Patch Reprojections for Non Local Methods

Salmon Joseph[a], Yann Strozecki[b]

[a]*LPMA / UMR 7599*
Université Paris Diderot Paris 7
Boîte courrier 7012 75251 PARIS Cedex 05 FRANCE
tel: +33 (0) 1 44 27 68 67
[b] *Équipe de Logique Mathématique*
Université Paris Diderot Paris 7
UFR de mathématiques case 7012
site Chevaleret 75205 PARIS Cedex 13 FRANCE
tel: +33 (0)1 44 27 56 14

Abstract

Since their introduction in image denoising, the family of non-local methods, whose Non-Local Means (NL-Means) is the most famous member, has proved its ability to challenge other powerful methods such as wavelet based approaches or variational techniques. Though simple to implement and efficient in practice, the classical NL-Means algorithm suffers from several limitations: noise artifacts are created around edges and regions with few repetitions in the image are not treated at all. In this paper, we present an easy to implement and time efficient modification of the NL-Means based on a better reprojection from the patches space to the original pixel space, specially designed to reduce the artifacts due to the *rare patch effect*. We compare the performance of several reprojection schemes on a toy example and on some classical natural images.

Keywords: Image denoising, Image restoration, Image edge analysis, Statistics, Non-Local Means
2000 MSC: 94A08 68U10 62G99

Email addresses: `salmon att math.jussieu.fr` (Salmon Joseph), `strozecki att logique.jussieu.fr` (Yann Strozecki)

URL: `http://www.math.jussieu.fr/~salmon` (Salmon Joseph), `http://www.logique.jussieu.fr/~strozecki/` (Yann Strozecki)

Preprint submitted to Signal Processing *August 9, 2011*

1. Introduction

In recent years, major progresses in image denoising have been recorded using patch-based methods. These approaches take advantage of the presence of a relatively large number of similar small sub-images, or *patches*, in natural pictures. Noise is then removed by averaging those similar patches.

To our knowledge, the first patch-based procedure was introduced in image processing by Efros and Leung [1] for texture synthesis, and then extended to inpainting in [2]. The introduction of patch-based methods in the context of image restoration is due to Buades *et al.* [3, 4, 5] (and independently to Awate and Whitaker [6]). The algorithm they developed, named Non-Local Means (NL-Means), and its variants based on Lepski's method [7, 8] give some of the best results in image denoising. These methods rely on the self-similarity of natural images. They outperform other standard approaches derived either from wavelets methods [9, 10] such as curvelets [11] and bandlets [12], or from variational methods [13, 14]. The wavelet type methods model the image as a sparse combination of some vectors from basis or frames adapted to image representation. They take advantage of the sparsity of the representation for fast implementation [15]. In variational methods one assumes that the original image is a smooth function. The image is recovered by minimizing the sum of a term representing the fidelity to the observed image and of a regularity constraint on the absolute norm of the gradient of the image.

To our knowledge, the two state-of-the-art methods in denoising [16, 17], though more sophisticated, are patch oriented, which emphasizes the importance of patch-based methods. The first method, called BM3D [16], reconstructs the image by first finding similar patches, then stacking those in a 3D signal, and eventually denoising the block by a classical wavelet thresholding procedure. In [17], the approach is the same, but instead of a fixed wavelet dictionary, the authors use a learned dictionary of patches and approximate the block using ℓ^0 or ℓ^1 regularization [18, 19, 20].

The principle of the NL-Means algorithm is the following (cf. also Fig.1). First, an image is transformed in a collection of patches. Then, an estimate of each patch in the image is provided by taking into account the similarity between patches. The patch estimates are weighted average of the noisy patches. The weights are controlled by the similarity between the target patch (the one to be denoised) and the other available candidate patches. Once this is done, one reprojects the information obtained from the patches

$$\begin{tikzcd}
\text{PATCH DOMAIN} & \left(\mathsf{P}^I_{\mathbf{x}}\right)_{\mathbf{x}\in\Omega} \arrow[r, "\text{Patches Denoising}"] & \left(\widehat{\mathsf{P}}^I_{\mathbf{x}}\right)_{\mathbf{x}\in\Omega} \arrow[d, "\text{Reprojection}"] \\
\text{PIXEL DOMAIN} & \left(I(\mathbf{x})\right)_{\mathbf{x}\in\Omega} \arrow[u, "\text{Patchization}"] & \left(\widehat{I}(\mathbf{x})\right)_{\mathbf{x}\in\Omega}
\end{tikzcd}$$

Figure 1: The denoising methods we consider can be decomposed in three steps: create the collection of patches (Patchization), process/denoise each patch of this collection (Patches Denoising), reproject the information in the original pixel domain (Reprojection). This paper is mainly about the last step.

to denoise the pixels themselves. This method can be interpreted as a kernel smoothing approach: in this context the NL-Means is a Nadaraya-Watson type estimator. In the original method [3], the kernel used is Gaussian but several authors propose to enlarge the variety of relevant kernels (see for instance [21]). We mainly focus on the flat kernel (cf. Fig.3) both for simplicity and computation speed.

Since the space of patches is of larger dimension than the original image space, there are many alternatives to recover pixel estimators from patch estimators. In this paper we investigate the two most popular reprojection procedures (cf. Sections 3.2 and 3.3) used to connect these spaces (cf. Fig.1) and we compare them with new ones. Among these new reprojections, we study a particularly efficient one called Weighted Average reprojection (WAV-reprojection). We show that this method both increases numerical performance, measured in PSNR (Peak Signal to Noise Ratio), and visual performance. This reprojection reduces the *halo of noise* (cf. Fig.4 and Fig.9) usually observed when applying Non Local methods. This phenomenon is due to the *rare patch effect*: some part of the image, such as the edges and the corners, contain little redundancy.

Another recently proposed idea consists in considering more general shapes such as disks, bands, pies, etc. instead of common square patches [22, 23]. This idea can be related to the reprojections we present, a shape being here a decentered square patch.

We also highlight the importance of using a multi-scale approach when using patch based methods. Better performance are reached by using patch sizes growing according to the noise level. Common patch sizes range from 7×7 in [3] for low level of noise up to 16×16 in [17] for strong noise. However with those relatively large patches, denoising "textons" (elementary motifs) of small size is impossible. So it is important to use at least a second smaller size to correctly handle parts of images made of small textons.

The rest of the paper is organized as follows: in Section 2 we introduce the framework and the original NL-Means method. In Section 3 we detail how to propagate the information obtained in the patches space into the pixels space. Among other methods, we investigate the WAV-reprojection already considered in a previous work of the authors [24]. Section 4 gives some insights on the importance of the reprojection step, from a practical, theoretical and computational point of view. In Section 5, we show the impact of using different sizes of patch while Section 6 ends the paper with numerical experiments comparing several reprojections methods on classical images and on a set of high-quality images.

2. Classical definition of the NL-Means

2.1. Framework, noise model and notations

In this paper, we are concerned with the problem of the restoration of noisy images. We assume that we are given a gray-scale image I which is a noisy version of an unobservable image $I^\star$. An image is defined as a real valued function from a domain (or index) Ω. In the sequel, the images are supposed to be rectangular of size $M \times N$. It means that the index set $\Omega \subset \mathbb{Z}^2$ can be chosen as $[\![0, M-1]\!] \times [\![0, N-1]\!]$, where the notation $[\![n_1, n_2]\!]$ stands for the integers interval $\{n_1, n_1+1, \cdots, n_2\}$. We sometime identify an $M \times N$ image I with the collection of pixel values, meaning $I = \big(I(\mathbf{x}), \mathbf{x} \in [\![0, M-1]\!] \times [\![0, N-1]\!]\big)$.

In this article we only deal with the additive white Gaussian noise:

$$I(\mathbf{x}) = I^\star(\mathbf{x}) + \varepsilon(\mathbf{x}) \,, \tag{1}$$

where $\mathbf{x} = (x, y) \in \Omega$ is any pixel in the image domain Ω and ε is a centered Gaussian noise with known variance σ^2. Traditional neighborhood filters treat model (1) as a problem of estimating a two-dimensional regression function that can be handled by various smoothing techniques (see e.g. [25] and the references therein).

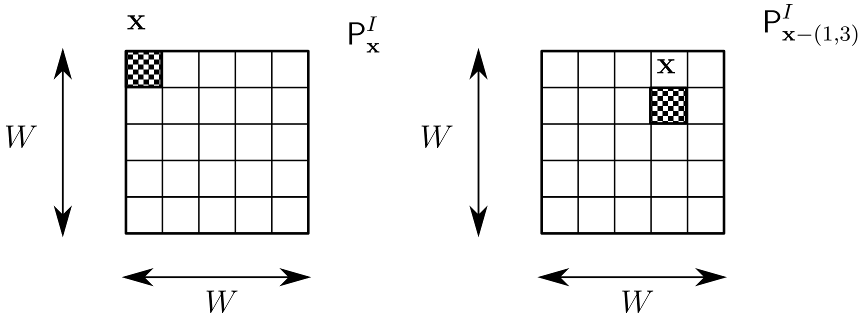

$\mathbf{x}$
$\mathsf{P}^I_{\mathbf{x}}$
W
W
$\mathsf{P}^I_{\mathbf{x}-(1,3)}$
$\mathbf{x}$
W
W

Figure 2: Illustration of a patch $\mathsf{P}^I_{\mathbf{x}}$ (with the convention that a patch is indexed by its upper left corner) and its slided/shifted version by the vector $\boldsymbol{\delta} = (1,3)$. In this example $W = 5$.

In [3] the approach is to replace the spatial regression by the regression of the pixel value on the values of its neighbors. This extends what is done for Bilateral Filtering [26] or for the Yaroslavsky's Filter [27] to the space of patches that we now define. Let $W = 2W_1 + 1$ (with $W_1 \in \mathbb{N}$) be some odd integer, then for some pixel $\mathbf{x} \in \Omega$, the patch $\mathsf{P}^I_{\mathbf{x}}$ is the sub-image of I having $\mathbf{x}$ as upper left corner (cf. Fig.2 for an illustration). For any pixel $\mathbf{x} \in \Omega$ (we omit the details for border pixels), one has

$$\mathsf{P}^{I,W}_{\mathbf{x}} = \mathsf{P}^I_{\mathbf{x}} = (I(\mathbf{x}+\boldsymbol{\tau}), \boldsymbol{\tau} \in [\![0, W-1]\!]^2), \tag{2}$$

where the exponent W is omitted when the patch width is fixed and that no confusion can occur. In the following, we will use the notation $\mathsf{P}^I_{\mathbf{x}-\boldsymbol{\delta}}$ to represent the patch having the pixel $\mathbf{x} - \boldsymbol{\delta}$ as upper left corner. We will have $\boldsymbol{\delta} \in [\![0, W-1]\!]^2$, therefore the pixel $\mathbf{x}$ will be in $\mathsf{P}^I_{\mathbf{x}-\boldsymbol{\delta}}$. The parameter $\boldsymbol{\delta}$ can be seen as a sliding or shifting parameter that is the patch is translated by $\boldsymbol{\delta}$.

Now, let us present the definition of the classical NL-Means estimator. For estimating the value $I^\star(\mathbf{x})$, a non-parametric regression estimation is carried out taking as response $I(\mathbf{x})$ and as covariate the patch $\mathsf{P}^I_{\mathbf{x}-\boldsymbol{\delta}_W}$ (centered on $\mathbf{x}$), where $\boldsymbol{\delta}_W = (W_1, W_1)$. Note that our unusual notation aims at simplifying the writing of the new reprojection methods introduced in Section 3.

