SIAM J. IMAGING SCIENCES
Vol. 4, No. 2, pp. 760–788

A Bias-Variance Approach for the Nonlocal Means*

Vincent Duval†, Jean-François Aujol‡, and Yann Gousseau†

Abstract. This paper deals with the parameter choice for the nonlocal means (NLM) algorithm. After basic computations on toy models highlighting the bias of the NLM, we study the bias-variance trade-off of this filter so as to highlight the need of a local choice of the parameters. Relying on Stein's unbiased risk estimate, we then propose an efficient algorithm to locally set these parameters, and we compare this method with the NLM with optimal global parameter.

Key words. nonlocal means, local bandwidth, Stein's unbiased risk estimate, bias-variance

AMS subject classification. 68U10

DOI. 10.1137/100790902

1. Introduction.

In recent years, patch-based methods have drawn a lot of attention in the image processing community. Inspired by the work of Efros and Leung [14] for the texture synthesis problem, Buades, Coll, and Morel have proposed the nonlocal means (or NLM) denoising algorithm in their seminal paper [7]. Their idea (also suggested independently in [1]) is to take advantage of self-similarities in images by comparing local neighborhoods (the "patches") across the whole image. Each pixel value is estimated as a weighted average of all others, and the pixels whose neighborhoods are the most similar to the neighborhood of the pixel to be denoised are given the largest weights. This novel point of view has stimulated many authors and is at the core of most of the state-of-the-art image denoising algorithms.

The aim of this paper is to discuss the choice of parameters of the standard NLM filter. Our theoretical study advocates for a local choice of the parameters within the image. Numerically, we address the problem of automatic parameter tuning by resorting to Stein's unbiased risk estimate (SURE) [31], which is a popular method of estimation of the risk, notably in the wavelet community [12], and which was recently introduced in the context of NLM by [37]. We propose an algorithm which automatically computes the best local parameters in the image so that we can propose a spatially adaptive version of the NLM filter.

To discuss the tuning of parameters of the NLM, we interpret this choice as a bias-variance dilemma. Contrary to the approach of [18], which also relies on a bias-variance analysis, we

*Received by the editors April 1, 2010; accepted for publication (in revised form) April 5, 2011; published electronically June 28, 2011. This work was supported by the grant FREEDOM (ANR07-JCJC-0048-01), "Movies, restoration and missing data."

http://www.siam.org/journals/siims/4-2/79090.html

†Telecom ParisTech, CNRS UMR 5141, F-75013 Paris, France (vincent.duval@telecom-paristech.fr, yann.gousseau@telecom-paristech.fr). The first author's research was supported by the French "Agence Nationale de la Recherche" (ANR) under grant NATIMAGES (ANR-08-EMER-009), "Adaptivity for natural images and texture representations."

‡IMB, Université Bordeaux 1, 351, cours de la Libération, F-33405 Talence cedex, France (Jean-Francois.Aujol@math.u-bordeaux1.fr). This author's research was supported by the French "Agence Nationale de la Recherche" (ANR) under grant NATIMAGES (ANR-08-EMER-009), "Adaptivity for natural images and texture representations."

focus mainly on the choice of the smoothing parameter rather than on the search window.

Notation. We consider an image u defined on a domain $\Omega \subset \mathbb{Z}^2$. Given an odd number s and a pixel $x \in \Omega$, we define the patch $U(x)$ of width s centered at x as the s^2-dimensional vector whose coordinates are the gray values of the pixels in a square neighborhood of x with side s:

$$U(x) = (u(x+j))_{|j|_\infty \leq \frac{s-1}{2}}. \tag{1.1}$$

The NLM filter compares patches in the image u to restore the value at pixel x according to the following formula:

$$NLu(x) = \sum_{y\in\Omega} w(x,y)u(y) = \frac{\sum_{y\in\Omega} e^{-\frac{\|U(x)-U(y)\|^2}{2h^2}} u(y)}{\sum_{y'\in\Omega} e^{-\frac{\|U(x)-U(y')\|^2}{2h^2}}}. \tag{1.2}$$

The ℓ^2 norm we use to compare the patches is *normalized:*

$$\|U(x) - U(y)\|^2 = \frac{1}{s^2} \sum_{|j|\leq\frac{s-1}{2}} (u(x+j) - u(y+j))^2,$$

so that h is homogeneous to a gray level.

Since the common habit is to restrict the above search for patches to a search window of side-length W around x, the sums in (1.2) may be replaced by sums over all $y \in \Omega$ such that $|x-y|_\infty \leq \frac{W-1}{2}$.

Bias-variance trade-off. Many authors (e.g., [18, 21]) use a χ^2 test to set the parameter h. They accept only patches that are likely to be exact replicas of the one they want to denoise: for instance, they choose the smallest h such that 99% of exact replicas contaminated by the noise are accepted. This leads to a linear relation between h and σ, and the experiments reported in [7, 34, 37] confirm that, in terms of the peak signal-to-noise ratio (PSNR), the best value of h is roughly proportional to σ. Still, we prove in this paper that there is interest in choosing the parameter h depending on the image. First, the rule of selecting $h = C\sigma$ is too rough: the visual difference between the results with the optimal h and the predicted value $C\sigma$ may be noticeable. Second, the optimal h widely varies among the different regions of an image (see section 4).

In fact, an exact replica is not always available (e.g., along contrasted curved edges or on isolated details), and it is sometimes necessary to smooth the image nonetheless, introducing some bias in order to reduce the variance. To fix ideas, let us momentarily assume that the weights are computed on the noise-free image (and thus are *deterministic*). If the variance σ^2 of the noise is small and the patch size is large,[1] this approximation makes sense. Indeed, if $\tilde{U}(z)$ denotes the patches of the noisy image $\tilde{u} = u + n$, with n independent and identically distributed (i.i.d.) Gaussian noise, we have[2] $\|\tilde{U}(x) - \tilde{U}(y)\|^2 \approx \mathbb{E}\|\tilde{U}(x) - \tilde{U}(y)\|^2 = \|U(x) - U(y)\|^2 + 2\sigma^2$.

[1] Roughly, we assume that $\frac{\sigma^2}{s}$ is small compared to the typical square distance $\|U(x) - U(y)\|^2$, which is typically 10^2, as illustrated in Figure 7.

[2] This approximation is not valid for $x = y$. However, the same qualitative conclusions can be drawn by slightly adapting the following discussion. We skip this for the sake of clarity.

The risk of denoising the pixel x is given by

$$\mathbb{E}|NL\tilde{u}(x) - u(x)|^2 = \mathbb{E}|NL\tilde{u}(x) - NLu(x)|^2 + \mathbb{E}|NLu(x) - u(x)|^2 + 2\mathbb{E}\left((NL\tilde{u} - NLu(x))(NLu(x) - u(x))\right).$$

The last term vanishes since $\mathbb{E}(NL\tilde{u}(x) - NLu(x)) = \sum_y \mathbb{E}(n(y))w(x,y) = 0$. The first term is the variance term: it is small when the smoothing parameter h is large. The second one is the bias term: it is small when h is small. Thus, the optimal choice of h is a trade-off between bias and variance. If the bias term of an image stays low for large intervals of h, this image will be easy to denoise with the NLM filter. We call such images *patch regular*. This property corresponds to a condition presented in [33] as a necessary assumption on the signal for NLM to work: *similar patches have similar central pixels*. This *patch regularity* will be clarified in the rest of the paper. Notice that a similar information-theoretic formulation can be found in [1], stating that conditionally to the rest of the patch, the entropy of the law of the central pixel is very low.

In practice we estimate the risk of NLM using SURE (recently introduced in [37] in the context of NLM), so we need not assume that the weights are deterministic.

Organization of the paper. The guiding principle of our study is the above bias-variance trade-off: section 2 illustrates the bias of the filter on toy models without noise. In section 3, we add noise and discuss the bias-variance trade-off in terms of the geometry of the image in the patch space. In section 4, we handle this trade-off locally, first by using an oracle which gives the main guidelines of an efficient denoising, then by relying on *Stein's unbiased risk estimate*, first proposed in [37] in the framework of NLM.

Related works. There are many contributions in the literature concerning the NLM, ranging from a better understanding of the algorithm to the proposition of brand new algorithms.

Variational frameworks for nonlocal image processing have been proposed in [19, 15, 16], in connection with total variation–based denoising. Let us also mention the work of Brox and Cremers [5] and Azzabou et al. [3] based on similar variational considerations. In addition, let us mention the combination of total variation filtering techniques and NLM proposed in [21] which reduces the visual artifacts of NLM, as in the present paper.

Generally, variational problems are solved using iterative methods. Since nonlocal filters require a lot of calculations, reducing their computation time is a crucial matter, not only for iterative denoising methods but also for real-time denoising, and several authors have proposed fast implementations of the NLM, including [10, 6].

Another point of view consists of studying images and the NLM filter in the patch space: in [32] and [36], the NLM filter is interpreted as one step of a heat equation in the patch space, and the connection with classical diffusion-based denoising algorithms is established. The analysis of the nonlocal heat equation is carried further in [23, 29], where a whole point of view on spectral analysis on manifolds is developed. In [24], Peyré shows that for several models of signals (e.g., smooth or cartoon), images lie on a manifold in the patch space, and he deduces restoration algorithms in that framework.

One may adopt a more statistical framework: in [17], the NLM is extended to deal with colored noise, and the choice of the weight function is discussed. In [2], the authors propose an adaptive denoising algorithm based on marginal posterior modes estimation, whereas

Deledalle, Denis, and Tupin [11] introduce an iterative algorithm based on maximum likelihood estimation which allows one to deal with noises other than Gaussian, e.g., multiplicative speckle noise. Moreover, a new light is shed on the classical NLM by Salmon and Le Pennec [27], who interpret it as the aggregation of estimators in a PAC-Bayesian approach. Also, statistical approaches have been proposed to select the parameters: in [13], a local selection of the parameter h is performed using the C_p statistic introduced by Mallows, and in [37] SURE is used to select a global optimal parameter.

In terms of performance, some of the best denoising results are obtained, to the best of our knowledge, by three different approaches that rely on nonlocality and patch-denoising: the BM3D algorithm, which is built on classical signal processing tools, proposed by [9], the one in [22] which is based on learning dictionaries of patches, and the one in [40] relying on principal component analysis (PCA).

2. Nonlocal means properties with basic examples. The aim of this section is to recap several qualitative properties of the NLM that are known in the literature. We consider toy examples that highlight the following properties:

1. The search window has an impact on the visual quality of the result.
2. A large patch size tends to blur objects.
3. Even periodic images are altered.
4. There is a loss of contrast depending on the occurrence of each pattern.
5. A weight with compact support instead of an exponential allows one to reduce the bias.
6. The less contrasted the details, the more they are degraded, and this relation is highly nonlinear.

The reader who is familiar with these properties may skip this section.

2.1. Isolated detail. Assume we want to denoise an isolated pattern modeled by a single crenel of size $\frac{T}{2}$ and intensity α. If the signal has length $N \gg T$, under assumptions described in the appendix, we have, inside the crenel,

$$NLu(x) = \alpha \frac{1 - e^{-r} - 2e^{-\frac{1}{2}(\frac{T}{2}+1)r} \cosh rx}{(1 - e^{-r}) \left(2 \sum_{j=0}^{\frac{T}{2}} e^{-rj} - 1 + (N - T - 1) e^{-r\frac{T}{2}} \right)}, \tag{2.1}$$

where $r = 2\frac{\alpha^2}{sh^2}$ and $x \in \left[-\frac{T}{4} + \frac{1}{2}, \frac{T}{4} - \frac{1}{2}\right]$ refers to the pixel position ($x = 0$ corresponds to the center of the crenel). The corresponding curve is a catenary that is farther from the crenel as r decreases (see Figure 1). The result when denoising an object depends on the size of the whole image: in the case of an infinite signal (or image) ($N \to +\infty$) we see that $NLu(x) = 0$ for all x! The phenomenon also appears when denoising lines in an image (see Figure 2), but this time the output is $O(\frac{1}{\sqrt{N}})$ instead of $O(\frac{1}{N})$ (where N is the number of pixels).

