Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

A principled approach to image denoising with similarity kernels involving patches

Arnaud De Decker *,[1], John A. Lee [2], Michel Verleysen

Université catholique de Louvain, Machine Learning Group, place du Levant 3, B-1348 Louvain-la-Neuve, Belgium

ARTICLE INFO

Available online 21 January 2010

Keywords:
Image denoising
Bilateral filtering
Local M-smoothers
Image patches
Mode estimation
Nonlocal means

ABSTRACT

Denoising is a cornerstone of image analysis and remains a very active research field. This paper deals with image filters that rely on similarity kernels to compute weighted pixel averages. Whereas similarities have been based on the comparison of isolated pixel values until recently, modern filters extend the paradigm to groups of pixels called patches. Significant quality improvements result from the mere replacement of pixel differences with patch-to-patch comparisons directly into the filter. Our objective is to cast this generalization within the framework of mode estimation. Starting from objective functions that are extended to patches, this leads us to slightly different formulations of filters proposed in the literature, such as the local M-smoothers, bilateral filters, and the nonlocal means. A fast implementation of these new filters relying on separable linear-time convolutions is detailed. Experiments show that this principled approach further improves the denoising quality without increasing the computational complexity.

1. Introduction

Digital images are massively produced and used in many domains, ranging from multimedia entertainment to professional applications in medicine, security, or geography, for instance. Considering each pixel of an image as a measurement with some statistical uncertainty emphasizes the importance of denoising in the process of image acquisition and generation. Statistical noise can stem from poor light conditions, short exposure, low photon detection efficiency, or electronic transmission faults. Depending on its origins, noise can be additive or multiplicative, Gaussian or Poissonian, or can require an even more complex model.

In this context, denoising techniques aim at reducing the statistical perturbations and recovering as well as possible the true underlying signal. An early and very popular approach was to achieve filtering in the frequency domain, just by trimming high-frequency components of the image spectrum. This computationally fast method has however a major drawback: it tends to smooth out the salient features of the signal, such as edges and textures. Wavelets [1–3] and other transformations in a combined space-frequency domain nicely address this issue and lead to very efficient filtering schemes. This paper rather focuses on nonlinear filtering methods in the space domain. Space domain filters often share many procedural resemblances and most of them work by computing weighted averages of similar pixels. Nevertheless, these methods can emerge from various paradigms. For instance, heat diffusion has inspired denoising techniques such as anisotropic diffusion [4,5] and several other techniques based on partial differential equations [6,7]. Total variation denoising (TV) [8–11] is a special case of image regularization methods that balances a smoothness measure and a fidelity term. The statistical nature of noise can also be addressed by robust statistics and mode estimation. The mean-shift procedure [12,13], the local M-smoothers [14,7] and bilateral filtering [15,16] implement this approach. Modeling images as random fields [18,19] provides yet another strategy.

In the recent years, filtering in the spatial domain has deeply evolved thanks to the introduction of patches. This term refers to small neighborhoods around each pixel, which are used to refine the computation of the pixel similarities. Extending filters to patches has considerably improved their denoising quality. Examples of patch-based filters are the nonlocal means (NLmeans) [20,21], unsupervised information-theoric adaptative filtering (UINTA) [22] and Bayesian approaches [27]. Adaptive patch sizes [23] and iterative updates [23–25] have also been investigated. Other developments of the NLmeans are related to kernel regression [26], probabilistic weights [24], efficient implementations with tree structures [28] or dictionaries [27].

* Corresponding author.
E-mail addresses: arnaud.dedecker@uclouvain.be (A. De Decker), john.lee@uclouvain.be (J.A. Lee).

[1] This author is funded by a Belgian F.R.I.A. grant.

[2] This author is with the Molecular Imaging and Experimental Radiotherapy service at StLuc University Hospital in Belgium. He is a Postdoctoral Researcher funded by the Belgian National Fund of Scientific Research (FNRS).

0925-2312/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2009.12.022

The connections between bilateral filtering, the mean-shift procedure and (iterative) NLmeans on one side, and with diffusion processes, partial differential equations, and spectral graph theory on the other side are pointed out in [29,30]. Robust kernels [25], variable width kernels [31] and specific patch-to-patch distances [27,32] are also reported in the literature.

This paper extends the preliminary ideas in [33]. It aims at casting patch-based filters within the framework of mode estimation and robust statistics. For this purpose, we start from local filtering and extend it to the local M-smoothers and bilateral filtering. For each pixel, these two filters iteratively estimate the closest intensity mode within a local neighborhood. In practice, these filters involve the product of two similarity kernels. The first one indicates whether two pixels are located close to each other in the pixel grid, whereas the second one measures the tonal or radiometric similarity between pixel intensities. Discarding the spatial kernel in bilateral filtering leads to the well-known mean-shift procedure [12,13,34,35]. Introducing patches in the aforementioned filters can be done in two different ways. The first one proceeds by replacing the pixel-to-pixel similarity directly in the expression of the filter. Many publications [20,23,22] follow this approach. Instead, we propose to introduce patches in the objective functions from which the filter expressions are derived. The analytical derivation starting from these generalized objective functions yields filters that entail weighted averages of patches instead of isolated pixels. For this reason we distinguish *patchwise* and *pixelwise* filtering in the following of this paper. In the case of the nonlocal means, patchwise versions are said to be vectorial [27,25] or block-based [20]. In [20], patchwise averaging of partly overlapping blocks is suggested as a way to reduce the computational cost of the nonlocal means. Fully overlapping blocks are used in [27], but the running time of the implemented filter is significantly higher than its pixelwise counterpart. We show that patchwise averaging with fully overlapping patches does not increase the computational cost of patch-based filters, as compared to classical pixelwise averaging described in the literature. Moreover, experiments with various images and several noise levels show that for each considered filter (local M-smoothers, bilateral filtering, nonlocal means), the patchwise versions slightly outperform the pixelwise implementations in a systematic way.

The remainder of this paper is organized as follows. Section 2 briefly defines the image model; it also describes the derivation of state-of-the-art filters with spatial and/or tonal similarity kernels, such as the local M-smoothers, bilateral filtering, and the mean-shift. Section 3 introduces patches in these filters. This leads to the nonlocal means and patch-based versions of local M-smoothers and bilateral filtering. For all these filters we give the patchwise update in addition to the classical pixelwise filter. Experiments and results are detailed and commented in Sections 4 and 5, respectively. Finally, Section 6 draws the conclusions.

2. Image model and nonlinear filters with spatial and/or tonal kernels

Let us assume that an image is a set of pixels located on a regular grid. Then let I denote the set of grid coordinates that are associated with each pixel of the image. Each pixel can then be uniquely identified on the grid by its coordinate vector $\mathbf{i} \in I$ and we (somewhat abusively) call it the **i**th pixel. The observed intensity of each pixel is given by

$$f_{\mathbf{i}} = u_{\mathbf{i}} + \varepsilon_{\mathbf{i}}, \tag{1}$$

where $u_{\mathbf{i}}$ is the noisefree value and $\varepsilon_{\mathbf{i}}$ is the noise component. The latter is independent and identically distributed for each pixel. More specifically, we assume that the noise is Gaussian with zero mean and standard deviation v, that is, $\varepsilon_{\mathbf{i}} \sim G(0, v^2)$.

A model with Gaussian i.i.d. noise can be too simple in many real situations. Noise can indeed result from a combination of several physical phenomena and the assumptions of independence and normality are often invalidated in practice. However, the knowledge of an appropriate noise model can lead to a data transformation whose purpose is to make noise (nearly) normal. For example, Fisz's and Anscombe's variance stabilizing transforms convert Poissonian noise into Gaussian noise [36], making classical filters applicable. Bilateral filtering with Fisz's transform is studied in [39] whereas patch-based filters with a generalization of Anscombe's transform are used in [17].

2.1. Isotropic local filter

Local filtering basically consists in replacing each pixel intensity with an average of the surrounding pixel intensities. Doing so is motivated by statistical considerations. If several measurements were available for each pixel, computing their average would yield a good estimate of the noisefree intensity. As pixels unfortunately contain a single measurement, involving neighboring pixels in the average might yield a useful surrogate, provided they share the same noisefree intensity. If this is not the case, for instance near an intensity jump, then there is a risk of introducing a bias in the filtered values. This justifies that only close neighbors of the pixel to be filtered bring a nonzero contribution in the average.

Working locally in the image entails the definition of a distance function on the image grid. Let $\|\mathbf{i}-\mathbf{j}\|_p$ denote the distance derived from the L_p norm between the **i**th and **j**th pixels on the grid. It is noteworthy that in order to avoid boundary effects, we consider that the image has a toroidal topology (in the two-dimensional case). In practice, the torus consists of four mirrored copies of the image. If the image depicts symbol p, then mirroring leads to $\begin{smallmatrix}\mathrm{p}&\mathrm{q}\\ \mathrm{b}&\mathrm{d}\end{smallmatrix}$. Next, the top and bottom borders of the resulting image are connected together, and so are its left and right borders. This produces a torus without any image discontinuity.

Given a distance on the grid, the neighborhood of the **i**th pixel can be defined as

$$N_{\mathbf{i}} = \{\mathbf{j} \text{ s.t. } \|\mathbf{i}-\mathbf{j}\|_p \le r\}, \tag{2}$$

where r is some radius. The toroidal topology ensures that $|N_{\mathbf{i}}| = |N_{\mathbf{j}}|$ for all **i** and **j**. In other words, all pixels have the same number of neighbors.

