data synthesis,” *Machine Intelligence Research*, 2023.
- [8] J. Liang, J. Cao, G. Sun, K. Zhang, L. Van Gool, and R. Timofte, “SwinIR: Image restoration using Swin Transformer,” in *International Conference on Computer Vision Workshops (ICCVW)*, 2021, pp. 1833–1844.
- [9] X. Mao, C. Shen, and Y.-B. Yang, “Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections,” in *Advances in Neural Information Processing Systems (NIPS)*, vol. 29, 2016.
- [10] Y. Chen and T. Pock, “Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1256–1272, 2017.
- [11] P. Liu, H. Zhang, K. Zhang, L. Lin, and W. Zuo, “Multi-level wavelet-CNN for image restoration,” in *Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018, pp. 773–782.
- [12] D. Liu, B. Wen, Y. Fan, C. C. Loy, and T. S. Huang, “Non-local recurrent network for image restoration,” in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 31, 2018.
- [13] T. Plötz and S. Roth, “Neural nearest neighbors networks,” in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 31, 2018.
- [14] S. Lefkimmiatis, “Non-local color image denoising with convolutional neural networks,” in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 3587–3596.
- [15] J. Xiao, X. Fu, M. Zhou, H. Liu, and Z.-J. Zha, “Random shuffle transformer for image restoration,” in *International Conference on Machine Learning (ICML)*, 2023, pp. 38 039–38 058.
- [16] J. Xiao, X. Fu, F. Wu, and Z.-J. Zha, “Stochastic window transformer for image restoration,” in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, 2022, pp. 9315–9329.
- [17] L. Kong, J. Dong, J. Ge, M. Li, and J. Pan, “Efficient frequency domain-based transformers for high-quality image deblurring,” in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 5886–5895.
- [18] H. Bai, S. Cheng, J. Tang, and J. Pan, “Learning a cascaded non-local residual network for super-resolving blurry images,” in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 223–232.
- [19] D. Martin, C. Fowlkes, D. Tal, and J. Malik, “A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics,” in *International Conference on Computer Vision (ICCV)*, vol. 2, 2001, pp. 416–423 vol.2.
- [20] S. Herbretreau and C. Kervrann, “Towards a unified view of unsupervised non-local methods for image denoising: the NL-Ridge approach,” in *IEEE International Conference on Image Processing (ICIP)*, 2022, pp. 3376–3380.
- [21] M. Lebrun, A. Buades, and J. M. Morel, “A nonlocal Bayesian image denoising algorithm,” *SIAM Journal on Imaging Sciences*, vol. 6, no. 3, pp. 1665–1688, 2013.
- [22] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Image denoising by sparse 3-D transform-domain collaborative filtering,” *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, 2007.
- [23] S. Gu, L. Zhang, W. Zuo, and X. Feng, “Weighted nuclear norm minimization with application to image denoising,” in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 2862–2869.
- [24] H. Liu, X. Liu, J. Lu, and S. Tan, “Self-supervised image prior learning with GMM from a single noisy image,” in *International Conference on Computer Vision (ICCV)*, 2021, pp. 2825–2834.
- [25] J. Xu, L. Zhang, and D. Zhang, “A trilateral weighted sparse coding scheme for real-world image denoising,” in *European Conference on Computer Vision (ECCV)*, 2018, pp. 20–36.
- [26] Y. Jo, S. Y. Chun, and J. Choi, “Rethinking Deep Image Prior for denoising,” in *International Conference on Computer Vision (ICCV)*, 2021, pp. 5087–5096.
- [27] Y. Quan, M. Chen, T. Pang, and H. Ji, “Self2Self with dropout: Learning self-supervised denoising from single image,” in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 1887–1895.
- [28] J. Lehtinen, J. Munkberg, J. Hasselgren, S. Laine, T. Karras, M. Aittala, and T. Aila, “Noise2Noise: Learning image restoration without clean data,” in *International Conference on Machine Learning (ICML)*, vol. 80, 2018, pp. 2965–2974.
- [29] N. Moran, D. Schmidt, Y. Zhong, and P. Coody, “Noisier2Noise: Learning to denoise from unpaired noisy data,” in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 12 061–12 069.
- [30] T. Pang, H. Zheng, Y. Quan, and H. Ji, “Reconstructed-to-Reconstructed: Unsupervised deep learning for image denoising,” in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 2043–2052.
- [31] A. Krull, T.-O. Buchholz, and F. Jug, “Noise2Void - Learning denoising from single noisy images,” in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 2124–2132.
