

Non-Local Methods with Shape-Adaptive Patches (NLM-SAP)

Charles-Alban Deledalle · Vincent Duval · Joseph Salmon

Received: date / Accepted: date

Abstract We propose in this paper an extension of the Non-Local Means (NL-Means) denoising algorithm. The idea is to replace the usual square patches used to compare pixel neighborhoods with various shapes that can take advantage of the local geometry of the image. We provide a fast algorithm to compute the NL-Means with arbitrary shapes thanks to the Fast Fourier Transform. We then consider local combinations of the estimators associated with various shapes by using Stein's Unbiased Risk Estimate (SURE). Experimental results show that this algorithm improve the standard NL-Means performance and is close to *state-of-the-art* methods, both in terms of visual quality and numerical results. Moreover, common visual artifacts usually observed by denoising with NL-Means are reduced or suppressed thanks to our approach.

Keywords Image denoising · non-local means · spatial adaptivity · aggregation · risk estimation · SURE

1 Introduction

As simple as it may seem, the classical problem of image denoising in the presence of Additive White Gaussian

Noise (AWGN) has drawn a lot of efforts from the image processing community. A wide variety of strategies were proposed during the last decades, from partial differential equations to transform-domain methods such as wavelets, approximation theory or stochastic analysis.

In fact, a first major difficulty in image denoising is to handle efficiently regular parts while preventing edges from being blurred. To this end, diffusion methods [42] and variational methods [48] have shown their effectiveness. The latter approaches introduce a prior model of the noise-free image and search for a trade-off between data fitting and regularization. It is crucial for these techniques to define a suitable prior that guarantees both the smoothness of the denoised image and the preservation of its structure. Total variation [48] is an example of a prior that enforces smoothness while preserving edges. However, such priors produce staircasing effect [40] and damage textures.

Donoho and Johnstone [17] have shown that spatial adaptive methods can be obtained by soft-thresholding wavelet decompositions. Wavelet shrinkage can be interpreted from a Bayesian perspective. It results from an ℓ_1 sparsity-promoting prior: the regularity of the signal is enforced by using sparsifying priors on the coefficients in the transform domain. Indeed, wavelet and cosine bases are well known to be able to capture most of a signal/image with very few coefficients, a good property for compression. For denoising applications, such orthogonal transforms lead to a separation of signal and noise. Noise can then be strongly suppressed by canceling the least significant coefficients. These types of methods culminated with the BLS-GSM algorithm [45]. Other improvements of wavelet oriented methods have been proposed such as Shape Adaptive Domains

Charles-Alban Deledalle
Institut Telecom ; Telecom ParisTech ; CNRS LTCI
46, rue Barrault 75634 Paris cedex 13, FRANCE
E-mail: charles-alban.deledalle@telecom-paristech.fr

Vincent Duval
Institut Telecom ; Telecom ParisTech ; CNRS LTCI
46, rue Barrault 75634 Paris cedex 13, FRANCE
E-mail: vincent.duval@telecom-paristech.fr

Joseph Salmon
Université Paris 7 - Diderot ; Laboratoire de Probabilité et
Modèles Aléatoires ; CNRS-UMR 7599
175 rue du Chevaleret 75013 Paris, FRANCE
E-mail: salmon@math.jussieu.fr

[20] and over-complete [53] or geometric [29] decompositions.

Spatial adaptivity can also be reached by considering adaptive neighbor filters, as the well known Sigma Filter [30], Yaroslavsky Filter [63] and Bilateral Filter [56]. Moreover, applications of Lepski's method [31] (known in signal processing as the ICI rule, Intersection of Confidence Intervals) to kernel smoothing are also known to perform well for denoising. Their theoretical performance is as good as the best wavelet shrinking methods: they reach adaptive rate in the minimax sense [22] for regular functions. In practice, Lepski's rule has successfully been applied to image denoising with Adaptive Weights Smoothing [43] or Shape Adaptive Windows [25] for instance.

Though very efficient at dealing with edges, such methods cannot proceed efficiently in textured regions. To overcome this drawback, many authors have proposed to work with small sub-images, called patches, to take into account the redundancy in natural images, especially in textured parts. The interest of using patches lies in their robustness to noise. The NL-Means (Non-Local Means) [7] and UINTA [2] algorithms are typical examples of this approach, as is their extension using Lepski's method [27]. Those algorithms consist in averaging similar pixels by measuring their similarity through patches. Iterative versions have also been proposed [2, 27, 6, 21, 16, 5] as well as mixed methods using Total Variation [34].

Another approach proposed by Aharon *et al.* [1] and extended in [37, 36] also uses patches to denoise the image. Though, for these methods, the key point is to get a good representation for each patch of the image by using dictionary learning techniques, well-suited for overcomplete representation (based for instance on ℓ_1 regularization or greedy algorithms). This leads to a *state-of-the-art* denoising method called Learned Simultaneous Sparse Coding (LSSC) [36]. It has to be noted that this algorithm incorporates a prior on images by initializing a dictionary of patches on a large clean dataset. Another *state-of-the-art* method in denoising is BM3D [11]. It also relies on patches and consists of a smart combination of classical filtering techniques, such as wavelet denoising and Wiener's Filter, applied in the space of patches. We refer to [26] for a recent review of neighborhood filters.

These methods are already quite efficient at dealing with smooth regions and textures. However, since they use patches with a fixed square shape and a fixed scale over the whole image, their performances may be limited for dealing with edges, mostly for edges with high contrast. Indeed, edges with high contrast present few redundancies and their denoising version suffer from a

persistence of residual noise: this is named the *noise halo*. In order to overcome this drawback, more directional priors may be considered, using locally chosen scales and orientations of shapes. Few attempts have been made to use several patch sizes (see [37] for learning with patches or [51] for the NL-Means).

As far as we know, the only work trying to handle variable shapes rather than simple square has recently been proposed by Dabov *et al.* [12] as a way to improve the BM3D algorithm. The authors propose to adapt the shapes used by the algorithm: they locally select a shape by applying Lepski's method, and then perform the same steps of the BM3D algorithm with these shapes rather than with common square blocks (i.e., patches).

Recently, spatial-adaptive methods for selecting the parameters of NL-means have been proposed in [19, 18]. Both methods locally select the parameters which minimize a local estimate of the risk (i.e., the Mean Square Error, MSE) by considering respectively Stein's Unbiased Risk Estimate (SURE) [54, 55] or the C_p criterion [38]. The use of SURE for NL-Means was originally proposed in order to select the best bandwidth parameter [58]. Applications of SURE emerged for choosing the smoothing parameter in families of linear estimates [33] such as for model selection, ridge regression, smoothing splines, etc. It was then widely used in the wavelet community after the introduction of the SURE-Shrink algorithm [17]. Solo [52] gave a general form of SURE for an estimator defined as a minimizer of a regular energy, especially for least square regression regularized by a Sobolev norm or the Total Variation. More recently, linear combinations of estimates based on SURE were considered [4] instead of the selection of a single one. Moreover, Ramani *et al.* [46] have described a Monte Carlo approach to evaluate SURE when a closed-form expression is not available or too computer-intensive. Our paper is in the same vein as [58, 19] and proposes to locally select or aggregate the best shapes.

Our contributions— The aim of this paper is to highlight the potential benefit of replacing the simple square patches with more general shapes, in the classical NL-Means filter. To this end, we propose a fast algorithm, based on the fast Fourier transform, which allows to compute the solution of the NL-Means for arbitrary patch shapes. Then we explain how to combine the estimators associated with each shape in a suitable way. We select or combine locally the shape-based estimates by measuring the performance of their associated denoisers with SURE. We coin such type of algorithms Non-Local Means with Shape-Adaptive Patches (NLM-SAP).

The main advantage of using adaptive patch shapes in the context of NL-Means is to reduce the *noise halo* produced around edges with high contrast. Our method is an improvement of the NL-Means taking into account the anisotropy of natural images. It is all the more relevant when the images to denoise present edges with high contrast for which the classical NL-Means fails. The difference with [12] is that we use a previously fixed family of shapes instead of learning them while processing the image.

Organization of the paper — We give in Section 2 a general overview of the NL-Means method. We introduce in Section 3 a more general framework using general shapes instead of square patches to measure the similarity between pixels. This leads in Section 4 to the natural problem of locally selecting or combining the best shapes in our NLM-SAP algorithm. In Section 5 we illustrate numerically, and above all visually, the gain in aggregating various shape-based estimates in a proper manner.

2 An overview of the NL-Means

In this section we define the model we use for the noisy image, and we recall the definition and basic facts about NL-Means. We also present some limitations and justify the need to introduce variable shapes instead of the standard fixed square patches.

2.1 The AWGN model

In this paper we are concerned with the problem of denoising. A noisy image \mathbf{Y} is given and assumed to be a version of an unobserved deterministic image \mathbf{f} corrupted by an Additive White Gaussian Noise (AWGN). Let $\Omega \subset \mathbb{Z}^2$ be the (bounded) indexing set of the pixels. For any pixel x in the grid Ω , the model has the following formulation:

$$\mathbf{Y}(x) = \mathbf{f}(x) + \boldsymbol{\varepsilon}(x), \quad (1)$$

where $\boldsymbol{\varepsilon}$ is a centered Gaussian noise with known variance σ^2 and the noise components $\boldsymbol{\varepsilon}(x)$ are independent. Even if we only focus on the case of AWGN, our methodology could be adapted to other noise models by considering NL-Means extensions such as [28, 16] and statistical results on unbiased risk estimation [24].

2.2 Definition and parameters

In this section we present the usual definition of the NL-Means procedure as introduced by Buades *et al.*

[7]. It is an averaging filter, i.e., for each pixel the output of the procedure is a weighted average of the whole image. The weights used are selected using a “metric” which determines whether two pixels are similar or not. For instance, with the Bilateral Filter [56], the “metric” consists in comparing both pixels values and pixel locations. The core idea of the NL-Means is to create a metric governed by patches surrounding each pixel, regardless of their position, i.e., non-local in the image space. For any pixel of interest x , an estimate of the value $\mathbf{f}(x)$ is given by:

$$\hat{\mathbf{f}}(x) = \frac{\sum_{x' \in \Omega} \omega(x, x') \mathbf{Y}(x')}{\sum_{x' \in \Omega} \omega(x, x')}, \quad (2)$$

where the weights $\omega(x, x')$ depend on patches around x and x' . The denominator is a normalizing factor which ensures the weights sum to one. For a fixed (odd) width p , a patch \mathbf{P}_x is a subimage of width p , centered around the pixel of interest x :

$$\mathbf{P}_x = \left(\mathbf{Y}(x + \tau), \tau \in \left[\left[-\frac{p-1}{2}, \frac{p-1}{2} \right] \right)^2 \right). \quad (3)$$

where $\llbracket n_1, n_2 \rrbracket = \{n_1, n_1 + 1, \dots, n_2\}$. With this notation, the original weights in the NL-Means are of the following form:

$$\omega(x, x') = \varphi \left(\frac{\|\mathbf{P}_x - \mathbf{P}_{x'}\|_{2,a}^2}{2h^2} \right), \quad (4)$$

where $h > 0$ is the bandwidth parameter, φ is the kernel used to measure similarity between patches, $\|\cdot\|_{2,a}$ is a weighted Euclidean norm using a Gaussian kernel, and a is the bandwidth that controls the concentration of the kernel around the central pixel. Let us now briefly recall the influence of each parameter (see [19] for a more extensive discussion on this subject).