Isotropic local filtering can then be obtained starting from the L_2 error

$$E_{\text{iso}}(\hat{\mathbf{u}}) = \frac{1}{2}\sum_{\mathbf{i}\in I}\sum_{\mathbf{j}\in N_{\mathbf{i}}} w_{\mathbf{ij}}(\hat{u}_{\mathbf{i}} - f_{\mathbf{j}})^2, \tag{3}$$

where $\hat{\mathbf{u}} = [\hat{u}_{\mathbf{i}}]_{\mathbf{i}\in I}$ denotes the denoised image and $w_{\mathbf{ij}}$ is a weight that depends on $\|\mathbf{i}-\mathbf{j}\|_2$. The global minimum of this error function is attained when its gradient vanishes. Therefore, if its partial derivative with respect to $\hat{u}_{\mathbf{k}}$ is equated to zero, then the closed-form solution

$$\hat{u}_{\mathbf{i}} = \frac{\sum_{\mathbf{j}\in N_{\mathbf{i}}} w_{\mathbf{ij}} f_{\mathbf{j}}}{\sum_{\mathbf{j}\in N_{\mathbf{i}}} w_{\mathbf{ij}}} \tag{4}$$

comes out. Choosing a softly decaying window such as

$$w_{\mathbf{ij}} = \exp\left(-\frac{\|\mathbf{i}-\mathbf{j}\|_2^2}{2\sigma^2}\right) \tag{5}$$

leads to a close approximation of Gaussian filtering, provided σ is small enough with respect to r. This ensures that the truncation due to the restriction to neighborhood $N_{\mathbf{i}}$ remains negligible. Other options may work as well, such as a hard window, that is, $w_{\mathbf{ij}}=1$.

The total photometry of the image is approximately preserved for zero-mean noise, as we have

$$\sum_{\mathbf{i}\in I}\hat{u}_{\mathbf{i}}^{t}=\sum_{\mathbf{i}\in I}f_{\mathbf{i}}\approx\sum_{\mathbf{i}\in I}u_{\mathbf{i}} \tag{6}$$

just by expanding $\hat{u}_{\mathbf{i}}^{t}$ with (4) and rearranging the terms of the sum in the left-hand side. The last approximation results from (1); the sum of ε_i (symmetric and zero mean noise) converges to zero for a large number of pixels.

In practice, the values of σ and/or r need to be chosen in order to attain a good tradeoff between residual noise variance and bias. In other words, noise should be attenuated without smoothing the underlying signal. Anisotropic variants of local filtering, such as the local M-smoothers and bilateral filtering elegantly simplifies the search for this tradeoff.

2.2. Local M-smoothers

The local M-smoothers (LMSs) [14,7,40] are inspired by robust statistics [37,38], which aim at deriving reliable estimators even in the presence of outliers. In order to reduce the influence of the latter in the calculations, standard L_2 estimators are refined in such a way that the contribution of outliers saturates. For instance, one can rewrite (3) as

$$E_{\mathrm{LMSs}}(\hat{\mathbf{u}})=\sum_{\mathbf{i}\in I}\sum_{\mathbf{j}\in N_{\mathbf{i}}}w_{\mathbf{ij}}\Psi_{\rho}((\hat{u}_{\mathbf{i}}-f_{\mathbf{j}})^{2}/2), \tag{7}$$

where

$$\Psi_{\rho}(z)=\rho^{2}(1-\exp(-z/\rho^{2})). \tag{8}$$

Observing that $\lim_{\rho\to\infty}\Psi_{\rho}(z^{2}/2)=z^{2}/2$, one easily sees the relationship with classical L_2 estimators. Function E_{LMSs} cannot be minimized in closed-form, for it is no longer quadratic. Instead, a gradient descent approach [39] allows us to write

$$\hat{u}_{\mathbf{i}}^{t+1}=\hat{u}_{\mathbf{i}}^{t}-\alpha^{t}\frac{\partial E_{\mathrm{LMSs}}}{\partial\hat{u}_{\mathbf{i}}}\Big|_{\hat{\mathbf{u}}^{t}} \tag{9}$$

$$=\hat{u}_{\mathbf{i}}^{t}-\alpha^{t}\sum_{\mathbf{j}\in N_{\mathbf{i}}}w_{\mathbf{ij}}\Psi'_{\rho}((u_{\mathbf{i}}^{t}-f_{\mathbf{j}})^{2}/2)(\hat{u}_{\mathbf{i}}^{t}-f_{\mathbf{j}}), \tag{10}$$

where t denotes the iteration index and α is the step size. Imposing

$$\alpha^{t}=1\Big/\sum_{\mathbf{j}\in N_{\mathbf{i}}}w_{\mathbf{ij}}\Psi'_{\rho}((\hat{u}_{\mathbf{i}}^{t}-f_{\mathbf{j}})^{2}/2) \tag{11}$$

leads to update rule

$$\hat{u}_{\mathbf{i}}^{t+1}=\frac{\sum_{\mathbf{j}\in N_{\mathbf{i}}}w_{\mathbf{ij}}\Psi'_{\rho}((\hat{u}_{\mathbf{i}}^{t}-f_{\mathbf{j}})^{2}/2)f_{\mathbf{j}}}{\sum_{\mathbf{j}\in N_{\mathbf{i}}}w_{\mathbf{ij}}\Psi'_{\rho}((\hat{u}_{\mathbf{i}}^{t}-f_{\mathbf{j}})^{2}/2).} \tag{12}$$

This particular value of the step size ensures that the total photometry of the image is approximately preserved. With an appropriate image padding, we have

$$\lim_{\rho\to\infty}\sum_{\mathbf{i}\in I}\hat{u}_{\mathbf{i}}^{t}=\sum_{\mathbf{i}\in I}f_{\mathbf{i}}\approx\sum_{\mathbf{i}\in I}u_{\mathbf{i}} \tag{13}$$

for all t. A similar limit can be expressed for $\rho\to 0$ as well. The initialization of (12) is given by $\hat{u}_{\mathbf{i}}^{0}=0$.

The same update rule can alternatively emerge from a fixed-point approach, wherein the partial derivatives are equated with zero. Function Ψ' in the update rule is called the radiometric or tonal kernel because it modulates the contribution of the neighboring pixels according to their intensity. This dependence on the surrounding pixel intensities makes the filter locally adaptive and anisotropic. Width ρ proves to be an additional parameter, compared to isotropic filtering. Its value is typically adjusted with respect to both the noise standard deviation and the height of the intensity jumps one wants to preserve.

The decaying influence of dissimilar pixel intensities makes the LMSs robust against outliers as well as an efficient mode estimator. This can be formally demonstrated by observing that the same update rule comes out of a hill-climbing procedure on a Parzen-window density estimator [34,40]. For this reason, the LMSs is particularly good at denoising piecewise constant signal.

2.3. Bilateral filtering and the mean-shift procedure

Bilateral filtering (BF) [15,16,40] happens to be a slightly different variant of the LMSs. More precisely, BF compares the intensity of the pixel to be filtered with surrounding filtered intensities instead of the noisy ones. Similarly, it computes an average of the filtered intensities rather than the noisy ones. These changes are summarized in update rule

$$\hat{u}_{\mathbf{i}}^{t+1}=\frac{\sum_{\mathbf{j}\in N_{\mathbf{i}}}w_{\mathbf{ij}}\Psi'_{\rho}((\hat{u}_{\mathbf{i}}^{t}-\hat{u}_{\mathbf{j}}^{t})^{2}/2)\hat{u}_{\mathbf{j}}^{t}}{\sum_{\mathbf{j}\in N_{\mathbf{i}}}w_{\mathbf{ij}}\Psi'((\hat{u}_{\mathbf{i}}^{t}-\hat{u}_{\mathbf{j}}^{t})^{2}/2).} \tag{14}$$

The initialization remains the same as for the LMSs and both techniques yield a different result starting from the second iteration.

As a matter of fact, LMSs are strictly local whereas BF is not. Averaging intensities that are already filtered entails a diffusion process. This explains why BF is even better than the LMSs at denoising piecewise constant signals, provided width ρ is small enough compared to the intensity jumps. If not, running BF for many iterations can eventually lead to a flat image; the diffusion process has to be stopped soon enough. Formally, one can observe that BF minimizes

$$E_{\mathrm{BF}}(\hat{\mathbf{u}})=\sum_{\mathbf{i}\in I}\sum_{\mathbf{j}\in N_{\mathbf{i}}}w_{\mathbf{ij}}\Psi_{\rho}((\hat{u}_{\mathbf{i}}-\hat{u}_{\mathbf{j}})^{2}/2), \tag{15}$$

which admits trivial solution $\hat{u}_{\mathbf{i}}=\hat{u}_{\mathbf{j}}$, for $\mathbf{i},\mathbf{j}\in I$.

Choosing $N_{\mathbf{i}}$ as the whole image in (14) and discarding the spatial kernel denoted by $w_{\mathbf{ij}}$ leads to the mean-shift procedure [12,13,34,35].

3. Patch-based filters

Although the LMSs and BF perform pretty well with piecewise constant images, the quality of their results significantly decreases if the image exhibits a more complex structure, with fine textures for instance. As these filters work as mode estimators, they poorly behave if the intensity distributions of neighboring pixels tend to overlap. This means that the signal modes are too close to be efficiently identified. In order to circumvent this difficulty, the problem can be reformulated in such a way that mode overlap is less likely to occur. A classical strategy consists in working in a higher dimensional space. As an illustration, consider two Gaussian modes in a high-dimensional space, with moderate overlap. Projecting this distribution along a single dimension is likely to increase the overlap. The reverse statement is true as well: adding more features or dimensions will probably decrease the mode overlap. In the case of images, each pixel turns out to be a dimension and therefore the idea of considering groups of pixels rather than isolated ones naturally emerges. It sounds particularly interesting to consider groups of adjacent pixels, which we call *patches*, for they are able to represent the local image texture. More formally, we define the patch centered on the $\mathbf{i}$th pixel as

$$P_{\mathbf{i}}=\{\mathbf{j}\ \text{s.t.}\ \|\mathbf{i}-\mathbf{j}\|_{\infty}\le p\}. \tag{16}$$

The L_∞ norm (or city-block distance) allows $P_{\mathbf{i}}$ to represent square block in the image, but any other choice is possible. The toroidal topology of the image (or an appropriate padding) guarantees that a patch can be constructed for each pixel. The intensities observed in a patch are denoted by $\mathbf{f_i} = [f_j]_{j \in P_{\mathbf{i}}}$. The square brackets indicate that intensities are arranged in a vector, always in the same order. Similarly, symbols $\mathbf{u_i}$ and $\hat{\mathbf{u}}_{\mathbf{i}}$ denote the noisefree value of the $\mathbf{i}$th patch and its estimate, respectively. The representation of patches by vectors allows us to compute the pairwise distances in the usual (Euclidean) way.