In practice, one usually restricts the set of patches to a search window around x, so that this dependence on N is changed into a dependence on W (hence point 1). The dependence on W was noted in [18, 13], and we believe this example strengthens their argument. The idea is that, when denoising a small detail, pixels are averaged with *any other pixels*. Because of the exponential function, the weights assigned to the wrong patches are small, but they are nonzero. If these patches are overwhelming, they will have a strong influence.

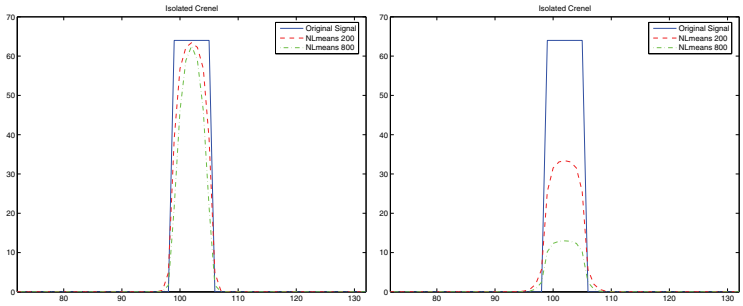


Figure 1. *Loss of isolated details. Left: an extract of a synthetic input signal and the result provided by the NLM filter; it is a catenary inside the crenel (see (2.1)). The size of the crenel is 7, its intensity $\alpha = 64$, the patch size is $s = 7$, $h = 20$. Depending on the total size of the signal ($N = 200$ or $N = 800$), the result does not vary much since $e^{-r\frac{T}{2}} \ll 1$. Right: same experiment, using a patch size $s = 15$. Since the patch size is larger than the pattern, the size of the signal has a large impact on the bias ($e^{-r\frac{T}{2}} \approx 1$).*

Figure 2. *Loss of lines and isolated details. Left: boat image with little noise ($\sigma = 5$). Middle: result of the NLM filter ($h = 6$, $s^2 = 7 \times 7$, search window 11×11). Right: same experiment, using a search window 61×61. Notice that several ropes vanish when the size of the search window increases (this should be seen on a computer screen).*

Two remedies have been proposed: use a small search window W or replace the exponential weights with functions with compact support (i.e., that vanish for $\|U(x)-U(y)\|^2/(2h^2)$ large enough) so that $e^{-r\frac{T}{2}}$ in (2.1) is replaced with zero (see, for instance, [17, 21, 26]). The connection between these methods is discussed in section 3.3. Let us also mention [18], where the bias is locally controlled by choosing W at each pixel.

Incidentally, notice that the larger the patch size, the larger the impact of W on the blurring of the detail (cf. point 2 and Figure 1).

2.2. Periodic crenel. The reader who is not familiar with the NLM filter might think that it is able to restore any periodic signal arbitrarily well. However, it can be shown that the only functions that are invariant by NLM are constant (a direct proof is given in [15]), so that even periodic signals must be altered. As a variation of the previous example, let us consider a quickly oscillating texture which can be modeled by a periodic crenel with period $T \ll s$ and intensity α. Under assumptions given in the appendix, the output of the filter is

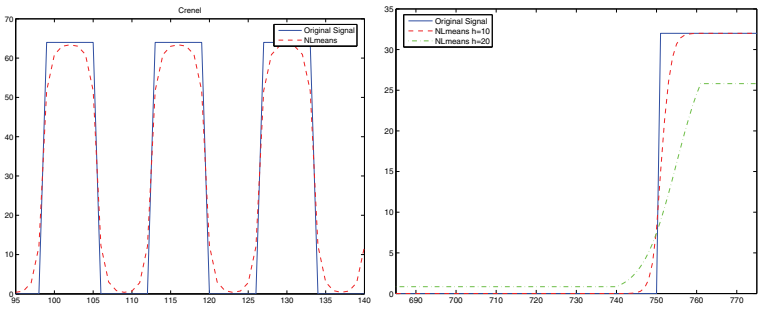


Figure 3. *Left: effect of NLM on a periodic crenel; the result is a piecewise catenary (see (2.2)). The period of the crenel is $T = 14$, its intensity is $\alpha = 64$, the patch size is $s = 21$, $h = 20$. Right: effect of NLM on a step signal. The size of the signal is $N = 1000$, 750 coefficients are zero, the other 250 have intensity $\alpha = 32$. The parameters are $s = 21$, $h = 10$ (dashed line), and $h = 20$ (dash-dot line). The length of the transition area is equal to the patch size. The bias at each pixel becomes more important as h increases and the pixel value becomes rare (see (2.3)).*

the following (see Figure 3):

$$NLu(x) = \frac{\alpha}{(1 - e^{-r\frac{T}{2}})(1 + e^{-r})} \left(1 - e^{-r} - 2e^{-\frac{1}{2}(\frac{T}{2}+1)r} \cosh rx\right), \tag{2.2}$$

where $r = \frac{\alpha^2}{Th^2}$. This shows that even periodic signals may suffer from bias (point 3).

2.3. Step edge. As a last toy example, let us consider two regions (one with θN black pixels, the other with $(1 - \theta)N$ pixels with intensity α, and $\theta \in (0, 1)$) delimited by a step edge. Taking the limit $N \to +\infty$, one may compute the output of the filter (see Figure 3):

$$NLu(x) = \frac{\alpha(1 - \theta)}{(1 - \theta) + \theta e^{(2b(x)-s)r}}, \tag{2.3}$$

where $b(x)$ is the number of black pixels in the patch centered at x (e.g., $b(x) = s$ if the patch of center x lies completely inside the black region). Observe that the gray levels of the two regions are shifted differently, depending on their number of pixels (point 4). Eventually, the transition width between the two regions is proportional to the patch size.

The previous three examples also show the nonlinear behavior of the filter with respect to the contrast (point 6). Two regions with the same geometry but different contrast are handled differently by the filter (notably for the rare patch effect exhibited in section 4.3).

3. Bias-variance trade-off in the patch space. Let us recall that our approach consists of setting the NLM parameters by considering the bias-variance trade-off. Whereas the previous section shed light on the bias of the estimator on toy models, we now focus on the bias-variance trade-off and carry out our study in the patch space.

3.1. Regularity and the patch space. The ease with which one may solve the bias-variance trade-off depends on the regularity of the image, which can be read in the patch space. Several authors [32, 23, 24, 35, 36, 29] interpret the behavior of the algorithm as a

diffusion on a manifold in the patch space (and, indeed, there is strong evidence that the patches of an image lie on a manifold). The patch application[3]

$$U : \begin{array}{ccc} \Omega & \longrightarrow & \mathcal{P} \subset \mathbb{R}^{s^2}, \\ x & \mapsto & (u(y), |y-x|_\infty \leq \frac{s-1}{2}), \end{array} \tag{3.1}$$

which maps every pixel x of the domain to the patch of center x in the patch space $\mathcal{P}$, gives a parametrization of a surface in the patch space (provided u is smooth enough).

However, the geometry of such surfaces is complicated as there are many self-intersections. Moreover, the geometry of the patch manifold does not take into account redundancies in the image. For instance, a closed surface may represent either a single pattern or a periodic one, while this difference is crucial for the NLM. Thus, inasmuch as we are interested in the bias-variance trade-off in the patch space, we focus on the mass distribution of the patch set rather than on its geometry, in a more elementary framework.

A measure in the patch space may be defined by pushing forward the Lebesgue measure of the spatial domain:

$$\forall A \in \mathcal{B}(\mathbb{R}^{s^2}),\ m(A) = \mathcal{L}^2(U^{-1}(A)).$$

If the search window is the whole image, the weights $w(x,y) = \frac{1}{C(x)} e^{-\frac{\|U(x)-U(y)\|^2}{2h^2}}$ (where $C(x)$ is the normalizing factor) depend only on the patch value $U(x)$ and not on the pixel position x itself. Therefore we can denote them by $W(U(x),U(y)) = \frac{1}{C(U(x))} e^{-\frac{\|U(x)-U(y)\|^2}{h^2}}$ and we may write

$$NLu(x) = \int_\Omega u(y) w(x,y) dy = \int_{\mathcal{P}} c(P) W(U(x),P) dm(P), \tag{3.2}$$

where c is the application that maps a patch to the value of its central pixel (i.e., $c(U(x)) = u(x)$).

Because of the normalization of the weights, the measure $W(U(x),P)dm(P)$ has total mass 1. The "bias" of the filter can therefore be expressed as

$$NLu(x) - u(x) = \int_{\mathcal{P}} \left(c(P) - c(U(x))\right) W(U(x),P) dm(P). \tag{3.3}$$

For instance, if an image has a measure m locally widely spread along the central pixel axis, it will be considerably modified by the NLM filter.

The patch regularity assumption may be reformulated by asking that the measure m make the above integral small.

Lipschitz regularity. A natural assumption to control the bias in (3.3) is to assume a Lipschitz regularity of the center: the support of $W(U(x),\cdot)m$ is contained in the set $\{P \in \mathcal{P}, |c(P) - u(x)| \leq k\|\dot{P} - \dot{U}(x)\|\}$, where $\dot{P} \in \mathbb{R}^{s^2-1}$ denotes the patch omitting its central pixel (i.e., $P = (c(P), \dot{P})$) and $k > 0$ is a constant. This provides the upper bound:

$$|NLu(x) - u(x)| \leq \frac{k}{\sqrt{1+k^2}} \int_{\mathcal{P}} \|P - U(x)\| W(U(x),P) dm(P).$$

[3]In this paragraph, we may assume that images are defined on a continuous domain. Otherwise, the Lebesgue measure may be replaced with the counting measure.

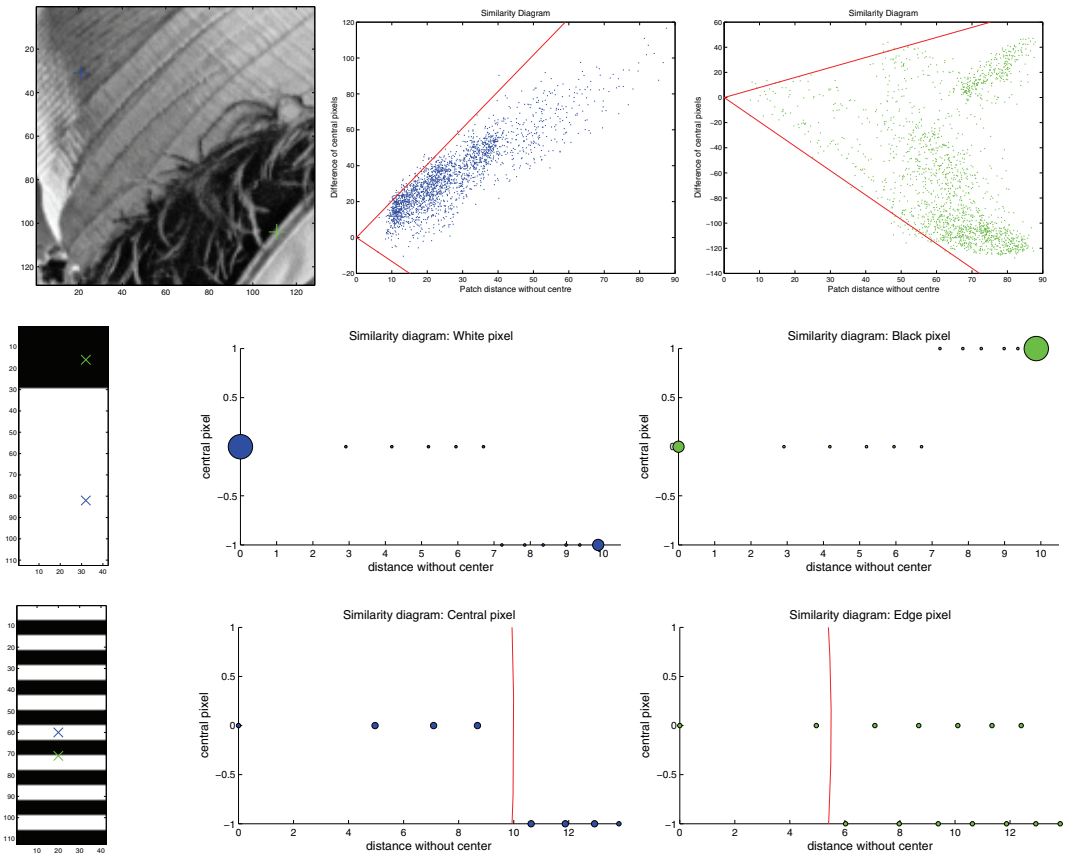


Figure 4. *Similarity diagrams for the pixels indicated by a blue or green cross. Top: a natural image. Middle: two regions. It is more difficult to reduce the variance without introducing bias for the minority pixels than for the majority. Bottom: stripes. Pixels near edges suffer from more bias than pixels at the center of each stripe. The Lipschitz constant k is larger near edges.*

However, this bound may be arbitrarily close to 255 (or any positive constant): consider the mass defined by the crenel signal of section 2.1, or the following one: $m = \frac{1}{N}\delta_{U(x)} + (1 - \frac{1}{N})\delta_{(0,0,\dots,0)}$, where $U(x) = (255, 255, \dots, 255)$. Again, since the exponential weights never cancel, the effect of overwhelming patches is out of control. On the contrary, if one uses kernels φ with compact support as proposed in [17] (e.g., $\varphi(x) = 0$ for $x \geq 1$), the bias is bounded by $\frac{\sqrt{2}k}{\sqrt{1+k^2}}h$.