Using kernel smoothing with patches, the NL-Means estimator of the pixel $\mathbf{x}$ can be written as

$$\hat{I}_{\mathrm{NLM}}(\mathbf{x}) = \frac{\displaystyle\sum_{\mathbf{x}' \in \Omega} K(\|\mathsf{P}^I_{\mathbf{x}-\boldsymbol{\delta}_W} - \mathsf{P}^I_{\mathbf{x}'-\boldsymbol{\delta}_W}\|/h) \cdot I(\mathbf{x}')}{\displaystyle\sum_{\mathbf{x}'' \in \Omega} K(\|\mathsf{P}^I_{\mathbf{x}-\boldsymbol{\delta}_W} - \mathsf{P}^I_{\mathbf{x}''-\boldsymbol{\delta}_W}\|/h)} , \tag{3}$$

where $\mathbf{x}'$ runs in Ω, K is a kernel function, $h > 0$ is the bandwidth parameter and $\|\cdot\|$ is a norm on $\mathbb{R}^{W^2}$. One restricts the summation in (3) to a searching zone $\Omega_R(\mathbf{x}) = \mathbf{x} + [\![-\frac{R-1}{2}, \frac{R-1}{2}]\!]^2$, a square centered on $\mathbf{x}$ of size $R \times R$ with R odd. This is done to avoid the violation of the stationarity assumption required to prove the convergence of the method [4]. Remark that substituting $\Omega_R(\mathbf{x})$ to Ω is also done to speed up the algorithm since the complexity of the naive algorithm decreases from $O(N^2M^2W^2)$ to $O(R^2NMW^2)$ for an image of size $M \times N$.

Defining the unnormalized NL-Means weights as

$$\theta^I_{\mathrm{NLM}}(\mathbf{x}, \mathbf{x}') = K(\|\mathsf{P}^I_{\mathbf{x}} - \mathsf{P}^I_{\mathbf{x}'}\|/h) , \tag{4}$$

we can then recast this estimator as a simple weighted average of the noisy pixels in the searching zone

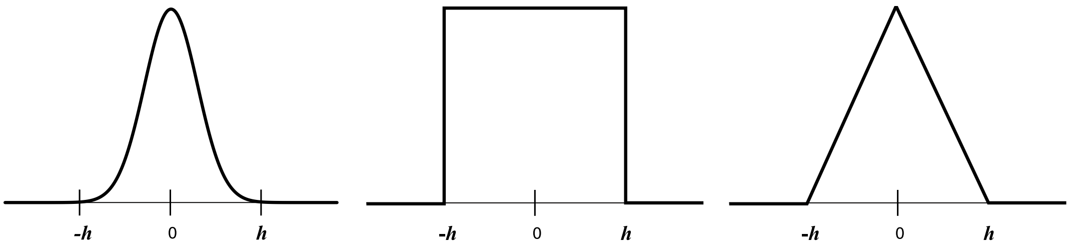
$$\hat{I}_{\mathrm{NLM}}(\mathbf{x}) = \sum_{\mathbf{x}' \in \Omega_R(\mathbf{x})} \frac{\theta^I_{\mathrm{NLM}}(\mathbf{x} - \boldsymbol{\delta}_W, \mathbf{x}' - \boldsymbol{\delta}_W)}{\displaystyle\sum_{\mathbf{x}'' \in \Omega_R(\mathbf{x})} \theta^I_{\mathrm{NLM}}(\mathbf{x} - \boldsymbol{\delta}_W, \mathbf{x}'' - \boldsymbol{\delta}_W)} \cdot I(\mathbf{x}') . \tag{5}$$

Different variants of NL-means estimators are obtained by changing the kernel functions K [21] or by changing the way the similarity between patches is measured. Default choices are the Gaussian kernel $K(x) = e^{-x^2}$ and the Euclidean norm denoted by $\|\cdot\|_2$. Other alternatives use PCA variants both to improve computation time [28, 29, 30, 31] and quality [32].

2.2. Choosing the kernel

The choice of the kernel should not drastically alter the performance of the method and h is a more crucial parameter [33]. However, there are several reasons why one should use a non-Gaussian kernel and especially a kernel with a compact support.

First a kernel with a compact support reduces a problem encountered by the classical NL-Means when the searching zone is too big [34]. Since



(a) Gaussian Kernel (b) Flat Kernel (c) Triangular Kernel

Figure 3: Three types of Kernel and their bandwidth h (scales are different)

many coefficients $\theta^I_{\mathrm{NLM}}(\mathbf{x}, \mathbf{x}')$ are close but not equal to zero, they decrease the impact of the good candidates, i.e., candidates with large $\theta^I_{\mathrm{NLM}}(\mathbf{x}, \mathbf{x}')$. Thus hard thresholding the small coefficients in (5) with a compact support kernel robusitifies the estimator and improves the quality of the procedure. In fact this phenomenon also limits the interest of using a large R, as noted by Tasdizen in [31]. He termed the method with a small searching window the *limited-range* implementation of the NL-Means. We would rather term it "Semi-Local Means" (see [35] where this name is first used), as no gain is obtained in practice with too large R. So, a small R (usually $10 \leq R \leq 30$) should not only be chosen to speed up the algorithm, but also to increase the performance on natural images.

Another drawback [36, 37, 34] of the classical NL-Means is that the weight of the central patch, i.e., $\theta^I_{\mathrm{NLM}}(\mathbf{x}, \mathbf{x})$, is overestimated. Indeed, one always compares the central patch with itself when $\mathbf{x} = \mathbf{x}'$ in (4), leading to a zero distance between patches. Thus whatever the kernel and the distance are, one has $\theta^I_{\mathrm{NLM}}(\mathbf{x}, \mathbf{x}) = K(0)$. Since the kernel K is usually chosen symmetric and non-increasing on $[0, +\infty)$, for any $\mathbf{x}, \mathbf{x}'$ the following inequality holds: $\theta^I_{\mathrm{NLM}}(\mathbf{x}, \mathbf{x}) \geq \theta^I_{\mathrm{NLM}}(\mathbf{x}, \mathbf{x}')$, and the gap might be big. To fix this problem, Buades et. al. proposed in their seminal paper [3] to assign to the weight of the central pixel the same value as the maximum weight calculated for the other pixels in the searching zone : $\theta^I_{\mathrm{NLM}}(\mathbf{x}, \mathbf{x}) = \max_{\mathbf{x}' \neq \mathbf{x}} \theta^I_{\mathrm{NLM}}(\mathbf{x}, \mathbf{x}')$. Thus, in regions with little redundancy (corner shapes for instance) this leads to only average the closest candidates with the central one. It was explained that this drawback can be overcome by using an unbiased estimator for the risk [34] or by using a k-nearest neighbors strategy [36]. Another alternative proposed in [38] is to use the maximum property only if the weight of the closest patch is big enough. However, with the flat kernel this problem does not occur since all good candidates have the same weights.

Now, let us justify that the weights should be almost uniform among the good candidates. Consider the *Oracle estimator* one would get using the true support $\Omega_{OR}(\mathbf{x}) = \{\mathbf{x}' \in \Omega_R(\mathbf{x}), \mathsf{P}_{\mathbf{x}}^{I^\star} = \mathsf{P}_{\mathbf{x}'}^{I^\star}\}$ instead of $\Omega_R(\mathbf{x})$ in (5) (see for instance [39] for more details on oracle estimators). Clearly, this is not an estimator since the oracle support $\Omega_{OR}(\mathbf{x})$ is unknown for the observer. Yet, the risk of the oracle estimator is an upper bound of the best performance one could achieve with the NL-Means. It also reflects the true spirit of the method: one should find the geometry of the underlying image and then average the noisy observations according to the estimated similarity. The oracle estimator is unbiased, so we need to use the weights $\theta^I(\mathbf{x}, \mathbf{x}')$ minimizing the variance in order to minimize its risk. Now, among weighted average estimators, the one with uniform weights for every pixel in $\Omega_{OR}(\mathbf{x})$, i.e., $\forall \mathbf{x}' \in \Omega_{OR}(\mathbf{x})$, $\theta^I(\mathbf{x}, \mathbf{x}') = 1/|\Omega_{OR}(\mathbf{x})|$ has the smallest variance.

From the statistical analysis of kernel smoothing methods, no result favors any kernel in all situations. Knowing the regularity of the underlying function to estimate, a kernel with the corresponding number of vanishing moments should be used. But in general, we do not know in advance the underlying regularity in the space of patches.

A practical solution consists in choosing the kernel leading to the fastest implementation. That is why we focus on the flat kernel described in Fig.3 (see also Section 3.6).

Another advantage of the flat kernel is that it leads to an easy statistical selection of the bandwidth h, by controlling a χ^2 variable as in [5]. The parameter h is the most crucial tuning parameter in the NL-Means method and is usually selected either manually or by cross-validation on a few images. It might be interesting to determine an automatic data driven selection rule. Two different approaches are proposed to select h for the Gaussian kernel case. Boulanger and Kervrann [7] apply the Lepskii's method [40] to find both a good searching zone and a good bandwidth. Van de Ville and Kocher [33] and Duval et al. [41] use a method based on the SURE (Stein Unbiased Risk Estimate) theory: they minimize the estimated risk of the procedure with regards to h on a discrete grid. Those methods provide good results in practice, though they increase the computation time. Indeed, the NL-Means algorithm should be run for several values of R or h depending on the method. However, for the flat kernel, we propose a simpler and faster solution.

Indeed one can test whether two patches $\mathsf{P}_{\mathbf{x}}^I$ and $\mathsf{P}_{\mathbf{x}'}^I$ "are independent noisy observations of the same original patch" under the assumption of Gaussian noise (neglecting covariances due to patches overlap). If two original patches are equal, $\|\mathsf{P}^I_{\mathbf{x}} - \mathsf{P}^I_{\mathbf{x}'}\|^2/2\sigma^2$ is $\chi^2(W^2)$ distributed. Thus one can accept with probability greater than $1-\alpha$ the hypothesis that the patches $\mathsf{P}^{I^*}_{\mathbf{x}}$ and $\mathsf{P}^{I^*}_{\mathbf{x}'}$ are equal if $\|\mathsf{P}^I_{\mathbf{x}} - \mathsf{P}^I_{\mathbf{x}'}\|^2/2\sigma^2 \leq q^{W^2}_{1-\alpha}$, where $q^{W^2}_{1-\alpha}$ is the $(1-\alpha)$-quantile of the $\chi^2(W^2)$ distribution. In practice, we used $\alpha = 0.01$ for our simulations on natural images. This leads to a quasi-automatic setting for the choice of the bandwidth: $h^2 = 2\sigma^2 q^{W^2}_{0.99}$.