3.1. Pixelwise patch-based filters

Starting from update rules (12) and (14), intuitive extensions to patches can be written as

$$\hat{u}_{\mathbf{i}}^{t+1} = \frac{\sum_{\mathbf{j} \in N_{\mathbf{i}}} w_{\mathbf{ij}} \Psi'_\rho(\|\hat{\mathbf{u}}_{\mathbf{i}}^t - \mathbf{f_j}\|_2^2/2) f_{\mathbf{j}}}{\sum_{\mathbf{j} \in N_{\mathbf{i}}} w_{\mathbf{ij}} \Psi'_\rho(\|\hat{\mathbf{u}}_{\mathbf{i}}^t - \mathbf{f_j}\|_2^2/2)} \tag{17}$$

and

$$\hat{u}_{\mathbf{i}}^{t+1} = \frac{\sum_{\mathbf{j} \in N_{\mathbf{i}}} w_{\mathbf{ij}} \Psi'_\rho(\|\hat{\mathbf{u}}_{\mathbf{i}}^t - \hat{\mathbf{u}}_{\mathbf{j}}^t\|_2^2/2) \hat{u}_{\mathbf{j}}^t}{\sum_{\mathbf{j} \in N_{\mathbf{i}}} w_{\mathbf{ij}} \Psi'(\|\hat{\mathbf{u}}_{\mathbf{i}}^t - \hat{\mathbf{u}}_{\mathbf{j}}^t\|_2^2/2),} \tag{18}$$

for the LMSs and BF, respectively. This modification involves patch comparisons in the tonal kernel Ψ'_ρ. Choosing a patch radius equal to zero trivially brings back the usual scalar filters. If the spatial kernel denoted $w_{\mathbf{ij}}$ is constant over the whole image, the first iteration of Eqs. (17) and (18) reduce to the nonlocal means [20].

Experiments confirm the expected performance gain (see Section 5). However, the introduction of patches directly into update rules (17) and (18) breaks the connection with error functions (7) and (15): one can no longer claim for sure that these modified update rules actually optimize some objective function. In the case of BF, an attempt to relate patch-based filtering to an error function can be found in [41]. Unfortunately, it involves hybrid patches that mix together noisy constant intensities coming from $\mathbf{f_i}$ and a central varying pixel taken from $\hat{\mathbf{u}}_{\mathbf{i}}^t$. A similar approach is developed in [23].

3.2. Patchwise patch-based filters

As an alternative approach, we propose to introduce patches in error functions (7) and (15) rather than in the update rules. In the case of the LMSs, we write the modified error function as

$$E_{\text{PWPBLMSs}}(\hat{\mathbf{u}}) = \sum_{\mathbf{i} \in I} \sum_{\mathbf{j} \in N_{\mathbf{i}}} w_{\mathbf{ij}} \Psi_\rho(\|\hat{\mathbf{u}}_{\mathbf{i}} - \mathbf{f_j}\|_2^2/2). \tag{19}$$

Next, we minimize it by gradient descent. For this purpose, we first notice that

$$\frac{\partial}{\partial u_{\mathbf{k}}} \frac{\|\mathbf{u_i} - \mathbf{f_j}\|_2^2}{2} = \begin{cases} u_{\mathbf{k}} - f_{\mathbf{j+k-i}} & \text{if} \quad \mathbf{k} \in P_{\mathbf{i}}, \\ 0 & \text{otherwise.} \end{cases} \tag{20}$$

The index of $\mathbf{f}$ is visually explained in Fig. 1. Therefore, the partial derivative of E_{PBLMSs} with respect to $\hat{u}_{\mathbf{k}}$ is given by

$$\frac{\partial E_{\text{PWPBLMSs}}}{\partial \hat{u}_{\mathbf{k}}} = \sum_{\mathbf{i} \in P_{\mathbf{k}}} \sum_{\mathbf{j} \in N_{\mathbf{i}}} w_{\mathbf{ij}} \Psi'_\rho(\|\hat{\mathbf{u}}_{\mathbf{i}} - \mathbf{f_j}\|_2^2/2)(\hat{u}_{\mathbf{k}} - f_{\mathbf{j+k-i}}) \tag{21}$$

and the gradient descent can be written as

$$u_{\mathbf{k}}^{t+1} = u_{\mathbf{k}}^t - \alpha^t \frac{\partial E_{\text{PBLMSs}}}{\partial \hat{u}_{\mathbf{k}}}\Big|_{\mathbf{u}^t} \tag{22}$$

$$= \left(1 - \alpha^t \sum_{\mathbf{i} \in P_{\mathbf{k}}} \sum_{\mathbf{j} \in N_{\mathbf{i}}} w_{\mathbf{ij}} \Psi'_\rho(\|\hat{\mathbf{u}}_{\mathbf{i}}^t - \mathbf{f_j}\|_2^2/2)\right) u_{\mathbf{k}}^t + \alpha^t \sum_{\mathbf{i} \in P_{\mathbf{k}}} \sum_{\mathbf{j} \in N_{\mathbf{i}}} w_{\mathbf{ij}} \Psi'_\rho(\|\hat{\mathbf{u}}_{\mathbf{i}}^t - \mathbf{f_j}\|_2^2/2) f_{\mathbf{j+k-i}}. \tag{23}$$

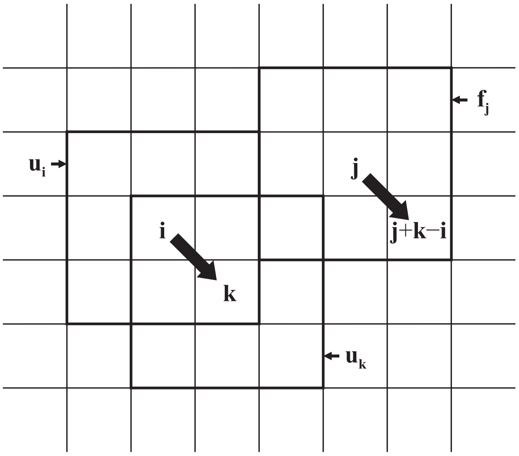

Fig. 1. Differentiating $\|\mathbf{u_i}\text{-}\mathbf{f_j}\|_2^2$ w.r.t. $u_{\mathbf{k}}$ is zero except if $\mathbf{k} \in P_{\mathbf{i}}$. In this case, translating the shift from $\mathbf{i}$ to $\mathbf{k}$ towards $\mathbf{j}$ allows us to determine which index in $P_{\mathbf{j}}$ is involved in the partial derivative.

Fixing the step size according to

$$\alpha^t = 1 \Bigg/ \sum_{\mathbf{i} \in P_{\mathbf{k}}} \sum_{\mathbf{j} \in N_{\mathbf{i}}} w_{\mathbf{ij}} \Psi'_\rho(\|\hat{\mathbf{u}}_{\mathbf{i}}^t - \mathbf{f_j}\|_2^2/2) \tag{24}$$

leads to the update rule given by

$$u_{\mathbf{k}}^{t+1} = \frac{\sum_{\mathbf{i} \in P_{\mathbf{k}}} \sum_{\mathbf{j} \in N_{\mathbf{i}}} w_{\mathbf{ij}} \Psi'_\rho(\|\hat{\mathbf{u}}_{\mathbf{i}}^t - \mathbf{f_j}\|_2^2/2) f_{\mathbf{j+k-i}}}{\sum_{\mathbf{i} \in P_{\mathbf{k}}} \sum_{\mathbf{j} \in N_{\mathbf{i}}} w_{\mathbf{ij}} \Psi'_\rho(\|\hat{\mathbf{u}}_{\mathbf{i}}^t - \mathbf{f_j}\|_2^2/2).} \tag{25}$$

As to patch-based BF, the same reasoning starting from

$$E_{\text{PWPBBF}}(\hat{\mathbf{u}}) = \sum_{\mathbf{i} \in I} \sum_{\mathbf{j} \in N_{\mathbf{i}}} w_{\mathbf{ij}} \Psi_\rho(\|\hat{\mathbf{u}}_{\mathbf{i}} - \hat{\mathbf{u}}_{\mathbf{j}}\|_2^2/2) \tag{26}$$

leads to

$$u_{\mathbf{k}}^{t+1} = \frac{\sum_{\mathbf{i} \in P_{\mathbf{k}}} \sum_{\mathbf{j} \in N_{\mathbf{i}}} w_{\mathbf{ij}} \Psi'_\rho(\|\hat{\mathbf{u}}_{\mathbf{i}}^t - \hat{\mathbf{u}}_{\mathbf{j}}^t\|_2^2/2) \hat{u}_{\mathbf{j+k-i}}^t}{\sum_{\mathbf{i} \in P_{\mathbf{k}}} \sum_{\mathbf{j} \in N_{\mathbf{i}}} w_{\mathbf{ij}} \Psi'_\rho(\|\hat{\mathbf{u}}_{\mathbf{i}}^t - \hat{\mathbf{u}}_{\mathbf{j}}^t\|_2^2/2).} \tag{27}$$

For both patch-based update rules, the initialization remains unchanged, i.e. $u_{\mathbf{k}}^0 = f_{\mathbf{k}}$ for all $\mathbf{k} \in I$. Classical LMSs and BF are found back if the patch radius is set to zero. Just as with the previously described pixelwise version, extending $N_{\mathbf{i}}$ to the whole image and letting $w_{\mathbf{ij}}=1$ leads to an iterative version of the nonlocal means [20].