Similarity diagram. A visual way to examine the regularity at a point is to plot the patch set as the couples $(\|\dot{P} - \dot{U}(x)\|, c(P) - u(x))$. This representation is inspired from the one in [29] where, in dimension 1, patches of size 2 are represented as couples $(u(y), u(y+1))$ for all y. It allows us to see if the estimation of each pixel will be very biased, and it contains the exact information that NLM needs to compute $NLu(x)$. Such *similarity diagrams* are shown in Figure 4 in the case of a natural image and in the case of the examples of sections 2.2 and 2.3. This shows that the regularity constant k varies in the image and is especially high near edges.

3.2. Bias-variance trade-off. The bias-variance considerations of section 1 may be translated to the patch space. Let $R_1 := \mathbb{E}|NL\tilde{u}(x) - NLu(x)|^2$ and $R_2 := \mathbb{E}|NLu(x) - u(x)|^2$. Since

$$R_1 = \mathbb{E}\left(\sum_y n(y)w(x,y)\right)^2 = \sigma^2 \sum_y (w(x,y))^2 = \sigma^2 \int_{\mathcal{P}} (W(U(x),P))^2\, dm(P), \tag{3.4}$$

the risk is given by

$$R_1 + R_2 = \sigma^2 \int_{\mathcal{P}} (W(U(x),P))^2\, dm(P) + \left(\int_{\mathcal{P}} (c(P) - c(U(x)))W(U(x),P) dm(P)\right)^2. \tag{3.5}$$

By the Cauchy–Schwarz inequality, the first term is minimal when the weights are uniform (i.e., $h \to +\infty$). On the contrary, if the image is patch regular, the second term is small for small values of h (since patches with a different central pixel are allowed, but they should be very far in the patch space; see Figure 4). This formula shows that the best choice for h is a trade-off between reducing the variance by taking a large number of pixels in the average, and not averaging pixels that belong to very different patches.

The interpretation of this quantity is even clearer when the kernel is not Gaussian but is given by an indicator function $w(x,y) = \frac{1}{C(x)}\mathbf{1}_{\|U(x)-U(y)\|^2 \le h^2}$, where $C(x)$ is an appropriate normalization factor. In the patch space, the weights can be written as $W(P,Q) = \frac{\mathbf{1}_{B_h(P)}(Q)}{m(B_h(P))}$, where $B_h(P)$ denotes a ball with radius h and center P, and $m(B_h(P))$ denotes its m-measure, i.e., the number of patches within distance h of P. Then, similar computations show that

$$R_1 + R_2 = \frac{\sigma^2}{m(B_{h'}(U(x)))} + \left(\frac{1}{m(B_{h'}(U(x)))}\int_{B_{h'}(U(x))} (c(P) - c(U(x)))dm(P)\right)^2, \tag{3.6}$$

where $h' = \sqrt{h^2 - 2\sigma^2}$ corresponds to a threshold on the noise-free image. The more regular the image is (e.g., the smaller the Lipschitz constant k is), the lower the minimum value of $R_1 + R_2$ is. Figure 4 shows the trade-off for different pixels. In the middle row, a threshold $h' = 7$ allows us to reduce the variance without introducing bias. However, the variance will decrease more for the majority pixels than for the minority ones. In the bottom row of Figure 4, it is immediately necessary to introduce bias near edges in order to reduce the variance, contrary to the center of stripes.

3.3. Bias-variance and the size of the search window. We have seen in sections 2.1 and 3.1 that the bias term depends on the size of the search window. Using a small search window is the common practice and the reason invoked is that, besides the speed-up, the result is visually better. However, on theoretical grounding, restricting the search of patches to a window for computational reasons is very different from doing this because we know that it will produce better results: the second approach contradicts the "nonlocal philosophy" introduced in [7]. If a similar object appears at the other side of the image, there is no reason not to use it to denoise a pattern. On the other hand, we have seen that kernels with compact support allow us to control the bias. The following experiment determines to what extent this is true.

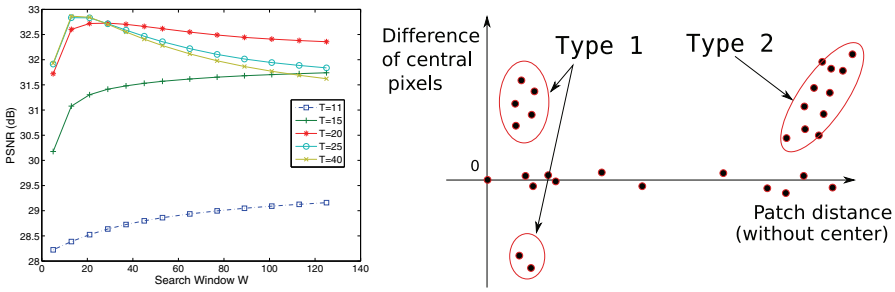


Figure 5. *Left: evolution of the PSNR as the size of the search window increases, for different values of the threshold T, on the boat image. Using a good threshold (e.g., $T = 20$) for the weights makes the algorithm more robust to changes in the search window. The parameters are $h = 10$, $s = 7$, $\sigma = 10$. Right: the different kinds of irrelevant pixels in a similarity diagram.*

To this end, we need to distinguish *two different kinds of pixels that may introduce bias* when denoising a pixel x (see Figure 5). *First*, there are pixels that belong to patches that are similar but have very different central values ($\|\dot{U}(x) - \dot{U}(y)\|$ is small, but $|u(x) - u(y)|$ is large): these patches are the cause of nonpatch regularity. We have seen that such pixels arise, for instance, at edges. *Second*, there are pixels that belong to very different patches (both $\|\dot{U}(x) - \dot{U}(y)\|$ and $|u(x) - u(y)|$ are large). These pixels have a small weight, but, as in section 2.1, it is nonzero. Contrary to pixels of the first type, we can get rid of them by using a truncated kernel.

We run the NLM filter on several images degraded by a noise with $\sigma = 10$ and look at the evolution of the PSNR when the size of the search window increases. However, we truncate the weights in the following way:

$$w(x,y) = \begin{cases} \frac{1}{C(x)} e^{-\frac{\|\tilde{U}(x)-\tilde{U}(y)\|^2}{2h^2}} & \text{if } \|\tilde{U}(x) - \tilde{U}(y)\| \le T, \\ 0 & \text{otherwise,} \end{cases} \tag{3.7}$$

where $C(x)$ is a normalizing factor and the parameter h is set to 10. Since the patch distance is normalized, for a threshold value $T = 255$, the algorithm is equivalent to the usual NLM filter. Typical curves for several values of T are shown in Figure 5. The PSNR first increases with W, since the variance decreases and the bias introduced on small neighborhoods is very low. When W gets large enough, many (but not all) pixels in the search window are not pertinent in denoising x, and the bias increases greatly: the PSNR drops.

The interesting point in this experiment is that when imposing a small threshold value T, the filter is almost insensitive to the increase of the search window. This shows that the pixels of the first kind are not prominent when the search window increases, and therefore the loss of PSNR without thresholding is due to the bias induced by pixels of the second kind.

As a consequence, images are mostly patch regular, and truncated weights make the algorithm more robust to the choice of the search window. In [17], the authors propose using kernels with compact support and show that this allows one to preserve textures better (which can also be understood in light of section 2.1). Let us stress the fact that it also makes the algorithm more robust to the choice of the search window.

Remark 3.1. It is shown in [34] that projecting the patches on their principal components

before computing the distances reduces the decay of the PSNR when W increases. Since the main advantage of this modified ℓ^2 distance is its improved robustness to noise, this suggests that some of the error committed when increasing W is due to the uncertainty on the distances (for large σ). However, the PCA does not solve the problem of nonzero weights (as in section 2.1), and we have performed several experiments highlighting the same bias problem for large W when using PCA.

3.4. Bias-variance and the patch size. In the literature (see [34, 37, 22]), the best results with strong noise are obtained with large patches. As noted in [29], using a large patch allows a more robust discrimination between areas that are not actually similar, which is interesting in the presence of noise. Let us illustrate this in an experiment displayed in Figure 6. We consider an image with mainly two kinds of patches, say two regions (with intensities α and 0), and a noise with standard deviation σ such that the two noisy regions are hard to distinguish. This time we compute the similarity diagram *on the noisy image*: the two regions cannot be discriminated with a 3×3 patch, but with size 15×15, two different clouds clearly appear.

The shapes of the similarity diagrams are explained by the Bienaymé–Chebyshev inequality. Indeed, let P (resp., Q) be a perfect gray (resp., black) patch and $\tilde{U}(x)$ be a noisy patch completely included in the gray region. Then

$$\forall \epsilon > 0,\ \mathbb{P}\left(\left|\|\tilde{U}(x) - P\|^2 - \sigma^2\right| > \epsilon\right) \leq \frac{2\sigma^4}{s^2\epsilon^2},$$

so that for a large patch size s^2, most patches of the gray (resp., black) regions lie near the sphere of radius σ and center P (resp., Q).

Since the two clusters are separated with a large patch size, a good threshold value h allows us to average the pixels of the first cluster only. This does not introduce bias if the first cluster actually corresponds to the pixels of the gray region, i.e., $\|\tilde{U}(x) - P\|^2 \leq \|\tilde{U}(x) - Q\|^2$. But if $n_i \sim \mathcal{N}(0,1)$ i.i.d. for $i \in \{1, \dots, s^2\}$,

$$\begin{aligned}\mathbb{P}\left(\|\tilde{U}(x) - Q\|^2 < \|\tilde{U}(x) - P\|^2\right) &= \mathbb{P}\left(\frac{1}{s^2}\sum_{i=1}^{s^2}(\alpha - \sigma n_i)^2 < \frac{1}{s^2}\sum_{i=1}^{s^2}(\sigma n_i)^2\right) \\ &= \mathbb{P}\left(\frac{\alpha s}{2\sigma} < \frac{1}{s}\sum_{i=1}^{s^2} n_i\right) = \int_{\frac{\alpha s}{2\sigma}}^{+\infty} e^{-t^2/2}\frac{dt}{\sqrt{2\pi}},\end{aligned} \tag{3.8}$$

so that when s is large, most patches are averaged with the correct cluster.

