MULTISCALE MODEL. SIMUL.
Vol. 4, No. 2, pp. 490–530

A REVIEW OF IMAGE DENOISING ALGORITHMS, WITH A NEW ONE*

A. BUADES†, B. COLL†, AND J. M. MOREL‡

Abstract. The search for efficient image denoising methods is still a valid challenge at the crossing of functional analysis and statistics. In spite of the sophistication of the recently proposed methods, most algorithms have not yet attained a desirable level of applicability. All show an outstanding performance when the image model corresponds to the algorithm assumptions but fail in general and create artifacts or remove image fine structures. The main focus of this paper is, first, to define a general mathematical and experimental methodology to compare and classify classical image denoising algorithms and, second, to propose a nonlocal means (NL-means) algorithm addressing the preservation of structure in a digital image. The mathematical analysis is based on the analysis of the "method noise," defined as the difference between a digital image and its denoised version. The NL-means algorithm is proven to be asymptotically optimal under a generic statistical image model. The denoising performance of all considered methods are compared in four ways; mathematical: asymptotic order of magnitude of the method noise under regularity assumptions; perceptual-mathematical: the algorithms artifacts and their explanation as a violation of the image model; quantitative experimental: by tables of L^2 distances of the denoised version to the original image. The most powerful evaluation method seems, however, to be the visualization of the method noise on natural images. The more this method noise looks like a real white noise, the better the method.

Key words. image restoration, nonparametric estimation, PDE smoothing filters, adaptive filters, frequency domain filters

AMS subject classification. 62H35

DOI. 10.1137/040616024

1. Introduction.

1.1. Digital images and noise. The need for efficient image restoration methods has grown with the massive production of digital images and movies of all kinds, often taken in poor conditions. No matter how good cameras are, an image improvement is always desirable to extend their range of action.

A digital image is generally encoded as a matrix of grey-level or color values. In the case of a movie, this matrix has three dimensions, the third one corresponding to time. Each pair $(i, u(i))$, where $u(i)$ is the value at i, is called a pixel, short for "picture element." In the case of grey-level images, i is a point on a two-dimensional (2D) grid and $u(i)$ is a real value. In the case of classical color images, $u(i)$ is a triplet of values for the red, green, and blue components. All of what we shall say applies identically to movies, three-dimensional (3D) images, and color or multispectral images. For the sake of simplicity in notation and display of experiments, we shall here be content with rectangular 2D grey-level images.

*Received by the editors September 30, 2004; accepted for publication (in revised form) January 10, 2005; published electronically July 18, 2005.
http://www.siam.org/journals/mms/4-2/61602.html

†Universitat de les Illes Balears, Anselm Turmeda, Ctra. Valldemossa Km. 7.5, 07122 Palma de Mallorca, Spain (vdmiabc4@uib.es, tomeu.coll@uib.es). These authors were supported by the Ministerio de Ciencia y Tecnologia under grant TIC2002-02172. During this work, the first author had a fellowship of the Govern de les Illes Balears for the realization of his Ph.D. thesis.

‡Centre de Mathématiques et Leurs Applications, ENS Cachan 61, Av du Président Wilson 94235 Cachan, France (morel@cmla.ens-cachan.fr). This author was supported by the Centre National d'Etudes Spatiales (CNES), the Office of Naval Research under grant N00014-97-1-0839, the Direction Générale des Armements (DGA), and the Ministère de la Recherche et de la Technologie.

The two main limitations in image accuracy are categorized as blur and noise. Blur is intrinsic to image acquisition systems, as digital images have a finite number of samples and must satisfy the Shannon–Nyquist sampling conditions [31]. The second main image perturbation is noise.

Each one of the pixel values $u(i)$ is the result of a light intensity measurement, usually made by a charge coupled device (CCD) matrix coupled with a light focusing system. Each captor of the CCD is roughly a square in which the number of incoming photons is being counted for a fixed period corresponding to the obturation time. When the light source is constant, the number of photons received by each pixel fluctuates around its average in accordance with the central limit theorem. In other terms, one can expect fluctuations of order $\sqrt{n}$ for n incoming photons. In addition, each captor, if not adequately cooled, receives heat spurious photons. The resulting perturbation is usually called "obscurity noise." In a first rough approximation one can write

$$v(i) = u(i) + n(i),$$

where $i \in I$, $v(i)$ is the observed value, $u(i)$ would be the "true" value at pixel i, namely the one which would be observed by averaging the photon counting on a long period of time, and $n(i)$ is the noise perturbation. As indicated, the amount of noise is signal-dependent; that is, $n(i)$ is larger when $u(i)$ is larger. In noise models, the normalized values of $n(i)$ and $n(j)$ at different pixels are assumed to be independent random variables, and one talks about "white noise."

1.2. Signal and noise ratios. A good quality photograph (for visual inspection) has about 256 grey-level values, where 0 represents black and 255 represents white. Measuring the amount of noise by its standard deviation, $\sigma(n)$, one can define the signal noise ratio (SNR) as

$$SNR = \frac{\sigma(u)}{\sigma(n)},$$

where $\sigma(u)$ denotes the empirical standard deviation of u,

$$\sigma(u) = \left(\frac{1}{|I|} \sum_{i \in I} (u(i) - \overline{u})^2 \right)^{\frac{1}{2}},$$

and $\overline{u} = \frac{1}{|I|} \sum_{i \in I} u(i)$ is the average grey-level value. The standard deviation of the noise can also be obtained as an empirical measurement or formally computed when the noise model and parameters are known. A good quality image has a standard deviation of about 60.

The best way to test the effect of noise on a standard digital image is to add a Gaussian white noise, in which case $n(i)$ are independently and identically distributed (i.i.d.) Gaussian real variables. When $\sigma(n) = 3$, no visible alteration is usually observed. Thus, a $\frac{60}{3} \simeq 20$ SNR is nearly invisible. Surprisingly enough, one can add white noise up to a $\frac{2}{1}$ ratio and still *see* everything in a picture! This fact is illustrated in Figure 1 and constitutes a major enigma of human vision. It justifies the many attempts to define convincing denoising algorithms. As we shall see, the results have been rather deceptive. Denoising algorithms see no difference between small details and noise, and therefore they remove them. In many cases, they create new distortions, and the researchers are so used to them that they have created a

FIG. 1. *A digital image with standard deviation* 55*, the same with noise added (standard deviation* 3*), the SNR therefore being equal to* 18*, and the same with SNR slightly larger than* 2*. In this second image, no alteration is visible. In the third, a conspicuous noise with standard deviation* 25 *has been added, but, surprisingly enough, all details of the original image still are visible.*

taxonomy of denoising artifacts: "ringing," "blur," "staircase effect," "checkerboard effect," "wavelet outliers," etc.

This fact is not quite a surprise. Indeed, to the best of our knowledge, all denoising algorithms are based on

- a noise model;
- a generic image smoothness model, local or global.

In experimental settings, the noise model is perfectly precise. So the weak point of the algorithms is the inadequacy of the image model. All of the methods assume that the noise is oscillatory and that the image is smooth or piecewise smooth. So they try to separate the smooth or patchy part (the image) from the oscillatory one. Actually, many fine structures in images are as oscillatory as noise is; conversely, white noise has low frequencies and therefore smooth components. Thus a separation method based on smoothness arguments only is hazardous.

1.3. The "method noise." All denoising methods depend on a filtering parameter h. This parameter measures the degree of filtering applied to the image. For most methods, the parameter h depends on an estimation of the noise variance σ^2. One can define the result of a denoising method D_h as a decomposition of any image v as

$$v = D_h v + n(D_h, v), \tag{1.1}$$

where

1. $D_h v$ is more smooth than v,
2. $n(D_h, v)$ is the noise guessed by the method.

Now it is not enough to smooth v to ensure that $n(D_h, v)$ will look like a noise. The more recent methods are actually not content with a smoothing but try to recover lost information in $n(D_h, v)$ [19, 25]. So the focus is on $n(D_h, v)$.

DEFINITION 1.1 (method noise). *Let u be a (not necessarily noisy) image and D_h a denoising operator depending on h. Then we define the* method noise of u *as the image difference*

$$n(D_h, u) = u - D_h(u). \tag{1.2}$$

This method noise should be as similar to a white noise as possible. In addition, since we would like the original image u not to be altered by denoising methods, the

method noise should be as small as possible for the functions with the right regularity.

According to the preceding discussion, four criteria can and will be taken into account in the comparison of denoising methods:

- A display of typical artifacts in denoised images.
- A formal computation of the method noise on smooth images, evaluating how small it is in accordance with image local smoothness.
- A comparative display of the *method noise* of each method on real images with $\sigma = 2.5$. We mentioned that a noise standard deviation smaller than 3 is subliminal, and it is expected that most digitization methods allow themselves this kind of noise.
- A classical comparison receipt based on noise simulation: it consists of taking a good quality image, adding Gaussian white noise with known σ, and then computing the best image recovered from the noisy one by each method. A table of L^2 distances from the restored to the original can be established. The L^2 distance does not provide a good quality assessment. However, it reflects well the relative performances of algorithms.

On top of this, in two cases, a proof of asymptotic recovery of the image can be obtained by statistical arguments.

1.4. Which methods to compare. We had to make a selection of the denoising methods we wished to compare. Here a difficulty arises, as most original methods have caused an abundant literature proposing many improvements. So we tried to get the best available version, while keeping the simple and genuine character of the original method: no hybrid method. So we shall analyze the following:

1. the Gaussian smoothing model (Gabor quoted in Lindenbaum, Fischer, and Bruckstein [17]), where the smoothness of u is measured by the Dirichlet integral $\int |Du|^2$;
2. the anisotropic filtering model (Perona and Malik [27], Alvarez, Lions, and Morel [1]);
3. the Rudin–Osher–Fatemi total variation model [30] and two recently proposed iterated total variation refinements [35, 25];
4. the Yaroslavsky neighborhood filters [41, 40] and an elegant variant, the SUSAN filter (Smith and Brady [33]);
5. the Wiener local empirical filter as implemented by Yaroslavsky [40];
6. the translation invariant wavelet thresholding [8], a simple and performing variant of the wavelet thresholding [10];
7. DUDE, the discrete universal denoiser [24], and the UINTA, unsupervised information-theoretic, adaptive filtering [3], two very recent new approaches;
8. the nonlocal means (NL-means) algorithm, which we introduce here.

This last algorithm is given by a simple closed formula. Let u be defined in a bounded domain $\Omega \subset \mathbb{R}^2$; then

$$NL(u)(\mathbf{x}) = \frac{1}{C(\mathbf{x})} \int e^{-\frac{(G_a * |u(\mathbf{x}+.) - u(\mathbf{y}+.)|^2)(0)}{h^2}} \, u(\mathbf{y}) \, d\mathbf{y},$$

where $\mathbf{x} \in \Omega$, G_a is a Gaussian kernel of standard deviation a, h acts as a filtering parameter, and $C(\mathbf{x}) = \int e^{-\frac{(G_a * |u(\mathbf{x}+.) - u(\mathbf{z}+.)|^2)(0)}{h^2}} d\mathbf{z}$ is the normalizing factor. In order to make clear the previous definition, we recall that

$$(G_a * |u(\mathbf{x}+.) - u(\mathbf{y}+.)|^2)(0) = \int_{\mathbb{R}^2} G_a(\mathbf{t}) |u(\mathbf{x}+\mathbf{t}) - u(\mathbf{y}+\mathbf{t})|^2 d\mathbf{t}.$$

This amounts to saying that $NL(u)(\mathbf{x})$, the denoised value at $\mathbf{x}$, is a mean of the values of all pixels whose Gaussian neighborhood looks like the neighborhood of $\mathbf{x}$.

1.5. What is left. We do not draw into comparison the hybrid methods, in particular the total variation + wavelets [7, 12, 18]. Such methods are significant improvements of the simple methods but are impossible to draw into a benchmark: their efficiency depends a lot upon the choice of wavelet dictionaries and the kind of image.

Second, we do not draw into the comparison the method introduced recently by Meyer [22], whose aim it is to decompose the image into a BV part and a texture part (the so called $u + v$ methods), and even into three terms, namely $u + v + w$, where u is the BV part, v is the "texture" part belonging to the dual space of BV, denoted by G, and w belongs to the Besov space $\dot{B}^{\infty}_{-1,\infty}$, a space characterized by the fact that the wavelet coefficients have a uniform bound. G is proposed by Meyer as the right space to model oscillatory patterns such as textures. The main focus of this method is not yet denoising. Because of the different and more ambitious scopes of the Meyer method [2, 36, 26], which makes it parameter- and implementation-dependent, we could not draw it into the discussion. Last but not least, let us mention the bandlet [15] and curvelet [34] transforms for image analysis. These methods also are separation methods between the geometric part and the oscillatory part of the image and intend to find an accurate and compressed version of the geometric part. Incidentally, they may be considered as denoising methods in geometric images, as the oscillatory part then contains part of the noise. Those methods are closely related to the total variation method and to the wavelet thresholding, and we shall be content with those simpler representatives.

1.6. Plan of the paper. Section 2 computes formally the *method noise* for the best elementary local smoothing methods, namely Gaussian smoothing, anisotropic smoothing (mean curvature motion), total variation minimization, and the neighborhood filters. For all of them we prove or recall the asymptotic expansion of the filter at smooth points of the image and therefore obtain a formal expression of the method noise. This expression permits us to characterize places where the filter performs well and where it fails. In section 3, we treat the Wiener-like methods, which proceed by a soft or hard threshold on frequency or space-frequency coefficients. We examine in turn the Wiener–Fourier filter, the Yaroslavsky local adaptive discrete cosine transform (DCT)-based filters, and the wavelet threshold method. Of course, the Gaussian smoothing belongs to both classes of filters. We also describe the universal denoiser DUDE, but we cannot draw it into the comparison, as its direct application to grey-level images is unpractical so far. (We discuss its feasibility.) Finally, we examine the UINTA algorithms whose principles stand close to the NL-means algorithm. In section 5, we introduce the NL-means filter. This method is not easily classified in the preceding terminology, since it can work adaptively in a local or nonlocal way. We first give a proof that this algorithm is asymptotically consistent (it gives back the conditional expectation of each pixel value given an observed neighborhood) under the assumption that the image is a fairly general stationary random process. The works of Efros and Leung [13] and Levina [16] have shown that this assumption is sound for images having enough samples in each texture patch. In section 6, we compare all algorithms from several points of view, do a performance classification, and explain why the NL-means algorithm shares the consistency properties of most of the aforementioned algorithms.

2. Local smoothing filters. The original image u is defined in a bounded domain $\Omega \subset \mathbb{R}^2$ and denoted by $u(\mathbf{x})$ for $\mathbf{x} = (x, y) \in \mathbb{R}^2$. This continuous image is usually interpreted as the Shannon interpolation of a discrete grid of samples [31] and is therefore analytic. The distance between two consecutive samples will be denoted by ε.

The noise itself is a discrete phenomenon on the sampling grid. According to the usual screen and printing visualization practice, we do not interpolate the noise samples n_i as a band limited function but rather as a piecewise constant function, constant on each pixel i and equal to n_i.

We write $|\mathbf{x}| = (x^2+y^2)^{\frac{1}{2}}$ and $\mathbf{x}_1.\mathbf{x}_2 = x_1x_2+y_1y_2$ as the norm and scalar product and denote the derivatives of u by $u_x = \frac{\partial u}{\partial x}$, $u_y = \frac{\partial u}{\partial y}$, and $u_{xy} = \frac{\partial^2 u}{\partial x \partial y}$. The gradient of u is written as $Du = (u_x, u_y)$ and the Laplacian of u as $\Delta u = u_{xx} + u_{yy}$.

2.1. Gaussian smoothing. By Riesz's theorem, image isotropic linear filtering boils down to a convolution of the image by a linear radial kernel. The smoothing requirement is usually expressed by the positivity of the kernel. The paradigm of such kernels is, of course, the Gaussian $\mathbf{x} \to G_h(\mathbf{x}) = \frac{1}{(4\pi h^2)} e^{-\frac{|\mathbf{x}|^2}{4h^2}}$. In that case, G_h has standard deviation h, and the following theorem is easily seen.

THEOREM 2.1 (Gabor 1960). *The image method noise of the convolution with a Gaussian kernel G_h is*

$$u - G_h * u = -h^2 \Delta u + o(h^2).$$

A similar result is actually valid for any positive radial kernel with bounded variance, so one can keep the Gaussian example without loss of generality. The preceding estimate is valid if h is small enough. On the other hand, the noise reduction properties depend upon the fact that the neighborhood involved in the smoothing is large enough, so that the noise gets reduced by averaging. So in the following we assume that $h = k\varepsilon$, where k stands for the number of samples of the function u and noise n in an interval of length h. The spatial ratio k must be much larger than 1 to ensure a noise reduction.