The difference between the pixelwise and patchwise versions of the patch-based filters is obvious. In the former, the outer sum over $P_{\mathbf{k}}$ is dropped and only the term associated with the central pixel of the patch remains. This shows that the pixelwise updates are approximations of the patchwise ones.

3.3. Computational complexity and implementation

At first glance, the patchwise updates seem to require more operations than the pixelwise ones, because they involve an additional sum. In order to show that this intuition is false, let us first compute the complexity of LMSs and BF without patches. Updates (12) and (14) are to be applied to whole image and are thus repeated $|I|$ times. Each update then requires $\mathcal{O}(|N_{\mathbf{i}}|)$ operations, in order to compute all filter weights and their sum in both their numerator and denominator. This leaves us with a total time complexity of $\mathcal{O}(|I||N_{\mathbf{i}}|)$.

The introduction of patches in the tonal kernel, such as in the pixelwise updates, increases the number of operations to compute each weight of the filter. Computing the patch-to-patch distance has a complexity of $\mathcal{O}(|P_\mathbf{i}|)$, instead of $\mathcal{O}(1)$ previously. This brings the total time complexity to $\mathcal{O}(|I||N_\mathbf{i}||P_\mathbf{i}|)$.

As to the patchwise updates, let us first notice that the terms in the numerator and denominator of (25) and (27) depend on indexes $\mathbf{i}$ and $\mathbf{j}$. A given pair $(\mathbf{i},\mathbf{j})$ occurs in the computation of several filtered intensities. More precisely, looking once again at Fig. 1 shows that this pair is involved in $\hat{u}_\mathbf{k}$ for $\mathbf{k}\in P_\mathbf{i}$. Starting from this observation, we rewrite the filtered intensity in (25) and (27) as

$$\hat{u}_\mathbf{i}^{t+1} = v_\mathbf{i}/n_\mathbf{i}, \tag{28}$$

where temporary variables $v_\mathbf{i}$ and $n_\mathbf{i}$ accumulate the terms of the numerator and denominator, respectively. Next, let us assume that these variables already account for all terms except those related to index pair $(\mathbf{i},\mathbf{j})$, with $\mathbf{i}$ fixed and $\mathbf{j}$ running in $N_\mathbf{i}$. Then, in the case of the LMSs, we accumulate the last terms by writing

$$\mathbf{n_i} \leftarrow \mathbf{n_i} + \sum_{\mathbf{j}\in N_\mathbf{i}} w_\mathbf{ij}\Psi'_\rho(\|\hat{\mathbf{u}}_\mathbf{i}^t - \mathbf{f_j}\|_2^2/2), \tag{29}$$

$$\mathbf{v_i} \leftarrow \mathbf{v_i} + \sum_{\mathbf{j}\in N_\mathbf{i}} w_\mathbf{ij}\Psi'_\rho(\|\hat{\mathbf{u}}_\mathbf{i}^t - \mathbf{f_j}\|_2^2/2)\mathbf{f_j}, \tag{30}$$

where $\mathbf{n_i}$ and $\mathbf{v_i}$ are as usual the patch values centered on the $\mathbf{i}$th pixel. If other terms than those associated to the given value of $\mathbf{i}$ are missing, then the two last updates can be applied for them as well. Eventually, we conclude that the updates can be applied for all $\mathbf{i}\in I$, starting from the initialization $n_\mathbf{i}=0$, $v_\mathbf{i}=0$. This means that computing (29) and (30) for all $\mathbf{i}\in I$, followed by (28) for all $\mathbf{i}\in I$ is equivalent to update (25). A similar reasoning can be followed for BF. In both cases, the computational complexity is distributed as follows. The computation of the patch-to-patch distance requires $\mathcal{O}(|P_\mathbf{i}|)$ operations. This leaves us with a total time complexity of $\mathcal{O}(|I||N_\mathbf{i}||P_\mathbf{i}|)$ in order to update $n_\mathbf{i}$ for all $\mathbf{i}$. On the other hand, updating $v_\mathbf{i}$ entails patch-to-patch additions. Fortunately enough, all patch elements are premultiplied by the same factor, meaning that the patch-to-patch distance and the patch-to-patch accumulation can be carried out sequentially. Therefore, the time complexity remains the same and experimental runtimes for the patchwise and pixelwise filters are not significantly different.

Provided the patches are rectangular blocks, the patch-to-patch distances and the patchwise update can be sped up by carrying them out with separable convolutions. Moreover, the convolution along each dimension of the image with a constant window can be implemented in linear time, by using accumulators. These computational simplifications have been described in [42] for the patch-to-patch distances; we have implemented the same tricks for the patchwise updates. More precisely, in the one-dimensional case, let us assume that the image is a vector with L pixels, the support of the spatial kernel encompasses $2M+1$ pixel, and that the block-shaped patch contains $2K+1$ pixels. The image must be padded with $2K+M$ pixels at both ends, so that we can fetch pixel intensities f_i and $\hat{u}_i^t$ with $1-2K-M\le i\le L+2K+M$; $\hat{u}_i^t$ is initialized to f_i. If $d(i,m)=\|\mathbf{u}_i-\mathbf{f}_{i+m}\|_2^2$ denotes the squared Euclidean patch-to-patch distance, with $-M\le m\le M$, then for fixed m we can compute

$$d(1-K,m) = \sum_{k=-K}^{K}(\hat{u}_{1+k}^t - f_{1+m+k})^2 \tag{31}$$

and for all $1-K<i\le L+K$ we iterate

$$d(i+1,m) = d(i,m) + (\hat{u}_{i+1+K}^t - f_{i+1+K+m})^2 - (\hat{u}_{i+1-K}^t - f_{i+1-K+m})^2. \tag{32}$$

Once this is done, we can obtain the terms involved in (29) and (30) by computing first

$$n(1,m) = \sum_{k=-K}^{K} w_m\Psi'_\rho(d(1+k,m)/2), \tag{33}$$

$$v(1,m) = \sum_{k=-K}^{K} w_m\Psi'_\rho(d(1+k,m)/2)\hat{u}_{1+m}^t, \tag{34}$$

and next, for all $1<i\le L$,

$$n(i+1,m) = n(i,m) + w_m\Psi'_\rho(d(i+1+K,m)/2) - w_m\Psi'_\rho(d(i+1-K,m)/2), \tag{35}$$

$$v(i+1,m) = v(i,m) + w_m\Psi'_\rho(d(i+1+K,m)/2)\hat{u}_{1+m}^t - w_m\Psi'_\rho(d(i+1-K,m)/2)\hat{u}_{1+m}^t, \tag{36}$$

where w_m is the spatial weight that corresponds to $w_\mathbf{ij}$ with $\mathbf{j}-\mathbf{i}=m$. An outer loop running over $-M\le m\le M$ allows us to compute

$$n_i = \sum_{m=-M}^{M} n(i,m), \quad v_i = \sum_{m=-M}^{M} v(i,m), \tag{37}$$

and eventually $\hat{u}_i^{t+1} = v_i/n_i$ for $1\le i\le L$. The sequence of all abovementioned steps amounts to a complexity of $\mathcal{O}(LMK)$ in this one-dimensional case. The generalization to two or more dimensions is straightforward but it requires much space and complicated notations. The most noticeable difference is the need to repeat the linear convolution for each dimension. The total time complexity is then $\mathcal{O}(|I||N_\mathbf{i}|D)$, where D is the image dimensionality.[3] The complexity does not depend on the patch size and only factor D makes it higher than with scalar filters.

4. Images and experimental setup

Nine filters are compared in order to discuss their respective advantages:

- classical (scalar) LMSs and BF,
- pixelwise patch-based (PB) LMSs, BF and NLmeans,
- patchwise patch-based (PWPB) LMSs, BF and PWNLmeans,
- SAFIR, a state-of-the-art filter [43,44].

The test images are shown in Figs. 2(a)–(c); they depict two ladies, namely Lena and Barbara, and boats in a port.

These three well known benchmarks include 512^2 pixels with grayscale intensities ranging from 0 to 255. Each pixel intensity is independently perturbated by Gaussian noise. Three noise levels are retained, with standard deviations equal to 5, 10 and 15, respectively. Each experiment involves $M=5$ noisy copies of each image for the optimization of the parameters, and 100 copies of each image for the final evaluation of the PSNR and its variance.

If u_i and $\hat{u}_\mathbf{i}^m$, respectively, denote the noisefree intensity of the $\mathbf{i}$th pixel and its estimate starting from the m th noisy images, then the denoising quality can be assessed by the mean square error (MSE) or the peak signal to noise ratio (PSNR). The MSE is defined here as

$$\mathrm{MSE}(\mathbf{u},\hat{\mathbf{u}}^m) = \frac{1}{|I|}\sum_{\mathbf{i}\in I}(u_i - \hat{u}_i^m)^2. \tag{38}$$

[3] The complexity is linear in D and thus lower than that in [42] because we perform all subtractions in (32) rather than in each call to the kernel function.

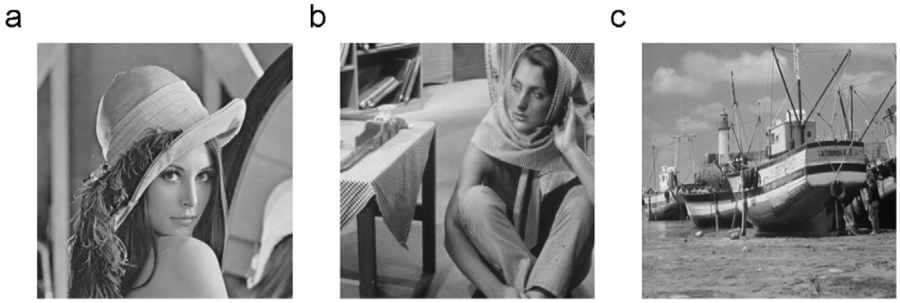

Fig. 2. Images used in this paper: (a) Len; (b) Barbara and (c) boat.