3. Reprojection from the patches space

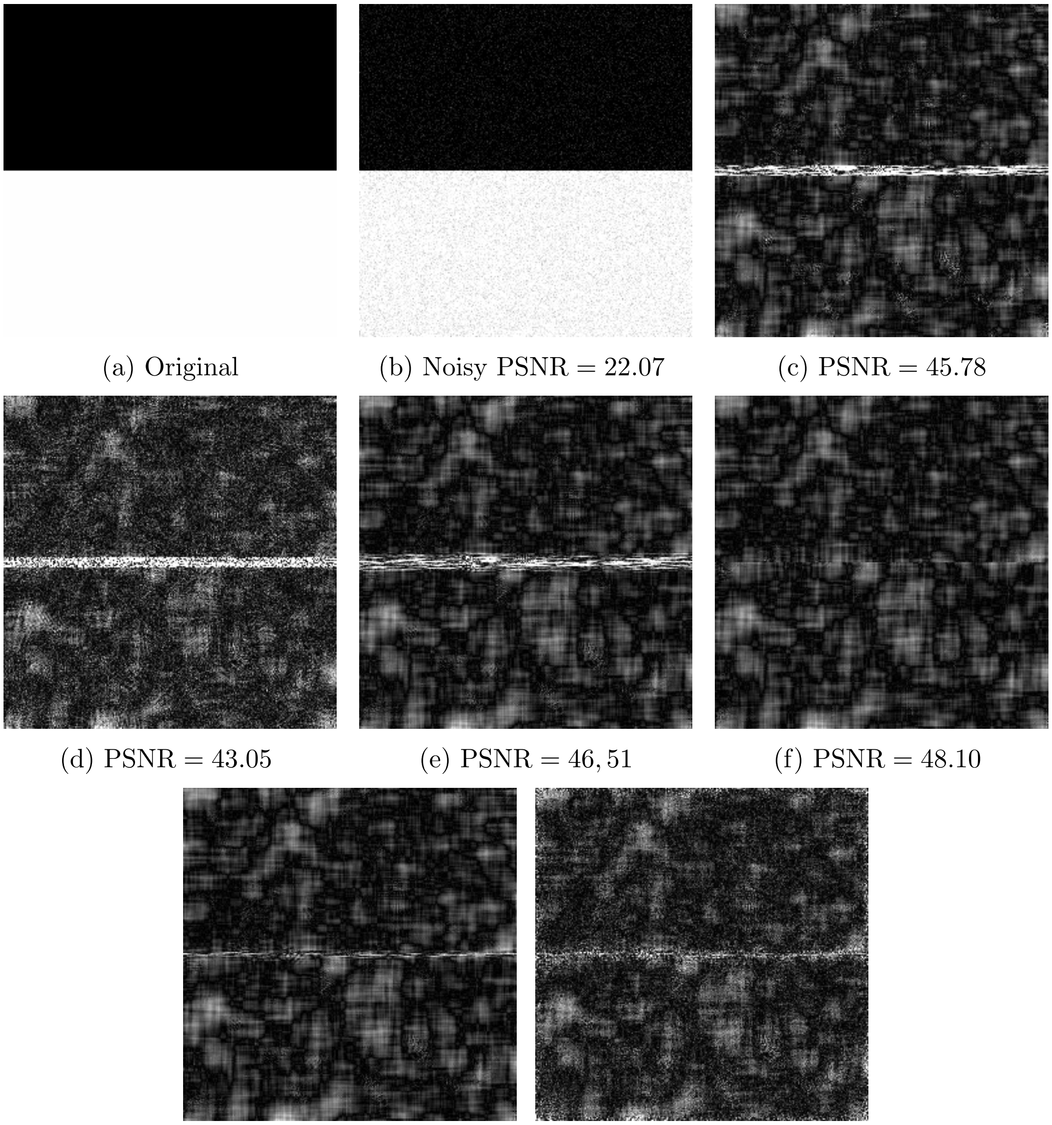
In this section we present several methods to reproject information from patches to pixels. The first two methods (Sections 3.2 and 3.3) present the reprojections already explored in [3] and in [7]. In Sections 3.4, 3.5 and 3.6, we propose new methods and detail why the WAV-reprojection should be used.

3.1. The rare patch effect

The NL-Means method achieves very good results on uniform region, where using the mean to recover the image is well asymptotically justified, according to the law of large numbers. But when the number of similar patches is small in some region (a corner or an edge for instance), one faces the *rare patch effect* [41]. This phenomenon leads for instance to degrading fine details in some complex textures.

Along edges NL-Means methods usually perform poorly (cf. Fig.4-c, 4-d, 4-e) and we observe a *halo of noise*. This noise occurs because the patches centered on the edge may only have a few similar patches. More precisely, the number of similar patches is linear in R in that case, whereas it is quadratic in R for constant regions.

We illustrate this effect with the "Jump" image (Fig.4-a) with $N = M = 256$, pixel values equal to 0 or 255, corrupted with $\sigma = 20$. On this example, it is easy to explain why the NL-Means method fails and the *halo of noise* is particularly visible. This image is also considered in [5] and in [42]. In the latter paper, the authors visualize the space of patches of a simple 1-D signal: a jump (a Heaviside type function). In this case the patches are of width $W = 2$ and can be written $\mathsf{P}^I_{\mathbf{x}} = (I(\mathbf{x}), I(\mathbf{x}+1))$. It is shown that the use of patches larger than 1×1 on such an image limits the number of misidentifications done by the method. Though, for a 1-D signal no problem



(a) Original (b) Noisy PSNR = 22.07 (c) PSNR = 45.78

(d) PSNR = 43.05 (e) PSNR = 46, 51 (f) PSNR = 48.10

(g) PSNR = 47.77 (h) PSNR = 46.15

Figure 4: Variant of NL-Means denoising with $R = 9, W = 9$, $\sigma = 20$, (a) original image, (b) noisy image. The following images are absolute difference between the original and the one denoised with (c) Flat kernel and central-reprojection, (d) Classical NL-Means $h = \sigma$, (e) Flat kernel and UAE-reprojection, (f) Flat kernel and Min-reprojection, (g) Flat kernel and WAV-reprojection, (h) two patch sizes $W_{\max} = 9, W_{\min} = 2$, $h^2_{\max} = 2\sigma^2 q_{0.99}^{W^2_{\max}}, h^2_{\min} = 2\sigma^2 q_{0.75}^{W^2_{\min}}$ and $h^2 = 2\sigma^2 q_{0.99}^{W^2}$ when the flat kernel is used.

occurs around the edge: there is only one patch that can mix values of the two regions. Now, we consider the image (a 2-D signal) of a simple jump as in Fig.4-b, and we focus on the denoising of a pixel on the edge. In Fig.5-c, we use patches of size 2 × 1 (vertical orientation) to visualize the repartition of the patches in the searching zone for such a pixel. It is obvious that a new problem emerges in 2-D: a third cluster appears, and the target patch is in the one with fewer candidates, corresponding to bi-color patches. Such a cluster does not exist in 1-D (see [42], Fig.3). In our context this means that denoising will be done by averaging very few pixels, thus leading to degraded performance. Worst, when the patch size is $W \times W$, $W + 1$ clusters emerge in $\mathbb{R}^{W^2}$, degrading performance in a band of size $W - 1$ near the edge.

We visualize the artifacts produced using the NL-Means on the "Jump" image in Fig.4-c by showing the absolute difference between the original and the denoised image. As predicted one can observe a band of noise of width $W - 1$ around the edge. Note also that this halo of noise does not depend on the value of h whether chosen globally or locally.

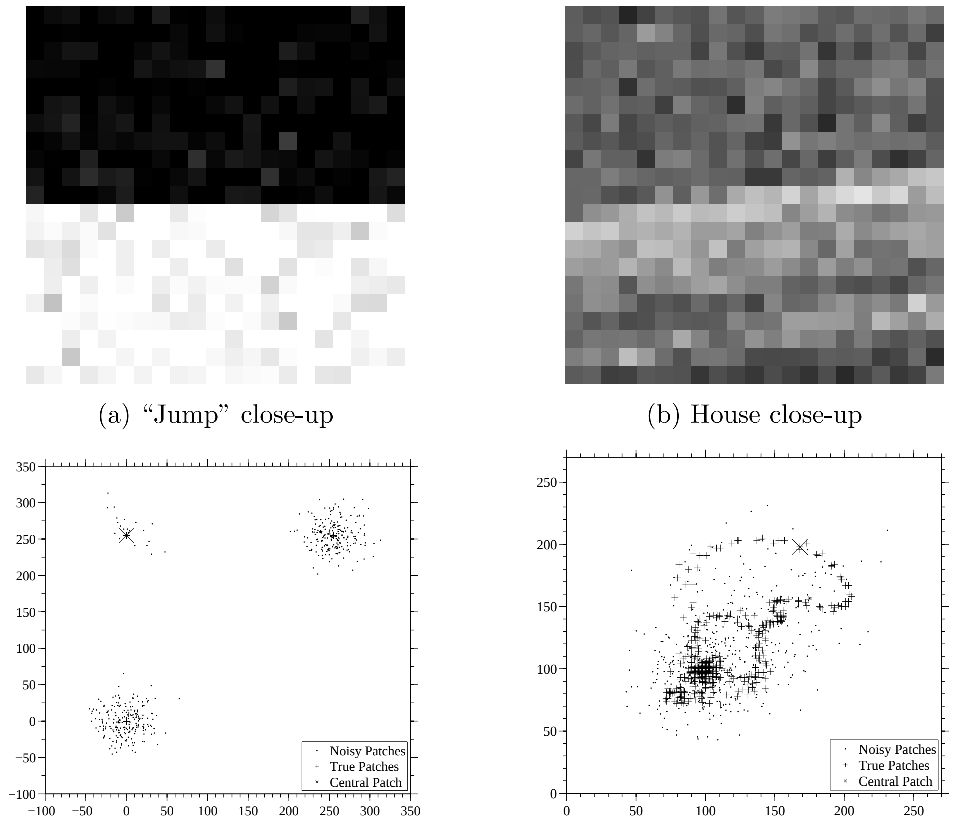
To solve this problem, one should more carefully take into account the information obtained on every patch of the image. Remind that patches are overlapping and every pixel belongs to W^2 patches. With this in mind, one can rewrite the NL-Means estimator in two parts. On the one hand, one determines a non-parametric estimator for every patch in the image using a weighted average with the same weights as in (4). This gives for each patch $\mathsf{P}^I_{\mathbf{x}}$ an estimator

$$\widehat{\mathsf{P}}^I_{\mathbf{x}} = \sum_{\mathbf{x}' \in \Omega_R(\mathbf{x})} \frac{\theta^I_{\mathrm{NLM}}(\mathbf{x}, \mathbf{x}') \cdot \mathsf{P}^I_{\mathbf{x}'}}{\displaystyle\sum_{\mathbf{x}'' \in \Omega_R(\mathbf{x})} \theta^I_{\mathrm{NLM}}(\mathbf{x}, \mathbf{x}'')} \,. \tag{6}$$

On the other hand, one needs to obtain a final estimate $\hat{I}(\mathbf{x})$ for each pixel $\mathbf{x}$ in the domain Ω. Due to overlaps, each pixel in the image belongs to W^2 patches (see for instance two such patches in Fig.7). This means that we have for each pixel W^2 estimates we can combine to obtain better final pixels estimators. This aggregation process will be referred to as reprojection. This new step is the last one in our denoising process (cf. Fig.1).

Recall that for each pixel $\mathbf{x} \in \Omega$ and each sliding parameter $\boldsymbol{\delta} \in [\![0, W-1]\!]^2$ one has $\mathsf{P}^I_{\mathbf{x}-\boldsymbol{\delta}}(\boldsymbol{\delta}) = I(\mathbf{x})$. Therefore, the collection of W^2 estimators for $\hat{I}(\mathbf{x})$ is $\left(\widehat{\mathsf{P}}^I_{\mathbf{x}-\boldsymbol{\delta}}(\boldsymbol{\delta})\right)_{\boldsymbol{\delta} \in [\![0, W-1]\!]^2}$.