The same holds for natural images: large patches are needed for their robustness to noise. An example is shown in Figure 7, where a small patch size induces a mottling effect. However, large patches make the algorithm more exposed to both bias and variance. First, as in section 2.1, using a large patch size reduces the importance of little contrasted small details, so that they are more blurred. Second, if the image has textures with highly contrasted transitions or curved and contrasted edges, using a too large patch will prevent the algorithm from finding redundancies and the variance will stay large (see Figure 8). A way to overcome both issues is to select the smoothing parameter h accordingly: when choosing the useful

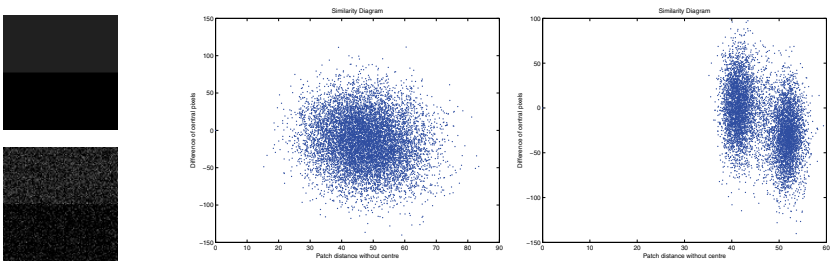


Figure 6. *Left: original (intensity* 0 *and* $\alpha = 32$*) and noisy image* ($\sigma = 30$*). Middle and right: similarity diagram for a pixel at the center of the gray region (the patch sizes are* 3×3 *and* 15×15*, respectively). With a large patch size one sees two clusters. Question: Does the closest cluster correspond to patches which were originally gray as well? When s is large, the answer is yes with high probability: in strong noise, a large patch size discriminates the two regions better than a small one.*

Figure 7. *Left and right: extract of the result of the NLM filter with* $s^2 = 3 \times 3$ *and* $s^2 = 5 \times 5$ *on a noisy image* ($\sigma = 10$*, h optimized for PSNR). Using a too small patch size makes the algorithm less robust to noise, and Lena's skin looks mottled. It looks smoother with patch size* 5×5*, but visual artifacts appear in the eye.*

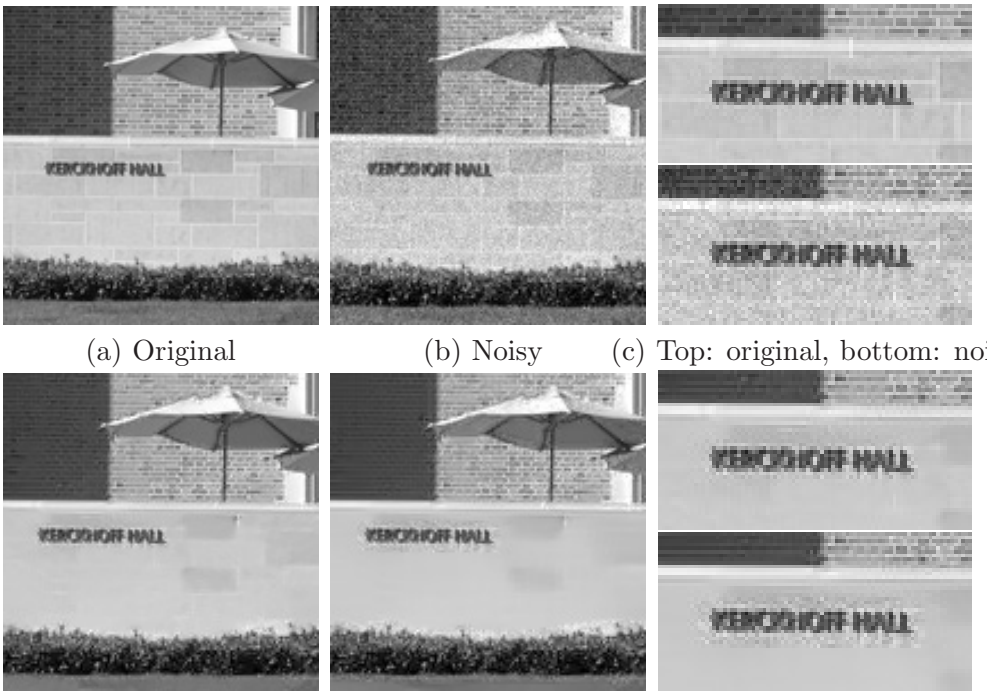
patches, be more selective in the first case and more tolerant in the second one. These two contradictory behaviors impose a local choice of h.

The patch size ideally should also be chosen depending on the local scale of the signal/image, but a locally defined h should make this choice less critical. In the next section we propose setting both the smoothing parameter and the patch size automatically.

4. Making the nonlocal means spatially adaptive. In this section, we propose locally selecting the parameters by solving the bias-variance trade-off. As the above discussion advocates for kernels with compact support (see also [17]), we consider general filters of the form

$$NLu(x) = \frac{\sum_{y\in\Omega} \varphi\left(\frac{\|U(x)-U(y)\|^2}{2h^2}\right) u(y)}{\sum_{y'\in\Omega} \varphi\left(\frac{\|U(x)-U(y')\|^2}{2h^2}\right)}, \tag{4.1}$$

with φ nonnegative and nondecreasing. With a kernel with compact support, the filter is more robust to the choice of W than it is with a global kernel, so that in the following, we fix W in advance and then we choose h locally. This approach is dual to that proposed by Kervrann and Boulanger [18], who fix h and then control the bias and variance of the filter by choosing the size of the search window.



(a) Original (b) Noisy (c) Top: original, bottom: noisy

(d) Patch size 5×5 (e) Patch size 9×9 (f) Top: 5×5, bottom: 9×9

Figure 8. *Choice of the patch size: original image* (a)*, noisy image* (b) *($\sigma = 10$), NLM filter with patch size $s^2 = 5 \times 5$* (d)*, and $s^2 = 9 \times 9$* (e)*. Zoomed versions are shown in* (c) *and* (f)*. Around the letters it is very difficult to find similar patches, and a noisy halo appears. A smaller patch size reduces the spread of the halo since it allows us to find similar patches for the furthest pixels. Another solution would have been to force a high smoothing parameter h (those used here were chosen to maximize the PSNR).*

4.1. Oracle estimation. To show that the behavior of the SURE-based method of section 4.2 is an approximation of the optimal behavior, we first build an oracle which has access to the *real* local squared error. Although this algorithm is not usable in practice, it gives an idea of which parameters should be used in each region and what can be expected in terms of visual quality.

For the oracle estimation only, we have used the indicator kernel $\varphi(x) = \mathbf{1}_{[0,1/2]}(x)$. This kernel has compact support, and it does not suffer from the overestimation of the *self-weights* $w(x,x)$ pointed out in [26]. Let us recall that for $y \neq x$, the distance between noisy patches $\|\tilde{U}(x) - \tilde{U}(y)\|^2 \approx \|U(x) - U(y)\|^2 + 2\sigma^2$ is increased by the noise level, whereas the *self-distance* $\|\tilde{U}(x) - \tilde{U}(x)\|^2$ is always zero. Therefore, with Gaussian weights, the weight $w(x,x) = \frac{1}{C(x)}$ is proportionally $e^{\frac{2\sigma^2}{2h^2}}$ times larger in the presence of noise than it would be without noise. As a consequence, some authors set $w(x,x)$ to $\frac{1}{C(x)} e^{-\frac{\delta^2}{2h^2}}$, where $\delta^2 = \min_{y \neq x} \|\tilde{U}(x) - \tilde{U}(y)\|^2$, or replace it with $\frac{1}{C(x)} e^{-\frac{\sigma^2}{h^2}}$ (see [26]). On the contrary, the indicator weights do not behave this way, and we need not give a special value to $w(x,x)$.

With this kernel, minimizing the bound (3.6) over the radius h' amounts to finding the

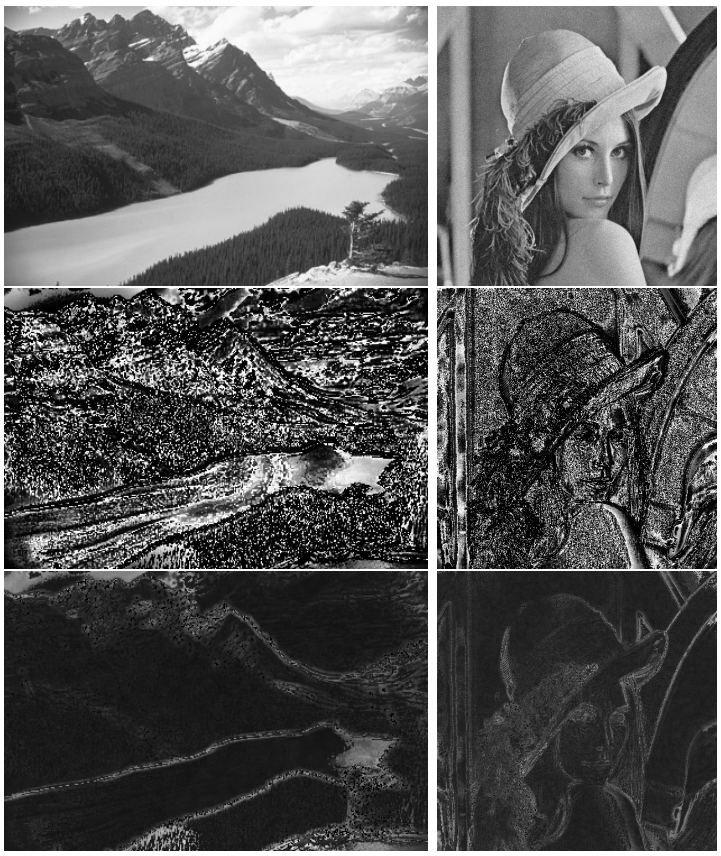
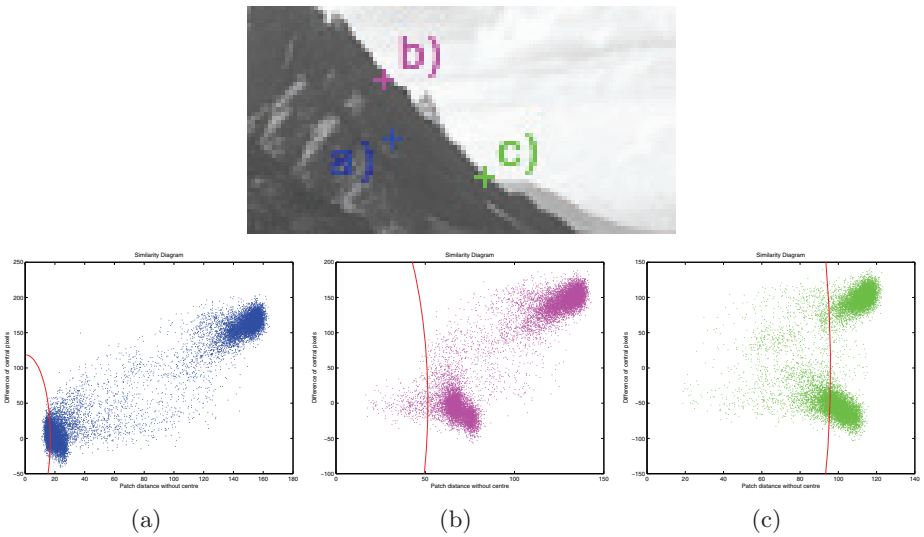


Figure 9. *Map of parameters prescribed by the oracle. The original images are degraded with $\sigma = 10$; the patch size is $s = 7$. From top to bottom: original images (without noise), map of the number of pixels in the mean, map of the corresponding h parameter. In the second row, the white regions represent a number of pixels $n_x \approx 3000$. In the third row, first image, the parameter h is approximately equal to 14 on the lake, while it ranges from 15–20 in the forest, and from 40–75 along the edges of the mountains and the lake. On the rocks it is around 30. Although the denoising of edges should be performed with very few pixels, the corresponding parameter h should be very large.*

optimal number of pixels $n_x := m(B_{h'}(U(x)))$ when denoising x. To build our oracle estimate, we compute the risk on the *noise-free image* u for each integer, and we keep the minimizing value n_x. Then we can estimate $u(x)$ from $\tilde{u}$ by averaging the centers of the n_x patches $\tilde{U}(y)$ that are nearest to $\tilde{U}(x)$ in Euclidean distance. In a nutshell, we use the oracle to define a map n_x and then compute the nonlocal filter on the noisy image $\tilde{u}$, keeping only the best n_x patches. Notice that this procedure is roughly equivalent to one iteration of [5] with the difference that the number of pixels is chosen locally, by an oracle.