Table 1
RMMSE, PSNR, standard deviation of the PSNR, and parameter values for all images with 100 repetitions, perturbated by additive i.i.d. Gaussian noise with standard deviation equal to 5.

	RMMSE	mean(PSNR)	std(PSNR)	T	ρ	σ	p
Lena ($v=5$)							
Noisy image	5.0032	34.1458	0.0749				
LMSs	3.4073	37.4826	0.0414	1	10.5968	4.8111	0
BF	3.4073	37.4826	0.0414	1	10.5968	4.8111	0
PBLMSs	3.0942	38.3198	0.0410	1	6.0353	4.4747	2
PWPBLMSs	3.0915	38.3274	0.0485	1	5.8082	1.4476	2
PBBF	3.0942	38.3198	0.0410	1	6.0353	4.4747	2
PWPBBF	3.0915	38.3274	0.0485	1	5.8082	1.4476	2
NLmeans	3.1777	38.0885	0.0349	1	5.2161	n.a	1
PWNLmeans	3.1726	38.1025	0.0361	1	5.0932	n.a	1
SAFIR	3.1523	38.1584	0.0371	4	n.a.	n.a.	6
Barbara ($v=5$)							
Noisy image	4.9971	34.1564	0.0763				
LMSs	4.0012	36.0870	0.0638	1	9.2913	3.7327	0
BF	4.0012	36.0870	0.0638	1	9.2913	3.7327	0
PBLMSs	3.5035	37.2408	0.0545	1	6.4241	4.8099	2
PWPBLMSs	3.5014	37.2460	0.0585	1	6.4704	4.1992	2
PBBF	3.5035	37.2408	0.0545	1	6.4241	4.8099	2
PWPBBF	3.5014	37.2460	0.0585	1	6.4604	4.1992	2
NLmeans	3.5289	33.2780	0.0507	1	5.2578	n.a	1
PWNLmeans	3.5258	33.2829	0.0523	1	5.0766	n.a	1
SAFIR	3.5598	37.1024	0.031	4	n.a.	n.a.	6
Boat ($v=5$)							
Noisy image	4.9967	34.1571	0.0638				
LMSs	3.9241	36.2560	0.0646	1	10.1310	2.9358	0
BF	3.9241	36.2560	0.0646	1	10.1310	2.9358	0
PBLMSs	3.6842	36.8039	0.0763	1	6.4056	3.2491	1
PWPBLMSs	3.6822	36.8087	0.0786	1	6.2953	3.1796	1
PBBF	3.6842	36.8039	0.0763	1	6.4056	3.2491	1
PWPBBF	3.6822	36.8087	0.0786	1	6.2953	3.1796	1
NLmeans	3.7213	36.7169	0.0669	1	5.3213	n.a	1
PWNLmeans	3.7177	36.7253	0.0695	1	5.1682	n.a	1
SAFIR	4.0460	35.9903	0.0431	3	n.a.	n.a.	6

T, number of iterations, ρ, radiometric kernel width, σ, spatial kernel width, p, patch radius (in pixels).

Table 2
RMMSE, PSNR, standard deviation of the PSNR, and parameter values for all images with 100 repetitions, perturbated by additive i.i.d. Gaussian noise with standard deviation equal to 10.

	RMMSE	mean(PSNR)	std(PSNR)	T	ρ	σ	p
Lena ($v=10$)							
Noisy image	10.0011	28.1298	0.0865				
LMSs	5.1218	33.9424	0.0591	1	25.9477	2.9327	0
BF	5.1218	33.9424	0.0591	1	25.9477	2.9327	0
PBLMSs	4.5251	35.0182	0.0542	1	12.3355	2.9156	2
PWPBLMSs	4.5153	35.0371	0.0576	2	8.2242	1.8088	2
PBBF	4.5251	35.0182	0.0542	1	12.3355	2.9156	2
PWPBBF	4.5150	35.0376	0.0590	2	8.2221	1.8301	2
NLmeans	4.6258	34.8271	0.0436	1	10.1293	n.a	2
PWNLmeans	4.6052	34.8658	0.0444	1	9.9683	n.a	2
SAFIR	4.4367	35.1897	0.0343	4	n.a.	n.a.	6
Barbara ($v=10$)							
Noisy image	10.0027	28.1285	0.0696				
LMSs	6.0288	32.5262	0.0591	1	26.8511	2.8409	0
BF	6.0288	32.5262	0.0591	1	26.8511	2.8409	0
PBLMSs	5.5583	33.2320	0.0652	1	12.6075	3.4908	2
PWPBLMSs	5.5458	33.2515	0.0591	1	11.6827	3.6703	2
PBBF	5.5583	33.2320	0.0652	1	12.6075	3.4908	2
PWPBBF	5.5458	33.2515	0.0591	1	11.6827	3.6703	2
NLmeans	5.6171	33.1406	0.0494	1	10.6273	n.a	2
PWNLmeans	5.6013	33.1650	0.0429	1	10.1083	n.a	2
SAFIR	5.3935	33.4935	0.0343	3	n.a.	n.a.	6
Boat ($v=10$)							
Noisy image	9.9983	28.1323	0.0844				
LMSs	6.8405	31.4290	0.0502	1	19.2729	4.0971	0
BF	6.8405	31.4290	0.0502	1	19.2729	4.0971	0
PBLMSs	5.5042	33.3169	0.0363	1	9.2067	11.2660	2
PWPBLMSs	5.4621	33.3836	0.0415	1	9.3434	8.6191	2
PBBF	5.5042	33.3169	0.0363	1	9.2067	11.2660	2
PWPBBF	5.4621	33.3836	0.0415	1	9.3434	8.6191	2
NLmeans	5.4968	33.3286	0.0346	1	9.5284	n.a	2
PWNLmeans	5.4782	33.3580	0.0362	1	9.1022	n.a	2
SAFIR	5.6425	33.1014	0.0385	3	n.a.	n.a.	6

T, number of iterations, ρ, radiometric kernel width, σ, spatial kernel width, p, patch radius (in pixels).

The M repetitions allow us to compute the square root of its average value, which is called the RMMSE:

$$\mathrm{RMMSE} = \sqrt{\frac{1}{M}\sum_{m=1}^{M} \mathrm{MSE}(\mathbf{u}, \hat{\mathbf{u}}^m)}. \tag{39}$$

Another widely used error criterion is the peak signal to noise ratio; it is defined as

$$\mathrm{PSNR} = 10\log_{10}\left(\frac{\max(\mathbf{f})^2}{MSE}\right) \tag{40}$$

and therefore depends on the intensity range of the image. In practice, since the images used in the experiments have an intensity range of 0–255, max($\mathbf{f}$) is replaced by 255 in (40).

The filter parameters are σ (the width of the spatial kernel), ρ (the width of the radiometric kernel), T (the number of iterations), and p (the patch radius). All four parameters have been optimized in order to maximize the PSNR. The parameters σ and ρ were initialized at 10 for all experiments and then optimized for each patch size, each and number of iterations. In the specific case of the NLmeans and PWNLmeans, the spatial kernel is constant over large 21-by-21 blocks (a true global

Table 3
RMMSE, PSNR, standard deviation of the PSNR, and parameter values for all images with 100 repetitions, perturbated by additive i.i.d. Gaussian noise with standard deviation equal to 15.

		RMMSE	mean(PSNR)	std(PSNR)	T	ρ	σ	p
Lena ($v=15$)	Fig. 3							
Noisy image	(b)	14.9981	24.6101	0.0826				
LMSs	(c)	6.4146	31.9874	0.0512	1	44.2942	3.2892	0
BF	(d)	6.4146	31.9874	0.0512	1	44.2942	3.2892	0
PBLMSs	(e)	5.5889	33.1843	0.0531	2	12.5253	3.5749	2
PWPBLMSs	(f)	5.5462	33.2509	0.0594	2	11.7869	4.6313	2
PBBF	(g)	5.5887	33.1846	0.0523	2	12.5114	3.5811	2
PWPBBF	(h)	5.5464	33.2506	0.0678	2	11.7814	4.6281	2
NLmeans	(i)	5.7116	32.9956	0.0418	1	13.2817	n.a	2
PWNLmeans	(j)	5.7003	33.0128	0.0421	1	12.4728	n.a	2
SAFIR	(k)	5.5541	33.2389	0.0764	4	n.a.	n.a.	6
Barbara ($v=15$)	Fig. 4							
Noisy image	(b)	15.0007	24.6086	0.0821				
LMSs	(c)	9.2642	28.7946	0.0677	1	28.5618	4.8083	0
BF	(d)	9.2642	28.7946	0.0677	1	28.5618	4.8083	0
PBLMSs	(e)	7.1058	31.0985	0.0741	1	12.2734	13.0870	2
PWPBLMSs	(f)	7.0581	31.1570	0.0770	1	12.4715	10.3464	2
PBBF	(g)	7.1058	31.0985	0.0741	1	12.2734	13.0870	2
PWPBBF	(h)	7.0581	31.1570	0.0770	1	12.4715	10.3464	2
NLmeans	(i)	7.1158	31.0863	0.0379	1	13.2215	n.a	2
PWNLmeans	(j)	7.0992	31.1066	0.0404	1	12.7798	n.a	2
SAFIR	(k)	6.9016	31.3516	0.0601	4	n.a.	n.a.	6
Boat ($v=15$)	Fig. 5							
Noisy image	(b)	14.9953	24.6117	0.0979				
LMSs	(c)	7.6514	30.4560	0.0485	1	42.8622	3.0811	0
BF	(d)	7.6514	30.4560	0.0485	1	42.8622	3.0811	0
PBLMSs	(e)	6.9901	31.2411	0.0508	2	14.1054	2.9948	2
PWPBLMSs	(f)	6.9442	31.2984	0.0550	2	12.9953	3.1038	2
PBBF	(g)	6.9895	31.2419	0.0468	2	14.1169	2.9893	2
PWPBBF	(h)	6.9440	31.2986	0.0414	2	13.0046	3.1070	2
NLmeans	(i)	7.3249	30.8348	0.0370	1	13.8132	n.a	1
PWNLmeans	(j)	7.2949	30.8704	0.0332	1	12.0148	n.a	2
SAFIR	(k)	6.999	31.2301	0.0449	4	n.a.	n.a.	6

T, number of iterations, ρ, radiometric kernel width, σ, spatial kernel width, p, patch radius (in pixels).

search is too expensive). The patch radius p was allowed to take any integer value between 0 and 7 and the number of iterations was chosen between 0 and 5. Once the optimal parameter values are determined, the quantitative results given in Section 5 stem from the evaluation of PSNR and RMMSE on a larger set of 100 noisy images.

