Secrets of image denoising cuisine

M. Lebrun M. Colom A. Buades J.M. Morel

M.L. and J.M.M.: CMLA, Ecole Normale Supérieure de Cachan, 61 avenue du Président Wilson 94235 Cachan cedex, France
M.C. and A. B. : Universitat de les Illes Balears, Crta. de Valldemossa, km 7.5, 07122 Palma de Mallorca, Spain

Abstract

Digital images are matrices of regularly spaced pixels, each containing a photon count. This photon count is a stochastic process due to the quantic nature of light. It follows that all images are noisy. Ever since digital images exist, numerical methods have been proposed to improve the signal to noise ratio. Such "denoising" methods require a noise model and an image model. It is relatively easy to obtain a noise model. As will be explained in the present paper, it is even possible to estimate it from a single noisy image.

Obtaining a convincing statistical image model is quite another story. Images reflect the world and are as complex as the world. Thus, any progress in image denoising signals a progress in our understanding of image statistics. The present paper contains an analysis of nine recent state of the art methods. This analysis shows that we are probably close to understanding digital images at a "patch" scale. Recent denoising methods use thorough non parametric estimation processes for 8×8 patches, and obtain surprisingly good denoising results.

The mathematical and experimental evidence of two recent articles suggests that we might even be close to the best attainable performance in image denoising ever. This suspicion is supported by a remarkable convergence of all analyzed methods. They certainly converge in performance. We intend to demonstrate that, under different formalisms, their methods are almost equivalent. Working in the 64-dimensional "patch space", all recent methods estimate local "sparse models" and restore a noisy patch by finding its likeliest interpretation knowing the noiseless patches.

The story will be told in an ordinate manner. Denoising methods are complex and have several indispensable ingredients. Noise model and noise estimation methods will be explained first. The four main image models used for denoising: the Markovian-Bayesian paradigm, the linear transform thresholding, the so-called image sparsity, and an image self-similarity hypothesis will be presented in continuation. The performance of all methods depends on three generic tools: colour transform, aggregation, and an "oracle" step. Their recipes will also be given. These preparations will permit to present, in a unified terminology, the complete recipes of nine different state of the art patch-based denoising methods. Three quality assessment recipes for denoising methods will also be proposed and applied to compare all methods. The paper presents an ephemeral state of the art in a burgeoning subject, but many of the presented recipes will remain useful. Most denoising recipes can be tested directly on any digital image at *Image Processing On Line*, `http://www.ipol.im/`.

Contents

Notation

- **i**, **j**, **r**, **s** image pixels
- $u(\mathbf{i})$ image value at **i**, denoted by $U(\mathbf{i})$ when the image is handled as a vector
- $\tilde{u}(\mathbf{i})$ noisy image value at **i**, written $\tilde{U}(\mathbf{i})$ when the image is handled as a vector
- $\hat{u}(\mathbf{i})$ restored image value, $\hat{U}(\mathbf{i})$ when the image is handled as a vector
- $n(\mathbf{i})$ noise at **i**
- N patch of noise in vector form
- m number of pixels **j** involved to denoise a pixel **i**
- P reference patch, Q, other patch compared to P
- $\tilde{P}$, $\tilde{Q}$ noisy patches
- $\hat{P}$ restored patch
- $w(\tilde{P},\tilde{Q}) = e^{-\frac{d^2(\tilde{P},\tilde{Q})}{C\sigma^2}}$ interaction weight between P and Q
- $d(\tilde{P},\tilde{Q})$ Euclidean distance between patches (considered as vectors of their values)
- σ standard deviation of white noise at each pixel
- $\kappa \times \kappa$: dimension of patches.
- $\lambda \times \lambda$: dimension of research zone in which similar patches are searched
- $\mathcal{N}(\mu, \mathbf{C})$ vectorial Gaussian distribution with mean vector μ and covariance matrix $\mathbf{C}$
- $\mathbb{P}(G)$ probability of an event G (in the image and noise stochastic models)
- $\underline{E}Q$: expectation (of a random patch Q)
- $\overline{P}$ empirical expectation of the patches similar to P
- Δ image Laplace operator (sum of the second derivatives in two orthogonal directions)
- $DCT(n_1, n_2)$ 2D discrete cosine transform at frequencies n_1, n_2
- p percentile value of a histogram (between 0% and 100%)
- $w \times w$ block size for estimating the noise
- $\hat{\sigma}$ estimated value of the noise
- b number of bins
- h result of a high-pass filter on $\tilde{u}$
- ∇ gradient (of an image)
- M total number of pixels or patches in the image, total number of patches in the patch space
- $\mathbf{C}_P$ covariance matrix (of patches similar to P or $\tilde{P}$)
- P_1 restored patch at the first application of a denoising algorithm
- P_2 restored patch at the second application of a denoising algorithm
- $\mathcal{B} = \{G_i\}_{i=1}^{M}$ orthonormal basis of $\mathbb{R}^M$
- $\mathbf{D}$ diagonal linear operator. Dictionary of patches considered as a matrix
- n_{dic} size of the dictionary (number of patches in it)
- $\mathbf{A}$ linear operator, applied to the image u encoded as a vector U
- $p(P)$ density function of patches
- K number of patch clusters. Number of wavelet coefficients
- k index in K
- Ω_k patch cluster
- i index of patch P_i (in a patch cluster, in the image)

1 Introduction

Most digital images and movies are currently obtained by a CCD device. The value $\tilde{u}(\mathbf{i})$ observed by a sensor at each pixel $\mathbf{i}$ is a Poisson random variable whose mean $u(\mathbf{i})$ would be the ideal image. The difference between the observed image and the ideal image $\tilde{u}(\mathbf{i}) - u(\mathbf{i}) = n(\mathbf{i})$ is called "shot noise". The standard deviation of the Poisson variable $\tilde{u}(\mathbf{i})$ is equal to the square root of the number of incoming photons $\tilde{u}(\mathbf{i})$ in the pixel captor $\mathbf{i}$ during the exposure time. The Poisson noise n adds up to a thermal noise and to an electronic noise which are approximately additive and white. On a motionless scene with constant lighting, $u(\mathbf{i})$ can be approached by simply accumulating photons for a long exposure time, and by taking the temporal average of this photon count, as illustrated in figure 1.

Accumulating photon impacts on a surface is therefore the essence of photography. The first Nicéphore Niépce photograph [34] was obtained after an eight hours exposure. The problem of a long exposure is the variation of the scene due to changes in light, camera motion, and incidental motions of parts of the scene. The more these variations can be compensated, the longer the exposure can be, and the more the noise can be reduced. If a camera is set to a long exposure time, the photograph risks motion blur. If it is taken with short exposure, the image is dark, and enhancing it reveals the noise.

A recently available solution is to take a burst of images, each with short-exposure time, and to average them after registration. This technique, illustrated in Fig. 1, was evaluated recently in a paper that proposes fusing bursts of images taken by cameras [28]. This paper shows that the noise reduction by this method is almost perfect: fusing m images reduces the noise by a $\sqrt{m}$ factor.

It is not always possible to accumulate photons. There are obstacles to this accumulation in astronomy, biological imaging and medical imaging. In day to day images, the scene is moving, which limits the exposure time. The main limitations to any imaging system are therefore the noise and the blur. In this review, experiments will be conducted on photographs of scenes taken by normal cameras. Nevertheless, the image denoising problem is a common denominator of all imaging systems.

A naive view of the denoising problem would be: how to estimate the ideal image, namely the mean $u(\mathbf{i})$, given only one sample $\tilde{u}(\mathbf{i})$ of the Poisson variable? The best estimate of this mean is of course this unique sample $\tilde{u}(\mathbf{i})$. Getting back a better estimate of $u(\mathbf{i})$ by observing only $\tilde{u}(\mathbf{i})$ is impossible. Getting a better estimate by using also the rest of the image is obviously an ill-posed problem. Indeed, each pixel receives photons coming from different sources.

Nevertheless, a glimpse of a solution comes from image formation theory. A well-sampled image u is band-limited [136]. Thus, it seems possible to restore the band-limited image u from its degraded samples $\tilde{u}$, as was proposed in 1966 in [73]. This classic Wiener-Fourier method consists in multiplying the Fourier transform by optimal coefficients to attenuate the noise. It results in a convolution of the image with a low-pass kernel.

From a stochastic viewpoint, the band-limitedness of u also implies that values $\tilde{u}(\mathbf{j})$ at neighboring pixels $\mathbf{j}$ of a pixel $\mathbf{i}$ are positively correlated with $\tilde{u}(\mathbf{i})$. Thus, these values can be taken into account to obtain a better estimate of $u(\mathbf{i})$. These values being nondeterministic, Bayesian approaches are relevant and have been proposed as early as 1972 in [133].

In short, there are two complementary early approaches to denoising, the Fourier method, and the Bayesian estimation.

The Fourier method has been extended in the past thirty years to other linear space-frequency transforms such as the windowed DCT [152] or the many wavelet transforms [114].

Being first parametric and limited to rather restrictive Markov random field models [69], the Bayesian method are becoming non-parametric. The idea for the recent non parametric Markovian estimation methods is a now famous algorithm to synthesize textures from examples [60]. The underlying Markovian assumption is that, in a textured image, the stochastic model for a given pixel $\mathbf{i}$ can be predicted from a local image neighborhood P of $\mathbf{i}$, which we shall call "patch".

The assumption for recreating new textures from samples is that there are enough pixels $\mathbf{j}$ similar to $\mathbf{i}$ in a texture image $\tilde{u}$ to recreate a new but similar texture u. The construction

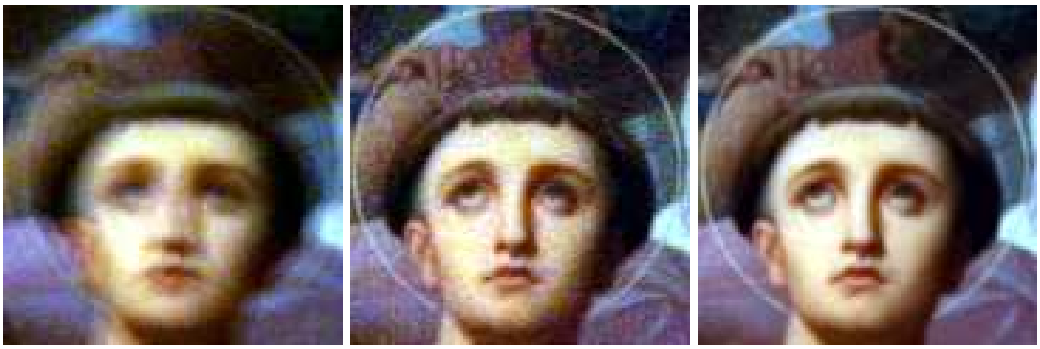


Figure 1: From left to right: (a) one long-exposure image (time=0.4 s, ISO=100), one of 16 short-exposure images (time=1/40 s, ISO=1600) and their average after registration. The long exposure image is blurry due to camera motion. (b) The middle short-exposure image is noisy. (c) The third image is about **four times** less noisy, being the result of averaging 16 short-exposure images. From [28].

of u is done by nonparametric sampling, amounting to an iterative copy-paste process. Let us assume that we already know the values of u on a patch P surrounding partially an unknown pixel $\mathbf{i}$. The Efros-Leung [60] algorithm looks for the patches $\tilde{P}$ in $\tilde{u}$ with the same shape as P and resembling P. Then a value $u(\mathbf{i})$ is sorted among the values predicted by $\tilde{u}$ at the pixels resembling $\mathbf{j}$. Indeed, these values form a histogram approximating the law of $u(\mathbf{i})$. This algorithm goes back to Shannon's theory of communication [136], where it was used for the first time to synthesize a probabilistically correct text from a sample.

As was proposed in [17], an adaptation of the above synthesis principle yields an image denoising algorithm. The observed image is the noisy image $\tilde{u}$. The reconstructed image is the denoised image $\hat{u}$. The patch is a square centered at $\mathbf{i}$, and the sorting yielding $u(\mathbf{i})$ is replaced by a weighted average of values at all pixels $\tilde{u}(\mathbf{j})$ similar to $\mathbf{i}$. This simple change leads to the "non-local means" algorithm, which can therefore be sketched in a few rows.

Algorithm 1 Non-local means algorithm

Input: noisy image $\tilde{u}$, σ noise standard deviation. **Output**: denoised image $\hat{u}$.
Set parameter $\kappa \times \kappa$: dimension of patches.
Set parameter $\lambda \times \lambda$: dimension of research zone in which similar patches are searched.
Set parameter C.
for each pixel $\mathbf{i}$ **do**
  Select a square reference sub-image (or "patch") $\tilde{P}$ around $\mathbf{i}$, of size $\kappa \times \kappa$.
  Call $\hat{P}$ the denoised version of $\tilde{P}$ obtained as a weighted average of the patches $\tilde{Q}$ in a square neighborhood of $\mathbf{i}$ of size $\lambda \times \lambda$. The weights in the average are proportional to $w(\tilde{P}, \tilde{Q}) = e^{-\frac{d^2(\tilde{P},\tilde{Q})}{C\sigma^2}}$ where $d(\tilde{P}, \tilde{Q})$ is the Euclidean distance between patches $\tilde{P}$ and $\tilde{Q}$.
end for
Aggregation: recover a final denoised value $\hat{u}(\mathbf{i})$ at each pixel $\mathbf{i}$ by averaging all values at $\mathbf{i}$ of all denoised patches $\hat{Q}$ containing $\mathbf{i}$

It was also proved in [17] that the algorithm gave the best possible mean square estimation if the image was modeled as an infinite stationary ergodic spatial process (see sec. 5.1 for an exact statement). The algorithm was called "non-local" because it uses patches $\tilde{Q}$ that are far away from $\tilde{P}$, and *even patches taken from other images*. NL-means was not the state of the art denoising method when it was proposed. As we shall see in the comparison section 6, the 2003 Portilla et al. [128] algorithm described in sec. 5.6 has a better PSNR performance. But quality criteria show that NL-means creates less artifacts than wavelet based methods. This may explain

why patch-based denoising methods have flourished ever since. By now, 1500 papers have been published on nonlocal image processing. Patch-based methods seem to achieve the best results in denoising. Furthermore, the quality of denoised images has become excellent for moderate noise levels. Patch-based image restoration methods are used in many commercial software.

An exciting recent paper in this exploration of nonlocal methods raises the following claim [92]: *For natural images, the recent patch-based denoising methods might well be close to optimality.* The authors use a set of 20000 images containing about 10^{10} patches. This paper provides a second answer to the question of absolute limits raised in [32], "Is denoising dead?". The Cramer-Rao type lower bounds on the attainable RMSE performance given in [32] are actually more optimistic: they allow for the possibility of a significant increase in denoising performance. The two types of performance bounds considered in [92] and [32] address roughly the same class of patch-based algorithms. It is interesting to see that these same authors propose denoising methods that actually approach these bounds, as we shall see in section 5.

The denoising method proposed in [92] is actually based on NL-means (algorithm 1), with the adequate parameter C to account for a Bayesian linear minimum mean square estimation (LMMSE) estimation of the noisy patch given a database of known patches. The only and important difference is that the similar patches Q are found on a database of 10^{10} patches, instead of on the image itself. Furthermore, by a simple mathematical argument and intensive simulations on the patch space, the authors are able to approach *the best average estimation error* which will *ever be attained by any patch-based denoising algorithm* (see sec. 5.4.)

These optimal bounds are nonetheless obtained on a somewhat restrictive definition of patch-based methods. A patch-based algorithm is understood as an algorithm that denoises each pixel by using the knowledge of: a) the patch surrounding it, and b) the probability density of all existing patches in the world. It turns out that state of the art patch-based denoising algorithms use more information taken in the image than just the patch. For example, most algorithms use the obvious but powerful trick to denoise all patches, and then to *aggregate* the estimation of all patches containing a given pixel to denoise it better. Conversely, these algorithms generally use much less information than a universal empirical law for patches. Nevertheless, the observation that at least one algorithm, BM3D [39] might be arguably very close to the best predicted estimation error is enlightening. Furthermore, doubling the size of the patch used in the [92] paper would be enough to cover the aggregation step. The difficulty is to get a faithful empirical law for 16×16 patches.

The "convergence" of all algorithms to optimality will be corroborated here by the thorough comparison of nine recent algorithms (section 6). These state of the art algorithms seem to attain a very similar qualitative and quantitative performance. Although they initially seem to rely on different principles, our final discussion will argue that these methods are equivalent.

Image restoration theory cannot be reduced to an axiomatic system, as the statistics of images are still a widely unexplored continent. Therefore, a complete theory, or a single final algorithm closing the problem are not possible. The problem is not fully formalized because there is no rigorous image model. Notwithstanding this limitation, rational recipes shared by all methods can be given, and the methods can be shown to rely on only very few principles. More precisely, this paper will present the following recipes, and compare them whenever possible:

- several families of noise estimation techniques (sec. 2);
- the four denoising principles in competition (sec. 3);
- three techniques that improve every denoising method (sec. 4);
- nine complete and recent denoising algorithms. For these algorithms complete recipes will be given (sec. 5);
- three complementary and simple recipes to evaluate and compare denoising algorithms (sec. 6).

Using the three comparison recipes, six emblematic or state of the art algorithms, based on reliable and public implementations, will be compared in sec. 6. This comparison is followed by a synthesis (sec. 7) hopefully demonstrating that, under very different names, the state of the art algorithms share the same principles.

Nevertheless, this convergence of results and techniques leaves several crucial issues unsolved. (This is fortunate, as no researcher likes finished problems.) With one exception, (the BLS-GSM algorithm, sec. 5.6), state of the art denoising algorithms are not multiscale. High noises and small noises also remain unexplored.

In a broader perspective, the success of image denoising marks the discovery and exploration of one of the first densely sampled high dimensional probability laws ever (numerically) accessible to mankind: the "patch space". For 8×8 patches, by applying a local PCA to the patches surrounding a given patch, one can deduce that this space has a dozen significant dimensions (the others being very thin). Exploring its structure, as was initiated in [87], seems to be the first step toward the statistical exploration of images. But, as we shall see, this local analysis of the *patch space* already enables state of the art image denoising.

Most denoising and noise estimation algorithms commented here will be available at the journal *Image Processing on Line,* `http://www.ipol.im/`. In this web journal, each algorithm is given a complete description, the corresponding source code, and can be run online on arbitrary images. By the time this paper is published, most results and techniques presented herewith will be effortlessly verifiable and reproducible online.

This introduction ends with a quick review of many contributions of interest seen recently about patch-based methods, which nevertheless fall beyond our limited scopes (sec. 1.1).

1.1 Miscellaneous "patch based" considerations and applications

Statistical validity This paper will compare patch-based algorithms on their structure, and on their practical performance, which is licit, in absence of a satisfactory mathematical or statistical model for digital images. Nonetheless, statistical arguments have also been developed to explore the validity of denoising algorithms. The statistical validity of NL-means is discussed in [141], [83] and [57] (where a Bayesian interpretation is proposed) or [151] where a bias of NL-means is corrected. [137] gives "a probabilistic interpretation and analysis of the method viewed as a random walk on the patch space". The most complete recent study is made in the realm of *Minimax approximation theory.* The Horizon class of images, which are piecewise constant with a sharp edge discontinuity [106] permits to perform an asymptotic analysis. The images are discontinuous across the edge and the edge itself is smooth, being in an $H^\alpha(C)$ class. A real function is in this class for $\alpha \geq 0$ if $|h^{([\alpha])}(t) - h^{([\alpha])}(s)| \leq C|t-s|^{\alpha-[\alpha]}$, where $[\alpha]$ is the integer part of α.

The principle is to measure the expected approximation rate of a denoising algorithm applied to m noisy samples of an image u in the horizon class. This image u is given by m samples, and these samples are perturbed by a white noise with variance σ^2. A denoising algorithm delivers a corrected function $\hat{u}$. The risk function of this algorithm is defined as the expectation $R_m(u, \hat{u})$ of the mean square distance of u and $\hat{u}$. Given a class of functions $\mathcal{F}$, the *minimax risk* is defined by

$$R_m(\mathcal{F}) = \inf_{\hat{u}} \sup_{u \in \mathcal{F}} R_m(u, \hat{u}),$$

where the inf is taken over all measurable estimators. It can be proven [108] that for $\alpha \geq 1$,

$$R_m(H^\alpha(C)) \simeq m^{-\frac{2\alpha}{\alpha+1}}. \tag{1}$$

For example for $\alpha = 2$, which corresponds to edges with bounded curvature, the optimal rate is $n^{-\frac{4}{3}}$. This result gives a sort of yardstick to measure, if not the performance, at least the theoretical limits of every denoising algorithm. This analysis has been conducted for several basic denoising methods including NL-means in [106]. The authors show that the decay rate is about m^{-1}, close to the one obtained with wavelet threshold denoising, better than rates of elementary filters such as a linear convolution, the median filter and the bilateral filter which have rates $m^{-\frac{2}{3}}$. The decay rate of NL-means is nonetheless away from the optimal minimax rate of $m^{-4/3}$, which is only attained for $\alpha = 2$ by the wedgelet transform. The same authors prove in [105] that an anisotropic nonlocal means (ANLM) algorithm is near minimax optimal for edge-dominated images from the Horizon class. The idea is to orient optimally rectangular thin blocks for performing the comparison. The algorithms improves on NL-means by approximately one decibel.

Other noise models. The present article focuses on algorithms removing white additive noise from digital optical images. There are other types of noise in other imaging systems. Thus, this study cannot account for the burgeoning variety of patch-based algorithms. Improvements or adaptations of NL-means have been proposed in cryo-electron microscopy [45], fluorescence microscopy [11], magnetic resonance imaging (MRI) [109], [8], [149], [117], multispectral MRI: [110], [16], and diffusion tensor MRI (DT-MRI) [148].

More invariance. Likewise, several papers have explored which degree of invariance could be applied to image patches. [160] explores a rotationally invariant block matching strategy improving NL-means, and [58] uses cross-scale (i.e., down-sampled) neighborhoods in the NL-means filter. See also [105], mentioned above as reaching better minimax limits. It uses oriented anisotropic patches. Self-similarity has also been explored in the Fourier domain for MRI in [112].

Fast patch methods Several papers have proposed fast and extremely fast (linear) NL-means implementations, by block pre-selection [98], [10], by Gaussian KD-trees to classify image patches [1], by SVD [121], by using the FFT to compute correlation between patches [146], by statistical arguments [38] and by approximate search [9], also used for optical flow.

Other image processing tasks The non-local denoising principle has also been expanded to most image processing tasks: *Demosaicking*, the operation which transforms the "R or G or B" raw image in each camera into an "R and G and B" image [25], [102], *movie colourization*, [65] and [93]; *image inpainting* by proposing a non local image inpainting variational framework with a unified treatment of geometry and texture [5] (see also [150]) ; *zooming* by a fractal like technique where examples are taken from the image itself at different scales [57]; *movie flicker stabilization* [51], compensating spurious oscillations in the colours of successive frames; *super-resolution*, an image zooming method fusing several frames from a video, or several low resolution photographs, into a larger image [129]. The main point of this super-resolution technique is that it gives up an explicit estimate of the motion, allowing actually for a multiple motion, since a block can look like several other patches in the same frame. The very same observation is made in [59] for devising a super-resolution algorithm, and also in [63], [41]. Other classic image nonlocal applications include image *contrast enhancement* by applying a reverse non local heat equation [20], and *Stereo vision*, by performing simultaneous non-local depth reconstruction and restoration of noisy stereo images [75].

The link to PDE's, variational variants The relationship of neighborhood filters to classic local PDE's has been discussed in [21] and [22] leading to an adaptation of NL-means which avoids the staircase effect. Nonlocal image-adapted differential operators and non-local variational methods are introduced in [84], which proposes to perform denoising and deblurring by non-local functionals. The general goal of this development is actually to give a variational form to all neighborhood filters, and to give a non local form to the total variation [134] as well. Several articles on deblurring have followed this variational line [78], [115], [70] (for image segmentation), [11] (in fluorescence microscopy), [158], again for nonlocal deconvolution and [96] for deconvolution and tomographic reconstruction. In [63], a paper dedicated to another notoriously ill-posed problem, the super-resolution, the non-local variational principle is viewed as "an emerging powerful family of regularization techniques", and the paper "proposes to use the example-based approach as a new regularizing principle in ill-posed image processing problems such as image super-resolution from several low resolution photographs." A particular notion of non-local PDE has emerged, whose coefficients are actually image-dependent. For instance, in [65] the image colourization is viewed as the minimization of a discrete partial differential functional on the weighted block graph. Thus, it can be seen either as a non-local heat equation on the image, or as a local heat equation on the space of image patches.

The geometric interpretation in a graph of patches In an almost equivalent framework, in [140] the set of patches is viewed as a weighted graph, and the weights of the edge between two patches centered at **i** and **j** respectively are decreasing functions of the block distances. Then a graph Laplacian can be calculated on this graph, seen as the sampling of a manifold, and NL-means can be interpreted as the heat equation on the set of blocks endowed with these weights. In the same way, the neighborhood filter can be associated with a heat equation on the image graph [125]. This approach is further extended to a variational formulation on patch graphs in [64]. In this same framework [20] proposed to perform image contrast enhancement by applying a non-local reverse heat equation. Finally, always in this non-local partial differential framework, [14] extends the Mumford-Shah image segmentation energy to contain a non-local self-similarity term replacing the usual Dirichlet term. The square of the gradient is replaced by the square of the non-local gradient.

2 Noise

2.1 Noise models

Most digital images and movies are obtained by a CCD device and the main source of noise is the so-called *shot noise*. Shot noise is inherent to photon counting. The value $\tilde{u}(\mathbf{i})$ observed by a sensor at each pixel $\mathbf{i}$ is a Poisson random variable whose mean would be the ideal image. The standard deviation of this Poisson distribution is equal to the square root of the number of incoming photons $\tilde{u}(\mathbf{i})$ in the pixel captor $\mathbf{i}$ during the exposure time. This noise adds up to a thermal noise and to an electronic noise which are approximately additive and white.

For sufficiently large values of $\tilde{u}(\mathbf{i})$, ($\tilde{u}(\mathbf{i}) > 1000$), the normal distribution $\mathcal{N}(\tilde{u}(\mathbf{i}), \sqrt{\tilde{u}(\mathbf{i})})$ with mean $\tilde{u}(\mathbf{i})$ and standard deviation $\sqrt{\tilde{u}(\mathbf{i})}$ is an excellent approximation to the Poisson distribution. If $\tilde{u}(\mathbf{i})$ is larger than 10, then the normal distribution still is a good approximation if an appropriate continuity correction is performed, namely $\mathbb{P}(\tilde{u}(\mathbf{i}) \leq a) \simeq \mathbb{P}(\tilde{u}(\mathbf{i}) \leq a + 0.5)$, where a is any non-negative integer.

Nevertheless, the pixel value is *signal dependent*, since its mean and variance depend on $\tilde{u}(\mathbf{i})$. To get back to the classic "white additive Gaussian noise" used in most researches on image denoising, a *variance-stabilizing transformation* can be applied: When a variable is Poisson distributed with parameter $\tilde{u}(\mathbf{i})$, its square root is approximately normally distributed with expected value of about $\sqrt{\tilde{u}(\mathbf{i})}$ and variance of about 1/4. Under this transformation, the convergence to normality is faster than for the untransformed variable[1]. The most classic VST is the Anscombe transform [3] which has the form $f(u_0) = b\sqrt{u_0 + c}$.

The denoising procedure with the standard variance stabilizing transformation (VST) procedure follows three steps,

1. apply VST to approximate homoscedasticity;
2. denoise the transformed data;
3. apply an inverse VST.

Note that the inverse VST is not just an algebraic inverse of the VST, and must be optimized to avoid bias [104].

Consider any additive signal dependent noisy image, obtained for example by the Gaussian approximation of a Poisson variable explained above. Under this approximation, the noisy image satisfies $\tilde{u} \simeq \tilde{u} + g(\tilde{u})n$ where $n \simeq \mathcal{N}(0, 1)$. We can search for a function f such that $f(\tilde{u})$ has uniform standard deviation,

$$f(\tilde{u}) \simeq f(\tilde{u}) + f'(\tilde{u})g(\tilde{u})n.$$

Forcing the noise term to be constant, $f'(\tilde{u})g(\tilde{u}) = c$, we get

$$f'(\tilde{u}) = \frac{c}{g(\tilde{u})},$$

[1] See `http://en.wikipedia.org/wiki/Poisson_distribution`.

and integrating

$$f(\tilde{u}) = \int_0^{\tilde{u}} \frac{c\,dt}{g(t)}.$$

When a linear variance noise model is taken, this transformation gives back an Anscombe transform. Most classical denoising algorithms can also be adapted to signal dependent noise. This requires varying the denoising parameters at each pixel, depending on the observed value $\tilde{u}(\mathbf{i})$. Several denoising methods indeed deal directly with the Poisson noise. Wavelet-based denoising methods [119] and [85] propose to adapt the transform threshold to the local noise level of the Poisson process. Lefkimmiatis et al. [91] have explored a Bayesian approach without applying a VST. Deledalle et al., [47] argue that for high noise level it is better to adapt NL-means than to apply a VST. These authors proposed to replace the Euclidean distance between patches by a likelihood estimation taking into account the noise model. This distance can be adapted to each noise model such as the Poisson, the Laplace or the Gamma noise [49], and to more complex (speckle) noise occurring in radar (SAR) imagery [50].

Nonetheless, dealing with a white uniform Gaussian noise makes the discussion on denoising algorithms far easier. The recent papers on the Anscombe transform [104] (for low count Poisson noise) and [66] (for Rician noise) argue that, when combined with suitable forward and inverse VST transformations, algorithms designed for homoscedastic Gaussian noise work just as well as ad-hoc algorithms signal-dependent noise models. This explains why in the rest of this paper the noise is assumed uniform, white and Gaussian, having previously applied, if necessary, a VST to the noisy image. This also implies that we deal with *raw* images, namely images as close as possible to the direct camera output before processing. Most reflex cameras, and many compact cameras nowadays give access to this raw image.

But there is definitely a need to denoise current image formats, which have undergone unknown alterations. For example, the JPEG-encoded images given by a camera contain a noise that has been altered by a complex chain of algorithms, ending with lossy compression. Noise in such images cannot be removed by the current state of the art denoising algorithms without a specific adaptation. The key is to have a decent noise model. For this reason, the fundamentals to estimate noise from a single image will be given in section 2.2.

2.2 Can noise be estimated from (just) one image?

Compared to the denoising literature, research on noise estimation is a poor cousin. Few papers are dedicated to this topic. Among the recent papers one can mention [162], which argues that images are scale invariant and therefore noise can be estimated by a deviation from this assumption. Unfortunately this method is not easily extendable to estimate scale dependent or signal dependent noise, like the one observed in most digital images in compressed format. As a rule of thumb, the noise model is relatively easy to estimate when the raw image comes directly from the imaging system, in which case the noise model is known and only a few parameters must be estimated. For this, efficient methods are described in [68], [67] for Poisson and Gaussian noise.

In this short review we will focus on methods that allow for local, signal and scale dependent noise. Indeed, one cannot denoise an image without knowing its noise model. It might be argued that the noise model comes with the knowledge of the imaging device. Nevertheless, the majority of images dealt with by the public or by scientists have lost this information. This loss is caused by format changes of all kinds, which may include resampling, denoising, contrast changes and compression. All of these operations change the noise model and make it *signal and scale dependent*.

The question that arises is why so many researchers are working so hard on denoising models, if their corpus of noisy images is so ill-informed.

It is common practice among image processing researchers to add the noise themselves to noise-free images to demonstrate the performance of a method. This proceeding permits to reliably evaluate the denoising performance, based on a controlled ground truth. Nevertheless the denoising performance may, after all, critically depend on how well we are able to estimate the noise. Most

Figure 2: Two examples of the ten noise-free images used in the tests: *computer* (left) and *traffic* (right).

world images are actually encoded with lossy JPEG formats. Thus, noise is partly removed by the compression itself. Furthermore, this removal is scale dependent. For example, the JPEG 1985 format divides the image into a disjoint set of 8×8 pixels blocks, computes their DCT, quantizes the coefficients and the small ones are replaced by zero. This implies that JPEG performs a frequency dependent threshold, equivalent to a basic Wiener filter. The same is true for JPEG 2000 (based on the wavelet transform).