Figure 9 shows the number of pixels n_x recommended by the oracle and the associated smoothing parameter h_x (that is, we display the norm $\|U(x) - U(y)\|$, where y is the last pixel taken into account). As expected, in very smooth regions the oracle selects as many pixels as possible, whereas in regions where the image is not patch regular (i.e., near edges) the oracle recommends using very few pixels. The case of textures falls in between. More surprising is the map of the corresponding h_x: the values prescribed near edges are much higher than in



(a) (b) (c)

Figure 10. *Similarity diagram for three different pixels. The red line indicates the ball of radius h_x prescribed by the oracle (the pixels at the right of this line are not taken into account in the mean). In the interior of homogeneous regions* (a)*, a small radius ($h_x \approx 17$) is sufficient to reduce the variance of the noise. Near edges* (b) *and* (c)*, one has to look very far in the patch space to find enough pixels to reduce the variance ($h_x \approx 50$ and 95, respectively). The threshold is especially large in the third case since the compensation of the darker and brighter pixels yields a very small bias.*

smooth regions or textured regions. In fact, even though the oracle prescribes very few pixels to reduce the variance term, one has to go very far in the patch space to gather enough pixels. This is illustrated in Figure 10 (notice that here these similarity diagrams are computed on the noisy image). Therefore, *one should use much higher values of h near edges.* Let us stress that this problem is *not* related to the overestimation of the self-weight mentioned above, since the indicator kernel does not suffer from this drawback.

Figure 11 shows the PSNR as a function of the size of the search window (cf. Figure 5). This time, the global trend of the PSNR is nondecreasing with the size of the search window. The conclusion is that there is no harm in computing *fully nonlocal* means, provided that we carefully choose the pixels in the means,[4] but there is also little gain in doing so unless the image has large smooth areas (like the *mountain* image). In general, the PSNR tends to stabilize for a side-length W greater than 25. Slights oscillations of the PSNR are due to the following balance: when increasing the search window, one adds to the mean several relevant pixels that help reduce the variance, but also a few pixels of the first kind (see section 3.3) which perturb the estimation of $u(x)$. However, they are very few, so that overall the PSNR is stabilized.

4.2. SURE. In this section, we show that one can approximate the behavior of the oracle by locally minimizing *Stein's unbiased risk estimate.* This is a way to handle the bias-variance

[4]Phrased this way, this statement appears to be a tautology, but one should remember that the expression (4.1) imposes a structure on the choice of the pixels, and it would not be true if pixels of the first kind (see section 3.3) were overwhelming in natural images.

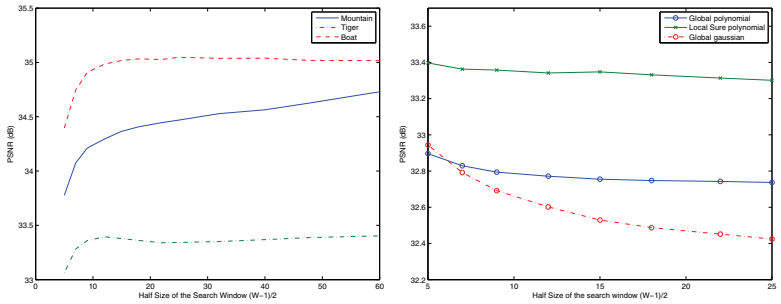


Figure 11. *Evolution of the PSNR as the size of the search window W increases. Left: for the oracle with indicator weights on different images. The patch size is $s = 7$; the noise level is $\sigma = 10$. With a locally defined smoothing parameter h, there is no or very little loss when using a large window, so that the choice of W is not a real issue, contrary to Figure 5. Right: for the local SURE filter on the image "mountain." The local SURE filter, the NLM with polynomial weight, and the classical NLM are displayed (for the last two filters, the parameter h was optimized for PSNR for each size of the search window). The local SURE filter is robust to the size of the search window, mainly because of the compact support of the weights.*

dilemma of section 3 without making the assumption that the weights are deterministic. The use of SURE to select the parameters is well known in the wavelet community [12, 41, 8, 4], but on the whole it is less widespread in the image processing community [30]: it is usually difficult to compute the necessary analytic expression of SURE. Recently, however, it was shown in [25] that one could replace this computation with a stochastic estimation. In the recent work [37] (which we discovered after deriving the results of section 4.2.1), Van De Ville and Kocher have shown that these computations are tractable in the case of the classical NLM. In this paper we extend this computation to compactly supported weights, and we use it to perform a local choice of the parameter h. Algorithmic issues are also discussed.

4.2.1. Estimation of the risk. Let us first recall the result by Stein (see [31]).

Proposition 4.1 (Stein). *Let $x \in \mathbb{R}$, $Z \sim \mathcal{N}(0, \sigma^2)$, and $Y = x + Z$. If $\gamma : \mathbb{R} \to \mathbb{R}$ is absolutely continuous, and*

(i) $\lim_{|z|\to\infty} \gamma(x+z)e^{-\frac{z^2}{2\sigma^2}} = 0$,
(ii) $\mathbb{E}(\gamma(x+Z))^2 < +\infty$ *and* $\mathbb{E}|\gamma'(x+Z)| < +\infty$,

then the risk of the estimator $\gamma(Y)$ of x is given by

$$\mathbb{E}|\gamma(Y) - x|^2 = \mathbb{E}\left[-\sigma^2 + 2\sigma^2\gamma'(Y) + (\gamma(Y) - Y)^2\right]. \tag{4.2}$$

The proof relies on an integration by parts. Let $\tilde{u}$ be the noisy image, $NL\tilde{u}$ the result of NLM applied to the noisy image using the noisy weights, and z the noise at pixel x (i.e., $\tilde{u}(x) = u(x) + z$). Then

$$J(x) := -\sigma^2 + 2\sigma^2\left(\frac{d}{dz}NL\tilde{u}(x)\right) + (NL\tilde{u}(x) - \tilde{u}(x))^2 \tag{4.3}$$

is an unbiased estimator of the risk at pixel x, and in [37], an analytic expression of J is given in the case of the Gaussian weights. The authors show that this estimator yields a very robust estimation of the global mean square error.

In the general case of a kernel φ with compact support, the middle term is rewritten as

$$\frac{d}{dz}\left(\sum_y \tilde{u}(y)\frac{\varphi_y}{C}\right) = \frac{\varphi(0)}{C} + \frac{1}{C}\sum_y \tilde{u}(y)\frac{\partial\varphi_y}{\partial z} - \frac{1}{C^2}\left(\sum_y \tilde{u}(y)\varphi_y\right)\left(\sum_w \frac{\partial\varphi_w}{\partial z}\right), \tag{4.4}$$

where

$$\varphi_y := \varphi\left(\frac{\|\tilde{U}(x)-\tilde{U}(y)\|^2}{2h^2}\right), \qquad C := C(x) = \sum_y \varphi_y,$$

and the derivative of φ_y is given by

$$\frac{\partial\varphi_y}{\partial z} = \frac{1}{h^2 s^d}(\tilde{u}(x)-\tilde{u}(y)-\underbrace{(\tilde{u}(x+i_0)-\tilde{u}(x))}_{\text{if } \exists i_0\in P,\ x=y+i_0})\varphi'_y.$$

The last two terms appear only when y belongs to the patch centered at x (i.e., $|x-y|_\infty \leq \frac{s-1}{2}$).

As with the Gaussian weights, this procedure yields a very reliable estimation of the (global) mean square error when it is summed over all pixels x in the image. Note that it is necessary to compute the NLM for each parameter to estimate the corresponding risk, and that (4.4) adapts straightforwardly to the trick of replacing $\varphi(0)$ with $\exp(-\frac{\sigma^2}{h^2})$ in the self-weight.

4.2.2. Local approach. In order to select local parameters, we use the estimation (4.3) to minimize the risk depending on the local content of the image (textured areas, smooth regions, etc.). Since the pointwise estimation of the risk is not robust, we need to locally average the estimations. The underlying assumption is that the risk is roughly homogeneous within each region (smooth/textured). One should find the right balance between having enough samples to estimate the risk and keeping a local estimation.

Considering a set of values $\{h_1, h_2, \dots h_n\}$ for the smoothing parameter, we compute for each value the output of the filter $(NL_{h_i}\tilde{u})_{i=1,\dots,n}$ and the associated risk map $(J_{h_i})_{i=1,\dots,n}$. We convolve each risk map J_{h_i} with a disk indicator or a Gaussian with small radius to have a more robust estimation of the local risk. Then we choose for each pixel x the value $h_i(x)$ that minimizes the convolved risk at pixel x, $J_{h_i}(x)$, and we retain the corresponding estimation $NL_{h_i(x)}\tilde{u}(x)$.

Implementation. The procedure we propose, called LBMRE (local bandwidth minimizer of the risk estimate), is described in Figure 12. It is necessary to compute many NLM filters, but this procedure is simpler than several methods proposed in the literature inasmuch as it is *not* iterative. The expensive computations of the patch distances need be performed only once (since all the filters work with the same input image), and as the other computations are independent, they can be parallelized. As an indication, our code takes 26 s to execute lines 1 to 15 (the rest is negligible) on a 256×384 image using a search window W of 23×23 and 64 values of h, on an Intel Core2 Duo 2.5GHz and 4Gb RAM. In addition, the speed can still be improved, since our C code (which uses SSE instructions to vectorize the computations) uses only one of the two cores. Additional tricks could be added, such as taking advantage of the fact that, for each pixel, if a weight is zero for some value h_1, it is necessarily zero for all $h_2 \leq h_1$. Moreover, several approaches proposed in the literature (such as the use of integral

Nonlocal means with local h (LBMRE)

for all pixel x **do**
  for all translation $k \in \mathbb{Z}^d$, $|k|_\infty \leq \frac{W-1}{2}$ **do**
    dist $\leftarrow \frac{\|U(x)-U(x+k)\|^2}{2}$
    for i=1 to n **do**
      $(\sum u\varphi)_i \leftarrow (\sum u\varphi)_i + u(x+k)\varphi(\frac{\text{dist}}{h_i^2})$,
      $(\sum \varphi)_i \leftarrow (\sum \varphi)_i + \varphi(\frac{\text{dist}}{h_i^2})$,
      $(\sum u\varphi')_i \leftarrow (\sum u\varphi')_i + u(x+k)\varphi'(\frac{\text{dist}}{h_i^2})$,
      $(\sum \varphi')_i \leftarrow (\sum \varphi')_i + \varphi'(\frac{\text{dist}}{h_i^2})$
    end for
  end for
  for i = 1 to n **do**
    $NL_i u(x) \leftarrow (\sum u\varphi)_i \,/\, (\sum \varphi)_i$,
    $J_i(x) \leftarrow \cdots$ (4.3)
  end for
end for
for i = 1 to n **do**
  $J_i \leftarrow J_i * G_r$
end for
for all pixel x **do**
  LBMRE$(x) \leftarrow NL_{i(x)}u(x)$, where $i(x) = \arg\min_i J_i(x)$
end for

Figure 12. *LBMRE algorithm.*

images to compute the patch distances in [10], or the cluster tree in [6] to accelerate the NLM) could be adapted.

4.3. Experimental results. In this section, we illustrate the differences between the NLM filter with optimal global parameter (i.e., using the value of h that minimizes the true MSE) and the NLM with local parameter estimated using SURE (LBMRE). The indicator oracle (section 4.1) is also shown. The NLM filters[5] used with SURE are given by the polynomial kernel: $\varphi(x) = \mathbf{1}_{[0,1]}(x)(1-(10x^6-24x^5+15x^4))$. Observe that this kernel is smooth enough to apply Proposition 4.1. In the following, the local and global versions share the same values for the parameters that are not locally selected and, unless otherwise stated, the size of the search window is set to 29×29 and the patch size is 7×7. Notice that the use of compactly supported weights already gives a better result than the original Gaussian ones, as explained in [17]. When the noise level is $\sigma = 10$, we take $\{h_1, h_2, \ldots, h_n\} = \{3, 3.5, \ldots, 34.5\}$ (see the previous paragraph) and scale these values proportionally when σ varies.

[5]We do not replace the self-weight $w(x,x)$ by the maximum of the weights $w(x,y)$. This would slightly reduce the rare patch effect described here, but it would not solve it because this effect is due to the configuration of the patch cloud (see Figure 10) and even the indicator weights suffer from it. This trick would also favor the loss of details. Moreover, its relevance is questioned in [26].