It is worth mentioning that the number of iterations of the filters was optimized according to criteria (39) and (40). Early stopping looks natural for bilateral filtering and other filters involving diffusion processes that lead to spurious steady states. In contrast, one might wonder why early stopping should be applied to filters with limited diffusion such as the local M-smoothers, whose steady state is intuitively not expected to be spurious. Actually, any filter can be considered to be a statistical estimator, for which the unknown parameter is the noisefree image. Though unknown by the estimator, the true parameter value can however be available to the experimenter for quality assessment purposes. Within this framework, a pertinent estimator is a function that owns the same optima as the quality criterion, without referring to the actual parameter value. In other words, the estimator is a surrogate of the quality criterion and it can be characterized by both its bias and variance. It is intuitively easy to observe that all filters considered in this paper are biased. They achieve weighted averages of noisy pixels or patches that come from different parts of the image. Thus, nothing prevents these filters to average together pixels or patches that have different distributions. This intuitively shows that filters with and without diffusion are biased, except if the radiometric width ρ is zero (no filtering at all). If $\rho=\infty$, the expected worst-case bias for the LMSs is the difference between the noisefree image and the noisefree image convolved with the spatial kernel. Similarly, the expected bias for BF is the difference between the noisefree image and the noisefree image convolved with the spatial kernel as many times as the number of iterations. In the case of an iterative estimator, a nonzero bias means that its steady state might be irrelevant. On the other hand, any intermediate state can be roughly seen as a mixture of the steady state (low variance but nonzero bias) with the initial noisy image (high variance but unbiased under reasonable assumption). Early stopping then proves to be an easy way to improve the bias/variance tradeoff. In machine learning, early stopping is a way to prevent overfitting, which turns out to be closely related to the abovementioned bias/variance tradeoff.

5. Results

This section details the results obtained with the various images and noise levels. Visual examples are only provided for the highest noise level as it provides the best support to evaluate the differences between the filters. The visual evaluation will be done

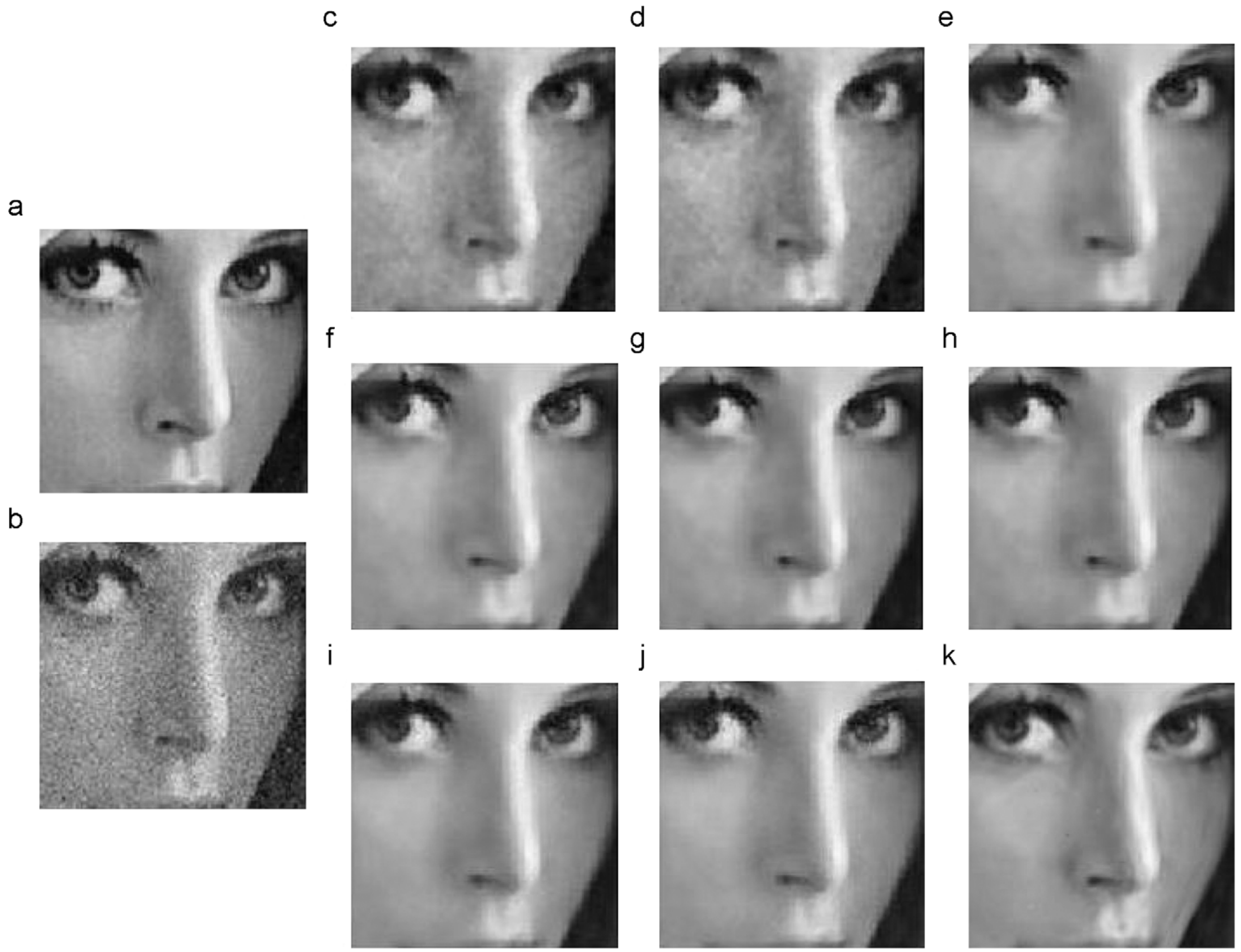

Fig. 3. Samples of Lena's image: (a) noisefree; (b) noisy with $v=15$; (c) LMSs; (d) BF; (e) PBLMSs; (f) PWPBLMSs; (g) PBBF; (h) PWPBBF; (i) NLmeans and (j) PWNLmeans.

for a selected part of the image in order to be able to analyze the behavior of each filter in a finer way.

5.1. *Images with additive Gaussian noise of standard deviation:* $v=5$

This first experiment features noise of very low intensity and is mainly used to control for eventual distortions, smoothings or artifacts generated during the filtering process. However, even with this small noise level, it is already possible to show that the filters perform differently. The RMMSE, along with the mean and standard deviation of the PSNR, are reported in Table 1 for all images. For this noise level, the best results are obtained with a single iteration, which explains the identical figures for the LMSs and BF, PBLMSs and PBBF, and PWPBLMSs and PWPBBF: in this case, these filters are identical. In all cases, the PSNRs given by scalar filters (LMSs and BF) are significantly lower than those of the patch-based filters. The scalar filters also use a wider σ and ρ than the patch-based filters. For the other filters (PBLMSs, PBBF, NLmeans), the PSNR values are systematically slightly higher for the patchwise versions than for the pixelwise ones. However, when having a look to the standard deviations, they cannot be considered to be statistically significant. The performances of the SAFIR filters are very close to the best results obtained by the principled filters. These results show that the patchwise version perform slightly better than the pixelwise versions, even if the small performance gain shows that the contribution of the additional terms in the patchwise filters remains limited, at least when the parameters of the filters (including the number of iterations) are optimized.

5.2. *Images with additive Gaussian noise of standard deviation:* $v=10$

The second experiment involves the same three images with a higher noise level, which significantly distort the image details. The fine features of the noisy images are now harder to see and the capability of the filters to recover these features becomes another criterion to consider. Table 2 indicates the RMMSE, mean and standard deviation of the PSNR and parameter values of each filter. As in the first experiment, the scalar filters perform systematically worse than the patch-based filters. For all images, the patchwise versions of the patch-based filters exhibit a slightly higher PSNR than their pixelwise counterparts. The results from the principled filters are comparable to those