Now, one has many alternatives to reproject from the space of patches



(a) "Jump" close-up

(b) House close-up

(c) Patch clusters ("Jump")

(d) Patch clusters (House)

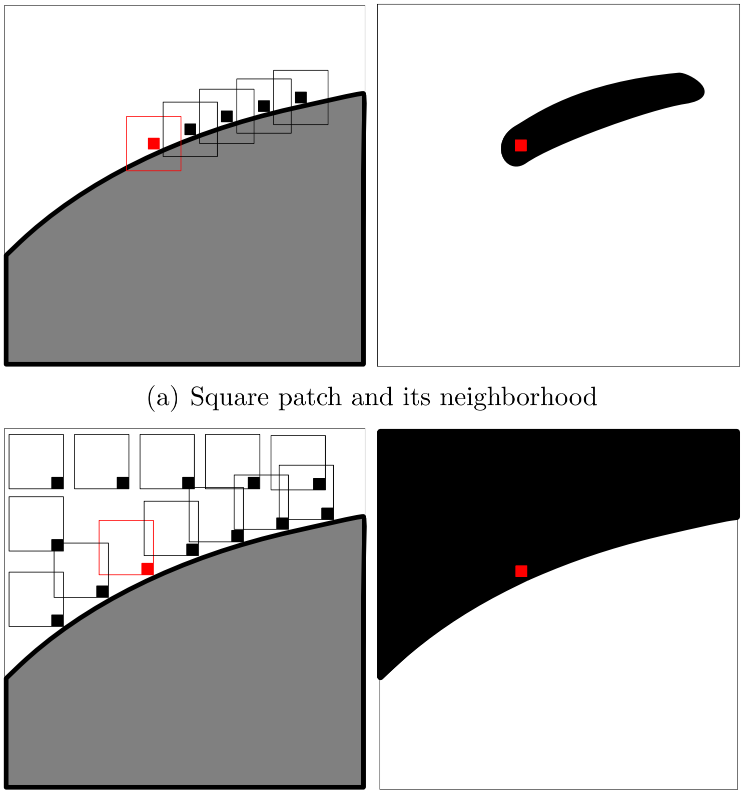
Figure 5: Top: (a) Searching zone for noisy 8-bits "Jump" image, (b) Searching zone for noisy 8-bits House image ($R = 21, \sigma = 20$). Bottom: corresponding scatter plot of the patches in the searching zone. The patches considered are vertical of size 2×1. Each axis in (c) and (d) corresponds to the value of one patch component.

to the pixels domain. One should notice that the way the candidates are aggregated can vary from pixel to pixel. Yet, the most commonly used reprojections proposed (cf. Sections 3.2 and 3.3) use the same aggregation scheme for all pixels, independently of their values or positions.

3.2. Central Reprojection

The original method [3] uses only the center of the denoised patch to estimate $I^*(\mathbf{x})$, leading to the estimator

$$\hat{I}_{\text{CENT}}(\mathbf{x}) = \widehat{\mathsf{P}}^{I}_{\mathbf{x}-\boldsymbol{\delta}_W}(\boldsymbol{\delta}_W) = \widehat{\mathsf{P}}^{I}_{\mathbf{x}-(W_1,W_1)}(W_1, W_1)\,. \tag{7}$$



(a) Square patch and its neighborhood

(b) Suitable decentered patch and its neighborhood

Figure 6: Examples of neighborhood associated with a square patch centered or conveniently decentered. The pixel of interest is in red and the selected pixels obtained by the two methods to denoise the red pixel are in black. In (a) the patch is centered and fewer similar patches candidates are found than in (b) with a decentered patch.

It is obvious that such a method loses an important part of the information provided by the patches. In particular, this reprojection suffers from the already mentioned *halo of noise*. We will show that one can gain a lot by using "sliding type" reprojections, i.e., reprojections taking into account the whole family $\left(\widehat{\mathsf{P}}^{I}_{\mathbf{x}-\boldsymbol{\delta}}(\boldsymbol{\delta})\right)_{\boldsymbol{\delta}\in[\![0,W-1]\!]^2}$ of estimators of $I^\star(\mathbf{x})$ (cf. Fig.7 for an example).

3.3. Uniform Average of Estimators reprojection (UAE)

A first improvement already proposed in [3, 43] is to average the different estimators of $I^*(\mathbf{x})$. Formally, the uniform average of estimator (UAE-reprojection estimator) is

$$\hat{I}_{\mathrm{UAE}}(\mathbf{x}) = \frac{1}{W^2} \sum_{\boldsymbol{\delta} \in [\![0, W-1]\!]^2} \widehat{\mathsf{P}}^{I}_{\mathbf{x}-\boldsymbol{\delta}}(\boldsymbol{\delta}) \ . \tag{8}$$

It already leads to important improvements in term of PSNR. Yet the *halo of noise* is still present in the "Jump" image (Fig.4-e) as in natural images (see the Cameraman's elbow in Fig 9-e).

3.4. Minimizing Variance-Reprojection

Now assume for a fixed $\boldsymbol{\delta} \in [\![0, W-1]\!]^2$ that the observed patches $\left(\mathsf{P}^{I}_{\mathbf{x}'-\boldsymbol{\delta}}\right)_{\mathbf{x}' \in \Omega_R(\mathbf{x})}$ used to define $\widehat{\mathsf{P}}^{I}_{\mathbf{x}-\boldsymbol{\delta}}$ satisfy the test "they are two noisy observations of the same original patch". This means that the selected patches can be considered of zero bias. Thus to minimize the expectation of the quadratic error, one only needs to choose the estimator with minimal variance. So, we define the Min-reprojection estimator as:

$$\hat{I}_{\mathrm{Min}}(\mathbf{x}) = \widehat{\mathsf{P}}^{I}_{\mathbf{x}-\hat{\boldsymbol{\delta}}}(\hat{\boldsymbol{\delta}}), \quad \text{with} \quad \hat{\boldsymbol{\delta}} = \underset{\boldsymbol{\delta} \in [\![0, W-1]\!]^2}{\arg\min} \ \mathrm{Var}\left(\widehat{\mathsf{P}}^{I}_{\mathbf{x}-\boldsymbol{\delta}}(\boldsymbol{\delta})\right) \ , \tag{9}$$

where $\mathrm{Var}\left(\widehat{\mathsf{P}}^{I}_{\mathbf{x}-\boldsymbol{\delta}}(\boldsymbol{\delta})\right)$ is the variance of $\widehat{\mathsf{P}}^{I}_{\mathbf{x}-\boldsymbol{\delta}}(\boldsymbol{\delta})$. The noise level σ^2 is known in our model, and if we assume that the patches behave as if they were independent, the variance in (9) equals

$$\sigma^2 \sum_{\mathbf{x}' \in \Omega_R(\mathbf{x})} \left(\theta^{I}_{\mathrm{NLM}}(\mathbf{x}, \mathbf{x}')\right)^2 / \Big(\sum_{\mathbf{x}'' \in \Omega_R(\mathbf{x})} \theta^{I}_{\mathrm{NLM}}(\mathbf{x}, \mathbf{x}'')\Big)^2 \ . \tag{10}$$

For the flat kernel, with $\mathbf{x}' = \mathbf{x} - \boldsymbol{\delta}$, the variance $\mathrm{Var}\left(\widehat{\mathsf{P}}^{I}_{\mathbf{x}'}(\boldsymbol{\delta})\right)$ is simply $\sigma^2 / \left|\left\{\mathbf{x}'' \in \Omega_R(\mathbf{x}'), \|\mathsf{P}^{I}_{\mathbf{x}'} - \mathsf{P}^{I}_{\mathbf{x}''}\| \leq h\right\}\right|$, where $|A|$ stands for the cardinal of A for any set A. This reprojection consists in selecting among the W^2 candidates the estimator based on the position that has the maximum number of similar patches in the searching zone.

Even though the improvement is quite important on the "Jump" image (cf. Fig.4-f), this type of reprojection performs poorly on natural images. The transition between areas having many similar patches and areas with fewer ones are too brutal. Hence edges are badly treated and appear crenelated (cf. Fig.9-f).

3.5. Minimizing Variance with Weighted Average Reprojection (WAV)

Let us now introduce a more refined method especially designed to reduce the *halo of noise*. The choice of weights based on variance criteria is made more robust by aggregating rather than by selecting the patches estimators. The WAV-reprojection estimator is a convex combination of the preliminary estimators. Formally,

$$\hat{I}_{\mathrm{WAV}}(\mathbf{x}) = \sum_{\boldsymbol{\delta} \in [\![0,W-1]\!]^2} \beta_{\boldsymbol{\delta}} \widehat{\mathsf{P}}^{I}_{\mathbf{x}-\boldsymbol{\delta}}\left(\boldsymbol{\delta}\right) \ , \tag{11}$$

with $\sum_{\boldsymbol{\delta} \in [\![0,W-1]\!]^2} \beta_{\boldsymbol{\delta}} = 1$, and $\beta_{\boldsymbol{\delta}} \geq 0$ for $\boldsymbol{\delta} \in [\![0, W-1]\!]^2$. The estimators $\left(\widehat{\mathsf{P}}^{I}_{\mathbf{x}-\boldsymbol{\delta}}\left(\boldsymbol{\delta}\right)\right)_{\boldsymbol{\delta} \in [\![0,W-1]\!]^2}$ are still handled as if they were unbiased. Therefore, to minimize the quadratic risk of the estimator in (11), one only needs to minimize the variance under the constraint that $\sum_{\boldsymbol{\delta} \in [\![0,W-1]\!]^2} \beta_{\boldsymbol{\delta}} = 1$. Using a Lagrangian formulation, this leads for all $\boldsymbol{\delta} \in [\![0, W-1]\!]^2$ to choose (cf. Appendix)

$$\beta_{\boldsymbol{\delta}} = \frac{\left[\mathrm{Var}\left(\widehat{\mathsf{P}}^{I}_{\mathbf{x}-\boldsymbol{\delta}}(\boldsymbol{\delta})\right)\right]^{-1}}{\sum_{\boldsymbol{\delta}' \in [\![0,W-1]\!]^2} \left[\mathrm{Var}\left(\widehat{\mathsf{P}}^{I}_{\mathbf{x}-\boldsymbol{\delta}'}(\boldsymbol{\delta}')\right)\right]^{-1}} \ .$$

For the flat kernel, thanks to the value of the approximated variance in that case (see Section 3.4), $\beta_{\boldsymbol{\delta}}$ is simply proportional to the number of averaged patches used to define $\widehat{\mathsf{P}}^{I}_{\mathbf{x}-\boldsymbol{\delta}}$.

This kind of aggregation procedure is also described in [16, 44] and in references therein, but in the context of aggregating wavelet-thresholded estimators for each patch (see also [45]). This point of view can be traced back to works on *Stacked Generalization*, *Stacked Regression* or *Generalized Ensemble Methods* [46, 47, 48].

3.6. Uniform Average of Candidates Reprojection (UAC)

The last reprojection we introduce, named UAC-reprojection, consists in making a uniform average of all the ”good patch candidates”. Formally,

$$\hat{I}_{\mathrm{UAC}}(\mathbf{x}) = \frac{\displaystyle\sum_{\boldsymbol{\delta}\in[\![0,W-1]\!]^2} \sum_{\mathbf{x}'\in\Omega_R(\mathbf{x})} \theta^I_{\mathrm{NLM}}(\mathbf{x}-\boldsymbol{\delta},\mathbf{x}'-\boldsymbol{\delta})\cdot I(\mathbf{x}')}{\displaystyle\sum_{\boldsymbol{\delta}'\in[\![0,W-1]\!]^2} \sum_{\mathbf{x}''\in\Omega_R(\mathbf{x})} \theta^I_{\mathrm{NLM}}(\mathbf{x}-\boldsymbol{\delta}',\mathbf{x}''-\boldsymbol{\delta}')} . \tag{12}$$

With the flat kernel, this estimator is exactly the same as the WAV-reprojection, justifying again our choice of the flat kernel. In particular it enables us to use a fast implementation of the UAC-reprojection based on “Summed Area Tables” [49] (also called “Integral Images” [50]) to compute the weights [51, 52]. This approach is not possible for the WAV-reprojection with other kernels.