In addition, the Poisson noise of a raw image is signal dependent. The typical image processing operations, demosaicking, white balance and tone curve (contrast change) alter this signal-dependency in a way which depends on the image itself.

In short:

- the noise model is different for each image;
- the noise is signal dependent;
- the noise is scale dependent;
- the knowledge of each dependence is crucial to denoise properly any given image which is not raw, and for which the camera model is available.

Thus, estimating JPEG noise is a complex and risky procedure, as well explained in [95] and [94]. It is argued in [44] that noise can be estimated by involving a denoising algorithm. Again, this procedure is probably too risky for noise and scale dependent signal.

This section, following [26], gives a concise review and a comparison of existing noise estimation methods. The classic methods estimate white homoscedastic noise only, but they can be adapted easily to estimate signal and scale dependent noise. To test the methods, a set of ten noise-free images was used. These noiseless images were obtained by taking snapshots with a reflex camera of scenes under good lighting conditions and with a low ISO level. This means that the number of photons reaching each captor was very high, and the noise level therefore small. To reduce further the noise level, the average of each block of 5×5 pixels was computed, reducing the noise by a 5 factor. Since the images are RGB, taking the mean of the three channels reduces the noise by a further $\sqrt{3}$ factor. The (small) initial noise was therefore reduced by a $5\sqrt{3} \simeq 8.66$ factor, and the images can be considered noise-free. Two images from this noiseless set can be seen in fig. 2. The size of each image is 704×469 pixels. For the uniform-noise tests, seven noise levels were applied to these noise-free images: $\sigma \in \{1, 2, 5, 10, 20, 50, 80\}$. Fig. 3 shows the result of adding white homoscedastic Gaussian noise with $\sigma \in \{1, 2, 5, 10, 20, 50, 80\}$ to the noise-free image *traffic*.

This study on noise estimation proceeds as follows: we review in detail in section 2.3 the method proposed in [26]. This method has all the features of the preceding methods, so we shall be able to make a rash review of them (section 2.4), followed by an overall comparison of all methods, at all noise levels. It follows that the Percentile method is the most accurate. Nevertheless, the estimation of very low noises remains slightly inaccurate, with some 20% error for noises below 2.

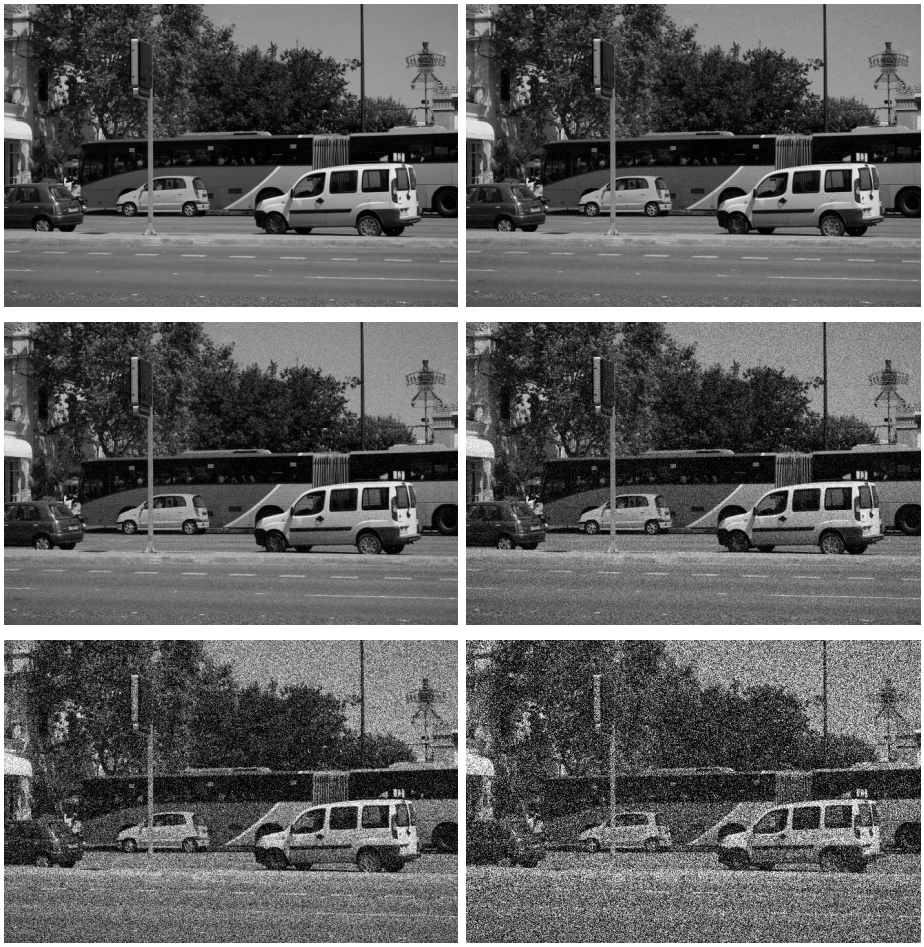


Figure 3: Result of adding white homoscedastic Gaussian noise with $\sigma \in \{2, 5, 10, 20, 50, 80\}$ to the noise-free image *traffic*. It may need a zoom in to perceive the noise for $\sigma = 2, 5$.

2.3 The Percentile method

The Percentile method, introduced in [126], is based on the fact that the histogram of the variances of all blocks in an image is affected by the edges and textures, but this alteration appears mainly on its rightmost part. The idea of the *percentile method* is to avoid the side effect of edges and textures by taking the variance of a very low percentile of the block variance histogram, and then to infer from it the real average variance of blocks containing only noise. This correction multiplies this variance by a factor that only depends on the choice of the percentile and the block size. As usual in all noise estimation methods, to reduce the presence of deterministic tendencies in the blocks, due to the signal, the image is first high passed. The commonly used high pass filters are differential operators or waveforms. The typical differential operators are directional derivatives, the Δ (Laplace) operator, its iterations $\Delta\Delta$, $\Delta\Delta\Delta$, ..., the wave forms are wavelet or DCT coefficients. All of them are implemented as discrete stencils. Filtering the image with such a local high pass filter operator removes smooth variations inside blocks, which increases the number of blocks where noise dominates and on which the variance estimate will be reliable. According to the performance tests, for observed $\hat{\sigma} < 75$ the best operator is the wave associated to the highest frequency coefficient of the transformed 2D DCT-II block with support 7×7 pixels.

The coefficient $\tilde{X}(6,6)$ of the 2D DCT-II of a 7×7 block P of the image is:

$$DCT(6,6) = \sum_{n_1=0}^{6} \sum_{n_2=0}^{6} F_7(n_1) F_7(n_2) P(n_1, n_2) \cos\left[\frac{\pi}{7}\left(n_1 + \frac{1}{2}\right)6\right] \cos\left[\frac{\pi}{7}\left(n_2 + \frac{1}{2}\right)6\right].$$

where

$$F_7(n) = \begin{cases} \frac{1}{\sqrt{7}}, \text{if } n = 0 \\ \sqrt{\frac{2}{7}}, \text{if } n \in \{1, \ldots, 6\} \end{cases}$$

Therefore, the values of the associated discrete filter are

$$F_7(n_1) F_7(n_2) \cos\left[\frac{\pi}{7}\left(n_1 + \frac{1}{2}\right)6\right] \cos\left[\frac{\pi}{7}\left(n_2 + \frac{1}{2}\right)6\right], n_1, n_2 \in \{0, 1, \ldots, 6\}.$$

These values must of course be normalized in order to keep the standard deviation of the data, by dividing each value by the root of the sum of the filter squared values.

The Percentile method computes the variances of overlapping $w \times w$ blocks in the high-pass filtered image. The means of the same blocks are computed from the original image (before the high pass). These means are classified into a disjoint union of variable intervals, in such a way that each interval contains (at least) 42000 elements. These measurements permit to construct, for each interval of means, a histogram of block variance of at least 42000 samples having their means in the interval. In each such variance histogram the percentile value is computed. It was observed that, for observed $\hat{\sigma} < 75$ and large images, the percentile $p = 0.5\%$, a block size $w = 21$ and a 7×7 support for the DCT transform give the best results. If $\hat{\sigma} \geq 75$, the percentile that should be used is the median, the block is still 21×21, but the support of the DCT should be 3×3.

This percentile value is of course lower than the real average block variance, and must be corrected by a multiplicative factor. This correction only depends on the percentile, block size and on the chosen high pass filter. Nevertheless, the constant is not easy to calculate explicitly, but can be learnt from simulations. For the 0.5% percentile, 21×21 pixels blocks and the DCT pre-filter operator with support 7×7, this empirical factor learnt on noise images was found to be 1.249441884. In summary, to each interval of means, a standard deviation is associated. The association mean→standard deviation yields a "noise curve" associated with the image. This noise curve predicts for each observed grey level value in the image its most likely underlying standard deviation due to noise. Optionally, the noise curve obtained on real images can be filtered. Indeed, it may present some peaks when variances measured for a given grey level interval belong to a highly-textured region. To filter the curve, the points that are above the segment that joins the points on the left and on the right are back-projected on that segment. In general, no more than

Image / $\hat{\sigma}$	$\sigma = 1$	$\sigma = 2$	$\sigma = 5$	$\sigma = 10$	$\sigma = 20$	$\sigma = 50$	$\sigma = 80$
bag	1.34	2.33	5.26	10.36	20.30	49.87	79.96
building1	1.12	2.17	5.24	10.14	20.48	50.19	80.45
computer	1.22	2.20	5.06	10.36	20.03	50.28	80.34
dice	1.11	2.00	5.01	10.03	20.02	49.95	79.79
flowers2	1.08	2.07	5.10	9.84	20.07	49.87	79.80
hose	1.15	2.13	5.10	10.15	20.06	49.99	79.99
leaves	1.51	2.43	5.38	10.29	19.82	50.07	80.04
lawn	1.57	2.50	5.57	10.48	20.42	50.05	79.92
stairs	1.42	2.27	5.19	10.15	19.96	49.92	79.93
traffic	1.25	2.35	5.33	10.61	20.64	50.10	80.29
Flat image	0.99	2.00	5.09	9.77	19.91	50.12	79.73

Table 1: Percentile method results on eleven noiseless images with white homoscedastic Gaussian noise added. The last image is simply flat. The real noise variance is σ. The estimated value is $\hat{\sigma}$. The noise estimation error is remarkably low on medium and large noise. It is nevertheless larger on very small noise (a $\sigma = 2$ noise is not visible with the naked eye). Indeed most photographed objects have everywhere some micro-texture (except perhaps sometimes in the blue sky which can be fully homogeneous). Such micro-textures are widespread and hardly distinguishable from noise. The parameters of the method are a 0.5% percentile, a 21×21 pixels block size, and the DCT has support 7×7. These parameters are valid if $\hat{\sigma} < 75$. If $\hat{\sigma} \geq 75$, the best parameters are: a 50% percentile, a 21×21 pixels block size and a DCT with support 3×3. Estimating the best parameters therefore requires a first estimation followed by a second one with the right parameters.

two filtering iterations are needed. For the comparative tests presented here, the curves were not filtered at all.

The pseudo-code for the percentile method is given in Algo. 2 and the results for the white homoscedastic Gaussian noise in Table 1. When the image is tested for white homoscedastic Gaussian noise, only one interval for all grey level means is used, whereas in the signal-dependent noise case, the grey level interval is divided into seven bins.

Algorithm 2 Percentile method algorithm.

PERCENTILE - Returns a list that relates the value of the image signal with its noise level. **Input** $\tilde{u}$ noisy image. **Input** b: number of bins. **Input** $w \times w$: block dimensions **Input** p: percentile. **Input** $filt$: filter iterations. **Output** (M, S): list made of pairs (mean, noise standard deviation) for each bin of grey level value.

$h = \text{FILTER}(\tilde{u})$. Apply high-pass filter to the image.
$a, v = \text{MEAN\_FILTERED\_VARIANCE}(\tilde{u}, h, w)$. Obtain the list of the block averages (in the original image $\tilde{u}$) and of the variances (of the filtered image h) for all $w \times w$ blocks.
Divide the block mean value list a into intervals (bins), having all the same number of elements. Keep for each interval the corresponding values in v.

$S = \emptyset$; $M = \emptyset$.
for each bin **do**
  $v = \text{Per}(\text{bin}, p)$. Get the p-percentile v of the block variances whose means belong to this bin.
  $m = \text{Mean}[\text{Per}(\text{bin}, p)]$. Get the mean of the block associated to that percentile.
  $S \leftarrow \sqrt{v}$. Store the standard deviation $\hat{\sigma}$.
  $M \leftarrow m$. Store mean.
end for
$S_c = \emptyset$. Corrected values.
for $s \in S$ **do**
  Apply correction C according to p, w and filter operator used.
  $s = Cs$. Correct direct estimate.
  $S_c \leftarrow s$.
end for
for $k = 1 \dots filt$ **do**
  $S_c[k] = \text{FILTER}(S_c[k], \text{filt})$. Filter the noise curve $filt$ times.
end for

Figure 4: Mosaic used to learn the correction values in the Percentile method.

The Percentile method with learning The percentile method with learning is essentially the same algorithm explained in section 2.3, with the difference that it tries to compensate the bias caused by edges and micro-texture in the image by learning a relationship between observed values $\hat{\sigma}$ and noise real values σ. The difference value $f(\sigma) = \hat{\sigma} - \sigma$ is called the *correction*, that is, the value that must be subtracted from the direct estimate $\hat{\sigma}$ without correction to get the final estimate (which we shall still call $\hat{\sigma} \approx \sigma$). These corrections depend on the structure of real images. A mosaic of several noise-free images is shown in Fig. 4. Simulated noise of standard deviations $\sigma = 0, \ldots, 100$ was added to these noiseless images. These images were selected randomly from a large database, to be statistically representative of the natural world, with textures, edges, flat regions, dark and bright regions. The correction learnt with these images is intended to be an average correction, that works for a broad range of natural images. It should of course be adapted to any particular set of images. Furthermore, the correction depends on the size of the image, and must be learnt for each size.

When the observed noise level is high enough ($\hat{\sigma} > 10$ for pixel intensities $u \in \{0, 1, \ldots 255\}$), the image gets dominated by the noise, that is, most of the variance measured is due to the noise and not due to the micro-textures and edges. It is therefore convenient to avoid applying the learnt corrections to direct estimates $\hat{\sigma}$ when $\hat{\sigma} > 10$. Thus, for $\hat{\sigma} > 10$, only the percentile correction is applied. Table 2 shows the $\hat{\sigma}$ values estimated with the Percentile with learning method. The correction learnt with the mosaic is only applied for $\sigma \in \{1, 2, 5, 10\}$.

2.4 A crash course on all other noise estimation methods

It is easier to explain the other methods after having explained in detail, as we did above, one method, namely the percentile method. Most noise estimation methods share the following features:

- they start by applying some high pass filter, which concentrates the image energy on boundaries, while the noise remains spatially homogeneous;
- they compute the energy on many blocks extracted from this high-passed image;
- they estimate the noise standard deviation from the values of the standard deviations of the blocks
- to avoid blocks contaminated by the underlying image, a statistics robust to (many) outliers must be applied. The methods therefore use the flattest blocks, which belong to a (low) percentile of the histogram of standard deviations of all blocks.

Table 3 shows a classification of the methods according the preceding criteria:

Image / $\hat{\sigma}$	$\sigma = 1$	$\sigma = 2$	$\sigma = 5$	$\sigma = 10$	$\sigma = 20$	$\sigma = 50$	$\sigma = 80$
bag	1.15	2.11	5.05	10.26	20.06	49.68	80.05
building1	0.95	1.97	5.00	10.42	20.32	49.99	80.27
computer	1.04	2.00	4.88	10.39	20.13	50.29	80.16
dice	0.91	1.84	4.81	10.01	19.90	49.76	79.60
flowers2	0.92	1.88	4.87	9.47	20.00	49.48	79.67
hose	0.99	1.93	4.89	10.08	19.97	49.73	79.71
leaves	1.36	2.26	5.17	10.28	20.03	49.80	79.92
lawn	1.35	2.29	5.36	10.37	20.26	50.07	79.88
stairs	1.20	2.10	4.95	10.11	20.10	49.92	79.86
traffic	1.04	2.06	5.06	10.75	20.64	49.91	80.05
Flat image	0.84	1.82	4.84	10.02	20.13	50.13	79.44

Table 2: Percentile with learning method results with white homoscedastic Gaussian noise added. The correction learnt with the mosaic is only applied for $\sigma \in \{1, 2, 5, 10\}$. This method, being local on blocks, extends immediately to estimate signal dependent noise and the performance is similar [26].

The first column is the choice of the high-pass filter, which can be a discrete differential operator of order two ($\frac{\partial^2}{\partial x \partial y}$) in the Estimation of Image Noise Variance (E.I.N.V.) method [130]). It is obtained as a composition of two forward discrete differences. Then we have a discrete Laplacian Δ [120] obtained as the difference between the current pixel value and the average of a discrete neighborhood, an order order three operator (a difference $\Delta_1 - \Delta_2$ of two different discretizations of the Laplacian [77]), a wave associated to a DCT coefficient [26], and sometimes a nonlinear discrete differential operator like in the Median method [120], which uses the difference between the image and its median value on a 3×3 block, thus equivalent to the curvature operator $curv$. The high-pass filter is previously applied to all pixels of the image. In the case of the DCT [127] the DCT is applied to a block centered on the reference pixel, and the highest frequency coefficients, for example $DCT(6, 7)$, $DCT(7, 6)$, $DCT(7, 7)$, are kept. The most primitive methods, the Block [89, 111], the Pyramid [113] and the Scatter method [88] do not apply any high pass filter. Nevertheless, since they compute block variances, they implicitly remove the mean from each block, which amounts to applying a high-pass filter of Laplacian type.

The second column gives the size of the block on which the standard deviation of the high-passed image is computed, which varies from 1 to 21. The pyramid method [113] uses standard deviations of blocks of all sizes and is unclassifiable. Two methods, F.N.V.E. [77] and the Gradient method [12, 145] do not compute any block standard deviation of the high-passed image before the final estimation.

The last column gives the value of the (low) percentile on which the block standard deviation are computed. When the slot contains "all", this means that the estimator is taking into account all the values.

The third column characterizes the estimator, for which there are several variants. The three compared percentile methods [26] use a very low percentile 0.5% of the block standard deviations. The Average, Median [120] and Block method [89, 111] use an 1% percentile of the gradient to select the blocks which variance is kept, while the high pass image is a higher order differential operator. The Pyramid [113] is instead quite complex, but uses overall all standard deviations of all possible blocks in the image. We give up giving its detailed algorithm. The F.N.V.E. [77] method has actually no outlier elimination, taking simply the root mean square of all samples of the high-passed image.

Rather than using a percentile of the block variance histogram followed by a compensation factor, several methods extract a mode, considering that the mode (peak of the histogram variance)

Method	Hi-pass	Block	estimator	percentile
Perc. learn. [26, 126]	DCT 7×7	**21**	block dev. at perc.	0.5%
Percentile [26, 126]	DCT 7×7	**21**	block dev. at perc.	0.5%
Block [89, 111]	none	7	mean of block dev	1%
Average [120]	Δ	3	mean of block dev	1% of grad. hist.
Median [120]	*curv*	3	mean of block dev	1% of grad. hist.
Scatter [88]	none	8	block dev at	block dev mode
Gradient [12, 145]	∇	1	$\lvert\nabla\rvert$ mode	all
E.I.N.V. [130]	$\frac{\partial^2}{\partial x \partial y}$	3	deconv. of block dev.	all
F.N.V.E. [77]	$\Delta_1 - \Delta_2$	1	RMS	all
DCT-MAD [53]	3-DCT	8	MAD of 3 DCT coef	all
DCT-mean [127]	3-DCT	8	mean of variances	all
Pyramid [113]	none	2^L	block dev	complex

Table 3: Table summarizing all methods. The abbreviation"block dev." means standard deviation of block, "at perc 1%" means that the chosen value is the one at which the 1% percentile is attained. "3-DCT" means the three highest frequency coefficients, namely $DCT(6,7)$, $DCT(7,6)$, $DCT(7,7)$. "DCT 7×7" means the DCT wave associated to the highest frequency coefficient of the 7×7 pixels support of the DCT-II transform of the block. MAD stands for median of absolute deviation (it is applied to the three DCT coefficients for all blocks.) The methods belong to three classes. The first main class (rows 1 to 5) does: high pass+ standard deviation of blocks+ low percentile. The second class (rows 6-7) replaces the percentile by a mode of the high-pass filter histogram. The rows 8-9-10-11 are more primitive and do a simple mean of the block variances of the high-pass filtered image. The last method is unclassifiable, and performs poorly.

must correspond to the noise. The Gradient method [12, 145] takes for $\hat{\sigma}$ the peak of the modulus of the gradient histogram. The Scatter [88] method, which also computes a mode when estimating white homoscedastic noise, namely the value at which the peak of the block standard deviations histogram is attained. The E.I.N.V. [130] method does a sort of iterative deconvolution of the histogram of block variances and also extracts its mode.

All of the values obtained by these methods are proportional to the noise standard deviation when the image is a white noise. Thus the final step, not mentioned in the table, is to apply a correction factor to get the final estimated noise standard deviation, as explained in the percentile method (sec. 2.3).

The comparison of the methods which use the highest DCT coefficients, DCT-mean [127] and DCT-MAD [53] where MAD stands for median value of absolute deviations, shows clearly the win with a robust estimator: the estimation is obtained by averaging the three MAD (median of absolute deviation) of the three highest frequency DCT coefficients for all blocks.

The ultimate choice for the methods is of course steered by their RMSE, namely the root mean square error between the estimated value of σ and σ itself, taken over a representative set of images. As Table 4 shows the ordering of methods by their RMSE is coherent and points to the percentile method as the best one. This method is still improved by learning. A good point justifying all methods is that they perform satisfactorily for all large noise values, down to $\sigma = 20$. But, with the exception of the Percentile method with learning, no method performs acceptably for $\sigma < 5$.

Method	$\sigma = 1$	$\sigma = 2$	$\sigma = 5$	$\sigma = 10$	$\sigma = 20$	$\sigma = 50$	$\sigma = 80$
Percentile	**0.309**	**0.276**	**0.265**	**0.315**	**0.293**	**0.130**	**0.229**
Percentile learning	**0.182**	**0.152**	**0.157**	0.364	0.240	0.248	0.270
Block	1.093	0.961	0.949	1.056	0.984	0.922	0.840
Average	2.669	2.556	2.375	2.165	1.771	1.227	0.874
Median	2.841	2.762	2.640	2.460	2.110	1.684	1.502
Scatter	4.533	4.013	3.141	2.290	1.436	1.488	1.862
Gradient	1.887	1.851	1.474	1.393	1.354	1.234	2.949
E.I.N.V.	1.406	1.159	0.924	0.842	0.656	0.450	0.557
F.N.V.E.	2.738	2.231	1.357	0.767	0.397	**0.196**	**0.225**
DCT-MAD	0.858	0.721	0.533	**0.356**	**0.239**	0.296	0.583
DCT-mean	1.895	1.469	0.837	0.462	0.316	0.355	0.726

Table 4: White homoscedastic Gaussian noise RMSE results for all methods and for varying σ. The Pyramid tests were omitted, being incomplete. Being obtained as an average on many noiseless images, the differences have been checked to be statistically significant. It is also apparent that the ranking of the compared methods may vary with the amount of noise. Nevertheless, the ranks of methods for noises larger than 20 is irrelevant, because all of them work at an acceptable level of precision. Thus, this ranking is mainly relevant for low noise levels, $\sigma = 1, 2, 5, 10$.

3 Four denoising principles

In this section, we will review the main algorithmic principles which have been proposed for noise removal. All of them use of course a model for the noise, which in our study will always be the Gaussian white noise. More interestingly, each principle implies a model for the ideal noiseless image. The Bayesian principle is coupled with a Gaussian (or a mixture of Gaussians) model for noiseless patches. Transform thresholding assumes that most image coefficients are high and sparse in a given well-chosen orthogonal basis, while noise remains white (and therefore with homoscedastic coefficients in any orthogonal basis). Sparse coding assumes the existence of a dictionary of patches on which most image patches can be decomposed with a sparse set of coefficients. Finally the averaging principle relies on an image self-similarity assumption. Thus four considered denoising principles are:

- Bayesian patch-based methods (Gaussian patch model);
- transform thresholding (sparsity of patches in a fixed basis);
- sparse coding (sparsity on a learned dictionary);
- pixel averaging and block averaging (image self-similarity).

As we will see in this review, the current state of the art denoising recipes are actually a smart combination of *all* of these ingredients.

3.1 Bayesian patch-based methods

Given u the noiseless ideal image and $\tilde{u}$ the noisy image corrupted with Gaussian noise of standard deviation σ so that

$$\tilde{u} = u + n, \tag{2}$$

the conditional distribution $\mathbb{P}(\tilde{u} \mid u)$ is

$$\mathbb{P}(\tilde{u} \mid u) = \frac{1}{(2\pi\sigma^2)^{\frac{M}{2}}} e^{-\frac{||u-\tilde{u}||^2}{2\sigma^2}}, \tag{3}$$

where M is the total number of pixels in the image.

In order to compute the probability of the original image given the degraded one, $\mathbb{P}(u \mid \tilde{u})$, we need to introduce a prior on u. In the first models [69], this prior was a parametric image model describing the stochastic behavior of a patch around each pixel by a Markov random field, specified by its Gibbs distribution. A Gibbs distribution for an image u takes the form

$$\mathbb{P}(u) = \frac{1}{Z} e^{-E(u)/T},$$

where Z and T are constants and E is called the energy function and writes

$$E(u) = \sum_{C \in \mathcal{C}} V_C(u),$$

where $\mathcal{C}$ denotes the set of cliques associated to the image and V_C is a potential function. The maximization of the *a posteriori* distribution writes by Bayes formula

$$\operatorname*{Arg\,max}_u \mathbb{P}(u \mid \tilde{u}) = \operatorname*{Arg\,max}_u \mathbb{P}(\tilde{u} \mid u)\mathbb{P}(u),$$

which is equivalent to the minimization of $-\log \mathbb{P}(u \mid \tilde{u})$,

$$\operatorname*{Arg\,min}_u \|u - \tilde{u}\|^2 + \frac{2\sigma^2}{T} E(u).$$

This energy writes as a sum of local derivatives of pixels in the image, thus being equivalent to a classical Tikhonoff regularization, [69] and [13].

Recent Bayesian methods have abandoned as too simplistic the global patch models formulated by an *a priori* Gibbs energy. Instead, the methods build local non parametric patch models learnt from the image itself, usually as a local Gaussian model around each given patch, or as a Gaussian mixture. The term "patch model" is now preferred to the terms "neighborhood" or "clique" previously used for the Markov field methods. In the nonparametric models, the patches are larger, usually 8×8, while the cliques were often confined to 3×3 neighborhoods. Given a noiseless patch P of u with dimension $\kappa \times \kappa$, and $\tilde{P}$ an observed noisy version of P, the same model gives by the independence of noise pixel values

$$\mathbb{P}(\tilde{P}|P) = c \cdot e^{-\frac{\|\tilde{P}-P\|^2}{2\sigma^2}} \tag{4}$$

where P and $\tilde{P}$ are considered as vectors with κ^2 components and $||P||$ denotes the Euclidean norm of P. Knowing $\tilde{P}$, our goal is to deduce P by maximizing $\mathbb{P}(P|\tilde{P})$. Using Bayes' rule, we can compute this last conditional probability as

$$\mathbb{P}(P|\tilde{P}) = \frac{\mathbb{P}(\tilde{P}|P)\mathbb{P}(P)}{\mathbb{P}(\tilde{P})}. \tag{5}$$

$\tilde{P}$ being observed, this formula can in principle be used to deduce the patch P maximizing the right term, viewed as a function of P. This is only possible if we have a probability model for P, and these models will be generally learnt from the image itself, or from a set of images. For example [33] applies a clustering method to the set of patches of a given image, and [161] applies it to a huge set of patches extracted from many images. Each cluster of patches is thereafter treated as a set of Gaussian samples. This permits to associate to each observed patch its likeliest cluster, and then to denoise it by a Bayesian estimation in this cluster. Another still more direct way to build a model for a given patch $\tilde{P}$ is to group the patches similar to $\tilde{P}$ in the image. Assuming that these similar patches are samples of a Gaussian vector yields a standard Bayesian restoration [86]. We shall now discuss this particular case, where all observed patches are noisy.

Why Gaussian? As usual when we dispose of several observations but of no particular guess on the form of the probability density, a Gaussian model is adopted. In the case of the patches Q similar to a given patch P, the Gaussian model has some pertinence, as it is assumed that many contingent random factors explain the difference between Q and P: other details, texture, slight lighting changes, shadows, etc. The Gaussian model in presence of a combination of many such random and independent factors is heuristically justified by the central limit theorem. Thus, for good or bad, assume that the patches Q similar to P follow a Gaussian model with (observable, empirical) covariance matrix $\mathbf{C}_P$ and (observable, empirical) mean $\overline{P}$. This means that

$$\mathbb{P}(Q) = c.e^{-\frac{(Q-\overline{P})^t \mathbf{C}_P^{-1}(Q-\overline{P})}{2}} \tag{6}$$

From (3) and (5) we obtain for each observed $\tilde{P}$ the following equivalence of problems:

$$\begin{aligned}
\max_P \ \mathbb{P}(P|\tilde{P}) &\Leftrightarrow \max_P \ \mathbb{P}(\tilde{P}|P)\mathbb{P}(P) \\
&\Leftrightarrow \max_P e^{-\frac{\|P-\tilde{P}\|^2}{2\sigma^2}} e^{-\frac{(P-\overline{P})^t \mathbf{C}_P^{-1}(P-\overline{P})}{2}} \\
&\Leftrightarrow \min_P \ \frac{\|P-\tilde{P}\|^2}{\sigma^2} + (P-\overline{P})^t \mathbf{C}_P^{-1}(P-\overline{P}).
\end{aligned}$$

This expression does not yield an algorithm. Indeed, the noiseless patch P and the patches similar to P are not observable. Nevertheless, we can observe the noisy version $\tilde{P}$ and compute the patches $\tilde{Q}$ similar to $\tilde{P}$. An empirical covariance matrix can therefore be obtained for the patches $\tilde{Q}$ similar to $\tilde{P}$. Furthermore, using (2) and the fact that P and the noise n are independent,

$$\mathbf{C}_{\tilde{P}} = \mathbf{C}_P + \sigma^2 \mathbf{I}; \qquad E\tilde{Q} = \overline{P}. \tag{7}$$

Notice that these relations assume that we searched for patches similar to $\tilde{P}$ at a large enough distance, to include all patches similar to P, but not too large either, because otherwise it can contain outliers. Thus the safe strategy is to search similar patches in a distance slightly larger than the expected distance caused by noise. If the above estimates are correct, our MAP (maximum *a posteriori* estimation) problem finally boils down by (7) to the following feasible minimization problem:

$$\max_P \ \mathbb{P}(P|\tilde{P}) \Leftrightarrow \min_P \ \frac{\|P-\tilde{P}\|^2}{\sigma^2} + (P-\overline{\tilde{P}})^t(\mathbf{C}_{\tilde{P}}-\sigma^2\mathbf{I})^{-1}(P-\overline{\tilde{P}}).$$

Differentiating this quadratic function with respect to P and equating to zero yields

$$P-\tilde{P}+\sigma^2(\mathbf{C}_{\tilde{P}}-\sigma^2\mathbf{I})^{-1}(P-\overline{\tilde{P}})=0.$$

Taking into account that $\mathbf{I}+\sigma^2(\mathbf{C}_{\tilde{P}}-\sigma^2\mathbf{I})^{-1}=(\mathbf{C}_{\tilde{P}}-\sigma^2\mathbf{I})^{-1}\mathbf{C}_{\tilde{P}}$, this yields

$$(\mathbf{C}_{\tilde{P}}-\sigma^2\mathbf{I})^{-1}\mathbf{C}_{\tilde{P}}P=\tilde{P}+\sigma^2(C_{\tilde{P}}-\sigma^2\mathbf{I})^{-1}\overline{\tilde{P}}.$$

and therefore

$$\begin{aligned} P &= \mathbf{C}_{\tilde{P}}^{-1}(\mathbf{C}_{\tilde{P}}-\sigma^2\mathbf{I})\tilde{P}+\sigma^2\mathbf{C}_{\tilde{P}}^{-1}\overline{\tilde{P}} \\ &= \tilde{P}+\sigma^2\mathbf{C}_{\tilde{P}}^{-1}(\overline{\tilde{P}}-\tilde{P}) \\ &= \overline{\tilde{P}}+\left[\mathbf{I}-\sigma^2\mathbf{C}_{\tilde{P}}^{-1}\right](\tilde{P}-\overline{\tilde{P}}) \\ &= \overline{\tilde{P}}+\left[\mathbf{C}_{\tilde{P}}-\sigma^2\mathbf{I}\right]\mathbf{C}_{\tilde{P}}^{-1}(\tilde{P}-\overline{\tilde{P}}) \end{aligned}$$

Thus we have proved that a restored patch $\hat{P}_1$ can be obtained from the observed patch $\tilde{P}$ by the one step estimation

$$\hat{P}_1=\overline{\tilde{P}}+\left[\mathbf{C}_{\tilde{P}}-\sigma^2\mathbf{I}\right]\mathbf{C}_{\tilde{P}}^{-1}(\tilde{P}-\overline{\tilde{P}}), \tag{8}$$

which resembles a local Wiener filter.