The bandwidth h : this parameter has a smoothing effect and plays the same role as the bandwidth for kernel methods in statistics (cf. [62] for more details). With our parametrization, the larger the bandwidth, the smoother the image becomes. Choosing this parameter is a difficult task and many solutions have been proposed in the literature. The simplest and most common one is to set a single h for the whole image, whose value is determined by cross validation on a small dataset of images. In [44, 27], the authors set this parameter according to the quantile of a χ^2 distribution, due to the particular metric they consider to compare pixels (or patches). Van De Ville and Kocher [58] calculate an unbiased risk estimate of the NL-Means to globally select the bandwidth h . Pursuing this idea, Duval *et al.* [19] consider a method based on



Fig. 1 Illustration of the *noise halo* appearing around edges with high contrast on images denoised by NL-means. The input noisy images was corrupted version of the noise-free images presented on Fig. 6 damaged by AWGN with standard deviation $\sigma = 20$. Noise halo arises from an abrupt lack of redundancy around edges sometimes referred as the *rare patch effect*.

the same approach but to locally select the bandwidth. The main interest of the last two approaches is that they give a closed-form expression of the SURE for NL-Means.

The search window size ℓ : the summation defined in Equation (2) is usually restricted to a search window around the pixel of interest: it is by convention an $\ell \times \ell$ square window. Defining $\Omega_\ell = \llbracket -\frac{\ell-1}{2}, \frac{\ell-1}{2} \rrbracket^2$, the search window centered on each pixel x is then $x + \Omega_\ell$. This was proposed in the seminal work [7] for computational acceleration. Though, some authors have also noticed that choosing locally the best search windows [27] or restricting the average over small ones (see [19] or [49] for more details) could benefit to the NL-Means procedure.

The kernel φ : function φ was chosen by Buades *et al.* [7] as $t \mapsto \exp(-t)$, but other choices may be considered, such as compactly supported smooth functions. It was noticed by some authors [23] that weights with compact support yield better results.

In this paper we have conducted our experiments with the classical exponential kernel and with the trapezoidal kernel as defined in [8]. In what follows, we provide results with the second kernel, since in practice we have observed better performance (up to 0.2dB) for all considered images. Moreover any other \mathcal{C}^1 kernel could be handled by our framework [19].

The patch size p : this parameter is generally chosen equal to 5, 7 or 9. Using a width $p = 1$ would lead to a method close to the Bilateral Filter [56] or Yaroslavsky Filter [63]. Moreover, in most papers the patch width is a global fixed parameter. Few works have tried to handle the difficult task of using several sizes of patches for a single image. To our knowledge, the first attempt was proposed in the context of learning patches by [37], using Support Vector Machines (SVM).

Our approach extends the notion of patches to the more general concept of shapes. Our goal is to better handle anisotropy in natural images using oriented patches. Indeed, it is well known that edges are not satisfactorily denoised when using square patches with the NL-Means. This leads to a *noise halo* around edges (see Fig. 1). We propose in the next section an algorithm to compute efficiently the solution of NL-means for arbitrary shapes. We also consider in details several natural collections of shapes. Numerical results are given in Section 5.

3 From patches to shapes: beyond the *rare patch effect*

In practice, we have seen that the original algorithm suffers from a *noise halo* around edges, due to an abrupt lack of redundancy of the image. This phenomenon is sometimes referred to as the *rare patch effect* (Fig. 1). Statistically, it leads to an NL-Means estimator with large variance around edges. Several solutions have already been proposed to handle this drawback [51, 19, 34] (see Section 3.2 for more details). Here, we generalize these approaches by considering general shapes rather than simple square patches (cf. Fig. 2).

In order to deal with patches of arbitrary shapes, we reformulate the way the distance between two pixels is measured in terms of patches. The weighted Euclidean distance $\|\cdot\|_{2,a}$ used in Eq. (4) can be generalized using the following expression:

$$d_{\mathbf{S}}^2(x, x') = \sum_{\tau \in \Omega} \mathbf{S}(\tau) (\mathbf{Y}(x + \tau) - \mathbf{Y}(x' + \tau))^2, \quad (5)$$

where \mathbf{S} encodes the shape we aim at. With this notation we can easily rewrite the original NL-Means with a simple \mathbf{S} by choosing:

$$\mathbf{S}(\tau) = \begin{cases} 1, & \text{if } \|\tau\|_\infty \leq \frac{p-1}{2}, \\ 0, & \text{otherwise,} \end{cases} \quad (6)$$

One of our contributions is to provide an efficient algorithm, based on the Fast Fourier Transform (FFT), to compute the distances in Eq. (5). Our implementation is independent of the shape, and can thus be applied with different shapes (see Section 3.1 for details). As soon as we consider the use of anisotropic shapes, and not just squares or disks centered on the pixel of interest, two questions emerge. The first one is how to choose the collection of shapes to consider. The second issue is to propose a way to combine the estimators provided by each shape.

The collection of shapes should be composed of more than one shape to locally take into account the geometrical properties of natural images. Consider for instance the use of a single vertically elongated patch. This could be interesting to handle vertical features, but eventually we would not optimally deal with horizontal details with only one oriented shape. So, the collection should be diversified and numerous enough to identify directional features (see Fig. 8 for a visual illustration). At the same time, it should remain small enough so that the algorithm is not computationally intensive.

3.1 Fast algorithm to handle shapes

In this section, we present a fast way to compute the NL-Means weights for general shapes, based on the 2D-FFT. It is inspired from works initiated in [60] and [15] to speed up the NL-Means algorithm. However, contrary to these approaches, ours can deal with non-square and/or non-binary patches, i.e., with general shapes \mathbf{S} . Like them, our method is independent of the shape size. Let us also mention that other fast implementations of the NL-Means have been proposed [35, 3, 41]: such methods use a pre-selection of the patches based either on statistical tests or gradients comparisons. We should however emphasize that the final estimates with those approaches are approximate solutions of the original NL-Means. Our method does not rely on such tricks and computes the exact NL-Means in the case of a square shape in Eq. (5).

In [60] and [15], the authors propose to compute the Euclidean distances using “Summed Area Tables” [10] (also called “Integral Images” [59]). This allows them to reduce the computational cost of the NL-Means from $O(|W| \cdot |\Omega| \cdot |P|)$ to $O(|W| \cdot |\Omega|)$, where $|W| = \ell^2$ is the number of pixels in the search window, $|\Omega|$ is the image domain size and $|P|$ is the patch size (we refer to [15] for more details). To compute these integral images, the authors change the original algorithm by swapping the two “for” loops: instead of considering all the shifts for each pixel, they consider all the pixels for each shift.

We use basically the same swapping trick. Notice that Equation (5) can be reformulated for any translation parameter δ (i.e., taking $x' = x + \delta$) as a discrete convolution:

$$\begin{aligned} d_{\mathbf{S}}^2(x, x + \delta) &= \sum_{\tau \in \Omega} \mathbf{S}(\tau) (\mathbf{Y}(x + \tau) - \mathbf{Y}(x + \delta + \tau))^2 \\ &= (\check{\mathbf{S}} \star \mathbf{\Delta}_{\delta})(x), \end{aligned} \quad (7)$$

where $(\check{\mathbf{S}}(\tau) = \mathbf{S}(-\tau)$, $\mathbf{\Delta}_{\delta}(x) = (\mathbf{Y}(x) - \mathbf{Y}(x + \delta))^2$ and \star is the convolution operator. This term can be interpreted as the correlation between the shape \mathbf{S} and the square difference of the observed image and the δ -shifted version. The convolution $\check{\mathbf{S}} \star \mathbf{\Delta}_{\delta}$ can be computed quickly thanks to the following relation:

$$\check{\mathbf{S}} \star \mathbf{\Delta}_{\delta} = \mathcal{F}^{-1}(\mathcal{F}(\check{\mathbf{S}})\mathcal{F}(\mathbf{\Delta}_{\delta})) = \mathcal{F}^{-1}(\overline{\mathcal{F}(\mathbf{S})}\mathcal{F}(\mathbf{\Delta}_{\delta})), \quad (8)$$

where \mathcal{F} is the 2D discrete Fourier transform (2D-FFT) and \mathcal{F}^{-1} is its inverse transform. According to Equation (8), and given a translation δ , we only need to perform one term by term multiplication in Fourier domain and two 2D-FFT (note that $\mathcal{F}(\mathbf{S})$ can be computed off-line). The repetition of this procedure for every translation δ covering the search window, leads to an algorithm (whose pseudo-code is detailed in Fig. 3) with a complexity of $O(|W| \cdot |\Omega| \cdot \log(|\Omega|))$. A similar algorithm using recursive filters was recently described in [9].

3.2 Families of shapes

The main purpose of this paper is to show that the use of different shapes allows to reduce the *rare patch effect*. This point of view is a generalization of the NL-Means based on square patches with the reprojection studied in [51], since each translated patch can be regarded as a decentered shape. Here, h is fixed and the challenge is to find shapes with enough similar candidates in the search window to reduce the noise.

We now present several types of families we have considered. The first collections consist of classical squares and disks shapes. Then, we propose more directional shapes such as *pie slices* and *bands* displayed in Fig. 2.

Squares: To begin with, we apply our framework to the most commonly used shapes, i.e., the square shapes of odd length (so the squares have centers we can consider). For instance, choosing:

$$\mathbf{S}(\tau) = \begin{cases} 1, & \text{if } \|\tau\|_{\infty} \leq \frac{p-1}{2}, \\ 0, & \text{otherwise,} \end{cases} \quad (9)$$