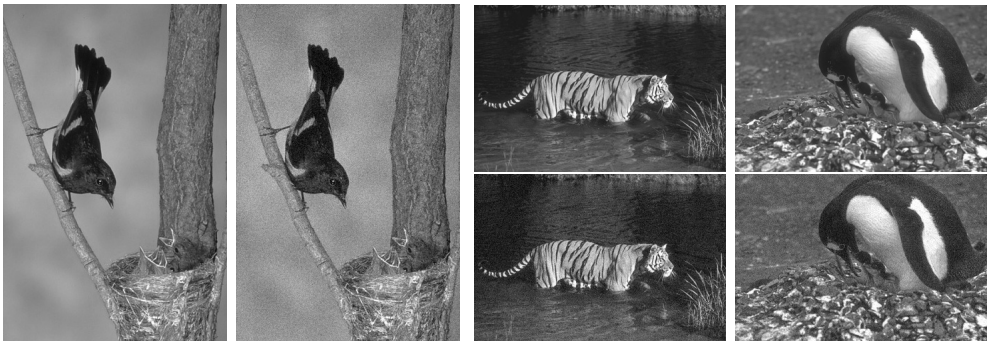


Figure 13. *Original and noisy images ($\sigma = 10$).*

Figure 14. *Bird image. From left to right: NLM with global smoothing parameter h optimized for PSNR, NLM with local h (LBMRE), zoom of the NLM with global h, local h, and their respective method noise $(\tilde{u} - NL\tilde{u})$. Along contrasted edges, the global NLM leaves a noisy halo. This "rare patch effect" becomes stronger as the edge is winding. The local choice of the smoothing parameter corrects this shortcoming.*

Original and noisy images are displayed in Figure 13. In Figure 14, it is shown that the local selection of h enables one to get rid of the *rare patch* effect, responsible for noisy halos around edges. The advantage of using the local approach is further illustrated in Figure 15, where one observes in particular that the macrotexture made of the tiger stripes is better preserved using a local selection of h. Both the PSNR and the structural similarity (SSIM) are improved in these experiments, as illustrated in Table 1 for $\sigma = 10$ and Table 2 for $\sigma = 20$. In these tables the proposed method is denoted as LBMRE with polynomial weights. The case of Gaussian weights is also shown, and the local bandwidth with exponentially weighted aggregation (LBEWA) is an alternative method presented in the conclusion.

Next, we display in Figure 16 the maps of prescribed values for h, first using the (ideal) oracle, then using the approach described in section 4.2.2. Of course, the values prescribed by the oracle are spatially more accurate than those using LBMRE. The corresponding denoising result is also better. It is therefore tempting to try to improve the estimation of h. We performed various attempts in this direction, for example, using some nonlocal regularization of the risk map, relying on weights computed on the noisy image. This did not improve the PSNR but, instead, yielded some visible artifacts, and we did not further pursue this direction. Observe also that some abnormally high h values are present on the map obtained from local

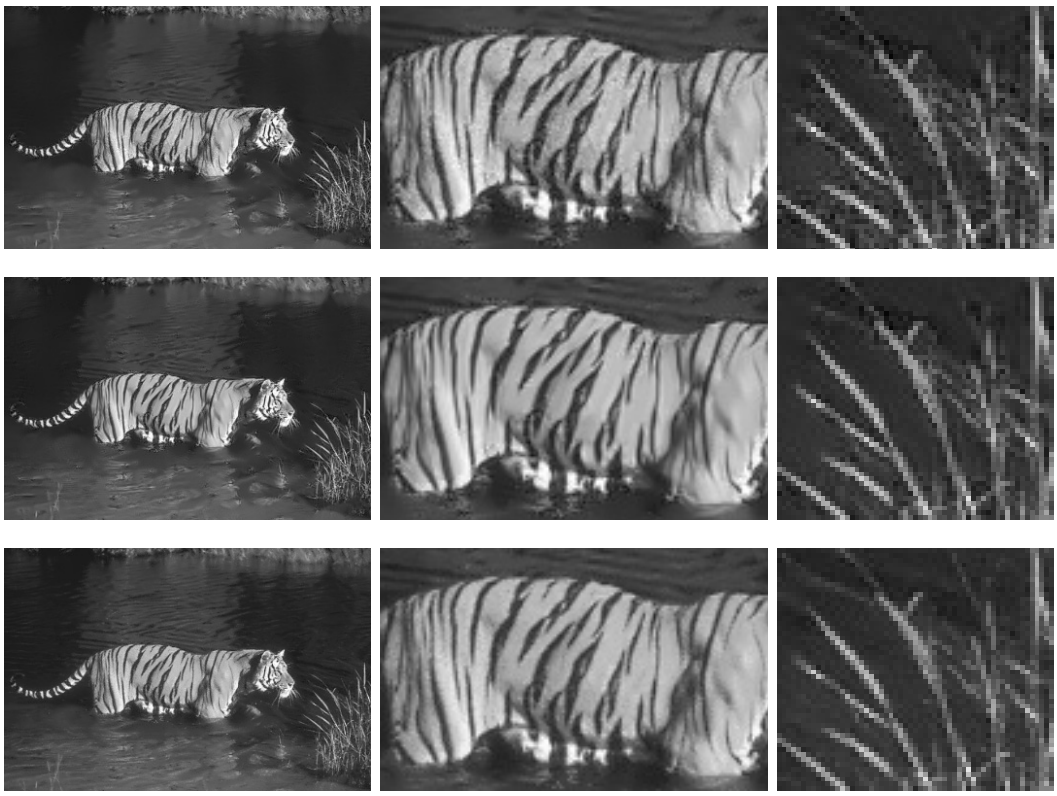


Figure 15. *Tiger image. From top to bottom: NLM with global parameter h optimized for PSNR, NLM with local h (LBMRE), oracle.*

SURE. They may be explained by very flat risk curves in these areas, as illustrated in the same figure, and therefore do not significantly impair the denoising process.

In the experiments of Figures 17 and 18, it is shown that the optimal local value of h near an edge strongly depends on its contrast. As a consequence there is no way to globally set h to efficiently denoise edges.

The experiment of Figure 19 shows that the LBMRE approach permits the reduction of noise halos and artifacts that appear when the patch size s is increased. Figure 20 displays an example where the local approach yields a better preservation of fine details. Finally, it is illustrated in Figures 21 and 22 that this ability to increase the value of s enables one to avoid the *mottling effect* (local intensity fluctuation due to a nonrobustness of small patch sizes) without washing out textures.

Eventually, we can also locally adapt the patch size using the local SURE. Figures 21 and 22 recap the trade-off one has to face when choosing the global patch size: a too small patch size yields a mottling effect, but it allows one to preserve details, whereas a large patch size produces smoother images except in regions where it brings the rare patch effect. Choosing h locally already makes this choice easier by allowing one to preserve details with large patch sizes. Therefore, the gain in letting s vary locally is visually small, although in our experiments it always provides a slightly higher PSNR.

Table 1
Comparison of the PSNR (dB)/SSIM [39] for the aggregation ($\sigma = 10$).

	NLM Gaussian weights		NLM polynomial weights			BM3D	NLM ind. w.
	Global h	Local h (LBMRE)	Global h	Local h (LBMRE)	Local h (LBEWA)	[9]	Local h Oracle
barbara	33.08/0.961	33.71/0.968	33.28/0.962	33.84/0.970	33.51/0.966	34.94/0.977	35.21/0.978
boat	31.84/0.938	32.59/0.952	32.13/0.943	32.76/0.955	32.55/0.949	33.86/0.966	33.95/0.963
bird	32.60/0.916	33.11/0.923	32.60/0.917	32.92/0.920	33.07/0.922	33.83/0.931	33.68/0.923
bridge	29.50/0.879	29.98/0.893	29.42/0.879	29.73/0.890	29.86/0.896	30.79/0.911	30.68/0.916
cameraman	32.35/0.910	33.26/0.922	32.52/0.912	33.07/0.921	33.14/0.920	34.08/0.932	34.19/0.925
country house	30.81/0.823	31.38/0.848	30.78/0.822	31.21/0.841	31.36/0.848	32.02/0.851	32.35/0.880
couple	31.92/0.931	32.58/0.951	32.23/0.941	32.81/0.955	32.47/0.946	34.00/0.967	33.99/0.965
fingerprint	30.34/0.979	30.76/0.987	30.37/0.983	30.69/0.987	30.69/0.989	32.48/0.991	31.94/0.989
flinstones	31.27/0.971	31.93/0.978	31.14/0.971	31.50/0.977	32.04/0.979	32.48/0.980	33.71/0.984
hill	30.55/0.854	31.08/0.868	30.53/0.847	30.90/0.863	31.01/0.870	31.81/0.883	32.13/0.902
house	34.55/0.879	35.20/0.896	34.88/0.887	35.46/0.900	35.10/0.891	36.59/0.918	36.52/0.915
lena	34.00/0.947	34.56/0.958	34.26/0.952	34.82/0.960	34.35/0.953	35.86/0.969	35.77/0.967
man	32.08/0.936	32.74/0.949	32.32/0.941	32.87/0.952	32.59/0.946	33.94/0.963	33.99/0.963
mandrill	30.28/0.931	31.02/0.949	30.51/0.930	31.33/0.951	30.98/0.950	33.18/0.966	32.18/0.963
peppers	32.89/0.903	33.54/0.912	33.17/0.906	33.67/0.914	33.41/0.908	34.72/0.927	35.18/0.936
tiger	31.59/0.846	32.28/0.855	31.72/0.846	32.31/0.858	32.17/0.850	33.42/0.887	33.41/0.886
ucla	30.43/0.928	31.03/0.940	30.48/0.932	30.78/0.937	30.93/0.939	31.63/0.948	31.68/0.952

Table 2
Comparison of the PSNR (dB)/SSIM [39] for the aggregation ($\sigma = 20$).

	NLM Gaussian weights		NLM polynomial weights			BM3D	NLM ind. w.
	Global h	Local h (LBMRE)	Global h	Local h (LBMRE)	Local h (LBEWA)	[9]	Local h Oracle
barbara	29.44/0.917	29.81/0.929	30.11/0.929	30.52/0.938	29.12/0.912	31.67/0.952	31.20/0.946
bird	28.79/0.813	29.28/0.836	28.91/0.826	29.36/0.837	28.95/0.813	30.18/0.859	29.85/0.825
boat	28.55/0.860	29.07/0.887	29.03/0.871	29.59/0.897	28.70/0.865	30.80/0.926	30.21/0.915
bridge	25.56/0.729	25.99/0.750	25.58/0.714	25.92/0.741	25.85/0.750	26.83/0.782	26.97/0.809
cameraman	28.73/0.830	29.47/0.852	28.96/0.841	29.57/0.854	29.22/0.833	30.52/0.878	30.28/0.844
couple	28.32/0.867	28.68/0.884	28.85/0.884	29.29/0.897	28.26/0.864	30.73/0.929	29.85/0.915
country house	27.71/0.703	28.06/0.724	27.77/0.702	28.12/0.724	27.93/0.716	28.88/0.743	28.88/0.772
fingerprint	26.59/0.938	26.86/0.953	27.02/0.951	27.21/0.957	26.52/0.948	28.86/0.972	28.38/0.969
flinstones	27.59/0.941	28.47/0.958	28.17/0.952	28.87/0.961	28.31/0.954	29.60/0.966	30.06/0.967
hill	27.13/0.724	27.48/0.739	27.35/0.721	27.70/0.743	27.16/0.725	28.54/0.778	28.53/0.800
house	31.60/0.841	31.87/0.844	32.32/0.851	32.56/0.851	31.30/0.826	33.77/0.869	32.81/0.838
lena	30.93/0.907	31.22/0.916	31.44/0.912	31.70/0.922	30.62/0.902	32.94/0.940	31.89/0.928
man	28.79/0.862	29.18/0.881	29.13/0.867	29.52/0.888	28.68/0.858	30.56/0.917	30.16/0.913
mandrill	26.59/0.840	26.90/0.863	26.91/0.836	27.36/0.872	26.67/0.847	29.11/0.912	28.18/0.906
peppers	29.38/0.845	29.83/0.851	29.83/0.854	30.26/0.858	29.33/0.830	31.38/0.885	31.04/0.867
tiger	28.29/0.741	28.82/0.758	28.46/0.742	29.02/0.764	28.63/0.746	30.01/0.806	29.58/0.776
ucla	26.81/0.824	27.28/0.855	26.73/0.829	27.19/0.855	27.08/0.844	27.94/0.875	28.12/0.892

4.4. Conclusion. We have proposed a bias-variance study of the NLM which shows the interest of using local parameters. Our LBMRE method automatically chooses the smoothing parameter h and the patch size s^2. Choosing the patch size s^2 locally provides hardly any visual improvement over a global (well-chosen) patch size: we suggest selecting a global patch size large enough to avoid the mottling effect (5×5 or 7×7) and a local parameter h.

