

Bayesian non-local means filter, image redundancy and adaptive dictionaries for noise removal

Charles Kervrann, Jérôme Boulanger, Pierrick Coupé

▶ **To cite this version:**

Charles Kervrann, Jérôme Boulanger, Pierrick Coupé. Bayesian non-local means filter, image redundancy and adaptive dictionaries for noise removal. SSVM'07 Proceedings of the 1st international conference on Scale space and variational methods in computer vision, May 2007, Ischia, Italy. pp.520-532. hal-00645444

HAL Id: hal-00645444
https://hal.science/hal-00645444

Submitted on 28 Nov 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Bayesian Non-Local Means Filter, Image Redundancy and Adaptive Dictionaries for Noise Removal

Charles Kervrann[1,3], Jérôme Boulanger[1,3], and Pierrick Coupé[2]

[1] INRIA, IRISA, Campus de Beaulieu, 35 042 Rennes, France
[2] Université de Rennes 1, IRISA, Campus de Beaulieu, 35 042 Rennes, France
[3] INRA - MIA, Domaine de Vilvert, 78 352 Jouy-en-Josas, France

Abstract. Partial Differential equations (PDE), wavelets-based methods and neighborhood filters were proposed as locally adaptive machines for noise removal. Recently, Buades, Coll and Morel proposed the *Non-Local (NL-) means filter* for image denoising. This method replaces a noisy pixel by the weighted average of other image pixels with weights reflecting the similarity between local neighborhoods of the pixel being processed and the other pixels. The *NL-means filter* was proposed as an intuitive *neighborhood filter* but theoretical connections to diffusion and non-parametric estimation approaches are also given by the authors. In this paper we propose another bridge, and show that the *NL-means filter* also emerges from the Bayesian approach with new arguments. Based on this observation, we show how the performance of this filter can be significantly improved by introducing adaptive local dictionaries and a new statistical distance measure to compare patches. The new *Bayesian NL-means filter* is better parametrized and the amount of smoothing is directly determined by the noise variance (estimated from image data) given the patch size. Experimental results are given for real images with artificial Gaussian noise added, and for images with real image-dependent noise.

1 Introduction

Denoising (or restoration) is still a widely studied and an unsolved problem in image processing. Many methods have been suggested in the literature, and a recent outstanding review of them can be found in [4]. Some of the more advanced methods are based on PDEs [28, 29, 33, 37] and aim at preserving the image details and local geometries while removing the undesirable noise; in general, an initial image is progressively approximated by filtered versions which are smoother or simpler in some sense. Other methods incorporate a neighborhood of the pixel under consideration and perform some kind of averaging on the gray values. One of the earliest examples for such filters has been presented by Lee [23] and a recent evolution is the so-called *bilateral filter* [34] with theoretical connections to local mode filtering [38], non-linear diffusion [3, 5] and nonlocal regularization approaches [27, 12].

However, natural images often contain many structured patterns which can be misclassified either as details to be preserved or noise, when usual neighborhood filters are applied. Very recently, the so-called *NL-means filter* has been proposed by Buades

et al. [4] that can deal with such a "structured" noise: for a given pixel, the restored gray value is obtained by the weighted average of the gray values of all pixels in the image; each weight is proportional to the similarity between the local neighborhood of the pixel being processed and the neighborhood corresponding to the other image pixels. A similar patch-based regularization approach based on the key idea of iteratively increasing a window at each pixel and adaptively weighting input data has been also proposed in [21] with excellent results on a commonly-used image database [31]. The success of the *NL-means filter* (see [21, 22, 26, 25, 7, 16, 2]), inspired by the Efros and Leung's exemplar-based approach for texture synthesis [11], is mainly related to image redundancy. A similar idea was early and independently proposed for Gaussian [9] and impulse noise [36, 39] removal in images, and more recently for image inpainting [8]. Similarities between image patches have been also used in the early 90's for texture segmentation [15, 20]. More recently, other recent denoising methods demonstrated that representations based on local image patches outperform the best denoising methods of the state-of-the-art [1, 21, 10, 18, 13]; in [32], patch-based Markov random field (MRF) models and learning techniques have been introduced to capture non-local pairwise interactions, and were successfully applied in image denoising.

In this paper, we present a new Bayesian motivation for the *NL-means filter* briefly described in Section 2. In Section 3, we adopt a blockwise (vectorial) representation and introduce spatially adaptive dictionaries in the modeling for better contrast restoration (Section 4). Using the proposed Bayesian framework, we revise the usual Euclidean distance used for patch comparison, yielding to a filter which is better parametrized, and with a higher performance. In Section 4, we also show how smooth parts in the image can be are better recovered if the restored image is "recycled" once. Experimental results on artificial and real images are presented in Section 5, and the performance is very close to the most competitive denoising methods. It is worth noting that the proposed modeling framework is general and could be used to restore images corrupted by non-Gaussian noises in applications such as biomedical imaging (microscopy, ultrasound imagery, ...) or remote sensing.

2 Image denoising with the NL-means filter

In this section, a brief overview of the *NL-means* method introduced in [4] is presented. Consider a gray-scale image $z = (z(x))_{x\in\Omega}$ defined over a bounded domain $\Omega \subset \mathbb{R}^2$, (which is usually a rectangle) and $z(x) \in \mathbb{R}_+$ is the noisy observed intensity at pixel $x \in \Omega$. The *NL-means filter* is defined as

$$NL\, z(x) = \frac{1}{C(x)} \sum_{y\in\Omega} w(x,y)\, z(y) \tag{1}$$

where $NL\, z(x)$ at pixel x is the weighted average of all gray values in the image and $C(x)$ is a normalizing factor, i.e. $C(x) = \sum_{y\in\Omega} w(x,y)$. The weights $w(x,y)$ defined as

$$w(x,y) = \exp\left(-\frac{1}{h^2}\int_{\mathbb{R}^2} G_a(t)|z(x+t)-z(y+t)|^2 dt\right) := \exp -\frac{\|\mathbf{z}(x)-\mathbf{z}(y)\|^2_{2,a}}{h^2} \tag{2}$$

express the amount of similarity between the vectorized image patches $\mathbf{z}(x)$ and $\mathbf{z}(y)$ (or neighborhoods) of each pair of pixels x and y involved in the computation. The decay parameter $h \approx 12\sigma$ acts as a filtering parameter. A Gaussian kernel $G_a(\cdot)$ of standard deviation a is used to take into account the distance between the central pixel and other pixels in the patch. In (2), the pixel intensities of a $\sqrt{n} \times \sqrt{n}$ square neighborhood $B(x)$ centered at pixel x, are taken and reordered lexicographically to form a n-dimensional vector $\mathbf{z}(x) := (z(x_k), x_k \in B(x)) \in \mathbb{R}^n$. In [4], it was shown that 7×7 patches are able to take care of the local geometries and textures seen in the image while removing undesirable distortions. The range of the search space in the *NL-means* algorithm can be as large as the whole image. In practice, it is necessary to reduce the total number of computed weights – $|\Omega|$ weights for each pixel – to improve the performance of the algorithm. This can be achieved by selecting patches in a *semi-local* neighborhood corresponding to a search window $\Delta(x)$ of 21×21 pixels. The *NL-means filter* we will now consider, is then defined as