Remark 1. *It is easily deduced that the expected estimation error is*

$$E||P-\hat{P}_1||^2=Tr\left[\left(\boldsymbol{C}_P^{-1}+\frac{\boldsymbol{I}}{\sigma^2}\right)^{-1}\right].$$

Sections 5.2, 5.3, 5.4, 5.5, 5.6, 5.9 will examine not less than **six Bayesian algorithms** deriving patch-based denoising algorithms from variants of (8). The first question when looking at this formula is obviously how the matrix $\mathbf{C}_{\tilde{P}}$ can be learnt from the image itself. Each method proposes a different notion to learn the patch model.

Of course, other, non Gaussian, Bayesian models are possible, depending on the patch density assumption. For example [132] assumes a local exponential density model for the noisy data, and gives a convergence proof to the optimal (Bayes) least squares estimator as the amount of data increases.

3.2 Transform thresholding

Classical transform coefficient thresholding algorithms like the DCT or the wavelet denoising use the observation that images are faithfully described by keeping only their large coefficients in a well-chosen basis. By keeping these large coefficients and setting to zero the small ones, noise should be removed and image geometry kept. By any orthogonal transform, the coefficients of an homoscedastic de-correlated noise remain de-correlated and homoscedastic. For example the wavelet or the DCT coefficients of a Gaussian white noise with variance σ^2 remain a Gaussian diagonal vector with variance σ^2. Thus, a threshold on the coefficients at, say, 3σ removes most of the coefficients that are only due to noise. (The expectation of these coefficients is assumed to

be zero.) The *sparsity* of image coefficients in certain bases is only an empirical observation. It is nevertheless invoked in most denoising and compression algorithms, which rely essentially on coefficient thresholds. The established image compression algorithms are based on the DCT (in the JPEG 1992 format) or, like the JPEG 2000 format [4], on biorthogonal wavelet transforms [35].

Let $\mathcal{B} = \{G_i\}_{i=1}^M$ be an orthonormal basis of $\mathbb{R}^M$, where M is the number of pixels of the noisy image $\tilde{U}$ (in staircase to recall that it is handled here as a vector). Then we have

$$\langle \tilde{U}, G_i \rangle = \langle U, G_i \rangle + \langle N, G_i \rangle , \tag{9}$$

where $\tilde{U}$, U and N denote respectively the noisy, original and noise images. We always assume that the noise values $N(\mathbf{i})$ are uncorrelated and homoscedastic with zero mean and variance σ^2. The following calculation shows that the noise coefficients in the new basis remain uncorrelated, with zero mean and variance σ^2:

$$\begin{aligned} E[\langle N, G_i \rangle \langle N, G_j \rangle] &= \sum_{\mathbf{r},\mathbf{s}=1}^{M} G_i(\mathbf{r}) G_j(\mathbf{s}) E[\mathbf{w}(\mathbf{r})\mathbf{w}(\mathbf{s})] \\ &= \langle G_i, G_j \rangle \sigma^2 = \sigma^2 \delta[j-i]. \end{aligned}$$

Each noisy coefficient $\langle \tilde{U}, G_i \rangle$ is modified independently and then the solution is estimated by the inverse transform of the new coefficients. Noisy coefficients are modified by multiplying by an attenuation factor $a(i)$ and the inverse transform yields the estimate

$$\mathbf{D}\tilde{U} = \sum_{i=1}^{M} a(i) \langle \tilde{U}, G_i \rangle G_i. \tag{10}$$

$\mathbf{D}$ is also called a *diagonal operator*. Noise reduction is achieved by attenuating or setting to zero small coefficients of order σ, assumedly due to noise, while the original signal is preserved by keeping the large coefficients. This intuition is corroborated by the following result.

Theorem 1. *The operator $\boldsymbol{D}_{inf}$ minimizing the mean square error (MSE),*

$$\boldsymbol{D}_{inf} = arg\min_{\boldsymbol{D}} E\{\|U - \boldsymbol{D}\tilde{U}\|^2\}$$

is given by the family $\{a(i)\}_i$, where

$$a(i) = \frac{|\langle U, G_i \rangle|^2}{|\langle U, G_i \rangle|^2 + \sigma^2}, \tag{11}$$

and the corresponding expected mean square error (MSE) is

$$E\{\|U - \boldsymbol{D}_{inf}\tilde{U}\|^2\} = \sum_{i=1}^{M} \frac{|\langle U, G_i \rangle|^2 \sigma^2}{|\langle U, G_i \rangle|^2 + \sigma^2}. \tag{12}$$

The previous optimal operator attenuates all noisy coefficients. If one restricts $a(i)$ to be 0 or 1, one gets a projection operator. In that case, a subset of coefficients is kept, and the rest are set to zero. The projection operator that minimizes the MSE under that constraint is obtained with

$$a(i) = \begin{cases} 1 & |\langle U, G_i \rangle|^2 \geq \sigma^2, \\ 0 & \text{otherwise} \end{cases}$$

and the corresponding MSE is

$$E\{\|U - \mathbf{D}_{inf}\tilde{U}\|^2\} = \sum_i \min(|\langle U, G_i \rangle|^2, \sigma^2). \tag{13}$$

A *transform thresholding* algorithm therefore keeps the coefficients with a magnitude larger than the noise, while setting the zero the rest. Note that both above mentioned filters are "ideal", or "oracular" operators. Indeed, they use the coefficients $\langle U, G_i \rangle$ of the original image, which are not known. These algorithms are therefore usually called *oracle filters*. We shall discuss their implementation in the next sections. For the moment, we shall introduce the classical thresholding filters, which approximate the oracle coefficients by using the noisy ones.

We call, as is classical, *Fourier–Wiener filter* the optimal operator (11) when $\mathcal{B}$ is a Fourier basis. By the use of the Fourier basis, global image characteristics may prevail over local ones and create spurious periodic patterns. To avoid this effect, the bases are usually more local, of the wavelet or block DCT type.

Sliding window DCT. The local adaptive filters were introduced by Yaroslavsky and Eden [152] and Yaroslavsky [154]. The noisy image is analyzed in a moving window, and at each position of the window its DCT spectrum is computed and modified by using the optimal operator (11). Finally, an inverse transform is used to estimate only the signal value in the central pixel of the window.

This method is called the *empirical Wiener filter*, because it approximates the unknown original coefficients $\langle u, G_i \rangle$ by using the identity

$$E|\langle \tilde{U}, G_i \rangle|^2 = |\langle U, G_i \rangle|^2 + \sigma^2$$

and thus replacing the optimal attenuation coefficients $a(i)$ by the family $\{\alpha(i)\}_i$,

$$\alpha(i) = \max \left\{ 0, \frac{|\langle \tilde{U}, G_i \rangle|^2 - c\sigma^2}{|\langle \tilde{U}, G_i \rangle|^2} \right\}.$$

where c is a parameter, usually larger than one.

Wavelet thresholding. Let $\mathcal{B} = \{G_i\}_i$ be a wavelet orthonormal basis [107]. The so-called *hard wavelet thresholding method* [54] is a (nonlinear) projection operator setting to zero all wavelet coefficients smaller than a certain threshold. According to the expression of the MSE of a projection operator (13), the performance of the method depends on the ability of the basis to approximate the image U by a small set of large coefficients. There has been a strenuous search for wavelet bases adapted to images [124].

Unfortunately, not only noise, but also image features can cause many small wavelet coefficients, which are nevertheless lower than the threshold. The brutal cancelation of wavelet (or DCT) coefficients near the image edges creates small oscillations, a Gibbs phenomenon often called *ringing*. Spurious wavelets can also be seen in flat parts of the restored image, caused by the undue cancelation of some of the small coefficients. These artifacts are sometimes called *wavelet outliers* [55]. These undesirable effects can be partially avoided with the use of a soft thresholding [52],

$$\alpha(i) = \begin{cases} \frac{\langle \tilde{U}, G_i \rangle - sgn(\langle \tilde{U}, G_i \rangle)\mu}{\langle \tilde{U}, G_i \rangle}, & |\langle \tilde{U}, G_i \rangle| \geq \mu, \\ 0 \quad \text{otherwise}, \end{cases}$$

The continuity of this soft thresholding operator reduces the Gibbs oscillation near image discontinuities.

Several orthogonal bases adapt better to image local geometry and discontinuities than wavelets, particularly the "bandlets" [124] and "curvelets" [139]. This tendency to adapt the transform locally to the image is accentuated with the methods adapting a different basis to each pixel, or selecting a few elements or "atoms" from a huge patch dictionary to linearly decompose the local patch on these atoms. This point of view is sketched in the next section on sparse coding.

3.3 Sparse coding

Sparse coding algorithms learn a redundant set $\mathbf{D}$ of vectors called *dictionary* and choose the right atoms to describe the current patch.

For a fixed patch size, the dictionary is encoded as a matrix of size $\kappa^2 \times n_{dic}$, where κ^2 is the number of pixels in the patch and $n_{dic} \geq \kappa^2$. The dictionary patches, which are columns of the matrix, are normalized (in Euclidean norm). This dictionary may collect usual orthogonal bases (discrete cosine transform, wavelets, curvelets ...), but also patches extracted (or learnt) from clean images or even from the noisy image itself.

The dictionary permits to compute a sparse representation α of each patch P, where α is a coefficient vector of size n_{dic}^2 satisfying $P \approx \mathbf{D}\alpha$. This sparse representation α can be obtained with an ORMP (orthogonal recursive matching pursuit) [37]. ORMP gives an approximate solution to the (NP-complete) problem

$$\underset{\alpha}{\text{Arg min}} ||\alpha||_0 \quad \text{such that} \quad ||P - \mathbf{D}\alpha||_2^2 \leq \kappa^2 (C\sigma)^2 \tag{14}$$

where $\|\alpha\|_0$ refers to the l^0 norm of α, i.e. the number of non-zero coefficients of α. This last constraint brings in a new parameter C. This coefficient multiplying the standard deviation σ guarantees that, with high probability, a white Gaussian noise of standard deviation σ on κ^2 pixels has an l^2 norm lower than $\kappa C\sigma$. The ORMP algorithm is introduced in [37]. Details on how this minimization can be achieved are given in the section describing the K-SVD algorithm 5.7. (It has been argued that the l^0 norm of the set of coefficients can be replaced by the much easier l^1 convex norm. This remark is the starting point of the compressive sampling method [29].)

In K-SVD and other current sparse coding algorithms, the previous denoising strategy is used as the first step of a two-steps algorithm. The selection step is iteratively combined with an update of the dictionary taking into account the image and the sparse codifications already computed. More details will be found in section 5.7 on the K-SVD algorithm.

Several of our referees have objected to considering *sparse coding* and *transform thresholding* as two different denoising principles. As models, both indeed assume the *sparsity* of patches in some well chosen basis. Nevertheless, some credit must be given to historical development. The notion of sparsity is associated with a recent and sophisticated variational principle, where the dictionary and the sparse decompositions are computed simultaneously. Transform thresholding methods existed before the term sparsity was even used. They simply pick a local wavelet or DCT basis and threshold the coefficients. In both algorithms, the sparsity is implicitly or explicitly assumed. But transform threshold methods use orthogonal bases, while the dictionaries are redundant. Furthermore, the algorithms are very different.

3.4 Image self-similarity leading to pixel averaging

The principle of many denoising methods is quite simple: they replace the colour of a pixel with an average of the colours of nearby pixels. It is a powerful and basic principle, when applied directly on noisy pixels with independent noise. If m pixels with the same colour (up to the fluctuations due to noise) are averaged the noise is reduced by a $\sqrt{m}$ factor.

The MSE between the true (unknown) value $u(\mathbf{i})$ of a pixel $\mathbf{i}$ and the value estimated by a weighted average of pixels $\mathbf{j}$ is

$$\begin{aligned} E\|u(\mathbf{i}) - \sum_{\mathbf{j}} w(\mathbf{j})\tilde{u}(\mathbf{j})\|^2 &= E\|\sum_{\mathbf{j}} w(\mathbf{j})(u(\mathbf{i}) - u(\mathbf{j})) - \sum_{\mathbf{j}} w(\mathbf{j})n(\mathbf{j})\|^2 \\ &= \sum_{\mathbf{j}} w(\mathbf{j})^2 (u(\mathbf{i}) - u(\mathbf{j}))^2 + \sigma^2 \sum_{\mathbf{j}} w(\mathbf{j})^2, \end{aligned} \tag{15}$$

where we assume that the noise, the image and the weights are independent and that the weights $\{w(\mathbf{j})\}_{\mathbf{j}}$ satisfy $\sum_{\mathbf{j}} w(\mathbf{j}) = 1$.

The above expression implies that the performance of the averaging depends on the ability to find many pixels $\mathbf{j}$ with an original value $u(\mathbf{j})$ close to $u(\mathbf{i})$. Indeed, the variance term $\sum_{\mathbf{j}} w(\mathbf{j})^2$

is minimized by a flat distribution probability $w(\mathbf{j}) = 1/m$, where m is the number of averaged pixels. The first term measures the bias caused by the fact that pixels do not have exactly the same deterministic value. Each method must find a tradeoff between the bias and variance terms of equation (15).

Averaging of spatially close pixels A first rather trivial idea is to average the closest pixels to a given pixel. This amounts to convolve the image with a fixed radial positive kernel. The paradigm of such kernels is the Gaussian kernel.

The convolution of the image with a Gaussian kernel ensures a fixed noise standard deviation reduction factor that equals the kernel standard deviation. Yet, nearby pixels do not necessarily share their colours. Thus, the first error term in (15) can quickly increase. This approach is valid only for pixels for which the nearby pixels have the same colour, that is, it only works inside the homogeneous image regions, but not for their boundaries.

Averaging pixels with similar colours A simple solution to the above mentioned dilemma is given by the sigma-filter [90] or neighborhood filter [153]. These filters average only nearby pixels of $\mathbf{i}$ having also a similar colour value. We shall denote these filters by *YNF*, (for Yaroslavsky neighborhood filter). Their formula is simply

$$YNF_{h,\rho}\tilde{u}(\mathbf{i}) = \frac{1}{C(\mathbf{i})} \sum_{\mathbf{j}\in B_\rho(\mathbf{i})} \tilde{u}(\mathbf{j})\, e^{-\frac{|\tilde{u}(\mathbf{i})-\tilde{u}(\mathbf{j})|^2}{h^2}}, \tag{16}$$

where $B_\rho(\mathbf{i})$ is a ball of center $\mathbf{i}$ and radius $\rho > 0$, $h > 0$ is the filtering parameter and $C(\mathbf{i}) = \sum_{\mathbf{j}\in B_\rho(\mathbf{i})} e^{-\frac{|\tilde{u}(\mathbf{j})-\tilde{u}(\mathbf{i})|^2}{h^2}}$ is the normalization factor. The parameter h controls the degree of colour similarity needed to be taken into account in the average. According to the Bayesian interpretation of the filter we should have $h = \sigma$. The filter (16), due to Yaroslavsky and Lee, has been reinvented several times, and has received the alternative names of *SUSAN filter* [138] and of *Bilateral filter* [142]. The relatively minor difference in these algorithms is that instead of considering a fixed spatial neighborhood $B_\rho(\mathbf{i})$, they weigh the spatial distance to the reference pixel $\mathbf{i}$ by a Gaussian.

Neighborhood filters choose the "neighboring" pixels by comparing their noisy colour. The weight distribution is therefore computed by using noisy values and is not independent of the noise. Therefore the error formula (15) is not applicable. For a flat zone and for a given pixel with colour value a, the nearby pixels with an intensity difference lower than h will be independent and identically distributed with a probability distribution which is the restriction of the Gaussian to the interval $(a-h, a+h)$. If the search zone (or spatial neighborhood) is broad enough, then the average value will tend to the expectation of this random variable. Thus, the increase of the search zone and therefore of the number of pixels being averaged beyond a reasonable value will not increase the noise reduction capability of the filter. More precisely, the asymptotic noise reduction factor is given in the next theorem, taken from [15].

Theorem 2. *Assume that $n(\boldsymbol{i})$ are i.i.d. with zero mean and variance σ^2, then a noise n filtered by the neighborhood filter YNF_h satisfies*

$$Var\, YNF_{h,\rho}\, n = f\left(\frac{h}{\sigma}\right)\sigma^2,$$

where

$$f(x) = \frac{1}{(2\pi)^{3/2}} \int_{\mathbb{R}} \frac{1}{\beta^2(a,x)} (e^{2xa}-1)^2 e^{(a+x)^2} e^{\frac{-a^2}{2}}\, da$$

and

$$\beta(a,x) = \frac{1}{\sqrt{2\pi}} \int_{a-x}^{a+x} e^{-t^2/2}\, dt\,.$$

The function $f(x)$ is a decreasing function with $f(0) = 1$ and $\lim_{x\to\infty} f(x) = 0$ (see plot in Fig. 5). The noise reduction increases with the ratio h/σ. We see that $f(x)$ is close to zero for values of x over 2.5 or 3, that is, values of h over $2.5\,\sigma$ or $3\,\sigma$. This corresponds to the values proposed in the original papers by Lee and Yaroslavsky. However, for a Gaussian variable, the probability of observing values at a distance to the average above 2.5 or 3 times the standard deviation is very small. Thus, taking these large values excessively increases the probability of mismatching pixels belonging in fact to other objects. This explains the observed decaying performance of the neighborhood filter when the noise standard deviation or the search zone $B(\mathbf{i}, \rho)$ increase too much.

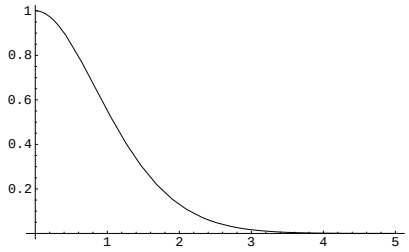


Figure 5: Noise reduction function $f(x)$ given by Theorem 2.

The image model underlying neighborhood filters is the *image self-similarity*, namely the presence in the image of pixels **j** which have the same law as **i**. We will introduce in section 5.1 the NL-means algorithm [17] which can be seen as an extension of the neighborhood filters attenuating their main drawbacks. In NL-means, the "neighborhood of a pixel **i**" is defined as any set of pixels **j** in the image such that a patch around **j** looks like a patch around **i**. In other terms NL-means estimates the value of **i** as an average of the values of all the pixels **j** whose neighborhood looks like the neighborhood of **i**.

4 Noise reduction, generic tools

This section a describes four generic tools that permit to increase the performance of *any* denoising principle. We shall illustrate them on DCT denoising. Starting from the application of a simple DCT transform threshold, the four generic tools will applied successively. We shall observe a dramatic improvement of the denoising performance. This observation is valid for all denoising principles.

4.1 Aggregation of estimates

Aggregation techniques combine for any pixel a set of m possible estimates. If these estimates were independent and had equal variance, then a uniform average would reduce this estimator variance by a factor m. Such an aggregation strategy was the main proposition of the *translation invariant wavelet thresholding algorithm* [36]. This method denoises several translations of the image by a wavelet thresholding algorithm and averages these different estimates once the inverse translation has been applied to the denoised images.

An interesting case is when one is able to estimate the variance of the m estimators. Statistical arguments lead to attribute to each estimator a weight inversely proportional to its variance [118]. For most denoising methods the variance of the estimators is high near image edges. When applied without aggregation, the denoising methods leave visible "halos" of residual noise near edges. For example in the sliding window DCT method, patches containing edges have many large DCT coefficients which are kept by thresholding. In flat zones instead, most DCT coefficients are canceled and the noise is completely removed. The proposition of [71] is to use the aggregation for DCT denoising, approximating the variance of each estimated patch by the number of non

zero coefficients after thresholding. In the online paper [72] one can test an implementation of DCT denoising. It actually uses an aggregation with uniform weights: "translation invariant DCT denoising is implemented by decomposing the image to sliding overlapping patches, calculating the DCT denoising in each patch, and then aggregating the denoised patches to the image averaging the overlapped pixels. The translation invariant DCT denoising significantly improves the denoising performance, typically from about 2 to 5 dB, and removes the block artifact".

The same risk of "halo" occurs with non-aggregated NL-means (section 5.1), since patches containing edges have many less similar instances in the image than flat patches. Thus the non-local averaging is made over less samples, and the final result keeps more noise near image edges. The same phenomenon occurs with BM3D (section 5.8) if the aggregation step is not applied [39]. As a consequence, an aggregation step is applied in all patch-based denoising algorithms. This weighted aggregation favors, at each pixel near an edge, the estimates given by patches which contain the pixel but do not meet the edge.

Aggregation techniques aim at a superior noise reduction by increasing the number of values being averaged for obtaining the final estimate or selecting those estimates with lower variance. Kervrann et al [82] considered the whole Bias+Variance decomposition in order to also adapt the search zone of neighborhood filters or of NL-means. Since the bias term depends on the original image, it cannot be computed in practice, and Kervrann et al. proposed to minimize both bias and variance by choosing the smallest spatial neighborhood attaining a stable noise reduction.

Another type of aggregation technique considers the risk estimate rather than the variance to locally attribute more weight to the estimators with small risks. In [144], Van De Ville and Kocher give a closed form expression of Stein's Unbiased Estimator of the Risk (SURE) for NL-Means. (See also generalizations of the SURE estimator to the non-Gaussian case in [131].) The aim is to select globally the best bandwidth for a given image. In [56], Duval et al. also use the SURE technique for minimizing the risk by selecting locally the bandwidth. Deledalle et al. [48] apply the same technique for combining the results of NL-means with different window sizes and shapes. A similar treatment can be found in [132], but with the assumption of a local exponential density for the noisy patches.

4.2 Iteration and "oracle" filters

Iterative strategies to remove residual noise would drift from the initial image. Instead, a first step denoised image can be used to improve the reapplication of the denoising method to the initial noisy image. In a second step application of a denoising principle, the denoised DCT coefficients, or the patch distances, can be computed in the first step denoised image. They are an approximation to the true measurements that would be obtained from the noise-free image. Thus, the first step denoised image is used as an "oracle" for the second step.

For averaging filters such as neighborhood filters or NL-means, the image u can be denoised in a first step by the method under consideration. This first step denoised image denoted by $\hat{u}_1$ is used for computing more accurate colour distances between pixels. Thus, the second step neighborhood filter is

$$YNF_{h,\rho}\tilde{u}(\mathbf{i}) = \frac{1}{C(\mathbf{i})} \sum_{\mathbf{j}\in B_\rho(\mathbf{j})} \tilde{u}(\mathbf{j}) e^{-\frac{|\hat{u}_1(\mathbf{j})-\hat{u}_1(\mathbf{i})|^2}{h^2}},$$

where $\tilde{u}$ is the observed noisy image and $\hat{u}_1$ the image previously denoised by (16).

Similarly, for linear transform Wiener-type methods, the image is first denoised by its classical definition, which amounts to approximate the oracle coefficients of Theorem 1 using the noisy ones. In a second iteration, the coefficients of the denoised image approximate the true coefficients of the noise-free image. Thus, the second step filter following the first step (10) is

$$\mathbf{D}\tilde{U} = \sum_i a(i)\ \langle \tilde{U}, G_i\rangle\ G_i\,, \quad \text{with} \quad a(i) = \frac{|\langle \hat{U}_1, G_i\rangle|^2}{|\langle \hat{U}_1, G_i\rangle|^2 + \sigma^2},$$

where $\hat{U}_1$ is the denoised image by applying a first time the thresholding algorithm to the observed image $\tilde{U}$.

Alternatives and extensions: "twicing" and Bregman iterations In the recent review paper [116], many denoising operators are formalized in a general linear framework, noting that they can be associated with a doubly stochastic diffusion matrix W with nonnegative coefficients. For example in NL-means, this matrix is obtained by the symmetrization of the matrix of the NL-means weights $w_{\tilde{P},\tilde{Q}}$ defined in Algorithm 1. Unless it is optimal, as is the case with an ideal Wiener filter, the matrix W associated with the denoising filter can be iterated. A study of MSE evolution with these iterations is proposed in [116] for several denoising operators, considering several different patch types (texture, edge, flat). Iteration is, however, different from the oracle iteration described above. In the oracle iteration, the matrix W is changed at each step, using its better estimate given by the previously denoised image. One does not generally observe much improvement by iterating the oracle method more than once. [116] points out another generic tool, used at least for total variation denoising, the so-called "twicing", term due to Tukey [143]. Instead of repeated applications of a filter, the idea is to process the residual obtained as the difference between the estimated image and the initial image. If the residuals contain some of the underlying signal, filtering them should recover part of it. The author shows that the Bregman iterations [123] used for improving total variation based denoising are a twicing and so is the matching pursuit method used in the K-SVD filter described in section 5.7.

4.3 Dealing with colour images

The straightforward strategy to extend denoising algorithms to colour or multivalued images is to apply the algorithm independently to each channel. The use of this simple strategy often introduces colour artifacts, easily detected by the eye. Two different strategies are observable in state of the art denoising algorithms.

Depending on the algorithm formulation, a vector-valued version dealing at the same time with all colour channels can be proposed. This solution is adopted by averaging filters like neighborhood filters or NL-means. These algorithms compute colour differences directly in the vector valued image, thus yielding a unified weight configuration which is applied to each channel.

The alternative option is to convert the usual RGB image to a different colour space where the independent denoising of each channel does not create noticeable colour artifacts. Most algorithms use the YUV system which separates the geometric and chromatic parts of the image. This change writes as a linear transform by multiplication of the RGB vector by the matrix

$$YUV = \begin{pmatrix} 0.30 & 0.59 & 0.11 \\ -0.15 & -0.29 & 0.44 \\ 0.61 & -0.51 & -0.10 \end{pmatrix}, \qquad Y_oU_oV_o = \begin{pmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{2} & 0 & -\frac{1}{2} \\ \frac{1}{4} & -\frac{1}{2} & \frac{1}{4} \end{pmatrix}$$

The second colour transform to the space $Y_oU_oV_o$ is an orthogonal transform. It has the advantage of maximizing the noise reduction of the geometric component, since this component is an average of the three colours. The geometric component is perceptually more important than the chromatic ones, and the presence of less noise permits a better performance of the algorithm in this part. It also permits a higher noise reduction on the chromatic components U_o and V_o, due to their observable regularity.

This latter strategy is adopted by transform thresholding filters for which the design of an orthonormal basis coupling the different colour channels is not trivial.

4.4 Trying all generic tools on an example

This section applies incrementally the previous generic denoising tools to the DCT sliding window to illustrate how these additional tools permit to drastically improve the algorithm performance. We start with the basic DCT "neighborhood filter" as proposed by Yaroslavsky [152]. Its principle is to denoise a patch around each pixel, and to keep only the central denoised pixel.

Fig. 6 displays the denoised images obtained by incrementally applying each of the following ingredients:

- Basic DCT thresholding algorithm by the neighborhood filter technique (keeping only the central pixel of the window). Each colour channel is treated independently.

- Use of an orthogonal geometric and chromatic decomposition colour system $Y_oU_oV_o$; grey parts are better reconstructed and colour artifacts are reduced.

- Uniform aggregation; the noise reduction is superior and isolated noise points are removed.

- Adaptive aggregation using the estimator variance; the noise reduction near edges is increased, "halo" effects are removed.

- Additional iteration using "oracle" estimation; residual noise is totally removed and the sharpness of details is increased.

The $PSNR$'s obtained by incrementally applying the previous strategies respectively are 26.85, 27.33, 30.65, 30.73, 31.25. This experiment illustrates how the use of these additional tools is crucial to achieve competitive results. This last version of the DCT denoising algorithm, incorporating all the proposed generic tools, will be the one used in the comparison section. A complete description of the algorithm can be found in Algorithm 3. The colour version of the algorithm applies the denoising independently to each $Y_oU_oV_o$ component. This version is therefore slightly better than the version online in [72], which does not use the oracle step.

Algorithm 3 DCT denoising algorithm. DCT coefficients lower than 3σ are canceled in the first step and a Wiener filter is applied in the "oracle" second step. The colour DCT denoising algorithm applies the current strategy independently to each $Y_oU_oV_o$ component.

Input: noisy image $\tilde{u}$, σ noise standard deviation.
Optional: prefiltered image $\hat{u}_1$ for "oracle" estimation.
Output: output denoised image.

Set parameter $\kappa = 8$: size of patches.
Set parameter $h = 3\,\sigma$: threshold parameter.

for each pixel **i** **do**

Select a square reference patch $\tilde{P}$ around **i** of size $\kappa \times \kappa$.
if $\hat{u}_1$ **then**
Select a square reference patch P_1 around **i** in $\hat{u}_1$.
end if
Compute the DCT transform of $\tilde{P}$.
if $\hat{u}_1$ **then**
Compute the DCT transform of P_1.
end if
if $\hat{u}_1$ **then**
Modify DCT coefficients of $\tilde{P}$ as

$$\tilde{P}(i) = \tilde{P}(i)\frac{P_1(i)^2}{P_1(i)^2 + \sigma^2}$$

else
Cancel coefficients of $\tilde{P}$ with magnitude lower than h.
end if
Compute the inverse DCT transform obtaining $\hat{P}$.
Compute the aggregation weight $w_{\tilde{P}} = 1/\#\{\text{number of non-zero } DCT \text{ coefficients}\}$.
end for

for each pixel **i** **do**
Aggregation: recover the denoised value at each pixel **i** by averaging all values at **i** of all denoised patches $\hat{Q}$ containing **i**, weighted by $w_{\tilde{Q}}$.
end for

Figure 6: Top: original and noisy images with an additive Gaussian white noise of standard deviation 25. Below, from top to bottom and left to right: crop of denoised images by sliding DCT thresholding filter and incrementally adding use of a $Y_oU_oV_o$ colour system, uniform aggregation, variance based aggregation and iteration with the "oracle" given by the first step. The corresponding PSNR are 26.85, 27.33, 30.65, 30.73, 31.25.

5 Detailed analysis of nine methods

In this section, we give a detailed description and analysis of nine denoising methods. Six of them, for which reliable faithful implementations are available, will be compared in section 6.