The effect of a Gaussian smoothing on the noise can be evaluated at a reference pixel $i = 0$. At this pixel,

$$G_h * n(0) = \sum_{i \in I} \int_{P_i} G_h(\mathbf{x}) n(\mathbf{x}) d\mathbf{x} = \sum_{i \in I} \varepsilon^2 G_h(i) n_i,$$

where we recall that $n(x)$ is being interpolated as a piecewise constant function, the P_i square pixels centered in i have size ε^2, and $G_h(i)$ denotes the mean value of the function G_h on the pixel i.

Denoting by $Var(X)$ the variance of a random variable X, the additivity of variances of independent centered random variables yields

$$Var(G_h * n(0)) = \sum_i \varepsilon^4 G_h(i)^2 \sigma^2 \simeq \sigma^2 \varepsilon^2 \int G_h(\mathbf{x})^2 d\mathbf{x} = \frac{\varepsilon^2 \sigma^2}{8\pi h^2}.$$

So we have proved the following theorem.

THEOREM 2.2. *Let $n(x)$ be a piecewise constant white noise, with $n(\boldsymbol{x}) = n_i$ on each square pixel i. Assume that the n_i are i.i.d. with zero mean and variance σ^2. Then the "noise residue" after a Gaussian convolution of n by G_h satisfies*

$$Var(G_h * n(0)) \simeq \frac{\varepsilon^2 \sigma^2}{8\pi h^2}.$$

In other terms, the standard deviation of the noise, which can be interpreted as the noise amplitude, is multiplied by $\frac{\varepsilon}{h\sqrt{8\pi}}$.

Theorems 2.1 and 2.2 traduce the delicate equilibrium between noise reduction and image destruction by any linear smoothing. Denoising does not alter the image at points where it is smooth at a scale h much larger than the sampling scale ε. The first theorem tells us that the *method noise* of the Gaussian denoising method is zero in harmonic parts of the image. A Gaussian convolution is optimal on harmonic functions and performs instead poorly on singular parts of u, namely edges or texture, where the Laplacian of the image is large. See Figure 3.

2.2. Anisotropic filters and curvature motion. The anisotropic filter (AF) attempts to avoid the blurring effect of the Gaussian by *convolving the image* u *at* $\boldsymbol{x}$ *only in the direction orthogonal to* $Du(\boldsymbol{x})$. The idea of such a filter goes back to Perona and Malik [27] and actually again to Gabor (quoted in Lindenbaum, Fischer, and Bruckstein [17]). Set

$$AF_h u(\mathbf{x}) = \int G_h(t) u\left(\mathbf{x} + t\frac{Du(\mathbf{x})^\perp}{|Du(\mathbf{x})|}\right) dt$$

for $\mathbf{x}$ such that $Du(\mathbf{x}) \neq 0$ and where $(x,y)^\perp = (-y,x)$ and $G_h(t) = \frac{1}{\sqrt{2\pi}h} e^{-\frac{t^2}{2h^2}}$ is the one-dimensional (1D) Gauss function with variance h^2. At points where $Du(\mathbf{x}) = 0$ an isotropic Gaussian mean is usually applied, and the result of Theorem 2.1 holds at those points. If one assumes that the original image u is twice continuously differentiable (C^2) at $\mathbf{x}$, the following theorem is easily shown by a second-order Taylor expansion.

THEOREM 2.3. *The image method noise of an anisotropic filter* AF_h *is*

$$u(\boldsymbol{x}) - AF_h u(\boldsymbol{x}) \simeq -\frac{1}{2}h^2 D^2 u\left(\frac{Du^\perp}{|Du|}, \frac{Du^\perp}{|Du|}\right) = -\frac{1}{2}h^2 |Du| curv(u)(\boldsymbol{x}),$$

where the relation holds when $Du(\boldsymbol{x}) \neq 0$.

By $curv(u)(\mathbf{x})$, we denote the curvature, i.e., the signed inverse of the radius of curvature of the level line passing by $\mathbf{x}$. When $Du(\mathbf{x}) \neq 0$, this means that

$$curv(u) = \frac{u_{xx}u_y^2 - 2u_{xy}u_x u_y + u_{yy}u_x^2}{(u_x^2 + u_y^2)^{\frac{3}{2}}}.$$

This method noise is zero wherever u behaves locally like a one-variable function, $u(x,y) = f(ax + by + c)$. In such a case, the level line of u is locally the straight line with equation $ax + by + c = 0$, and the gradient of f may instead be very large. In other terms, *with anisotropic filtering, an edge can be maintained.* On the other hand, we have to evaluate the Gaussian noise reduction. This is easily done by a 1D adaptation of Theorem 2.2. Notice that the noise on a grid is not isotropic; so the Gaussian average when Du is parallel to one coordinate axis is made roughly on $\sqrt{2}$ more samples than the Gaussian average in the diagonal direction.

THEOREM 2.4. *By anisotropic Gaussian smoothing, when* ε *is small enough with respect to* h*, the noise residue satisfies*

$$Var\,(AF_h(n)) \leq \frac{\varepsilon}{\sqrt{2\pi}h}\sigma^2.$$

In other terms, the standard deviation of the noise n is multiplied by a factor at most equal to $(\frac{\varepsilon}{\sqrt{2\pi}h})^{1/2}$*, this maximal value being attained in the diagonals.*

Proof. Let L be the line $\mathbf{x} + t\frac{Du^{\perp}(\mathbf{x})}{|Du(\mathbf{x})|}$ passing by $\mathbf{x}$, parameterized by $t \in \mathbb{R}$, and denote by P_i, $i \in I$, the pixels which meet L, $n(i)$ the noise value, constant on pixel P_i, and ε_i the length of the intersection of $L \cap P_i$. Denote by $g(i)$ the average of $G_h(\mathbf{x} + t\frac{Du^{\perp}(\mathbf{x})}{|Du(\mathbf{x})|})$ on $L \cap P_i$. Then one has

$$AF_h n(\mathbf{x}) \simeq \sum_i \varepsilon_i n(i) g(i).$$

The $n(i)$ are i.i.d. with standard variation σ, and therefore

$$Var(AF_h(n)) = \sum_i \varepsilon_i^2 \sigma^2 g(i)^2 \leq \sigma^2 \max(\varepsilon_i) \sum_i \varepsilon_i g(i)^2.$$

This yields

$$\text{Var } (AF_h(n)) \leq \sqrt{2}\varepsilon\sigma^2 \int G_h(t)^2 dt = \frac{\varepsilon}{\sqrt{2\pi}h}\sigma^2. \qquad \square$$

There are many versions of AF_h, all yielding an asymptotic estimate equivalent to the one in Theorem 2.3: the famous median filter [14], an inf-sup filter on segments centered at $\mathbf{x}$ [5], and the clever numerical implementation of the mean curvature equation in [21]. So all of those filters have in common the good preservation of edges, but they perform poorly on flat regions and are worse there than a Gaussian blur. This fact derives from the comparison of the noise reduction estimates of Theorems 2.1 and 2.4 and is experimentally patent in Figure 3.

2.3. Total variation. The total variation minimization was introduced by Rudin and Osher [29] and Rudin, Osher, and Fatemi [30]. The original image u is supposed to have a simple geometric description, namely a set of connected sets, the objects, along with their smooth contours, or edges. The image is smooth inside the objects but with jumps across the boundaries. The functional space modeling these properties is $BV(\Omega)$, the space of integrable functions with finite total variation $TV_\Omega(u) = \int |Du|$, where Du is assumed to be a Radon measure. Given a noisy image $v(\mathbf{x})$, the above-mentioned authors proposed to recover the original image $u(\mathbf{x})$ as the solution of the constrained minimization problem

$$\arg\min_u TV_\Omega(u), \tag{2.1}$$

subject to the noise constraints

$$\int_\Omega (u(\mathbf{x}) - v(\mathbf{x}))d\mathbf{x} = 0 \quad \text{and} \quad \int_\Omega |u(\mathbf{x}) - v(\mathbf{x})|^2 d\mathbf{x} = \sigma^2.$$

The solution u must be as regular as possible in the sense of the total variation, while the difference $v(\mathbf{x}) - u(\mathbf{x})$ is treated as an error, with a prescribed energy. The constraints prescribe the right mean and variance to $u - v$ but do not ensure that it is similar to a noise (see a thorough discussion in [22]). The preceding problem is naturally linked to the unconstrained problem

$$\arg\min_u TV_\Omega(u) + \lambda \int_\Omega |v(\mathbf{x}) - u(\mathbf{x})|^2 d\mathbf{x} \tag{2.2}$$

for a given Lagrange multiplier λ. The above functional is strictly convex and lower semicontinuous with respect to the weak-star topology of BV. Therefore the minimum exists, is unique, and is computable (see, e.g., [6]). The parameter λ controls the tradeoff between the regularity and fidelity terms. As λ gets smaller the weight of the regularity term increases. Therefore λ is related to the degree of filtering of the solution of the minimization problem. Let us denote by $TVF_\lambda(v)$ the solution of problem (2.2) for a given value of λ. The Euler–Lagrange equation associated with the minimization problem is given by

$$(u(\mathbf{x}) - v(\mathbf{x})) - \frac{1}{2\lambda} curv(u)(\mathbf{x}) = 0$$

(see [29]). Thus, we have the following theorem.

THEOREM 2.5. *The image method noise of the total variation minimization* (2.2) *is*

$$u(\boldsymbol{x}) - TVF_\lambda(u)(\boldsymbol{x}) = -\frac{1}{2\lambda} curv(TVF_\lambda(u))(\boldsymbol{x}).$$

As in the anisotropic case, straight edges are maintained because of their small curvature. However, details and texture can be oversmoothed if λ is too small, as is shown in Figure 3.

2.4. Iterated total variation refinement. In the original total variation model the removed noise, $v(\mathbf{x}) - u(\mathbf{x})$, is treated as an error and is no longer studied. In practice, some structures and texture are present in this error. Several recent works have tried to avoid this effect [35, 25].

2.4.1. The Tadmor–Nezzar–Vese approach. In [35], the authors have proposed to use the Rudin–Osher–Fatemi model iteratively. They decompose the noisy image, $v = u_0 + n_0$, by the total variation model. So taking u_0 to contain only geometric information, they decompose by the very same model $n_0 = u_1 + n_1$, where u_1 is assumed to be again a geometric part and n_1 contains less geometric information than n_0. Iterating this process, one obtains $u = u_0 + u_1 + u_2 + \cdots + u_k$ as a refined geometric part and n_k as the noise residue. This strategy is in some sense close to the matching pursuit methods [20]. Of course, the weight parameter in the Rudin–Osher–Fatemi model has to grow at each iteration, and the authors propose a geometric series $\lambda, 2\lambda, \ldots, 2^k\lambda$. In that way, the extraction of the geometric part n_k becomes twice more taxing at each step. Then the new algorithm is as follows:

1. Starting with an initial scale $\lambda = \lambda_0$,

$$v = u_0 + n_0, \qquad [u_0, n_0] = \arg\min_{v=u+n} \int |Du| + \lambda_0 \int |v(\mathbf{x}) - u(\mathbf{x})|^2 d\mathbf{x}.$$

2. Proceed with successive applications of the dyadic refinement $n_j = u_{j+1} + n_{j+1}$,

$$[u_{j+1}, n_{j+1}] = \arg\min_{n_j=u+n} \int |Du| + \lambda_0 2^{j+1} \int |n_j(\mathbf{x}) - u(\mathbf{x})|^2 d\mathbf{x}.$$

3. After k steps, we get the following hierarchical decomposition of v:

$$\begin{aligned} v &= u_0 + n_0 \\ &= u_0 + u_1 + n_1 \\ &= ..... \\ &= u_0 + u_1 + \cdots + u_k + n_k. \end{aligned}$$

The denoised image is given by the partial sum $\sum_{j=0}^{k} u_j$, and n_k is the noise residue. This is a multilayered decomposition of v which lies in an intermediate scale of spaces, in between BV and L^2. Some theoretical results on the convergence of this expansion are presented in [35].

2.4.2. The Osher et al. approach. The second algorithm due to Osher et al. [25] also consists of an iteration of the original model. The new algorithm is as follows:

1. First, solve the original total variation model
$$u_1 = \arg\min_{u\in BV}\left\{\int |\nabla u(\mathbf{x})|d\mathbf{x} + \lambda\int (v(\mathbf{x}) - u(\mathbf{x}))^2 d\mathbf{x}\right\}$$
to obtain the decomposition $v = u_1 + n_1$.
2. Perform a correction step to obtain
$$u_2 = \arg\min_{u\in BV}\left\{\int |\nabla u(\mathbf{x})|d\mathbf{x} + \lambda\int (v(\mathbf{x}) + n_1(x) - u(\mathbf{x}))^2 d\mathbf{x}\right\},$$
where n_1 is the noise estimated by the first step. The correction step adds this first estimate of the noise to the original image and raises the decomposition $v + n_1 = u_2 + n_2$.
3. Iterate: compute u_{k+1} as a minimizer of the modified total variation minimization,
$$u_{k+1} = \arg\min_{u\in BV}\left\{\int |\nabla u(\mathbf{x})|d\mathbf{x} + \lambda\int (v(\mathbf{x}) + n_k(x) - u(\mathbf{x}))^2 d\mathbf{x}\right\},$$
where
$$v + n_k = u_{k+1} + n_{k+1}.$$

Some results are presented in [25] which clarify the nature of the above sequence:

- $\{u_k\}_k$ converges monotonically in L^2 to v, the noisy image, as $k \to \infty$.
- $\{u_k\}_k$ approaches the noise-free image monotonically in the Bregman distance associated with the BV seminorm, at least until $\|u_{\bar{k}} - u\| \le \sigma^2$, where u is the original image and σ is the standard deviation of the added noise.

These two results indicate how to stop the sequence and choose $u_{\bar{k}}$. It is enough to proceed iteratively until the result gets noisier or the distance $\|u_{\bar{k}} - u\|^2$ gets smaller than σ^2. The new solution has more details preserved, as Figure 3 shows.

The above iterated denoising strategy being quite general, one can make the computations for a linear denoising operator T as well. In that case, this strategy
$$T(v + n_1) = T(v) + T(n_1)$$
amounts to saying that the first estimated noise n_1 is filtered again and its smooth components are added back to the original, which is in fact the Tadmor–Nezzar–Vese strategy.

2.5. Neighborhood filters. The previous filters are based on a notion of spatial neighborhood or proximity. *Neighborhood filters* instead take into account grey-level values to define neighboring pixels. In the simplest and more extreme case, the denoised value at pixel i is an average of values at pixels which have a grey-level value close to $u(i)$. The grey-level neighborhood is therefore
$$B(i,h) = \{j \in I \mid u(i) - h < u(j) < u(i) + h\}.$$

This is a fully nonlocal algorithm, since pixels belonging to the whole image are used for the estimation at pixel i. This algorithm can be written in a more continuous form,

$$NF_h u(\mathbf{x}) = \frac{1}{C(\mathbf{x})} \int_\Omega u(\mathbf{y}) e^{-\frac{|u(\mathbf{y})-u(\mathbf{x})|^2}{h^2}} d\mathbf{y},$$

where $\Omega \subset \mathbb{R}^2$ is an open and bounded set, and $C(\mathbf{x}) = \int_\Omega e^{-\frac{|u(\mathbf{y})-u(\mathbf{x})|^2}{h^2}} d\mathbf{y}$ is the normalization factor. The first question to address here is the consistency of such a filter, namely, how close the denoised version is to the original when u is smooth.