Remark that in the general case, the pixel estimator $\hat{I}_{\mathrm{UAC}}(\mathbf{x})$ is different from $\hat{I}_{\mathrm{UAE}}(\mathbf{x})$, that assigns uniform weights to the estimators and that can be written

$$\hat{I}_{\mathrm{UAE}}(\mathbf{x}) = \frac{1}{W^2} \sum_{\boldsymbol{\delta}\in[\![0,W-1]\!]^2} \sum_{\mathbf{x}'\in\Omega_R(\mathbf{x})} \frac{\theta^I_{\mathrm{NLM}}(\mathbf{x}-\boldsymbol{\delta},\mathbf{x}'-\boldsymbol{\delta})\cdot I(\mathbf{x}')}{\displaystyle\sum_{\mathbf{x}''\in\Omega_R(\mathbf{x})} \theta^I_{\mathrm{NLM}}(\mathbf{x}-\boldsymbol{\delta},\mathbf{x}''-\boldsymbol{\delta})} .$$

4. Insights on reprojections improvements

Let us detail in this section several improvements provided by the reprojection framework.

4.1. Practical improvements

To denoise $I(\mathbf{x})$, the pixels of the image can help in two different ways. First they can be in the searching zone $\Omega_R(\mathbf{x})$, those pixels are averaged in the summation in (5). They can also be in what we call the influence zone $\Omega_{\mathrm{In}}(\mathbf{x})$. This zone consists of the pixels that govern the weights $\theta^I_{\mathrm{NLM}}(\mathbf{x}-\boldsymbol{\delta},\mathbf{x}'-\boldsymbol{\delta})$, for $\mathbf{x}'\in\Omega_R(\mathbf{x})$ and $\boldsymbol{\delta}\in[\![0,W-1]\!]^2$ in (5). Obviously, for any reprojection (with patches such that $W>1$) one has $\Omega_{\mathrm{In}}(\mathbf{x})\supset\Omega_R(\mathbf{x})$. For the central reprojection $|\Omega_{\mathrm{In}}(\mathbf{x})|=(R+W-1)^2>R^2$ (cf. Fig.7-a), but $\Omega_{\mathrm{In}}(\mathbf{x})$ increases for “sliding type” reprojection since $|\Omega_{\mathrm{In}}(\mathbf{x})|=(R+2W-2)^2>(R+W-1)^2$ (cf. Fig.7-b). In this case, by considering every patch to which a pixel belongs, denoising can be done more efficiently since the influence zone of each pixel is increased. This drives us to choose a smaller searching zone than in the literature on NL-Means: we already get good performance (cf. Tab.3) choosing $R = 9$ (while in [3], R is equal to 21).

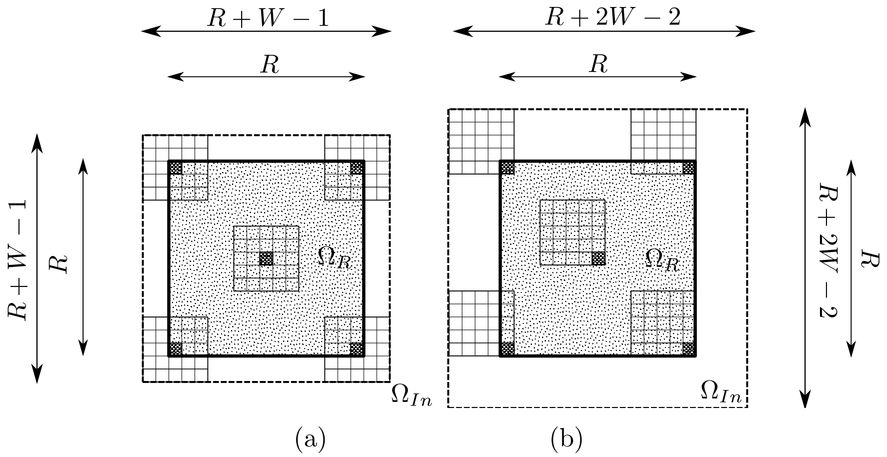


Figure 7: Searching zone ($R = 15$) and influence zone with central-reprojection (a) and with WAV-reprojection (b).

The concept of influence zone also appears in [36] for explaining that the influence of pixels propagates when one applies the NL-Means several times to an image. At a given step, two pixels $\mathbf{x}$ and $\mathbf{x}'$ with distance $2R$ can influence each other if there exists a third pixel $\mathbf{x}''$, at distance R from the two others, that has non-zero weights with respect to them at the previous step.

Moreover with a sliding type reprojection, W is no longer artificially constrained to odd values and the borders are easily handled, since a pixel always belong to at least one patch (the extreme case being a pixel in a corner of the image).

In practice, we can check that the WAV-reprojection we propose limits the artifacts encountered by the classical NL-Means method. Let us focus again on the "Jump" image. Using our WAV-reprojection with flat kernel, we have almost eliminated the *halo of noise* (cf. Fig.4-g). Expressed in term of PSNR, improvements are also important: for the classical NL-Means PSNR = 43.05, whereas for the flat UAE-reprojection PSNR = 46.51 and for the flat WAV-reprojection PSNR = 47.77 (parameters used are given in Fig.4). We give a more detailed simulation study of this impact of reprojections in Section 6.

4.2. Theoretical improvements: a toy model

We analyze here in details the example of the "Jump" image with N^2 pixels ($N = 256$). With this type of image, we measure the performance of two kinds of "Oracle NL-Means". Assume one has an oracle that can decide whether two noisy patches are noisy versions of the same patch, meaning one has access to $\Omega_{OR}(\mathbf{x}) = \{\mathbf{x}' \in \Omega_R(\mathbf{x}), \mathsf{P}^{I^\star}_{\mathbf{x}} = \mathsf{P}^{I^\star}_{\mathbf{x}'}\}$ as in Section 2.2. Based on this information, an oracle patch estimator is just the average of the selected noisy patches. From this oracle patch estimator, one can theoretically study oracle pixel estimators obtained using either the central-reprojection $\hat{I}_{\mathrm{ORC}}$, as defined in Section 3.2, or the Min-reprojection $\hat{I}_{\mathrm{ORM}}$, as defined in Section 3.3. To simplify the calculations, we neglect here border issues.

Let us first focus on $\hat{I}_{\mathrm{ORC}}$. One has two types of pixels to consider. First there are the pixels whose searching zone is included in a constant part of the image (say the black part). In that case we can find exactly R^2 candidates. Assuming the patches are independent, $\hat{I}_{\mathrm{ORC}}(\mathbf{x})$ is $\mathcal{N}(0, \frac{\sigma^2}{R^2})$ distributed. On the other hand, there are two sub-cases when the searching zone intersect the edge. If the centered patch intersects the edge, we have only R patches selected by the oracle: those translated along the direction of the edge. So in that case the estimator is $\mathcal{N}(0, \frac{\sigma^2}{R})$ distributed and there are $N(W-1)$ such pixels in the image. If the centered patch does not intersects the edge, the number of good candidates ranges from $R(R-1)$ to $R(R - \frac{W-1}{2})$. So we can write

$$\mathbb{E}\|\hat{I}_{\mathrm{ORC}} - I^\star\|_N^2 = \left(1 - \frac{R-1}{N}\right)\frac{\sigma^2}{R^2} \quad + \frac{\sigma^2}{NR}(W-1) + \frac{2}{N}\sum_{i=1}^{\frac{W+1}{2}} \frac{\sigma^2}{R(R-i)}\ , \tag{13}$$

with the notation $\|\cdot\|_N^2 = \|\cdot\|^2/N^2$.

For $\hat{I}_{\mathrm{ORM}}$ there are still two kinds of pixels to consider. The first ones are treated as before, but for pixels around the edge things are better now: sliding the patch in the direction orthogonal to the edge increases the number of similar (constant) patches found in the searching zone. So

$$\mathbb{E}\|\hat{I}_{\mathrm{ORM}} - I^\star\|_N^2 = \left(1 - \frac{R-1}{N}\right)\frac{\sigma^2}{R^2} + \frac{2}{N}\sum_{i=1}^{\frac{R-1}{2}} \frac{\sigma^2}{R(R-i)}\ . \tag{14}$$

Now remind that for our 8-bits images PSNR $= 10\log\frac{255^2}{MSE}$. With $\sigma = 20, R = 21, W = 9$, the oracle central-reprojection PSNR is 46.45, while for

Algorithm 1: WAV-reprojection with Flat Kernel

```
Data: $I, R, W, h$
Result: The filtered image $\hat{I}$
begin
    Initialize the variables $\hat{I}$, $Z$ and Dist2
    forall $\boldsymbol{\delta} \in [\![-(R-1)/2, (R-1)/2]\!]^2$ do
1       Pre-compute$(I, \mathrm{Dist2}, \boldsymbol{\delta}, W)$
        forall $\mathbf{x} \in \Omega$ do
            if $\mathrm{Dist2}(\mathbf{x})^2 \leq h^2$ then
2               forall $\mathbf{x}' \in P^I_{\mathbf{x}}$ do
                    $Z(\mathbf{x}') \leftarrow Z(\mathbf{x}') + 1$
                    $\hat{I}(\mathbf{x}') \leftarrow \hat{I}(\mathbf{x}') + I(\mathbf{x}' + \boldsymbol{\delta})$
                    $Z(\mathbf{x}' + \boldsymbol{\delta}) \leftarrow Z(\mathbf{x}' + \boldsymbol{\delta}) + 1$
3                   $\hat{I}(\mathbf{x}' + \boldsymbol{\delta}) \leftarrow \hat{I}(\mathbf{x}' + \boldsymbol{\delta}) + I(\mathbf{x}')$
    forall $\mathbf{x} \in \Omega$ do
        $\hat{I}(\mathbf{x}) \leftarrow \hat{I}(\mathbf{x})/Z(\mathbf{x})$
    return $\hat{I}$
end
```

the Min-reprojection PSNR is $48,42$. In practice our reprojections with the flat kernel are close to the oracle value: respectively 45.78 and 48.10 for the central and Min-reprojections. The gain of using a more refined reprojection is then confirmed both in theory and in practice.

4.3. Computational improvements

We have already seen that sliding-reprojections have a larger influence zone. Thus the best parameter R for those reprojections is smaller than the best parameter for the classical method. This speeds up the algorithm, since its complexity quadratically depends on R.

4.3.1. Implementation and complexity

Implementation was done in C, and following the philosophy of reproducible research, the open source code (mex-file) is available on-line at `http://people.math.jussieu.fr/~salmon/code/index_codes.php`. Recall that Ω is the set of pixels positions in the image, and that $\mathsf{P}^{I,W}_{\mathbf{x}}$ is the patch of size W with $\mathbf{x}$ as upper-left pixel. In Algorithm 1 we use three matrices of size $M \times N$:

- Z, the normalization of each pixel

- Dist2, the square distances between patches
- $\hat{I}$, the denoised image.

Note that in Algortihm 1 sometimes $\mathbf{x}+\boldsymbol{\delta}$ or $\mathbf{x}'$ are outside the image. Most authors deal with this border issue by increasing the size of their image either by a symmetric or a toroidal extension. We prefer not to take into account those elements, because it introduces artificial information and does not improve the quality of the denoising. Our implementation uses a pre-computation step (at line 1) to compute all distances between patches in the same way as in [51, 52]. After the execution of the procedure Pre-compute$(I, \text{Dist2}, \boldsymbol{\delta}, W)$, for all $\mathbf{x}$ the value $\|\mathsf{P}^{I,W}_{\mathbf{x}} - \mathsf{P}^{I,W}_{\mathbf{x}+\boldsymbol{\delta}}\|^2$ is stored in Dist2$(\mathbf{x})$.