$$NL_h z(x) = \frac{1}{C(x)} \sum_{y \in \Delta(x)} e^{-\|\mathbf{z}(x)-\mathbf{z}(y)\|^2/h^2} z(y), \;\; C(x) = \sum_{y \in \Delta(x)} e^{-\|\mathbf{z}(x)-\mathbf{z}(y)\|^2/h^2} \quad (3)$$

where, for the sake of simplicity, $\|\cdot\|$ denotes the usual ℓ^2-norm. In practice, it is just required to set the $\sqrt{n} \times \sqrt{n}$ patch size, the search space $\Delta(x)$ and the filtering parameter h. Buades *et al.* showed that this filter substantially outperforms the *bilateral filter* [34] and other iterative approaches [33].

Since, several accelerated versions of this filter have been proposed [26, 7]. In [4], Buades *et al.* recommended the *vectorial (or block-based) NL-means filter* defined as

$$NL_h \mathbf{z}(x) = \frac{1}{C(x)} \sum_{y \in \Delta(x)} e^{-\|\mathbf{z}(x)-\mathbf{z}(y)\|^2/h^2} \mathbf{z}(y), \;\; C(x) = \sum_{y \in \Delta(x)} e^{-\|\mathbf{z}(x)-\mathbf{z}(y)\|^2/h^2}, \quad (4)$$

which amounts to simultaneously restore pixels of a whole patch $\mathbf{z}(x)$ from nearby patches. The restored value at a pixel x is finally obtained by averaging the different estimators available at that position [4]. This filter can been considered as a fast implementation of the *NL-means filter*, especially if the blocks are picked up on a subsampled grid of pixels. In this paper, the proposed filter is inspired by this intuitive *vectorial NL-means filter* [4], but also by other recent accelerated versions [26, 7], as explained in the next sections. The related Bayesian framework enables to establish the relationships between these algorithms, to justify some underlying statistical assumptions and to give keys to set the control parameters of the *NL-means filter*. It is worth noting that this framework could be also used to remove noise in applications for which the noise distribution is assumed to be known and non-Gaussian.

3 A Bayesian risk formulation

In a probabilistic setting, the image denoising problem is usually solved in a discrete setting. The estimation problem is then to guess a n-dimensional patch $\mathbf{u}(x)$ from its noisy version $\mathbf{z}(x)$ observed at point x. Typically, the unknown vectorized image patch

$\mathbf{u}(x)$ is defined as $\mathbf{u}(x) := (u(x_k), x_k \in B(x)) \in \mathbb{R}^n$ where $B(x)$ defines the $\sqrt{n}\times\sqrt{n}$ square neighborhood of point x and the pixels in $\mathbf{u}(x)$ are ordered lexicographically. Let us suppose now that $\mathbf{u}(x)$ is unknown but we can observe $\mathbf{z}(x) = f(\mathbf{u}(x), \mathbf{v}(x))$ where $\mathbf{z}(x) := (z(x_k), x_k \in B(x))$, $\mathbf{v}(x)$ represents noise and $f(\cdot)$ is a linear or a non-linear function related to the image formation process. The noise $\mathbf{v}(x)$ is a random vector which components are iid, and $\mathbf{u}(x)$ is considered as a stochastic vector with some unknown probability distribution function (pdf).

Conditional mean estimator To compute the optimal Bayesian estimator for the vector $\mathbf{u}(x)$, it is necessary to define an appropriate loss function $L(\mathbf{u}(x), \widehat{\mathbf{u}}(x))$ which measures the loss associated with choosing an estimator $\widehat{\mathbf{u}}(x)$ when the true vector is $\mathbf{u}(x)$. The optimal estimator $\widehat{\mathbf{u}}_{opt}(x)$ is found by minimizing the posterior expected loss

$$\mathbb{E}[L(\mathbf{u}(x), \widehat{\mathbf{u}}(x))] = \sum_{\mathbf{u}(x)\in\Lambda} L(\mathbf{u}(x), \widehat{\mathbf{u}}(x))\, p(\mathbf{u}(x)|\mathbf{z}(x)),$$

taken with respect to the posterior distribution $p(\mathbf{u}(x)|\mathbf{z}(x))$ and Λ denotes the large space of all configurations of $\mathbf{u}(x)$ (e.g $|\Lambda| = 256^n$ if $u(x) \in \{0, \cdots, 255\}$). The loss function used in most cases is $L(\mathbf{u}(x), \widehat{\mathbf{u}}(x)) = 1 - \delta(\mathbf{u}(x), \widehat{\mathbf{u}}(x))$ where the δ function equals 1 if $\mathbf{u}(x) = \widehat{\mathbf{u}}(x)$ and 0 otherwise. Minimizing $\mathbb{E}[L(\mathbf{u}(x), \widehat{\mathbf{u}}(x))]$ is then equivalent to choose $\widehat{\mathbf{u}}_{opt}(x) = \arg\max p(\mathbf{u}(x)|\mathbf{z}(x))$, with the motivation that this should correspond to the most likely vector given the observation $\mathbf{z}(x)$. However, this loss function $L(\mathbf{u}(x), \widehat{\mathbf{u}}(x))$ may not be the most appropriate since it assigns 0 cost only to the perfect solution and unit cost to all other estimators. Another thought would be to use a cost function that depends on the number of pixels that are in error such as $L(\mathbf{u}(x), \widehat{\mathbf{u}}(x)) = \|\mathbf{u}(x) - \widehat{\mathbf{u}}(x)\|^2$. Assuming this quadratic loss function, the optimal Bayesian estimator is then

$$\widehat{\mathbf{u}}_{opt}(x) = \arg\min_{\widehat{\mathbf{u}}(x)} \sum_{\mathbf{u}(x)} \|\mathbf{u}(x) - \widehat{\mathbf{u}}(x)\|^2\, p(\mathbf{u}(x)|\mathbf{z}(x)) = \sum_{\mathbf{u}(x)} \mathbf{u}(x)\, p(\mathbf{u}(x)|\mathbf{z}(x)).$$

Referred as the conditional mean estimator, $\widehat{\mathbf{u}}_{opt}(x)$ can be also written as

$$\widehat{\mathbf{u}}_{opt}(x) = \sum_{\mathbf{u}(x)} \mathbf{u}(x) \frac{p(\mathbf{u}(x), \mathbf{z}(x))}{p(\mathbf{z}(x))} = \frac{\sum_{\mathbf{u}(x)} \mathbf{u}(x) p(\mathbf{z}(x)|\mathbf{u}(x)) p(\mathbf{u}(x))}{\sum_{\mathbf{u}(x)} p(\mathbf{z}(x)|\mathbf{u}(x)) p(\mathbf{u}(x))} \tag{5}$$

by using the Bayes' and marginalization rules, and $p(\mathbf{z}(x)|\mathbf{u}(x))$ and $p(\mathbf{u}(x))$ respectively denote the distribution of $\mathbf{z}(x)|\mathbf{u}(x)$ and the prior distribution of $\mathbf{u}(x)$.