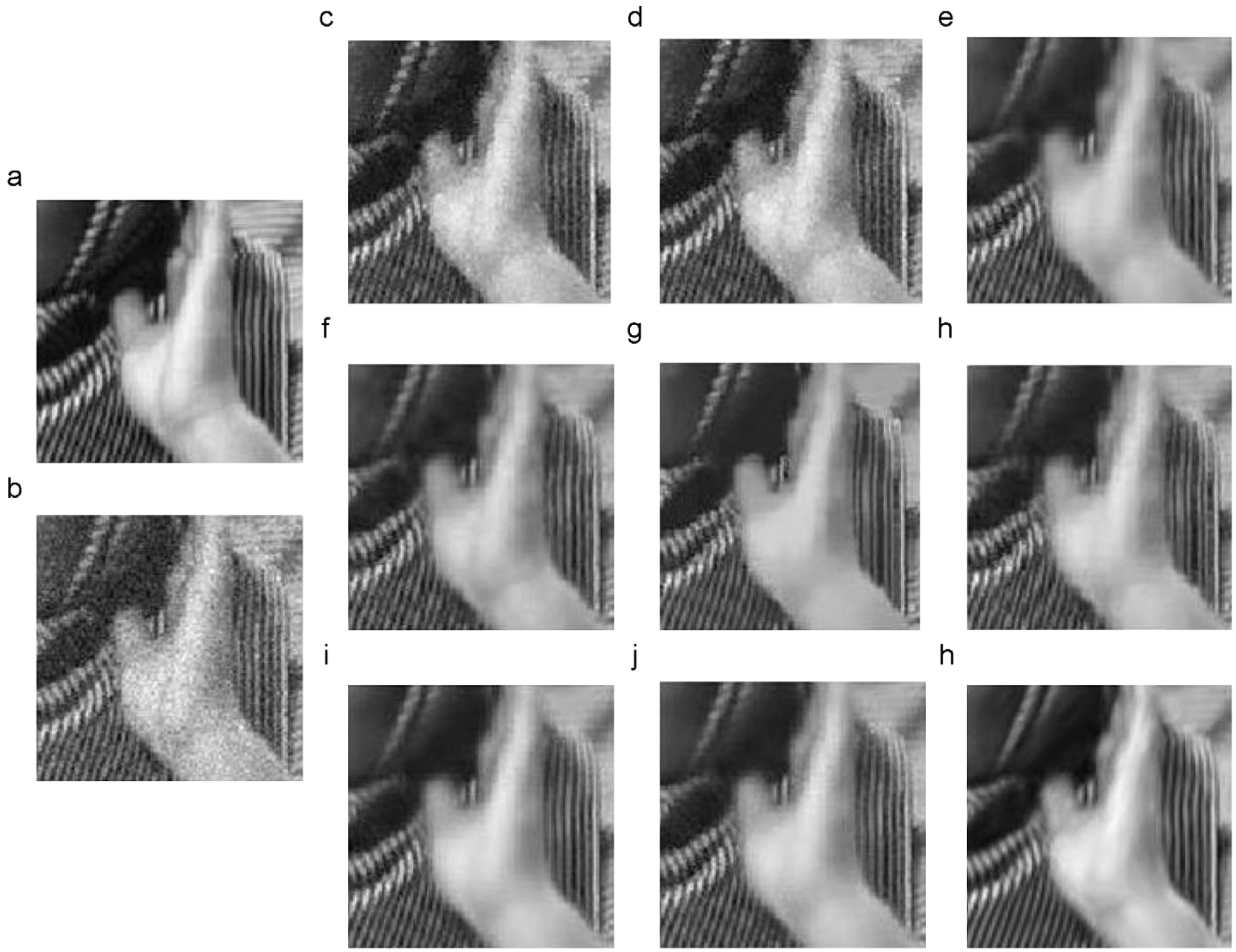

Fig. 4. Samples of Barbara's image: (a) noisefree; (b) noisy with $v=15$; (c) LMSs; (d) BF; (e) PBLMSs; (f) PWPBLMSs; (g) PBBF; (h) PWPBBF; (i) NLmeans and (j) PNLmeans.

obtained by the SAFIR filter. Even if the performance gain is not statistically significant, it is again present for all tested filters, and on all images. For this experiment, optimum values of the PSNR for the patch-based filters are found with patch radius equal to 2.

5.3. Images with additive Gaussian noise of standard deviation: $v=15$

In this third experiment, the standard deviation of the noise is high enough to completely absorb the fine details in the images. In Lena's picture, the noise distorts the textures of the hat fabric, eyelashes, and some parts of the feathers. In Barbara's picture, the features on the hand and wrists are lost in the noise, while the stripes are salient enough to be recovered by the filters. In the boat image, the features on the boats and the sea are also lost due to noise and should be hard to recover. Table 3 indicates the RMMSE, mean and standard deviation of the PSNR and parameter values for each image and filter. Once again, patchwise averaging in patch-based filters slightly outperform their pixelwise versions on all images even if the performance gain is not statistically significant. The lower PSNR is given by the scalar filters.

Typical Lena, Barabara and boat images are displayed in Figs. 3–5. Scalar filters attain their best RMMSE value by merely achieving a light denoising; images (c) and (d) remain noisy. The visual feeling for images (e)–(i) is much more satisfying and barely frothy. Patch-based filters can attenuate noise without distorting the image details, whereas the tradeoff for scalar filters consists of a light denoising that prevents intensities from biases. Most of the details that are not completely lost in the noise are nicely preserved. However, the skin and wrist in the Barbara picture and the sea and boats in the boat image are smoothed by the denoising process. The images show that none of the filters noticeably distorts the images or generates artifacts at this level of noise.

6. Conclusions

Efficient image filtering can be carried out by computing weighted averages of similar pixels. In recent publications, the similarity kernel compares image patches rather than single pixels, leading to the so-called patch-based filters. Within the framework of robust statistics and mode estimation, this generalization into a higher-dimensional space allows for an improved

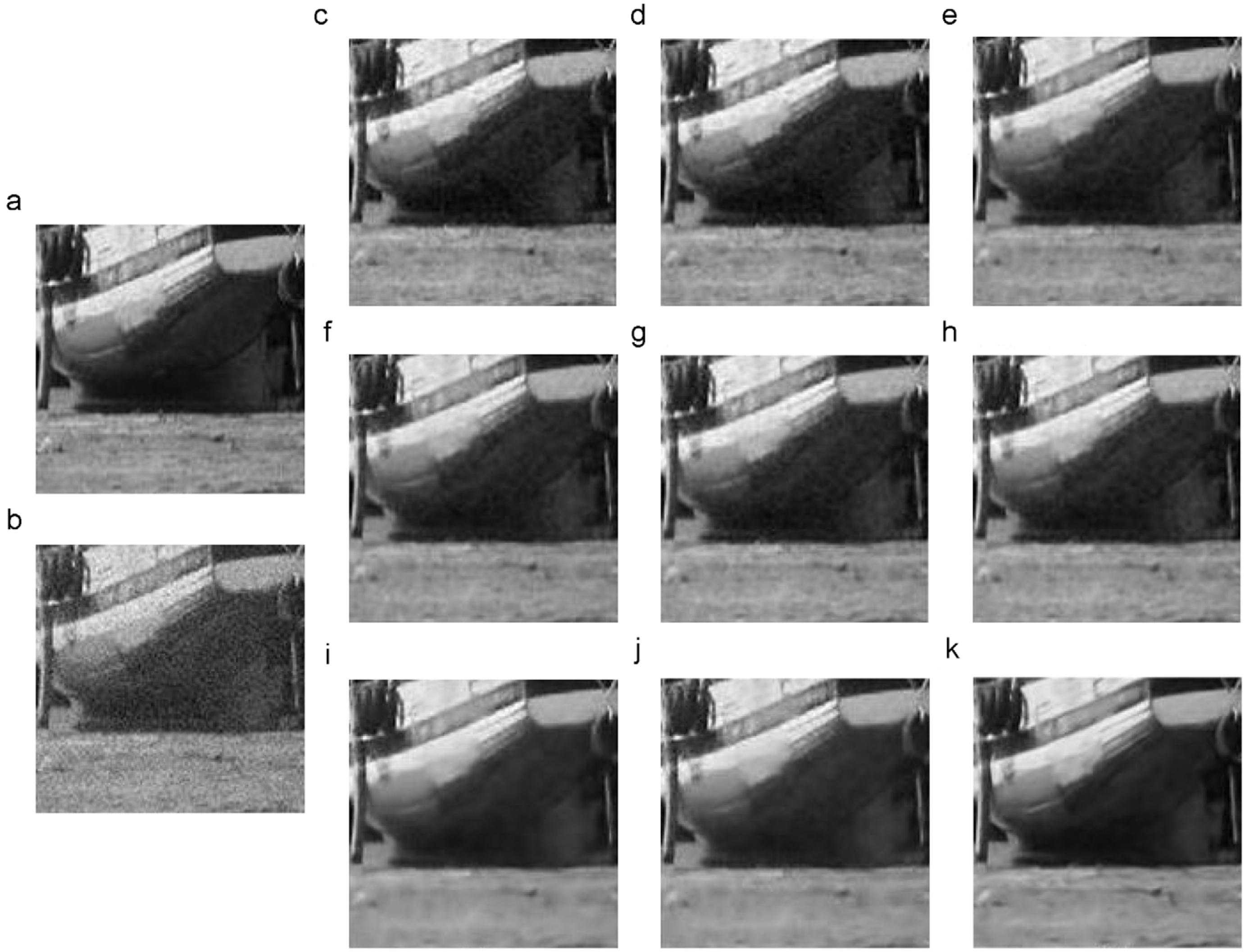

Fig. 5. Samples of boat image: (a) noisefree; (b) noisy with $v = 15$; (c) LMSs; (d) BF; (e) PBLMSs; (f) PWPBLMSs; (g) PBBF; (h) PWPBBF; (i) NLmeans and (j) PWNLmeans.

mode discrimination and thus increased performance. Patches can actually be introduced either directly in the final expression of the filters or in their underlying objective functions. This paper has followed the second approach, which leads to filters that achieve weighted averages of whole patches instead of averages of pixels in the usual filters; the latter can then be seen as pixelwise approximations of the patchwise filters. We demonstrate that this modification does not increase the time complexity of the considered filters and we describe an implementation relying on separable linear-time convolutions. Moreover, experiments with several images, filters, and noise levels show that patch-based filters with patchwise averaging slightly outperform the corresponding pixelwise versions in terms of RMSE and PSNR. As the pixelwise approximations provide no computational speedup, we therefore recommend using patchwise averages in patch-based filters.

References

[1] R. Charnigo, J. Sun, R. Muzic, A semi-local paradigm for wavelet denoising, IEEE Transactions on Image Processing 15 (3) (2006) 666–677.

[2] R. Coifman, D. Donoho, Translation-invariant de-noising, Wavelets and Statistics (1995) 125–150.

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

[4] P. Perona, J. Malik, Scale-space and edge-detection using anisotropic diffusion, IEEE Transactions on Pattern Analysis and Machine Intelligence 12 (7) (1990) 629–639.