In previous works, the pre-computation decreases the complexity down to $O(N^2R^2)$ which does not depend anymore on the size W of the patches. Though in our method when we find two similar patches, we update the whole patch (at line 2) instead of a single pixel. Therefore the worst case complexity is still $O(N^2R^2W^2)$.

But on most images, the number of patches $\mathsf{P}^{I,W}_{\mathbf{x}+\boldsymbol{\delta}}$ which are similar to $\mathsf{P}^{I,W}_{\mathbf{x}}$ is very small with regard to R^2. Therefore in practice, the computational time of our algorithm for WAV-reprojection is very close to the one for central-reprojection and does not depend much on W.

It is important to remark that the implementation of the UAC-reprojection is simpler and clearer to code than the UAE-reprojection. It is also faster, since the forementioned pre-computation can be used only for this reprojection.

Finally, it is very easy to exploit the symmetry of the problem, meaning that when we find that $\mathsf{P}^{I}_{\mathbf{x}}$ and $\mathsf{P}^{I}_{\mathbf{x}+\boldsymbol{\delta}}$ are similar, we update the values of the pixels both in $\mathsf{P}^{I}_{\mathbf{x}}$ and in $\mathsf{P}^{I}_{\mathbf{x}+\boldsymbol{\delta}}$ in $\hat{I}$ (at line 3). This trick has been noted in [21] and speeds up the algorithm by a factor 2.

4.3.2. Speeding up the NL-Means

To further speed up the NL-Means algorithms, many techniques have already been proposed. Among them one can do the main loop only on a fraction $\frac{1}{k}$ of the pixels, and still get $\frac{W^2}{k}$ estimators for each pixel, as in [7]. This sub-sampling method is a trade-off: one divides the execution time by a factor k while degrading the performance. However for small k this degradation is not significant, we observe a loss smaller than 0.2 dB for all

Table 1: Computing time (Mex/Matlab) for NL-Means with WAV-reprojection, flat kernel, a searching zone of width $R = 9$, and either one patch size $W_{\max} = 9$ (WAV) or two patch sizes $W_{\max} = 9, W_{\min} = 2$ (2-WAV). BM3D [16] is run with the standard 'nc' and 'high' parameters and K&B [7] is run using `cimgmatlab_safir`(ImNoisy,4,7,113.5,3,sigma,1). Results are average time for 10 realizations on Cameraman (256×256) and Barbara (512×512), run on an Intel(R) Core(TM) i7 CPU M620@2.67 GHz with 4 Gb of ram.

pixels	WAV	2-WAV	BM3D ('nc')	BM3D ('high')	K&B
256^2	1.22 s.	1.34 s.	1.22 s.	2.38 s.	18.11 s.
512^2	4.23s.	6.04 s.	5.06 s.	10.37 s.	53.02 s.

noises and all images considered choosing $k = 4$. Moreover, sub-sampling combined with the WAV-reprojection still reduces the *halo of noise*.

Algorithm 1 is implemented quite efficiently, but without low level optimizations (assembly language inside C) or parallelization (SIMD and multicore). For reference, Table 1 provides the computing time of our methods as well as the one of Safir and BM3D (Matlab+mex for all methods). One can check that our methods are particularly competitive in term of speed.

5. Combining different patch sizes

Another drawback of patch-oriented denoising algorithm is the problem appearing when the textons are smaller than the patch size. At worst no patch candidate is found in the searching zone, leading eventually the estimator to be the observed noisy image. Since usually one size of patch is used for all images, that might happen frequently. For instance, with cartoon-type images such as Flinstones, best performance is reached with $W = 1$, that is using Yarolavsky's Filter [27], when the noise level is small ($\sigma \leq 5$).

Thus a crucial issue is to aggregate estimators based on different sizes of patches to recover different scales of details. We introduce a simple method for aggregating estimators with two sizes of patches $W_{\max}$ and $W_{\min}$. Get $\hat{I}_{W_{\min}}$ and $\hat{I}_{W_{\max}}$ by any NL-Means procedure and keep the normalizations $Z_{\min}(\mathbf{x})$ and $Z_{\max}(\mathbf{x})$ for each pixel (i.e., the denominator in equation (3); see Algorithm 1 for more details). Then, one can average $\hat{I}_{W_{\max}}$ and $\hat{I}_{W_{\min}}$ with weights that are proportional to theses normalizations, re-weighted re-

Table 2: Evaluation of NL-Means variants: Flat and Gaussian kernel for central-reprojection. Others use the flat kernel : UAE-reprojection, WAV-reprojection, WAV-reprojection with two sizes of patches (2-WAV). For all procedures $W = 9, R = 9$, $h = \sigma$ (Gaussian), $h^2 = 2\sigma^2 q_{0.99}^{W^2}$ (flat) and $W_{\text{min}} = 2$ and $h^2_{\text{min}} = 2\sigma^2 q_{0.75}^{W^2_{\text{min}}}$ (2-WAV).

$\sigma = 20$	Flat	Gaussian	Flat UAE	Flat WAV	2-WAV
Barbara	28.77	29.04	30.03	30.17	30.04
Boats	28.51	28.82	29.52	29.57	29.71
Bridge	24.42	25.72	25.52	25.86	26.44
Camera.	27.62	28.22	28.76	29.18	29.74
Couple	28.16	28.53	29.22	29.32	29.44
Fingerp.	25.04	25.83	26.57	27.14	27.11
Flinst.	26.06	26.44	27.31	27.92	28.34
Hill	26.34	27.33	27.47	27.62	27.95
House	31.22	30.97	32.28	32.29	32.32
Lena	31.11	31.17	32.11	32.14	32.14
Man	28.54	29.11	29.60	29.58	29.81
Peppers	28.68	28.88	30.10	30.33	30.56

spectively by $1/W_{\text{max}}$ and $1/W_{\text{min}}$. For any pixel $\mathbf{x}$ the final estimator is

$$\hat{I}_{\text{min,max}}(\mathbf{x}) = \frac{\frac{Z_{\text{min}}(\mathbf{x})}{W_{\text{min}}} \hat{I}_{W_{\text{min}}}(\mathbf{x}) + \frac{Z_{\text{max}}(\mathbf{x})}{W_{\text{max}}} \hat{I}_{W_{\text{max}}}(\mathbf{x})}{\frac{Z_{\text{min}}(\mathbf{x})}{W_{\text{min}}} + \frac{Z_{\text{max}}(\mathbf{x})}{W_{\text{max}}}} \,. \tag{15}$$

Remark that for the WAV-reprojection with flat kernel, the normalization is just the number of (possibly repeated) pixels candidates used to denoise $\mathbf{x}$.

We divide by the patch widths in order to favor the larger, since we are more confident in denoising with large patches. Moreover, the selection needs to be more conservative with the small patches, because misidentifications occur more often. So, one should use a smaller bandwidth h for smaller patches: in practice we select $\alpha_{\text{min}} = 0.75$ in all our experiments.

Here we propose a simple and efficient solution to aggregate estimators with two different sizes of patch. Other solutions combining several patch sizes has been proposed in [22, 23], relying on unbiased risk estimation for combining/choosing various sizes (or shapes). Though the improvement is obtained through a higher computational cost.

Table 3: Evaluation of the NL-Means, for 2-WAV reprojection, with flat kernel, patch sizes $W_{\max} = 9, W_{\min} = 2$, and $R = 9$ $h^2_{\max} = 2\sigma^2 q^{W^2}_{0.99}$, $h^2_{\max} = 2\sigma^2 q^{W^2}_{0.75}$.

σ	5	10	20	50	100
Barbara	36.76	33.03	30.01	24.97	21.71
Boats	36.27	32.89	29.71	25.16	22.25
Bridge	34.87	30.14	26.47	22.45	20.22
Camera.	37.74	33.49	29.69	24.93	20.93
Couple	36.62	32.99	29.43	24.65	21.99
Fingerp.	34.62	29.86	27.15	22.54	18.72
Flinst.	35.91	31.64	28.36	23.30	18.88
Hill	35.58	31.44	28.07	24.10	21.97
House	38.61	35.26	32.20	26.84	23.12
Lena	37.89	34.97	32.04	27.32	23.74
Man	36.95	33.17	29.81	25.62	22.88
Peppers	37.31	33.89	30.69	25.14	21.29

6. Numerical Experiments

On natural images the UAE-reprojection is already quite an improvement. The WAV-reprojection procedure brings an additional gain of order 0.5dB on Bridge, Cameraman, Fingerprint and Flinstones while the performance for the other images is mostly the same. We present the example of Cameraman ($N = M = 256$) with $\sigma = 20$ and $W = 9, R = 21$. We get PSNR = 28, 16 for the original NL-Means (with $h = \sigma$, and the Gaussian kernel). Using the flat kernel, PSNR = 27.60 for central-reprojection, PSNR = 28.65 for UAE-reprojection and PSNR = 28.84 for WAV-reprojection (all with $h^2 = 2\sigma^2 q^{W^2}_{0.99}$). Better results are achieved using two size of patches, with WAV-reprojection and flat kernel ($W_{\max} = 9, W_{\min} = 2, h^2_{\max} = 2\sigma^2 q^{W^2_{\max}}_{0.99}, h^2_{\min} = 2\sigma^2 q^{W^2_{\min}}_{0.75}$). Note that visually, the importance of this two patch sizes procedure is particularly relevant. We present the performance of the different variation of the NL-Means that we have studied in Tab.2. It illustrates the advantages of the WAV-reprojection with respect to the others.

It is also important to test denoising methods using high quality images, since for instance images such as Cameraman, Lena and some other classical ones are of bad quality. For instance, the Cameraman image suffers from a black horizontal scratch in the bottom. Also testing on a big number of images makes the manual tuning of parameters a lot trickier, which makes

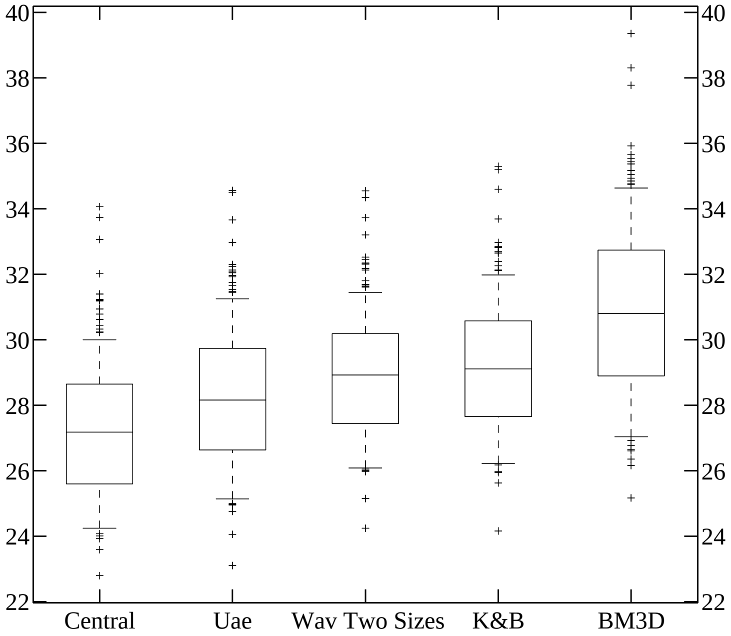


Figure 8: Box-plots of PNSR obtained for several methods on Condat's image database (150 images). From left to right: NL-Means with central-reprojection, UAE-reprojection and WAV-Reprojection with two size of patches (same settings as in Tab.3), Kervrann and Boulanger method's [7] and BM3D [16].

the comparison of methods much fairer. The 150 images we used in Fig.8 are available at `http://www.greyc.ensicaen.fr/~lcondat/imagebase.html`, and one is provided in Fig.10. We give in Fig.8 a box plot of five methods for a Gaussian noise of variance $\sigma = 20$: the first three are NL-Means variants with flat kernel: Central-reprojection, UAE-reprojection, and WAV-reprojection with two size of patches, all with $W = 9, R = 9, h^2 = 2\sigma^2 q_{0.99}^{W^2}$ and for the two sizes approach $h_{\max}^2 = 2\sigma^2 q_{0.99}^{W_{\max}^2}, h_{\min}^2 = 2\sigma^2 q_{0.75}^{W_{\min}^2}$. Then we compare with Kervrann and Boulanger method's [7] and BM3D method [16] both with standard parameters.