LEMMA 2.6. *Suppose u is Lipschitz in Ω and $h > 0$; then $C(\boldsymbol{x}) \geq O(h^2)$.*

Proof. Given $\mathbf{x}, \mathbf{y} \in \Omega$, by the mean value theorem, $|u(\mathbf{x}) - u(\mathbf{y})| \leq K|\mathbf{x} - \mathbf{y}|$ for some real constant K. Then $C(\mathbf{x}) = \int_\Omega e^{-\frac{|u(\mathbf{y})-u(\mathbf{x})|^2}{h^2}} d\mathbf{y} \geq \int_{B(\mathbf{x},h)} e^{-\frac{|u(\mathbf{y})-u(\mathbf{x})|^2}{h^2}} d\mathbf{y} \geq e^{-K^2} O(h^2)$. ∎

PROPOSITION 2.7 (method noise estimate). *Suppose u is a Lipschitz bounded function on Ω, where Ω is an open and bounded domain of $\mathbb{R}^2$. Then $|u(\boldsymbol{x}) - NF_h u(\boldsymbol{x})| = O(h\sqrt{-\log h})$ for h small, $0 < h < 1$, $\boldsymbol{x} \in \Omega$.*

Proof. Let $\mathbf{x}$ be a point of Ω, and for a given B and h, $B, h \in \mathbb{R}$, consider the set $D_h = \{\mathbf{y} \in \Omega \mid |u(\mathbf{y}) - u(\mathbf{x})| \leq Bh\}$. Then

$$\begin{aligned} |u(\mathbf{x}) - NF_h u(\mathbf{x})| \leq & \frac{1}{C} \int_{D_h} e^{-\frac{|u(\mathbf{y})-u(\mathbf{x})|^2}{h^2}} |u(\mathbf{y}) - u(\mathbf{x})| d\mathbf{y} \\ & + \frac{1}{C} \int_{D_h^c} e^{-\frac{|u(\mathbf{y})-u(\mathbf{x})|^2}{h^2}} |u(\mathbf{y}) - u(\mathbf{x})| d\mathbf{y}. \end{aligned}$$

On one hand, considering that $\int_{D_h} e^{-\frac{|u(\mathbf{y})-u(\mathbf{x})|^2}{h^2}} d\mathbf{y} \leq C(\mathbf{x})$ and $|u(\mathbf{y}) - u(\mathbf{x})| \leq Bh$ for $\mathbf{y} \in D_h$ one sees that the first term is bounded by Bh. On the other hand, considering that $e^{-\frac{|u(\mathbf{y})-u(\mathbf{x})|^2}{h^2}} \leq e^{-B^2}$ for $\mathbf{y} \notin D_h$, $\int_{D_h^c} |u(\mathbf{y}) - u(\mathbf{x})| d\mathbf{y}$ is bounded, and by Lemma 2.6, $C \geq O(h^2)$, one deduces that the second term has an order $O(h^{-2} e^{-B^2})$. Finally, choosing B such that $B^2 = -3 \log h$ yields

$$|u(\mathbf{x}) - NF_h u(\mathbf{x})| \leq Bh + O(h^{-2} e^{-B^2}) = O(h\sqrt{-\log h}) + O(h),$$

and so the method noise has order $O(h\sqrt{-\log h})$. ∎

The Yaroslavsky neighborhood filters [40, 41] consider mixed neighborhoods $B(i,h) \cap B_\rho(i)$, where $B_\rho(i)$ is a ball of center i and radius ρ. So the method takes an average of the values of pixels which are both close in grey-level and spatial distance. This filter can be easily written in a continuous form as

$$YNF_{h,\rho}(\mathbf{x}) = \frac{1}{C(\mathbf{x})} \int_{B_\rho(\mathbf{x})} u(\mathbf{y}) e^{-\frac{|u(\mathbf{y})-u(\mathbf{x})|^2}{h^2}} d\mathbf{y},$$

where $C(\mathbf{x}) = \int_{B_\rho(\mathbf{x})} e^{-\frac{|u(\mathbf{y})-u(\mathbf{x})|^2}{h^2}} d\mathbf{y}$ is the normalization factor. In [33] the authors present a similar approach where they do not consider a ball of radius ρ but weigh the distance to the central pixel, obtaining the following close formula:

$$\frac{1}{C(\mathbf{x})} \int u(\mathbf{y}) e^{-\frac{|\mathbf{y}-\mathbf{x}|^2}{\rho^2}} e^{-\frac{|u(\mathbf{y})-u(\mathbf{x})|^2}{h^2}} d\mathbf{y},$$

where $C(\mathbf{x}) = \int e^{-\frac{|\mathbf{y}-\mathbf{x}|^2}{\rho^2}} e^{-\frac{|u(\mathbf{y})-u(\mathbf{x})|^2}{h^2}} d\mathbf{y}$ is the normalization factor.

First, we study the method noise of the $YNF_{h,\rho}$ in the 1D case. In that case, u denotes a 1D signal.

THEOREM 2.8. *Suppose $u \in \mathcal{C}^2((a,b))$, $a,b \in \mathbb{R}$. Then, for $0 < \rho \ll h$ and $h \to 0$,*

$$u(s) - YNF_{h,\rho}u(s) \simeq -\frac{\rho^2}{2} f\left(\frac{\rho}{h}|u'(s)|\right) u''(s),$$

where

$$f(t) = \frac{\frac{1}{3} - \frac{3}{5}t^2}{1 - \frac{1}{3}t^2}.$$

Proof. Let $s \in (a,b)$ and $h, \rho \in \mathbb{R}^+$. Then

$$u(s) - YNF_{h,\rho}(s) = -\frac{1}{\int_{-\rho}^{\rho} e^{-\frac{(u(s+t)-u(s))^2}{h^2}}\, dt} \int_{-\rho}^{\rho} (u(s+t) - u(s)) e^{-\frac{(u(s+t)-u(s))^2}{h^2}}\, dt.$$

If we take the Taylor expansion of $u(s+t)$ and the exponential function e^{-x^2} and we integrate, then we obtain that

$$u(s) - YNF_{h,\rho}(s) \simeq -\frac{\frac{\rho^3 u''}{3} - \frac{3\rho^5 u'^2 u''}{5h^2}}{2h - \frac{2\rho^3 u'^2}{3h^2}}$$

for ρ small enough. The method noise follows from the above expression. ∎

The previous result shows that the neighborhood filtering method noise is proportional to the second derivative of the signal. That is, it behaves like a weighted heat equation. The function f gives the sign and the magnitude of this heat equation. Where the function f takes positive values, the method noise behaves as a pure heat equation, while where it takes negative values, the method noise behaves as a reverse heat equation. The zeros and the discontinuity points of f represent the singular points where the behavior of the method changes. The magnitude of this change is much larger near the discontinuities of f producing an amplified shock effect. Figure 2 displays one experiment with the 1D neighborhood filter. We iterate the algorithm on a sine signal and illustrate the shock effect. For the two intermediate iterations u_{n+1}, we display the signal $f(\frac{\rho}{h}|u_n'|)$ which gives the sign and magnitude of the heat equation at each point. We can see that the positions of the discontinuities of $f(\frac{\rho}{h}|u_n'|)$ describe exactly the positions of the shocks in the further iterations and the final state. These two examples corroborate Theorem 2.8 and show how the function f totally characterizes the performance of the 1D neighborhood filter.

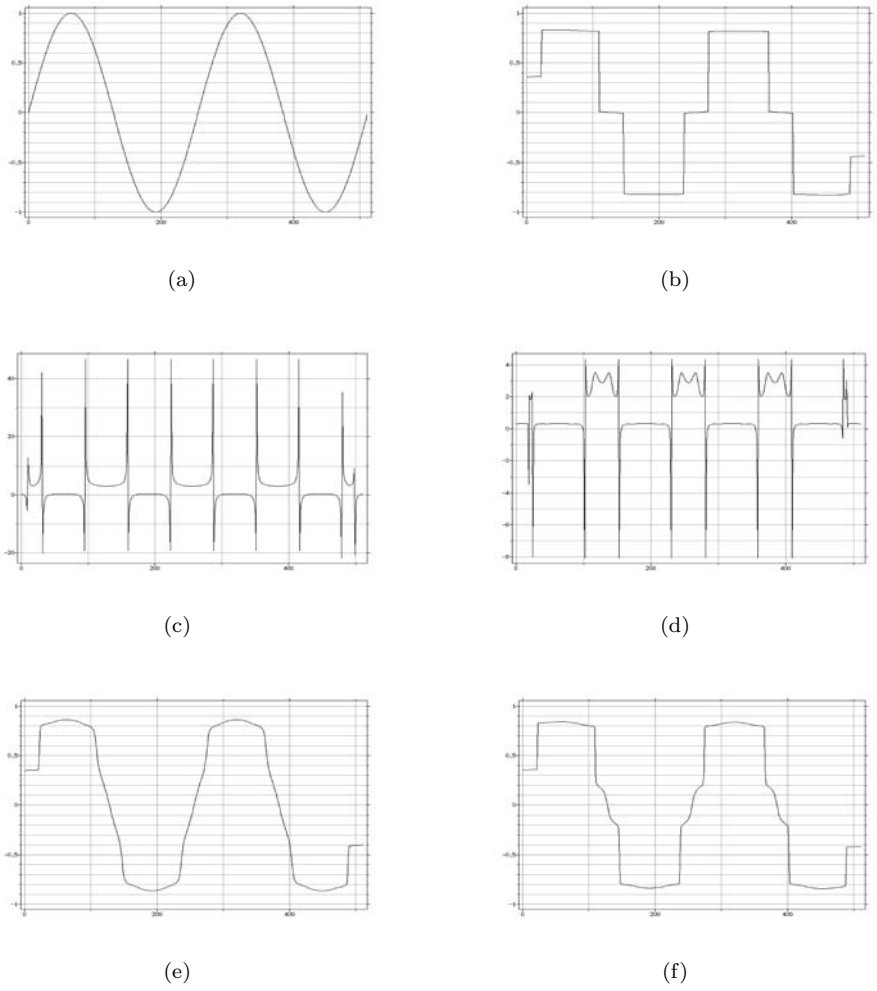
Next we give the analogous result for 2D images.

THEOREM 2.9. *Suppose $u \in \mathcal{C}^2(\Omega)$, $\Omega \subset \mathbb{R}^2$. Then, for $0 < \rho \ll h$ and $h \to 0$,*

$$u(\boldsymbol{x}) - YNF_{h,\rho}(\boldsymbol{x}) \simeq -\frac{\rho^2}{8}\left(g\left(\frac{\rho}{h}|Du|\right) u_{\eta\eta} + h\left(\frac{\rho}{h}|Du|\right) u_{\xi\xi}\right),$$

where

$$u_{\eta\eta} = D^2u\left(\frac{Du}{|Du|}, \frac{Du}{|Du|}\right), \quad u_{\xi\xi} = D^2u\left(\frac{Du^\perp}{|Du|}, \frac{Du^\perp}{|Du|}\right)$$



(a) (b) (c) (d) (e) (f)

FIG. 2. *1D neighborhood filtering experience. We iterate the filter on the sine signal until it converges to a steady state. We show the input signal* (a) *and the final state* (b). (e) *and* (f) *display two intermediate states.* (c) *and* (d) *display the signal* $f(\frac{\rho}{h}|u_n'|)$ *which gives the magnitude and signal of the heat equation leading to* (e) *and* (f). *These two signals describe exactly the positions of the shocks in the further iterations and the final state.*

and

$$g(t) = \frac{1 - \frac{3}{2}t^2}{1 - \frac{1}{4}t^2}, \qquad h(t) = \frac{1 - \frac{1}{2}t^2}{1 - \frac{1}{4}t^2}.$$

Proof. Let $\mathbf{x} \in \Omega$ and $h, \rho \in \mathbb{R}^+$. Then

$$u(\mathbf{x}) - YNF_{h,\rho}(\mathbf{x}) = -\frac{1}{\int_{B_\rho(0)} e^{-\frac{(u(\mathbf{x}+\mathbf{t})-u(\mathbf{x}))^2}{h^2}}\, d\mathbf{t}} \int_{B_\rho(0)} (u(\mathbf{x}+\mathbf{t}) - u(\mathbf{x}))e^{-\frac{(u(\mathbf{x}+\mathbf{t})-u(\mathbf{t}))^2}{h^2}}\, d\mathbf{t}.$$

We take the Taylor expansion of $u(\mathbf{x}+\mathbf{t})$ and the exponential function e^{-y^2}. Then

we take polar coordinates and integrate, obtaining

$$u(\mathbf{x}) - YNF_{h,\rho}(\mathbf{x}) \simeq \frac{1}{\pi\rho^2 - \frac{\rho^4\pi}{4h^2}(u_x^2+u_y^2)}\left(\frac{\pi\rho^4}{8}\Delta u - \frac{\pi\rho^6}{16h^2}\left(u_x^2u_{xx}+u_y^2u_{xx}\right.\right.$$
$$\left.\left. + u_x^2u_{yy}+u_x^2u_{xx}\right) - \frac{\pi\rho^6}{8h^2}(u_x^2u_{xx}+2u_xu_yu_{xy}+u_y^2u_{yy})\right)$$

for ρ small enough. By grouping the terms of above expression, we get the desired result. □

The neighborhood filtering method noise can be written as the sum of a diffusion term in the tangent direction $u_{\xi\xi}$, plus a diffusion term in the normal direction, $u_{\eta\eta}$. The sign and the magnitude of both diffusions depend on the sign and the magnitude of the functions g and h. Both functions can take positive and negative values. Therefore, both diffusions can appear as a directional heat equation or directional reverse heat equation, depending on the value of the gradient. As in the 1D case, the algorithm performs like a filtering/enhancing algorithm, depending on the value of the gradient. If $B_1 = \sqrt{2}/\sqrt{3}$ and $B2 = \sqrt{2}$, respectively, denote the zeros of the functions g and h, we can distinguish the following cases:

- When $0 < |Du| < B_2\frac{h}{\rho}$ the algorithm behaves like the Perona–Malik filter [27]. In a first step, a heat equation is applied, but when $|Du| > B_1\frac{h}{\rho}$ the normal diffusion turns into a reverse diffusion, enhancing the edges, while the tangent diffusion stays positive.
- When $|Du| > B_2\frac{h}{\rho}$ the algorithm differs from the Perona–Malik filter. A heat equation or a reverse heat equation is applied, depending on the value of the gradient. The change of behavior between these two dynamics is marked by an asymptotical discontinuity leading to an amplified shock effect.

FIG. 3. *Denoising experience on a natural image. From left to right and from top to bottom: noisy image (standard deviation* 20*), Gaussian convolution, anisotropic filter, total variation minimization, Tadmor–Nezzar–Vese iterated total variation, Osher et al. iterated total variation, and the Yaroslavsky neighborhood filter.*

3. Frequency domain filters. Let u be the original image defined on the grid I. The image is supposed to be modified by the addition of a signal independent white noise N. N is a random process where $N(i)$ are i.i.d. with zero mean and have constant

variance σ^2. The resulting noisy process depends on the random noise component, and therefore it is modeled as a random field V,

$$V(i) = u(i) + N(i). \tag{3.1}$$

Given a noise observation $n(i)$, $v(i)$ denotes the observed noisy image,

$$v(i) = u(i) + n(i). \tag{3.2}$$

Let $\mathcal{B} = \{g_\alpha\}_{\alpha \in A}$ be an orthogonal basis of $\mathbb{R}^{|I|}$. The noisy process is transformed as

$$V_{\mathcal{B}}(\alpha) = u_{\mathcal{B}}(\alpha) + N_{\mathcal{B}}(\alpha), \tag{3.3}$$

where

$$V_{\mathcal{B}}(\alpha) = \langle V, g_\alpha \rangle, \quad u_{\mathcal{B}}(\alpha) = \langle u, g_\alpha \rangle, \quad N_{\mathcal{B}}(\alpha) = \langle N, g_\alpha \rangle$$

are the scalar products of V, u, and N with $g_\alpha \in \mathcal{B}$. The noise coefficients $N_{\mathcal{B}}(\alpha)$ remain uncorrelated and with zero mean, but the variances are multiplied by $\|g_\alpha\|^2$:

$$\begin{aligned} E[N_{\mathcal{B}}(\alpha) N_{\mathcal{B}}(\beta)] &= \sum_{m,n \in I} g_\alpha(m) g_\beta(n) E[N(m)N(n)] \\ &= \langle g_\alpha, g_\beta \rangle \sigma^2 = \sigma^2 \|g_\alpha\|^2 \delta[\alpha - \beta]. \end{aligned}$$

Frequency domain filters are applied independently to every transform coefficient $V_{\mathcal{B}}(\alpha)$, and then the solution is estimated by the inverse transform of the new coefficients. Noisy coefficients $V_{\mathcal{B}}(\alpha)$ are modified to $a(\alpha)V_{\mathcal{B}}(\alpha)$. This is a nonlinear algorithm because $a(\alpha)$ depends on the value $V_{\mathcal{B}}(\alpha)$. The inverse transform yields the estimate

$$\hat{U} = DV = \sum_{\alpha \in A} a(\alpha)\ V_{\mathcal{B}}(\alpha)\ g_\alpha. \tag{3.4}$$

D is also called a *diagonal operator*. Let us look for the frequency domain filter D which minimizes a certain estimation error. This error is based on the square Euclidean distance, and it is averaged over the noise distribution.