5.1 Non-local means

The Non-local means (NL-means) algorithm tries to take advantage of the redundancy of most natural images. The redundancy, or self-similarity hypothesis is that for every small patch in a natural image one can find several similar patches in the same image, as illustrated in figures 7 and 8. This similarity is true for patches whose centers stand at a one pixel distance of the center of the reference patch. In that case the self-similarity boils down to a local image regularity assumption. Such a regularity is guaranteed by Shannon-Nyquist's sampling conditions, which require the image to be blurry. In a much more general sense inspired by neighborhood filters, one can define as "neighborhood of a pixel **i**" any set of pixels **j** in the image such that a patch around **j** looks like a patch around **i**. All pixels in that neighborhood can be used for predicting the value at **i**, as was first shown in [61] for the synthesis of texture images. This self-similarity hypothesis is a generalized periodicity assumption. The use of self-similarities is actually well-known in information theory from its foundation. In his 1948 Mathematical Theory of Communication, Shannon [136] analyzed the local self-similarity (or redundancy) of natural written language, and gave probably the first stochastic text synthesis algorithm. The Efros-Leung texture synthesis method adapted this algorithm to images, and NL-Means [18] seems to be first adaptation of the same idea to denoising[2]

[2]Nevertheless, some researchers have pointed out to us the report [46] as giving an early intuition that intuition could use signal redundancy. This very short paper describes an experiment in a few sentences. It suggests that

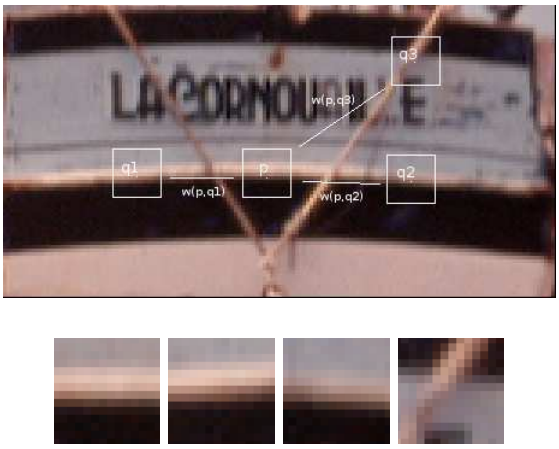


Figure 7: $q1$ and $q2$ have a large weight because their similarity windows are similar to that of p. On the other side the weight $w(p, q3)$ is much smaller because the intensity grey values in the similarity windows are very different.

NL-means denoises a square reference patch $\tilde{P}$ around $\mathbf{i}$ of dimension $\kappa \times \kappa$ by replacing it by an average of all similar patches $\tilde{Q}$ in a square neighborhood of $\mathbf{i}$ of size $\lambda \times \lambda$. To do this, a normalized Euclidean distance between $\tilde{P}$ and $\tilde{Q}$, $d(\tilde{P}, \tilde{Q}) = \frac{1}{\kappa^2}\|\tilde{P} - \tilde{Q}\|^2$ is computed for all patches $\tilde{Q}$ is the search neighborhood. Then the weighted average is

$$\hat{P} = \frac{\sum_{\tilde{Q}} \tilde{Q} e^{-\frac{d(\tilde{P},\tilde{Q})^2}{h^2}}}{\sum_{\tilde{Q}} e^{-\frac{d(\tilde{P},\tilde{Q})^2}{h^2}}}.$$

The thing that helps NL-means over the neighborhood filters is the concentration of the noise law, as the number of pixels increases. Because the distances are computed on many patch samples instead of only one pixel, the fluctuations of the quadratic distance due to the noise are reduced.

Related attempts: [147] proposed a "universal denoiser" for digital images. The authors prove that this denoiser is universal in the sense "of asymptotically achieving, without access to any information on the statistics of the clean signal, the same performance as the best denoiser that does have access to this information". In [122] the authors presented an implementation valid for binary images with an impulse noise, with excellent results. Awate and Whitaker [7] also proposed a method whose principles stand close to NL-means, since the method involves comparison between patches to estimate a restored value. The objective of the algorithm is to denoise the image by decreasing the randomness of the image.

A consistency theorem for NL-means. NL-means is intuitively consistent under stationarity conditions, namely if one can find many samples of every image detail. It can be proved [24] that if the image is a fairly general stationary and mixing random process, for every pixel $\mathbf{i}$, NL-means converges to the conditional expectation of $\mathbf{i}$ knowing its neighborhood, which is the best Bayesian estimate.

NL-means as an extension of previous methods. A Gaussian convolution preserves only flat zones, while contours and fine structure are removed or blurred. Anisotropic filters instead preserve straight edges, but flat zones present many artifacts. One could think of combining these two methods to improve both results. A Gaussian convolution could be applied in flat zones, while an anisotropic filter could be applied on straight edges. Still, other types of filters should be

region redundancy on both sides of an edge can be detected, and used for image denoising. Nevertheless, no algorithm is specified in this paper.

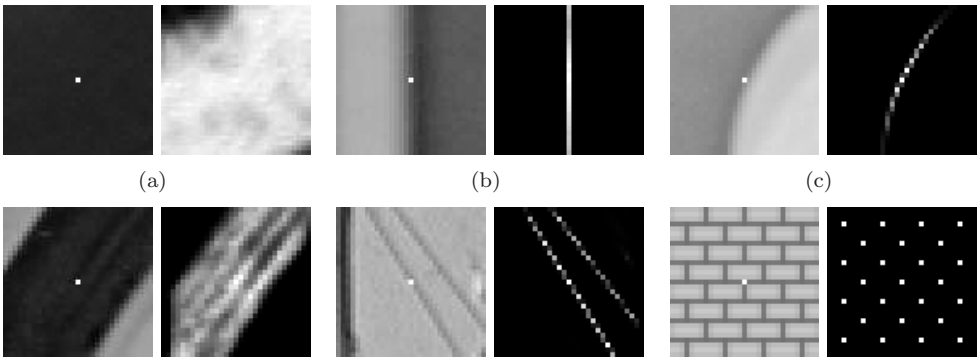


Figure 8: On the right-hand side of each pair, one can see the weight distribution used to estimate a centered patch of the left image by NL-means. (a) In flat zones, the weights are uniformly distributed, NL-means acts like a low pass isotropic filter. (b) On straight edges, the weights are distributed in the direction of the edge (like for anisotropic filters). (c) On curved edges, the weights favor pixels belonging to the same contour. (d) In a flat neighborhood, the weights are distributed in a grey-level neighborhood (exactly like for neighborhood filters). In the cases of (e) and (f), the weights are distributed across the more similar configurations, even though they are far away from the observed pixel. This behavior justifies the "non local" appellation.

designed to specifically restore corners, or curved edges, or periodic texture. Figure 8 illustrates how NL-means chooses the right weight configuration for each sort of image self-similarity.

NL-means is easily extended to the denoising of image sequences and video, involving indiscriminately pixels belonging to a space-time neighborhood. The algorithm favors pixels with a similar local configuration. When the similar configuration moves, so do the weights. Thus, as shown in [23] the algorithm is able to follow moving similar configurations without any explicit motion computation (see Fig. 9).

Indeed, this fact contrasts with previous classical movie denoising algorithms, which were motion compensated. The underlying idea of motion compensation is the existence of a "ground truth" for the physical motion. Legitimate information about the colour of a given pixel should exist only along its physical trajectory. Yet, one of the major difficulties in motion estimation is the ambiguity of trajectories, the so-called *aperture problem*. The aperture problem, viewed as a general phenomenon of movies, can be positively interpreted in the following way: There are many pixels in the next or previous frames which resemble the current pixel. Thus, it seems sound to use not just one trajectory, but rather *all similar pixels* to the current pixel across time and space as NL-means does (see [23] for more details on this discussion).

Algorithm 4 NL-means algorithm (parameter values for κ, λ are indicative).

Input: noisy image $\tilde{u}$, σ noise standard deviation.
Output: output denoised image.

Set parameter $\kappa = 3$: size of patches.
Set parameter $\lambda = 31$: size of research zone for which similar patches are searched.
Set parameter $h = 0.6\,\sigma$: bandwidth filtering parameter.

for each pixel **i** **do**

Select a square reference patch $\tilde{P}$ around **i** of dimension $\kappa \times \kappa$.

Set $\hat{P} = 0$ and $\hat{C} = 0$.

for each patch $\tilde{Q}$ in a square neighborhood of **i** of size $\lambda \times \lambda$ **do**

Compute the normalized Euclidean distance between $\tilde{P}$ and $\tilde{Q}$, $d(\tilde{P}, \tilde{Q}) = \frac{1}{\kappa^2}\|\tilde{P} - \tilde{Q}\|^2$.

Accumulate $\tilde{Q}e^{-\frac{d(\tilde{P},\tilde{Q})^2}{h^2}}$ to $\hat{P}$ and $e^{-\frac{d(\tilde{P},\tilde{Q})^2}{h^2}}$ to $\hat{C}$.

end for

Normalize the average patch $\hat{P}$ by dividing it by the sum of weights $\hat{C}$

end for

for each pixel x **do**
Aggregation: recover the denoised value at each pixel **i** by averaging all values at **i** of all denoised patches $\hat{Q}$ containing **i**
end for

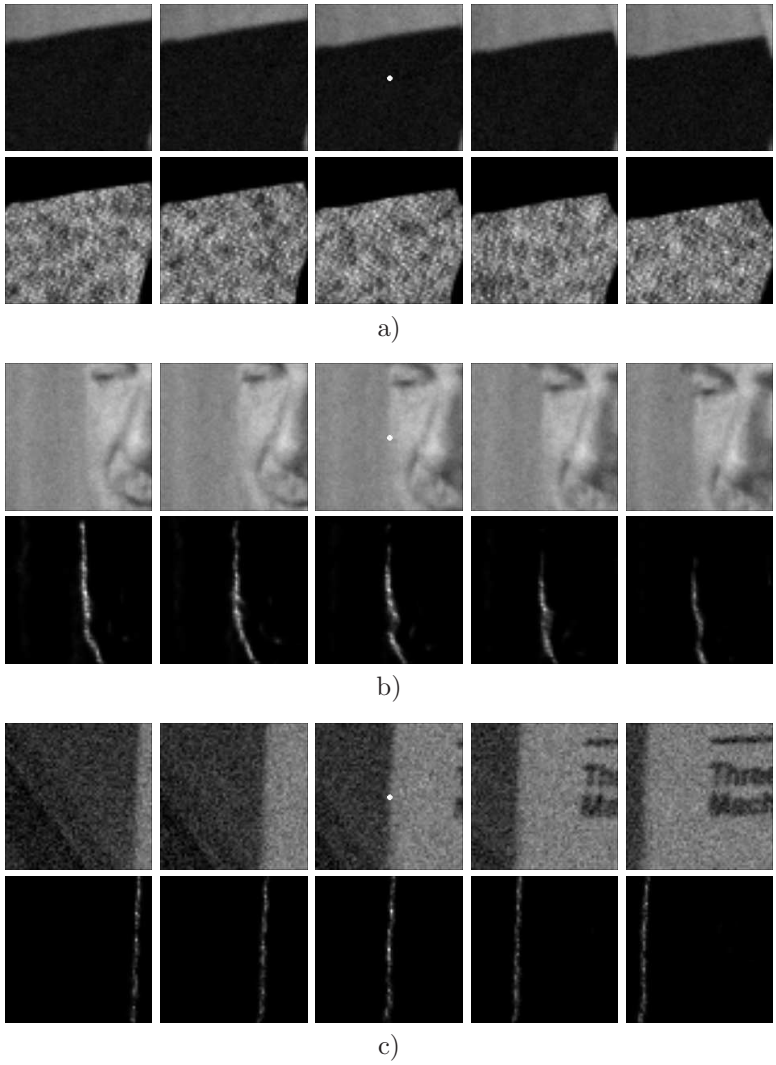


Figure 9: Weight distribution of NL-means applied to a movie. In a), b) and c) the first row shows a five frames image sequence. In the second row, the weight distribution used to estimate the central pixel (in white) of the middle frame is shown. The weights are equally distributed over the successive frames, including the current one. They actually involve all the candidates for the motion estimation instead of picking just one per frame. The aperture problem can be taken advantage of for a better denoising performance by involving more pixels in the average.

5.2 Non-local Bayesian denoising

It is apparent that (8) given in section 3.1,

$$\hat{P}_1 = \overline{\tilde{P}} + \left[\mathbf{C}_{\tilde{P}} - \sigma^2\mathbf{I}\right]\mathbf{C}_{\tilde{P}}^{-1}(\tilde{P} - \overline{\tilde{P}}),$$

gives by itself a denoising algorithm, provided we can compute the patch expectations and patch covariance matrices. We shall now explain how the Non-local Bayes algorithm proposed in [86] does it. Let $\mathcal{P}(\tilde{P})$ be the set of patches $\tilde{Q}$ similar to the patch $\tilde{P}$, which have obtained with a suitably chosen tolerance threshold, so that we can assume that they represent noisy versions of the patches similar to P. Then, by the law of large numbers, we have

$$\mathbf{C}_{\tilde{P}} \simeq \frac{1}{\#\mathcal{P}(\tilde{P}) - 1}\sum_{\tilde{Q}\in\mathcal{P}(\tilde{P})}\left(\tilde{Q} - \overline{\tilde{P}}\right)\left(\tilde{Q} - \overline{\tilde{P}}\right)^t, \qquad \overline{\tilde{P}} \simeq \frac{1}{\#\mathcal{P}(\tilde{P})}\sum_{\tilde{Q}\in\mathcal{P}(\tilde{P})}\tilde{Q}. \tag{17}$$

Nevertheless, the selection of similar patches at the first step is not optimal and can be improved in a second estimation step where the first step estimation is used as oracle. Thus, in a second step, where all patches have been denoised at the first step, all the denoised patches can be used again to obtain an estimation $\mathbf{C}_{\hat{P}_1}$ for $\mathbf{C}_P$, the covariance of the cluster containing P, and a new estimation of $\overline{\tilde{P}}$, the average of patches similar to $\tilde{P}$. Indeed, the patch similarity is better estimated with the denoised patches. Then it follows from (7) and (8) that we can obtain a second better denoised patch,

$$\hat{P}_2 = \overline{\tilde{P}}^1 + \mathbf{C}_{\hat{P}_1}\left[\mathbf{C}_{\hat{P}_1} + \sigma^2\mathbf{I}\right]^{-1}(\tilde{P} - \overline{\tilde{P}}^1) \tag{18}$$

where

$$\mathbf{C}_{\hat{P}_1} \simeq \frac{1}{\#\mathcal{P}(\hat{P}_1) - 1}\sum_{\hat{Q}_1\in\mathcal{P}(\hat{P}_1)}\left(\hat{Q}_1 - \overline{\tilde{P}}^1\right)\left(\hat{Q}_1 - \overline{\tilde{P}}^1\right)^t, \qquad \overline{\tilde{P}}^1 \simeq \frac{1}{\#\mathcal{P}(\hat{P}_1)}\sum_{\hat{Q}_1\in\mathcal{P}(\hat{P}_1)}\tilde{Q}. \tag{19}$$

We write the denoised patches $\overline{\tilde{P}}$ in (17) and $\overline{\tilde{P}}^1$ in (18). Indeed, in (18), the denoised version of $\tilde{P}$ computed as the average of noisy patches $\tilde{Q}$ whose denoised patch is similar to $\hat{P}_1$.

In short, the estimates (8) and (18) appear equivalent, but they are not in practice. $\mathbf{C}_{\hat{P}_1}$, obtained after a first denoising step, is a better estimation than $\mathbf{C}_{\tilde{P}}$. Furthermore, $\overline{\tilde{P}}^1$ is a more accurate mean than $\overline{\tilde{P}}$. It uses a better evaluation of patch similarities. All above quantities being computable from the noisy image, we obtain the two step algorithm 5.

As pointed out in [27], the Nonlocal Bayes algorithm only is an interpretation (with some generic improvements like the aggregation) of the PCA based algorithm proposed in [157]. This paper has a self-explanatory title: "Two-stage image denoising by principal component analysis with local pixel grouping." It is equivalent to apply a PCA on the patches similar to $\tilde{P}$, followed by a Wiener filter on the coefficients of $\tilde{P}$ on this PCA, or to apply formula (8) with the covariance matrix of the similar patches. Indeed the PCA computes nothing but the eigenvalues of the empirical covariance matrix. Thus, the method in [157] gets its Bayesian interpretation. A study on the compared performance of local PCA versus global PCA for TSID is actually proposed in [31].

5.3 Patch-based near-optimal image denoising (PLOW)

While in the Non-local Bayes method of section 5.2 a local model is estimated in a neighborhood of each patch, in the PLOW [33] method the idea is to learn from the image a sufficient number of patch clusters, actually 15, and to apply the LMMSE estimate to each patch *after* having assigned it to one of the clusters obtained by clustering. Thus, this empirical-Bayesian algorithm starts by clustering the patches by the classic K-means clustering algorithm. To take into account that similar patches can actually have varying contrast, the inter-patch distance is photometrically

Algorithm 5 Non local Bayes image denoising

Input: noisy image
Output: denoised image

for all patches $\tilde{P}$ of the noisy image **do**

Find a set $\mathcal{P}(\tilde{P})$ of patches $\tilde{Q}$ similar to $\tilde{P}$.

Compute the expectation $\overline{\tilde{P}}$ and covariance matrix $\mathbf{C}_{\tilde{P}}$ of these patches by

$$\mathbf{C}_{\tilde{P}} \simeq \frac{1}{\#\mathcal{P}(\tilde{P}) - 1} \sum_{\tilde{Q} \in \mathcal{P}(\tilde{P})} \Big(\tilde{Q} - \overline{\tilde{P}}\Big)\Big(\tilde{Q} - \overline{\tilde{P}}\Big)^t, \qquad \overline{\tilde{P}} \simeq \frac{1}{\#\mathcal{P}(\tilde{P})} \sum_{\tilde{Q} \in \mathcal{P}(\tilde{P})} \tilde{Q}.$$

Obtain the first step estimation:

$$\hat{P}_1 = \overline{\tilde{P}} + \left[\mathbf{C}_{\tilde{P}} - \sigma^2 \mathbf{I}\right] \mathbf{C}_{\tilde{P}}^{-1} (\tilde{P} - \overline{\tilde{P}}).$$

end for
Obtain the pixel value of the basic estimate image $\hat{u}_1$ as an average of all values of all denoised patches $\hat{Q}_1$ which contain $\mathbf{i}$.
for all patches $\tilde{P}$ of the noisy image **do**

Find a new set $\mathcal{P}_1(\tilde{P})$ of noisy patches $\tilde{Q}$ similar to $\tilde{P}$ by comparing their denoised "oracular" versions Q_1 to P_1.

Compute the new expectation $\overline{\tilde{P}}^1$ and covariance matrix $\mathbf{C}_{\hat{P}_1}$ of these patches:

$$\mathbf{C}_{\hat{P}_1} \simeq \frac{1}{\#\mathcal{P}(\hat{P}_1) - 1} \sum_{\hat{Q}_1 \in \mathcal{P}(\hat{P}_1)} \left(\hat{Q}_1 - \overline{\tilde{P}}^1\right)\left(\hat{Q}_1 - \overline{\tilde{P}}^1\right)^t, \qquad \overline{\tilde{P}}^1 \simeq \frac{1}{\#\mathcal{P}(\hat{P}_1)} \sum_{\hat{Q}_1 \in \mathcal{P}(\hat{P}_1)} \tilde{Q}.$$

Obtain the second step patch estimate

$$\hat{P}_2 = \overline{\tilde{P}}^1 + \mathbf{C}_{\hat{P}_1} \left[\mathbf{C}_{\hat{P}_1} + \sigma^2 \mathbf{I}\right]^{-1} (\tilde{P} - \overline{\tilde{P}}^1).$$

end for
Obtain the pixel value of the denoised image $\hat{u}(\mathbf{i})$ as an average of all values of all denoised patches $\hat{Q}_2$ which contain $\mathbf{i}$.

neutral, and the authors call it a "geometric distance". The clustering phase is accelerated by a dimension reduction obtained by applying a principal component analysis to the patches. The clustering is therefore a segmentation of the set of patches, and the denoising of each patch is then performed within its cluster. Since each cluster contains geometrically similar, but not necessarily photometrically similar patches, the method identifies for each patch in the cluster the photometrically similar patches as those whose quadratic distance to the reference patch are within the bounds allowed by noise. Then a LMMSE [81] estimate is obtained for the reference patch by a variant of (8). The algorithm uses a first phase, which performs a first denoising before constituting the clusters. Thus the main phase is actually using the first phase as oracle to get the covariance matrices of the sets of patches.

Algorithm 6 Algorithm 1: PLOW denoising

Input: image in vector form $\tilde{U}$.
Output: denoised image in vector form $\hat{U}$.

Set parameters: patch size $\kappa \times \kappa = 11 \times 11$, number of clusters $K = 15$;
Estimate noise standard deviation $\hat{\sigma}$ by $\hat{\sigma} = 1.4826\text{median}(|\nabla\tilde{U} - \text{median}(\nabla\tilde{U})|)$;
Set parameter: $h^2 = 1.75\hat{\sigma}^2\kappa^2$;
Pre-filter image $\tilde{U}$ to obtain a pilot estimate $\hat{U}_1$;
Extract overlapping patches of size $\kappa \times \kappa$, $\tilde{Q}$ from $\tilde{U}$ and $\hat{Q}_1$ from U_1;
Geometric clustering with K-Means of the patches in $\hat{U}_1$ (using a variant of PCA for the patches). The distance is a geometric distance, photometrically neutral.
for each patch cluster Ω_k **do**
  Estimate from the patches $\hat{Q}_1 \in \Omega_k$ the mean patch $\overline{P}_k \simeq \sum_{\hat{Q}_1 \in \Omega_k} \hat{Q}_1$ and the cluster covariance $\mathbf{C}_P^k$.
  for each patch $\hat{Q}_{1,i} \in \Omega_k$ **do**
    Consider its associated noisy patch $\tilde{Q}_i$. Identify photometrically similar patches $\tilde{Q}_j$ in the cluster as those with a quadratic distance to $\tilde{Q}_i$ within the bounds allowed by noise, namely $\gamma^2 + 2\kappa^2\hat{\sigma}^2$, with $\gamma = \gamma(\kappa)$ a "small" threshold.
    Compute similarity weights $w_{ij} = e^{-\frac{||\tilde{Q}_i - \tilde{Q}_j||^2}{h^2}}$.
    Compute the slightly more complex than usual LMMSE estimator for the noisy patch $\tilde{Q}_i$, (because the cluster contains patches that are geometrically similar but not necessarily photometrically similar):

$$\hat{Q}_i = \overline{P} + \left[\mathbf{I} - \left(\sum_j w_{ij}\mathbf{C}_P^k + \mathbf{I}\right)^{-1}\right] \sum_j \frac{w_{ij}}{\sum_j w_{ij}} (\tilde{Q}_j - \overline{P}).$$

  end for
end for
At each pixel aggregate multiple estimates from all $\hat{P}$ containing it, with weights given as inverses of the variance of each estimator.

5.4 Inherent bounds in image denoising

By "Shotgun" patch denoising methods, we mean methods that intend to denoise patches by a fully non-local algorithm, in which the patch is compared to a patch model obtained from a large or *very large* patch set. The "sparse-land" methods intend to learn from a single image or from a small set of images a sparse patch dictionary on which to decompose any given patch. The shotgun methods learn instead from a very large patch set extracted from tens of thousands of images (up to 10^{10} patches). Then the patch is denoised by deducing its likeliest estimate from the set of all patches. In the case of [161], this patch space is organized as a Gaussian mixture with about 200 components. Shotgun methods have started being used in several image restoration methods. For example in [74], for image inpainting, with an explicit enough title: "Scene completion using millions of photographs".

The approach of [92] is to define the simplest universal "shotgun" method, where a huge set of patches is used to estimate the upper limits a patch-based denoising method will ever reach. The results support the "near optimality of state of the art denoising results", the results obtained by the BM3D algorithm being only 0.1 decibel away from optimality for methods using small patches (typically 8×8.)

This experiment uses to evaluate the MMSE a set of 20, 000 images from the LabelMe dataset [135]. The method, even if certainly not practical, is of exquisite simplicity. Given a clean patch P the noisy patch $\tilde{P}$ with Gaussian noise of standard deviation σ has probability distribution

$$\mathbb{P}(\tilde{P} \mid P) = \frac{1}{(2\pi\sigma^2)^{\frac{\kappa^2}{2}}} e^{-\frac{||P-\tilde{P}||^2}{2\sigma^2}}, \tag{20}$$

where κ^2 is the number of pixels in the patch. Then given a noisy patch $\tilde{P}$ its optimal estimator for the Bayesian minimum squared error (MMSE) is by Bayes' formula

$$\hat{P} = \mathbb{E}[P \mid \tilde{P}] = \int \mathbb{P}(P \mid \tilde{P}) P dP = \int \frac{\mathbb{P}(\tilde{P} \mid P)}{\mathbb{P}(\tilde{P})} \mathbb{P}(P) P dP. \tag{21}$$

Using a huge set of M natural patches (with a distribution supposedly approximating the real natural patch density), we can approximate the terms in (21) by $\mathbb{P}(P)dP \simeq \frac{1}{M}$ and $\mathbb{P}(\tilde{P}) \simeq \frac{1}{M}\sum_i \mathbb{P}(\tilde{P} \mid P_i)$, which in view of (20) yields

$$\hat{P} \simeq \frac{\frac{1}{M}\sum_i \mathbb{P}(\tilde{P} \mid P_i) P_i}{\frac{1}{M}\sum_i \mathbb{P}(\tilde{P} \mid P_i)}.$$

Thus the final MMSE estimator is nothing but the exact application of NL-means, denoising each patch by matching it to the huge patch database. Clearly this is not just a theoretical algorithm. Web based application could provide a way to denoise online any image by organizing a huge patch data base. The final algorithm is summarized in Algorithm 7.

The main focus of [92] is, however, as we mentioned, elsewhere: it uses this shotgun denoising to estimate universal upper and lower bounds of the attainable PSNR by any patch based denoising algorithm. More precisely, the algorithm gives upper and lower bounds to the following problem:

Given a noisy patch $\tilde{P}$, given the law $p(P)$ of all patches in the world, find the best possible estimate (in the sense of MMSE). The shotgun algorithm gives a best possible estimate for *any patch based denoising algorithm* of this kind.

The upper bound obtained by the authors turns out to be very close to results obtained with BM3D (see sec. 5.8), and the authors conclude that for small window sizes, or moderate to high noise levels, the chase for the best denoising algorithm might be close to the finish. More precisely, only fractions of decibels separate the current best algorithms from these demonstrated upper bounds. The EPLL method [161] can be viewed as a first (slightly) more practical realization of this quasi-optimality by a shotgun algorithm, and there is no doubt that other more practical ones will follow. We now describe how the [92] lower and upper bounds can be estimated from a sufficient set of natural images.

Algorithm 7 Shotgun NL-means

Input: Noisy image $\tilde{u}$ in vectorial form.
Input: Very large set of M patches P_i extracted from a large set of noiseless natural images.
Output: Denoised image $\hat{u}$.

for all patches $\tilde{P}$ extracted from $\tilde{u}$ **do**
Compute the MMSE denoised estimate of $\tilde{P}$

$$\hat{P} \simeq \frac{\sum_{i=1}^{M} \mathbb{P}(\tilde{P} \mid P_i) P_i}{\sum_{i=1}^{M} \mathbb{P}(\tilde{P} \mid P_i)}$$

where $\mathbb{P}(\tilde{P} \mid P_i)$ is known from (20).
end for
At each pixel $\mathbf{i}$ get $\hat{u}(\mathbf{i})$ as $\hat{P}(\mathbf{i})$, where the patch P is centered at $\mathbf{i}$.
(optional Aggregation) : for each pixel $\mathbf{j}$ of u, compute the denoised version $\hat{u}_{\mathbf{j}}$ as the average of all values $\hat{P}(\mathbf{j})$ for all patches containing $\mathbf{j}$. (This step in not considered in [92].)

The MSE for a given denoising algorithm can be obtained by randomly sampling patches P, then add noise to generate noisy patches $\tilde{P}$, and measure the reconstruction error $\|P - \hat{P}\|^2$. Then the mean reconstruction error is

$$\text{MSE} = \int \mathbb{P}(P) \int \mathbb{P}(\tilde{P} \mid P) \|P - \hat{P}\|^2 d\tilde{P} dP. \tag{22}$$

Conversely, one can start from a noisy patch $\tilde{P}$, measure the variance of $\mathbb{P}(P \mid \tilde{P})$ around it. This amounts according to the authors of [92] to compute the sum of weighted distances between the restored $\hat{P}$ and all possible P explanations:

$$\text{MSE} = \int \mathbb{P}(\tilde{P}) \int \mathbb{P}(P \mid \tilde{P}) \|P - \hat{P}\|^2 d\tilde{P} dP. \tag{23}$$

This last equation follows from (22) by the Bayes rule. For each noisy $\tilde{P}$ one can define its MMSE

$$\text{MMSE}(\tilde{P}) = E[\|\hat{P} - \tilde{P}\|^2 \mid \tilde{P}] = \int \mathbb{P}(P \mid \tilde{P})(P - \hat{P})^2 dP. \tag{24}$$

The main interest of this formulation is that it permits to prove that the MMSE is, of all denoising algorithms, the one that minimizes the overall MSE. Indeed, differentiating (23) with respect to $\hat{P}$ yields back the MMSE estimator (21). The best overall MMSE achievable by any given denoising algorithm therefore is

$$\text{MMSE} = \int \mathbb{P}(\tilde{P}) E[\|\hat{P} - \tilde{P}\|^2 \mid \tilde{P}] = \int \mathbb{P}(\tilde{P}) \mathbb{P}(P \mid \tilde{P})(P - \hat{P})^2 dP d\tilde{P}. \tag{25}$$

The goal of [92] is to bound the MMSE from below, ignoring of course the probability distribution $\mathbb{P}(P)$, but having enough samples of it. The main idea is to derive an upper and a lower bound on the MMSE from the two MSE formulations (22)-(23). Given a set of M clean and noisy pairs $\{(P_j, \tilde{P}_j)\}$, $j = 1 \ldots, M$ and another independent set of N clean patches $\{P_i\}$, $i = 1 \ldots, N$, both randomly sampled from natural images the proposed estimates are

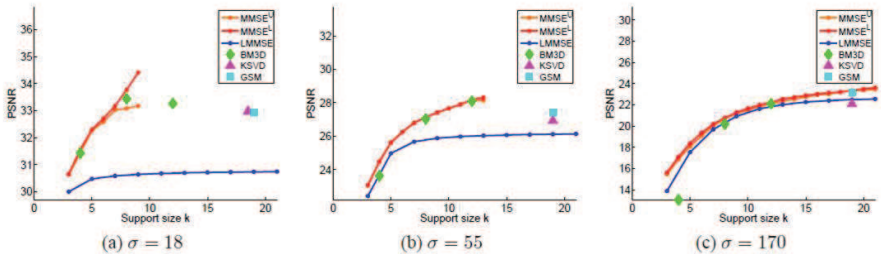
$$\text{MMSE}^U = \frac{1}{M} \sum_j \|\hat{P}_j - P_j\|^2 \tag{26}$$

and

$$\text{MMSE}^L = \frac{1}{M} \sum_j \frac{\sum_i \mathbb{P}(\tilde{P}_j \mid P_i) \|\hat{P}_j - P_i\|^2}{\sum_i \mathbb{P}(\tilde{P}_j \mid P_i)}. \tag{27}$$

A striking feature of both estimates is that $MMSE^U$ uses the explicit knowledge of the original noise-free patch P_j, while MMSE^L does not involve it. Since MMSE^U simply measures the error for a given denoising algorithm, it obviously provides an upper bound for the MMSE of any other denoising algorithm. As the authors observe, MMSE^U and MMSE^L are random variables that depend on the choice of the samples. When the sample size approaches infinity, both converge to the exact MMSE. Nevertheless, [92] gives a simple proof that, for a finite sample, in expectation, MMSE^U and MMSE^L provide upper and lower bounds on the best possible MMSE. When both MMSE^U and MMSE^L coincide, they provide an accurate estimate of the optimal denoising possible with a given patch size.

For very high noise levels, the authors of [92] also tried to apply the linear minimum mean square error (LMMSE) estimator (or Wiener filter) using only the second order statistics of the data, by fitting a single k^2 dimensional Gaussian to the set of M image $k \times k$ patches. They conclude that even this simple approach is close to optimal for large noise.