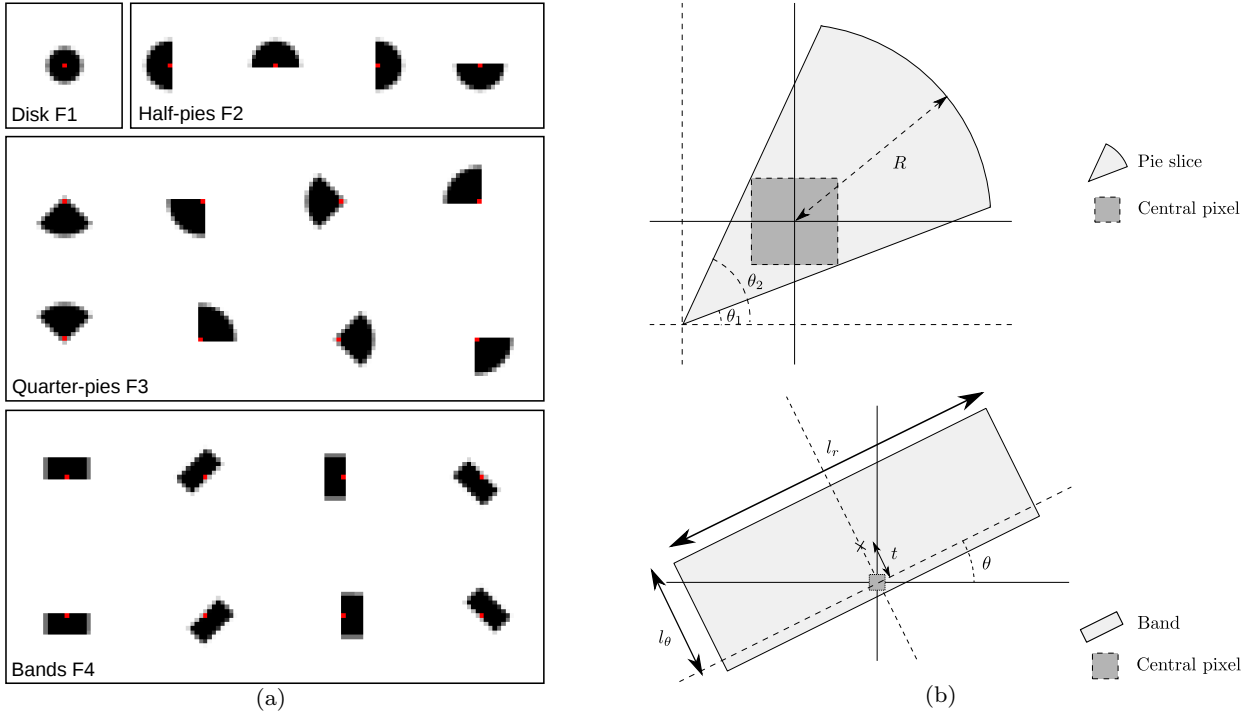


Fig. 2 (a) Examples of shapes considered. The “central” pixel is shown in red, dark pixels illustrate high weights. Shapes are grouped in four categories: F1. the isotropic *disk* family, F2. the *half-pies* family, F3. the *quarter-pies* family and F4. the *bands* family. (b) Parametrization of the *pie* slices and *bands*.

leads to the classical (simplified) NL-Means definition with square patches of size $p \times p$ and distance between patches measured by the Euclidean norm.

Gaussian: The original, but less common choice, is to set:

$$S(\tau) = \begin{cases} \exp(-(\tau_1^2 + \tau_2^2)/2a^2), & \text{if } \|\tau\|_\infty \leq \frac{p-1}{2}, \\ 0, & \text{otherwise.} \end{cases} \quad (10)$$

Equation (10) means that the norm $\|\cdot\|_{2,a}$ is used to measure the distance between patches. This limits the influence of square patches corners and leads to a more isotropic comparison between patches.

Disks: Disk shapes are defined in the same way, using the Euclidean norm instead:

$$S(\tau) = \begin{cases} 1, & \text{if } \|\tau\|_2 \leq \frac{p-1}{2}, \\ 0, & \text{otherwise.} \end{cases} \quad (11)$$

A non-binary version may also be defined for pixels crossed by the boundary.

Pie slices: We study a family of shapes, denoted as “pie”, whose elements are defined with three

parameters: two angles and a radius. These shapes represent a portion of a disk delimited by two lines and surrounding the discrete central pixel.

Bands: This family of shapes is simply composed of rectangles, potentially rotated and decentered with respect to the pixel of interest.

3.3 Connection with previous work

One of our main concern is to address the *rare patch effect* of the NL-Means algorithm. Different methods have been designed to limit this drawback or to improve the NL-Means in terms of quality or speed. The first attempt was proposed to speed-up the algorithm. The idea is to denoise patch by patch rather than pixel by pixel. Taking into account patches overlaps, a fast implementation of the NL-Means is reached by using a sub-sampled grid of pixels [7,28]. Quality improvement can also be obtained by properly using overlapping patches. Indeed, we get $|P|$ estimates for each pixel (where $|P|$ is the number of elements in a patch). Some authors [7,27] simply propose to uniformly average those $|P|$ estimates while a weighted average is performed in [51] (see Section 4.1 and 4.2). In our frame-

Algorithm 2D-FFT NL-Means for an arbitrary shape**Inputs:** noisy image \mathbf{Y} , 2D-FFT of the shape $\mathcal{F}(\mathbf{S})$ **Parameters:** search window W , bandwidth h **Output:** estimated image $\hat{\mathbf{f}}$ Initialize accumulator images \mathbf{A} and \mathbf{B} to zero**for all** shift vector δ in the search window W **do** Compute the square difference image Δ_δ

$$\Delta_\delta(x) := (\mathbf{Y}(x) - \mathbf{Y}(x + \delta))^2 \text{ for all pixels } x \text{ in } \Omega$$

 Compute the 2D-FFT $\mathcal{F}(\Delta_\delta)$ Perform the convolution of Δ_δ by the shape $\tilde{\mathbf{S}}$

$$d_{\tilde{\mathbf{S}}}^2(\cdot, \cdot + \delta) \leftarrow \left(\mathcal{F}^{-1} \left(\overline{\mathcal{F}(\tilde{\mathbf{S})}} \mathcal{F}(\Delta_\delta) \right) \right) (\cdot)$$

 $\triangleright O(|\Omega| \cdot \log |\Omega|)$ operations using 2D-FFT**for all** pixels x in Ω **do**

Compute the weights

$$\omega(x, x + \delta) = \varphi \left(\frac{d_{\tilde{\mathbf{S}}}^2(x, x + \delta)}{2h^2} \right)$$

Update the accumulators

$$\mathbf{A}(x) \leftarrow \mathbf{A}(x) + \omega(x, x + \delta) \mathbf{Y}(x + \delta)$$

$$\mathbf{B}(x) \leftarrow \mathbf{B}(x) + \omega(x, x + \delta)$$

end for**end for**Final (normalized) estimator $\hat{\mathbf{f}}(x) = \frac{\mathbf{A}(x)}{\mathbf{B}(x)}$ for all pixel x Note: the central pixel ($\delta = 0$) is treated as a special case \triangleright see Section 5 for details

Fig. 3 NL-Means pseudo-code for an arbitrary patch shape \mathbf{S} . Pre-computations (based on 2D-FFT) of distances between shapes from the noisy image and shapes from its shifted version leads to a smaller complexity of $O(|W| \cdot |\Omega| \cdot \log |\Omega|)$, independent of the shape \mathbf{S} .

work, these blockwise approaches are equivalent to use $|\mathbf{P}|$, possibly decentered, square shapes (cf. Fig. 4).

Other methods have been introduced to reduce the *noise halo*. In [34], the authors use a total variation-based pre-filtering of the image and set locally its parameter so that the NL-Means find enough similar patches. In [19], the authors aim to select locally the bandwidth parameter h using SURE (introduced in details in Section 4.3) to select enough patches according to a bias-variance trade-off.

4 Aggregation of shape-based estimates

In this section we investigate several ways to aggregate the NL-Means estimators based on different shapes of “patches”. We have extended the standard square shape to other shapes such as *disks*, *pies* or *bands* (see Section

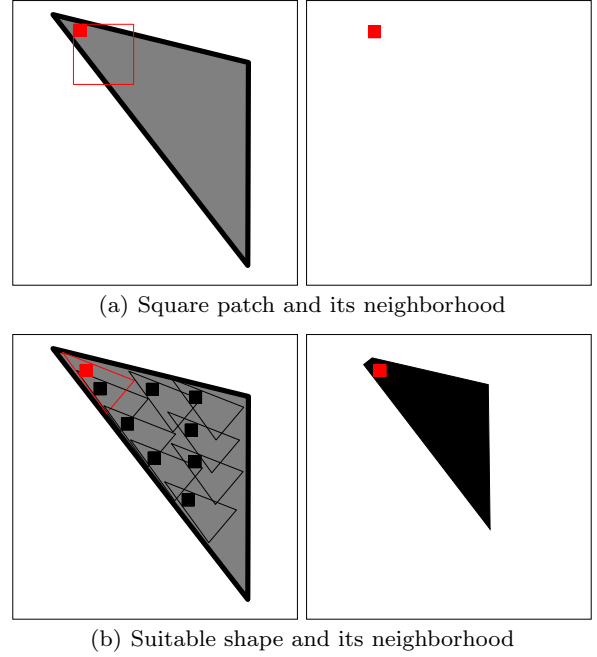


Fig. 4 Examples of neighborhood associated with a square patch or a suitable shape. If the patch is square (a) fewer similar patches candidates are found than with the shape (b). The pixel of interest is in red and the selected pixels obtained by the two methods to denoise the red pixel are in black.

3.2). Thus, the new goal in this context is to determine how to locally take the most of each proposed denoiser.

Assume that for any pixel x in the image, we have built a collection of K pixel estimators $\hat{\mathbf{f}}_1(x), \dots, \hat{\mathbf{f}}_K(x)$ based on different shapes, and that estimates of their corresponding performances are available. With this information at hand, we can address different aggregation/optimization problems. Indeed, the ways of combining the estimators at hand may depend on the theoretical aggregation problem we aim to solve (as described by [39] and [57]):

- (S) The selection problem: finding the best estimator among $\hat{\mathbf{f}}_1(x), \dots, \hat{\mathbf{f}}_K(x)$.
- (L) The linear problem: finding the best linear combination of $\hat{\mathbf{f}}_1(x), \dots, \hat{\mathbf{f}}_K(x)$.
- (C) The convex problem: finding the best convex combination of $\hat{\mathbf{f}}_1(x), \dots, \hat{\mathbf{f}}_K(x)$.

4.1 Uniformly weighted aggregation (UWA)

This is the simplest form of aggregation that we consider. The idea is to give the same weight to any shape-based estimator. With few shapes it is already an improvement in practice (both in term of PSNR and SSIM, see Table 3), but as the number of shapes increases, we can take into account positions that are

completely irrelevant. It is in essence the method proposed in [7] and referred to as blockwise NL-means in [27], with shapes being decentered square patches. Such an aggregation procedure still suffers from the *rare patch effect* (see Fig. 1, second line, for a visual illustration). We can express for any pixel x the uniform estimate $\hat{\mathbf{f}}_{\text{UWA}}(x)$ as the mean of the shape-based estimates:

$$\hat{\mathbf{f}}_{\text{UWA}}(x) := \frac{1}{K} \sum_{k=1}^K \hat{\mathbf{f}}_k(x). \quad (12)$$

In our experiments, this simple and commonly used method can be considered as the benchmark to be improved when using more complex aggregating strategies.

4.2 Variance-based decision, Weighted Average (WAV)

A possible way to limit the halo effect is to adapt WAV-reprojection [51] to general shapes. The idea, also proposed by Dabov *et al.* [11] in a different context, is to perform a weighted average of the estimates $\hat{\mathbf{f}}_1(x), \dots, \hat{\mathbf{f}}_K(x)$. To limit the *noise halo*, each weight should be chosen inversely proportional to the (approximate) variance of the corresponding estimator. In the context of NL-means, this approximate variance can be obtained in closed-form in the same way as in [27], assuming that the coefficients $\omega(x, x')$ in Equation (2) can be treated as deterministic. Measuring the performances of the estimators in term of variance is well justified since the halo effect results to the high variance of our estimators around the edges (see [51]). However, it tends to over-smooth the edges and the thin details since it does not consider the bias of each estimator.