DEFINITION 3.1. *Let u be the original image, N be a white noise, and $V = u + N$. Let D be a frequency domain filter. Define the* risk of D *as*

$$r(D, u) = E\{\|u - DV\|^2\}, \tag{3.5}$$

where the expectation is taken over the noise distribution.

The following theorem, which is easily proved, gives the diagonal operator D_{inf} that minimizes the risk,

$$D_{inf} = \arg\min_D r(D, u).$$

THEOREM 3.2. *The operator D_{inf} which minimizes the risk is given by the family $\{a(\alpha)\}_\alpha$, where*

$$a(\alpha) = \frac{|u_{\mathcal{B}}(\alpha)|^2}{|u_{\mathcal{B}}(\alpha)|^2 + \|g_\alpha\|^2 \sigma^2}, \tag{3.6}$$

and the corresponding risk is

$$r_{inf}(u) = \sum_{s\in S} \|g_\alpha\|^4 \frac{|u_{\mathcal{B}}(\alpha)|^2\sigma^2}{|u_{\mathcal{B}}(\alpha)|^2 + \|g_\alpha\|^2\sigma^2}. \tag{3.7}$$

The previous optimal operator attenuates all noisy coefficients in order to minimize the risk. If one restricts $a(\alpha)$ to be 0 or 1, one gets a projection operator. In that case, a subset of coefficients is kept, and the rest gets canceled. The projection operator that minimizes the risk $r(D,u)$ is obtained by the family $\{a(\alpha)\}_\alpha$, where

$$a(\alpha) = \begin{cases} 1 & |u_{\mathcal{B}}(\alpha)|^2 \geq \|g_\alpha\|^2\sigma^2, \\ 0 & \text{otherwise} \end{cases}$$

and the corresponding risk is

$$r_p(u) = \sum \|g_\alpha\|^2 \min(|u_{\mathcal{B}}(\alpha)|^2, \|g_\alpha\|^2\sigma^2).$$

Note that both filters are ideal operators because they depend on the coefficients $u_{\mathcal{B}}(\alpha)$ of the original image, which are not known. We call, as classical, *Fourier–Wiener filter* the optimal operator (3.6) where $\mathcal{B}$ is a Fourier basis. This is an ideal filter, since it uses the (unknown) Fourier transform of the original image. By the use of the Fourier basis (see Figure 4), global image characteristics may prevail over local ones and create spurious periodic patterns. To avoid this effect, the basis must take into account more local features, as the wavelet and local DCT transforms do. The search for the ideal basis associated with each image is still open. At the moment, the way seems to be a dictionary of basis instead of one single basis [19].

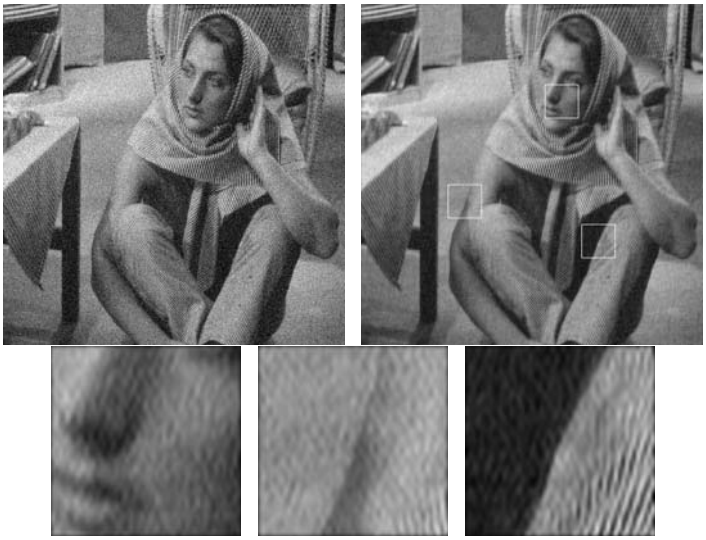


FIG. 4. *Fourier–Wiener filter experiment. Top left: Degraded image by an additive white noise of* $\sigma = 15$. *Top right: Fourier–Wiener filter solution. Bottom: Zoom on three different zones of the solution. The image is filtered as a whole, and therefore a uniform texture is spread all over the image.*

3.1. Local adaptive filters in transform domain. The local adaptive filters have been introduced by Yaroslavsky and Eden [41] and Yaroslavsky [42]. In this case, the noisy image is analyzed in a moving window, and in each position of the window its spectrum is computed and modified. Finally, an inverse transform is used to estimate only the signal value in the central pixel of the window.

Let $i \in I$ be a pixel and $W = W(i)$ a window centered in i. Then the DCT transform of W is computed and modified. The original image coefficients of W, $u_{\mathcal{B},W}(\alpha)$, are estimated, and the optimal attenuation of Theorem 3.2 is applied. Finally, only the center pixel of the restored window is used. This method is called the *empirical Wiener filter*. In order to approximate $u_{\mathcal{B},W}(\alpha)$, one can take averages on the additive noise model, that is,

$$E|V_{\mathcal{B},W}(\alpha)|^2 = |u_{\mathcal{B},W}(\alpha)|^2 + \sigma^2\|g_\alpha\|^2.$$

Denoting by $\mu = \sigma\|g_\alpha\|$, the unknown original coefficients can be written as

$$|u_{\mathcal{B},W}(\alpha)|^2 = E|V_{\mathcal{B},W}(\alpha)|^2 - \mu^2.$$

The observed coefficients $|v_{\mathcal{B},W}(\alpha)|^2$ are used to approximate $E|V_{\mathcal{B},W}(\alpha)|^2$, and the estimated original coefficients are replaced in the optimal attenuation, leading to the family $\{a(\alpha)\}_\alpha$, where

$$a(\alpha) = \max\left\{0, \frac{|v_{\mathcal{B},W}(\alpha)|^2 - \mu^2}{|v_{\mathcal{B},W}(\alpha)|^2}\right\}.$$

Denote by $EWF_\mu(i)$ the filter given by the previous family of coefficients. The method noise of the $EWF_\mu(i)$ is easily computed, as proved in the following theorem.

THEOREM 3.3. *Let u be an image defined in a grid I, and let $i \in I$ be a pixel. Let $W = W(i)$ be a window centered in the pixel i. Then the method noise of the $EWF_\mu(i)$ is given by*

$$u(i) - EWF_\mu(i) = \sum_{\alpha\in\Lambda} v_{\mathcal{B},\mathcal{W}}(\alpha)\ g_\alpha(i) + \sum_{\alpha\notin\Lambda} \frac{\mu^2}{|v_{\mathcal{B},\mathcal{W}}(\alpha)|^2} v_{\mathcal{B},\mathcal{W}}(\alpha)\ g_\alpha(i),$$

where $\Lambda = \{\alpha \mid\ |v_{\mathcal{B},\mathcal{W}}(\alpha)| < \mu\}$.

The presence of an edge in the window W will produce a great number of large coefficients, and, as a consequence, the cancelation of these coefficients will produce oscillations. Then spurious cosines will also appear in the image under the form of chessboard patterns; see Figure 5.

3.2. Wavelet thresholding. Let $\mathcal{B} = \{g_\alpha\}_{\alpha\in A}$ be an orthonormal basis of wavelets [20]. Let us discuss two procedures modifying the noisy coefficients, called *wavelet thresholding methods* (Donoho and Johnstone [10]). The first procedure is a projection operator which approximates the ideal projection (3.6). It is called a *hard thresholding* and cancels coefficients smaller than a certain threshold μ,

$$a(\alpha) = \begin{cases} 1 & |v_{\mathcal{B}}(\alpha)| > \mu, \\ 0 & \text{otherwise.} \end{cases}$$

Let us denote this operator by $HWT_\mu(v)$. This procedure is based on the idea that the image is represented with large wavelet coefficients, which are kept, whereas the noise

is distributed across small coefficients, which are canceled. The performance of the method depends on the capacity of approximating u by a small set of large coefficients. Wavelets are, for example, an adapted representation for smooth functions.

THEOREM 3.4. *Let u be an image defined in a grid I. The method noise of a hard thresholding $HWT_\mu(u)$ is*

$$u - HWT_\mu(u) = \sum_{\{\alpha || u_{\mathcal{B}}(\alpha)| < \mu\}} u_{\mathcal{B}}(\alpha) g_\alpha.$$

Unfortunately, edges lead to a great amount of wavelet coefficients lower than the threshold but not zero. The cancelation of these wavelet coefficients causes small oscillations near the edges, i.e., a Gibbs-like phenomenon. Spurious wavelets can also be seen in the restored image due to the cancelation of small coefficients; see Figure 5. This artifact will be called *wavelet outliers*, as it is introduced in [11]. Donoho [9] showed that these effects can be partially avoided with the use of a soft thresholding,

$$a(\alpha) = \begin{cases} \frac{v_{\mathcal{B}}(\alpha) - sgn(v_{\mathcal{B}}(\alpha))\mu}{v_{\mathcal{B}}(\alpha)}, & |v_{\mathcal{B}}(\alpha)| \geq \mu, \\ 0 & \text{otherwise}, \end{cases}$$

which will be denoted by $SWT\mu(v)$. The continuity of the soft thresholding operator better preserves the structure of the wavelet coefficients, reducing the oscillations near discontinuities. Note that a soft thresholding attenuates all coefficients in order to reduce the noise, as an ideal operator does. As we shall see at the end of this paper, the L^2 norm of the method noise is lessened when replacing the hard threshold by a soft threshold. See Figures 5 and 15 for a comparison of both method noises.

THEOREM 3.5. *Let u be an image defined in a grid I. The method noise of a soft thresholding $SWT_\mu(u)$ is*

$$u - SWT_\mu(u) = \sum_{\{\alpha || u_{\mathcal{B}}(\alpha)| < \mu\}} u_{\mathcal{B}}(\alpha) g_\alpha + \mu \sum_{\{\alpha || u_{\mathcal{B}}(\alpha)| > \mu\}} sgn(u_{\mathcal{B}}(\alpha))\ g_\alpha.$$

A simple example can show how to fix the threshold μ. Suppose the original image u is zero; then $v_{\mathcal{B}}(\alpha) = n_{\mathcal{B}}(\alpha)$, and therefore the threshold μ must be taken over the maximum of noise coefficients to ensure their suppression and the recovery of the original image. It can be shown that the maximum amplitude of a white noise has a high probability of being smaller than $\sigma\sqrt{2\log|I|}$. It can be proved that the risk of a wavelet thresholding with the threshold $\mu = \sigma\sqrt{2\log|I|}$ is near the risk r_p of the optimal projection; see [10, 20].

THEOREM 3.6. *The risk $r_t(u)$ of a hard or soft thresholding with the threshold $\mu = \sigma\sqrt{2\log|I|}$ is such that for all $|I| \geq 4$*

$$r_t(u) \leq (2\log|I| + 1)(\sigma^2 + r_p(u)). \tag{3.8}$$

The factor $2\log|I|$ is optimal among all the diagonal operators in $\mathcal{B}$, that is,

$$\lim_{|I| \to \infty} \inf_{D \in D_{\mathcal{B}}} \sup_{u \in \mathbb{R}^{|I|}} \frac{E\{\|u - DV\|^2\}}{\sigma^2 + r_p(u)} \frac{1}{2\log|I|} = 1. \tag{3.9}$$

In practice the optimal threshold μ is very high and cancels too many coefficients not produced by the noise. A threshold lower than the optimal is used in the experiments and produces much better results; see Figure 5. For a hard thresholding the threshold is fixed to $3 * \sigma$. For a soft thresholding this threshold still is too high; it is better fixed at $\frac{3}{2}\sigma$.

3.3. Translation invariant wavelet thresholding. Coifman and Donoho [8] improved the wavelet thresholding methods by averaging the estimation of all translations of the degraded signal. Calling $v^p(i)$ the translated signal $v(i-p)$, the wavelet coefficients of the original and translated signals can be very different, and they are not related by a simple translation or permutation,

$$v^p_{\mathcal{B}}(\alpha) = \langle v(n-p), g_\alpha(n)\rangle = \langle v(n), g_\alpha(n+p)\rangle.$$

The vectors $g_\alpha(n+p)$ are not in general in the basis $\mathcal{B} = \{g_\alpha\}_{\alpha\in A}$, and therefore the estimation of the translated signal is not related to the estimation of v. This new algorithm yields an estimate $\hat{u}^p$ for every translated v^p of the original image,

$$\hat{u}^p = Dv^p = \sum_{\alpha\in A} a(\alpha) v^p_{\mathcal{B}}(\alpha) g_\alpha. \tag{3.10}$$

The translation invariant thresholding is obtained by averaging all these estimators after a translation in the inverse sense,

$$\frac{1}{|I|}\sum_{p\in I} \hat{u}^p(i+p), \tag{3.11}$$

and will be denoted by $TIWT(v)$.

The Gibbs effect is considerably reduced by the translation invariant wavelet thresholding (see Figure 5), because the average of different estimations of the image reduces the oscillations. This is therefore the version we shall use in the comparison section. Recently, Durand and Nikolova [11] have actually proposed an efficient variational method finding the best compromise to avoid the three common artifacts in total variation methods and wavelet thresholding, namely the staircasing, the Gibbs effect, and the wavelet outliers. Unfortunately, we could not draw the method into the comparison.

FIG. 5. *Denoising experiment on a natural image. From left to right and from top to bottom: noisy image (standard deviation* 20*), Fourier–Wiener filter (ideal filter), the DCT empirical Wiener filter, the wavelet hard thresholding, the soft wavelet thresholding, and the translation invariant wavelet hard thresholding.*

4. Statistical neighborhood approaches. The methods we are going to consider are very recent attempts to take advantage of an image model learned from the image itself. More specifically, these denoising methods attempt to learn the statistical relationship between the image values in a window around a pixel and the pixel value at the window center.

4.1. DUDE, a universal denoiser. The recent work by Weissman et al. [38] has led to the proposition of a "universal denoiser" for digital images. The authors assume that the noise model is fully known, namely the probability transition matrix $\Pi(a,b)$, where $a,b \in A$, the finite alphabet of all possible values for the image. In order to fix ideas, we shall assume as in the rest of this paper that the noise is additive Gaussian, in which case one simply has $\Pi(a,b) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(a-b)2}{2\sigma^2}}$ for the probability of observing b when the real value was a. The authors also fix an error cost $\Lambda(a,b)$ which, to fix ideas, we can take to be a quadratic function $\Lambda(a,b) = (a-b)^2$, namely, the cost of mistaking a for b.

The authors fix a neighborhood shape, say, a square discrete window deprived of its center i, $\tilde{N}_i = N_i \setminus \{i\}$ around each pixel i. Then the question is, Once the image has been observed in the window $\tilde{N}_i$, what is the best estimate we can make from the observation of the full image?

The authors propose the following algorithm:

- Compute, for each possible value b of $u(i)$, the number of windows N_j in the image such that the restrictions of u to $\tilde{N}_j$ and $\tilde{N}_i$ coincide and the observed value at the pixel j is b. This number is called $m(b, N_i)$, and the line vector $(m(b,N_i))_{b\in A}$ is denoted by $m(N_i)$.
- Then compute the denoised value of u at i as

$$\tilde{u}(i) = \arg\min_{b\in A} m(N_i)\Pi^{-1}(\Lambda_b \otimes \Pi_{u(i)}),$$

  where $w \otimes v = (w(b)v(b))$ denotes the vector obtained by multiplying each component of u by each component of v, $u(i)$ is the observed value at i, and we denote by X_a the a-column of a matrix X.

The authors prove that this denoiser is universal in the sense "of asymptotically achieving, without access to any information on the statistics of the clean signal, the same performance as the best denoiser that does have access to this information." In [24] the authors present an implementation valid for binary images with an impulse noise, with excellent results. The reason of these limitations in implementation are clear: First, the matrix Π is of very low dimension and invertible for impulse noise. If instead we consider as above a Gaussian noise, then the application of Π^{-1} amounts to deconvolving a signal by a Gaussian, which is a rather ill-conditioned method. All the same, it is doable, while the computation of m certainly is not for a large alphabet, such as the one involved in grey-tone images (256 values). Even supposing that the learning window N_i has the minimal possible size of 9, the number of possible such windows is about 256^9, which turns out to be much larger than the number of observable windows in an image (whose typical size amounts to 10^6 pixels). Actually, the number of samples can be made significantly smaller by quantizing the grey-level image and by noting that the window samples are clustered. Anyway, the direct observation of the number $m(N_i)$ in an image is almost hopeless, particularly if it is corrupted by noise.