Bayesian filter and image redundancy Ideally, we would like to know the pdfs $p(\mathbf{z}(x)|\mathbf{u}(x))$ and $p(\mathbf{u}(x))$ to compute $\widehat{\mathbf{u}}_{opt}(x)$ for each point x in the image from a large number of "repeated" observations (i.e. images). Unfortunately, we have only one image at our disposal, meaning that we have to adopt another way of estimating these pdfs. Due to the fact that the pdfs $p(\mathbf{z}(x)|\mathbf{u}(x))$ and $p(\mathbf{u}(x))$ cannot be obtained from a number of observations at the same point x, we choose to use the observations at a number of neighboring points taken in a semi-local neighborhood (or window) $\Delta(x)$. The window $\Delta(x)$ needs to be not too large since the remote samples are likely less

significant and can originate from other spatial contexts. We then assume that this set of nearby samples may be considered as a set of samples from $p(\mathbf{u}(x)|\mathbf{z}(x))$.

More formally, we suppose that $p(\mathbf{u}(x)|\mathbf{z}(x))$ is unknown, but we have a set $\{\mathbf{u}(x_1), \mathbf{u}(x_2), \cdots, \mathbf{u}(x_{N(x)})\}$ of $N(x)$ posterior samples taken in $\Delta(x)$. In what follows, $|\Delta(x)|$ is fixed for all the pixels but the size $N(x) \leq |\Delta(x)|$ is spatially varying since irrelevant and unlikely samples in $\Delta(x)$ are preliminarily discarded. From this set, we start by examining the prior distribution $p(\mathbf{u}(x))$. A first natural ideal would be to introduce MRFs and Gibbs distributions to capture interactions between pixels in the image patch, but the MRF framework involves the computationally intensive estimation of additional *hyperparameters* which must be likely adapted to each spatial position. Due to the huge domain space Λ, a computational alternative is then to set $p(\mathbf{u}(x))$ to uniform, i.e. $p(\mathbf{u}(x)) = 1/N(x)$. This means there is no preference to choose a vector $\mathbf{u}(x_i)$ in the set $\{\mathbf{u}(x_1), \cdots, \mathbf{u}(x_{N(x)})\}$ assumed to be composed of $N(x)$ preliminarily selected "similar" patches. Then, we have the following approximations (see [17]):

$$\frac{1}{N(x_i)} \sum_{j=1}^{N(x_i)} \mathbf{u}(x_j) p(\mathbf{z}(x_i)|\mathbf{u}(x_j)) \xrightarrow{P} \sum_{\mathbf{u}(x)} \mathbf{u}(x) p(\mathbf{z}(x)|\mathbf{u}(x)) p(\mathbf{u}(x)),$$

$$\frac{1}{N(x_i)} \sum_{j=1}^{N(x_i)} p(\mathbf{z}(x_i)|\mathbf{u}(x_j)) \xrightarrow{P} \sum_{\mathbf{u}(x)} p(\mathbf{z}(x)|\mathbf{u}(x)) p(\mathbf{u}(x)),$$

and we can propose a reasonable estimator $\widehat{\mathbf{u}}_N(x_i)$ for $\widehat{\mathbf{u}}_{opt}(x)$:

$$\widehat{\mathbf{u}}_N(x_i) = \frac{\frac{1}{N(x_i)} \sum_{j=1}^{N(x_i)} \mathbf{u}(x_j) p(\mathbf{z}(x_i)|\mathbf{u}(x_j))}{\frac{1}{N(x_i)} \sum_{j=1}^{N(x_i)} p(\mathbf{z}(x_i)|\mathbf{u}(x_j))}. \tag{6}$$

Nevertheless, we do not have the set $\{\mathbf{u}(x_1), \cdots, \mathbf{u}(x_{N(x)})\}$, but only a spatially varying dictionary $\mathcal{D}(x) = \{\mathbf{z}(x_1), \cdots, \mathbf{z}(x_{N(x)})\}$ composed of noisy versions is available. A way to solve this problem will be then to substitute $\mathbf{z}(x_j)$ to $\mathbf{u}(x_j)$ in (6) as shown in the next section. In a second step, this estimator will be refined by substituting the "aggregated" estimator computed from $\widehat{\mathbf{u}}_N(x_j)$ (see below) to $\mathbf{u}(x_j)$. Indeed, the restored patch at pixel x_j is a better approximation of $\mathbf{u}(x_j)$ than the noisy patch $\mathbf{z}(x_j)$ used as a *"pilot"* estimator, and the performance will be improved at the second iteration.

Aggregation of estimators The estimator (6) requires spatially sliding windows over the whole image for image reconstruction. Hence, a set of L (constant for uniform image sub-sampling) concurrent scalar values $\widehat{u}_{N,1}(x_i), \widehat{u}_{N,2}(x_i), \cdots, \widehat{u}_{N,L}(x_i)$ is calculated for the same pixel x_i due to the overlapping between patches. This set of competing estimators must be fused or *aggregated* into the one final estimator $\widetilde{u}(x_i)$ at pixel x_i. Actually, when no estimator is a "clear winner", one may prefer to combine the different estimators $\widehat{u}_{N,1}(x_i), \widehat{u}_{N,2}(x_i), \cdots, \widehat{u}_{N,L}(x_i)$ and a natural approach, well-grounded in statistics [6], is to use a convex or linear combination of estimators [19, 10]. Here, our *aggregate* estimator $\widetilde{u}_N(x_i)$ is simply defined as the average of competing estimators:

$$\widetilde{u}_N(x_i) = \frac{1}{L} \sum_{l=1}^{L} \widehat{u}_{N,l}(x_i). \tag{7}$$

In practice, patches are picked up on a sub-sampled grid (e.g. factor 3) of pixels to speed up the algorithmic procedure (e.g. factor 8), while preserving a good visual quality.