4.3 SURE-based decisions

A way to take the bias into account is to consider the risk estimate rather than the variance to locally attribute more weight to the estimators with small risks. In [58], Van De Ville and Kocher give a closed-form expression of Stein's Unbiased Estimator of the Risk (SURE) for the NL-Means. They aim at selecting globally the best bandwidth for a given image. Here, our approach is different, despite the use of the same tool. Indeed, our choice of shape is done locally (i.e., for each pixel), since it is very unlikely that a single shape should be optimal for a whole natural image. Our method is closer to the one proposed by Duval *et al.* [19]. The authors rely on SURE to locally determine the parameters: the bandwidth h and the patch size p .

They have shown that a local choice of h reduces the visual artifacts, especially the *rare patch effect*. Let us now rephrase Stein's Lemma [54, 55] in our NLM-SAP framework.

Proposition 1 (Stein) *Let $\mathbf{f}(x) \in \mathbb{R}$, $\varepsilon(x) \sim \mathcal{N}(0, \sigma^2)$, and $\mathbf{Y}(x) = \mathbf{f}(x) + \varepsilon(x)$. Denote by $\gamma(\mathbf{Y}(x))$ an estimate of $\mathbf{f}(x)$ and assume $\gamma : \mathbb{R} \rightarrow \mathbb{R}$ is absolutely continuous, and*

- i) $\lim_{|z| \rightarrow \infty} \gamma(\mathbf{f}(x) + z) e^{-\frac{z^2}{2\sigma^2}} = 0$,
- ii) $\mathbb{E}(\gamma(\mathbf{f}(x) + \varepsilon(x)))^2 < +\infty$, and
- iii) $\mathbb{E}|\gamma'(\mathbf{f}(x) + \varepsilon(x))| < +\infty$.

Then the following relation holds:

$$\mathbb{E}|\gamma(\mathbf{Y}(x)) - \mathbf{f}(x)|^2 = \mathbb{E} \left((\gamma(\mathbf{Y}(x)) - \mathbf{Y}(x))^2 + 2\sigma^2 \frac{\partial \gamma(\mathbf{Y}(x))}{\partial \varepsilon(x)} \right) - \sigma^2. \quad (13)$$

Proposition 1 provides an unbiased estimate of the risk for the K shape-based denoised values $\hat{\mathbf{f}}_k(x)$ at each pixel x . Indeed, if γ is one of the shape-based estimators and if the kernel φ is such that γ is absolutely continuous, then

$$\mathbf{r}_k(x) = (\hat{\mathbf{f}}_k(x) - \mathbf{Y}(x))^2 + 2\sigma^2 \frac{\partial \hat{\mathbf{f}}_k(x)}{\partial \varepsilon(x)} - \sigma^2, \quad (14)$$

is an unbiased estimate of the risk at pixel x for the k -th shape-based estimate, i.e., $\mathbb{E}(\mathbf{r}_k(x)) = \mathbb{E}|\hat{\mathbf{f}}_k(x) - \mathbf{f}(x)|^2$.

The main contribution of [58, 19] is that they give a closed-form expression of $\frac{\partial \hat{\mathbf{f}}_k(x)}{\partial \varepsilon(x)}$ for NL-Means. Indeed, thanks to Eq. (2) and Eq. (4), its expression can be recast in the following form:

$$\begin{aligned} \frac{\partial \hat{\mathbf{f}}_k(x)}{\partial \varepsilon(x)} &= \frac{\varphi(0)}{C_x} + \frac{1}{C_x} \sum_{x'} \mathbf{Y}(x') \frac{\partial \omega(x, x')}{\partial \varepsilon(x)} \\ &\quad - \left(\frac{1}{C_x} \sum_{x'} \mathbf{Y}(x') \omega(x, x') \right) \left(\frac{1}{C_x} \sum_{x''} \frac{\partial \omega(x, x'')}{\partial \varepsilon(x)} \right) \end{aligned} \quad (15)$$

where $C_x = \sum_{x'} \omega(x, x')$ is a normalization constant. In our NLM-SAP framework, our shape-based norm defined in Eq. (5) leads to the following expression of the derivative of the weights $\omega(x, x')$:

$$\begin{aligned} \frac{\partial \omega(x, x')}{\partial \varepsilon(x)} &= \frac{1}{h^2} \varphi' \left(\frac{d_{\mathbf{S}}^2(x, x')}{2h^2} \right) (\mathbf{S}(0) [\mathbf{Y}(x) - \mathbf{Y}(x')] \\ &\quad + \mathbf{S}(x - x') [\mathbf{Y}(x) - \mathbf{Y}(2x - x')]) \end{aligned} \quad (16)$$

where \mathbf{S} encodes the shape of our k -th shape-based estimator. Finally, combining equations (14), (15) and (16)

leads to an unbiased estimate of the risk of our NLM-SAP denoiser.

Now, since we are using K shapes, we get $\mathbf{r}_1(x), \dots, \mathbf{r}_K(x)$, unbiased risk estimates respectively for the shape-based estimators $\hat{\mathbf{f}}_1(x), \dots, \hat{\mathbf{f}}_K(x)$.

4.3.1 Minimizing the risk of linear combinations

Suppose we want to solve the linear problem of aggregation. A natural way to find a good linear combination is to solve the following problem:

$$\hat{\mathbf{f}}_{\text{LIN}}(x) := \sum_{k=1}^K \alpha_k^* \hat{\mathbf{f}}_k(x), \quad (17)$$

where $\alpha^* := \arg \min_{\alpha \in \mathbb{R}^K} \mathbb{E} \left(\sum_{k=1}^K \alpha_k \hat{\mathbf{f}}_k(x) - \mathbf{f}(x) \right)^2$.

Here, the linearity of the combination allows to use Stein's Lemma (Proposition 1). It provides an unbiased estimate of the risk of the weighted average estimate:

$$\begin{aligned} \mathbb{E} \left(\sum_{k=1}^K \alpha_k \hat{\mathbf{f}}_k(x) - \mathbf{f}(x) \right)^2 &= \\ \mathbb{E} \left(\left(\sum_{k=1}^K \alpha_k \hat{\mathbf{f}}_k(x) - \mathbf{Y}(x) \right)^2 + 2\sigma^2 \sum_{k=1}^K \alpha_k \frac{\partial \hat{\mathbf{f}}_k(x)}{\partial \varepsilon(x)} \right) &- \sigma^2. \end{aligned}$$

Therefore, neglecting the term that does not depend on α in the last equation, the optimal weights α^* can be obtained by solving the following:

$$\begin{aligned} \alpha^*(x) = \arg \min_{\alpha \in \mathbb{R}^K} \sum_{k,k'} \alpha_k \alpha_{k'} \mathbb{E}(\hat{\mathbf{f}}_k(x) \hat{\mathbf{f}}_{k'}(x)) &+ \\ \sum_{k=1}^K \alpha_k \left(2\sigma^2 \mathbb{E} \left(\frac{\partial \hat{\mathbf{f}}_k(x)}{\partial \varepsilon(x)} \right) - 2\mathbb{E}(\mathbf{Y}(x) \hat{\mathbf{f}}_k(x)) \right). \end{aligned}$$

Using the first order optimality conditions, the last problem amounts to solving a linear system in α . This type of estimator is known to perform quite well for wavelet thresholding estimation where it is referred to as the SURE-LET (for SURE-Linear Expansion of Thresholds, see [4]). However, in our framework, this method is not applicable. Indeed, since we work pixelwise, we have only one sample of the SURE for each pixel. This leads to an ill-conditioned system to solve, and the calculated α^* behaves poorly. Algebraically, this is due to the fact that the matrix $[\hat{\mathbf{f}}_k(x) \hat{\mathbf{f}}_{k'}(x)]_{k,k'=1,\dots,K}$ is of rank one, and thus the system to solve is ill-conditioned. Our attempts to regularize the problem (e.g. with Tikhonov regularization known as “Ridge Regression” in statistics) or to solve the problem blockwise instead of pixelwise did not yield satisfying results. More generally, the problem of using

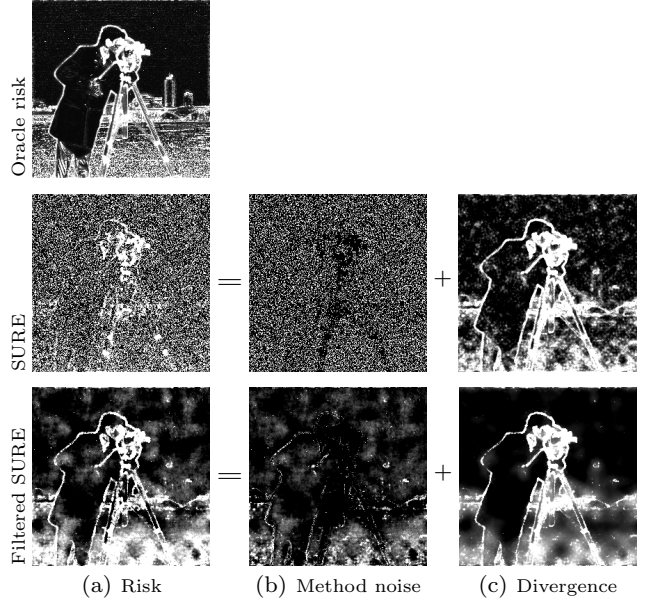


Fig. 5 Maps of the risk associated to a circular shape: (first line) the oracle risk map, (second line) the SURE map without regularization and (third line) the SURE map with Yaroslavsky regularization. Second and third rows correspond to the decomposition (14).

SURE to take a local decision for each pixel x is difficult since this estimator has large oscillations (see for instance Fig. 5). In the next paragraph, we present how to regularize the risk maps, i.e., the “images” $\mathbf{r}_1, \dots, \mathbf{r}_K$.

4.3.2 Regularizing the risk maps

In practice, the estimation of the risk given by (14) is too noisy to guide a local choice of the shape (see Figure 5). To make it more robust, it is necessary to locally regularize the risk maps, so as to approximate at each pixel the expectations used in Eq. (14): our aim is to find estimates $\hat{\mathbf{r}}_k(x)$ close to the true risks $\mathbb{E}[(\hat{\mathbf{f}}_k(x) - \mathbf{f}(x))^2]$. These true risks will be referred to as the oracle risks since in our simulations we will compute them using the true image.

In [19], the convolution of the risk map is an efficient way to estimate the local risk in view of setting h since on both sides of an edge a large value of h should be used. Here, the anisotropy of the shapes implies that on one side of an edge the risk may be low whereas it may be very high on the other side. Since the convolution diffuses the risks on both sides of the edges, any comparison of the risks associated with each shape becomes unstable.

In order to average the risks on each side of edges separately, we have adopted a variant of the Yaroslavsky filter [63]. A natural idea is to use the noisy image $\mathbf{Y}(x)$ to regularize the risk.