4.2. The UINTA algorithm. Awate and Whitaker [3] have proposed a method whose principles stand close to the NL-means algorithm, since, as in DUDE, the

method involves comparison between subwindows to estimate a restored value. The objective of the algorithm "UINTA, for unsupervised information-theoretic, adaptive filtering" is to denoise the image, decreasing the randomness of the image. The algorithm proceeds as follows:

- Assume that the $(2d+1)\times(2d+1)$ windows in the image are realizations of a random vector Z. The probability distribution function of Z is estimated from the samples in the image,

$$p(z) = \frac{1}{|A|}\sum_{z_i \in A} G_\sigma(z - z_i),$$

  where $z \in \mathbb{R}^{(2d+1)\times(2d+1)}$, G_σ is the Gaussian density function in dimension n with variance σ^2,

$$G_\sigma(\mathbf{x}) = \frac{1}{(2\pi)^{\frac{n}{2}}\sigma} e^{-\frac{\|\mathbf{x}\|^2}{\sigma^2}},$$

  and A is a random subset of windows in the image.
- Then the authors propose an iterative method which minimizes the entropy of the density distribution,

$$E_p \log p(Z) = -\int p(z)\log p(z) dz.$$

  This minimization is achieved by a gradient descent algorithm of the previous energy function.

The denoising effect of this algorithm can be understood, as it forces the probability density to concentrate. Thus, groups of similar windows tend to assume a more and more similar configuration which is less noisy. The differences of this algorithm with NL-means are patent, however. This algorithm creates a global interaction between all windows. In particular, it tends to favor big groups of similar windows and to remove small groups. To that extent, it is a global homogenization process and is quite valid if the image consists of a periodic or quasi-periodic texture, as is patent in the successful experiments shown in this paper. The spirit of this method is to define a new, information theoretically oriented *scale space*. In that sense, the gradient descent must be stopped before a steady state. The time at which the process is stopped gives us the scale of randomness of the filtered image.

5. NL-means algorithm. The local smoothing methods and the frequency domain filters aim at a noise reduction and at a reconstruction of the main geometrical configurations but not at the preservation of the fine structure, details, and texture. Due to the regularity assumptions on the original image of previous methods, details and fine structures are smoothed out because they behave in all functional aspects as noise. The NL-means algorithm we shall now discuss tries to take advantage of the high degree of redundancy of any natural image. By this, we simply mean that every small window in a natural image has many similar windows in the same image. This fact is patent for windows close by, at one pixel distance, and in that case we go back to a local regularity assumption. Now in a very general sense inspired by the neighborhood filters, one can define as "neighborhood of a pixel i" any set of pixels j in the image such that a window around j looks like a window around i. All pixels in that neighborhood can be used for predicting the value at i, as was first shown in [13] for 2D images. This first work has inspired many variants for the restoration

of various digital objects, in particular 3D surfaces [32]. The fact that such a self-similarity exists is a regularity assumption, actually more general and more accurate than all regularity assumptions we have considered in section 2. It also generalizes a periodicity assumption of the image.

Let v be the noisy image observation defined on a bounded domain $\Omega \subset \mathbb{R}^2$, and let $\mathbf{x} \in \Omega$. The NL-means algorithm estimates the value of $\mathbf{x}$ as an average of the values of all the pixels whose Gaussian neighborhood looks like the neighborhood of $\mathbf{x}$,

$$NL(v)(\mathbf{x}) = \frac{1}{C(\mathbf{x})} \int_\Omega e^{-\frac{(G_a * |v(\mathbf{x}+.)-v(\mathbf{y}+.)|^2)(0)}{h^2}} \, v(\mathbf{y}) \, d\mathbf{y},$$

where G_a is a Gaussian kernel with standard deviation a, h acts as a filtering parameter, and $C(\mathbf{x}) = \int_\Omega e^{-\frac{(G_a * |v(\mathbf{x}+.)-v(\mathbf{z}+.)|^2)(0)}{h^2}} d\mathbf{z}$ is the normalizing factor. We recall that

$$(G_a * |v(\mathbf{x}+.) - v(\mathbf{y}+.)|^2)(0) = \int_{\mathbb{R}^2} G_a(\mathbf{t}) |v(\mathbf{x}+\mathbf{t}) - v(\mathbf{y}+\mathbf{t})|^2 d\mathbf{t}.$$

Since we are considering images defined on a discrete grid I, we shall give a discrete description of the NL-means algorithm and some consistency results. This simple and generic algorithm and its application to the improvement of the performance of digital cameras are the object of an European patent application [4].

5.1. Description. Given a discrete noisy image $v = \{v(i) \mid i \in I\}$, the estimated value $NL(v)(i)$ is computed as a weighted average of all the pixels in the image,

$$NL(v)(i) = \sum_{j \in I} w(i,j) v(j),$$

where the weights $\{w(i,j)\}_j$ depend on the similarity between the pixels i and j and satisfy the usual conditions $0 \le w(i,j) \le 1$ and $\sum_j w(i,j) = 1$.

In order to compute the similarity between the image pixels, we define a *neighborhood system* on I.

DEFINITION 5.1 (neighborhoods). *A* neighborhood system *on I is a family $\mathcal{N} = \{\mathcal{N}_i\}_{i \in I}$ of subsets of I such that for all $i \in I$,*

(i) $i \in \mathcal{N}_i$,

(ii) $j \in \mathcal{N}_i \Rightarrow i \in \mathcal{N}_j$.

The subset $\mathcal{N}_i$ is called the neighborhood or the similarity window of i. We set $\tilde{\mathcal{N}}_i = \mathcal{N}_i \setminus \{i\}$.

The similarity windows can have different sizes and shapes to better adapt to the image. For simplicity we will use square windows of fixed size. The restriction of v to a neighborhood $\mathcal{N}_i$ will be denoted by $v(\mathcal{N}_i)$:

$$v(\mathcal{N}_i) = (v(j), j \in \mathcal{N}_i).$$

The similarity between two pixels i and j will depend on the similarity of the intensity grey-level vectors $v(\mathcal{N}_i)$ and $v(\mathcal{N}_j)$. The pixels with a similar grey-level neighborhood to $v(\mathcal{N}_i)$ will have larger weights on the average; see Figure 6.

In order to compute the similarity of the intensity grey-level vectors $v(\mathcal{N}_i)$ and $v(\mathcal{N}_j)$, one can compute a Gaussian weighted Euclidean distance, $\|v(\mathcal{N}_i) - v(\mathcal{N}_j)\|^2_{2,a}$. Efros and Leung [13] showed that the L^2 distance is a reliable measure for the comparison of image windows in a texture patch. Now this measure is so much more adapted

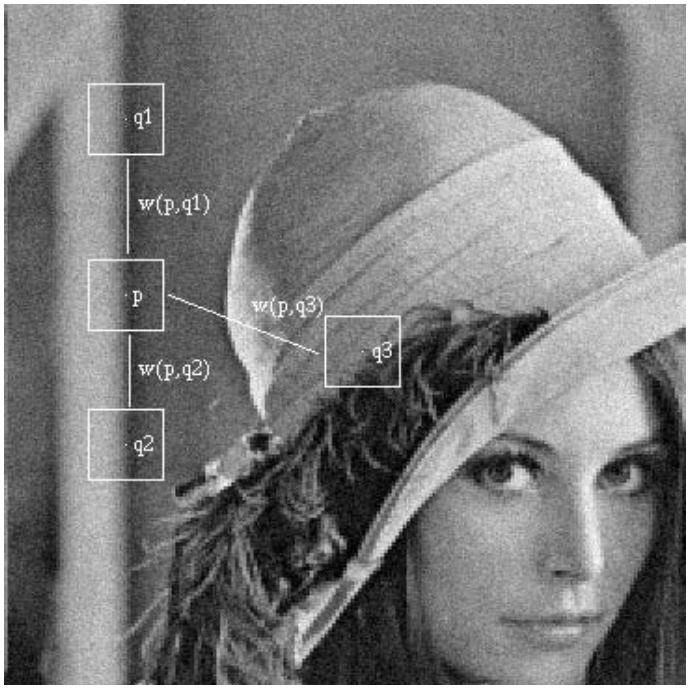


FIG. 6. *$q1$ and $q2$ have a large weight because their similarity windows are similar to that of p. On the other side the weight $w(p,q3)$ is much smaller because the intensity grey values in the similarity windows are very different.*

to any additive white noise such that a noise alters the distance between windows in a uniform way. Indeed,

$$E\|v(\mathcal{N}_i) - v(\mathcal{N}_j)\|_{2,a}^2 = \|u(\mathcal{N}_i) - u(\mathcal{N}_j)\|_{2,a}^2 + 2\sigma^2,$$

where u and v are, respectively, the original and noisy images and σ^2 is the noise variance. This equality shows that, in expectation, the Euclidean distance preserves the order of similarity between pixels. So the most similar pixels to i in v also are expected to be the most similar pixels to i in u. The weights associated with the quadratic distances are defined by

$$w(i,j) = \frac{1}{Z(i)} e^{-\frac{\|v(\mathcal{N}_i) - v(\mathcal{N}_j)\|_{2,a}^2}{h^2}},$$

where $Z(i)$ is the normalizing factor $Z(i) = \sum_j e^{-\frac{\|v(\mathcal{N}_i) - v(\mathcal{N}_j)\|_{2,a}^2}{h^2}}$ and the parameter h controls the decay of the exponential function, and therefore the decay of the weights, as a function of the Euclidean distances.

5.2. A consistency theorem for NL-means. The NL-means algorithm is intuitively consistent under stationarity conditions, saying that one can find many samples of every image detail. In fact, we shall be assuming that the image is a fairly general stationary random process. Under these assumptions, for every pixel i, the NL-means algorithm converges to the conditional expectation of i knowing its neighborhood. In the case of an additive or multiplicative white noise model, this expectation is in fact the solution to a minimization problem.

Let X and Y denote two random vectors with values on $\mathbb{R}^p$ and $\mathbb{R}$, respectively. Let f_X, f_Y denote the probability distribution functions of X, Y, and let f_{XY} denote the joint probability distribution function of X and Y. Let us recall briefly the definition of the conditional expectation.

DEFINITION 5.2.

(i) *Define the probability distribution function of Y conditioned to X as*

$$f(y \mid x) = \begin{cases} \frac{f_{XY}(x,y)}{f_X(x)} & \text{if } f_X(x) > 0, \\ 0 & \text{otherwise} \end{cases}$$

for all $x \in \mathbb{R}^p$ and $y \in \mathbb{R}$.

(ii) *Define the* conditional expectation *of Y given $\{X = x\}$ as the expectation with respect to the conditional distribution $f(y \mid x)$,*

$$E[Y \mid X = x] = \int y\, f(y \mid x)\, dy,$$

for all $x \in \mathbb{R}^p$.

The conditional expectation is a function of X, and therefore a new random variable $g(X)$, which is denoted by $E[Y \mid X]$.

Now let V be a random field and $\mathcal{N}$ a neighborhood system on I. Let Z denote the sequence of random variables $Z_i = \{Y_i, X_i\}_{i \in I}$, where $Y_i = V(i)$ is real valued and $X_i = V(\tilde{\mathcal{N}}_i)$ is $\mathbb{R}^p$ valued. Recall that $\tilde{\mathcal{N}}_i = \mathcal{N}_i \setminus \{i\}$.

Let us restrict Z to the n first elements $\{Y_i, X_i\}_{i=1}^n$. Let us define the function $r_n(x)$,

$$r_n(x) = R_n(x)/\hat{f}_n(x), \tag{5.1}$$

where

$$\hat{f}_n(x) = \frac{1}{nh^p} \sum_{i=1}^n K\left(\frac{X_i - x}{h}\right), \qquad R_n(x) = \frac{1}{nh^p} \sum_{i=1}^n \phi(Y_i) K\left(\frac{X_i - x}{h}\right), \tag{5.2}$$

ϕ is an integrable real valued function, K is a nonnegative kernel, and $x \in \mathbb{R}^p$.

Let X and Y be distributed as X_1 and Y_1. Under this form the NL-means algorithm can be seen as an instance for the exponential operator of the Nadaraya–Watson estimator [23, 37]. This is an estimator of the conditional expectation $r(x) = E[\phi(Y) \mid X = x]$. Some definitions are needed for the statement of the main result.

DEFINITION 5.3. *A stochastic process $\{Z_t \mid t = 1, 2, \ldots\}$, with Z_t defined on some probability space $(\Omega, \mathcal{A}, \mathcal{P})$, is said to be (strict-sense) stationary if for any finite partition $\{t_1, t_2, \ldots, t_n\}$ the joint distributions $F_{t_1,t_2,\ldots,t_n}(x_1, x_2, \ldots, x_n)$ are the same as the joint distributions $F_{t_1+\tau,t_2+\tau,\ldots,t_n+\tau}(x_1, x_2, \ldots, x_n)$ for any $\tau \in \mathbb{N}$.*

In the case of images, this stationary condition amounts to saying that as the size of the image grows, we are able to find in the image many similar patches for all the details of the image. This is a crucial point in understanding the performance of the NL-means algorithm. The following mixing definition is a rather technical condition. In the case of images, it amounts to saying that regions become more independent as their distance increases. This is intuitively true for natural images.

DEFINITION 5.4. *Let Z be a stochastic and stationary process $\{Z_t \mid t = 1, 2, \ldots, n\}$, and, for $m < n$, let $\mathbb{F}_m^n$ be the σ-field induced in Ω by the r.v.'s Z_j, $m \le j \le n$. Then the sequence Z is said to be β-mixing if for every $A \in \mathbb{F}_1^k$ and every $B \in \mathbb{F}_{k+n}^\infty$*

$$|P(A \cap B) - P(A)P(B)| \le \beta(n), \quad \text{with } \beta(n) \to 0, \text{ as } n \to \infty.$$

The following theorem establishes the convergence of r_n to r; see Roussas [28]. The theorem is established under the stationary and mixing hypothesis of $\{Y_i, X_i\}_{i=1}^{\infty}$ and asymptotic conditions on the decay of ϕ, $\beta(n)$, and K. This set of conditions will be denoted by H, and it is more carefully detailed in the appendix.

THEOREM 5.5 (conditional expectation theorem). *Let $Z_j = \{X_j, Y_j\}$ for $j = 1, 2, \ldots$ be a strictly stationary and mixing process. For $i \in I$, let X and Y be distributed as X_i and Y_i. Let J be a compact subset $J \subset \mathbb{R}^p$ such that*

$$\inf\{f_X(x); x \in J\} > 0.$$

Then, under hypothesis H*,*

$$\sup[\psi_n |r_n(x) - r(x)|; x \in J] \to 0 \quad a.s.,$$

where ψ_n are positive norming factors.

Let v be the observed noisy image, and let i be a pixel. Taking for ϕ the identity, we see that $r_n(v(\tilde{\mathcal{N}}_i))$ converges to $E[V(i) \mid V(\tilde{\mathcal{N}}_i) = v(\tilde{\mathcal{N}}_i)]$ under stationary and mixing conditions of the sequence $\{V(i), V(\tilde{\mathcal{N}}_i)\}_{i=1}^{\infty}$.

In the case where an additive or multiplicative white noise model is assumed, the next result shows that this conditional expectation is in fact the function of $V(\tilde{\mathcal{N}}_i)$ that minimizes the mean square error with the original field U.

THEOREM 5.6. *Let V, U, N_1, and N_2 be random fields on I such that $V = U + N_1 + g(U)N_2$, where N_1 and N_2 are independent white noises. Let $\mathcal{N}$ be a neighborhood system on I. Then we have the following:*

(i) *$E[V(i) \mid V(\tilde{\mathcal{N}}_i) = x] = E[U(i) \mid V(\tilde{\mathcal{N}}_i) = x]$ for all $i \in I$ and $x \in \mathbb{R}^p$.*

(ii) *The real value $E[U(i) \mid V(\tilde{\mathcal{N}}_i) = x]$ minimizes the mean square error*

$$\min_{g^* \in \mathbb{R}} E[(U(i) - g^*)^2 \mid V(\tilde{\mathcal{N}}_i) = x] \tag{5.3}$$

for all $i \in I$ and $x \in \mathbb{R}^p$.