4 Bayesian NL-means filter

As explained before, to compute $\widehat{\mathbf{u}}_N(x_i)$, we first substitute $\mathbf{z}(x)$ to $\mathbf{u}(x)$ in (6). This yields the following estimator

$$\frac{\frac{1}{N(x_i)}\sum_{j=1}^{N(x_i)} p(\mathbf{z}(x_i)|\mathbf{z}(x_j))\mathbf{z}(x_j)}{\frac{1}{N(x_i)}\sum_{j=1}^{N(x_i)} p(\mathbf{z}(x_i)|\mathbf{z}(x_j))} \approx \widehat{\mathbf{u}}_N(x_i) \tag{8}$$

which can be computationally calculated provided the pdfs are known. It is worth noting that $p(\mathbf{z}(x_i)|\mathbf{z}(x_j))$ is not well defined if $x_i = x_j$ and it could be recommended to set $p(\mathbf{z}(x_i)|\mathbf{u}(x_i)) \approx \max_{x_j \neq x_i} p(\mathbf{z}(x_i)|\mathbf{z}(x_j))$ in (8) (see [22]). The central data point involved in the computation of its own average is then re-weighted to get the higher weight. Actually, the probability to detect an exact copy of $\mathbf{z}(x_i)$ corrupted by noise in the neighborhood tends to 0 because the space of $\sqrt{n} \times \sqrt{n}$ patches is huge and, to be consistent, it is necessary to limit the influence of the central patch.

In the remainder of this section, we shall now consider the usual image model

$$\mathbf{z}(x) = \mathbf{u}(x) + \mathbf{v}(x) \tag{9}$$

where $\mathbf{v}(x)$ is an additive white Gaussian noise with variance σ^2. We will further assume that the likelihood can be factorized as $p(\mathbf{z}(x_i)|\mathbf{z}(x_j)) = \prod_{k=1}^{n} p(z(x_{i,k})|z(x_{j,k}))$ with $x_{i,k} \in B(x_i)$ and $x_{j,k} \in B(x_j)$. It follows that $\mathbf{z}(x_i)|\mathbf{z}(x_j)$ follows a multivariate normal distribution, $\mathbf{z}(x_i)|\mathbf{z}(x_j) \sim \mathcal{N}(\mathbf{z}(x_j), \sigma^2\mathbf{I}_n)$ where $\mathbf{I}_n$ is the n-dimensional identity matrix. From (8), the filter adapted to white Gaussian noise is then given by

$$\frac{1}{C(x_i)}\sum_{j=1}^{N(x_i)} e^{-\|\mathbf{z}(x_i)-\mathbf{z}(x_j)\|^2/(2\sigma^2)}\mathbf{z}(x_j) \;\text{ with }\; C(x_i) = \sum_{j=1}^{N(x_i)} e^{-\|\mathbf{z}(x_i)-\mathbf{z}(x_j)\|^2/(2\sigma^2)}. \tag{10}$$

If we arbitrarily set $N(x_i) = N$ to a constant value and $h^2 = 2\sigma^2$, this filter is nothing else than the *vectorial NL-means filter* given in (4). However, it is recommended to set $h \approx 12\sigma$ in [4] to produce satisfying denoising results. In our experiments, it is also confirmed that $h \approx 5\sigma$ is good choice if we use (3) and (4) for denoising. Actually, the filtering parameter h is actually set to a higher value than the expected value $\sqrt{2}\sigma$ in practical imaging. In the next sections, we shall see how this parameter can be better interpreted and theoretically estimated.

Spatially adaptive dictionaries The filter (10) can be refined if the adaptive dictionary $\mathcal{D}(x_i) = \{\mathbf{z}(x_1), \cdots, \mathbf{z}(x_{N(x_i)})\}$ around x_i is reliably obtained using an *off-line* procedure. Since $\mathcal{D}(x_i)$ is assumed to be composed of noisy versions of the more likely samples from the posterior distribution $p(\mathbf{u}(x_i)|\mathbf{z}(x_i))$, the irrelevant image patches in $\Delta(x_i)$ must be discarded in a preliminary step. Consequently, the size $N(x_i)$ is adaptive according to local spatial contexts and a simple way to detect these unwanted samples can be based on local statistical features between images patches. In our experiments, we have consider two basic features, that is the mean $m(\mathbf{z}(x)) = n^{-1}\sum_{k=1}^{n} z(x_k)$ and the variance $\text{var}(\mathbf{z}(x)) = n^{-1}\sum_{k=1}^{n}(z(x_k) - m(\mathbf{z}(x)))^2$ of a vectorized patch $\mathbf{z}(x) := (z(x_k), x_k \in B(x))$.

Intuitively, $\mathbf{z}(x_j)$ will be discarded from the local dictionary $\mathcal{D}(x_i)$ if the mean $m(\mathbf{z}(x_j))$ is too 'far" from the mean $m(\mathbf{z}(x_i))$ when $\mathbf{z}(x_j)$ and $\mathbf{z}(x_i)$ are compared. More formally, if $|m(\mathbf{z}(x_j)) - m(\mathbf{z}(x_i))| > \lambda_\alpha \sigma/\sqrt{n}$, where $\lambda_\alpha \in \mathbb{R}_+$ is chosen as a quantile of the standard normal distribution, the hypothesis that the two patches belong to the same "population" is rejected. Hence, setting $\lambda_\alpha = 3$ given $\mathbb{P}(|m(\mathbf{z}(x_j)) - m(\mathbf{z}(x_i))| \leq \lambda_\alpha \sigma/\sqrt{n}) = 1 - \alpha$, yields to $\alpha = 2(1 - \Phi(\lambda_\alpha/\sqrt{2})) = 0.034$ where Φ means the Laplace distribution.

Similarly, the variance $\text{var}(\mathbf{z}(x_j))$ is expected to be close to the variance $\text{var}(\mathbf{z}(x_i))$ for the central patch $\mathbf{z}(x_i)$. A F-test[4] is used and the ratio $F = \frac{\max(\text{var}(\mathbf{z}(x_j)),\text{var}(\mathbf{z}(x_i)))}{\min(\text{var}(\mathbf{z}(x_j)),\text{var}(\mathbf{z}(x_i)))}$ is compared to a threshold $T_{\beta,n-1}$ to determine if the value falls within the zone of acceptance of the hypothesis that the variances are equal. The threshold $T_{\beta,n-1}$ is the critical value for the F-distribution with $n-1$ degrees of freedom for each patch and a significance level β. Typically, when 7×7 patches are compared, we have $\mathbb{P}(F > T_{0.05,48} = 1.6) = 0.05$. If the ratio F exceeds the value $T_{\beta,n-1}$, the sample $\mathbf{z}(x_j)$ is discarded from the local dictionary $\mathcal{D}(x_i)$.

This formal description is related to the intuitive approach proposed in [26, 7, 16] to improve the performance of the *NL-means filter*.