(a) $\sigma = 18$ (b) $\sigma = 55$ (c) $\sigma = 170$

Figure 10: Comparing [92] the PSNR ($= -10 \log_{10}(\text{MMSE})$) of several denoising algorithms (K-SVD [100], BM3D [39], Gaussian Scale Mixture [128]) compared to the PNSR predicted by MMSE^U, MMSE^U. The performance of all algorithms is bounded by the MMSE^U estimate, but BM3D approaches this upper bound by fractional dB values. Nevertheless, the performance bounds consider more restrictive patch based algorithms than the class BM3D belongs to. Thus the actual gap to optimality may be higher.

5.5 The expected patch log likelihood (EPLL) method

The patch Gaussian mixture model This other shotgun method [161] is an almost literal application of the piecewise linear estimator (PLE) method [156], see section 5.9). But it is shotgun, namely applied to a huge set of patches instead of the image itself. A Gaussian mixture model is learnt from a set of 2.10^6 patches, sampled from the Berkeley database with their mean removed. The 200 mixture components with zero means and full covariance matrices are obtained using the EM (expectation maximization) algorithm. This training took about 30 hours with a public MATLAB code[3]. Thus were learnt: 200 means (actually they are all zero), 200 full covariance matrices and 200 mixing weights which constitute the Gaussian mixture model of this set of patches. Fig 11 shows some six bases extracted from the Gaussian mixture. Each one shows the patches that are eigenvectors of some of the covariance matrices, sorted by eigenvalue.

Once the Gaussian mixture is learnt, the denoising method maximizes the Expected Patch Log Likelihood (EPLL) while being close to the corrupted image in a way which is dependent on the (linear) corruption model. Given an image U (in vector form) the EPLL of U under prior $\mathbb{P}$ is defined by

$$EPLL_{\mathbb{P}}(U) = \sum_i \log \mathbb{P}(\mathbf{P}_i U)$$

where $\mathbf{P}_i$ is a matrix which extracts the i-th patch P_i from the image U out of all overlapping patches, while $\log \mathbb{P}(\mathbf{P}_i X)$ is the likelihood of the i-th patch under the prior $\mathbb{P}$. Assuming a patch

[3] http://www.mathworks.com/matlabcentral/fileexchange/26184-em-algorithm-for-gaussian-mixture-model.

Figure 11: Eigenvectors of six randomly selected covariance matrices from the learnt Gaussian mixture model, sorted by eigenvalue from largest to smallest, from [161]. The authors notice the similarity of these basis elements to DCT, but also that many seem to model texture boundaries and edges at various orientations.

location in the image is chosen uniformly at random, EPLL can be interpreted as the expected log likelihood of a patch in the image (up to a multiplication by $1/M$). Given a corrupted image $\tilde{U}$ in vector form and a model of image corruption of the form $||\mathbf{A}U - \tilde{U}||^2$, the restoration is made by minimizing

$$f_{\mathbb{P}}(U|\tilde{U}) = \frac{\lambda}{2}||\mathbf{A}U - \tilde{U}||^2 - EPLL_{\mathbb{P}}(U).$$

According to the authors, "This equation has the familiar form of a likelihood term and a prior term, but note that $EPLL_{\mathbb{P}}(U)$ is not the log probability of a full image. Since it sums over the *log* probabilities of all overlapping patches, it "double counts" the log probability. Rather, it is the expected log likelihood of a randomly chosen patch in the image."

The optimization is made by "half quadratic splitting" which amounts to introduce auxiliary patch variables Z^i, $i = 1, \dots M$, one for each patch P_i, and to minimize alternatively the auxiliary functional

$$C_{\mathbb{P},\beta}(U, \{Z^i\}|\tilde{U}) := \frac{\lambda}{2}||\mathbf{A}U - \tilde{U}||^2 + \frac{\beta}{2}\sum_i ||\mathbf{P}_i U - Z^i||^2 - \log \mathbb{P}(Z^i).$$

Solving for U given $\{Z^i\}$ amounts to the inversion

$$U = \left(\lambda \mathbf{A}^T\mathbf{A} + \beta \sum_i \mathbf{P}_i^T \mathbf{P}_i\right)^{-1} \left(\lambda \mathbf{A}^T \tilde{U} + \beta \sum_i \mathbf{P}_i^T Z^i\right).$$

In the case of denoising, $\mathbf{A}$ is simply the identity, and the above formula boils down to computing for each pixel $\mathbf{j}$ a denoised value $U(\mathbf{j})$ as a weighted average over all patches P_i containing this given pixel $\mathbf{j}$ of the noisy pixel value $\tilde{U}(\mathbf{j})$ and of the patch denoised values $Z_i(\mathbf{j})$:

$$U(\mathbf{j}) = \frac{\lambda \tilde{U}(\mathbf{j}) + \sum_{P_i \ni \mathbf{j}} Z_i(\mathbf{j})}{\lambda + \beta k^2}, \tag{28}$$

where k^2 is the patch size.

Then, solving for $\{Z_i\}$ given U amounts to solving a MAP (maximum a posteriori) problem of estimating the most likely patch under the prior $\mathbb{P}$, given $\mathbf{P}_i U$ and parameter β.

The Gaussian mixture model being known, calculating the log likelihood of a given patch is trivial:

$$\log \mathbb{P}(Q) = \log\left(\sum_{k=1}^{K} \pi_k \mathcal{N}(Q|\mu_k, \mathbf{C}_k)\right),$$

where π_k are the mixing weights for each of the mixture component, μ_k and $\mathbf{C}_k$ are the corresponding mean and covariance matrix.

Given a noisy patch $\tilde{Q}$, the MAP estimate is computed with the following procedure:

Algorithm 8 Patch restoration once the Patch Gaussian mixture is known

for each noisy patch $\tilde{Q}$ **do**

Compute the conditional mixture weights $\pi'_k = \mathbb{P}(k \mid \tilde{Q})$ (given by EM);

Pick the component k with the highest conditional mixing weight: $k_{max} = \max_k \pi'_k$;

The MAP estimate $\hat{Q}$ is a Wiener solution for the k_{max}-th component:

$$\hat{Q} = \left(\mathbf{C}_{k_{max}} + \sigma^2 \mathbf{I}\right)^{-1} \left(\mathbf{C}_{k_{max}} \tilde{Q} + \sigma^2 \mu_{k_{max}}\right).$$

end for

The authors comment that this is one iteration of the "hard version" of the EM algorithm for finding the modes of a Gaussian mixture [30]. The method can be used for denoising and several experiments seems to indicate that it equals the performance of BM3D and LLSC [100].

5.6 The Portilla et al. wavelet neighborhood denoising (BLS-GSM)

The basic idea of this algorithm is modeling a noiseless "wavelet coefficient neighborhood", P, by a Gaussian scale mixture (GSM) which is defined as

$$P = \sqrt{z}U$$

where U is a zero-mean Gaussian random vector and z is an independent positive scalar random variable. The wavelet coefficient neighborhood turns out to be a patch of an oriented channel of the image at a given scale, complemented with a coefficient of the channel at the same orientation and the next lower scale. Thus, we adopt again the patch notation P. (Arguably, this method is the first patch-based method.) Using a GSM model for P estimated from the image itself, the method makes a Bayes least square (BLS) estimator. For this reason, the method will be called here BLS-GSM (Bayes least square estimate of Gaussian scale mixture; the authors called it simply BLS.) Without loss of generality it is assumed that $Ez = 1$ and therefore the random variables U and P have similar covariances. To use the GSM model for wavelet patch denoising, the noisy input image is first decomposed into a wavelet pyramid, and each image of the pyramid will be separately denoised. The resulting denoised image is obtained by the reconstruction algorithm from the wavelet coefficients. To avoid ringing artifacts in the reconstruction, a redundant version of the wavelet transform, the so-called steerable pyramid, is used. For a $n_1 \times n_2$ image, the pyramid, $\mathcal{P}$, is generated on $\log_2(\min(n_1, n_2) - 4)$ scales and eight orientations using the following procedure. First the input image is decomposed into one low-pass and eight oriented high-pass component images using two polar filters in quadrature in the Fourier domain (the sum of their squares is equal to 1). The Fourier domain being represented in polar coordinates (r, θ), the low pass and high pass isotropic filters are

$$l(r) = \begin{cases} 1 & 0 \le r < 0.5; \\ \cos(\frac{\pi}{2}(-\log_2 r - 1)) & 0.5 \le r < 1; \\ 0 & 1 \le r \le \sqrt{2}; \end{cases} \tag{29}$$

and

$$h(r) = \begin{cases} 0 & 0 \le r < 0.5; \\ \cos(\frac{\pi}{2}(\log_2 r)) & 0.5 \le r < 1; \\ 1 & 1 \le r \le \sqrt{2}. \end{cases} \tag{30}$$

The high pass filter h is decomposed again into eight oriented components,

$$a_k(r, \theta) = h(r)g_k(\theta), \quad k \in [0, K-1], \tag{31}$$

where $K = 8$, and

$$g_k(\theta) = \frac{(K-1)}{\sqrt{K[2(K-1)]}} \left[2\cos\left(\theta - \frac{\pi k}{K}\right)\right]^{K-1}. \tag{32}$$

Then the steerable pyramid is generated by iteratively applying the a_k filters to the result of the low-pass filter to obtain bandpass images, and calculating the residual using the l filter followed by sub-sampling. For example in the case of a 512×512 image we have a 5 scales pyramid consisting of 49 sub-bands: 8 high-pass oriented sub-bands, from $\mathcal{P}^1$ to $\mathcal{P}^8$, 8 bandpass oriented sub-bands for each scale, from $\mathcal{P}^9$ to $\mathcal{P}^{48}$, in addition to one lowpass non-oriented residual subband, $\mathcal{P}^{49}$. (WLOG we shall keep this 49 number as landmark, but this number depends of course on the image size). Assume now that the image has been corrupted by independent additive Gaussian noise. Therefore, a typical neighborhood of wavelet coefficients can be represented as

$$\tilde{P} = P + N = \sqrt{z}U + N, \tag{33}$$

where noise, N, and P are considered to be independent. Define $p_s(i,j)$ as the sample at position (i,j) of the sub-band $\mathcal{P}^s$, the subbands being enumerated as (e.g.) $s = 1, \ldots, 49$. The neighborhood of the wavelet coefficient $p_s(i,j)$ is composed of its spatial neighbors for the same sub-band s. It could have contained also coefficients from other sub-bands at the same scale as $p_s(i,j)$ but with different orientations, and could finally also contain sub-band coefficients from the adjacent scales, up and down. Surprisingly, the final neighborhood is quite limited: The authors sustain that the best efficiency is reached with a 3×3 spatial block around $p_s(i,j)$, supplemented with one coefficient at the same location and at the next coarser scale (considering its up-sampled parent by interpolation) with the same orientation. Hence, the neighborhood size is 10 and contains only $\{p_s(i-1,j-1), \ldots, p_s(i+1,j+1), p_{s+8}(i,j)\}$. There are two exceptions for this: first is the neighborhood of coarsest scale coefficients (without any coarser scale) has necessarily only 9 surrounding coefficients. Second, the boundary coefficients are processed using special steps described below. Using the observed noisy vector, $\tilde{P}$, an estimation of P, can be obtained by

$$E(P \mid \tilde{P}) = \int_0^\infty \mathbb{P}(z \mid \tilde{P}) E(P \mid \tilde{P}, z) dz.$$

This estimation is the Bayesian denoised value of the reference coefficient. The integral is computed numerically on experimentally obtained sampled intervals of z. Here, only 13 equally spaced values of z in the interval $[\ln(z_{min}), \ln(z_{max})] = [-20.5, 3.5]$ are used. Therefore $E(P \mid \tilde{P})$ is computed as

$$E(P \mid \tilde{P}) = \sum_{i=1}^{13} \mathbb{P}(z_i \mid \tilde{P}) E(P \mid \tilde{P}, z_i). \tag{34}$$

The only question left is: how to compute the conditional probability and the conditional expectation, $\mathbb{P}(z_i \mid \tilde{P})$ and $E(P \mid \tilde{P}, z_i)$. For each sub-band, $\mathcal{P}^s$ except the low-pass residual, $\mathcal{P}^{49}$ which remains unchanged, define $\mathbf{C}_N^s$ and $\mathbf{C}_{\tilde{P}}^s$, respectively the noise and the observation covariance matrices of the wavelet neighborhood. If n^s denotes the size of neighborhood at sub-band $\mathcal{P}^s$ (n^s therefore is 10 or 9 as explained above), $\mathbf{C}_N^s$ is a $n^s \times n^s$ matrix which can be experimentally generated by first decomposing a delta function $\sigma\delta$ on the steerable pyramid. Here σ is the known noise variance and δ is an $n_1 \times n_2$ image defined by

$$\delta(i,j) = \begin{cases} 1 & (i,j) = (\frac{n_1}{2}, \frac{n_2}{2}), \\ 0 & \text{otherwise.} \end{cases}$$

(This covariance matrix is equal to the covariance of the white noise defined as a band-limited function obtained by randomizing uniformly the phase of the Fourier coefficients of the discrete Dirac mass δ.) Using the steerable pyramid decomposition of $\sigma\delta$, define $\mathbf{N}_s$ as the matrix which has for rows all neighborhoods of the sub-band $\mathcal{P}_s$. This is a matrix with n_s columns and $(n_1-2)(n_2-2)$ rows. (Subtracting 2 is for eliminating the boundary coefficients). The covariance $\mathbf{C}_N^s$ matrix of the neighborhood samples for each sub-band is computed as

$$\mathbf{C}_N^s = \frac{\mathbf{N}_s^T \mathbf{N}_s}{(n_1-2)(n_2-2)},$$

where $(.)^T$ stands for matrix transposition. Since all the noise removal steps are calculated for each sub-band separately, in the following we skip the superscript s to simplify the notation. Similarly but using the pyramid of observed noisy samples, $\mathbf{C}_{\tilde{P}}$ can be computed. Using (33) and the assumption $Ez = 1$, for each sub-band s we have

$$\mathbf{C}_U = \mathbf{C}_{\tilde{P}} - \mathbf{C}_N.$$

$\mathbf{C}_U$ can be forced to be positive semi-definite by setting to zero all of its negative eigenvalues. We can now calculate $E(P \mid \tilde{P}, z_i)$. Using the fact that P and $\underline{\mathrm{N}}$ are Gaussian independent variables and also that the noise is additive, $E(P \mid \tilde{P}, z_i)$ is simply a local Wiener estimate:

$$E(P \mid \tilde{P}, z) = \frac{z\mathbf{C}_U}{z\mathbf{C}_U + \mathbf{C}_N} \tilde{P},$$

where the matrix fraction notation is understood as $\frac{\mathbf{C}}{\mathbf{W}} := \mathbf{C}\mathbf{W}^{-1}$. Clearly it would be cumbersome to compute as many matrix inversions as z_i's. Fortunately, with a bit of linear algebra this computation can be rendered common to all z_i. Define $\{\mathbf{Q}, \mathbf{\Lambda}\}$ as the eigenvectors and eigenvalues of $\mathbf{S}^{-1}\mathbf{C}_U\mathbf{S}^{-T}$, where $\mathbf{S}_{n^s \times n^s}$ is the symmetric square root of $\mathbf{C}_N$, $\mathbf{C}_N = \mathbf{S}\mathbf{S}^T$. So we have $\mathbf{S}^{-1}\mathbf{C}_U\mathbf{S}^{-T} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$. Furthermore, set $\mathbf{M} = \mathbf{S}\mathbf{Q}, \quad \mathbf{v} = \mathbf{M}^{-1}\tilde{P}$. Then we have

$$\begin{aligned}
E(P \mid \tilde{P}, z) &= \frac{z\mathbf{C}_U}{z\mathbf{C}_U + \mathbf{C}_N}\tilde{P} \\
&= \frac{z\mathbf{C}_U}{z\mathbf{C}_U + \mathbf{S}\mathbf{S}^T}\tilde{P} \\
&= \frac{z\mathbf{C}_U}{\mathbf{S}(z\mathbf{S}^{-1}\mathbf{C}_U\mathbf{S}^{-T} + \mathbf{I})\mathbf{S}^T}\tilde{P} \\
&= \frac{z\mathbf{C}_U}{\mathbf{S}\mathbf{Q}(z\mathbf{\Lambda} + \mathbf{I})\mathbf{Q}^T\mathbf{S}^T}\tilde{P} \\
&= z\mathbf{C}_U\mathbf{S}^{-T}\mathbf{Q}(z\mathbf{\Lambda} + \mathbf{I})^{-1}\mathbf{Q}^T\mathbf{S}^{-1}\tilde{P} \\
&= z\mathbf{S}\mathbf{S}^{-1}\mathbf{C}_U\mathbf{S}^{-T}\mathbf{Q}(z\mathbf{\Lambda} + \mathbf{I})^{-1}\mathbf{Q}^T\mathbf{S}^{-1}\tilde{P} \\
&= z\mathbf{S}\mathbf{Q}\mathbf{\Lambda}(z\mathbf{\Lambda} + \mathbf{I})^{-1}\mathbf{Q}^T\mathbf{S}^{-1}\tilde{P} \\
&= z\mathbf{M}\mathbf{\Lambda}(z\mathbf{\Lambda} + \mathbf{I})^{-1}\mathbf{v}.
\end{aligned}$$

The interest is that one can calculate $\mathbf{M}$, $\mathbf{\Lambda}$ and $\mathbf{v}$ once for each subband. The scalar final formulation of the above equation is

$$E(P \mid \tilde{P}, z_i)_c = \sum_{j=1}^{n^s} \frac{z_i m_{c,j}\lambda_{j,j}v_j}{z_i\lambda_{j,j} + 1}, \tag{35}$$

where $m_{c,j}$, $\lambda_{j,j}$ and v_j are the elements of $\mathbf{M}$, $\mathbf{\Lambda}$ and $\mathbf{v}$ respectively, and c is the index of the reference coefficient in the neighborhood.

The second component of (34) is $\mathbb{P}(z_i \mid \tilde{P})$, which can be obtained using the Bayes rule ($p_z(z)$ denotes the density function of the random variable z:

$$\mathbb{P}(z_i \mid \tilde{P}) = \frac{\mathbb{P}(\tilde{P} \mid z_i)p_z(z_i)}{\int_0^\infty \mathbb{P}(\tilde{P} \mid \alpha)p_z(\alpha)d\alpha}$$

or its discrete form

$$\mathbb{P}(z_i \mid \tilde{P}) = \frac{\mathbb{P}(\tilde{P} \mid z_i)p_z(z_i)}{\sum_{j=1}^{13} \mathbb{P}(\tilde{P} \mid z_j)p_z(z_j)}. \tag{36}$$

where the density of observed noisy neighborhood vector $\tilde{P}$ conditioned on z_i is a zero-mean Gaussian with covariance

$$\mathbf{C}_{\tilde{P}|z_i} := z_i\mathbf{C}_U + \mathbf{C}_N,$$

so that

$$\mathbb{P}(\tilde{P} \mid z_i) = \frac{e^{-\frac{-\tilde{P}^T(z\mathbf{C}_U + \mathbf{C}_N)^{-1}\tilde{P}}{2}}}{\sqrt{|z\mathbf{C}_U + \mathbf{C}_N|}} \quad .$$

Using the above definitions of $\mathbf{v}$ and $\mathbf{\Lambda}$ and the same simplifications as for $E(P \mid \tilde{P}, z_i)$ we obtain

$$\mathbb{P}(\tilde{P} \mid z_i) = \frac{e^{-\frac{1}{2}\sum_{j=1}^{n^s} \frac{v_j^2}{z_j\lambda_{j,j}+1}}}{\sqrt{\Pi_{j=1}^{n^s}(z_i\lambda_{j,j} + 1)}}, \tag{37}$$

The only question left is the form of $p_z(z)$. The authors, after a somewhat puzzling discussion, adopt "a non-informative Jeffrey prior" $p_z(z) \simeq \frac{1}{z}$. Since this function cannot be a density, being non integrable, the function is actually cut off to zero near $z = 0$.

To summarize, the Portilla et al. algorithm is:

Algorithm 9 Portilla et al. wavelet neighborhood denoising (BLS-GSM)

Input: noisy image
Output: denoised image

Parameters: $n_1 \times n_2$ the image size, number of pyramid scales $\log_2(\min(n_1, n_2) - 4)$.
Parameter s, enumeration of all oriented channels at each scale (8 per scale).
Establish n^s, dimension of wavelet neighborhood coefficient (10 or 9).

Apply the wavelet pyramid (29)-(32), respectively to the noise image δ and to the observed image.
Regroup the obtained wavelet coefficients to obtain $\tilde{P}^s$, the wavelet coefficient neighborhoods of rank s and N^s the noise wavelet coefficient neighborhoods of rank s.
for each filter index s **do**
Compute $\mathbf{C}_N^s$ and $\mathbf{C}_{\tilde{P}}^s$, noise and observation covariance matrices of N_s and $\tilde{P}_s$. (In the sequel the subscript s is omitted.) Deduce $\mathbf{C}_U = \mathbf{C}_{\tilde{P}} - \mathbf{C}_N$.
Compute $\{\mathbf{Q}, \mathbf{\Lambda}\}$ the eigenvectors and eigenvalues of $\mathbf{S}^{-1}\mathbf{C}_U\mathbf{S}^{-T}$, where $\mathbf{S}$ is the symmetric square root of $\mathbf{C}_N$, $\mathbf{C}_N = \mathbf{S}\mathbf{S}^T$.
end for
for each wavelet coefficient neighborhood $\tilde{P}$ and $i \in \{1, \ldots, 13\}$ **do**
Compute $\mathbf{M} = \mathbf{S}\mathbf{Q}, \quad \mathbf{v} = \mathbf{M}^{-1}\tilde{P}$
Using (35) obtain $E(P \mid \tilde{P}, z_i)_c = \sum_{j=1}^{n^s} \frac{z_i m_{c,j} \lambda_{j,j} v_j}{z_i \lambda_{j,j} + 1}$, where $m_{c,j}$, $\lambda_{j,j}$ and v_j are the elements of $\mathbf{M}$, $\mathbf{\Lambda}$ and $\mathbf{v}$ respectively, and c is the index of the reference coefficient in the neighborhood.
Apply (36) to get $\mathbb{P}(z_i \mid \tilde{P}) = \frac{\mathbb{P}(\tilde{P}|z_i)p_z(z_i)}{\sum_{j=1}^{13}\mathbb{P}(\tilde{P}|z_j)p_z(z_j)}$, using the value obtained by (37) for $\mathbb{P}(\tilde{P} \mid z_i) = \frac{e^{-\frac{1}{2}\sum_{j=1}^{n^s}\frac{v_j^2}{z_j\lambda_{j,j}+1}}}{\sqrt{\Pi_{j=1}^{n^s}(z_i\lambda_{j,j}+1)}}$.
By (34) finally obtain $E(P \mid \tilde{P}) = \sum_{i=1}^{13} \mathbb{P}(z_i \mid \tilde{P}) E(P \mid \tilde{P}, z_i)$ where $p_z(z) \simeq \frac{1}{z}$ and z_i are quantized uniformly on the interval $[\ln(z_{min}); \ln(z_{max})] = [-20.5, 3.5]$.
end for
Reconstruct the restored image from its restored neighborhood coefficients $E(P \mid \tilde{P})$ by the inverse steerable pyramid.

As we shall see in the synthesis, in spite of its formalism, this method is actually extremely similar to other patch-based Bayesian methods. It has received a more recent extension, reaching state of the art performance, in [97]. This paper proposes an extension of the above method modeling the wavelet coefficients as a global random field of Gaussian scale mixtures.

5.7 K-SVD

The K-SVD method was introduced in [2] where the whole objective was to optimize the quality of sparse approximations of vectors in a learnt dictionary. Even if this article noticed the interest of the technique in image processing tasks, it is in [62] that a detailed study has been led on the denoising of grey-level images. Then, the adjustment to colour images has been treated in [103]. Let us notice that this last article proved that the K-SVD method can also be useful in other image processing tasks, such as non-uniform denoising, demoisaicing and inpainting. For a detailed description of K-SVD the reader is referred to [99] and [101].

The algorithm is divided in three steps. In the two first steps an optimal dictionary and a sparse representation is built for each patch in the image, using among other tools a singular value decomposition (SVD). In the last step, the restored image is built by aggregating the computed sparse representations of all image patches. The algorithm requires an initialization of the dictionary which is updated during the process. The dictionary initialization may contain usual orthogonal basis (discrete cosine transform, wavelets...), or patches from clean images or even from the noisy image itself.

The first step looks for sparse representations of all patches of size κ^2 in the noisy image in vector form $\tilde{U}$ using a fixed dictionary $\mathbf{D}$. A dictionary is represented as a matrix of size $\kappa^2 \times n_{dic}$, with $n_{dic} \geq \kappa^2$, whose columns (the "atoms of the dictionary") are normalized (in Euclidean norm). For each noisy patch $\mathbf{R_i}\tilde{U}$, (where the index $\mathbf{i}$ indicates that the top left corner of the patch is the pixel $\mathbf{i}$, and $\mathbf{R_i}$ is the matrix extracting the patch vector from $\tilde{U}$) a "sparse" column vector $\alpha_\mathbf{i}$ (of size n_{dic}) is calculated by optimization. This vector of coefficients should have only a few non-zero coefficients, the distance between $\mathbf{R_i}\tilde{U}$ and its sparse approximation $\mathbf{D}\alpha_\mathbf{i}$ remaining as small as possible. The dictionary allows one to compute a sparse representation $\alpha_\mathbf{i}$ of each patch $\mathbf{R_i}\tilde{U}$. These sparse vectors are assembled in a matrix α with κ^2 rows and N_p columns where N_p is the number of patches of dimension κ^2 of the image.

More precisely, an ORMP (Orthogonal Recursive Matching Pursuit) gives an approximate solution of the (NP-complete) problem

$$\underset{\alpha_\mathbf{i}}{\text{Arg min}} \, ||\alpha_\mathbf{i}||_0 \quad \text{such that} \quad ||\mathbf{R_i}\tilde{U} - \mathbf{D}\alpha_\mathbf{i}||_2^2 \leq \kappa^2 (C\sigma)^2 \tag{38}$$

where $\|\alpha_\mathbf{i}\|_0$ refers to the l^0 norm of $\alpha_\mathbf{i}$, i.e. the number of non-zero coefficients of $\alpha_\mathbf{i}$. The additional constraint guarantees that the residual has an l^2 norm lower than $\kappa C\sigma$. C is an user parameter. The second step tries to update one by one the columns of the dictionary $\mathbf{D}$ and the representations α to improve the overall fidelity of the patch approximation. The goal is to decrease the quantity

$$\sum_\mathbf{i} \|\mathbf{D}\alpha_\mathbf{i} - \mathbf{R_i}\tilde{U}\|_2^2 \tag{39}$$

while keeping the sparsity of the vectors $\alpha_\mathbf{i}$. We will denote by $\hat{d}_l$ $(1 \leq l \leq n_{dic})$ the columns of the dictionary $\hat{\mathbf{D}}$. First, the quantity (39) is minimized without taking care of the sparsity. The atom $\hat{d}_l$ and the coefficients $\hat{\alpha}_\mathbf{i}(l)$ are modified to make the approximations of all the patches more efficient. For each $\mathbf{i}$, introduce the residue

$$e_\mathbf{i}^l = \mathbf{R_i}\tilde{U} - \hat{\mathbf{D}}\hat{\alpha}_\mathbf{i} + \hat{d}_l \hat{\alpha}_\mathbf{i}(l) \tag{40}$$

which is the error committed by deciding not to use $\hat{d}_l$ any more in the representation of the patch $R_\mathbf{i}\tilde{U}$. Thus $e_\mathbf{i}^l$ is a vector of size κ^2.

These residues are grouped together in a matrix $\mathbf{E}_l$ (whose columns are indexed by $\mathbf{i}$). The values of the coefficients $\hat{\alpha}_{\mathbf{i}}(l)$ are also grouped in a row vector denoted by $\hat{\alpha}^l$. Therefore, $\mathbf{E}_l$ is a matrix of size $\kappa^2 \times N_p$ (recall that N_p is the total number of patches in the image) and $\hat{\alpha}^l$ is a row vector of size N_p. We must try to find a new $\hat{d}_l$ and a new row vector $\hat{\alpha}^l$ minimizing

$$\sum_{\mathbf{i}} \|\hat{\mathbf{D}}\hat{\alpha}_{\mathbf{i}} - \hat{d}_l\hat{\alpha}_{\mathbf{i}}(l) + d_l\alpha^l - \mathbf{R}_{\mathbf{i}}\tilde{U}\|_2^2 = ||\mathbf{E}_l - d_l\alpha^l||_F^2 \tag{41}$$

where the squared Frobenius norm $\|\mathbf{M}\|_F^2$ refers to the sum of the squared elements of $\mathbf{M}$. This Frobenius norm is also equal to the sum of the squared (Euclidean) norms of the columns, and one can be convinced that minimizing (41) amounts to reduce the approximation error caused by $\hat{d}_l$. It is well-known that the minimization of such a Frobenius norm consists in a rank-one approximation, which always admits a solution, practically given by the singular value decomposition (SVD). Using the SVD of $\mathbf{E}_l$:

$$\mathbf{E}_l = \mathbf{U}\mathbf{\Delta}\mathbf{V}^T$$

(where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices and $\mathbf{\Delta}$ is non-negative and decreasing), the updated values of $\hat{d}_l$ and $\hat{\alpha}^l$ are respectively the first column of $\mathbf{U}$ and the first column of $\mathbf{V}$ multiplied by $\mathbf{\Delta}(1,1)$.

After K iterations of these two steps, a denoised patch $\hat{\mathbf{D}}\hat{\alpha}_{\mathbf{i}}$ is available for each patch position $\mathbf{i}$, where $\hat{\mathbf{D}}$ is the final updated dictionary. The third and last (aggregation) step consists in merging the denoised versions of all patches of the image in order to obtain a global estimate. This is achieved by solving the minimization problem

$$\hat{U} = \underset{U_0 \in \mathbb{R}^M}{\operatorname{Arg\,min}} \ \lambda||U_0 - \tilde{U}||_2^2 \ + \ \sum_{\mathbf{i}} ||\hat{\mathbf{D}}\hat{\alpha}_{\mathbf{i}} - \mathbf{R}_{\mathbf{i}}U_0||_2^2 \ ,$$

by the closed formula

$$\hat{U} = \left(\lambda\mathbf{I} + \sum_{\mathbf{i}} \mathbf{R}_{\mathbf{i}}^T\mathbf{R}_{\mathbf{i}}\right)^{-1} \left(\lambda\tilde{U} + \sum_{\mathbf{i}} \mathbf{R}_{\mathbf{i}}^T\hat{\mathbf{D}}\hat{\alpha}_{\mathbf{i}}\right) \ . \tag{42}$$

This amounts for each pixel to average its initial noisy value with the average of all estimates obtained with all patches containing it. The parameter λ controls the tradeoff between these two values and thus measures the fidelity to the initial noisy image.

Mairal et al. [103] proposed to directly extend the algorithm to vector valued images instead of converting the colour image to another colour system decorrelating geometry and chromaticity. The previous algorithm is applied on column vectors which are a concatenation of the R,G,B values. In this way, the algorithm, when updating the dictionary, takes into account the inter-channel correlation. We shall detail the algorithm for grey level images, the colour version simply requires an adaptation of the Euclidean norm to the colour space.

Algorithm 10 K-SVD algorithm for grey level images

Input: noisy image $\tilde{u}$, $\tilde{U}$ in vector form, noise standard deviation σ.
Input: κ^2, dimension of patches (number of pixels).
Input n_{dic}, dictionary size, K iteration number of the dictionary optimization.
Input: initial patch dictionary $\mathbf{D}_{init}$ as matrix with n_{dic} columns and κ^2 rows.
Output: output image in vector form $\hat{U}$.