- [32] S. Laine, T. Karras, J. Lehtinen, and T. Aila, “High-quality self-supervised deep image denoising,” in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 32, 2019.
- [33] J. Batson and L. Royer, “Noise2Self: Blind denoising by self-supervision,” in *International Conference on Machine Learning (ICML)*, vol. 97, 2019, pp. 524–533.
- [34] M. Zhussip, S. Soltanayev, and S. Y. Chun, “Training deep learning based image denoisers from undersampled measurements without ground truth and without image prior,” in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 10 255–10 264.
- [35] V. Lempitsky, A. Vedaldi, and D. Ulyanov, “Deep image prior,” in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 9446–9454.
- [36] Y. Mansour and R. Heckel, “Zero-Shot Noise2Noise: Efficient image denoising without any data,” in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 14 018–14 027.
- [37] J. Lequyer, R. Philip, A. Sharma, W.-H. Hsu, and L. Pelletier, “A fast blind zero-shot denoiser,” *Nature Machine Intelligence*, vol. 4, no. 11, pp. 953–963, 2022.
- [38] W. Dong, L. Zhang, G. Shi, and X. Li, “Nonlocally centralized sparse representation for image restoration,” *IEEE Transactions on Image Processing*, vol. 22, no. 4, pp. 1620–1630, 2013.
- [39] W. Dong, G. Shi, and X. Li, “Nonlocal image restoration with bilateral variance estimation: A low-rank approach,” *IEEE Transactions on Image Processing*, vol. 22, no. 2, pp. 700–711, 2013.

- [40] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Transactions on Image Processing*, vol. 15, no. 12, pp. 3736–3745, 2006.
- [41] C. Kervrann, "PEWA: Patch-based exponentially weighted aggregation for image denoising," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 27, 2014.
- [42] Q. Jin, I. Grama, C. Kervrann, and Q. Liu, "Nonlocal means and optimal weights for noise removal," *SIAM Journal on Imaging Sciences*, vol. 10, no. 4, pp. 1878–1920, 2017.
- [43] M. Delbracio and P. Milanfar, "Inversion by direct iteration: An alternative to denoising diffusion for image restoration," *arXiv preprint arXiv:2303.11435*, 2023.
- [44] J. Salmon and Y. Strozeccki, "From patches to pixels in non-local methods: Weighted-average reprojction," in *IEEE International Conference on Image Processing (ICIP)*, 2010, pp. 1929–1932.
- [45] M. Lebrun, M. Colom, A. Buades, and J.-M. Morel, "Secrets of image denoising cuisine," *Acta Numerica*, vol. 21, no. 4, pp. 475 – 576, 2012.
- [46] S. Herbreteau and C. Kervrann, "DCT2net: an interpretable shallow CNN for image denoising," *IEEE Transactions on Image Processing*, vol. 31, pp. 4292–4305, 2022.
- [47] S. Mohan, J. L. Vincent, R. Manzorro, P. Crozier, C. Fernandez-Granda, and E. Simoncelli, "Adaptive denoising via GainTuning," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 34, 2021, pp. 23 727–23 740.
- [48] T. Blu and F. Luisier, "The SURE-LET approach to image denoising," *IEEE Transactions on Image Processing*, vol. 16, no. 11, pp. 2778–2786, 2007.
- [49] D. Van De Ville and M. Kocher, "SURE-based Non-Local Means," *IEEE Signal Processing Letters*, vol. 16, no. 11, pp. 973–976, 2009.
- [50] Y.-Q. Wang and J.-M. Morel, "SURE guided Gaussian mixture image denoising," *SIAM Journal on Imaging Sciences*, vol. 6, no. 2, pp. 999–1034, 2013.
- [51] A. Krishnamoorthy and D. Menon, "Matrix inversion using Cholesky decomposition," in *Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA)*, 2013, pp. 70–72.
- [52] T. Hastie, R. Tibshirani, and J. H. Friedman, *The Elements of Statistical Learning: data mining, inference, and prediction*. Springer, 2009, vol. 2.
- [53] T. Plötz and S. Roth, "Benchmarking denoising algorithms with real photographs," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2750–2759.
- [54] J.-B. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 5197–5206.
- [55] J.-L. Starck, F. D. Murtagh, and A. Bijaoui, *Image Processing and Data Analysis: the multiscale approach*. Cambridge University Press, 1998.
- [56] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 32, 2019.