New statistical distance measure for patch comparison In this section, we propose to revise the distance used for patch comparison, yielding to a *NL-means filter* which is better parametrized. In (3) and (4), it is implicitly assumed that $\mathbf{z}(x_i)|\mathbf{z}(x_j) \sim \mathcal{N}(\mathbf{z}(x_j), \frac{1}{2}h^2\mathbf{I}_n)$. Actually, this hypothesis is valid only for non-overlapping and statistically independent patches, but most of patches overlapped in $\Delta(x)$ since $\Delta(x)$ is not so large (e.g 21×21 pixels). At the opposite, if $\mathbf{z}(x_j)$ is horizontally (or vertically) shifted by one pixel from the location of $\mathbf{z}(x_j)$, $\mathbf{z}(x_i)|\mathbf{z}(x_j)$ is expected to follow a multivariate Laplacian distribution. However, this statistical hypothesis does not hold true for arbitrary locations of overlapping patches in $\Delta(x)$. The adjustment of the decay parameter $h \approx 5\sigma$ in (3) to a value higher that the expected value $\sqrt{2}\sigma$ is probably related to the fact that the two compared patches are not independent. Note that some pixel values are in common in the two vectors but at different locations.

Hence, $p(\mathbf{z}(x_i)|\mathbf{z}(x_j))$ must be carefully examined and we propose the following definition for the likelihood: $p(\mathbf{z}(x_i)|\mathbf{z}(x_j)) \propto e^{-\phi(\|\mathbf{z}(x_i)-\mathbf{z}(x_j)\|)}$. Typically, we can choose $\phi(t) = t^2$ or $\phi(t) = |t|$ (or a scaled version) to compare patches. Here, we examine the distribution of $\|\mathbf{z}(x_i) - \mathbf{z}(x_j)\|$ from the local dictionary $\mathcal{D}(x_i)$ to determine ϕ. First, it can be observed that $\mathbb{E}[\|\mathbf{z}(x_i) - \mathbf{z}(x_j)\|]$ is non-zero in most cases and the probability to find an exact copy of $\mathbf{z}(x_i)$ in $\Delta(x)$ tends to 0, especially if $\Delta(x)$ is large. The maximum of the assumed zero-mean multivariate Gaussian distribution in (3) should be then "shifted" to $\mathbb{E}[\|\mathbf{z}(x_i)-\mathbf{z}(x_j)\|]$. However, this training step could be hard in practice since it must adapted to each spatial position, and we propose instead to use asymptotic results. Actually, we have already assumed $(z(x_{i,k})-z(x_{j,k})) \sim \mathcal{N}(0, 2\sigma^2)$ when two pixels taken in $\mathbf{z}(x_i)$ and $\mathbf{z}(x_j) \in \mathcal{D}(x_i)$ are compared. Hence, the normalized distance $\text{dist}(\mathbf{z}(x_i), \mathbf{z}(x_j)) = \|\mathbf{z}(x_i) - \mathbf{z}(x_j)\|^2/(2\sigma^2)$ follows a chi-square χ^2_n distribution with n degrees of freedom. For n large ($n \geq 25$), it can be proved that $\sqrt{2\,\text{dist}(\mathbf{z}(x_i), \mathbf{z}(x_j))}$ is approximately normally distributed with mean $\sqrt{2n-1}$ and

[4] The F-distribution is used to compare the variance of two independent samples from a normally distributed population.

unit variance:

$$p\left(\sqrt{2\,\mathrm{dist}(\mathbf{z}(x_i),\mathbf{z}(x_j))}\right) \propto \exp -\frac{1}{2}\left(\sqrt{2\,\mathrm{dist}(\mathbf{z}(x_i),\mathbf{z}(x_j))}-\sqrt{2n-1}\right)^2 \tag{11}$$
$$\propto \exp -\left(\frac{\|\mathbf{z}(x_i)-\mathbf{z}(x_j)\|^2}{2\sigma^2}-\frac{\|\mathbf{z}(x_i)-\mathbf{z}(x_j)\|}{(\sigma/\sqrt{2n-1})}+\frac{2n-1}{2}\right).$$

Accordingly, we define the likelihood as $p(\mathbf{z}(x_i)|\mathbf{z}(x_j)) \propto \exp -\phi(\|\mathbf{z}(x_i)-\mathbf{z}(x_j)\|)$ and choose $\phi(t) = at^2 - b|t| + c$ with $a = 1/(2\sigma^2)$, $b = \sqrt{2n-1}/\sigma$ and $c = (2n-1)/2$ depending only on the patch size n and the noise variance σ^2. From our experiments, it was confirmed that no additional adjustment parameter is necessary provided the noise variance is robustly estimated and the performance is maximum for the true noise variance as expected. Figure 1 (bottom-right) shows the functions e^{-t^2/h^2} and $e^{-(|t|/\sigma-\sqrt{2n-1})^2/2}$ by setting $n = 49$, $\sigma^2 = 1$ and $h = 5\sigma$, and then illustrates how data points are currently weighted when the original *NL-means filter* and the so-called *Adaptive NL-means filter* (*ANL*) defined as

$$ANL_{\sigma,n}\mathbf{z}(x_i) = \frac{\displaystyle\sum_{j=1}^{N(x_i)} \exp -\frac{1}{2}\left(\frac{\|\mathbf{z}(x_i)-\mathbf{z}(x_j)\|}{\sigma}-\sqrt{2n-1}\right)^2 \mathbf{z}(x_j)}{\displaystyle\sum_{j=1}^{N(x_i)} \exp -\frac{1}{2}\left(\frac{\|\mathbf{z}(x_i)-\mathbf{z}(x_j)\|}{\sigma}-\sqrt{2n-1}\right)^2}. \tag{12}$$

where $N(x_i) = \#\{\mathcal{D}(x_i)\}$, are applied. Note that the data at point x_i should participate significantly to the weighted average. Accordingly, since $p(\mathbf{z}(x_i)|\mathbf{z}(x_j))$ tends to 0 when $x_i = x_j$ in (12), we arbitrarily decide to set $p(\mathbf{z}(x_i)|\mathbf{z}(x_i)) \approx \max_{x_j \neq x_i} p(\mathbf{z}(x_i)|\mathbf{z}(x_j))$ as explained before (see also [22]).