[5] M.J. Black, G. Sapiro, D.H. Marimont, D. Heeger, Robust anisotropic diffusion, IEEE Transactions on Image Processing 7 (3) (1998) 421–432.

[6] J. Weickert, Anisotropic Diffusion in Image Processing, ECMI Series, Teubner-Verlag, Stuttgart, 1998.

[7] G. Winkler, V. Aurich, K. Hahn, A. Martin, K. Rodenacker, Noise reduction in images: some recent edge-preserving methods, Mathematik, Informatik und Statistik, Statistik, Sonderforschungsbereich 386 (1998).

[8] S. Osher, M. Burger, D. Goldfarb, J. Xu, W. Yin, An iterative regularization method for total variation-based image restoration, Multiscale Modeling & Simulation 4 (2) (2005) 460–489.

[9] S. Osher, E. Fatemi, Nonlinear total variation based noise removal algorithms, Physica D 60 (1992) 259–268.

[10] C.R. Vogel, M.E. Oman, Iterative methods for total variation denoising, Journal on Scientific Computing 17 (1) (1996) 227–238.

[11] E. Tadmor, S. Nezzar, L. Vese, A multiscale image representation using hierarchical (BV,L^2) decompositions, Multiscale Modeling & Simulation 2 (4) (2004) 554–579.

[12] Y. Cheng, Mean shift, mode seeking, and clustering, IEEE Transactions on Pattern Analysis and Machine Intelligence 17 (8) (1995) 790–799.

[13] D. Comaniciu, P. Meer, Mean shift: a robust approach toward feature space analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (5) (2002) 603–619.

[14] C.K. Chu, I.K. Glad, F. Godtliebsen, J.S. Marron, Edge-preserving smoothers for image processing, Journal of the American Statistical Association 93 (442) (1998) 526–541.

[15] C. Tomasi, R. Manduchi, Bilateral filtering for gray and color images, in: International Conference on Computer Vision, Bombay, India, 1998, pp. 839–846.

[16] M. Elad, On the origin of the bilateral filter and ways to improve it, IEEE Transactions on Image Processing 11 (10) (2002) 1141–1151.

[17] J. Boulanger, Ch. Kervrann, P. Bouthemy, P. Elbau, J.-B. Sibarita, J. Salamero. Patch-based non-local functional for denoising fluorescence microscopy image sequences. IEEE Transactions on Medical Imaging 28 (12), 2009.

[18] S. Roth, M.J. Black, Fields of experts, International Journal of Computer Vision 82 (2) (2009) 205–229.

[19] S. Roth, M.J. Black, Steerable random fields, in: International Conference on Computer Vision, Rio de Janeiro, Brazil, October 2007, pp. 1–8.

[20] A. Buades, B. Coll, J.M. Morel, A review of image denoising algorithms, with a new one, Multiscale Modeling & Simulation 4 (2) (2005) 490–530.

[21] A. Buades, B. Coll, J.-M. Morel, The staircasing effect in neighborhood filters and its solution, IEEE Transactions on Image Processing 15 (6) (2006) 1499–1505.

[22] S.A. Awate, R.T. Whitaker, Image denoising with unsupervised, information-theoric, adaptative filtering, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (3) (2006) 364–376.

[23] C. Kervrann, J. Boulanger, Optimal spatial adaptation for patch-based image denoising, IEEE Transactions on Image Processing 15 (10) (2006) 2866–2878.

[24] C.-A. Deledalle, L. Denis, F. Tupin, Iterative weighted maximum likelihood denoising with probabilistic patch-based weights, IEEE Transactions on Image Processing 18 (12), 2009.

[25] B. Goossens, H. Luong, A. Pizurica, W. Philips, An improved non-local denoising algorithm, in: International Workshop on Local and Non-Local Approximation in Image Processing, Lausanne, Switzerland, 2008, pp. 25–29.

[26] P. Chatterjee, P. Milanfar, A generalization of non-local means via kernel regression, in: Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, vol. 6814, March 2008.

[27] C. Kervrann, J. Boulanger, P. Coup, Bayesian non-local means filter, image redundancy and adaptative dictionaries for noise removal, in: Conference on Scale-Space and Variational Methods, vol. 4485, Ischia, Italy, 2007, pp. 520–532.

[28] T. Brox, O. Kleinschmidt, D. Cremers, Efficient nonlocal means for denoising of textural patterns, IEEE Transactions on Image Processing 17 (7) (2008) 1083–1092.

[29] G. Gilboa, S. Osher, Nonlocal operators with applications to image processing, Multiscale Modeling & Simulation 7 (3) (2008) 1005–1028.

[30] G. Gilboa, S. Osher, Nonlocal linear image regularization and supervised segmentation, Multiscale Modeling & Simulation 6 (2) (2007) 595–630.

[31] N. Azzabou, N. Paragios, F. Cao, F. Guichard, Variable bandwidth image denoising using image-based noise models, in: IEEE Conference in Computer Vision and Pattern Recognition, Minneapolis, MN, June 2007, pp. 1–7.

[32] T. Tasdizen, Principal components for non-local means image denoising, in: IEEE International Conference on Image Processing, (ICIP), San Diego, CA, October 2008, pp. 1728–1731.

[33] A. De Decker, J.A. Lee, M. Verleysen, Patch-based bilateral filter and local m-smoother for image denoising, in: European Symposium on Artificial Neural Networks—Advances in Computational Intelligence and Learning (ESANN), Bruges, Belgium, April 2009, pp. 95–100.

[34] R. van den Boomgaard, J. van de Weijer, On the equivalence of local-mode finding, robust estimation and mean-shift analysis as used in early vision tasks, in: IEEE Conference on Pattern Recognition, vol. 3, Quebec, Canada, August 2002, pp. 927–930.

[35] D. Barash, D. Comaniciu, A common framework for nonlinear diffusion, adaptive smoothing, bilateral filtering and mean-shift, Image and Video Computing 22 (1) (2004) 73–81.

[36] P. Fryzlewicz, G.P. Nason, A Haar–Fisz algorithm for poisson intensity estimation, Journal of Computational and Graphical statistics 13 (3) (2004) 621–638.

[37] P.J. Huber, Robust Statistics. Wiley Series in Probability and Mathematical Statistics, Wiley, New York, 1981.

[38] F.R. Hampel, E.M. Ronchetti, P.J. Rousseeuw, W.A. Stahel, Robust Statistics, Wiley, New York, 2005.

[39] J.A. Lee, X. Geets, V. Gregoire, A. Bol, Edge-preserving filtering of images with low photon counts, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (6) (2008) 1014–1027.

[40] P. Mrazek, J. Weickert, A. Bruhn, Robust Estimation and smoothing with Spatial and Tonal Kernels. Geometric Properties for Incomplete Data, Springer, Berlin, 2006.

[41] S. Kindermann, S. Osher, P.W. Jones, Deblurring and denoising of images by nonlocal functionals, Multiscale Modeling & Simulation 4 (4) (2005) 1091–1115.

[42] J. Darbon, A. Cunha, T.F. Chan, S. Osher, G. Jensen, Fast nonlocal filtering applied to electron cryomicroscopy, in: IEEE International Symposium on Biomedical Imaging, Paris, France, May 2008, pp. 1331–1334.

[43] Ch. Kervrann, J. Boulanger, Local adaptivity to variable smoothness for exemplar-based image denoising and representation, International Journal of Computer Vision 79 (1) (2008) 45–69.

[44] Ch. Kervrann, J. Boulanger, Optimal spatial adaptation for patch-based image denoising, IEEE Transactions on Image Processing 15 (10) (2006) 2866–2878.

Arnaud De Decker was born in Brussels, Belgium, in 1982. He received his M.Sc. degree in physics (space, climate and earth physics) in 2005 and an M.Sc. in statistics in 2006, both from the Université catholique de Louvain (UCL, Belgium). His main research interests are ECG automatic annotation, medical image denosing and deblurring, and automatic tumor and organs delineation on medical images. He is now a Ph.D. student in the machine learning group of the Polytechnic School at UCL.

John Aldo Lee was born in 1976 in Brussels, Belgium. He received the M.Sc. degree in Applied Sciences (Computer Engineering) in 1999 and the Ph.D. degree in Applied Sciences (Machine Learning) in 2003, both from the Université catholique de Louvain (UCL, Belgium). His main interests are nonlinear dimensionality reduction, intrinsic dimensionality estimation, independent component analysis, clustering, and vector quantization. He is a member of the UCL Machine Learning Group and is now a Research Associate with the Belgian FNRS (Fonds National de la Recherche Scientifique). His current work aims at developing specific image enhancement techniques for positron emission tomography in the Molecular Imaging and Experimental Radiotherapy Department of the Saint-Luc University Hospital (Belgium).

Michel Verleysen was born in 1965 in Belgium. He received the M.S. and the Ph.D. degrees in Electrical Engineering from the Université catholique de Louvain (Belgium) in 1987 and 1992, respectively. He was an Invited Professor at the Swiss Ecole Polytechnique Fédérale de Lausanne (E.P.F.L.), Switzerland, in 1992, at the Université d'Evry Val d'Essonne (France) in 2001, and at the Université Paris 1–Panthéon-Sorbonne in 2002–2004. He is a former Research Director with the Belgian FNRS (Fonds National de la Recherche Scientifique) and a Professor at the Université catholique de Louvain. He is Editor-in-Chief of the Neural Processing Letters journal, Chairman of the Annual European Symposium on Artificial Neural Networks (ESANN) Conference, Associate Editor of the IEEE Transactions on Neural Networks journal, and member of the editorial board and program committee of several journals and conferences on neural networks and learning. He is the author or the co-author of about 200 scientific papers in international journals and books or communications to conferences with reviewing committee. He is the co-author of the scientific popularization book on artificial neural networks in the series "Que Sais-Je?," in French. His research interests include machine learning, artificial neural networks, self-organization, time-series forecasting, nonlinear statistics, adaptive signal processing, and high-dimensional data analysis.