The estimate of the risk in (14) is actually considered as a random variable and the SURE map is then a random field that can be decomposed into two terms: a divergence term $\mathbf{D}_k(x) = 2\sigma^2 \frac{\partial \hat{\mathbf{f}}_k(x)}{\partial \boldsymbol{\varepsilon}(x)}$ and the square of the method noise [7]. This divergence term has little variance compared to the noisy image and to the method noise (see Figure 5) and contains all the needed information to guide the averaging process: it is uniformly high in the regions where the halo effect is likely to appear, whereas it is low in smooth regions. As a consequence, we obtain better results by using the following regularized version of the risk:

$$\hat{\mathbf{r}}_k(x) = \frac{1}{C(x)} \sum_{x'} \mathbb{1}_{\{|\mathbf{D}_k(x) - \mathbf{D}_k(x')| \leq \kappa\}} \mathbf{r}_k(x'), \quad (18)$$

where the sum is taken over a small neighborhood of x , $C(x)$ is a normalizing constant and κ is a bandwidth parameter. Fig. 5 shows that this regularization procedure provides smooth risk maps, following edges of the underlying noise-free image, and finer than without regularization. We have displayed the oracle risk map and the estimated risk map provided by SURE and the Yaroslavsky regularization. For illustration purpose, we also show the decomposition of SURE as the sum of the square of the method noise and the divergence term.

Other attempts to regularize the risk map were performed (median filter, variants of Perona-Malik diffusion and NL-means). Yaroslavsky regularization provides the best trade-off between computing time, visual and numerical results, and we have adopted this approach.

4.3.3 Minimizer of the risk estimates (MRE)

With our measure of performance, the most natural way to address the selection problem (S) is to select the shape that minimizes the local risk estimate we have at hand:

$$\hat{\mathbf{f}}_{\text{MRE}}(x) := \hat{\mathbf{f}}_{k^*}(x) \quad \text{where} \quad k^* = \arg \min_k \mathbf{r}_k(x). \quad (19)$$

Such a selection rule is all the more relevant as the shapes are really different. In most cases, selecting the locally optimal shape yields satisfying results, but it is sometimes more appropriate to combine some of the best performing estimators as in the next paragraph.

4.3.4 Exponentially weighted aggregation (EWA)

In many cases, it might be better to combine several estimators rather than just selecting one. Especially, it happens to be relevant if the best estimators (in term of evaluated risk) are diversified enough or if the risk

of the MRE was wrongly under-estimated. Thus, we have also used the statistical method of Exponentially Weighted Aggregation as introduced by Leung and Barron [32]. This method has been theoretically studied in [13, 14] and adapted for patch-based denoising in [50]. It consists in aggregating the estimators by performing a weighted average with weights based on the confidence attributed to each estimator, measured in term of the risk. More precisely:

$$\hat{\mathbf{f}}_{\text{EWA}}(x) := \sum_{k=1}^K \alpha_k \hat{\mathbf{f}}_k(x), \quad (20)$$

$$\text{with} \quad \alpha_k = \frac{\exp(-\mathbf{r}_k(x)/T)}{\sum_{k'=1}^K \exp(-\mathbf{r}_{k'}(x)/T)}.$$

The temperature parameter $T > 0$ is a smoothing parameter, that controls the confidence attributed to the risk estimates. If $T \rightarrow \infty$, then the EWA is simply the uniform aggregate $\hat{\mathbf{f}}_{\text{UWA}}$ defined before. Conversely, when $T \rightarrow 0$, then $\hat{\mathbf{f}}_{\text{EWA}} \rightarrow \hat{\mathbf{f}}_{\text{MRE}}$. Most theoretical works about EWA (see [32, 13, 14]) recommend a large temperature parameter $T = 4\sigma^2$ under few assumptions (like independence) on the estimators $\hat{\mathbf{f}}_1(x), \dots, \hat{\mathbf{f}}_K(x)$. In practice, since assumptions on the estimators family may not be satisfied, we have used a smaller value, $T = 0.4\sigma^2$.

5 Numerical and Visual Results

This section presents quantitative and qualitative results obtained on four images synthetically corrupted by AWGN. Unless otherwise specified, the corrupted images are obtained from four 256×256 noise-free images presented on Fig. 6: the famous *cameraman* image and *city*, *lake* and *windmill*¹. These images are particularly interesting in the study of our proposed Non-Local Means with Adaptive Patch Shapes (NLM-SAP) since they present edges with high contrast for which the classical NL-Means suffer from the *rare patch effect* (see Fig. 1).

In all the experiments, **unless otherwise specified**, the NLM-SAP is used with the following default parameters:

- search window: width $\ell = 11$ px,
- shape family: 15 shapes from families F1 and F2 on Fig. 2.a with shape areas of 12.5, 25 and 50 px²,
- aggregation: EWA with $T = 0.4\sigma^2$,
- risk regularization: Yaroslavsky regularization with search window of size 11×11 px² and κ is proportional to the estimated standard deviation of the divergence map.

¹ three sub-images extracted from Laurent Condat's database: <http://www.greyc.ensicaen.fr/~lcondat/>



Fig. 6 Chosen 256×256 noise-free images for our experiments, from left to right: *cameraman*, *city*, *lake* and *windmill*. These images present edges with high contrast for which the classical NL-Means suffer from the *rare patch effect*.

Table 1 Gain by using multi-scale isotropic shapes in terms of PSNR and SSIM values (PSNR/SSIM). Circular patch shapes respectively with area of 12.5, 25 and 50 px^2 are used and their results are compared to the one obtained when using a combination of these three scales of patches.

	Cameraman	City	Lake	Windmill
Noisy input image	22.13/0.400	22.13/0.567	22.13/0.456	22.13/0.385
Patch shape area of 12.5 px^2	29.59 /0.822	28.11/0.873	28.68/0.849	30.91/0.879
Patch shape area of 25 px^2	29.38/0.828	27.94/0.880	28.46/0.855	30.72/0.895
Patch shape area of 50 px^2	29.06/0.825	27.59/0.879	28.33/0.857	30.35/0.899
Combination of these three scales	29.58/ 0.844	28.32 / 0.897	28.93 / 0.878	31.08 / 0.912

As soon as we consider shapes of different areas, the parameter h has to adapt to the sizes of the shapes. Since we use a trapezoidal kernel, we set the bandwidth parameter to $h^2 = 2\sqrt{8\sigma^4/|S|}$ as suggested in [8] where $|S|$ is the equivalent size of the shape $|S| = \frac{(\sum S(\tau))^2}{\sum S(\tau)^2}$.

The main limitation in computing time is due to the number K of shapes required by our NLM-SAP algorithm. We need to perform K times an NL-Means like algorithm, i.e., one for each shape. Thanks to our FFT acceleration, the computing time required for one shape, whatever the shape, is of about 2s for a 256×256 image with a Matlab implementation on an Intel Pentium 64-bit, 3.00 GHz. By comparison, the naive Matlab implementation of NL-Means takes about 100s, for square patches of area $7 \times 7 \text{ px}^2$. The computation of one local SURE map, using Yaroslavsky filtering, takes about 0.2s per shape. Finally, NLM-SAP² using 15 shapes leads to a computing time of about 32s which is still less than the naive Matlab implementation of NL-Means.

Visual results are given to assess the denoising qualities relative to the different settings of NLM-SAP and to compare NLM-SAP with other denoising approaches (see Fig. 12). Numerical criteria support our claims: the

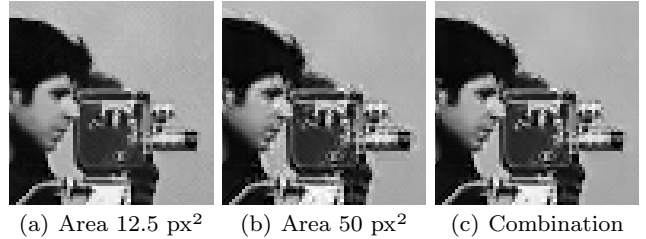


Fig. 7 Results obtained with circular shapes of different scales (12.5 px^2 , 50 px^2 , combination of 12.5, 25 and 50 px^2). The smallest patch size provide best PSNR but has more artifacts in smooth regions, whereas larger ones suffer from the noise halo. Combination of scales limits those two issues.

Peak Signal to Noise Ratio (PSNR) defined by

$$\text{PSNR}(\hat{\mathbf{f}}, \mathbf{f}) = 10 \log_{10} \frac{255^2}{\frac{1}{|\Omega|} \sum_{x \in \Omega} (\mathbf{f}(x) - \hat{\mathbf{f}}(x))^2}, \quad (21)$$

and the Structural Similarity (SSIM) defined in [61]. The SSIM is between 0 and 1 and a value closed to 1 means that the estimated image has a similar structure to the noise-free image.

5.1 Behavior of NLM-SAP

In this section, we will study the behavior of NLM-SAP according to some parameters such as the type of

² our Matlab implementation will be available on-line when this paper is published

Table 2 Gain by using anisotropic or mixture of isotropic and anisotropic shapes in terms of PSNR and SSIM values (PSNR/SSIM). The studied patch shapes are the isotropic *disks*, the *half-pies*, the *quarter-pies*, the *bands* and some combination of them (see Fig. 2.a).

	Cameraman	City	Lake	Windmill
Noisy input image	22.13/0.400	22.13/0.567	22.13/0.385	22.13/0.456
Patches with <i>disk</i> shapes (family F1)	29.58/ 0.844	28.32/ 0.897	28.93/ 0.878	31.08/0.912
Patches with <i>half-pie</i> shapes (family F2)	29.72 /0.843	28.48/0.896	29.00/0.877	31.29/0.912
Patches with <i>quarter-pie</i> shapes (family F3)	29.64/0.842	28.27/0.891	28.89/0.875	31.24/0.912
Patches with <i>band</i> shapes (family F4)	29.72/0.841	28.45/0.894	28.98/0.875	31.36/0.912
Combination of shape families F1 and F2	29.74/ 0.844	28.53 / 0.897	29.04 / 0.878	31.32/ 0.913
Combination of shape families F1, F2, F3 and F4	29.75 /0.842	28.49/0.895	29.02/0.876	31.40 / 0.913

family, the type of aggregation and the type of risk regularization. Each noisy image is corrupted by AWGN with standard deviation $\sigma = 20$.

Table 1 and Figure 7 illustrate the gain of performance to use multi-scale patch shapes instead of using only one fixed size. In this experiment, we consider three circular shapes of areas: 12.5, 25 and 50 px². Comparatively, for the original version of NL-means, the authors suggest to use square patches of fixed size $7 \times 7 = 49$ px². Surprisingly, using the smallest shapes provides always the best PSNR. It means that the bias and the *noise halos* introduced by using large patches are actually more penalizing than the remaining noise left by the use of small patches. The aggregation of these three scales of shapes with our NLM-SAP methods improves slightly the PSNR. Visually speaking, using only small isotropic patches already decreases the *rare patch effect* while using too large isotropic shapes produces a strong *noise halo* effect. However, the level of noise is much more decreased by using large shapes than small ones in homogeneous areas. Finally, combining different scales of isotropic patch shapes leads to a diminution of both the level of noise and the halo effect. This is well reflected in Tab. 1 by the gain in term of the SSIM criterion which provides quality measurements closer to our perception system. We will see in the following that the results can still be improved by considering both multi-scale and anisotropic patch shapes.