Collect all noisy patches of dimension κ^2 in column vectors $\mathbf{R_i}\tilde{U}$
Set $\hat{\mathbf{D}} = \mathbf{D}_{init}$.

for k=1 to K **do**

An ORMP is applied to the vectors $\mathbf{R_i}\tilde{U}$ in a such way that a vector of sparse coefficients $\hat{\alpha}_\mathbf{i}$ is obtained verifying $\mathbf{R_i}\tilde{U} \approx \mathbf{D}\hat{\alpha}_\mathbf{i}$.

Introduce $\omega_l = \{\, \mathbf{i} \,|\, \hat{\alpha}_\mathbf{i}(l) \neq 0 \,\}$;
For $\mathbf{i} \in \omega_l$, obtain the residue

$$e_\mathbf{i}^l = \mathbf{R_i}\tilde{U} - \hat{\mathbf{D}}\hat{\alpha}_\mathbf{i} + \hat{d}_l\hat{\alpha}_\mathbf{i}(l) \;;$$

Put these column vectors together in a matrix $\mathbf{E}_l$. Values $\hat{\alpha}_\mathbf{i}(l)$ are also assembled in a row vector denoted by $\hat{\alpha}^l$ for $\mathbf{i} \in \omega_l$;
Update $\hat{d}_l$ and $\hat{\alpha}^l$ as solutions of the minimization problem :

$$(\hat{d}_l, \hat{\alpha}^l) = \underset{d_l, \alpha^l}{\operatorname{Arg\,min}} \|\mathbf{E}_l - d_l\alpha^l\|_F^2 \;.$$

A truncated SVD is applied to the matrix $\mathbf{E}_l$. It provides partially $\mathbf{U}$, $\mathbf{V}$ (orthogonal matrices) and $\mathbf{\Delta}$ (filled in with zeroes except on its first diagonal), such that $\mathbf{E}_l = \mathbf{U}\mathbf{\Delta}\mathbf{V}^T$. Then $\hat{d}_l$ is defined again as the first column of $\mathbf{U}$ and $\hat{\alpha}^l$ as the first column of $\mathbf{V}$ multiplied by $\mathbf{\Delta}(1,1)$.
end for

Aggregation: for each pixel the final result $\hat{U}$ in vector form is obtained thanks to the weighted aggregation:

$$\hat{U} = \left(\lambda\mathbf{I} + \sum_\mathbf{i} \mathbf{R_i^t}\mathbf{R_i}\right)^{-1} \left(\lambda\tilde{U} + \sum_\mathbf{i} \mathbf{R_i^t}\hat{\mathbf{D}}\hat{\alpha}_\mathbf{i}\right) \;.$$

5.8 BM3D

BM3D is a sliding window denoising method extending DCT denoising and NL-means. Instead of adapting locally a basis or choosing from a large dictionary, it uses a fixed basis. The main difference with DCT denoising is that a set of similar patches are used to form a 3D block which is filtered by using a 3D transform, hence its name *Collaborative filtering*. The method has four steps: a) finding the image patches similar to a given image patch and grouping them in a 3D block b) 3D linear transform of the 3D block; c) shrinkage of the transform spectrum coefficients; d) inverse 3D transformation. This 3D filter therefore filters out simultaneously all 2D image patches in the 3D block. By attenuating the noise, collaborative filtering reveals even the finest details shared by the grouped patches. The filtered patches are then returned to their original positions and an adaptive aggregation procedure is applied by taking into account the number of kept coefficients per patch during the thresholding process (see section 4 for more details on aggregation).

The first collaborative filtering step is much improved in a second step using an oracle Wiener filtering. This second step mimics the first step, with two differences. The first difference is that it compares the filtered patches instead of the original patches like described in section 4. The second difference is that the new 3D group (built with the unprocessed image samples, but using the patch distances of the filtered image) is processed by an oracle Wiener filter using coefficients from the denoised image obtained at the first step to approximate the true coefficients given by Theorem 1. The final aggregation step is identical to those of the first step.

The algorithm is extended to colour images through the $Y_oU_oV_o$ colour system. The previous strategy is applied independently to each channel with the exception that similar patches are always selected by computing distances in the channel Y_o.

Algorithm 11 BM3D first iteration algorithm for grey images.

Input: noisy image $\tilde{u}$, σ, noise standard deviation.
Output: output basic estimation $\hat{u}_1$ of the denoised image.

Set parameter $\kappa \times \kappa = 8 \times 8$: dimension of patches.
Set parameter $\lambda \times \lambda = 39 \times 39$: size of search zone in which similar patches are searched.
Set parameter $N_{max} = 16$: maximum number of similar patches retained during the grouping part.
Set parameter $s = 3$: step in both rows and columns between two reference patches.
Set parameter $\lambda_{3D} = 2.7$: coefficient used for the hard thresholding.
Set parameter $\tau = 2500$ (if $\sigma > 40, \tau = 5000$): threshold used to determine similarity between patches.

for each pixel **i**, with a step s in rows and columns **do**

Select a square reference patch $\tilde{P}$ around **i** of size $\kappa \times \kappa$.

Look for square patches $\tilde{Q}$ in a square neighborhood of **i** of size $\lambda \times \lambda$ having a distance to $\tilde{P}$ lower than τ.

if there are more than N_{max} similar patches **then**
keep only the N_{max} closest similar patches to $\tilde{P}$ according to their Euclidean distance.
else
keep 2^p patches, where p is the largest integer such that 2^p is smaller than the number of similar patches
end if

A 3D group $\mathcal{P}(\tilde{P})$ is built with those similar patches.

A bi-orthogonal spline wavelet (Bior 1.5) is applied on every patch contained in $\mathcal{P}(\tilde{P})$.

A Walsh-Hadamard transform is then applied along the third dimension of the 3D group $\mathcal{P}(\tilde{P})$.

A hard thresholding with threshold $\lambda_{3D}\sigma$ is applied to $\mathcal{P}(\tilde{P})$. An associated weight $w_{\tilde{P}}$ is computed :

$$w_{\tilde{P}} = \begin{cases} (N_{\tilde{P}})^{-1} & N_{\tilde{P}} \geq 1 \\ 1 & N_{\tilde{P}} = 0 \end{cases}$$

where $N_{\tilde{P}}$ is the number of retained (non-zero) coefficients.

The estimate $\hat{u}_1^{\tilde{Q},\tilde{P}}$ for each pixel **i** in similar patches $\tilde{Q}$ of the 3D group $\mathcal{P}(\tilde{P})$ is then obtained by applying the inverse of the Walsh-Hadamard transform along the third dimension, followed by the inverse of the bi-orthogonal spline wavelet on every patches of the 3D group.
end for

for each pixel **i do**
Aggregation: recover the denoised value at **i** by averaging all estimates of all patches $\tilde{Q}$ in all 3D groups $\mathcal{P}(\tilde{P})$ containing **i**, the weights being given by the $w_{\tilde{P}}$.
end for

Algorithm 12 BM3D second iteration algorithm for grey images.

Input: noisy image $\tilde{u}$, σ, noise standard deviation.
Input: basic estimation $\hat{u}_1$ obtained at the first step.
Output: final denoised image $\hat{u}$.

Set parameter $\kappa \times \kappa = 8 \times 8$ (up to 12 for high noise level): dimension of patches.
Set parameter $\lambda \times \lambda = 39 \times 39$: size of search zone in which similar patches are searched.
Set parameter $N_{max} = 32$: maximum number of similar patches retained during the grouping part.
Set parameter $s = 3$: step in both rows and columns between two reference patches.
Set parameter $\tau = 400$ (if $\sigma > 40, \tau = 3500$): threshold used to determinate similarity between patches.

for each pixel **i**, with a step s in rows and columns **do**

Take the square reference patches $\tilde{P}$ and $\hat{P}_1$ centered at **i**, of size $\kappa \times \kappa$ in the initial and basic estimation images.

Look for square patches $\hat{Q}_1$ in a square neighborhood of **i** of size $zsize \times zsize$ having a distance lower than τ in the basic estimate image $\hat{u}_1$.

if there are more than N_{max} similar patches **then**
keep only the N_{max} closest similar patches to $\hat{P}_1$ according to their Euclidean distance.
else
keep 2^p patches, where p is the largest integer such that 2^p is smaller than the number of similar patches
end if

Two 3D groups $\mathcal{P}(\tilde{P})$ and $\mathcal{P}(\hat{P}_1)$ are built with those similar patches, one from the noisy image $\tilde{u}$ and one from the basic estimate image $\hat{u}_1$.

A 2D DCT is applied on every patch contained in $\mathcal{P}(\tilde{P})$ and $\mathcal{P}(\hat{P}_1)$.

A Walsh-Hadamard transform is then applied along the third dimension of $\mathcal{P}(\tilde{P})$ and $\mathcal{P}(\hat{P}_1)$.

Denoting by τ_{3D} the 3D transform (2D DCT followed by the Walsh-Hadamard transform) applied on the 3D group, compute the Wiener coefficient
$\omega_{\tilde{P}} = \frac{|\tau_{3D}(\mathcal{P}(\hat{P}_1)|^2}{|\tau_{3D}(\mathcal{P}(\hat{P}_1)|^2 + \sigma^2}$.

The Wiener collaborative filtering of $\mathcal{P}(\tilde{P})$ is realized as the element-by-element multiplication of the 3D transform of the noisy image $\tau_{3D}(\mathcal{P}(\tilde{P}))$ with the Wiener coefficients $\omega_{\tilde{P}}$.

An associated weight $w_{\tilde{P}}$ is computed :

$$w_{\tilde{P}} = \begin{cases} (\|\omega_{\tilde{P}}\|_2)^{-2} & \|\omega_{\tilde{P}}\|_2 > 0 \\ 1 & \|\omega_{\tilde{P}}\|_2 = 0 \end{cases}$$

The estimate $\hat{u}_2^{\tilde{Q},\tilde{P}}$ for each pixel **i** in similar patches $\tilde{Q}$ of the 3D group $\mathcal{P}(\tilde{P})$ is then obtained by applying the inverse of the 1D Walsh-Hadamard transform along the third dimension, followed by the inverse of the 2D DCT on every patch of the 3D group.
end for

for each pixel **i do**
Aggregation: Recover the denoised value $\hat{u}(\mathbf{i})$ at **i** by averaging all estimates of patches $\tilde{Q}$ in all 3D groups $\mathcal{P}(\tilde{P})$ containing **i**, using the weights $\omega_{\tilde{P}}$.
end for

Here we described the basic implementation given in its seminal paper, and which will also be used in the comparison section. Yet, BM3D has several more recent variants that improve its performance. Like for NL-means, there is a variant with shape-adaptive patches [40]. In this algorithm denominated BM3D-SAPCA, the sparsity of image representation is improved in two aspects. First, it employs image patches (neighborhoods) which can have data-adaptive shape. Second, the PCA bases are obtained by eigenvalue decomposition of empirical second-moment matrices that are estimated from groups of similar adaptive-shape neighborhoods. This method improves BM3D especially in preserving image details and introducing very few artifacts. The anisotropic shape-adaptive patches are obtained using the 8-directional LPA-ICI techniques [80].

The very recent development of BM3D is presented in [79], [43], where it is generalized to become a generic image restoration tool, including deblurring.

5.9 The piecewise linear estimation (PLE) method

The ambitious Bayesian restoration model proposed in [155] and [156] is a general framework for restoration, including denoising, deblurring, and inpainting. An image is decomposed into overlapping patches $\tilde{P}_i = \mathbf{A}_i P_i + N_i$ where $\mathbf{A}_i$ is the degradation operator restricted to the patch i, P_i is the original patch, $\tilde{P}_i$ the degraded one, and N_i the noise restricted to the patch. Since we are studying only the denoising problem, we shall take for $\mathbf{A}_i$ the identity. The (straightforward) extension including a linear perturbation operator is out of our scope.

The patch density law is modeled as a mixture of Gaussian distributions $\{\mathcal{N}(\mu_k, \mathbf{C}_k)\}_{1\leq k\leq K}$ parametrized by their means μ_k and covariance matrices $\mathbf{C}_k$. Thus each patch $\tilde{P}_i$ is assumed independently drawn from one of these Gaussians with an unknown index k and a density function

$$p(P_i) = \frac{1}{(2\pi)^{\frac{\kappa^2}{2}} |\mathbf{C}_{k_i}|^{\frac{1}{2}}} e^{-\frac{1}{2}(P_i-\mu_k)^T \mathbf{C}_{k_i}^{-1}(P_i-\mu_k)}.$$

Estimating all patches P_i from their noisy observations $\tilde{P}_i$ amounts to solve the following problems:

- to estimate the Gaussian parameters $(\mu_k, \mathbf{C}_k)_{1\leq k\leq K}$ from the degraded data $\tilde{P}_i$;
- to identify the index k_i of the Gaussian distribution generating the patch P_i;
- to estimate P_i from its corresponding Gaussian distribution $(\mu_{k_i}, \mathbf{C}_{k_i})$ and from its noisy version $\tilde{P}_i$.

In consequence PLE [156]) has two distinct steps in the estimation procedure. In an E-step (E for Estimate), the Gaussian parameters $(\mu_k, \mathbf{C}_k)_k$ are known and for each patch the maximum *a posteriori* (MAP) estimate $\hat{P}_i^k$ is computed with each Gaussian model. Then the best Gaussian model k_i is selected to obtain the estimate $\hat{P}_i = \hat{P}_i^{k_i}$.

In the M-step (M for Model), the Gaussian model selection k_i and the signal estimates $\hat{f}_i$ are assumed known for all patches i, and permit to estimate again the Gaussian models $(\mu_k, \mathbf{C}_k)_{1\leq k\leq K}$. According to the terminology of section 4.2, this section gives the *oracle* permitting to estimate in the E-step the patches by a Wiener type filter.

For each image patch with index i the patch estimation and its model selection is obtained by maximizing the *log* a-posteriori probability $\mathbb{P}(P_i \mid \tilde{P}_i, k)$,

$$\begin{aligned}
(\hat{P}_i, k_i) &= \arg\max_{P,k} \log \mathbb{P}(P_i \mid \tilde{P}_i, \mathbf{C}_k) && (43)\\
&= \arg\max_{P,k} \Big(\log \mathbb{P}(\tilde{P}_i \mid P_i, \mathbf{C}_k) + \log \mathbb{P}(P_i \mid \mathbf{C}_k) \Big) && (44)\\
&= \arg\min_{P_i,k} \Big(\|P_i - \tilde{P}_i\|^2 + \sigma^2 (P_i-\mu_k)^T \mathbf{C}_k^{-1} (P_i-\mu_k) + \sigma^2 \log|\mathbf{C}_k| \Big) && (45)
\end{aligned}$$

where the second equality follows from the Bayes rule and the third one assumes a white Gaussian noise with diagonal matrix $\sigma^2\mathbf{I}$ (of the dimension of the patch) and $P_i \simeq \mathcal{N}(\mu_k, \mathbf{C}_k)$. This minimization can be made first over P_i, which amounts to a linear filter, and then over k, which is a simple comparison of a small set of real values. The index k being fixed, the optimal P_i^k satisfies

$$P_i^k = \arg\min_{P_i} \left(\|P_i - \tilde{P}_i\|^2 + \sigma^2 (P_i - \mu_k)^T \mathbf{C}_k^{-1} (P_i - \mu_i) + \log |\mathbf{C}_k| \right)$$

and therefore

$$P_i^k = \mu_k + (\mathbf{I} + \sigma^2 \mathbf{C}_{k_i}^{-1})^{-1} (\tilde{P}_i - \mu_k),$$

which is the formula (18) already seen in section 5.2. Then the best Gaussian model k_i is selected as

$$k_i = \arg\min_{k} \left(\|P_i^k - \tilde{P}_i\|^2 + \sigma^2 (P_i^k - \mu_k)^T \sigma_k^{-1} (P_i^k - \mu_i) + \log |\mathbf{C}_k| \right).$$

Assuming now that for each patch P_i its model k_i and its estimate $\hat{P}_i$ are known, the next question is to give for each k the maximum likelihood estimate for $(\mu_k, \mathbf{C}_k)$ knowing all the patches assigned to the k-th cluster $\mathcal{C}_k$, namely,

$$(\mu_k, \mathbf{C}_k) = \arg\max_{\mu_k, \mathbf{C}_k} \log \mathbb{P}(\{\hat{P}_i\}_{i \in \mathcal{C}_k} \mid \mu_k, \mathbf{C}_k).$$

This yields the empirical estimate

$$\mu_k = \frac{1}{\#\mathcal{C}_k} \sum_{i \in \mathcal{C}_k} \hat{P}_i, \quad \mathbf{C}_k = \frac{1}{\#\mathcal{C}_k - 1} \sum_{i \in \mathcal{C}_k} (\hat{P}_i - \mu_k)(\hat{P}_i - \mu_k)^T,$$

which are the estimates (19) also used in section 5.2.

Finally the above MAP-EM algorithm is iterated and the authors observe that the MAP probability of the observed signals $\mathbb{P}(\{\hat{P}_i\}_i \mid \{\tilde{P}_i\}_i, \{\mu_k, \mathbf{C}_k\}_k)$ always increases. The clusters and the patch estimates converge. Nevertheless, this algorithm requires a good initialization. Noticing that having the adequate Gaussians describing the patch space amounts to have a good set of PCA bases for intuitive patch clusters, the authors create 19 orthogonal bases in the following way: one of them, say $k = 0$, is the classic DCT basis and corresponds to the "texture cluster". The others are obtained by fixing 18 uniformly sampled directions in the plane. For each direction, PCA is applied to a set of patches extracted from a synthetic image containing an edge in that direction. The PCA yields an oriented orthonormal basis. In short, the initial clusters segment the patch set in 18 classes of patches containing an edge or an oriented texture, and one class containing the more isotropic patches.

The study in this paper gives an interpretation of the patch dictionary methods such as K-SVD and fuses them with Bayesian methods and the Wiener method. In particular the paper shows how the K-SVD method actually learns patches that are quite similar to oriented patches obtained by the above procedure, as illustrated in Fig. 12. This analysis structures the synthetic view proposed in section 7.

6 Comparison of denoising algorithms

In this section we shall compare the following "state of the art" denoising algorithms: the sliding DCT filter as specified in Algorithm 3, the wavelet neighborhood Gaussian scale mixture (BLS-GSM) algorithm, as specified in Algorithm 9, the classical vector valued NL-means as specified in Algorithm 4, the BM3D algorithm as specified in Algorithms 11 and 12, the K-SVD denoising method as described in Algorithm 10 and the Non-local Bayes algorithm as specified in Algorithm 5. These algorithms have been chosen for two reasons. First they have a public and completely transparent code available, which is in agreement with their present description. Second, they all

Algorithm 13 Piecewise linear estimation (PLE)

Input: noisy image $\tilde{u}$ given by the family of its noisy patches $(\tilde{P}_i)_i$, initial set of 19 Gaussian models $\mathcal{N}(\mu_k, \mathbf{C}_k)$ obtained as: a) the 18 PCAs of the patches of 18 synthetic edge images, each with a different orientation; b) a Gaussian model with a diagonal covariance matrix on the DCT basis.
Output: denoised image $\hat{u}$

E-STEP
for all patches $\tilde{P}_i$ of the noisy image **do**

  for each k **do**
    Estimate the MAP of P_i knowing k: $P_i^k = \mu_k + (\mathbf{I} + \sigma^2 \mathbf{C}_k^{-1})^{-1} \tilde{P}_i$.
  end for
  Select the best Gaussian model k_i for P_i as
  $k_i = \arg\min_k \left(\|P_i^k - \tilde{P}_i\|^2 + \sigma^2 (P_i^k - \mu_k)^T \mathbf{C}_k^{-1} (P_i^k - \mu_i) + \log |\mathbf{C}_k| \right)$.
  Obtain the best estimate of P_i knowing the Gaussian models $(\mu_k, \mathbf{C}_k)$, $\hat{P}_i = P_i^{k_i}$.
end for

M-STEP
for all k **do**
  Compute the expectation μ_k and covariance matrix $\mathbf{C}_k$ of each Gaussian by
  $\mu_k = \frac{1}{\#\mathcal{C}_k} \sum_{i \in \mathcal{C}_k} \hat{P}_i, \quad \mathbf{C}_k = \frac{1}{\#\mathcal{C}_k - 1} \sum_{i \in \mathcal{C}_k} (\hat{P}_i - \mu_k)(\hat{P}_i - \mu_k)^T$.
end for
Iterate **E-STEP** and **M-STEP**
Aggregation: Obtain the pixel value of the denoised image $u(\mathbf{i})$ as a weighted average of all values of all denoised patches P_i which contain $\mathbf{i}$.

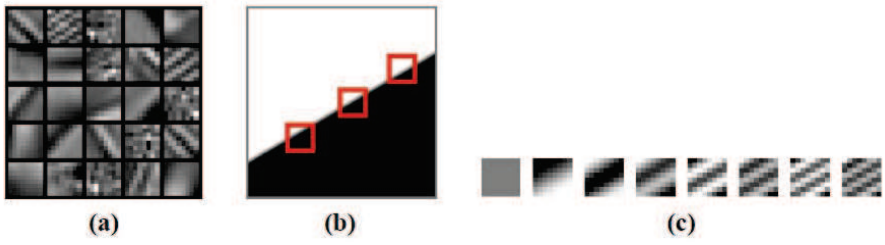


Figure 12: Taken in [156], this figure shows : (a) typical dictionary atoms learnt from the classic image Lena with K-SVD; (b)-(d) the numerical procedure to create one of the oriented PCAs; (b) a synthetic edge image. Patches 8×8 touching the edge are used to calculate an initial PCA basis; (c) the first 8 patches of the PCA basis (ordered by the larger eigenvalue).

represent distinct denoising principles and therefore illustrate the methodological progress and the diversity of denoising principles.

The comparison, using when possible the public IPOL algorithms `http://www.ipol.im/`, will be based on four quantitative and qualitative criteria: the visualization of the *method noise*, namely the part of the image that the algorithm has taken out as noise, the visual verification of the *noise to noise* principle, and the *mean square error* or *PSNR* tables. Last but not least the *visual quality* of the restored images must of course be the ultimate criterion. It is easily seen that a single criterion is not enough to judge a restoration method. A good denoising solution must have a high performance under all mentioned criteria.

6.1 "Method noise"

The difference between the original image and its filtered version shows the "noise" removed by the algorithm. This procedure was introduced in [19] and this difference was called *method noise* by the authors. The authors pointed out that the method noise should look like a noise, at least in case of additive white noise. A visual inspection of this difference tells us which geometrical features or details have been unduly removed from the original. Only human perception is able to detect these unduly removed structures in the "method noise". Furthermore for several classical algorithms like the Gaussian convolution, anisotropic filters, neighborhood filters or wavelet thresholding algorithms, a closed formula permits to analyze the method noise mathematically and thus gives an explanation of observed structured image differences when applying the denoising method [24]. Such an analysis is unfortunately not available and not easy for the state of the art algorithms which are compared in this section. The degree of complexity of each method does not allow for a mathematical study of the method noise. Therefore the evaluation of this criterion will be based only on visual inspection.

When the standard deviation of the added noise is higher than contrast in the original image, a visual exploration of the method noise is nevertheless not reliable. Image features in the method noise may be hidden in the removed noise. For this reason, the evaluation of the method noise should not rely on experiments where a white noise with standard deviation larger than 5 or 10 has been added to the original.

Fig. 13 displays the method noise for the state of the art algorithms being compared in this section, when a Gaussian white noise of standard deviation $\sigma = 5$ has been added. The image differences have been rescaled from $[-4\sigma, 4\sigma]$ to $[0, 255]$ for visualization purposes, and values outside this range have been saturated. By a first visual inspection, it is noticed that all methods have a difference similar to a white noise. This is an outstanding properties of these algorithms, which is not shared by classical denoising techniques such as anisotropic filtering, total variation minimization or wavelet thresholding (see [17] for a more detailed study). It is also immediately observed that the magnitude of the method noise of NL-means and K-SVD is larger than for the rest of the methods. This is corroborated by the standard deviation of each residual noise (see Fig. 13), which is around 5.7 for NL-means and K-SVD, around 4.7 for DCT denoising and around 4.25 for the other algorithms. DCT-denoising, BLS-GSM, BM3D and NL-Bayes keep the transform coefficients that are larger than the ones predicted by noise. This explains why they remove little noise in textured or edge regions. This fact can be easily noticed in Fig. 14 where a piece of the residual noise of Fig. 13 has been enlarged. The amplitude of the noise removed by NL-means and K-SVD is uniform all over the image, while it depends on the underlying image for the rest of the algorithms.

6.2 The "noise to noise" principle

The *noise to noise* principle, introduced in [23], requires that a denoising algorithm transforms white noise into white noise. This paradoxical requirement seems to be the best way to characterize artifact-free algorithms. The transformation of a white noise into any correlated signal creates

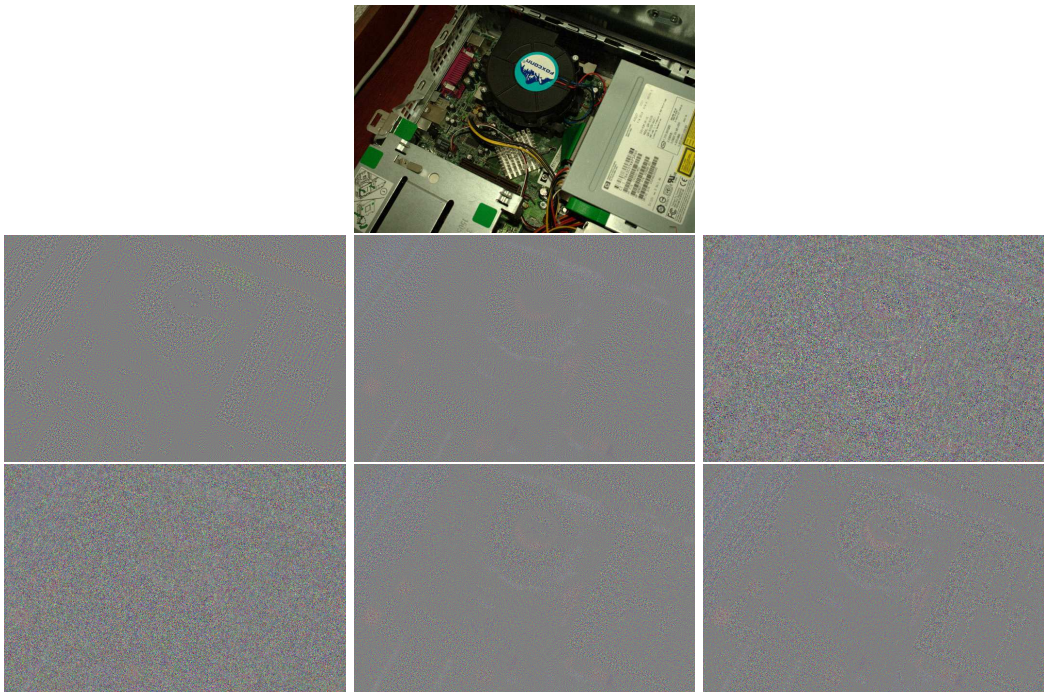


Figure 13: Display of method noise. The noisy image was obtained by adding a Gaussian white noise of standard deviation 5. From top to bottom and left to right: slightly noisy image, DCT sliding window (std = 4.69), BLS-GSM (std = 4.28), NL-means (std = 5.78), K-SVD (std = 5.67), BM3D (std = 4.25) and Non-local Bayes (std = 4.28). All methods have a difference similar to a white noise even if the magnitude of the NL-means and K-SVD differences is larger. This is corroborated by the standard deviation of each residual noise. Due to the thresholding nature of DCT, BLS-GSM, BM3D and NL-Bayes, which alter little the coefficients larger than the ones predicted by noise, noise is not removed in textured and edge zones. This can be easily noticed in Fig. 14 where a piece of the residual noises has been enlarged.

structure and artifacts. Only white noise is perceptually devoid of structure, as was pointed out by Attneave [6].

The noise to noise of classical denoising algorithms was studied in [23], where it was shown that neighborhood filters and, asymptotically, NL-means transform a white noise into a white noise. The convolution with a Gauss kernel keeps the low frequencies and cancels the high ones. Thus, the filtered noise actually shows big grains due to its prominent low frequencies. Noise filtered by a wavelet or DCT thresholding is no more a white noise. The few coefficients with a magnitude larger than the threshold are spread all over the image. The pixels which do not belong to the support of one of these coefficients are set to zero. The visual result is a constant image with superposed wavelets or cosines if the DCT is used. The mathematical analysis of the rest of algorithms is not feasible due to its degree of complexity. Thus, only a visual inspection of this filtered noise is possible.

The methodology adopted to process the *noise to noise* and to show it is the following:

- Since most recent methods process colour images (except BLS-GSM), the *noise to noise* is applied on a colour flat image, i.e. an image with three channels with slightly different[4] values: $RGB = (127, 128, 129)$;
- To reduce the variations due to the random nature of the noise, the tests are performed on

[4]Values are different on each channel in order to force the algorithm to consider this image as a colour image, and not a grey image with a single channel.

relatively large noise images. The chosen size is 1024×1024. The PSNR and RMSE results become then fairly independent of the simulated noise;

- Noise is added on each channel independently. It therefore is a colour noise, and its standard deviation is equal to 30 on each channel of the flat original image;
- All compared algorithms are processed on this noisy image;
- The denoised image is displayed. The mean on every channel is set to 128, and the difference to this mean is enhanced by a factor 5. A small part with size 256×256 of the denoised image is shown after zoom in in 15.

The results in PSNR and RMSE are summarized in the following table:

Method	PSNR	RMSE
NL-Bayes	45.45	1.36
BM3D	45.03	1.43
NL-means	41.45	2.16
TV denoising	41.06	2.26
DCT denoising	40.91	2.30
K-SVD	38.44	3.05

The "order" of performance of the methods is almost respected, except for TV denoising, which shows a really good result compared to K-SVD. Fig. 15 displays the filtered noise images by several state of the art algorithms.

As expected, threshold-based methods present noticeable artifacts, in particular DCT denoising and BM3D. The NL-means result reflects the size of the search zone, and therefore leaves behind a low-frequency oscillation. Despite its good results, TV denoising presents a lot of artifacts which do not look like noise, and are uglier than the K-SVD artifacts. Only NL-Bayes has no artifacts. Indeed, it detects flat patches and replaces them by their mean. This trick could actually be applied to all algorithms. Last but not least, each method leaves a sizeable low-frequency noise, which could be removed with a multi-scale approach.

6.3 Comparing visual quality

The visual quality of the restored image is obviously a necessary, if not sufficient, criterion to judge the performance of a denoising algorithm. It permits to control the absence of artifacts and the correct reconstruction of edges, texture and fine structure. Figures 17-19 display the noisy and denoised images for the algorithms under comparison for noise standard deviations of 20, 30 and 40.

Figure 17 presents an image with straight edges and flat and fine structures with a noise of standard deviation 20. The main artifacts are noticeable in the DCT, BLS-GSM and K-SVD denoised images. These are the most local algorithms and therefore have more trouble in removing the low frequencies of the noise. As a consequence, the denoised images present many low frequency colour artifacts in flat and dark zones. These artifacts are noticeable for all these algorithms even if all use a different strategy to deal with colour images. DCT uses the $Y_oU_oV_o$, K-SVD a vector valued algorithm and BLS-GSM is applied independently to each RGB component. NL-means does not suffer of these noise low frequency problems, but it leaves some isolated noise points on non-repetitive structures, mainly on corners. These isolated noise points could be attenuated by using the $Y_oU_oV_o$ colour space instead of the vector valued algorithm. In this experience, BM3D and Non-Local Bayes give a similar performance and superior to the rest of algorithms.

Figures 18 and 19 illustrate again the low frequency colour artifacts of DCT, BLS-GSM and K-SVD. In these figures, DCT and BLS-GSM also suffer of a strong Gibbs effect near all image

boundaries. This Gibbs effect is nearly not noticeable in the denoised image by K-SVD, since the use of the whole dictionary permits to better reconstruct edges when the right atoms are present in the dictionary. The NL-means denoised image has no visual artifacts but is more blurred than those given by BM3D and Non-Local Bayes, that have a clearly superior performance to the rest of the algorithms. The BM3D denoised image has some Gibbs effect near edges, which sometimes degrades the visual quality of the solution. Non-Local bayes image shows no artifacts. It preserves often better textures than BM3D, by which the trees and vegetation can be slightly blurred by the use of the linear transform threshold.