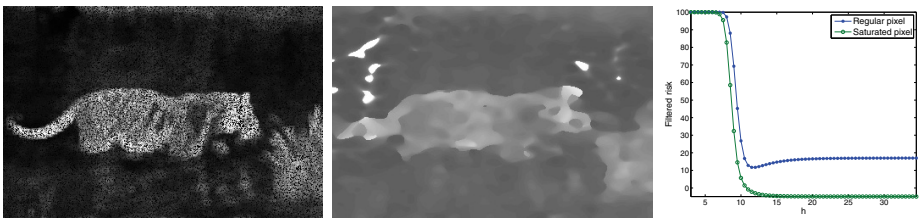


Figure 16. *Map of the prescribed value of h using the oracle (left) and the filtered SURE (middle). The middle map is rough, but it shows the same general behavior as the left one. In some areas, the chosen h is very high since the filtered SURE is flat when h varies (right). It has no visual consequence.*

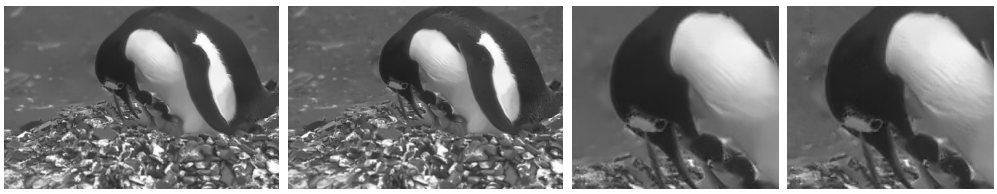


Figure 17. *Penguin image. From left to right: NLM with global parameter optimized for PSNR, NLM with local h (LBMRE), zoom of the same results.*

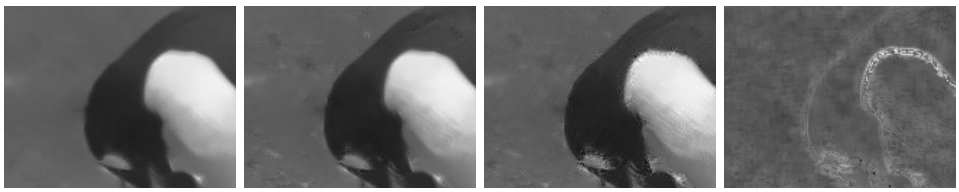


Figure 18. *Experiment with $\sigma = 20$. Left: NLM filter ($h = 30$ optimized for the PSNR). The global optimal parameter is too high for the least contrasted edges, so that, as in (2.3), they are blurred. Middle left: NLM with local h (LBMRE). Along the least contrasted edge, the chosen value of h is about 24. If we set the global parameter to 24 (middle right), these edges become sharp, but the more contrasted edges become noisy. Right: map of h prescribed by the indicator oracle. The more contrasted the edge, the higher h should be.*

The main improvement brought by the locality is to remove the "rare patch effect." This is definitely a visual improvement, but since the concerned regions, along edges or highly contrasted textures, are not prominent in images, the gain on the global PSNR/SSIM is moderate. A second, less striking, improvement is the better preservation of the contrast of details and textures. However, a limitation of the method is that the decision of the local SURE might yield small visual artifacts in some regions.

As a future work, we plan to investigate the benefit of combining the estimators associated with different parameters instead of keeping the minimizer of the estimated risk as we do in this paper. In the context of NLM, recent works have shown that considering a linear combination of estimators instead of the best one often leads to better results (see in particular [28, 38]). Figures 23 and 24 show a preliminary result using an exponentially weighted aggregation (see [20] or, in the context of NLM, [27]) of the NLM for different values of h. Instead of considering

(a) Original (b) Global h, $s^2 = 5 \times 5$ (c) Global h, $s^2 = 7 \times 7$

(d) Noisy $\sigma = 10$ (e) Local h, $s^2 = 5 \times 5$ (f) Local h, $s^2 = 7 \times 7$

Figure 19. *Couple image (only an extract is shown). Top: extract of the original image* (a)*, NLM with global parameter h: the patch size is $s^2 = 5 \times 5$ in* (b) *(PSNR* 32.46 *dB) and $s^2 = 7 \times 7$ in* (c) *(*32.14 *dB). In both cases, h was chosen to maximize the PSNR. Bottom: noisy image* (d) *($\sigma = 10$), NLM with local h (LBMRE): the patch size is $s^2 = 5 \times 5$ in* (e) *(*32.77 *dB) and $s^2 = 7 \times 7$ in* (f) *(*32.70 *dB). Notice how the face, the tie, and the shoulder are smoother with the local h (the reader should zoom in on this picture) with both patch sizes. Yet the contrast of the wall is not lost.*

the minimizer, we take

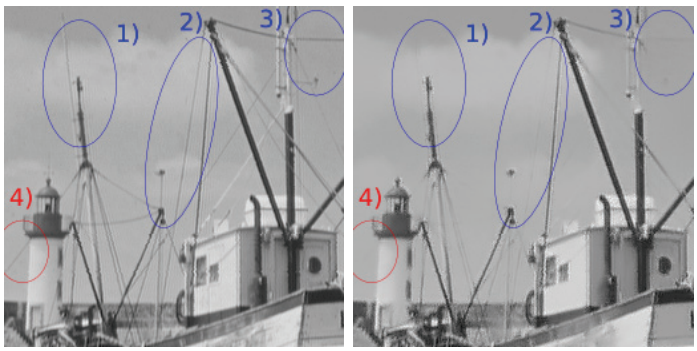
$$\mathrm{LBEWA}(x) = \frac{\sum_{i=1}^{n} \exp\left(-J_i(x)/T\right) NL_i\tilde{u}(x)}{\sum_{i=1}^{n} \exp\left(-J_i(x)/T\right)} \tag{4.5}$$

with $T = 0.5\sigma^2$. As before, everything is performed locally. This procedure leads to slightly noisier images than the method described in this paper, but the preservation of textures is much better. We find that the visual result is drastically improved. Yet, it is less stable in terms of PSNR: the performance is sometimes better than the local h described in this paper, or sometimes below the best global h (see LBEWA in Tables 1 and 2).

Appendix: Assumptions on the toy signals of section 2. For the isolated crenel and the periodic crenel, we assume that

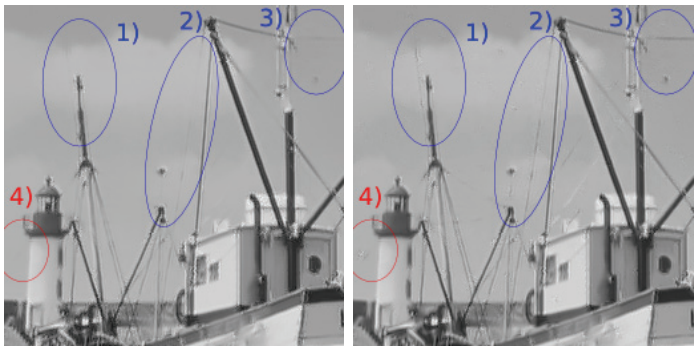
- the crenel size is $T/2 = p$, where p is an odd number;
- the size of the search window is infinite, or it is a multiple of T.

We assume moreover that the patch size is $s = T/2$ in the case of the isolated crenel, and $s = \left(k + \frac{1}{2}\right) T$, for $k \in \mathbb{N}$, in the case of the periodic crenel (i.e., the period of the signal is small compared to the patch size). In both cases, the computation of the distances leads to a triangle signal, so that the denoised signal is the quotient of geometric sums.



(a) Original image (without noise).

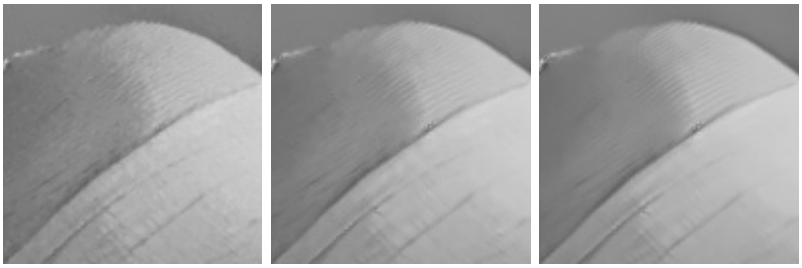
(b) Gaussian weights, global h.



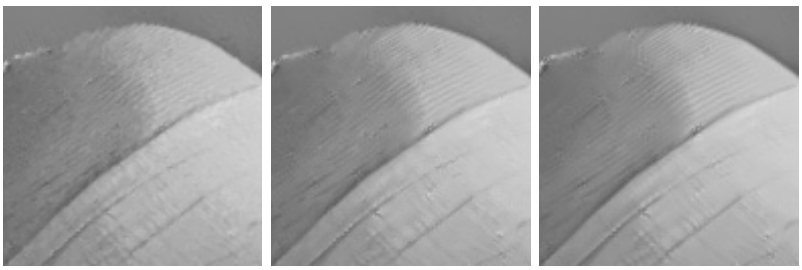
(c) Polynomial weights, global h.

(d) Polynomial weights, local h.

Figure 20. *Comparison of the NLM on a noisy image ($\sigma = 10$). In* (b) *and* (c)*, the parameter h is optimized for PSNR. The difference between them is barely visible. In regions* (1)*,* (2)*, and* (3)*, the adaptivity using SURE* (d) *allows us to reconstruct fine structures such as ropes and antennas. However, the filter leaves a noisy spot in region* (4) *when trying to preserve a fine rope.*

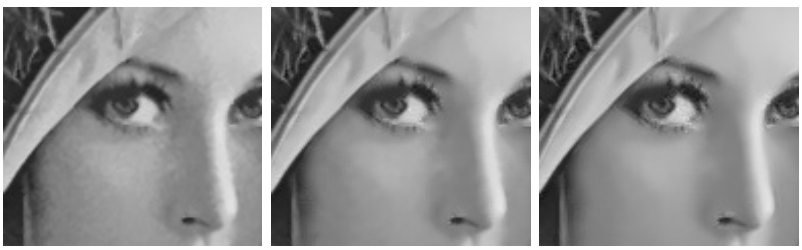


(a) Global h, $s^2 = 3 \times 3$ (b) Global h, $s^2 = 5 \times 5$ (c) Global h, $s^2 = 7 \times 7$

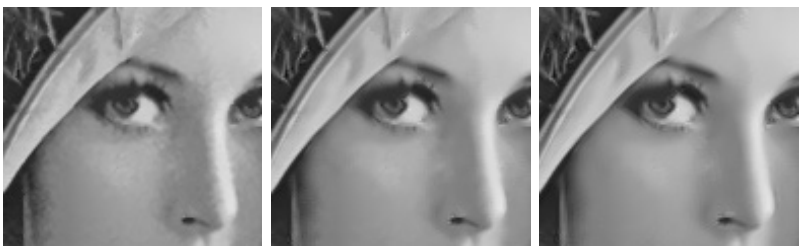


(d) Local h, $s^2 = 3 \times 3$ (e) Local h, $s^2 = 5 \times 5$ (f) Local h, $s^2 = 7 \times 7$

Figure 21. *Patch size and textures. Top: NLM with different patch sizes s^2 using a global parameter h optimized for PSNR. Bottom: local parameter h using LBMRE. The least contrasted textures are better preserved with a small patch size. However, what looks like texture with patch size 3×3 might as well be the mottling artifact (see below). With the local parameter h, the textures are preserved even with a large patch size.*



(a) Global h, $s^2 = 3 \times 3$ (b) Global h, $s^2 = 5 \times 5$ (c) Global h, $s^2 = 7 \times 7$



(d) Local h, $s^2 = 3 \times 3$ (e) Local h, $s^2 = 5 \times 5$ (f) Local h, $s^2 = 7 \times 7$