(iii) *The expected random variable $E[U(i) \mid V(\tilde{\mathcal{N}}_i)]$ is the function of $V(\tilde{\mathcal{N}}_i)$ that minimizes the mean square error*

$$\min_{g} E[U(i) - g(V(\tilde{\mathcal{N}}_i))]^2. \tag{5.4}$$

Given a noisy image observation $v(i) = u(i) + n_1(i) + g(u(i))n_2(i)$, $i \in I$, where g is a real function and n_1 and n_2 are white noise realizations, the NL-means algorithm is the function of $v(\tilde{\mathcal{N}}_i)$ that minimizes the mean square error with the original image $u(i)$.

5.3. Experiments with NL-means. The NL-means algorithm chooses for each pixel a different average configuration adapted to the image. As we explained in the previous sections, for a given pixel i, we take into account the similarity between the neighborhood configuration of i and all the pixels of the image. The similarity between pixels is measured as a decreasing function of the Euclidean distance of the similarity windows. Due to the fast decay of the exponential kernel, large Euclidean distances lead to nearly zero weights, acting as an automatic threshold. The decay of the exponential function, and therefore the decay of the weights, is controlled by the parameter h. Empirical experimentation shows that one can take a similarity window of size 7×7 or 9×9 for grey-level images and 5×5 or even 3×3 in color images with little noise. These window sizes have shown to be large enough to be robust to

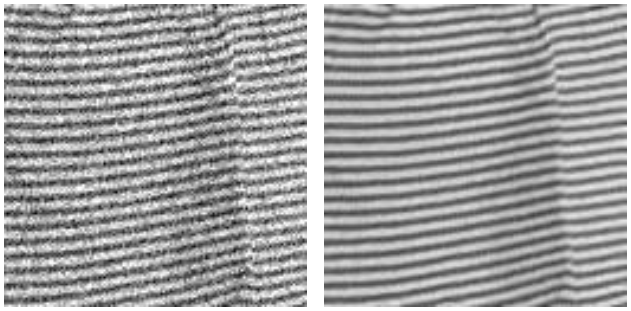


FIG. 7. *NL-means denoising experiment with a nearly periodic image. Left: Noisy image with standard deviation* 30. *Right: NL-means restored image.*

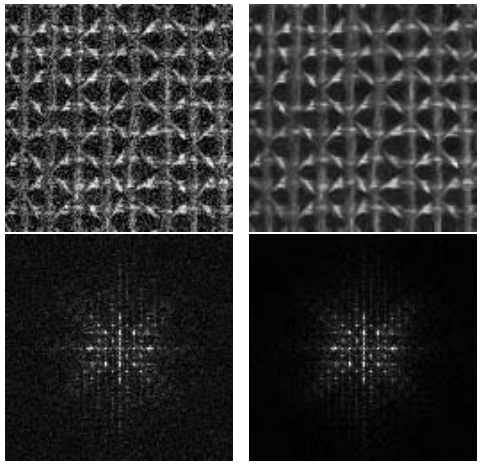


FIG. 8. *NL-means denoising experiment with a Brodatz texture image. Left: Noisy image with standard deviation* 30. *Right: NL-means restored image. The Fourier transform of the noisy and restored images show how main features are preserved even at high frequencies.*

noise and at the same time to be able to take care of the details and fine structure. Smaller windows are not robust enough to noise. Notice that in the limit case, one can take the window reduced to a single pixel i and therefore get back to the Yaroslavsky neighborhood filter. We have seen experimentally that the filtering parameter h can take values between $10 * \sigma$ and $15 * \sigma$, obtaining a high visual quality solution. In all experiments this parameter has been fixed to $12 * \sigma$. For computational aspects, in the following experiments the average is not performed in all the images. In practice, for each pixel p, we consider only a squared window centered in p and size 21×21 pixels. The computational cost of the algorithm and a fast multiscale version are addressed in section 5.5.

Due to the nature of the algorithm, the most favorable case for the NL-means algorithm is the periodic case. In this situation, for every pixel i of the image one can find a large set of samples with a very similar configuration, leading to a noise reduction and a preservation of the original image; see Figure 7 for an example.

Another case which is ideally suitable for the application of the NL-means algorithm is the textural case. Texture images have a large redundancy. For a fixed configuration many similar samples can be found in the image. In Figure 8 one can see

FIG. 9. *NL-means denoising experiment with a natural image. Left: Noisy image with standard deviation* 20. *Right: Restored image.*

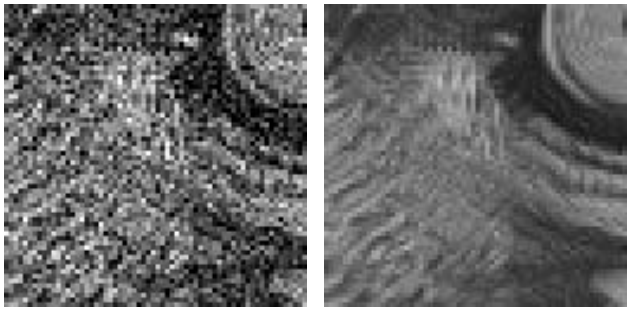


FIG. 10. *NL-means denoising experiment with a natural image. Left: Noisy image with standard deviation* 35. *Right: Restored image.*

an example with a Brodatz texture. The Fourier transform of the noisy and restored images shows the ability of the algorithm to preserve the main features even in the case of high frequencies.

The NL-means algorithm is not only able to restore periodic or texture images. Natural images also have enough redundancy to be restored. For example, in a flat zone, one can find many pixels lying in the same region and with similar configurations. In a straight or curved edge a complete line of pixels with a similar configuration is found. In addition, the redundancy of natural images allows us to find many similar configurations in far away pixels. Figures 9 and 10 show two examples on two well-known standard processing images. The same algorithm applies to the restoration of color images and films; see Figure 11.

5.4. Testing stationarity: A soft threshold optimal correction. In this section, we describe a simple and useful statistical improvement of the NL-means algorithms, with a technique similar to the wavelet thresholding. The stationarity assumption of Theorem 5.5 is not true everywhere, as each image may contain exceptional, nonrepeated structures. Such structures can be blurred out by the algorithm. The NL-means algorithm, and actually every local averaging algorithm, must involve a detection phase and special treatment of nonstationary points. The principle of such a correction is quite simple and directly derived from other thresholding methods, such as the *SWT* method.

Let us estimate the original value at a pixel i, $u(i)$, as the mean of the noisy grey levels $v(j)$ for $j \in J \subset I$. In order to reduce the noise and restore the original value,

FIG. 11. *NL-means denoising experiment with a color image. Left: Noisy image with standard deviation* 15 *in every color component. Right: Restored image.*

pixels $j \in J$ should have a nonnoisy grey level $u(j)$ similar to $u(i)$. Assuming this fact,

$$\hat{u}(i) = \frac{1}{|J|}\sum_{j\in J} v(j) \simeq \frac{1}{|J|}\sum_{j\in J} u(i) + n(j) \to u(i) \text{ as } |J| \to \infty,$$

because the average of noise values tends to zero. In addition,

$$\frac{1}{|J|}\sum_{j\in J}(v(j) - \hat{u}(i))^2 \simeq \frac{1}{|J|}\sum_{j\in J} n(j)^2 \to \sigma^2 \text{ as } |J| \to \infty.$$

If the averaged pixels have a nonnoisy grey-level value close to $u(i)$, as expected, then the variance of the average should be close to σ^2. If it is a posteriori observed that this variance is much larger than σ^2, this fact can hardly be caused only by the noise. This means that the original grey-level values of the averaged pixels were very different. At those pixels, a more conservative estimate is required, and therefore the estimated value should be averaged with the noisy one. The next result tells us how to compute this average.

THEOREM 5.7. *Let X and Y be two real random variables. Then the linear estimate $\hat{Y}$,*

$$\hat{Y} = EY + \frac{Cov(X,Y)}{VarX}(X - EY),$$

minimizes the square error

$$\min_{a,b\in\mathbb{R}} E[(Y - (a + bX))^2].$$

In our case, $X = Y + N$, where N is independent of Y, with zero mean and variance σ^2. Thus,

$$\hat{Y} = EY + \frac{VarY}{VarY + \sigma^2}(X - EY),$$

which is equal to

$$\hat{Y} = EX + \max\left(0, \frac{VarX - \sigma^2}{VarX}\right)(X - EX).$$

FIG. 12. *Optimal correction experience. Left: Noisy image. Middle: NL-means solution. Right: NL-means corrected solution. The average with the noisy image makes the solution noisier, but details and fine structure are better preserved.*

This strategy can be applied to correct any local smoothing filter. However, a good estimate of the mean and the variance at every pixel is needed. That is not the case for the local smoothing filters of section 2. This strategy can instead be satisfactorily applied to the NL-means algorithm. As we have shown in the previous section, the NL-means algorithm converges to the conditional mean. The conditional variance can also be computed by the NL-means algorithm, by taking $\phi(x) = x^2$ in Theorem 5.5, and then computing the variance as $EX^2 - (EX)^2$. In Figure 12 one can see an application of this correction.

5.5. Fast multiscale versions.

5.5.1. Plain multiscale. Let us now make some comments on the complexity of NL-means and how to accelerate it. One can estimate the complexity of an unsophisticated version as follows. If we take a similarity window of size $(2f+1)^2$, and since we can restrict the search of similar windows in a larger "search window" of size $(2s+1)^2$, the overall complexity of the algorithm is $N^2 \times (2f+1)^2 \times (2s+1)^2$, where N^2 is the number of pixels of the image. As for practical numbers, we took in all experiments $f = 3$, $s = 10$, so that the final complexity is about $49 \times 441 \times N^2$. For a 512×512 image, this takes about 30 seconds on a normal PC. It is quite desirable to expand the size of the search window as much as possible, and it is therefore useful to give a fast version. This is easily done by a multiscale strategy, with little loss in accuracy.

MULTISCALE ALGORITHM.

1. Zoom out the image u_0 by a factor 2 by a standard Shannon subsampling procedure. This yields a new image u_1. For convenience, we denote by (i, j) the pixels of u_1 and by $(2i, 2j)$ the even pixels of the original image u_0.
2. Apply NL-means to u_1, so that with each pixel (i, j) of u_1, a list of windows centered in $(i_1, j_1), \ldots, (i_k, j_k)$ is associated.
3. For each pixel of u_0, $(2i+r, 2j+s)$ with $r, s \in \{0, 1\}$, we apply the NL-means algorithm. But instead of comparing with all the windows in a searching zone, we compare only with the nine neighboring windows of each pixel $(2i_l, 2j_l)$ for $l = 1, \ldots, k$.
4. This procedure can be applied in a pyramid fashion by subsampling u_1 into u_2, and so on. In fact, it is not advisable to zoom down more than twice.

By zooming down by just a factor 2, the computation time is divided by approximately 16.

5.5.2. By blocks. Let I be the 2D grid of pixels, and let $\{i_1, \ldots, i_n\}$ be a subset of I. For each i_k, let $W_k \subset I$ be a neighborhood centered in i_k, $W_k = i_k + B_k$, where B_k gives the size and shape of the neighborhood. Let us suppose that each W_k is a

connected subset of I, such that $I = W_1 \cup W_2 \cup \cdots \cup W_n$, and where we allow the intersections between the neighborhoods to be nonempty.

Then for each W_k we define the vectorial NL-means as

$$NL(W_k) = \frac{1}{C_k} \sum_{j \in I} v(j + B_k) e^{-\frac{\|v(i_k+B_k)-v(j+B_k)\|_2^2}{h^2}},$$

where $C_k = \sum_{j \in I} e^{-\frac{\|v(i_k+B_k)-v(j+B_k)\|_2^2}{h^2}}$ and h acts as a filtering parameter. We note that $NL(W_k)$ is a vector of the same size as W_k. In contrast with the NL-means algorithm, we compute a nonweighted L^2 distance, since we restore at the same time a whole neighborhood and do not want to give privilege to any point of the neighborhood.

In order to restore the value at a pixel i, we take into account all W_k containing i, $A_i = \{k \mid i \in W_k\}$, and define

$$NL(i) = \frac{1}{|A_i|} \sum_{k \in A_i} NL(W_k)(i).$$

The overlapping of these neighborhoods permits a regular transition in the restored image and avoids block effects.

This variant by blocks of NL-means allows a better adaptation to the local image configuration of the image and, at the same time, a reduction of the complexity. In order to illustrate this reduction, let us describe the simplest implementation:

- Let $N \times N$ be the size of the image, and set $i_k = (kn, kn)$ for $k = 1, \ldots, (N - n)/n$.
- Consider the subset $B = \{i = (x_i, y_i) \mid |x_i| \leq m \text{ and } |y_i| \leq m\}$ and $W_k = i_k + B$ for all k. We take $m > n/2$ in order to have a nonempty intersection between neighboring subsets W_k.
- If we take a squared neighborhood B of size $(2m+1)^2$, and since we can restrict the search of similar windows in a larger "search window" of size $(2s + 1)^2$, the overall complexity of the algorithm is $(2m + 1)^2 \times (2s + 1)^2 \times (\frac{N-n}{n})^2$.

Taking $n = 9$ reduces the computation time of the original algorithm by more than 81.

6. Discussion and comparison.

6.1. NL-means as an extension of previous methods. As was said before, the Gaussian convolution preserves only flat zones, while contours and fine structure are removed or blurred. Anisotropic filters instead preserve straight edges, but flat zones present many artifacts. One could think of combining these two methods to improve both results. A Gaussian convolution could be applied in flat zones, while an anisotropic filter could be applied on straight edges. Still, other types of filters should be designed to specifically restore corners or curved edges and texture. The NL-means algorithm seems to provide a feasible and rational method to automatically take the best of each mentioned algorithm, reducing for every possible geometric configuration the image method noise. Although we have not computed explicitly the image method noise, Figure 13 illustrates how the NL-means algorithm chooses in each case a weight configuration corresponding to one of the previously analyzed filters. In particular, according to this set of experiments, we can consider that the consistency results given in Theorems 2.1, 2.3, and 2.5 are all valid for this algorithm.

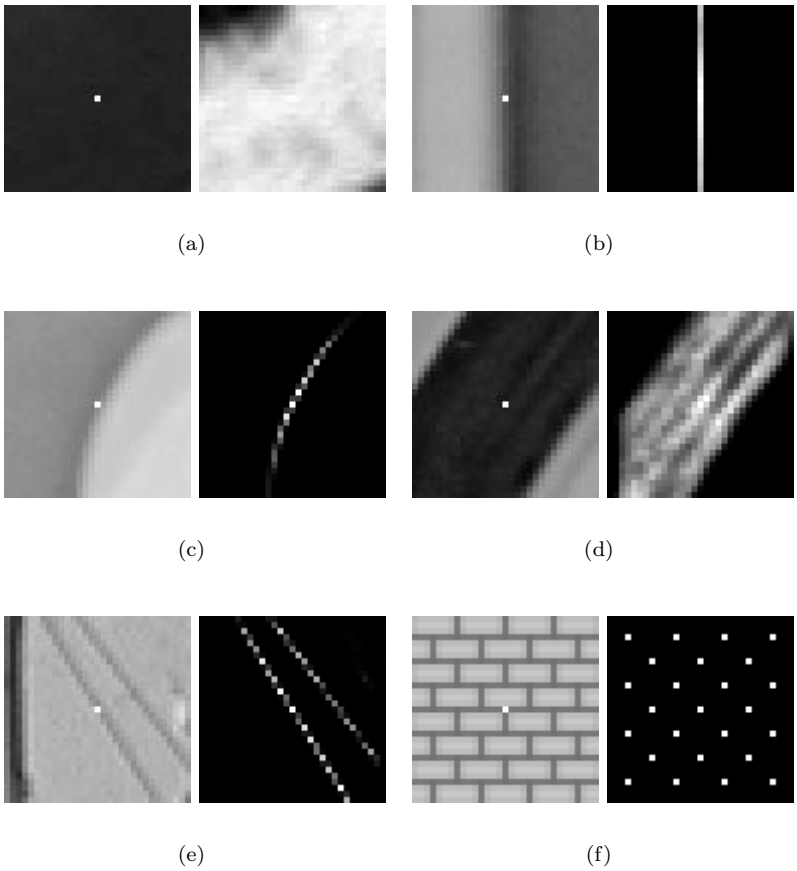


FIG. 13. *On the right-hand side of each pair, we display the weight distribution used to estimate the central pixel of the left image by the NL-means algorithm.* (a) *In flat zones, the weights are distributed as a convolution filter (as a Gaussian convolution).* (b) *In straight edges, the weights are distributed in the direction of the level line (as the mean curvature motion).* (c) *On curved edges, the weights favor pixels belonging to the same contour or level line, which is a strong improvement with respect to the mean curvature motion.* (d) *In a flat neighborhood, the weights are distributed in a grey-level neighborhood (as with a neighborhood filter). In the cases of* (e) *and* (f)*, the weights are distributed across the more similar configurations, even though they are far away from the observed pixel. This shows a behavior similar to a nonlocal neighborhood filter or to an ideal Wiener filter.*

In Figure 14 we display the probability distributions used to restore a noisy pixel. The images are the same of Figure 13. The comparison of both figures illustrates how the probability distribution is perturbed by the addition of a white noise. In this case, the probability distribution is still adapted to the local configuration of the image, and the main structures of Figure 13 are well preserved.