In short, the visual quality of DCT, BLS-GSM and K-SVD is inferior to that of NL-means, BM3D and NL-Bayes, because of strong colour noise low frequencies in flat zones, and of a Gibbs effect. NL-means does not show noticeable artifacts but the denoised image is more blurred than those of BM3D and Non-Local Bayes. BM3D still has some Gibbs effect due to the use of a single basis for all pixels and a slightly inferior noise reduction, compared to Non-Local Bayes.

6.4 Comparing by PSNR

The mean square error is the square of the Euclidean distance between the original image and its estimate. In the denoising literature an equivalent measurement, up to a decreasing scale change, is the PSNR,

$$PSNR = 10 \log_{10} \left(\frac{255^2}{MSE} \right).$$

These numerical quality measurements are the most objective, since they do not rely on any visual interpretation. Tables 5 and 6 display the PSNR of state of the art denoising methods using the images in Fig. 16 and several values of σ from 2 to 40.

Before jumping to conclusions, we would like to point out that such a PSNR comparison is just informative, and cannot lead to an objective ranking of algorithms. Indeed, what is really needed is a comparison of denoising principles. To compare them, these denoising principles must be implemented in denoising recipes containing several ingredients. Since the PSNR difference between recipes is tight, the way such or such generic tool is implemented, and the degree of sophistication with which each principle is implemented do matter. For example, two of our readers have pointed out to us[5] that an experimental analysis carried out exclusively on color images does not permit a comparison between the different strategies devised to take advantage of spatial redundancy. They suggest to complement the denoising results on color images with experiments on grayscale images. Then it would be possible to: 1) compare the degree of success of these different denoising principles in exploiting spatial redundancy; 2) evaluate the effectiveness of the various ways in which these grayscale algorithms are extended to color data.

In short, these authors do not share our analysis herewith, and the way conclusions can be drawn from the experimental results, because these results are very much influenced by the way color data is treated while much of the conclusions are applied about the relative effectiveness in exploiting spatial redundancy.

For the same reasons, these authors also disagree with the taxonomy summarized in table 7, where it seems that the extension to color is to be considered as a feature of a particular algorithm. Some methods are applied to color data in a very simple non-adaptive way and thus cannot be expected to fully decorrelate the color channels. This is for instance the case of BM3D, which uses a YUV/Opp color transformation. Data-adaptive color transformations for multispectral data are considered in [42]. This adaptive method provides substantially better results than a standard color transformation.

Another reason for being cautious, is that all methods with some existence have actually variants, and we are using the basic algorithms as they were announced in their seminal paper. For example, it is shown in [76] that BM3D can be slightly improved for heavy noise > 40 by changing the method parameters.

[5] Alessandro Foi, Vladimir Katkovnik, personal communication.

	$\sigma = 2$					
	NL-Bayes	BM3D	BLS-GSM	K-SVD	NL-means	DCT denoising
Alley	**45.42**	44.95	-	41.51	42.75	44.58
Computer	**45.96**	45.22	44.69	44.52	44.03	44.54
Dice	**49.00**	48.86	48.59	47.79	48.51	48.39
Flowers	**47.77**	47.31	47.12	47.09	46.36	47.05
Girl	**47.56**	47.40	47.14	47.28	46.96	46.76
Traffic	**45.33**	44.56	44.15	43.80	43.55	44.26
Trees	**43.51**	43.07	-	42.05	42.22	42.95
Valldemossa	**45.17**	44.68	44.41	40.08	43.33	44.50
Mean	**46.22**	45.76	-	44.27	44.71	45.37
	$\sigma = 5$					
	NL-Bayes	BM3D	BLS-GSM	K-SVD	NL-means	DCT denoising
Alley	**39.24**	38.95	-	38.45	37.18	38.37
Computer	**40.69**	39.98	39.30	39.58	38.86	39.03
Dice	**46.09**	45.80	45.21	45.27	45.12	45.22
Flowers	**43.44**	42.99	42.76	43.09	42.05	42.78
Girl	**44.26**	44.03	43.70	43.59	43.44	43.36
Traffic	**39.70**	38.67	38.10	38.75	37.50	38.21
Trees	**36.70**	36.10	-	35.61	34.69	35.76
Valldemossa	**38.73**	38.33	38.02	37.87	35.94	37.94
Mean	**41.11**	40.61	-	40.28	39.35	40.08
	$\sigma = 10$					
	NL-Bayes	BM3D	BLS-GSM	K-SVD	NL-means	DCT denoising
Alley	**35.05**	34.82	-	34.29	33.53	34.22
Computer	**36.58**	36.28	35.47	35.79	35.44	35.34
Dice	**43.30**	43.02	42.21	41.71	42.06	42.22
Flowers	**39.52**	39.49	39.10	39.31	38.49	39.03
Girl	**41.69**	41.45	41.14	40.29	40.42	40.55
Traffic	**34.93**	34.54	33.92	34.69	33.89	34.11
Trees	**36.70**	36.10	-	35.61	29.42	30.92
Valldemossa	**38.73**	38.33	38.02	37.87	32.02	33.45
Mean	**37.06**	36.83	-	36.31	35.66	36.23

Table 5: PSNR table for $\sigma = 2$, 5 and 10. Only the three first digits are actually significant; the last one may vary with different white noise realizations.

In short, the following PSNR comparison on color images must be taken for what it is; it gives some hints and these hints depend on the particular implementation of the denoising principles. We observe in the results that DCT denoising, GLS-GSM, K-SVD and NL-means have a similar PSNR. The relative performance of the methods depends on the kind of image and on noise level σ. On average, K-SVD and BLS-GSM are slightly superior to the other two, even if this is not the case visually, where K-SVD and BLS-GSM have a poor visual quality compared to NL-means. In all cases, BM3D and Non-local Bayes have a better PSNR performance than the others. Because of a superior noise reduction in flat zones and the presence of less artifacts of Non-local Bayes, the PSNR of BM3D is slightly inferior to Non-local Bayes. BM3D seems to retain the best conservation of detail. Some ringing artefacts near boundaries can probably be eliminated by the same trick as Non-local Bayes, namely detecting and giving a special treatment to flat 3D groups.

7 Synthesis

We have showed that all methods either already use, or should adopt the same three *generic denoising tools* described in section 4. Since all methods denoise not just the pixel, but a whole

	$\sigma = 20$					
	NL-Bayes	BM3D	BLS-GSM	K-SVD	NL-means	DCT denoising
Alley	**31.36**	31.23	-	30.55	29.94	30.21
Computer	**33.08**	32.71	31.89	31.96	31.59	31.45
Dice	**40.19**	39.93	39.00	37.23	38.17	38.67
Flowers	**35.87**	35.85	35.34	35.24	34.56	34.89
Girl	**38.92**	38.71	38.49	36.36	36.81	37.27
Traffic	**31.14**	30.83	30.14	30.70	30.12	29.98
Trees	**27.22**	26.92	-	26.88	26.28	26.27
Valldemossa	**29.81**	29.57	26.97	29.08	28.37	28.91
Mean	**33.45**	33.22	-	32.25	31.98	32.20
	$\sigma = 30$					
	NL-Bayes	BM3D	BLS-GSM	K-SVD	NL-means	DCT denoising
Alley	**29.42**	29.33	-	28.60	27.58	28.25
Computer	**31.00**	30.67	29.90	29.84	28.98	29.20
Dice	**38.20**	37.88	37.05	36.52	37.18	35.89
Flowers	33.67	**33.73**	33.19	33.54	32.66	32.46
Girl	**37.12**	36.97	36.91	35.38	35.54	34.67
Traffic	**29.08**	28.87	28.20	28.60	27.40	27.87
Trees	**24.95**	24.64	-	24.52	23.29	23.83
Valldemossa	**27.51**	27.30	26.97	26.80	25.55	26.48
Mean	**31.37**	31.17	-	30.48	29.77	29.83
	$\sigma = 40$					
	NL-Bayes	BM3D	BLS-GSM	K-SVD	NL-means	DCT denoising
Alley	**28.16**	28.08	-	27.29	26.30	27.14
Computer	**29.55**	29.15	28.52	28.25	27.31	27.44
Dice	**36.91**	36.28	35.50	34.49	35.31	33.06
Flowers	31.94	**32.10**	31.68	31.90	30.99	30.80
Girl	**36.09**	35.62	35.61	33.73	34.03	32.01
Traffic	**27.67**	27.50	26.93	27.19	26.01	26.49
Trees	**23.35**	23.17	-	23.06	21.91	22.46
Valldemossa	**27.51**	25.78	25.50	25.28	24.10	25.08
Mean	**30.15**	29.71	-	28.90	28.25	28.05

Table 6: PSNR table for $\sigma = 20$, 30 and 40.

neighborhood, they give several evaluations for each pixel. Thus, they all use an aggregation step. There is only one method for which the aggregation is not explicitly stated as such, the wavelet neighborhood (BLS-GSM) algorithm. Nevertheless, a closer examination shows that it denoises not one, but some 49 wavelet channels for a 512×512 image. The used wavelet transform is redundant. Thus, an aggregation is implicit in its final reconstruction step from all channels. BLS-GSM is also patch-based. Indeed, each "wavelet neighborhood" contains a 3×3 patch of a wavelet channel, complemented with one more sample from the down-scale channel sharing the same orientation. Thus, like the others, this algorithm builds Bayesian estimates of patches. The difference is that the patches belong to the wavelet channels. Each one of these channels is denoised separately, before the reconstruction of the image from its wavelet channels.

In short, even if the BLS-GSM formalization looks at first different from the other algorithms, it relies on similar principles: it estimates patch models to denoise them, and aggregates the results. But, it also is the only multiscale algorithm among those considered here. Indeed, it denoises the image at all scales. Furthermore, it introduces a scale interaction. These features are neglected in the other algorithms and might make a significant difference in future algorithms.

It may be asked why its performance is slightly inferior to that of the current state of the art algorithms. First of all, this algorithm, like many wavelet based algorithms, has not proposed a good solution to deal with colour. Applying the colour space tool of section 4.3 can probably bring a PSNR improvement. The paper does not specify if there is an aggregation step, but a first aggregation step is possible (the second aggregation being implicit in the reconstruction step from all channels, that are redundant). Indeed, each wavelet channel patch contains ten coefficients, and these coefficients are therefore estimated ten times. These estimates might be aggregated.

7.1 The synoptic table

Table 7 shows a synopsis of the ten methods that have been thoroughly discussed. The classification criteria are:

The denoising principle of the method. Our task here is to show that, in spite of the different language used by each method, the underlying principles actually converge. The dominant principle is to compute a linear minimum least square estimator (LMMSE) after building a Bayesian patch model. As a matter of fact, even if this is not always explicit, *all* methods follow very closely the same LMMSE estimator principle. For example the DCT threshold is nothing but a Wiener thresholding version of the Bayesian LMMSE. This threshold is used because the DCT of the underlying noiseless image is actually unknown. The same argument applies for Nonlocal Means, which was interpreted as an LMMSE in section 5.1. A close examination of K-SVD can convince a practitioner that this algorithm is very close to EPL, PLOW or EPLL, and conversely. Indeed, the patch clustering performed in these three algorithms interprets the patch space as a redundant dictionary. Each cluster is treated by a Bayesian estimator as a Gaussian vector, for which an orthogonal eigenvector basis is computed. This basis is computed from the cluster patches by PCA. Thus, EPL, PLOW and EPLL actually deliver a dictionary, which is the union of several orthogonal bases of patches. EPL, PLOW and EPLL select for each noisy patch one of the bases, on which the patch will be sparse. In short, like K-SVD, they compute for each patch a sparse representation in an over-complete dictionary. In this argument, we follow the simple and intelligent interpretation proposed with the PLE method in [156], [155]. Their method was summarized by the authors as follows:

> An image representation framework based on structured sparse model selection is introduced in this work. The corresponding modeling dictionary is comprised of a family of learnt orthogonal bases. For an image patch, a model is first selected from this dictionary through linear approximation in a best basis, and the signal estimation is then calculated with the selected model. The model selection leads to a guaranteed near optimal denoising estimator. The degree of freedom in the model selection is

equal to the number of the bases, typically about 10 for natural images, and is significantly lower than with traditional over-complete dictionary approaches, stabilizing the representation.

From the algorithmic viewpoint, EPLL is a variant of PLE, but used in a different setting. The comparison of these two almost identical Gaussian mixture models is of particular interest. EPLL is applied to a huge set of patches (of the order of 10^{10}) united in some 200 clusters. PLE is applied with 19 clusters learnt each from some 64 patches. Thus, the open question is: how many clusters and how many learning patches are actually necessary to obtain the best PSNR? The disparity between these figures is certainly too large to be realistic.

We must finally wonder if transform thresholding methods fit into the united view of all algorithms. The Bayesian-Gaussian estimate used by most mentioned algorithms can be interpreted as a Wiener filter on the eigenvector basis of the Gaussian. It includes sometimes a threshold (to avoid negative eigenvalues for the covariance matrix of the Gaussian vector). Thus, the only difference between Bayesian-Gaussian methods and the classic transform thresholding is that in the Bayesian methods the orthogonal basis is adapted to each patch. Therefore, they appear to be a direct extension of transform thresholding methods, and have logically replaced them. BM3D combines several linear transform thresholds (2D-bior 1.5, 2D-DCT, 1D-Walsh-Hadamard), applied to the 3D block obtained by grouping similar patches. Clearly, it has found by a rather systematic exploration the right 2D orthogonal bases, and therefore does not need to estimate them for each patch group.

We shall now reunite two groups of methods that are only superficially different. Non-local Means, Non-local Bayes, Shotgun-NL, and BM3D denoise a patch after comparing it to a group of similar patches. The other five patch-based Bayesian methods *do not perform a search for similar patches.*

These other patch methods, PLE, PLOW, EPLL, BLS-GSM and K-SVD, process globally the "patch space" and construct patch models. Nevertheless, this difference is easily reduced. Indeed, EPL, PLOW and EPLL segment the patch space into a sufficient number of clusters, each one endowed with a rich structure (an orthonormal basis). Thus, the patches contributing to the denoising of a given patch estimation are not compared to each other, but they are compared to the clusters. Similarly, the dictionary based methods like K-SVD propose over-complete dictionaries learnt from the image or from a set of images. Finding the best elements of the dictionary to decompose a given patch, as K-SVD does, amounts to classify this patch. This is what is suggested by the authors of PLE in [156]: the dictionary is a list of orthogonal bases which are initiated by sets of oriented edges. Each basis is therefore associated with an orientation (plus one associated with the DCT). Thus PLE is very similar to BLS-GSM, which directly applies a set of oriented filters. Another link between the Bayesian method and sparse modeling is elaborated in [159].

Patches The second column in the classification table 7 indicates the number of patches used for the denoising method, and where they are found. The trivial DCT uses only the current patch to denoise it; Non-local Means, Non-local Bayes and BM3D compare the reference patch with a few hundred patches in its spatial neighborhood; PLE, PLOW, BLS-GSM and K-SVD compare each noisy patch to a learnt model of all image patches; finally Shotgun-NL and EPLL involve in the estimation a virtually infinite number of patches. Surprisingly enough, the performance of all methods are relatively similar. Thus, the huge numbers used to denoise in Shotgun-NL and EPLL clearly depend on the fact that the patches were not learnt from the image itself, and their number can arguably be considerably reduced.

Size (of patches) The third column in our synoptic table compares the patch sizes. All methods without an exception try to deduce the correct value of a given pixel $\mathbf{i}$ by using a neighborhood of $\mathbf{i}$ called patch. This patch size goes from 3×3 to 8×8, with a strong dominance of 8×8 patches. Nevertheless, the size of the patches obviously depends on the amount of noise and should be adapted to the noise standard deviation. For very large noises, a size 8×8 can be insufficient,

Method	Denoising principle	Patches	size	Aggr.	Oracle	Colour
DCT	transform threshold	one	8	yes	yes	yes
Non-local Means	average	neighborhood	3	yes	yes	no
Non-local Bayes	Bayes	neighborhood	3-7	yes	yes	yes
PLOW	Bayes, 15 clusters	image	11	yes	yes	yes
Shotgun-NL	Bayes	10^{10} patches	3-20	yes	no	no
EPLL	Bayes, 200 clusters	2.10^{10} patches	8	yes	yes	yes
BLS-GSM	Bayes in GSM	Image	3	yes	no	no
K-SVD	sparse dictionary	Image	8	yes	yes	yes
BM3D	transform threshold	neighborhood	8-12	yes	yes	yes
PLE	Bayes, 19 clusters	Image	8	yes	yes	yes

Table 7: Synoptic table of all considered methods.

while for small noises small patches might be better. As a matter of fact, all articles focus on noise standard deviations around 30 (most algorithms are tested for σ between 20 and 80). There is little work on small noise (below 10). For large noise, above 50, most algorithms do not deliver a satisfactory result and most papers show denoising results for $20 \leq \sigma \leq 40$. This may also explain the homogeneity of the patch size.

Aggregation, Oracle, Colour A good sign of maturity of the methods is that the three generic improvement tools described in section 4 are used by most methods. When a "no" is present in the table on these three columns, this indicates that the method can probably be substantially improved with little effort by using the corresponding tool. Shotgun-NL and BLS-GSM can probably gain some decibels by aggregation and by the Oracle strategy.

The algorithms compared by their complexity and their information Current research is focusing on getting the best ever, perhaps even the best denoising results, for ever. We have followed this track and have completely disregarded the complexity issue in this comparison. For example, the "shotgun" patch methods are not reproducible in acceptable time. Yet, "all is fair in love and war". The question of how to get the best acceptable results must be solved first, by every possible means, before fast algorithms are devised. On the other hand, the complexity does not seem to be a serious obstacle. Indeed, several of the mentioned algorithms are already realizable, and five of them are even functioning online at Image Processing online `http://www.ipol.im`. Among them, at least two give state of the art results. Thus, we hold the view that complexity is not a central issue in the current debate. Another question which emerged in this study is the *amount of information needed to achieve optimal denoising.* Here, we have observed that the methods do the splits. The simplest one (DCT denoising) uses only one image patch and get results only 1dB far away from optimal results. The classic nonlocal methods only use a larger neighborhood of a given pixel, in spite of their "nonlocal" epithet. Then, an intermediate class of methods uses simultaneously all image patches. The shotgun methods use virtually all existing image patches in the world. The fact that the performance gap between them is so small seems to indicate that all obtain a decent estimate of the "patch space" around each given image patch. This also means that, arguably, there is enough information for that in just one image.

7.2 Conclusion

There seems to be currently only one image denoising paradigm, which generalizes and unites the transform thresholding method with a Markovian Bayesian estimation theory. This unification is complete when the patch space is assumed to be a Gaussian mixture, each Gaussian implicitly giving a different adapted patch orthonormal basis.

This method is almost optimal, and denoises satisfactorily images for an interval of standard deviations of 5 to 40. (These figures are valid for current image formats, with range in $[0, 255]$). Thin noises (below 5) and large noises (above 50) are largely unexplored. They may require new tools or a different theory. Are they important? The answer is yes, because for several applications, for example photogrammetry and stereovision, the precision varies like the inverse of the SNR. Thus, even with good quality stereo pairs, it is relevant to decrease the noise level. As for large noises, it may be argued that the only thing that really matters is the second order statistics of natural images, and one can obtain near optimal denosing by a global Wiener filter. But all existing filters leave behind too many artifacts and must be reconsidered for high noise.

The multiscale aspect of denoising is explored only on three dyadic scales (since most patch methods use 8×8 patches), which may be insufficient. The success of denoising methods is only one step forward in the statistical exploration of images, and in particular in the exploration of the huge "patch space". Its structure remains widely unknown, and we ignore its geometry. There is little doubt that it is not just a sum of Gaussians, or a Gaussian scale mixture.

Last but not least, are image denoising algorithms close to achieve their optimal bounds? In our opinion, on the ranges of noise that we have tested, the image visual improvement obtained by state of the art denoising methods is undeniable. It is even spectacular. On movies, which are much more redundant, this effect is still more impressive. Nevertheless, can we take the arguments developed in [92] and [32] and conclude that the current methods are almost optimal? The arguments given in favor of this view in [92] are very interesting, because they give a method to estimate the optimal bounds for *all* patch-based methods. Nevertheless, a closer examination shows that the existing methods are probably farther away from optimality than explained in this paper. Indeed, all state of the art patch-based methods use the aggregation step which doubles the size of the neighborhood effectively used in the estimation. It follows that their comparison to the shotgun Bayesian estimate using only the knowledge of each given patch to denoise it, is unfair to shotgun NL-means. A fair comparison would be obtained by applying a shotgun NL-means to larger patches, namely 16×16. The question is whether this is possible, or if we face a dimensionality curse.

Acknowledgements The authors wish to thank Richard Baraniuk, Alessandro Foi, Vladimir Katkovnik, Peyman Milanfar, Boaz Nadler, Boshra Rajaei, Guillermo Sapiro, Eero Simoncelli, Yair Weiss and Guoshen Yu for valuable comments, which have been systematically integrated in the text. Research partially financed by the MISS project of Centre National d'Etudes Spatiales, the Office of Naval research under grant N00014-97-1-0839 and by the European Research Council, advanced grant "Twelve labours".

References

[1] A. Adams, N. Gelfand, J. Dolson, and M. Levoy. Gaussian kd-trees for fast high-dimensional filtering. In *ACM Transactions on Graphics (TOG)*, volume 28, page 21. ACM, 2009.

[2] M. Aharon, Michael Elad, and A. Bruckstein. K-SVD: Design of dictionaries for sparse representation. *IEEE Transactions on Image Processing*, pages 9–12, 2005.

[3] F. J. Anscombe. The transformation of Poisson, binomial and negative-binomial data. *Biometrika*, 35(3):246–254, 1948.

[4] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies. Image coding using wavelet transform. *Image Processing, IEEE Transactions on*, 1(2):205–220, 1992.

[5] P. Arias, V. Caselles, and G. Sapiro. A variational framework for non-local image inpainting. *Proc. of EMMCVPR. Springer, Heidelberg*, 2009.

[6] F. Attneave. Some informational aspects of visual perception. *Psychological review*, 61(3):183–193, 1954.

[7] S.P. Awate and R.T. Whitaker. Unsupervised, information-theoretic, adaptive image filtering for image restoration. *IEEE Trans. PAMI*, 28(3):364–376, 2006.

[8] C. Barillot and F. Rennes. An optimized blockwise non local means denoising filter for 3D magnetic resonance images. *Transactions on Medical Imaging*, page 18, 2007.

[9] C. Barnes, E. Shechtman, A. Finkelstein, and D.B. Goldman. Patchmatch: a randomized correspondence algorithm for structural image editing. In *ACM Transactions on Graphics (TOG)*, volume 28, page 24. ACM, 2009.

[10] R.C. Bilcu and M. Vehvilainen. Combined Non-Local averaging and intersection of confidence intervals for image denoising. In *15th IEEE Intern. Conf. on Image Processing*, pages 1736–1739, 2008.

[11] J. Boulanger, J.B. Sibarita, C. Kervrann, and P. Bouthemy. Non-parametric regression for patch-based fluorescence microscopy image sequence denoising. In *5th IEEE Int. Symp. on Biomedical Imaging: From Nano to Macro, 2008. ISBI 2008*, pages 748–751, 2008.

[12] R. Bracho and AC Sanderson. Segmentation of images based on intensity gradient information. In *Proc. IEEE Computer Soc. Conf. on Computer Vision and Pattern Recognition*, pages 19–23, 1985.

[13] Pierre Brémaud. *Markov chains: Gibbs fields, Monte Carlo simulation, and queues*, volume 31. springer verlag, 1999.

[14] X. Bresson and TF Chan. Non-local unsupervised variational image segmentation models. *UCLA CAM Report*, pages 08–67, 2008.

[15] A. Buades. *Image and film denoising by non-local means*. PhD thesis, 2006.

[16] A. Buades, A. Chien, JM Morel, and S. Osher. Topology preserving linear filtering applied to medical imaging. *SIAM Journal on Imaging Science*, 1(1):26–50, 2008.

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

[18] A. Buades, B. Coll, and JM Morel. Image data process by image noise reduction and camera integrating the means for implementing this process. *French Patent 0404837*.

[19] A. Buades, B. Coll, and J.M. Morel. A non local algorithm for image denoising. *IEEE Computer Vision and Pattern Recognition*, 2:60–65, 2005.

[20] A. Buades, B. Coll, and J.M. Morel. Image enhancement by non-local reverse heat equation. *Preprint CMLA*, 22, 2006.

[21] A. Buades, B. Coll, and J.M. Morel. Neighborhood filters and PDE's. *Numerische Mathematik*, 105(1):1–34, 2006.

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

[23] A. Buades, B. Coll, and J.M. Morel. Nonlocal image and movie denoising. *International Journal of Computer Vision*, 76(2):123–139, 2008.

[24] A. Buades, B. Coll, J.M. Morel, et al. A review of image denoising algorithms, with a new one. *Multiscale Modeling and Simulation*, 4(2):490–530, 2006.

[25] A. Buades, B. Coll, JM Morel, and C. Sbert. Self-Similarity Driven Color Demosaicking. *IEEE Transactions on Image Processing*, 18(6):1192–1202, 2009.

[26] A. Buades, M. Colom, and J.M. Morel. Multiscale signal dependent noise estimation. *Image Processing On Line (http:www.ipol.im)*.

[27] A. Buades, M. Lebrun, and J.M. Morel. Implementation of the "non-local bayes" image denoising algorithm. *Image Processing On Line (http:www.ipol.im.*

[28] A. Buades, Y. Lou, JM Morel, and Z. Tang. A note on multi-image denoising. In *Local and Non-Local Approximation in Image Processing, 2009. LNLA 2009. International Workshop on*, pages 1–15. IEEE, 2009.

[29] E.J. Candès and M.B. Wakin. An introduction to compressive sampling. *Signal Processing Magazine, IEEE*, 25(2):21–30, 2008.

[30] M.A. Carreira-Perpinan. Mode-finding for mixtures of gaussian distributions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(11):1318–1323, 2000.

[31] Joseph Salmon Charles-Alban Deledalle and Arnak Dalalyan. Image denoising with patch based pca: local versus global. In *Proceedings of the British Machine Vision Conference*, pages 25.1–25.10. BMVA Press, 2011. http://dx.doi.org/10.5244/C.25.25.

[32] P. Chatterjee and P. Milanfar. Is denoising dead? *Image Processing, IEEE Transactions on*, 19(4):895–911, 2010.

[33] P. Chatterjee and P. Milanfar. Patch-based near-optimal image denoising. *IEEE transactions on image processing: a publication of the IEEE Signal Processing Society*, 2011.

[34] C. Chevalier, G. Roman, and J.N. Niepce. *Guide du photographe.* C. Chevalier, 1854.

[35] A. Cohen, I. Daubechies, and J.C. Feauveau. Biorthogonal bases of compactly supported wavelets. *Communications on pure and applied mathematics*, 45(5):485–560, 1992.

[36] R.R. Coifman and D.L. Donoho. Translation-invariant de-noising. *Lecture Notes In Statistics*, pages 125–125, 1995.

[37] S.F. Cotter, R. Adler, R.D. Rao, and K. Kreutz-Delgado. Forward sequential algorithms for best basis selection. In *Vision, Image and Signal Processing, IEE Proceedings*, volume 146, pages 235–244, 1999.

[38] P. Coupe, P. Yger, and C. Barillot. Fast non local means denoising for 3D MRI images. *MICCAI (2)*, pages 33–40.

[39] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3D transform-domain collaborative filtering. *IEEE Transactions on image processing*, 16:2007, 2007.

[40] K. Dabov, A. Foi, V. Katkovnik, K. Egiazarian, et al. Bm3d image denoising with shape-adaptive principal component analysis. *Proc. of the Workshop on Signal Processing with Adaptive Sparse Structured Representations, Saint-Malo, France*, April 2009.

[41] A. Danielyan, A. Foi, V. Katkovnik, and K. Egiazarian. Image And Video Super-Resolution Via Spatially Adaptive Block-Matching Filtering. In *Proceedings of International Workshop on Local and Non-Local Approximation in Image Processing*, 2008.

[42] A. Danielyan, A. Foi, V. Katkovnik, and K. Egiazarian. Denoising of multispectral images via nonlocal groupwise spectrum-pca. *Proc. of The fifth European Conference on Colour in Graphics, Imaging, and Vision and of the 12th International Symposium on Multispectral Colour Science held at University of Eastern Finland, Joensuu, Finland*, June 2010.

[43] A. Danielyan, V. Katkovnik, and K. Egiazarian. Bm3d frames and variational image deblurring. *IEEE Transactions on Image Processing*, 2012.

[44] Aram Danielyan and A. Foi. Noise variance estimation in nonlocal transform domain. In *Proceedings of International Workshop on Local and Non-Local Approximation in Image Processing, LNLA 2009*.

[45] J. Darbon, A. Cunha, T.F. Chan, S. Osher, and G.J. Jensen. Fast nonlocal filtering applied to electron cryomicroscopy. In *5th IEEE Int. Symp. on Biomedical Imaging: From Nano to Macro*, pages 1331–1334, 2008.

[46] J.S. De Bonet. Noise reduction through detection of signal redundancy. *Rethinking artificial intelligence*, 1997.

[47] C.A. Deledalle, L. Denis, and F. Tupin. Nl-insar: Nonlocal interferogram estimation. *Geoscience and Remote Sensing, IEEE Transactions on*, 49(4):1441–1452, 2011.

[48] C.A. Deledalle, V. Duval, and J. Salmon. Non-local methods with shape-adaptive patches (nlm-sap). *Journal of Mathematical Imaging and Vision*, pages 1–18, 2010.

[49] C.A. Deledalle, F. Tupin, and L. Denis. Poisson nl means: Unsupervised non local means for poisson noise. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pages 801–804. IEEE, 2010.

[50] C.A. Deledalle, F. Tupin, and L. Denis. Polarimetric sar estimation based on non-local means. In *Geoscience and Remote Sensing Symposium (IGARSS), 2010 IEEE International*, pages 2515–2518. IEEE, 2010.

[51] J. Delon and A. Desolneux. Flicker stabilization in image sequences. *hal.archives-ouvertes.fr*, 2009.

[52] D.L. Donoho. De-noising by soft-thresholding. *IEEE Transactions on Information Theory*, 41(3):613–627, 1995.

[53] D.L. Donoho and I.M. Johnstone. Adapting to unknown smoothness via wavelet shrinkage. *Journal of the american statistical association*, pages 1200–1224, 1995.

[54] D.L. Donoho and J.M. JOHNSTONE. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455, 1994.

[55] S. Durand and M. Nikolova. Restoration of wavelet coefficients by minimizing a specially designed objective function. In *Proc. IEEE Workshop on Variational, Geometric and Level Set Methods in Computer Vision*, pages 145–152, 2003.

[56] V. Duval, J.F. Aujol, and Y. Gousseau. A bias-variance approach for the nonlocal means. *SIAM Journal on Imaging Sciences*, 4:760, 2011.

[57] M. Ebrahimi and E.R. Vrscay. Solving the Inverse Problem of Image Zooming Using" Self-Examples". *Lecture Notes in Computer Science*, 4633:117, 2007.

[58] M. Ebrahimi and E.R. Vrscay. Examining the role of scale in the context of the non-local-means filter. *Lecture Notes in Computer Science*, 5112:170–181, 2008.

[59] M. Ebrahimi and E.R. Vrscay. Multi-frame super-resolution with no explicit motion estimation. In *Proceedings of the 2008 International Conference on Image Processing, Computer Vision, and Pattern Recognition*, 2008.

[60] A. Efros and T. Leung. Texture synthesis by non parametric sampling. In *Proc. Int. Conf. Computer Vision*, volume 2, pages 1033–1038, 1999.

[61] A.A. Efros and T.K. Leung. Texture synthesis by non-parametric sampling. In *International Conference on Computer Vision*, volume 2, pages 1033–1038. Corful, Greece, 1999.

[62] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, 15(12):3736–3745, 2006.