Table 2 gives numerical results obtained by using different families. The compared families are the ones presented on Fig. 2.a, i.e., the *disks*, the *half-pies*, the *quarter-pies* and the *bands*. Combination of these families are also studied. Our experiments show that most suitable shape families, both in terms of PSNR and SSIM, have to contain isotropic shapes, directional shapes and various scales of shapes. Increasing the number of shapes does not necessarily improve the denoising quality. Using 15 shapes from families F1 and F2 with the three different scales, seems to be a good trade-off between computing time and denoising quality. Figure 8 illustrates why using directional shapes is important to reduce the *rare patch effect*. Indeed, each oriented



Fig. 8 Eight denoised images obtained for different oriented patch shapes. The proposed final aggregate is in the center. Each denoiser provides good performances in a specific target direction but suffers from *noise halos* in the other directions. The final (central) aggregate takes advantage of every oriented-denoiser to provide high quality restored edges. The patch shape used is indicated in white.

patch shape enables the restoration of edges in the target direction but is inappropriate in the other directions. Then a fine aggregation of them leads to high quality restoration of edges in all directions. Figure 9 displays weight maps induced by using patches with only one fixed square shape (i.e., NL-Means) compared to patches with adaptive scales and orientations (i.e., NLM-SAP). For NL-Means, all the weights are concentrated around the target pixel: this is the *rare patch effect*. For NLM-SAP, the weights are more spread, and other similar pixels are detected thanks to multi-scale and anisotropic patch shapes. It is clear that the limitation of the *rare patch effect* around edges with high contrast leads to a good reduction of the *noise halo*.

Table 3 Comparisons of different aggregation procedures in terms of PSNR and SSIM values (PSNR/SSIM). The compared aggregation types are UWA, WAV, MRE and EWA.

	Cameraman	City	Lake	Windmill
Noisy input image	22.13/0.400	22.13/0.567	22.13/0.385	22.13/0.456
Uniform Weighted Average (UWA)	29.60/0.843	28.29/0.895	29.01/ 0.879	31.04/0.910
Weighted Average based on Variance (WAV)	29.64/0.841	28.15/0.887	28.69/0.868	31.10/0.910
Minimizer of the Risk Estimate (MRE)	29.40/0.838	27.99/0.889	28.34/0.864	30.84/0.907
Exponentially Weighted Average (EWA)	29.74/0.844	28.53/0.897	29.04/0.878	31.32/0.912

Table 4 Comparisons of regularization procedures of the risk maps in terms of PSNR and SSIM values (PSNR/SSIM). The compared regularization procedures are the ones using the noisy risk maps directly (i.e., SURE maps), the convolved risk maps and the risk maps obtained by Yaroslavsky regularization.

	Cameraman	City	Lake	Windmill
Noisy input image	22.13/0.400	22.13/0.567	22.13/0.456	22.13/0.385
Noisy risk maps (SURE maps)	29.13/0.817	27.41/0.865	28.38/0.846	30.15/0.872
Convolved risk maps	29.71/ 0.845	28.49/0.898	29.13/0.881	31.26/ 0.912
Yaroslavsky risk maps	29.74/0.844	28.53/0.897	29.04/0.878	31.32/0.912
True risk maps (MSE maps provided by an oracle)	32.09/0.880	32.31/0.938	32.27/0.922	34.43/0.935

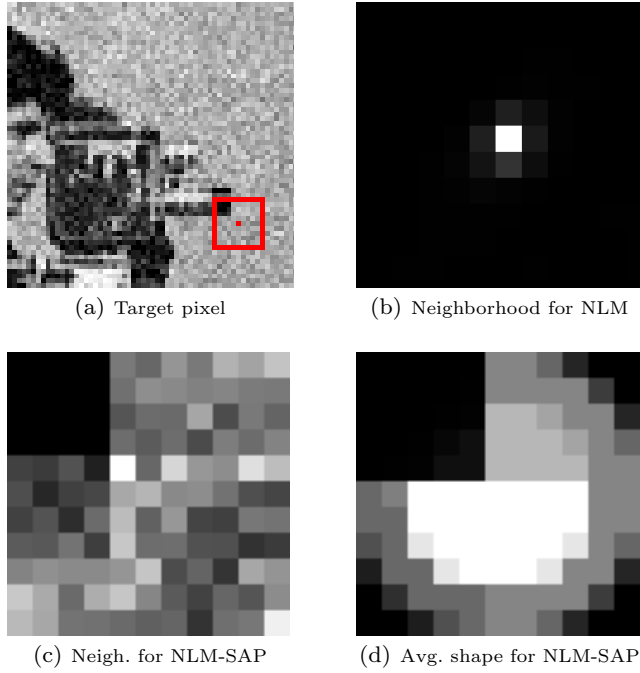
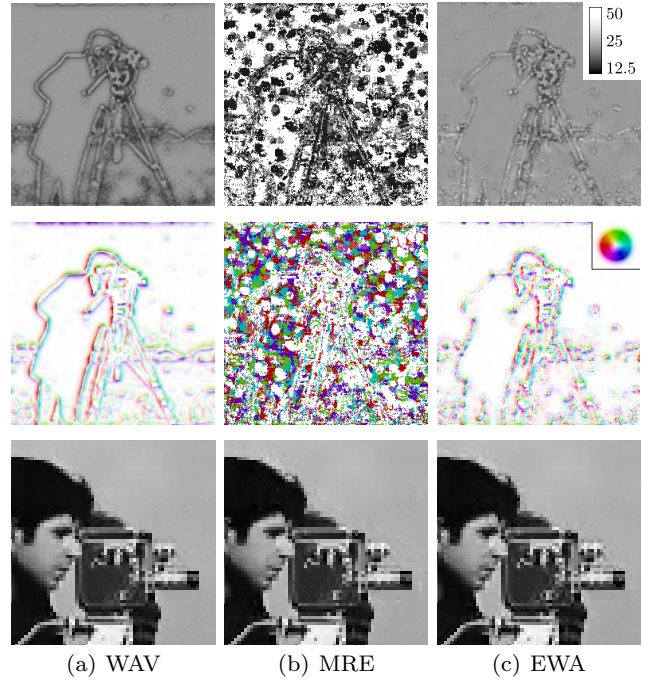
**Fig. 9** Illustration of the diminution of the *rare patch effect*. (a) The noisy image with an highlighted target pixel and its neighborhood. (b) The associated maps of weights obtained by using only square patches of fixed size (i.e., NL-Means). (c) The associated maps of weights with multi-scale and anisotropic patch shapes (i.e., NLM-SAP with F1 and F2). (d) Weighted average of the shapes combined by NLM-SAP.

Table 3 presents the numerical performances associated with four aggregation procedures: UWA, WAV, MRE and EWA. As expected, EWA provides best results, in terms of PSNR and SSIM, since compared to the other three it combines estimates with the best bias-variance trade-off. The local behaviors of NLM-SAP for

**Fig. 10** (top) Average areas and (bottom) average orientations of selected shapes for different aggregation procedures on a noisy realization of the *cameraman* image. From left to right: WAV, MRE and EWA aggregations. The average areas and the average orientations are represented using colors whose legends are given on the top right corners.

WAV, MRE and EWA are presented on Fig. 10. The average areas and the average orientations of the selected shapes are given for the *cameraman* image. It summarizes for all pixels the information of the average shape as given in Fig. 9.d. In smooth regions, anisotropic shapes are not necessarily worse than isotropic ones (like disks or squares). In fact all shapes with the same

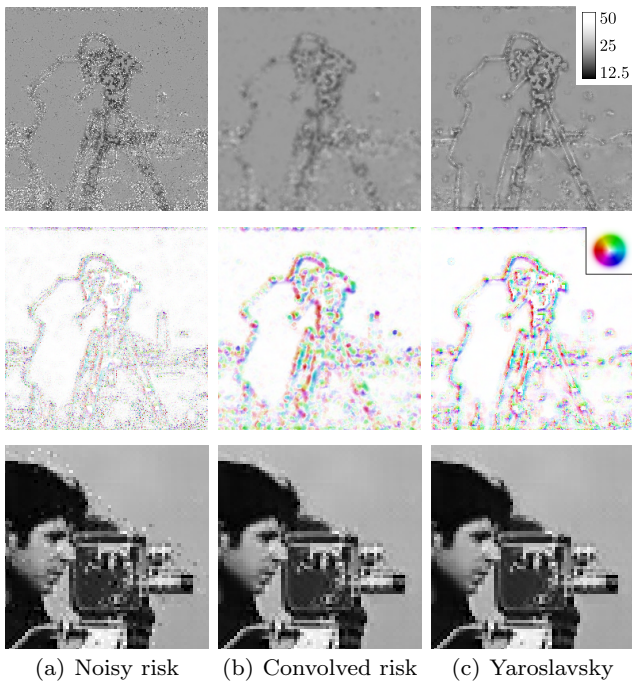


Fig. 11 (top) Average areas and (middle line) average orientations of selected shapes for different risk maps on a noisy realization of the *cameraman* image. (bottom) Corresponding results focused on the cameraman’s neck. From left to right, results using the noisy risk maps (i.e., SURE maps), the convolved risk maps and the risk maps obtained by Yaroslavsky regularization. The average areas and the average orientations are represented using colors whose legends are given on the top right corners.

size should perform equally: weights provided by WAV and EWA are close to uniform distributions. The selected patch shapes clearly adapt to the local scale and orientation of the image geometry. The chosen sizes of the shapes are smaller around edges and textured areas than in homogeneous areas. The chosen orientation follows the orthogonal orientation of the shapes which is consistent with the remarks given in Section 3. Compared to EWA, MRE suffers from brutal transitions, since it selects only one shape per pixel, while EWA evolves in a smoother way due to the weighted combination of shapes for each pixel.

Finally, we have studied the influence of the regularization of the risk maps on the aggregation results. Three methodologies are compared: aggregation using the noisy risk maps (i.e., SURE maps), the convolved risk maps (using a disk kernel of radius 4) and the risk maps obtained by Yaroslavsky regularization. Table 4 gives the corresponding numerical performances and Fig. 11 illustrates the behavior of each type of risk map regularization. The risk maps based on convolution and Yaroslavsky of SURE maps provide the best results in terms of PSNR and SSIM. However, the choice

of the local sizes and orientations of the patch shapes is more relevant in the maps obtained by Yaroslavsky regularization, in terms of scale adaptivity, feature directions and spatial coherency. Using Yaroslavsky filtering, the NLM-SAP acts as expected by selecting big sizes of shapes, even around edges, since the shape orientations have been chosen properly to reduce the *rare patch effect*. By comparison, the convolution forces the size of shapes to be small around edges since it cannot select properly the suitable orientations. This slight differences of behaviors can be noticed around the camera (cf. Figure 11). Other regularization strategies have been investigated (median filter, NL-Means, anisotropic diffusion) but we have not found striking improvements. However, Tab. 4 shows that there is still a gap of numerical performance between regularizations of the risk maps and an “oracle risk map” defined as the image of local square errors associated to each shape: $(\hat{\mathbf{f}}_k(x) - \mathbf{f}(x))^2$.

5.2 Comparisons with *state-of-the-art* methods

In this section, the proposed NLM-SAP approach is compared to *state-of-the-art* denoising methods on a large dataset of standard images at different noise levels σ . Comparisons have been performed with the classical (pixelwise) NL-Means, the blockwise NL-Means using either UWA or WAV reprojection (with square patches of a single scale) the pixelwise NL-means using SURE-based adaptive bandwidth selection [19], a refinement of the NL-Means by Goossens *et al.* [23], the Block-Matching and 3D filtering (BM3D) denoiser [11], and our proposed NLM-SAP approach.