Our proposed method improves on previous NL-Means variants. Our performance is on par with the method in [7], but is still clearly outperformed by BM3D. Visual comparisons (Fig.10) give the same hierarchy in the methods tested. However, one should notice that the parameters in our method are not optimized with respect to the noise level, though numerical study show that larger patches should be used when the amount of noise increases.

Moreover, the number of tuning parameters we have used remains as small as possible.

We have also provided a comparison of the computing time of those algorithms. It is provided in Table 1. One can check that our algorithm is competitive with the BM3D algorithm in term of speed and outperforms the Safir algorithm described in [7]. Remark that we have compared versions which are non-optimized for speed, in particular we do not use sub-sampling in any of the three algorithms.

7. Conclusion

In this paper, we have presented new methods to combine the information obtained by patch-based approaches in the context of denoising. Aggregating in a more refined way the various pixel estimates due to patch overlaps, we can increase the performance of classical NL-Means procedures. This is particularly true for the method called WAV-reprojection, where the weight of each pixel estimate is choosen inversely proportionnal to its estimated variance. We have illustrated the performance of our new WAV-reprojection on a toy example and on natural images. Moreover we have explained in details how the different types of reprojection reduce the *noise halo*. We have also shown that using at least two sizes of patches improves the performance significantly. The extension of our framework has also lead to research on more general shapes as in [22, 23].

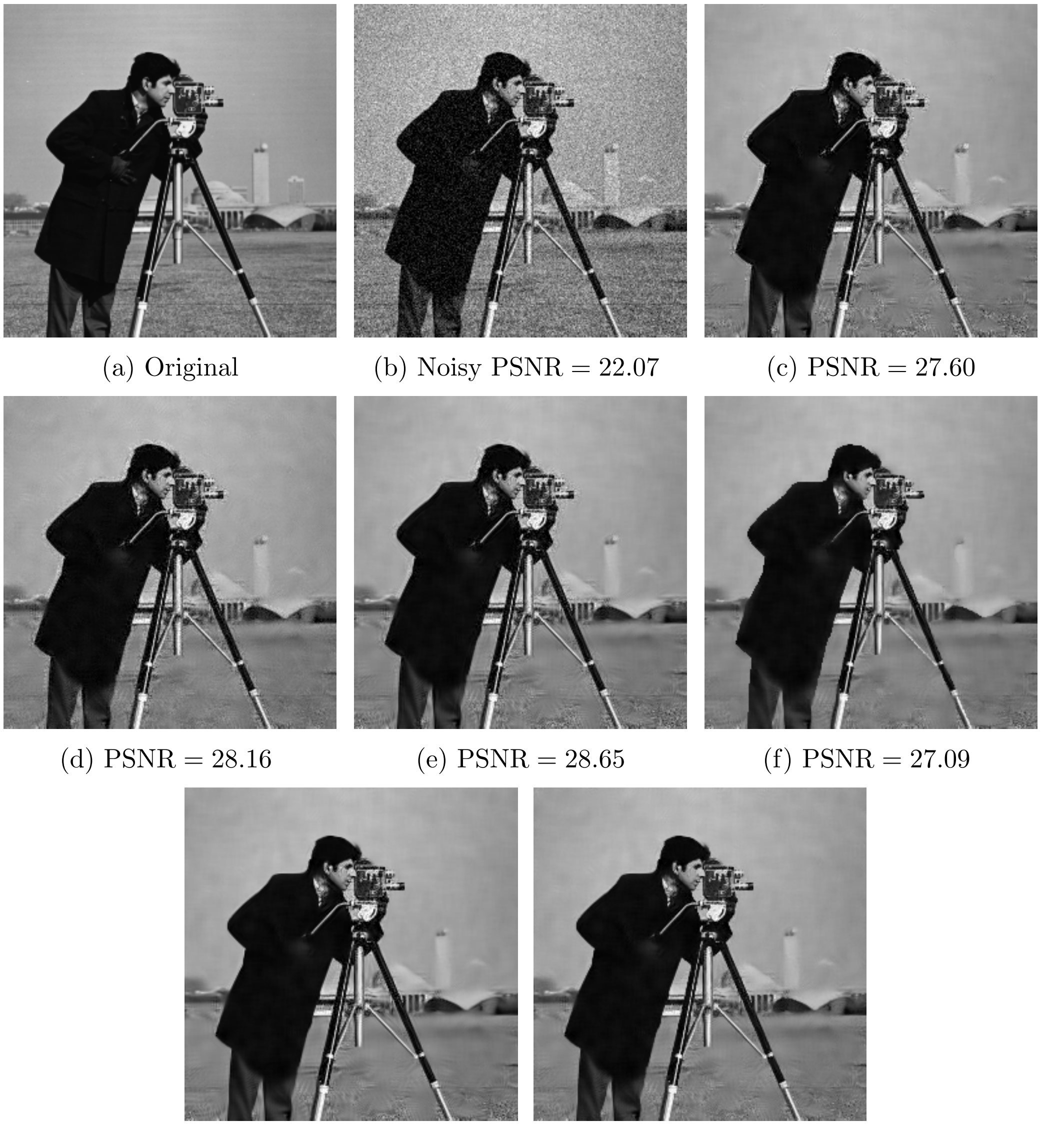
Appendix

We explain in detail how to determine the weights for the WAV-reprojection. From Equation (11), and assuming the pixels estimators $\left(\widehat{\mathsf{P}}^{I}_{\mathbf{x}-\boldsymbol{\delta}}(\boldsymbol{\delta})\right)_{\boldsymbol{\delta}\in[\![0,W-1]\!]^2}$ behave as if they were decorrelated, the variance of $\hat{I}_{\mathrm{WAV}}$ can be written

$$\mathrm{Var}\left(\hat{I}_{\mathrm{WAV}}(\mathbf{x})\right) = \mathrm{Var}\left[\sum_{\boldsymbol{\delta}\in[\![0,W-1]\!]^2} \beta_{\boldsymbol{\delta}}\widehat{\mathsf{P}}^{I}_{\mathbf{x}-\boldsymbol{\delta}}(\boldsymbol{\delta})\right] ,$$

$$\mathrm{Var}\left(\hat{I}_{\mathrm{WAV}}(\mathbf{x})\right) = \sum_{\boldsymbol{\delta}\in[\![0,W-1]\!]^2} \beta_{\boldsymbol{\delta}}^2 \,\mathrm{Var}\left[\widehat{\mathsf{P}}^{I}_{\mathbf{x}-\boldsymbol{\delta}}(\boldsymbol{\delta})\right] .$$

Thus to minimize the expectation of the quadratic error of the weighted average estimator, we only need to minimize the variance under the constraint



(a) Original

(b) Noisy PSNR = 22.07

(c) PSNR = 27.60

(d) PSNR = 28.16

(e) PSNR = 28.65

(f) PSNR = 27.09

(g) PSNR = 29.08

(h) PSNR = 29.64

Figure 9: Variant of NL-Means denoising with $R = 9, W = 9$, $\sigma = 20$ (a) original image, (b) noisy image, (c) Flat kernel and central-reprojection, (d) Classical NL-Means $h = \sigma$, (e) Flat kernel and UAE-reprojection, (f) Flat kernel and Min-reprojection, (g) Flat kernel and WAV-reprojection, (h) two patch sizes $W_{\max} = 9, W_{\min} = 2$, $h^2_{\max} = 2\sigma^2 q_{0.99}^{W^2_{\max}}, h^2_{\min} = 2\sigma^2 q_{0.75}^{W^2_{\min}}$ and $h^2 = 2\sigma^2 q_{0.99}^{W^2}$ when the flat kernel is used.

that $\sum_{\boldsymbol{\delta}\in[\![0,W-1]\!]^2}\beta_{\boldsymbol{\delta}}=1$. The dual (Lagrangian) form is

$$\mathcal{L}=\sum_{\boldsymbol{\delta}\in[\![0,W-1]\!]^2}\beta_{\boldsymbol{\delta}}^2\operatorname{Var}\left[\widehat{\mathsf{P}}^{I}_{\mathbf{x}-\boldsymbol{\delta}}(\boldsymbol{\delta})\right]+\lambda\left(\sum_{\boldsymbol{\delta}\in[\![0,W-1]\!]^2}\beta_{\boldsymbol{\delta}}-1\right)\ .$$

The first order conditions gives for any $\boldsymbol{\delta}$:

$$\beta_{\boldsymbol{\delta}}=-\lambda/\left(2\operatorname{Var}\left[\widehat{\mathsf{P}}^{I}_{\mathbf{x}-\boldsymbol{\delta}}(\boldsymbol{\delta})\right]\right)\ .$$

Since the coefficients are normalized, for all $\boldsymbol{\delta}\in[\![0,W-1]\!]^2$ we get

$$\beta_{\boldsymbol{\delta}}=\frac{\left[\operatorname{Var}\left(\widehat{\mathsf{P}}^{I}_{\mathbf{x}-\boldsymbol{\delta}}(\boldsymbol{\delta})\right)\right]^{-1}}{\sum_{\boldsymbol{\delta}'\in[\![0,W-1]\!]^2}\left[\operatorname{Var}\left(\widehat{\mathsf{P}}^{I}_{\mathbf{x}-\boldsymbol{\delta}'}(\boldsymbol{\delta}')\right)\right]^{-1}}\ .$$

With the flat kernel, the weight $\beta_{\boldsymbol{\delta}}$ is proportional to the number of selected patches to denoise $\mathsf{P}^{I}_{\mathbf{x}-\boldsymbol{\delta}}$ (since the variance of the mean of n i.i.d random variables is just the variance of one the variables divided by n).

Acknowledgment

The authors would like to thank Vincent Duval and Charles-Alban Deledalle for fruitful discussions and Jérôme Boulanger and Charles Kervrann for providing us with the experimental results of their algorithms.

References

[1] A. A. Efros, T. Leung, Texture Synthesis by Non-parametric Sampling, in: ICCV, 1033–1038, 1999.

[2] A. Criminisi, P. Pérez, K. Toyama, Region filling and object removal by exemplar-based image inpainting, IEEE Trans. Image Process. 13 (9) (2004) 1200–1212.

[3] A. Buades, B. Coll, J.-M. Morel, A review of image denoising algorithms, with a new one, Multiscale Model. Simul. 4 (2) (2005) 490–530.

[4] A. Buades, Image and movie denoising by non local means, Ph.D. thesis, Universitat de les Illes Balears, 2006.

[5] A. Buades, B. Coll, J.-M. Morel, Nonlocal Image and Movie Denoising, Int. J. Comput. Vision 76 (2) (2008) 123–139.

[6] S. P. Awate, R. T. Whitaker, Unsupervised, Information-Theoretic, Adaptive Image Filtering for Image Restoration, IEEE Trans. Pattern Anal. Mach. Intell. 28 (3) (2006) 364–376.