6.2. Comparison. In this section we shall compare the different algorithms based on three well-defined criteria: the method noise, the mean square error, and the visual quality of the restored images. Note that every criterion measures a different aspect of the denoising method. It is easy to show that only one criterion is not enough to judge the restored image, and so one expects a good solution to have a high performance under the three criteria.

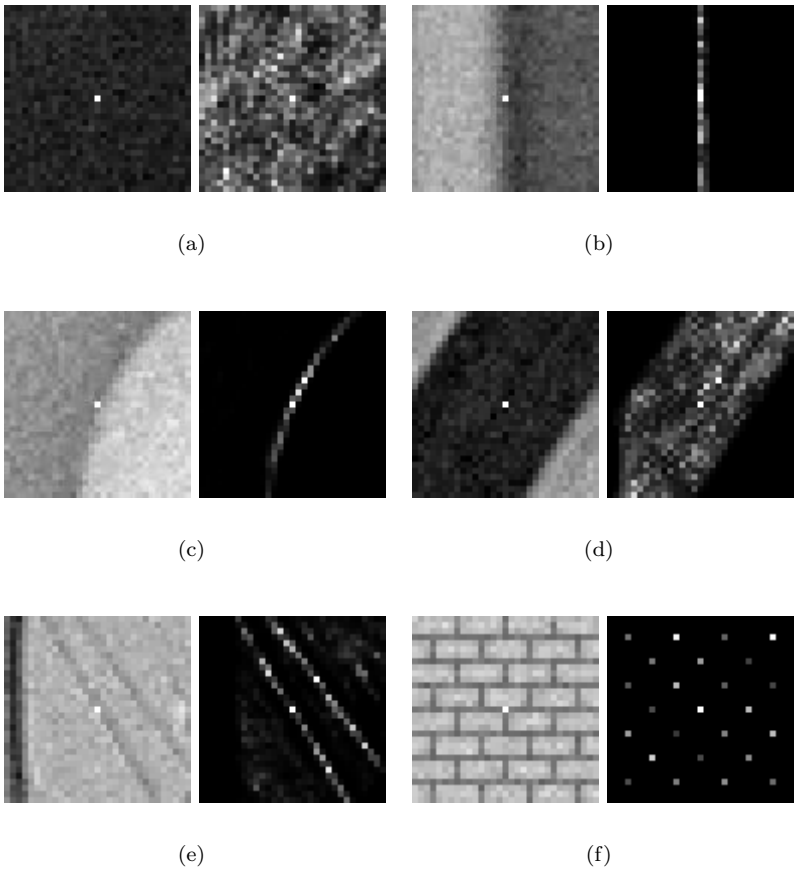


FIG. 14. *On the right-hand side of each pair, we display the weight distribution used to estimate the central pixel of the left image by the NL-means algorithm. Images are obtained by adding a Gaussian noise of standard deviation* 12.5 *to those of Figure* 13.

6.2.1. Method noise comparison. In previous sections we have defined the method noise and computed it for the different algorithms. Remember that the denoising algorithm is applied on the original (slightly noisy) image. A filtering parameter, depending mainly on the standard deviation of the noise, must be fixed for most algorithms. Let us fix $\sigma = 2.5$: we can suppose that any digital image is affected by this amount of noise since it is not visually noticeable.

The method noise tells us which geometrical features or details are preserved by the denoising process and which are eliminated. In order to preserve as many features as possible of the original image, the method noise should look as much as possible like white noise. Figures 15–17 display the method noise of the different methods for a set of standard natural images. Let us comment on them briefly.

- The Gaussian filter method noise highlights all important features of the image, such as texture, contours, and details. All these features have a large Laplacian and are therefore modified by the application of the algorithm; see Theorem 2.1.
- As announced in Theorem 2.3, the anisotropic filter method noise displays

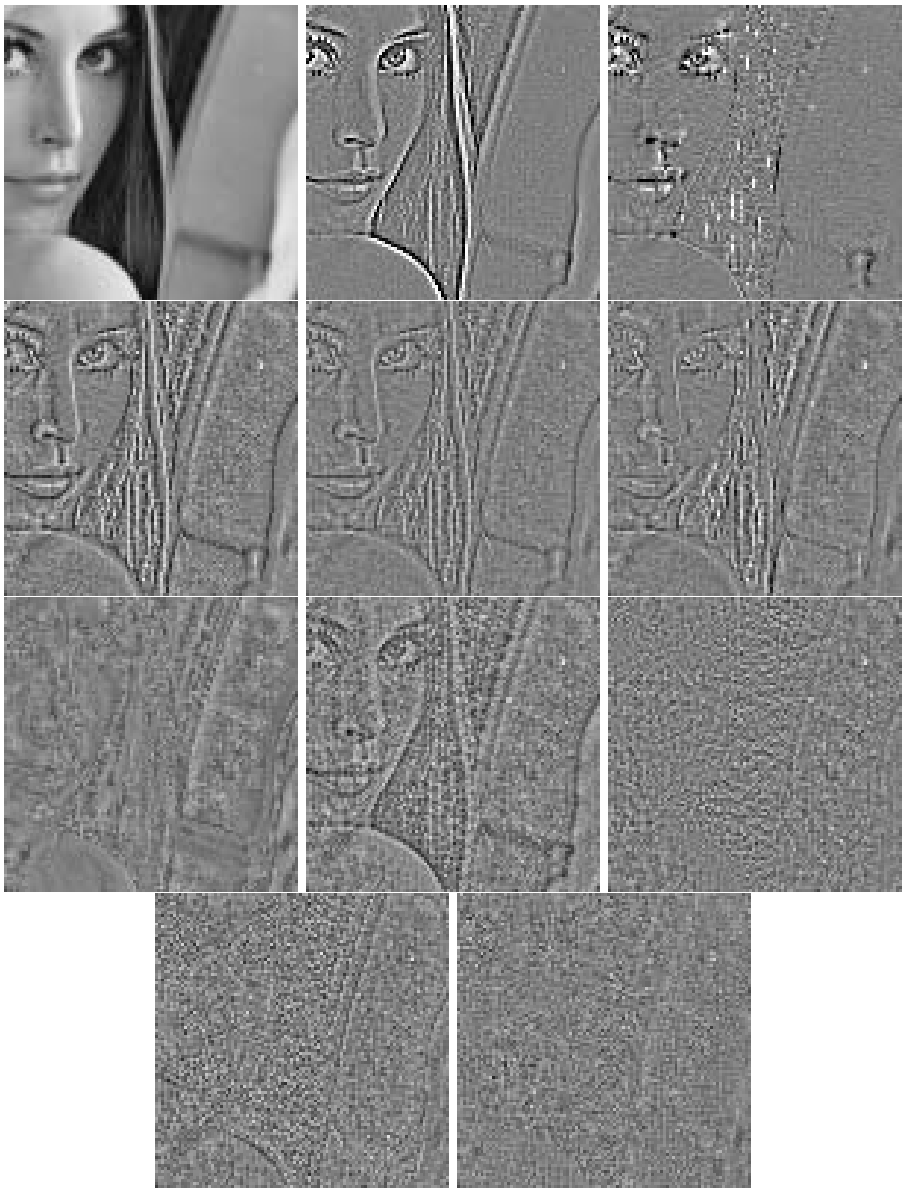


FIG. 15. *Image method noise. From left to right and from top to bottom: original image, Gaussian convolution, mean curvature motion, total variation, Tadmor–Nezzar–Vese iterated total variation, Osher et al. total variation, neighborhood filter, soft TIWT, hard TIWT, DCT empirical Wiener filter, and the NL-means algorithm.*

the corners and high frequency features. The straight edges are instead not to be seen: they have a low curvature.

- The total variation method modifies most structures and details of the image. Even straight edges are not well preserved.
- The iterated total variation refinements improve the total variation method noise. Both strategies try to reduce the geometry present in the removed noise, adding it back to the restored image, and therefore reducing the method noise.

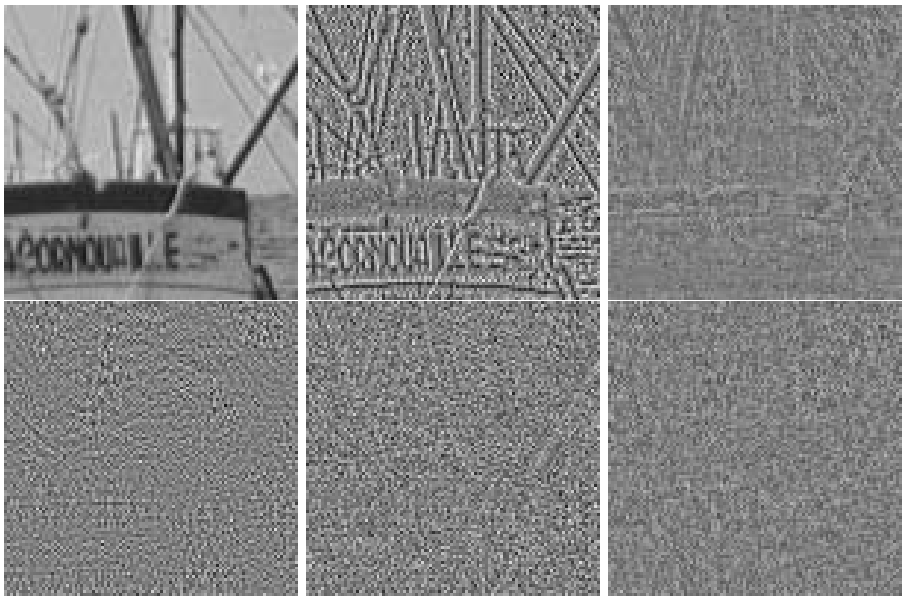


FIG. 16. *Image method noise. From left to right and from top to bottom: original image, total variation, neighborhood filter, hard TIWT, DCT empirical Wiener filter, and the NL-means algorithm.*

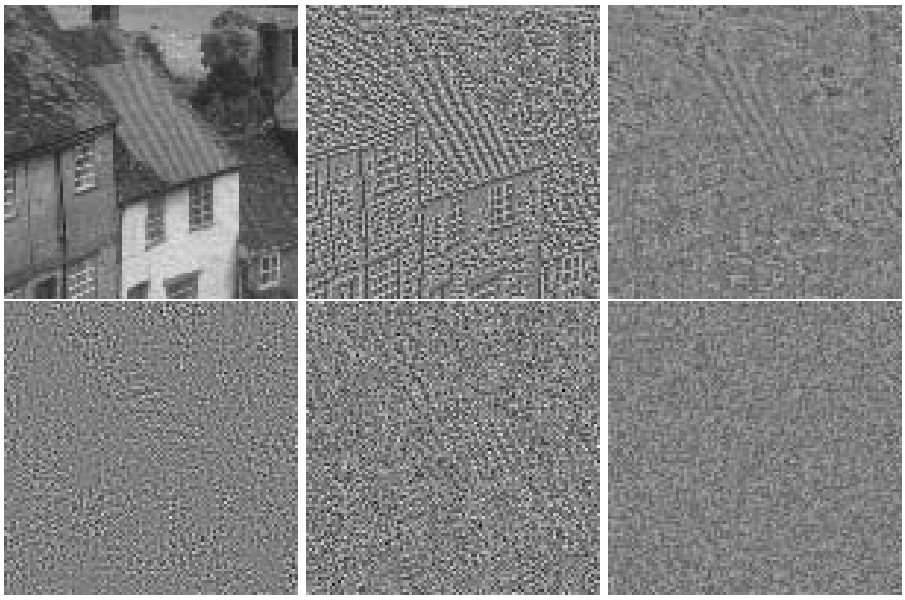


FIG. 17. *Image method noise. From left to right and from top to bottom: original image, total variation, neighborhood filter, hard TIWT, DCT empirical Wiener filter, and the NL-means algorithm.*

- The neighborhood filter preserves flat objects and contrasted edges, while edges with a low contrast are not kept. In any case, the contours, texture, and details seem to be well preserved.

- The *TIHWT* method noise is concentrated on the edges and high frequency features. These structures lead to coefficients of large enough value but lower than the threshold. They are removed by the algorithm. The average of the application to all translated versions reduces the method noise, and structures are hardly noticeable.
- The *TISWT* method noise presents much more structure than the hard thresholding. Indeed, the method noise is not only based on the small coefficients but also on an attenuation of the large ones, leading to a high alteration of the original image.
- It is difficult to find noticeable structure in the DCT empirical Wiener filter method noise. Only some contours are noticeable. In general, this filter seems to perform much better than all local smoothing filters and other frequency domain filters. Its results are similar to those of a hard stationary wavelet thresholding.
- The NL-means method noise looks more like a white noise.

6.2.2. Visual quality comparison. As commented on before, the visual quality of the restored image is another important criterion to judge the performance of a denoising algorithm. Let us present some experiments on a set of standard natural images. The objective is to compare the visual quality of the restored images, the nonpresence of artifacts, and the correct reconstruction of edges, texture, and fine structure. Figures 18–21 present these experiences comparing the visual quality of previous methods.

Figure 18 illustrates the fact that a nonlocal algorithm is needed for the correct reconstruction of periodic images. Local smoothing filters and local frequency filters are not able to reconstruct the wall pattern. Only the NL-means algorithm and the global Fourier–Wiener filter reconstruct the original texture. The Fourier–Wiener filter is based on a global Fourier transform which is able to capture the periodic structure of the image in a few coefficients. Now, in practice, this is an ideal filter because the Fourier transform of the original image is used. Figure 13(e) shows how the NL-means method chooses the correct weight configuration and explains the correct reconstruction of the wall pattern.

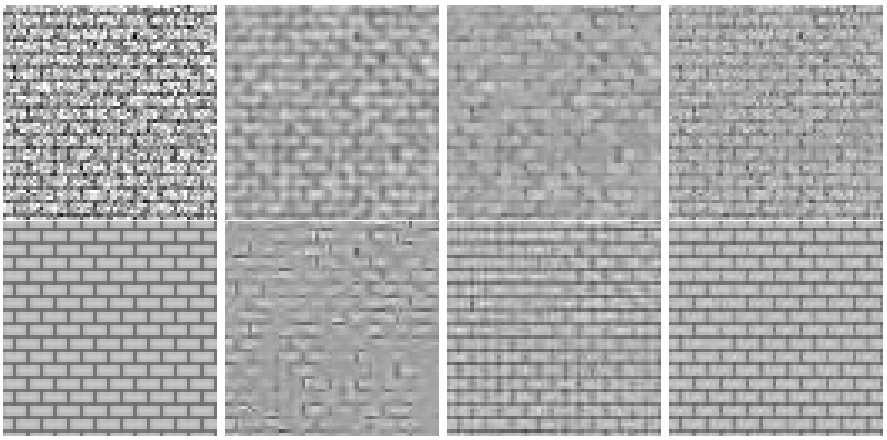


FIG. 18. *Denoising experience on a periodic image. From left to right and from top to bottom: noisy image (standard deviation* 35*), Gauss filtering, total variation, neighborhood filter, Wiener filter (ideal filter), hard TIWT, DCT empirical Wiener filtering, and the NL-means algorithm.*

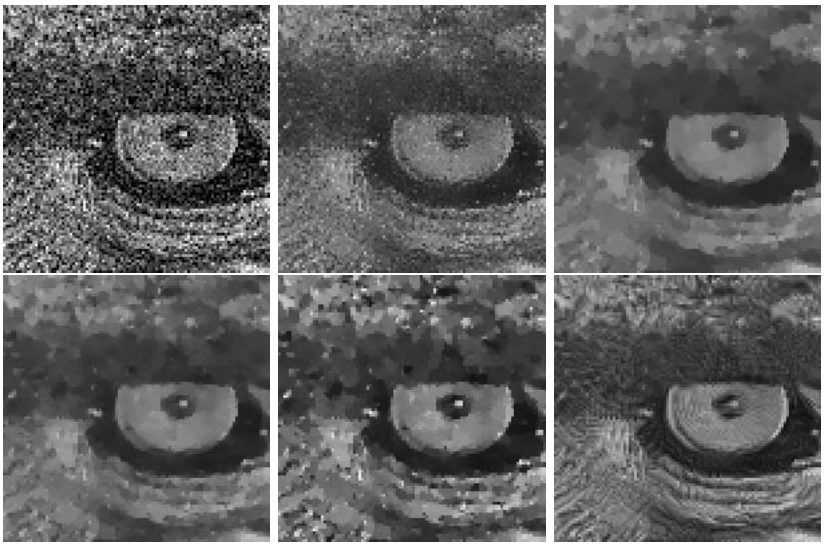


FIG. 19. *Denoising experience on a natural image. From left to right and from top to bottom: noisy image (standard deviation* 35*), neighborhood filter, total variation, Tadmor–Nezzar–Vese iterated total variation, Osher et al. iterated total variation, and the NL-means algorithm.*

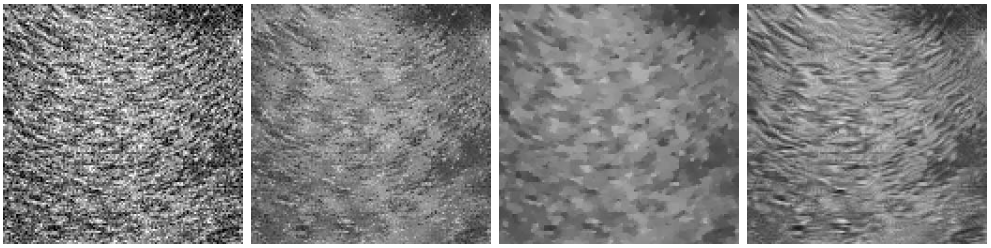


FIG. 20. *Denoising experience on a natural image. From left to right and from top to bottom: noisy image (standard deviation* 35*), neighborhood filter, total variation, and the NL-means algorithm.*

Figures 19 and 20 illustrate the difficulty of local smoothing filters for recovering stochastic patterns. The high degree of noise present in the image makes the local comparisons of the neighborhood filter noise-dependent. As a consequence, noise and texture are not well differentiated. The regularity assumption involved in the bounded variation makes it unsuitable for the restoration of textures which are filtered as noise. Iterated total variation refinements improve the total variation minimization and recover part of the excessively filtered texture.