Table 5 shows that NLM-SAP brings a gain of PSNR of about 1 dB compared to the classical NL-Means (for $\sigma \leq 20$). The SSIM is also usually increased. The BM3D approach leads to better numerical results than all Non Local based approaches. Figure 12 gives the visual results. While the blockwise NL-Means using UWA aggregation and the refinement of the NL-Means [23] illustrate the *rare patch effect* by the presence of *noise halos*, BM3D and NLM-SAP have reduced a lot this phenomenon. Our NLM-SAP provides smooth results with accurate details, such as the cameraman’s head, the house windows, the windmill blades, the tree-trunk and the car. Visually, the quality of images obtained with NLM-SAP challenges those obtained with BM3D.

Table 5 Comparisons of denoising approaches for different degradation levels in terms of PSNR and SSIM values (PSNR/SSIM). The compared methods are the classical (pixelwise) NL-Means [7], the blockwise NL-Means using UWA reprojection for square patches the blockwise NL-Means using WAV reprojection for square patches, the pixelwise NL-means using SURE-based adaptive bandwidth selection [19], the BM3D denoiser [11], and our proposed NLM-SAP approach.

	NL-Means	UWA NLM	WAV NLM	Duval <i>et al.</i>	Goossens <i>et al.</i>	BM3D	NLM-SAP
$\sigma = 5$							
barbara	36.04/0.986	36.28/0.987	36.48/0.986	37.06/0.986	37.95/0.988	38.15/0.989	36.94/0.986
boat	35.41/0.983	35.61/0.983	35.76/0.983	36.13/0.982	36.92/0.985	37.15/0.986	36.34/0.984
bridge	34.33/0.958	34.37/0.958	34.43/0.959	34.68/0.961	35.21/0.965	35.31/0.967	34.76/0.963
cameraman	36.71/0.949	36.97/0.952	37.29/0.956	37.44/0.956	37.73/0.958	38.14/0.961	37.67/0.957
city	35.65/0.963	35.92/0.965	36.32/0.971	36.69/0.974	36.43/0.971	37.45/0.978	37.14/0.976
couple	35.73/0.983	35.98/0.984	36.21/0.984	36.47/0.983	37.03/0.985	37.35/0.986	36.70/0.983
fingerprint	34.15/0.997	34.16/0.997	34.16/0.997	34.81/0.996	36.48/0.997	36.38/0.997	34.41/0.997
flinstones	34.85/0.990	35.00/0.990	35.15/0.990	35.21/0.990	35.74/0.991	36.05/0.991	35.62/0.991
hill	34.74/0.940	34.83/0.942	34.94/0.943	35.11/0.943	35.73/0.950	35.96/0.952	35.43/0.948
lake	36.46/0.960	36.78/0.963	37.10/0.969	37.19/0.970	37.53/0.971	38.12/0.976	37.77/0.974
lena	37.07/0.981	37.36/0.982	37.56/0.982	37.77/0.981	38.36/0.983	38.52/0.984	37.90/0.981
man	35.87/0.982	36.13/0.983	36.40/0.982	36.76/0.982	37.31/0.984	37.68/0.985	36.98/0.982
mandril	34.85/0.986	34.94/0.987	35.02/0.986	35.65/0.983	37.50/0.988	37.39/0.989	35.41/0.985
windmill	37.75/0.970	38.15/0.972	38.75/0.978	39.27/0.979	38.75/0.978	39.87/0.983	39.43/0.980
$\sigma = 10$							
barbara	32.23/0.969	32.61/0.971	32.89/0.970	33.85/0.970	34.29/0.973	34.90/0.977	33.69/0.970
boat	32.00/0.956	32.41/0.958	32.61/0.955	32.77/0.955	33.21/0.962	33.85/0.967	32.99/0.953
bridge	29.08/0.884	29.23/0.889	29.40/0.891	29.70/0.887	30.46/0.904	30.66/0.906	30.03/0.896
cameraman	32.13/0.909	32.47/0.913	32.85/0.921	33.11/0.920	33.52/0.926	34.05/0.930	33.50/0.923
city	30.60/0.922	31.00/0.926	31.49/0.941	32.07/0.948	32.01/0.943	33.14/0.955	32.73/0.952
couple	31.99/0.952	32.41/0.954	32.67/0.952	32.81/0.955	33.25/0.959	33.93/0.967	33.07/0.948
fingerprint	28.77/0.988	28.83/0.988	28.95/0.988	30.66/0.986	32.14/0.990	32.41/0.990	30.44/0.988
flinstones	30.33/0.976	30.71/0.978	31.07/0.977	31.50/0.977	31.68/0.978	32.40/0.980	31.85/0.978
hill	30.32/0.859	30.66/0.869	30.96/0.871	30.93/0.863	31.43/0.877	31.85/0.883	31.49/0.871
lake	31.64/0.919	32.10/0.924	32.42/0.936	32.56/0.936	32.95/0.940	33.62/0.949	33.22/0.943
lena	34.08/0.962	34.47/0.964	34.65/0.962	34.81/0.961	35.34/0.965	35.79/0.969	35.00/0.959
man	32.14/0.953	32.53/0.955	32.75/0.951	32.87/0.951	33.34/0.958	33.90/0.963	33.20/0.949
mandril	30.11/0.954	30.26/0.955	30.39/0.952	31.29/0.950	32.73/0.960	33.09/0.966	31.11/0.948
windmill	33.00/0.938	33.48/0.941	34.06/0.953	35.05/0.957	34.62/0.958	35.81/0.966	35.24/0.958
$\sigma = 20$							
barbara	29.87/0.936	30.30/0.939	30.31/0.937	30.62/0.939	30.95/0.946	31.76/0.953	30.41/0.930
boat	29.29/0.892	29.63/0.893	29.55/0.886	29.59/0.897	29.92/0.902	30.81/0.927	29.67/0.877
bridge	25.68/0.739	26.11/0.756	26.17/0.743	25.89/0.738	26.20/0.761	26.76/0.775	26.24/0.728
cameraman	28.59/0.823	29.01/0.831	29.23/0.838	29.58/0.856	29.49/0.852	30.34/0.871	29.74/0.844
city	26.58/0.863	27.07/0.868	27.30/0.877	27.85/0.893	28.00/0.893	29.06/0.912	28.53/0.897
couple	29.03/0.892	29.42/0.895	29.41/0.889	29.25/0.893	29.82/0.903	30.67/0.927	29.37/0.877
fingerprint	26.48/0.958	26.94/0.960	27.16/0.958	27.20/0.957	27.75/0.965	28.80/0.972	27.45/0.951
flinstones	27.19/0.958	27.70/0.959	28.03/0.955	28.92/0.961	28.54/0.962	29.55/0.966	29.04/0.960
hill	27.50/0.733	27.86/0.745	27.78/0.735	27.62/0.741	27.99/0.756	28.51/0.779	27.83/0.724
lake	27.78/0.854	28.23/0.861	28.18/0.865	28.48/0.872	28.82/0.881	29.38/0.894	29.04/0.878
lena	31.61/0.926	31.99/0.928	31.95/0.924	31.67/0.922	32.37/0.932	32.98/0.940	31.92/0.918
man	29.34/0.886	29.64/0.887	29.54/0.880	29.55/0.889	29.81/0.895	30.52/0.915	29.62/0.872
mandril	27.02/0.869	27.31/0.872	27.35/0.864	27.33/0.867	28.22/0.884	29.04/0.910	27.45/0.846
windmill	29.36/0.883	29.94/0.890	30.18/0.901	31.44/0.917	30.85/0.917	32.06/0.935	31.32/0.912
$\sigma = 40$							
barbara	26.65/0.855	26.78/0.858	26.67/0.856	26.86/0.866	27.49/0.879	28.04/0.896	26.04/0.831
boat	26.26/0.781	26.38/0.784	26.21/0.779	26.27/0.789	26.74/0.800	27.64/0.848	26.06/0.762
bridge	23.01/0.529	23.14/0.540	23.02/0.533	23.16/0.568	23.15/0.552	23.98/0.615	22.91/0.511
cameraman	25.89/0.707	26.11/0.724	25.88/0.724	26.29/0.768	26.62/0.766	27.26/0.801	26.28/0.749
city	23.50/0.733	23.75/0.746	23.42/0.736	23.77/0.766	24.04/0.779	25.25/0.829	23.70/0.747
couple	25.73/0.769	25.87/0.772	25.75/0.767	25.67/0.775	26.33/0.792	27.43/0.849	25.39/0.737
fingerprint	23.45/0.855	23.78/0.859	23.72/0.856	23.93/0.885	24.08/0.878	25.27/0.926	23.07/0.817
flinstones	24.76/0.903	24.86/0.898	24.63/0.890	25.29/0.915	25.35/0.916	26.07/0.933	25.02/0.888
hill	24.57/0.555	24.73/0.568	24.67/0.564	24.63/0.583	24.97/0.590	25.87/0.659	24.43/0.540
lake	24.71/0.715	24.88/0.732	24.67/0.727	24.99/0.759	25.25/0.774	25.86/0.805	24.79/0.742
lena	28.31/0.846	28.51/0.850	28.42/0.848	28.37/0.853	29.14/0.868	29.77/0.883	28.22/0.839
man	26.42/0.779	26.52/0.781	26.45/0.778	26.46/0.785	26.83/0.792	27.57/0.833	26.30/0.761
mandril	23.75/0.678	23.83/0.677	23.74/0.671	23.98/0.707	24.16/0.705	25.22/0.789	23.39/0.633
windmill	26.41/0.769	26.59/0.787	26.36/0.789	27.46/0.837	27.23/0.839	28.04/0.878	26.65/0.815



Fig. 12 Comparisons of the visual denoising performances of the proposed NLM-SAP approach and other *state-of-the-art* methodologies ($\sigma = 20$). From top to bottom, the input noisy images, the results obtained by the classical (pixelwise) NL-Means [7], the blockwise NL-Means using UWA reprojection for square patches the BM3D denoiser [11], and our proposed NLM-SAP approach.

5.3 Conclusions on the experiments

The best results were obtained by using the NLM-SAP with the default parameters given in the previous section. We have noticed that another version of the algorithm could be appealing to users particularly interested in fast and simple implementation. It consists in replacing the EWA aggregation with a WAV aggregation using only three scales of isotropic shapes. The latter aggregation scheme is parameter-free and its computing time is shorter, at the price of a slight decrease of the PSNR. Both versions will be available on the authors' websites.

6 Conclusion

In this paper, we have addressed the problem of the *rare patch effect* arising in the NL-Means procedure and responsible of the noisy halos created around edges with high contrast. The proposed solution consists in substituting the square patches of fixed size by spatially adaptive patch shapes. A fast implementation of NL-Means, based on FFT calculations, has been proposed in this context to handle any kind of patch shape with arbitrary scale. Thanks to this acceleration, different estimates are obtained by using different patch shapes, typically one isotropic patch shape and four edge oriented patch shapes, all of them with three different scales. We have extended SURE-based approaches to aggregate properly these different shape-based estimates in a spatially adaptive way. To get an efficient locally adaptive filter, we have shown that the SURE-based risk maps require to be regularized and that Yaroslavsky regularization can be used to this purpose. Simulations have shown that exponentially weighted aggregation based on the regularized risk maps of the different shape-based estimates could lead to both numerical and visual improvements (the *noise halo* is suppressed around edges). Visually, our method challenges other NL-Means improvements we have considered in our comparisons. It is still out-performed by BM3D in terms of PSNR and SSIM.