Bayesian NL-means filter and plugin estimator In the previous sections, $\mathbf{z}(x_j)$ was substituted to $\mathbf{u}(x_j)$ in (6) to give (8) and further (12). Now, we are free to substitute the vector $\widetilde{\mathbf{u}}_{ANL}(x_j)$ of aggregated estimators (computed from the set of restored blocks $\{ANL_{\sigma,n}\mathbf{z}(x_j)\}$, see (7)) to $\mathbf{u}(x_j)$. This *plugin ANL* estimator defined as

$$ANL_{\sigma,n}\widetilde{\mathbf{u}}_{ANL}(x_i) = \frac{\displaystyle\sum_{j=1}^{N(x_i)} \exp -\frac{1}{2}\left(\frac{2\,\|\mathbf{z}(x_i)-\widetilde{\mathbf{u}}_{ANL}(x_j)\|}{\sigma}-\sqrt{2n-1}\right)^2 \widetilde{\mathbf{u}}_{ANL}(x_j)}{\displaystyle\sum_{j=1}^{N(x_i)} \exp -\frac{1}{2}\left(\frac{2\,\|\mathbf{z}(x_i)-\widetilde{\mathbf{u}}_{ANL}(x_j)\|}{\sigma}-\sqrt{2n-1}\right)^2} \tag{13}$$

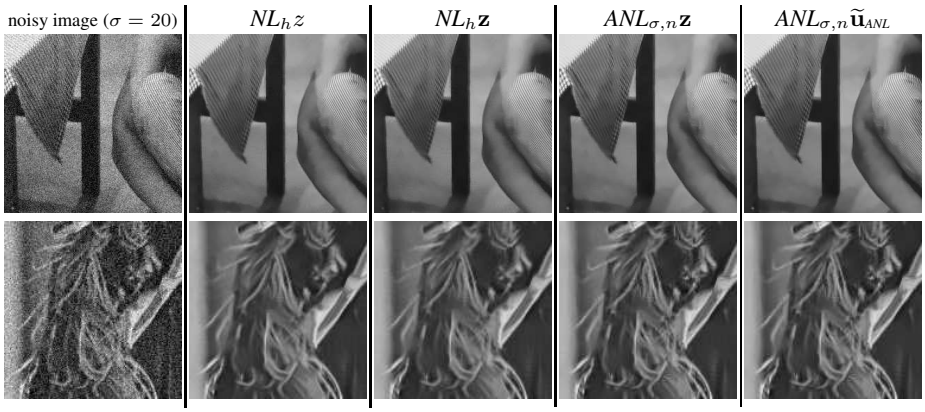
is expected to improve the restored image since $\widetilde{\mathbf{u}}_{ANL}(x_j)$ is a better approximation of $\mathbf{u}(x_j)$ than $\mathbf{z}(x_j)$. In (13) the restored image is recycled but the weights is a re-scaled function (theoretically by a factor 2) of the distance between the *"pilot"* estimator $\widetilde{\mathbf{u}}_{ANL}(x_j)$ given by (12) and the input vector $\mathbf{z}(x_i)$. The estimators are finally aggregated to produce the final restored image (see (7)).

5 Experiments

In this section, we evaluate the performance of different versions of the Bayesian filter and the original *NL-means filter* using the peak signal-to-noise ratio (PSNR in db) defined as PSNR $= 10\log_{10}(255^2/\text{MSE})$ with MSE $= |\Omega|^{-1}\sum_{x\in\Omega}(z_0(x)-\widehat{u}(x))^2$ where z_0 is the noise-free original image. We also use the *"method noise"* described in [4] which consists in computing the difference between the input noisy image and its denoised version. The *NL-means filter* (see (3)) was applied with $h = 5\sigma$ and our experiments have been performed with 7×7 patches and 15×15 search windows, corresponding to the best visual results and the best PSNR values. For all the presented results, we set $T_{0.05,n-1} = 1.6$ and $\lambda_{0.034} = 3$ to build spatially adaptive dictionaries.

The potential of the estimation method is mainly illustrated with the 512×512 *Lena* and *Barbara* images corrupted by an additive white-Gaussian noise (WGN) (PSNR = 22.13 db, $\sigma = 20$). We compared the original *NL-means* algorithm with the proposed modifications and Fig. 1 shows the denoising results using several filters ($n = 7\times 7$ and $N = 15\times 15$): *i)* the *NL-means filter* $NL_h z$ when the similarity is only measured by the Euclidean distance (see (3)); *ii*) the *vectorial NL-means filter* $NL_h\mathbf{z}$ with sliding blocks but no spatial sub-sampling (see (4)); *iii)* our *Adaptive NL-means filter* $ANL_{\sigma,n}\mathbf{z}$ which includes adaptive dictionaries and a similarity measure based on the new distance (see (12)); *iv)* the plugin *Adaptive NL-means filter* $ANL_{\sigma,n}\widetilde{\mathbf{u}}_{ANL}$ (see (13)). In most cases, the PSNR values are slightly affected if a spatial sub-sampling (by a factor 3) is used in the implementation, but the time computing is drastically reduced (speed is 8 times less than before): the implementation of the *fast Adaptive NL-means filter* took 10 sec on a single-CPU PC 2.0 Ghz running linux, and the full *Adaptive NL-means filter* (with no spatial sub-sampling) took 75 sec for denoising a 512×512 image (see table in Fig. 1). In these practical examples, the use of spatially adaptive dictionaries enables to enhance contrasts. Note that the residual noise in flat zones is more reduced with no additional blur, when $ANL_{\sigma,n}\widetilde{\mathbf{u}}_{ANL}$ is applied. In Fig. 2, we modified the estimation of noise variance to assess the sensitivity of this parameter on filtering results. In general, the PSNR value is maximum with the true value (e.g. $\sigma = 20$ in Fig. 2) but decreases if this value is under-estimated or over-estimated. In Fig. 2, the estimated noise component is similar to a simulated white Gaussian noise but contains few geometric structures if we under-estimate or over-estimate the noise variance. To evaluate the performance of those filters, we reported the PSNR values for different versions of the *NL-means filter*. In table 1, the numerical results are improved using our filter, with performance very close to competitive patch-based denoising methods. Note that the best results (PSNR values) were recently obtained by filtering in 3D transform domain and combining sliding-window transform processing with block-matching [10].

In the second part of experiments, we have applied $ANL_{\sigma,n}\widetilde{\mathbf{u}}_{ANL}$ to restore an old photograph (Fig. 3 - left column); in that case, the noise variance is estimated from image data (see [21]). Nevertheless, in real digital imaging, images are better described by the following model $z(x) = u(x)+u^{\gamma}(x)\,\varepsilon(x)$ where the sensor noise $u^{\gamma}(x)\,\varepsilon(x)$ is defined as a function of $u(x)$ and $\varepsilon(x)$ is a zero-mean Gaussian noise of variance σ^2. Accordingly, the noise in bright regions is higher than in dark regions (see Fig. 2 - second column). From experiments on real digital images [14], it was confirmed that $\gamma \approx 0.5$ ($\gamma = 0$ corresponds to WGN in the previous experiments). Accordingly, we modify the



	Timings	*Lena* 512×512	*Barbara* 512×512
$NL_h z$	58.1 sec	31.85 db	30.27 db
$NL_h \mathbf{z}$	96.2 sec	32.04 db	30.49 db
$ANL_{\sigma,n}\mathbf{z}$	75.2 sec	32.51 db	30.79 db
$ANL_{\sigma,n}\widetilde{\mathbf{u}}_{ANL}$	173.3 sec	**32.63** db	**30.88** db
Fast $ANL_{\sigma,n}\mathbf{z}$	10.6 sec	32.36 db	30.61 db
Fast $ANL_{\sigma,n}\widetilde{\mathbf{u}}_{ANL}$	21.2 sec	32.49 db	30.71 db