Figure 21 shows that the frequency domain filters are well adapted to the recovery of oscillatory patterns. Although some artifacts are noticeable in both solutions, the stripes are well reconstructed. The DCT transform seems to be more adapted to this type of texture, and stripes are a little better reconstructed. For a much more detailed comparison between sliding window transform domain filtering methods and wavelet threshold methods, we refer the reader to [39]. Figure 22 shows that although the total variation minimization is not adapted to the restoration of oscillatory patterns, the iterated total variation approaches improve the restored image and reduce this drawback. The NL-means algorithm also performs well on this type of texture, due

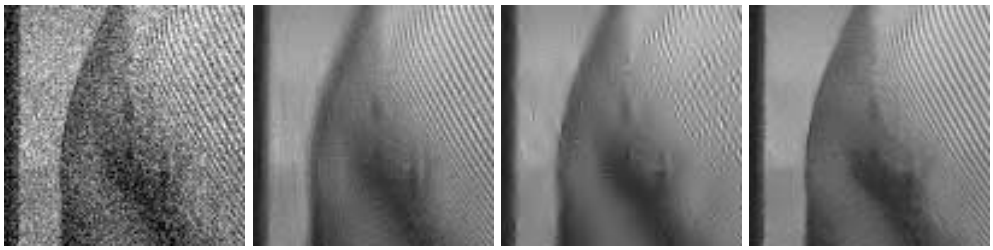


FIG. 21. *Denoising experience on a natural image. From left to right and from top to bottom: noisy image (standard deviation* 25*), the DCT empirical Wiener filter, hard TIWT, and the NL-means algorithm.*

FIG. 22. *Denoising experience on a natural image. From left to right and from top to bottom: noisy image (standard deviation* 25*), the total variation minimization, the Tadmor–Nezzar–Vese iterated total variation, and the Osher et al. iterated total variation.*

to its high degree of redundancy.

Finally, in Figures 23 and 24 we show a real denoising experiment. The NL-means algorithm is applied to a natural image taken in poor light conditions. The NL-means algorithm seems to denoise the image, keeping the main structures and details.

6.2.3. Mean square error comparison. The mean square error is the square of the Euclidean distance between the original image and its estimate. This numerical quality measurement is the more objective one, since it does not rely on any visual interpretation. Table 1 shows the mean square error of the different denoising methods with the images presented in this paper. This error table seems to corroborate the observations made for the other criteria. One sees, for example, how the frequency domain filters have a lower mean square error than the local smoothing filters. One also sees that in the presence of periodic or textural structures the empirical Wiener filter based on a DCT transform performs better than the wavelet thresholding; see also Figures 18 and 21. Note that, in the presence of periodic or stochastic patterns, the NL-means mean square error is significantly more precise than the other algorithms. Of course, the errors presented in this table cannot be computed in a real denoising problem. Let us remark that a small error does not guarantee a good visual quality of the restored image. The mean square error by itself would not be meaningful, and all previous quality criteria are also necessary to evaluate the performance of denoising methods.

Appendix. The set H of assumptions necessary for the statement of Theorem 5.5 are the following:

(H1) The sequence of random vectors $Z_i = \{Y_i, X_i\}_{i=1}^{\infty}$, where Y_i is real valued and X_i is $\mathbb{R}^p$ valued, form a strictly stationary sequence.

FIG. 23. *Denoising experience on a natural noisy image. Top: Original image, Deià village (Mallorca). Bottom: NL-means filtered image.*

(H2) The sequence $\{Z_i\}$ is β-*mixing*, and the sequence $\beta(n)$ satisfies the following summability requirement: $\beta^* = \sum_{j=1}^{\infty} \beta(n) < \infty$.

(H3) Let $\alpha = \alpha(n)$ be a positive integer, and let $\mu = \mu(n)$ be the largest positive integer for which $2\alpha\mu \leq n$. Then

$$\limsup[1 + 6e^{\frac{1}{2}}\beta^{1/(\mu+1)}(\alpha)]^{\mu} < \infty.$$

(H4) $\|x\|^p K(x) \to 0$, as $x \to \infty$, where the norm $\|x\|$ of $x = (x_1, \ldots, x_p)$ is defined by $\|x\| = \max(|x_1|, \ldots, |x_p|)$.

FIG. 24. *Detail of Figure* 23. *Top: Original detail. Bottom: NL-means filtered detail.*

TABLE 1

Mean square error table. A smaller mean square error indicates that the estimate is closer to the original image. The numbers have to be compared on each row. The square of the number on the left-hand column gives the real variance of the noise. By comparing this square to the values on the same row, it is quickly checked that all studied algorithms indeed perform some denoising. This is a sanity check! In general, the comparison performance corroborates the previously mentioned quality criteria.

Image	σ	GF	AF	TV	YNF	EWF	TIHWT	NL-means
Boat	8	53	38	39	39	33	28	23
Lena	20	120	114	110	129	105	81	68
Barbara	25	220	216	186	176	111	135	72
Baboon	35	507	418	365	381	396	365	292
Wall	35	580	660	721	598	325	712	59

(H5) (i) ϕ is a real valued Borel function defined on $\mathbb{R}$ such that $E|\phi(Y)|^s < \infty$ for some $s > 1$.

(ii)

$$\sup\left[\int_{\mathbb{R}} |\phi(y)|^s f_{XY}(x,y)dy;\ x \in \mathbb{R}^p\right] = C < \infty.$$

(H6) (i) For any point x and x' in $\mathbb{R}^p$ and for some positive constant C (independent of these points),

$$|K(x) - K(x')| \le C\|x - x'\|.$$

(ii) $\int \|x\|K(x)dx < \infty$.

(H7) For any point x in $\mathbb{R}^p$, there are positive constants $C(x)$ such that, for all $x' \in \mathbb{R}^p$ and with J being as in (H8),

(i)

$$\|f_X(x) - f_X(x')\| \le C(x)\|x - x'\|, \qquad \sup[C(x);\ x \in J] < \infty.$$

(ii)

$$\|\psi(x) - \psi(x')\| \le C(x)\|x - x'\|, \qquad \sup[C(x);\ x \in J] < \infty,$$

where $r(x) = E[\phi(Y) \mid X = x]$ and $\psi(x) = r(x)f_X(x)$.

(H8) There exists a compact subset J of $\mathbb{R}^p$ such that

$$\inf[f_X(x);\ x \in J] > 0.$$

Acknowledgments. We thank François Malgouyres, Stéphane Mallat, Yves Meyer, Stanley Osher, Guillermo Sapiro, and Luminita Vese for valuable conversations and comments. Several experiments in this paper have been performed thanks to the public software libraries MegaWave, Wavelab, and Advanced Image Processing Lab.

REFERENCES

[1] L. Alvarez, P.-L. Lions, and J.-M. Morel, *Image selective smoothing and edge detection by nonlinear diffusion.* II, SIAM J. Numer. Anal., 29 (1992), pp. 845–866.

[2] J.-F. Aujol, G. Aubert, L. Blanc-Féraud, and A. Chambolle, *Image decomposition into a bounded variation component and an oscillating component*, J. Math. Imaging Vision, 22 (2005), pp. 71–88.

[3] S. A. Awate and R. T. Whitaker, *Image denoising with unsupervised, information-theoretic, adaptive filtering*, in Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, 2005, to appear.

[4] A. Buades, B. Coll, and J.-M. Morel, *Procédé de traitement de données d'image, par réduction de bruit d'image, et caméra intégrant des moyens de mise en oeuvre du procédé (Image data process by image noise reduction and camera integrating the means for implementing this process)*, French patent application, serial number: 0404837.

[5] F. Catté, F. Dibos, and G. Koepfler, *A morphological scheme for mean curvature motion and applications to anisotropic diffusion and motion of level sets*, SIAM J. Numer. Anal., 32 (1995), pp. 1875–1909.

[6] A. Chambolle and P. L. Lions, *Image recovery via total variation minimization and related problems*, Numer. Math., 76 (1997), pp. 167–188.

[7] T. F. Chan and H. M. Zhou, *Total variation improved wavelet thresholding in image compression*, in Proceedings of the IEEE International Conference on Image Processing, Vol. 2, Vancouver, BC, Canada, 2000, pp. 391–394.

[8] R. R. Coifman and D. Donoho, *Translation-invariant de-noising*, in Wavelets and Statistics, Springer-Verlag, New York, 1995, pp. 125–150.

[9] D. Donoho, *De-noising by soft-thresholding*, IEEE Trans. Inform. Theory, 41 (1995), pp. 613–627.

[10] D. Donoho and I. Johnstone, *Ideal spatial adaptation via wavelet shrinkage*, Biometrika, 81 (1994), pp. 425–455.

[11] S. Durand and M. Nikolova, *Restoration of wavelet coefficients by minimizing a specially designed objective function*, in Proceedings of the IEEE Workshop on Variational and Level Set Methods in Computer Vision, 2003.

[12] S. Durand and J. Froment, *Reconstruction of wavelet coefficients using total variation minimization*, SIAM J. Sci. Comput., 24 (2003), pp. 1754–1767.

[13] A. Efros and T. Leung, *Texture synthesis by non parametric sampling*, in Proceedings of the IEEE International Conference on Computer Vision, Vol. 2, Corfu, Greece, 1999, pp. 1033–1038.

[14] F. Guichard, J. M. Morel, and R. Ryan, *Image Analysis and P.D.E.'s*, preprint.

[15] E. Le Pennec and S. Mallat, *Geometrical image compression with bandelets*, in Proceedings of the SPIE 2003, Vol. 5150, Lugano, Switzerland, 2003, pp. 1273–1286.

[16] E. Levina, *Statistical Issues in Texture Analysis*, Ph.D. thesis, UC-Berkeley, Berkeley, CA, 2002.

[17] M. Lindenbaum, M. Fischer, and A. M. Bruckstein, *On Gabor contribution to image enhancement*, Pattern Recognition, 27 (1994), pp. 1–8.

[18] S. Lintner and F. Malgouyres, *Solving a variational image restoration model which involves L^∞ constraints*, Inverse Problems, 20 (2004), pp. 815–831.

[19] F. Malgouyres, *A noise selection approach of image restoration*, in Proceedings of Wavelet Applications in Signal and Image Processing IX, SPIE Proc. Ser. 4478, SPIE, Bellingham, WA, 2001, pp. 34–41.

[20] S. Mallat, *A Wavelet Tour of Signal Processing*, Academic Press, San Diego, CA, 1997.

[21] B. Merriman, J. Bence, and S. Osher, *Diffusion generated motion by mean curvature*, in Proceedings of the Geometry Center Workshop, Minneapolis, MN, 1992.

[22] Y. Meyer, *Oscillating Patterns in Image Processing and Nonlinear Evolution Equations*, Univ. Lecture Ser. 22, AMS, Providence, RI, 2002.
[23] É. A. Nadaraya, *On non-parametric estimates of density functions and regression curves*, Theory Probab. Appl., 10 (1964), pp. 186–190.
[24] E. Ordentlich, G. Seroussi, S. Verdú, M. Weinberger, and T. Weissman, *A discrete universal denoiser and its application to binary images*, in Proceedings of the IEEE International Conference on Image Processing, Vol. 1, Barcelona, Spain, 2003, pp. 117–120.
[25] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin, *An iterative regularization method for total variation-based image restoration*, Multiscale Model. Simul., 4 (2005), pp. 460–489.
[26] S. Osher, A. Solé, and L. Vese, *Image decomposition and restoration using total variation minimization and the H^{-1} norm*, Multiscale Model. Simul., 1 (2003), pp. 349–370.
[27] P. Perona and J. Malik, *Scale space and edge detection using anisotropic diffusion*, IEEE Trans. Patt. Anal. Mach. Intell., 12 (1990), pp. 629–639.
[28] G. G. Roussas, *Nonparametric regression estimation under mixing conditions*, Stochastic Process. Appl., 36 (1990), pp. 107–116.
[29] L. Rudin and S. Osher, *Total variation based image restoration with free local constraints*, in Proceedings of the IEEE International Conference on Image Processing, Vol. 1, Austin, TX, 1994, pp. 31–35.
[30] L. Rudin, S. Osher, and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Phys. D, 60 (1992), pp. 259–268.
[31] C. Shannon and W. Weaver, *The Mathematical Theory of Communication*, University of Illinois Press, Champaign, IL, 1998.
[32] A. Sharf, M. Alexa, and D. Cohen-Or, *Context-based surface completion*, in Proceedings of the 31st International Conference on Computer Graphics and Interactive Techniques, Los Angeles, CA, 2004, pp. 878–887.
[33] S. M. Smith and J. M. Brady, *SUSAN - a new approach to low level image processing*, International Journal of Computer Vision, 23 (1997), pp. 45–78.
[34] J. Starck, E. Candès, and D. Donoho, *The curvelet transform for image denoising*, IEEE Trans. Image Process., 11 (2000), pp. 670–684.
[35] E. Tadmor, S. Nezzar, and L. Vese, *A multiscale image representation using hierarchical (BV, L^2) decompositions*, Multiscale Model. Simul., 2 (2004), pp. 554–579.
[36] L. Vese and S. Osher, *Modeling textures with total variation minimization and oscillating patterns in image processing*, J. Sci. Comput., 19 (2003), pp. 553–572.
[37] G. S. Watson, *Smooth regression analysis*, Sankhyā Ser. A, 26 (1964), pp. 359–372.
[38] T. Weissman, E. Ordentlich, G. Seroussi, S. Verdu, and M. Weinberger, *Universal discrete denoising: Known channel*, IEEE Trans. Inform. Theory, 51 (2005), pp. 5–28.
[39] L. Yaroslavsky, K. Egiazarian, and J. Astola, *Transform domain image restoration methods: Review, comparison and interpretation*, in Nonlinear Image Processing and Pattern Analysis, SPIE, Bellingham, WA, 2001, pp. 155–169.
[40] L. P. Yaroslavsky, *Digital Picture Processing. An Introduction*, Springer-Verlag, Berlin, 1985.
[41] L. Yaroslavsky and M. Eden, *Fundamentals of Digital Optics*, Birkhäuser Boston, Boston, MA, 1996.
[42] L. Yaroslavsky, *Local adaptive image restoration and enhancement with the use of DFT and DCT in a running window*, in Proceedings of Wavelet Applications in Signal and Image Processing IV, SPIE Proc. Ser. 2825, Denver, CO, 1996, pp. 1–13.