[7] C. Kervrann, J. Boulanger, Optimal Spatial Adaptation for Patch-Based Image Denoising, IEEE Trans. Image Process. 15 (10) (2006) 2866–2878.

[8] J. Boulanger, C. Kervrann, P. Bouthemy, P. Elbau, J.-B. Sibarita, J. Salamero, Patch-based nonlocal functional for denoising fluorescence microscopy image sequences., IEEE transactions on medical imaging 29 (2) (2010) 442–454.

[9] D. L. Donoho, I. M. Johnstone, Ideal spatial adaptation by wavelet shrinkage, Biometrika 81 (3) (1994) 425–455.

[10] J. Portilla, V. Strela, M. Wainwright, E. P. Simoncelli, Image denoising using scale mixtures of Gaussians in the wavelet domain, IEEE Trans. Image Process. 12 (11) (2003) 1338–1351.

[11] J.-L. Starck, E. J. Candès, D. L. Donoho, The curvelet transform for image denoising, IEEE Trans. Image Process. 11 (6) (2002) 670–684.

[12] E. Le Pennec, S. Mallat, Sparse geometric image representations with bandelets, IEEE Trans. Image Process. 14 (4) (2005) 423–438.

[13] P. Perona, J. Malik, Scale Space and Edge Detection Using Anisotropic Diffusion, IEEE Trans. Pattern Anal. Mach. Intell. 12 (1990) 629–639.

[14] G. Sapiro, Geometric Partial Differential Equations and Image Analysis, Cambridge University Press, 2001.

[15] S. Mallat, A wavelet tour of signal processing, Elsevier/Academic Press, Amsterdam, the sparse way, With contributions from Gabriel Peyré, 2009.

[16] K. Dabov, A. Foi, V. Katkovnik, K. O. Egiazarian, Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering, IEEE Trans. Image Process. 16 (8) (2007) 2080–2095.

[17] J. Mairal, F. Bach, J. Ponce, G. Sapiro, A. Zisserman, Non-local sparse models for image restoration, in: ICCV, 2272–2279, 2009.

[18] S. S. Chen, D. L. Donoho, Atomic decomposition by basis pursuit, in: SPIE, 1995.

[19] R. Tibshirani, Regression Shrinkage and Selection via the Lasso, J. Roy. Statist. Soc. Ser. B 58 (1) (1996) 267–288.

[20] S. S. Chen, D. L. Donoho, M. A. Saunders, Atomic decomposition by basis pursuit, SIAM J. Sci. Comput. 20 (1) (1998) 33–61 (electronic), ISSN 1064-8275.

[21] B. Goossens, H. Q. Luong, A. Pizurica, W. Philips, An improved non-local denoising algorithm, in: LNLA, 143–156, 2008.

[22] C.-A. Deledalle, V. Duval, J. Salmon, Non-local Methods with Shape-Adaptive Patches (NLM-SAP), Journal of Mathematical Imaging and Vision (2011) 1–18ISSN 0924-9907.

[23] C.-A. Deledalle, V. Duval, J. Salmon, Anisotropic Non-Local Means with Spatially Adaptive Patch Shapes, in: SSVM, 2011.

[24] J. Salmon, Y. Strozecki, From Patches to Pixels in Non-Local methods: Weighted-Average Reprojection, in: ICIP, 1929–1932, 2010.

[25] J. Polzehl, V. G. Spokoiny, Image denoising: pointwise adaptive approach, Ann. Statist. 31 (1) (2003) 30–57.

[26] C. Tomasi, R. Manduchi, Bilateral Filtering for Gray and Color Images, in: ICCV, 839–846, 1998.

[27] L. P. Yaroslavsky, Digital picture processing, vol. 9 of *Springer Series in Information Sciences*, Springer-Verlag, Berlin, 1985.

[28] N. Azzabou, N. Paragios, F. Guichard, Image Denoising Based on Adapted Dictionary Computation, in: ICIP, 109–112, 2007.

[29] T. Tasdizen, Principal components for non-local means image denoising, in: ICIP, 1728–1731, 2008.

[30] J. Orchard, M. Ebrahimi, A. Wong, Efficient nonlocal-means denoising using the SVD, in: ICIP, 1732–1735, 2008.

[31] T. Tasdizen, Principal neighborhood dictionaries for nonlocal means image denoising, IEEE Trans. Image Process. 18 (12) (2009) 2649–2660.

[32] D. Van De Ville, M. Kocher, Non-Local Means with Dimensionality Reduction and SURE-Based Parameter Selection, IEEE Trans. Image Process. (99) (2011) 1–1.

[33] D. Van De Ville, M. Kocher, SURE-Based Non-Local Means, IEEE Signal Process. Lett. 16 (2009) 973–976.

[34] J. Salmon, On two parameters for denoising with Non-Local Means, IEEE Signal Process. Lett. 17 (2010) 269–272.

[35] T. Brox, O. Kleinschmidt, D. Cremers, Efficient Nonlocal Means for Denoising of Textural Patterns, IEEE Trans. Image Process. 17 (7) (2008) 1083–1092.

[36] T. Brox, D. Cremers, Iterated Nonlocal Means for Texture Restoration, in: SSVM, vol. 4485 of *Lecture Notes in Computer Science*, 13–24, 2007.

[37] S. Zimmer, S. Didas, J. Weickert, A rotationally invariant block matching strategy improving image denoising with non-local means, in: LNLA, 2008.

[38] V. Doré, M. Cheriet, Robust NL-Means Filter With Optimal Pixel-Wise Smoothing Parameter for Statistical Image Denoising, IEEE Trans. Signal Process. 57 (2009) 1703–1716.

[40] O. V. Lepski, E. Mammen, V. G. Spokoiny, Optimal spatial adaptation to inhomogeneous smoothness: an approach based on kernel estimates with variable bandwidth selectors, Ann. Statist. 25 (3) (1997) 929–947.

[42] A. Singer, Y. Shkolnisky, B. Nadler, Diffusion interpretation of nonlocal neighborhood filters for signal denoising., SIAM J. Imaging Sci. 2 (1) (2009) 118–139.

[43] C. Kervrann, J. Boulanger, P. Coupé, Bayesian Non-local Means Filter, Image Redundancy and Adaptive Dictionaries for Noise Removal, in: SSVM, vol. 4485, 520–532, 2007.

[44] V. Katkovnik, A. Foi, K. O. Egiazarian, J. T. Astola, From Local Kernel to Nonlocal Multiple-Model Image Denoising, Int. J. Comput. Vision 86 (1) (2010) 1–32.

[45] O. G. Guleryuz, Weighted Averaging for Denoising With Overcomplete Dictionaries, IEEE Trans. Image Process. 16 (12) (2007) 3020–3034.

[46] D. H. Wolpert, Stacked generalization, Neural Networks 5 (2) (1992) 241–259.

[47] M. P. Perrone, Improving regression estimation: Averaging methods for variance reduction with extensions to general convex measure optimization, Ph.D. thesis, Brown University, Providence, RI, USA, 1993.

[48] L. Breiman, Stacked regressions, Mach. Learn. 24 (1) (1996) 49–64.

[49] F. C. Crow, Summed-area tables for texture mapping, in: SIGGRAPH, 207–212, 1984.

[50] P. Viola, M. Jones, Rapid Object Detection using a Boosted Cascade of Simple Features, CVPR 1 (2001) 511–518.

[51] J. Wang, Y.-W. Guo, Y. Ying, Y.-L. Liu, Q.-S. Peng, Fast Non-Local Algorithm for Image Denoising, in: ICIP, 1429–1432, 2006.

[52] J. Darbon, A. Cunha, T. F. Chan, S. Osher, G. J. Jensen, Fast Nonlocal Filtering Applied to Electron Cryomicroscopy, in: ISBI, 1331–1334, 2008.

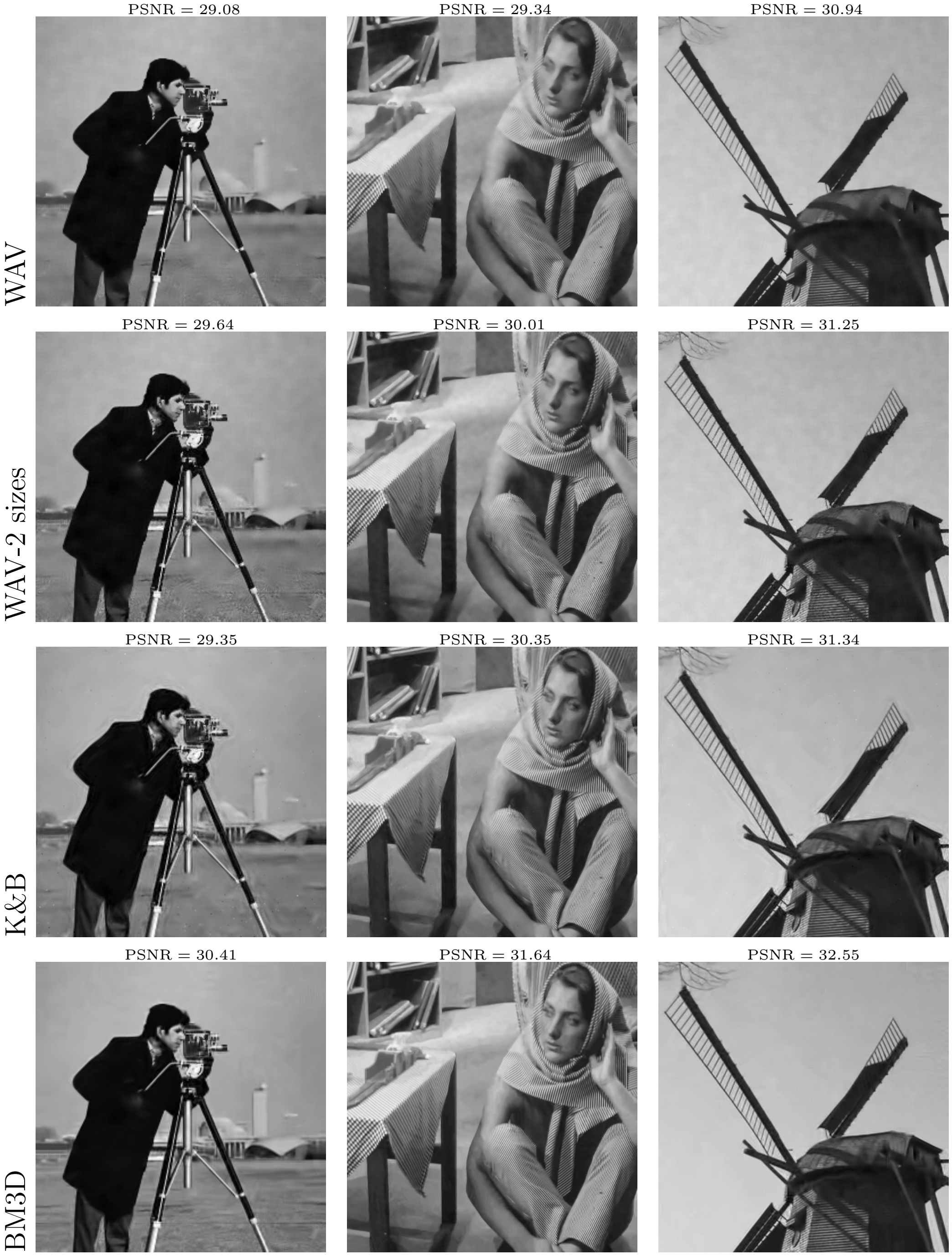


Figure 10: Comparisons on Cameraman, Barabara and Windmill, $\sigma = 20$. Top to bottom: WAV-reprojection, WAV-reprojection with two sizes, K&B [7] and BM3D [16].