Future work could be to reduce computation time of the algorithm by choosing more convenient shapes or parallel implementations. Another extension might be to apply our framework to more general kinds of noise, by adapting results given by [24,47] on unbiased risk estimates for general exponential families.

Acknowledgements The authors would like to thanks Jean-François Aujol, Loïc Denis, Yann Gousseau, Erwan Le Pennec and Florence Tupin for their interesting comments and criticisms.

References

1. Aharon, M., Elad, M., Bruckstein, A.: K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans. Signal Process.* **54**(11), 4311–4322 (2006)
2. Awate, S.P., Whitaker, R.T.: Unsupervised, information-theoretic, adaptive image filtering for image restoration. *IEEE Trans. Pattern Anal. Mach. Intell.* **28**(3), 364–376 (2006)
3. Bilcu, R.C., Vehvilainen, M.: Combined non-local averaging and intersection of confiden intervals for image denoising. In: *ICIP*, pp. 1736–1739 (2008)
4. Blu, T., Luisier, F.: The SURE-LET approach to image denoising. *IEEE JIP* **16**(11), 2778–2786 (2007)
5. Boulanger, J., Kervrann, C., Bouthemy, P., Elbau, P., Sibarita, J.B., Salamero, J.: Patch-based nonlocal functional for denoising fluorescence microscopy image sequences. *IEEE transactions on medical imaging* **29**(2), 442–454 (2010)
6. Brox, T., Kleinschmidt, O., Cremers, D.: Efficient nonlocal means for denoising of textural patterns. *IEEE Trans. Image Process.* **17**(7), 1083–1092 (2008)
7. Buades, A., Coll, B., Morel, J.M.: A review of image denoising algorithms, with a new one. *Multiscale Model. Simul.* **4**(2), 490–530 (2005)
8. Buades, A., Coll, B., Morel, J.M.: Non-local means denoising. *Image Processing on Line* (2009). URL http://www.ipol.im/pub/alg/bcm_non_local_means_denoising/
9. Condat, L.: A Simple Trick to Speed Up and Improve the Non-Local Means. submitted (2010)
10. Crow, F.C.: Summed-area tables for texture mapping. In: *SIGGRAPH*, pp. 207–212 (1984)
11. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.O.: Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Trans. Image Process.* **16**(8), 2080–2095 (2007)
12. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.O.: BM3D image denoising with shape-adaptive principal component analysis. In: *Proc. Workshop on Signal Processing with Adaptive Sparse Structured Representations (SPARS'09)* (2009)
13. Dalalyan, A.S., Tsybakov, A.B.: Aggregation by exponential weighting, sharp oracle inequalities and sparsity. In: *COLT*, pp. 97–111 (2007)
14. Dalalyan, A.S., Tsybakov, A.B.: Aggregation by exponential weighting, sharp pac-bayesian bounds and sparsity. *Mach. Learn.* **72**(1-2), 39–61 (2008)
15. Darbon, J., Cunha, A., Chan, T.F., Osher, S., Jensen, G.J.: Fast nonlocal filtering applied to electron cryomicroscopy. In: *ISBI*, pp. 1331–1334 (2008)
16. Deledalle, C.A., Denis, L., Tupin, F.: Iterative weighted maximum likelihood denoising with probabilistic patch-based weights. *IEEE Trans. Image Process.* **18**(12), 2661–2672 (2009)
17. Donoho, D.L., Johnstone, I.M.: Adapting to unknown smoothness via wavelet shrinkage. *J. Amer. Statist. Assoc.* **90**(432), 1200–1224 (1995)
18. Doré, V., Cheriet, M.: Robust NL-Means Filter With Optimal Pixel-Wise Smoothing Parameter for Statistical Image Denoising. *IEEE Trans. Signal Process.* **57**, 1703–1716 (2009)
19. Duval, V., Aujol, J.F., Gousseau, Y.: On the parameter choice for the non-local means. *Tech. Rep. hal-00468856, HAL* (2010)

20. Foi, A., Katkovnik, V., Egiazarian, K.O.: Pointwise shape-adaptive DCT for high-quality denoising and de-blocking of grayscale and color images. *IEEE Trans. Image Process.* **16**(5), 1395–1411 (2007)
21. Gilboa, G., Osher, S.: Nonlocal operators with applications to image processing. *Multiscale Model. Simul.* **7**(3), 1005–1028 (2008)
22. Goldenshluger, A., Nemirovski, A.S.: On spatially adaptive estimation of nonparametric regression. *Math. Methods Statist.* **6**(2), 135–170 (1997)
23. Goossens, B., Luong, H.Q., Pizurica, A., Philips, W.: An improved non-local denoising algorithm. In: *LNLA*, pp. 143–156 (2008)
24. Hudson, H.M.: A natural identity for exponential families with applications in multiparameter estimation. *Ann. Statist.* **6**(3), 473–484 (1978)
25. Katkovnik, V., Foi, A., Egiazarian, K.O., Astola, J.T.: Directional varying scale approximations for anisotropic signal processing. In: *EUSIPCO*, pp. 101–104 (2004)
26. Katkovnik, V., Foi, A., Egiazarian, K.O., Astola, J.T.: From local kernel to nonlocal multiple-model image denoising. *Int. J. Comput. Vision* **86**(1), 1–32 (2010)
27. Kervrann, C., Boulanger, J.: Optimal spatial adaptation for patch-based image denoising. *IEEE Trans. Image Process.* **15**(10), 2866–2878 (2006)
28. Kervrann, C., Boulanger, J., Coupé, P.: Bayesian non-local means filter, image redundancy and adaptive dictionaries for noise removal. In: *SSVM*, vol. 4485, pp. 520–532 (2007)
29. Le Pennec, E., Mallat, S.: Sparse geometric image representations with bandelets. *IEEE Trans. Image Process.* **14**(4), 423–438 (2005)
30. Lee, J.S.: Digital image smoothing and the sigma filter. *Computer Vision, Graphics, and Image Processing* **24**(2), 255–269 (1983)
31. Lepski, O.V., Mammen, E., Spokoiny, V.G.: Optimal spatial adaptation to inhomogeneous smoothness: an approach based on kernel estimates with variable bandwidth selectors. *Ann. Statist.* **25**(3), 929–947 (1997)
32. Leung, G., Barron, A.R.: Information theory and mixing least-squares regressions. *IEEE Trans. Inf. Theory* **52**(8), 3396–3410 (2006)
33. Li, K.C.: From Stein’s unbiased risk estimates to the method of generalized cross validation. *Ann. Statist.* **13**(4), 1352–1377 (1985)
34. Louchet, C., Moisan, L.: Total variation as a local filter. To appear (2010)
35. Mahmoudi, M., Sapiro, G.: Fast image and video denoising via nonlocal means of similar neighborhoods. *IEEE Signal Process. Lett.* **12**, 839–842 (2005)
36. Mairal, J., Bach, F., Ponce, J., Sapiro, G., Zisserman, A.: Non-local sparse models for image restoration. *ICCV* (2009)
37. Mairal, J., Sapiro, G., Elad, M.: Learning multiscale sparse representations for image and video restoration. *Multiscale Model. Simul.* **7**(1), 214–241 (2008)
38. Mallows, C.L.: Some comments on *c.p.* *Technometrics* **15**(4), 661–675 (1973)
39. Nemirovski, A.S.: Topics in non-parametric statistics, *Lecture Notes in Math.*, vol. 1738. Springer, Berlin (2000)
40. Nikolova, M.: Local strong homogeneity of a regularized estimator. *SIAM Journal on Applied Mathematics* **61**(2), 633–658 (2000)
41. Pang, C., Au, O., Dai, J., Yang, W., F.Zou: A fast nl-means method in image denoising based on the similarity of spatially sampled pixels. In: *MMSP* (2009)
42. Perona, P., Malik, J.: Scale space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Anal. Mach. Intell.* **12**, 629–639 (1990)
43. Polzehl, J., Spokoiny, V.G.: Adaptive weights smoothing with applications to image restoration. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **62**(2), 335–354 (2000)
44. Polzehl, J., Spokoiny, V.G.: Propagation-separation approach for local likelihood estimation. *Probab. Theory Related Fields* **135**(3), 335–362 (2006)
45. Portilla, J., Strela, V., Wainwright, M., Simoncelli, E.P.: Image denoising using scale mixtures of gaussians in the wavelet domain. *IEEE Trans. Image Process.* **12**(11), 1338–1351 (2003)
46. Ramani, S., Blu, T., Unser, M.: Monte-Carlo SURE: a black-box optimization of regularization parameters for general denoising algorithms. *IEEE Trans. Image Process.* **17**(9), 1540–1554 (2008)
47. Raphan, M., Simoncelli, E.P.: Learning to be Bayesian without supervision. In: *NIPS*, vol. 19, p. 1145 (2007)
48. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. *Phys. D* **60**(1–4), 259–268 (1992)
49. Salmon, J.: On two parameters for denoising with Non-Local Means. *IEEE Signal Process. Lett.* **17**, 269–272 (2010)
50. Salmon, J., Le Pennec, E.: NL-Means and aggregation procedures. In: *ICIP*, pp. 2977–2980 (2009)
51. Salmon, J., Strobecki, Y.: From patches to pixels in non-local methods: Weighted-Average reprojection. In: *ICIP* (2010)
52. Solo, V.: A sure-fired way to choose smoothing parameters in ill-conditioned inverse problems. In: *ICIP*, vol. 3, pp. 89–92 (1996)
53. Starck, J.L., Candès, E.J., Donoho, D.L.: The curvelet transform for image denoising. *IEEE Trans. Image Process.* **11**(6), 670–684 (2002)
54. Stein, C.M.: Estimation of the mean of a multivariate distribution. In: *Proc. Prague Symp. Asymptotic Statist.* (1973)
55. Stein, C.M.: Estimation of the mean of a multivariate normal distribution. *Ann. Statist.* **9**(6), 1135–1151 (1981)
56. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: *ICCV*, pp. 839–846 (1998)
57. Tsybakov, A.B.: Optimal rates of aggregation. In: *COLT*, pp. 303–313 (2003)
58. Van De Ville, D., Kocher, M.: SURE-based Non-Local Means. *IEEE Signal Process. Lett.* **16**, 973–976 (2009)
59. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. *CVPR* **1**, 511–518 (2001)
60. Wang, J., Guo, Y.W., Ying, Y., Liu, Y.L., Peng, Q.S.: Fast non-local algorithm for image denoising. In: *ICIP*, pp. 1429–1432 (2006)
61. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Signal Process.* **13**(4), 600–612 (2004)
62. Wasserman, L.: All of Nonparametric Statistics (Springer Texts in Statistics). Springer (2007)
63. Yaroslavsky, L.P.: Digital picture processing, *Springer Series in Information Sciences*, vol. 9. Springer-Verlag, Berlin (1985)