Figure 22. *Patch size and robustness to noise (same experiment as in Figure 21). With a too small patch size, the algorithm leaves too much noise: Lena's skin looks mottled. As the patch size increases, this effect reduces, but the rare patch artifact appears. With a local h, the rare patch effect is reduced, which allows us to use large patches.*

(a) Original image (b) Noisy Image ($\sigma = 10$)

(c) Global h (d) Local h, LBMRE

(e) Local h, LBEWA (f) BM3D

Figure 23. *Comparison on the country house image ($\sigma = 10$).*

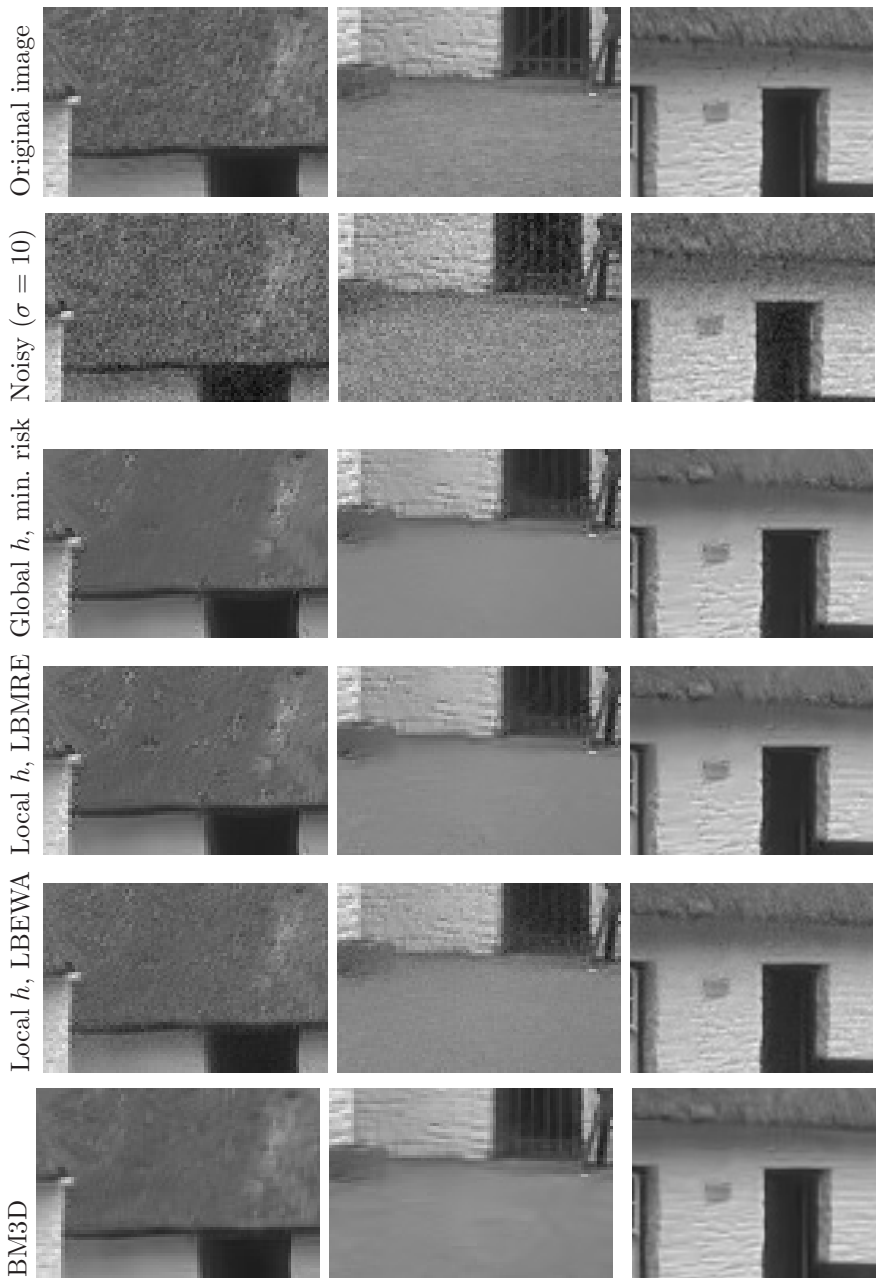


Figure 24. *Zoom of Figure* 23.

Acknowledgments. The first author thanks Joseph Salmon, Charles Deledalle, and Julien Rabin for fruitful discussions about this work.

REFERENCES

[1] S. P. Awate and R. T. Whitaker, *Unsupervised, information-theoretic, adaptive image filtering for image restoration*, IEEE Trans. Pattern Anal. Mach. Intell., 28 (2006), pp. 364–376.

[2] N. Azzabou, N. Paragios, and F. Guichard, *Uniform and textured regions separation in natural images towards MPM adaptive denoising*, in Proceedings of the 1st International Conference on Scale Space and Variational Methods in Computer Vision, Springer-Verlag, Berlin, Heidelberg, 2007, pp. 418–429.

[3] N. Azzabou, N. Paragios, F. Guichard, and F. Cao, *Variable bandwidth image denoising using image-based noise models*, in Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, 2007, pp. 1–7.

[4] T. Blu and F. Luisier, *The SURE-LET approach to image denoising*, IEEE Trans. Image Process., 16 (2007), pp. 2778–2786.

[5] T. Brox and D. Cremers, *Iterated nonlocal means for texture restoration*, in Proceedings of the 1st International Conference on Scale Space and Variational Methods in Computer Vision, Lecture Notes in Comput. Sci. 4485, F. Sgallari, A. Murli, and N. Paragios, eds., Springer-Verlag, Berlin, 2007, pp. 13–24.

[6] T. Brox, O. Kleinschmidt, and D. Cremers, *Efficient nonlocal means for denoising of textural patterns*, IEEE Trans. Image Process., 17 (2008), pp. 1083–1092.

[7] A. Buades, B. Coll, and J. Morel, *A review of image denoising algorithms, with a new one*, Multiscale Model. Simul., 4 (2005), pp. 490–530.

[8] P. L. Combettes and J.-C. Pesquet, *Wavelet-constrained image restoration*, Int. J. Wavelets Multiresolut. Inf. Process., 2 (2004), p. 371–389.

[9] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazaran, *Image denoising by sparse 3D-transform-domain collaborative filtering*, IEEE Trans. Image Process., 16 (2007), pp. 2080–2095.

[10] J. Darbon, A. Cunha, T.-F. Chan, S. Osher, and G. Jensen, *Fast nonlocal filtering applied to electron cryomicroscopy*, in Proceedings of the 5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro, 2008, pp. 1331–1334.

[11] C.-A. Deledalle, L. Denis, and F. Tupin, *Iterative weighted maximum likelihood denoising with probabilistic patch-based weights*, IEEE Trans. Image Process., 18 (2009), pp. 2661–2672.

[12] D. Donoho and I. Johnstone, *Adapting to unknown smoothness via wavelet shrinkage*, J. Amer. Statist. Assoc., 90 (1995), pp. 1200–1224.

[13] V. Dore and M. Cheriet, *Robust NL-means filter with optimal pixel-wise smoothing parameter for statistical image denoising*, IEEE Trans. Signal Process., 57 (2009), pp. 1703–1716.

[14] A. A. Efros and T. K. Leung, *Texture synthesis by non-parametric sampling*, in Proceedings of the IEEE International Conference on Computer Vision, Corfu, Greece, 1999, pp. 1033–1038.

[15] G. Gilboa and S. Osher, *Nonlocal linear image regularization and supervised segmentation*, Multiscale Model. Simul., 6 (2007), pp. 595–630.

[16] G. Gilboa and S. Osher, *Nonlocal operators with applications to image processing*, Multiscale Model. Simul., 7 (2008), pp. 1005–1028.

[17] B. Goossens, Q. Luong, A. Pizurica, and W. Philips, *An improved non-local denoising algorithm*, in Proceedings of the International Workshop on Local and Non-Local Approximation in Image Processing, 2008.

[18] C. Kervrann and J. Boulanger, *Local adaptivity to variable smoothness for exemplar-based image denoising and representation.*, Int. J. Comput. Vision, 79 (2008), pp. 45–69.

[19] S. Kindermann, S. Osher, and P. W. Jones, *Deblurring and denoising of images by nonlocal functionals*, Multiscale Model. Simul., 4 (2005), pp. 1091–1115.

[20] G. Leung and A. R. Barron, *Information theory and mixing least-squares regressions*, IEEE Trans. Inform. Theory, 52 (2006), pp. 3396–3410.

[21] C. Louchet and L. Moisan, *Total variation as a local filter*, SIAM J. Imaging Sci., 4 (2011), pp. 651–694.

[22] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, *Non-local sparse models for image restoration*, in Proceedings of the 12th IEEE International Conference on Computer Vision, 2009, pp. 2272–2279.
[23] G. Peyré, *Image processing with nonlocal spectral bases*, Multiscale Model. Simul., 7 (2008), pp. 703–730.
[24] G. Peyré, *Manifold models for signals and images*, Computer Vis. Image Underst., 113 (2009), pp. 249–260.
[25] S. Ramani, T. Blu, and M. Unser, *Monte-Carlo sure: A black-box optimization of regularization parameters for general denoising algorithms*, IEEE Trans. Image Process., 17 (2008), pp. 1540–1554.
[26] J. Salmon, *On two parameters for denoising with non-local means*, IEEE Signal Process. Lett., 17 (2010), pp. 269–272.
[27] J. Salmon and E. Le Pennec, *An aggregator point of view on NL-Means*, in Wavelets XIII, Proceedings of the SPIE, Vol. 7446, 2009, 74461E.
[28] J. Salmon and Y. Strozecki, *From patches to pixels in non-local methods: Weighted-average reprojection*, in Proceedings of the 17th IEEE International Conference on Image Processing, 2010, pp. 1929–1932.
[29] A. Singer, Y. Shkolnisky, and B. Nadler, *Diffusion interpretation of nonlocal neighborhood filters for signal denoising*, SIAM J. Imaging Sci., 2 (2009), pp. 118–139.
[30] V. Solo, *A sure-fired way to choose smoothing parameters in ill-conditioned inverse problems*, in Proceedings of the IEEE International Conference on Image Processing, 1996, pp. 89–92.
[31] C. Stein, *Estimation of the mean of a multivariate normal distribution*, Ann. Statist., 9 (1981), pp. 1135–1151.
[32] A. Szlam, *Non-Stationary Analysis on Datasets and Applications*, Ph.D. thesis, Yale University, New Haven, CT, 2006.
[33] A. Szlam, *Non-Local Means for Audio Denoising*, UCLA CAM Report 08-56, University of California, Los Angeles, CA, 2008.
[34] T. Tasdizen, *Principal neighborhood dictionaries for nonlocal means image denoising*, IEEE Trans. Image Process., 18 (2009), pp. 2649–2660.
[35] D. Tschumperlé and L. Brun, *Image denoising and registration by PDE's on the space of patches*, in Proceedings of the International Workshop on Local and Non-Local Approximation in Image Processing (LNLA'08), Lausanne, Switzerland, 2008.
[36] D. Tschumperlé and L. Brun, *Non-local image smoothing by applying anisotropic diffusion PDE's in the space of patches*, in Proceedings of the 16th IEEE International Conference on Image Processing (ICIP'09), Cairo, Egypt, 2009.
[37] D. Van De Ville and M. Kocher, *SURE based non-local means*, IEEE Signal Process. Lett., 16 (2009), pp. 973–976.
[38] D. Van De Ville and M. Kocher, *Non-local means with dimensionality reduction and SURE-based parameter selection*, IEEE Trans. Image Process., to appear.
[39] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, *Image quality assessment: From error visibility to structural similarity*, IEEE Trans. Signal Process., 13 (2004), pp. 600–612.
[40] L. Zhang, W. Dong, D. Zhang, and G. Shi, *Two-stage image denoising by principal component analysis with local pixel grouping*, Pattern Recogn., 43 (2010), pp. 1531–1549.
[41] X.-P. Zhang and Z.-Q. Luo, *A new time-scale adaptive denoising method based on wavelet shrinkage*, in ICASSP '99: Proceedings of the 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing, IEEE Computer Society, Washington, DC, 1999, pp. 1629–1632.