[63] M. Elad and D. Datsenko. Example-based regularization deployed to super-resolution reconstruction of a single image. *The Computer Journal*, 2007.

[64] A. Elmoataz, O. Lezoray, and S. Bougleux. Nonlocal discrete regularization on weighted graphs: a framework for image and manifold processing. *IEEE Transactions on Image Processing*, 17(7):1047–1060, 2008.

[65] A. Elmoataz, O. Lezoray, S. Bougleux, and V.T. Ta. Unifying local and nonlocal processing with partial difference operators on weighted graphs. In *International Workshop on Local and Non-Local Approximation in Image Processing*, 2008.

[66] A. Foi. Noise estimation and removal in mr imaging: The variance-stabilization approach. In *Biomedical Imaging: From Nano to Macro, 2011 IEEE International Symposium on*, pages 1809–1814. IEEE, 2011.

[67] A. Foi, S. Alenius, V. Katkovnik, and K. Egiazarian. Noise measurement for raw-data of digital imaging sensors by automatic segmentation of non-uniform targets. *IEEE Sensors Journal*, 7(10):1456–1461, 2007.

[68] Alessandro Foi, Mejdi Trimeche, Vladimir Katkovnik, and Karen Egiazarian. Practical poissonian-gaussian noise modeling and fitting for single-image raw-data. *IEEE TRANSACTIONS ON IMAGE PROCESSING*, (10):1737–1754, 2008.

[69] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions and the bayesian restoration of images. *IEEE Pat. Anal. Mach. Intell.*, 6:721–741, 1984.

[70] G. Gilboa and S. Osher. Nonlocal linear image regularization and supervised segmentation. *Multiscale Modeling and Simulation*, 6(2):595–630, 2008.

[71] O.G. Guleryuz. Weighted averaging for denoising with overcomplete dictionaries. *Image Processing, IEEE Transactions on*, 16(12):3020–3034, 2007.

[72] Guillermo Sapiro Guoshen Yu. DCT image denoising: a simple and effective image denoising algorithm. *Image Processing On Line)*, 2011.

[73] S.R.J.L. HARRIS. Image evaluation and restoration. *JOSA*, 56(5):569–570, 1966.

[74] J. Hays and A.A. Efros. Scene completion using millions of photographs. In *ACM Transactions on Graphics (TOG)*, volume 26, page 4. ACM, 2007.

[75] Y.S. Heo, K.M. Lee, and S.U. Lee. Simultaneous depth reconstruction and restoration of noisy stereo images using non-local pixel distribution. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[76] Y. Hou, C. Zhao, D. Yang, and Y. Cheng. Comments on image denoising by sparse 3-d transform-domain collaborative filtering. *Image Processing, IEEE Transactions on*, 20(1):268–270, 2011.

[77] J. Immerkaer. Fast noise variance estimation. *Computer Vision and Image Understanding*, 64(2):300–302, 1996.

[78] M. Jung and LA Vese. Nonlocal variational image deblurring models in the presence of Gaussian or impulse noise, 2009.

[79] V. Katkovnik, A. Danielyan, and K. Egiazarian. Decoupled inverse and denoising for image deblurring: variational BM3D-frame technique. In *Proceedings of IEEE International Conference on Image Processing (ICIP, 2011)*, 2011.

[80] V.I.A. Katkovnik, V. Katkovnik, K. Egiazarian, and J. Astola. *Local approximation techniques in signal and image processing*. Society of Photo Optical.

[81] S.M. Kay. Fundamentals of statistical signal processing: Estimation theory, 1993.

[82] C. Kervrann and J. Boulanger. Local Adaptivity to Variable Smoothness for Exemplar-Based Image Regularization and Representation. *International Journal of Computer Vision*, 79(1):45–69, 2008.

[83] C. Kervrann, J. Boulanger, and P. Coupe. Bayesian non-local means filter, image redundancy and adaptive dictionaries for noise removal. *Lecture Notes In Computer Science*, 4485:520, 2007.

[84] S. Kindermann, S. Osher, and P.W. Jones. Deblurring and denoising of images by nonlocal functionals. *Multiscale Modeling and Simulation*, 4(4):1091–1115, 2006.

[85] E. D. Kolaczyk. Wavelet shrinkage estimation of certain Poisson intensity signals using corrected thresholds. *Statist. Sin.*, 9:119–135, 1999.

[86] M. Lebrun, A. Buades, and J.M. Morel. Study and analysis of NL-PCA. *Image Processing on Line*. ipol.im . Workshop, 2011.

[87] A.B. Lee, K.S. Pedersen, and D. Mumford. The nonlinear statistics of high-contrast patches in natural images. *International Journal of Computer Vision*, 54(1):83–103, 2003.

[88] J. S Lee and K. Hoppel. Noise modelling and estimation of remotely-sensed images. *Proceedings of the International Geoscience and Remote Sensing Symposium*, vol. 2, pp. 10051008., 1989.

[89] J.S. Lee. Refined filtering of image noise using local statistics. *Computer graphics and image processing*, 15(4):380–389, 1981.

[90] J.S. Lee. Digital image smoothing and the sigma filter. *Computer Vision, Graphics, and Image Processing*, 24(2):255–269, 1983.

[91] Stamatios Lefkimmiatis, Petros Maragos, and George Papandreou. Bayesian inference on multiscale models for poisson intensity estimation: Application to photo-limited image denoising. *IEEE Transactions on Image Processing*, 18(8):1724–1741, 2009.

[92] A. Levin and B. Nadler. Natural image denoising: Optimality and inherent bounds. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2833–2840. IEEE, 2011.

[93] O. Lezoray, V.T. Ta, and A. Elmoataz. Nonlocal graph regularization for image colorization. *International Conference on Pattern Recognition*, 2008.

[94] C. Liu, W.T. Freeman, R. Szeliski, and S.B. Kang. Noise estimation from a single image. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 901–908. Ieee, 2006.

[95] C. Liu, R. Szeliski, S.B. Kang, C.L. Zitnick, and W.T. Freeman. Automatic estimation and removal of noise from a single image. *IEEE transactions on pattern analysis and machine intelligence*, 30(2):299–314, 2008.

[96] Y. Lou, X. Zhang, S. Osher, and A. Bertozzi. Image recovery via nonlocal operators. *UCLA CAM Reports (08-35)*.

[97] S. Lyu and E.P. Simoncelli. Modeling multiscale subbands of photographic images with fields of gaussian scale mixtures. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(4):693–706, 2009.

[98] M. Mahmoudi and G. Sapiro. Fast image and video denoising via nonlocal means of similar neighborhoods. *IEEE Signal Processing Letters*, 12(12):839–842, 2005.

[99] J. Mairal. *Représentations parcimonieuses en apprentissage statistique, traitement d'image et vision par ordinateur*. PhD thesis, 2010.

[100] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Non-local sparse models for image restoration. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 2272–2279. IEEE, 2009.

[101] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Non-local sparse models for image restoration. In *ICCV'09*, pages 2272–2279, 2009.

[102] J. Mairal, M. Elad, and G. Sapiro. Sparse representation for color image restoration. *Image Processing, IEEE Transactions on*, 17(1):53–69, 2008.

[103] J. Mairal, M. Elad, and G. Sapiro. Sparse representation for color image restoration. *IEEE Transactions on image processing*, 17(1):53–69, 2008.

[104] M. Makitalo and A. Foi. Optimal inversion of the Anscombe transformation in low-count Poisson image denoising. *Image Processing, IEEE Transactions on*, 20(1):99–109, 2011.

[105] A. Maleki, M. Narayan, and R.G. Baraniuk. Anisotropic nonlocal means denoising. *Arxiv preprint arXiv:1112.0311*, 2011.

[106] A. Maleki, M. Narayan, and R.G. Baraniuk. Suboptimality of nonlocal means for images with sharp edges. *Arxiv preprint arXiv:1111.5867*, 2011.

[107] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic press, 1999.

[108] E. Mammen and A.B. Tsybakov. Asymptotical minimax recovery of sets with smooth boundaries. *The Annals of Statistics*, pages 502–524, 1995.

[109] J.V. Manjón, J. Carbonell-Caballero, J.J. Lull, G. García-Martí, L. Martí-Bonmatí, and M. Robles. MRI denoising using Non-Local Means. *Medical Image Analysis*, 12(4):514–523, 2008.

[110] J.V. Manjón, M. Robles, and N.A. Thacker. Multispectral MRI Denoising Using Non-Local Means. In *Proc. MIUA*, volume 7, pages 41–45.

[111] G.A. Mastin. Adaptive filters for digital image noise smoothing: An evaluation*. *Computer Vision, Graphics, and Image Processing*, 31(1):103–121, 1985.

[112] G.S. Mayer, E.R. Vrscay, M.L. Lauzon, B.C. Goodyear, and J.R. Mitchell. Self-similarity of Images in the Fourier Domain, with Applications to MRI. *Lecture Notes in Computer Science*, 5112:43–52, 2008.

[113] P. Meer, J.M. Jolion, and A. Rosenfeld. A fast parallel algorithm for blind estimation of noise variance. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 12(2):216–223, 1990.

[114] Y. Meyer. Wavelets-algorithms and applications. *Wavelets-Algorithms and applications Society for Industrial and Applied Mathematics Translation., 142 p.*, 1, 1993.

[115] M. Mignotte. A non-local regularization strategy for image deconvolution. *Pattern Recognition Letters*, 29(16):2206–2212, 2008.

[116] P. Milanfar. A tour of modern image filtering. *Invited feature article to IEEE Signal Processing Magazine (preprint at http://users. soe. ucsc. edu/ milanfar/publications/)*, 2011.

[117] B. Naegel, A. Cernicanu, J.N. Hyacinthe, M. Tognolini, and J.P. Vallée. SNR enhancement of highly-accelerated real-time cardiac MRI acquisitions based on non-local means algorithm. *Medical Image Analysis*, 2009.

[118] A. Nemirovski. Topics in non-parametric statistics. *Lectures on probability theory and statistics (Saint-Flour, 1998)*, 1738:85–277, 2000.

[119] Robert D. Nowak and Richard G. Baraniuk. Wavelet-domain filtering for photon imaging systems. *IEEE Transactions on Image Processing*, 8(5):666–678, 1997.

[120] S.I. Olsen. Estimation of noise in images: An evaluation. *CVGIP: Graphical Models and Image Processing*, 55(4):319–323, 1993.

[121] J. Orchard, M. Ebrahimi, and A. Wong. Efficient Non-Local-Means Denoising using the SVD. In *Proceedings of The IEEE International Conference on Image Processing*, 2008.

[122] E. Ordentlich, G. Seroussi, S. Verdu, M. Weinberger, and T. Weissman. A discrete universal denoiser and its application to binary images. In *International Conference on Image Processing*, volume 1, 2003.

[123] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin. Using geometry and iterated refinement for inverse problems (1): Total variation based image restoration. *Department of Mathematics, UCLA, LA, CA*, 90095:04–13, 2004.

[124] E. Le Pennec and S. Mallat. Geometrical image compression with bandelets. In *Proceedings of the SPIE 2003*, volume 5150, pages 1273–1286, 2003.

[125] G. Peyré. Manifold models for signals and images. *Computer Vision and Image Understanding*, 113(2):249–260, 2009.

[126] N. N. Ponomarenko, V. V. Lukin, M. S. Zriakhov, A. Kaarna, and J. T. Astola. An automatic approach to lossy compression of AVIRIS images. *IEEE International Geoscience and Remote Sensing Symposium*, 2007.

[127] N.N. Ponomarenko, V.V. Lukin, S.K. Abramov, K.O. Egiazarian, and J.T. Astola. Blind evaluation of additive noise variance in textured images by nonlinear processing of block dct coefficients. In *Proceedings of SPIE*, volume 5014, page 178, 2003.

[128] J. Portilla, V. Strela, M.J. Wainwright, and E.P. Simoncelli. Image denoising using scale mixtures of gaussians in the wavelet domain. *Image Processing, IEEE Transactions on*, 12(11):1338–1351, 2003.

[129] M. Protter, M. Elad, H. Takeda, and P. Milanfar. Generalizing the non-local-means to super-resolution reconstruction. *IEEE Transactions on Image Processing*, 2008.

[130] K. Rank, M. Lendl, and R. Unbehauen. Estimation of image noise variance. In *Vision, Image and Signal Processing, IEE Proceedings-*, volume 146, pages 80–84. IET, 1999.

[131] M. Raphan and E.P. Simoncelli. Learning to be bayesian without supervision. *Advances in neural information processing systems*, 19:1145, 2007.

[132] M. Raphan and E.P. Simoncelli. An empirical bayesian interpretation and generalization of nl-means. Technical report, Technical Report TR2010-934, Computer Science Technical Report, Courant Inst. of Mathematical Sciences, New York University, 2010.

[133] W.H. Richardson. Bayesian-based iterative method of image restoration. *JOSA*, 62(1):55–59, 1972.

[134] L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60(1-4):259–268, 1992.

[135] B.C. Russell, A. Torralba, K.P. Murphy, and W.T. Freeman. Labelme: a database and web-based tool for image annotation. *International journal of computer vision*, 77(1):157–173, 2008.

[136] C.E. Shannon. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1):3–55, 2001.

[137] A. Singer, Y. Shkolnisky, and B. Nadler. Diffusion interpretation of nonlocal neighborhood filters for signal denoising. *SIAM J. Imaging Sci*, 2(1):118–139, 2009.

[138] S.M. Smith and J.M. Brady. SUSANA new approach to low level image processing. *International Journal of Computer Vision*, 23(1):45–78, 1997.

[139] J.L. Starck, E.J. Candes, and D.L. Donoho. The curvelet transform for image denoising. *Image Processing, IEEE Transactions on*, 11(6):670–684, 2002.

[140] A.D. Szlam, M. Maggioni, and R.R. Coifman. A general framework for adaptive regularization based on diffusion processes on graphs. *Yale technichal report*, 2006.

[141] N.A. Thacker, J.V. Manjon, and P.A. Bromiley. A Statistical Interpretation of Non-Local Means. In *5th International Conference on Visual Information Engineering*, pages 250–255, 2008.

[142] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Computer Vision, 1998. Sixth International Conference on*, pages 839–846, 1998.

[143] J.W. Tukey. Exploratory data analysis. 1977. *Massachusetts: Addison-Wesley*, 1976.

[144] D. Van De Ville and M. Kocher. Sure-based non-local means. *Signal Processing Letters, IEEE*, 16(11):973–976, 2009.

[145] H. Voorhees and T. Poggio. Detecting textons and texture boundaries in natural image. In *Proceedings of the First International Conference on Computer Vision London*, pages 250–258. IEEE, Washington, DC, 1987.

[146] J. Wang, Y. Guo, Y. Ying, Y. Liu, and Q. Peng. Fast non-local algorithm for image denoising. In *2006 IEEE International Conference on Image Processing*, pages 1429–1432, 2006.

[147] T. Weissman, E. Ordentlich, G. Seroussi, S. Verdu, and MJ Weinberger. Universal discrete denoising: Known channel. *IEEE Transactions on Information Theory*, 51(1):5–28, 2005.

[148] N. Wiest-Daesslé, S. Prima, P. Coupé, S.P. Morrissey, and C. Barillot. Non-local means variants for denoising of diffusion-weighted and diffusion tensor MRI. *Lecture Notes in Computer Science*, 4792:344, 2007.

[149] N. Wiest-Daessle, S. Prima, P. Coupé, S.P. Morrissey, and C. Barillot. Rician noise removal by non-local means filtering for low signal-to-noise ratio MRI: Applications to DT-MRI. *Lecture Notes in Computer Science*, 5242:171–179, 2008.

[150] A. Wong and J. Orchard. A nonlocal-means approach to exemplar-based inpainting. In *15th IEEE International Conference on Image Processing, 2008*, pages 2600–2603, 2008.

[151] H. Xu, J. Xu, and F. Wu. On the biased estimation of nonlocal means filter. In *2008 IEEE International Conference on Multimedia and Expo*, pages 1149–1152, 2008.

[152] L. Yaroslavsky and M. Eden. Fundamentals of Digital Optics, 2003.

[153] L. P. Yaroslavsky. *Digital Picture Processing.* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1985.

[154] L.P. Yaroslavsky. Local adaptive image restoration and enhancement with the use of DFT and DCT in a running window. In *Proceedings of SPIE*, volume 2825, page 2, 1996.

[155] G. Yu, G. Sapiro, and S. Mallat. Image modeling and enhancement via structured sparse model selection. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pages 1641–1644. IEEE, 2010.

[156] G. Yu, G. Sapiro, and S. Mallat. Solving Inverse Problems with Piecewise Linear Estimators: From Gaussian Mixture Models to Structured Sparsity. *Arxiv preprint arXiv:1006.3056*, 2010.

[157] L. Zhang, W. Dong, D. Zhang, and G. Shi. Two-stage image denoising by principal component analysis with local pixel grouping. *Pattern Recognition*, 43(4):1531–1549, 2010.

[158] X. Zhang, M. Burger, X. Bresson, and S. Osher. Bregmanized nonlocal regularization for deconvolution and sparse reconstruction. *UCLA CAM Report*, pages 09–03, 2009.

[159] M. Zhou, H. Yang, G. Sapiro, D. Dunson, and L. Carin. Dependent hierarchical beta process for image interpolation and denoising. In *International Conference on Artificial Intelligence and Statistics*, 2011.

[160] S. Zimmer, S. Didas, and J. Weickert. A rotationally invariant block matching strategy improving image denoising with non-local means. In *Proc. 2008 International Workshop on Local and Non-Local Approximation in Image Processing*, 2008.

[161] D. Zoran and Y. Weiss. From learning models of natural image patches to whole image restoration.

[162] D. Zoran and Y. Weiss. Scale invariance and noise in natural images. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 2209–2216. IEEE, 2009.

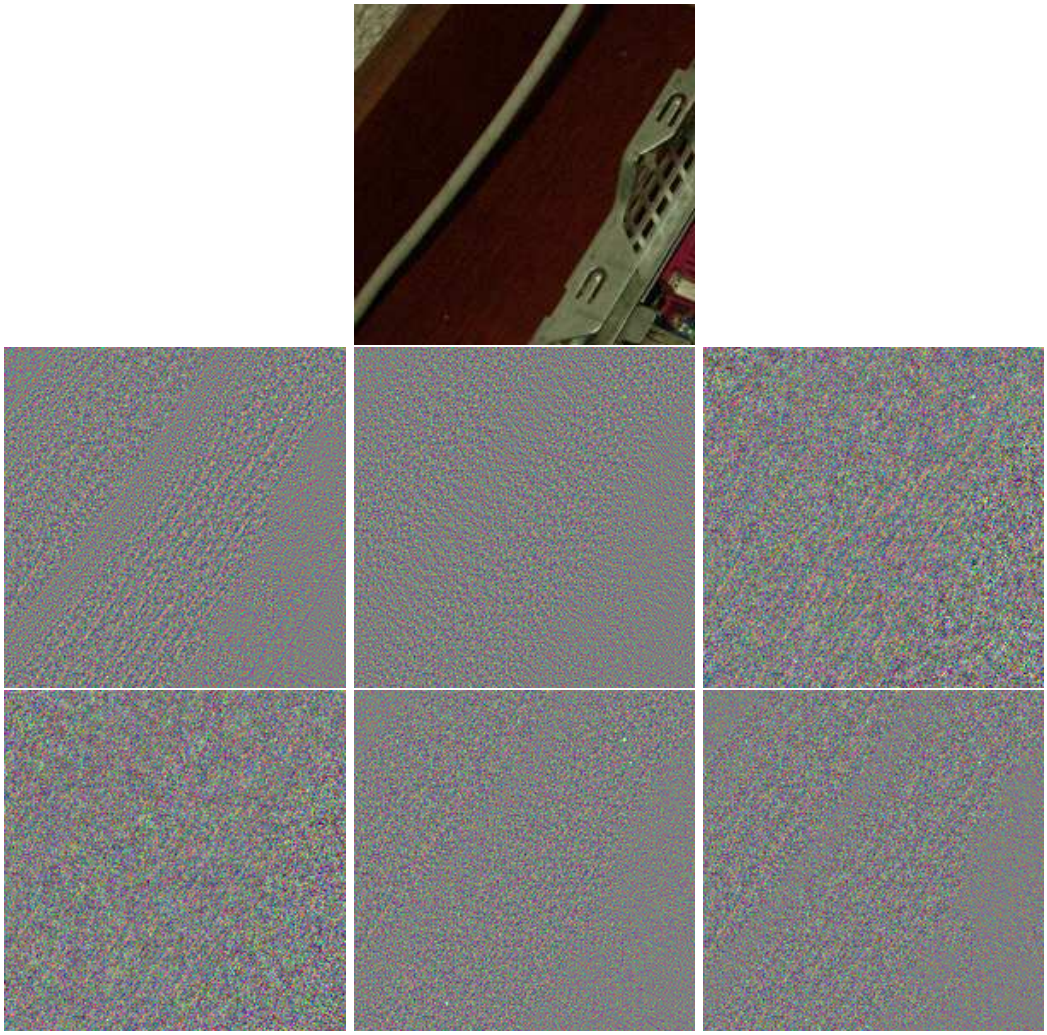


Figure 14: Enlargement of the method noise difference of Fig. 13. From top to bottom and left to right: slightly noisy image, and the method noise for DCT sliding window, BLS-GSM, NL-means, K-SVD, BM3D and Non-local Bayes. The amplitude of the noise removed by NL-means and K-SVD is uniform all over the image while it is region dependent for the rest of the algorithms. Threshold based algorithms prefer to keep noisy values nearly untouched on highly textured or edge zones.

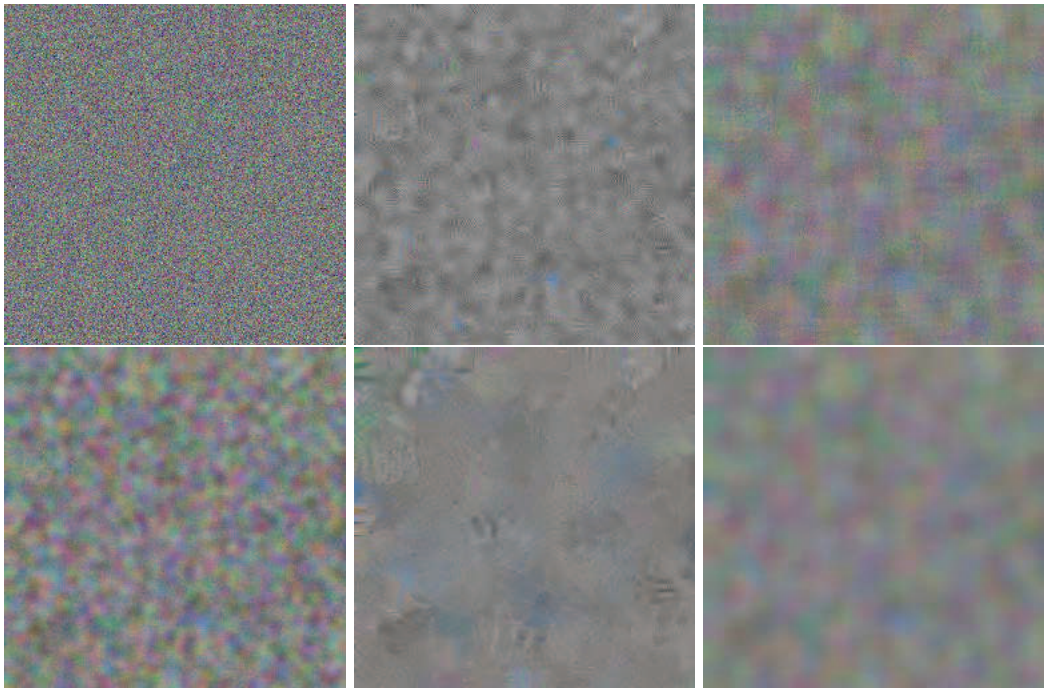


Figure 15: The noise to noise principle: a three-channels colour noise image filtered by the state of the art methods. From top to bottom and left to right: the noise image (flat, with independent homoscedastic noise added on each channel). Then, this same image denoised by DCT sliding window, NL-means, K-SVD, BM3D and Non-local Bayes. The more the denoised image of a noise image looks like a noise image the better. Indeed, structured noise creates artifacts. BSM-GSM was not compared because we lack a colour version for this algorithm. None of the methods gives a satisfactory result: they all create a lower frequency oscillation or local artifacts for DCT and BM3D. Only multiscale version could cope with the low frequency remaining noise.

Figure 16: A set of noiseless images used for the comparison tests.

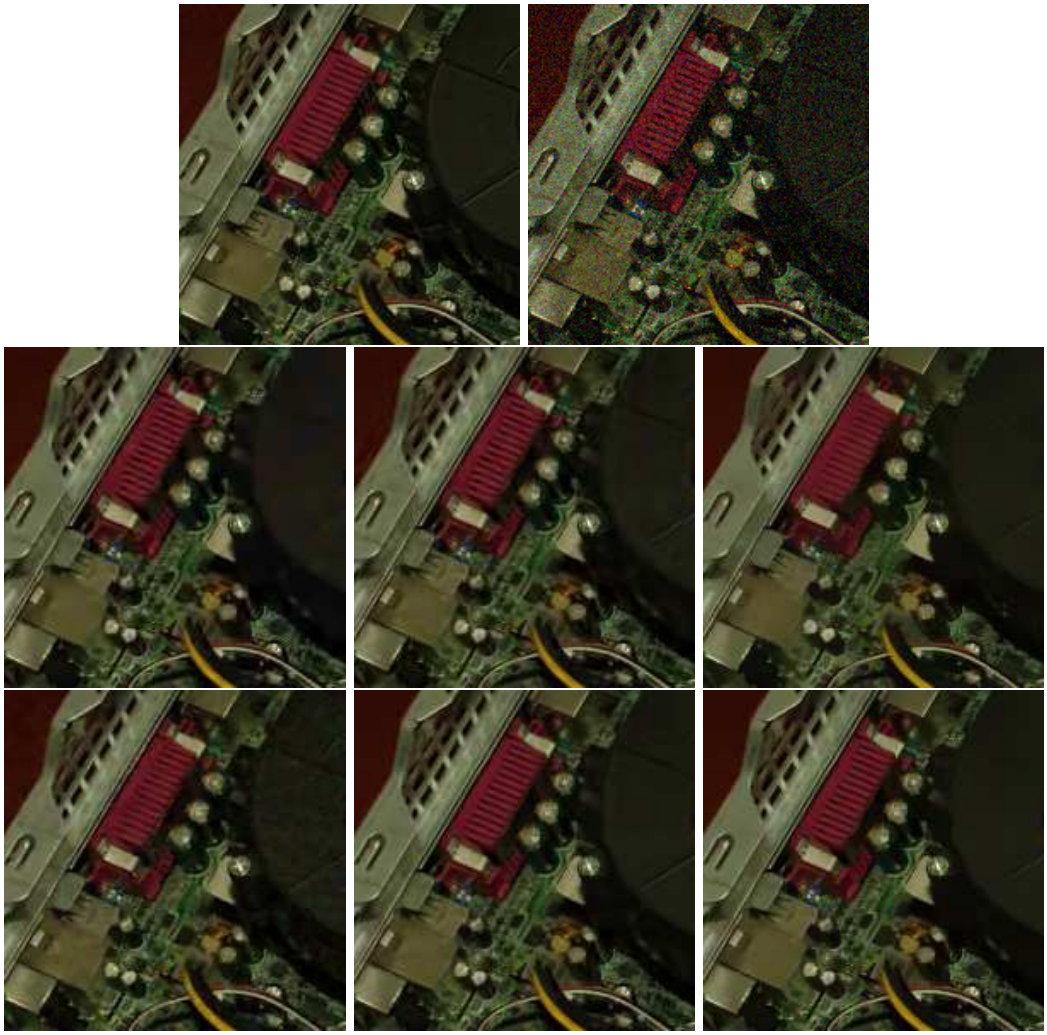


Figure 17: Visual quality comparison. The noisy image was obtained adding a Gaussian white noise of standard deviation 20. From top to bottom and left to right: original, noisy, DCT sliding window, BLS-GSM, NL-means, K-SVD, BM3D and Non-local Bayes.

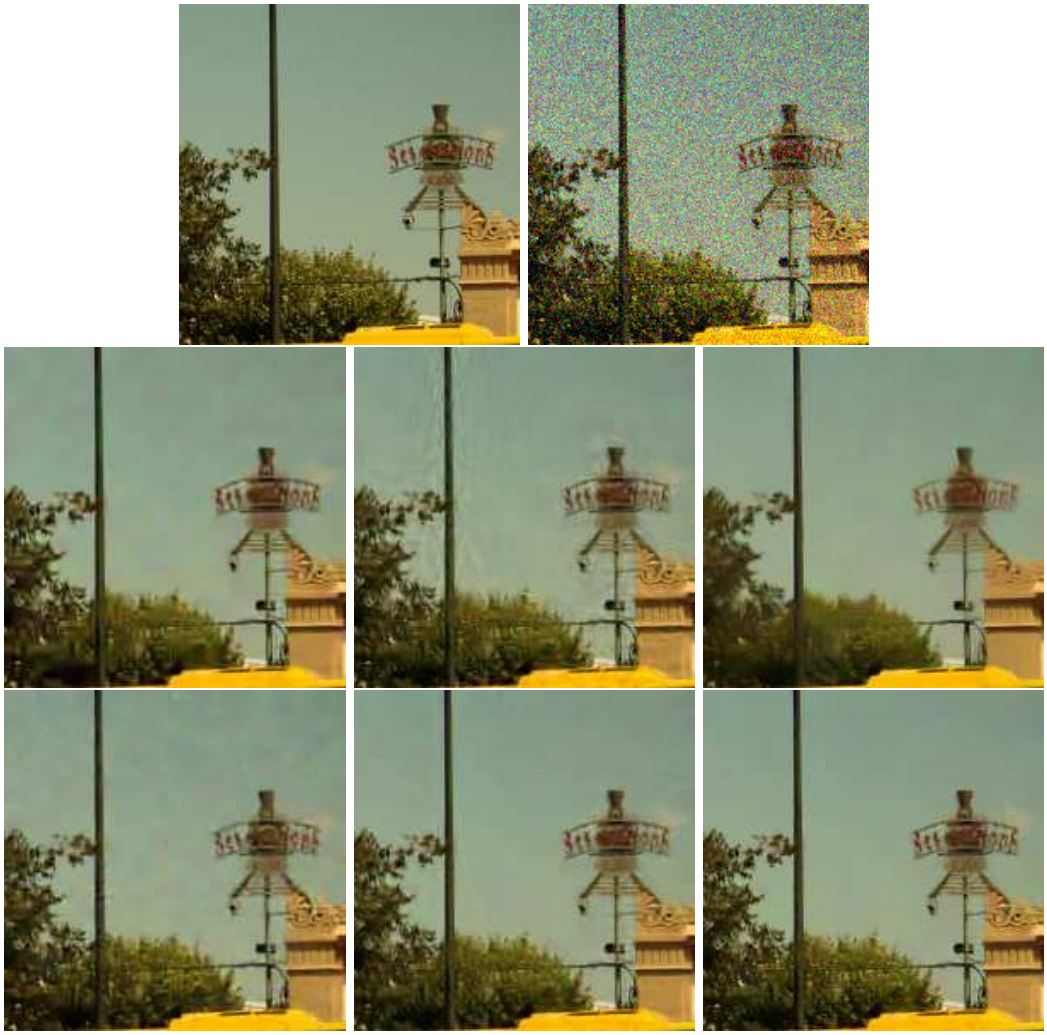


Figure 18: Comparison of visual quality. The noisy image was obtained adding a Gaussian white noise of standard deviation 30. From top to bottom and left to right: original, noisy, DCT sliding window, BLS-GSM, NL-means, K-SVD, BM3D and Non-local Bayes.

Figure 19: Comparison of visual quality. The noisy image was obtained adding a Gaussian white noise of standard deviation 40. From top to bottom and left to right: original, noisy, DCT sliding window, BLS-GSM, NL-means, K-SVD, BM3D and Non-local Bayes.