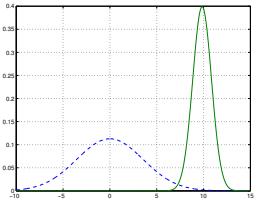


Fig. 1. Comparisons of different *NL-means filters*. From top to bottom and from left to right: part of noisy images (*Barbara, Lena*, $\sigma = 20$), *NL-means filter*, *vectorial NL-means filter*, *Adaptive NL-means filter*, plugin *Adaptive NL-means filter*; numerical results for each filter; comparison of exponential weights for the original *NL-means filter* (blue dashed line) and for the *Adaptive NL-means filter* (solid green line) (see text).

normalized distance as follows: $\text{dist}(\mathbf{z}(x_i), \mathbf{z}(x_j)) = \|\mathbf{z}(x_i) - \mathbf{z}(x_j)\|^2/(2\sigma^2\mathbf{z}(x_j))$ with $\sigma \in [1.5, 3]$. Moreover, this model $z(x) = u(x) + \sqrt{u(x)}\,\varepsilon(x)$ has already been considered to denoise log-compressed ultrasonic images [24]. Preliminary results of $ANL_{\sigma,n}\widetilde{\mathbf{u}}_{ANL}$ with this model is shown in Fig. 3, when applied to two log-compressed ultrasonic images and a cryo-Electronic Microscopy image (cross-section of a microtubule (10-30 nm)) where brights areas are smoother than dark areas.

6 Conclusion

We have described a Bayesian motivation for the *NL-means filter* and justify some intuitive adaptations described in previous papers [4, 26, 7]. The proposed framework yields to a filter which is better parametrized: the size of the adaptive dictionary and the noise variance are computed from the input image, and the patch size must be large enough. Our Bayesian approach has been used to remove image-dependent noise and could be adapted in applications with appropriate noise distributions. A more thoroughly evaluation with other methods [1, 2, 35, 16], and recent improvements of the *NL-means filter* described in [5], would be also desirable.

Acknowledgments. We thank B. Delyon and P. Pérez for fruitful conversations and comments.

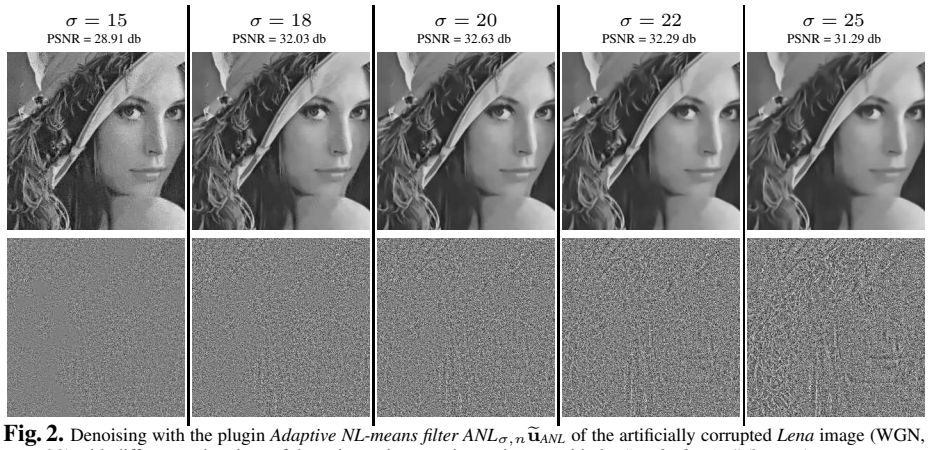


Fig. 2. Denoising with the plugin *Adaptive NL-means filter* $ANL_{\sigma,n}\widetilde{\mathbf{u}}_{ANL}$ of the artificially corrupted *Lena* image (WGN, $\sigma = 20$) with different estimations of the noise variance and experiments with the *"method noise"* (bottom).

Image σ/PSNR	*Lena* 20/22.13	*Barbara* 20/22.18	*Boats* 20/22.17	*House* 20/22.11	*Peppers* 20/22.19
$ANL_{\sigma,n}\widetilde{\mathbf{u}}_{ANL}$	**32.63**	**30.88**	**30.16**	**33.24**	**30.75**
NL-means filter	31.85	30.27	29.42	32.24	29.86
Dabov *et al.* [10]	33.03	31.77	30.65	33.54	30.87
Elad *et al.* [13]	32.38	30.83	30.36	33.20	30.82
Kervrann *et al.* [21]	32.64	30.37	30.12	32.90	30.59
Portilla *et al.* [31]	32.66	30.32	30.38	32.39	30.31
Roth *et al.* [32]	31.92	28.32	29.85	32.17	30.58
Rudin *et al.* [33]	30.48	27.07	29.02	31.03	28.51

σ/PSNR	*Lena* 512^2	*Barbara* 512^2	*Boats* 512^2	*House* 256^2	*Peppers* 256^2
5 / 34.15	37.98	36.93	36.39	38.89	37.13
10 / 28.13	35.25	33.82	33.18	35.67	33.87
15 / 24.61	33.68	32.21	31.45	34.23	32.06
20 / 22.11	32.63	30.88	30.16	33.24	30.75
25 / 20.17	31.55	29.77	29.11	32.30	29.77
50 / 14.15	27.51	24.91	25.13	27.64	23.84

Table 1. left: performance of different methods when applied to test noisy (WGN) images (*NL-means filter* $NL_h z$ is implemented as in (3) and the maximum weighting $w(x,y)$ at $x = y$ is given by $w(x,x) = max_{x \neq y} w(x,y)$) ; right: performance of the plugin *Adaptive NL-means filter* $ANL_{\sigma,n}\widetilde{\mathbf{u}}_{ANL}$ for different signal-to-noise ratios (WGN).

References

1. Awate, S.P., Whitaker, R.T.: Higher-order image statistics for unsupervised, information-theoretic, adaptive, image filtering. CVPR'05, San Diego (2005)
2. Azzabou, N., Paragios, N., Guichard, F.: Random walks, constrained multiple hypothesis and image enhancement. ECCV'06, Graz (2006)
3. Barash, D., Comaniciu, D.: A Common framework for nonlinear diffusion, adaptive smoothing, bilateral filtering and mean shift. Image Vis. Comp. **22** (2004) 73-81
4. Buades, A., Coll, B., Morel, J.M.: A review of image denoising algorithms, with a new one. Multiscale Modeling and Simulation. **4** (2005) 490-530
5. Buades, A., Coll, B., Morel, J.M.: The staircasing effect in neighborhood filters and its solution. IEEE T. Image Process. **15** (2006)
6. Bunea, F., Nobel, A.B.: Sequential procedures for aggregating arbitrary estimators of a conditional mean *(under revision)* (2005)
7. Coupé P., Yger, P. Barillot, C.: Fast non-local means denoising for 3D MR images. MICCAI'06, Copenhagen (2006)
8. Criminisi, A., Pérez, P., Toyama, K.: Region filling and object removal by exemplar-based inpainting. IEEE T. Image Process. **13** (2004) 1200-1212
9. De Bonet, J.S.: Noise reduction through detection of signal redundancy. Rethinking Artificial Intelligence, MIT AI Lab (1997)
10. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising with block-matching and 3D filtering. In Electronic Imaging'06, Proc. SPIE 6064, no. 6064A-30, San Jose, California USA, 2006.
11. Efros, A., Leung, T.: Texture synthesis by non-parametric sampling. ICCV'99, Kerkyra (1999)
12. Elad, M.: On the bilateral filter and ways to improve it. IEEE T. Image Process. **11** (2002) 1141-1151
13. Elad, M. Aharon, M.: Image denoising via learned dictionaries and sparse representation. CVPR'06, New-York (2006)
14. Faraji, H. and MacLean, J.W.: CCD noise removal in digital images. IEEE T. Image Process. **15**(9) (2006) 2676-2685
15. Geman, D., Geman, S., Graffigne, C., Dong, P: Boundary detection by constrained optimization. IEEE T. Patt. Anal. Mach. Intell. **12** (1990), 609-628
16. Gilboa, G., Osher, S.: Nonlocal linear image regularization and supervised segmentation. UCLA CAM Report 06-47 (2006)

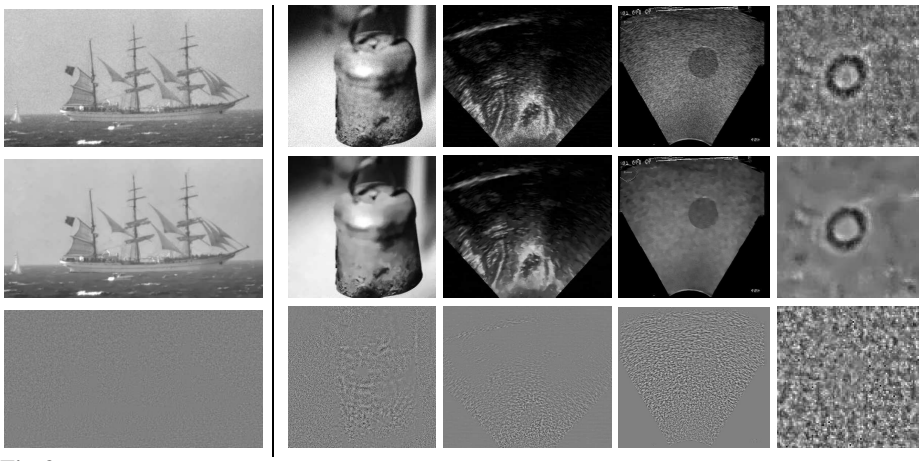


Fig. 3. Denoising of real noisy images. From top to bottom: original images, denoised images, *"method noise"* experiments. The old photograph (left) has been denoised using an usual additive noise and the four other images (right) are denoised using the image-dependent noise model ($\gamma = 0.5$, see text).

17. Godtliebsen, F., Spjotvoll, E., Marron, J.S.: A nonlinear Gaussian filter applied to images with discontinuities. J. Nonparametric Stat. **8** (1997) 21-43
18. Hirakawa, K., Parks, T.W.: Image denoising using total least squares. IEEE T. Image Process. **15**(9) (2006) 2730-2742
19. Katkovnik, V., Egiazarian, K., Astola, J.: Adaptive window size image denoising based on intersection of confidence intervals (ICI) rule. J. Math. Imag. Vis. **16** (2002) 223-235
20. Kervrann, C., Heitz, F.: A Markov random field model-based approach to unsupervised texture segmentation using local and global spatial statistics. IEEE T. Image Process. **4** (1995) 856-862
21. Kervrann, C., Boulanger, J. Unsupervised patch-based image regularization and representation. ECCV'06, Graz (2006)
22. Kinderman, S., Osher, S. Jones, P.W.: Deblurring and denoising of images by nonlocal functionals. Multiscale Modeling and Simulation. **4** (2005) 1091-1115
23. Lee, J.S.: Digital image smoothing and the sigma filter. Comp. Vis. Graph. Image Process. **24** (1983) 255-269
24. Loupas, T., McDicken, W.N., Allan, P.L.: An adaptive weighted median filter for speckle suppression in medical ultrasonic images. IEEE T. Circ. Syst. **36** (1989) 129-135
25. Lukin, A.: A multiresolution approach for improving quality of image denoising algorithms. ICASSP'06, Toulouse (2006)
26. Mahmoudi, M.; Sapiro, G.: Fast image and video denoising via nonlocal means of similar neighborhoods. IEEE Signal Processing Letters. **12** (2205) 839-842
27. Mrazek, P., Weickert, J., Bruhn, A.: On robust estimation and smoothing with spatial and tonal kernels. Preprint 51, U. Bremen (2004)
28. Mumford, D., Shah, J.: Optimal approximations by piecewise smooth functions and variational problems. Comm. Pure and Appl. Math. **42** (1989) 577-685
29. Perona. P., Malik, J.: Scale space and edge detection using anisotropic diffusion. IEEE T. Patt. Anal. Mach. Intell. **12** (1990) 629-239
30. Polzehl, J., Spokoiny, V.: Adaptive weights smoothing with application to image restoration. J. Roy. Stat. Soc. B **62** (2000) 335-354
31. Portilla, J., Strela, V., Wainwright, M., Simoncelli, E.: Image denoising using scale mixtures of Gaussians in the wavelet domain. IEEE T. Image Process. **12** (2003) 1338-1351
32. Roth, S., Black, M.J.: Fields of experts: a framework for learning image priors with applications. CVPR'05, San Diego (2005)
33. Rudin, L., Osher, S., Fatemi, E.: Nonlinear Total Variation based noise removal algorithms. Physica D (2992) **60** (1992) 259-268
34. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. ICCV'98, Bombay (1998)
35. Tschumperlé, D.: Curvature-preserving regularization of multi-valued images using PDE's. ECCV'06, Graz (2006)
36. Wang, Z., Zhang, D.: Restoration of impulse noise corrupted images using long-range correlation. IEEE Signal Processing Letters. **5** (1998) 4-6
37. Weickert, J.: Anisotropic Diffusion in Image Processing. Teubner-Verlag, Stuttgart (1998)
38. van de Weijer, J., van den Boomgaard, R.: Local mode filtering. CVPR'01, Kauai (2001)
39. Zhang, D., Wang, Z.: Image information restoration based on long-range correlation. IEEE T. Circ. Syst. Video Technol. **12** (2002) 331-341