Acta Numerica (2012), pp. 475–576
doi:10.1017/S0962492912000062
© Cambridge University Press, 2012
Printed in the United Kingdom

Secrets of image denoising cuisine*

M. Lebrun
CMLA, Ecole Normale Supérieure de Cachan,
61 Avenue du Président Wilson,
94235 Cachan CEDEX, France
E-mail: marc.lebrun.ik@gmail.com

M. Colom
Universitat de les Illes Balears,
Crta de Valldemossa, km 7.5,
07122 Palma de Mallorca, Spain
E-mail: Miguel.Colom@cmla.ens-cachan.fr

A. Buades
Universitat de les Illes Balears,
Crta de Valldemossa, km 7.5,
07122 Palma de Mallorca, Spain
and
MAP5, Université Paris Descartes,
45 Rue des Saints Pères,
75270 Paris CEDEX 06, France
E-mail: toni.buades@uib.es

J. M. Morel
CMLA, Ecole Normale Supérieure de Cachan,
61 Avenue du Président Wilson,
94235 Cachan CEDEX, France
E-mail: morel@cmla.ens-cachan.fr

Digital images are matrices of equally spaced pixels, each containing a photon count. This photon count is a stochastic process due to the quantum nature of light. It follows that all images are noisy. Ever since digital images have existed, 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.

* Colour online for monochrome figures available at journals.cambridge.org/anu.

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

Obtaining a convincing statistical image model is quite another story. Images reflect the world and are just as complex. Thus, any progress in image denoising implies 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 optimal performance in image denoising. This suspicion is supported by a remarkable convergence of all analysed methods. They certainly converge in performance. We intend to demonstrate that, under different formalisms, their image models 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, given the noiseless patches.

The story will be told in an ordered fashion. 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 are the Markovian–Bayesian paradigm, linear transform thresholding, so-called image sparsity, and an image self-similarity hypothesis, which will also be presented. 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 us 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 rapidly changing subject, but many of the presented recipes will remain useful. Most denoising recipes can be tested directly on any digital image in the web journal *Image Processing On Line*.

CONTENTS

1 Introduction 478
2 Noise 487
3 Four denoising principles 500
4 Noise reduction: generic tools 511
5 Detailed analysis of nine methods 516
6 Comparison of denoising algorithms 547
7 Synthesis 561
References 567

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

Notation

- $\mathbf{i}$, $\mathbf{j}$, $\mathbf{r}$, $\mathbf{s}$ image pixels
- $u(\mathbf{i})$ image value at $\mathbf{i}$, written as $U(\mathbf{i})$ when the image is handled as a vector
- $\tilde{u}(\mathbf{i})$ noisy image value at $\mathbf{i}$, or $\tilde{U}(\mathbf{i})$ when the image is handled as a vector
- $\hat{u}(\mathbf{i})$ restored image value, or $\hat{U}(\mathbf{i})$ when the image is handled as a vector
- $n(\mathbf{i})$ noise at $\mathbf{i}$
- N patch of noise in vector form
- m number of pixels $\mathbf{j}$ involved in denoising a pixel $\mathbf{i}$
- P reference patch; Q, a second patch compared to P
- $\tilde{P}$, $\tilde{Q}$ noisy patches
- $\hat{P}$ restored patch
- $w(\tilde{P},\tilde{Q}) = \mathrm{e}^{-\frac{d^2(\tilde{P},\tilde{Q})}{C\sigma^2}}$ the 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 search zone for similar patches
- $\mathcal{N}(\mu, \mathbf{C})$ 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)
- $\mathbb{E}Q$ 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)
- $\mathrm{DCT}(n_1, n_2)$ 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_{dict} 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)

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

1. Introduction

Most digital images and movies are currently obtained by a CCD[1] 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 is the sum of thermal noise and electronic noise, which are approximately additive and white. For 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 1.1.

Accumulating photon impacts on a surface is therefore the essence of photography. The first photograph, by Nicéphore Niépce (Chevalier, Roman and Niépce 1854), was obtained from an eight-hour exposure. The problem of long exposure is the variation of the scene due to changes in light, camera motion, and incidental motion of parts of the scene. The more these variations can be compensated for, 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 a short exposure, the image is dark, and enhancing it will reveal noise.

A recently available solution is to take a burst of images, each with a short exposure time, and to average them. This technique, illustrated in Figure 1.1, was evaluated recently in a paper that proposes fusing bursts of images taken by cameras (Buades, Lou, Morel and Tang 2009*b*). This paper shows that the noise reduction by this method is almost perfect: fusing m images reduces the noise by a factor of $\sqrt{m}$.

It is not always possible to accumulate photons. There are obstacles to this accumulation in astronomy, biological imaging and medical imaging. In everyday images the scene moves, and this limits the exposure time. The main limitations to any imaging system are therefore noise and 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 do we 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 also using the rest of the image is obviously an ill-posed problem. Indeed, each pixel receives photons coming from different sources.

[1] Charge coupled device, *i.e.*, the 'digital film' at the heart of a digital camera.

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press



(a) (b) (c)

Figure 1.1. (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 Buades *et al.* (2009*b*).

Nevertheless, one hint for the solution comes from image formation theory. A well-sampled image u is band-limited (Shannon 2001). Thus, it seems possible to restore the band-limited image u from its degraded samples $\tilde{u}$, as proposed by Harris (1966). 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 neighbouring 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 $u(\mathbf{i})$. Since these values are non-deterministic, Bayesian approaches are relevant and were proposed as early as Richardson (1972).

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

The Fourier method has been extended in the past 30 years to other linear space-frequency transforms, such as the windowed discrete cosine transform (Yaroslavsky and Eden 2003) or the many wavelet transforms (Meyer 1993).

Bayesian methods were initially parametric and limited to rather restrictive Markov random field models (Geman and Geman 1984), but are now becoming non-parametric. The idea behind the recent non-parametric Markovian estimation methods is a now famous algorithm to synthesize textures from examples (Efros and Leung 1999). 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 neighbourhood P of $\mathbf{i}$, which we shall call a 'patch'.

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

Algorithm 1 Non-local means (NL-means) algorithm

Input. Noisy image $\tilde{u}$, noise standard deviation σ.
Output. Denoised image $\hat{u}$.
Set parameter $\kappa \times \kappa$: dimension of patches.
Set parameter $\lambda \times \lambda$: dimension of search zone for similar patches.
Set parameter C.
for each pixel **i do**
Select a square reference sub-image (or 'patch') $\tilde{P}$ around **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 neighbourhood of **i** of size $\lambda \times \lambda$. The weights in the average are proportional to

$$w(\tilde{P}, \tilde{Q}) = \mathrm{e}^{-\frac{d^2(\tilde{P},\tilde{Q})}{C\sigma^2}},$$

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 **i** by averaging all values at **i** of all denoised patches $\hat{Q}$ containing **i**.

The assumption when recreating new textures from samples is that there are enough pixels **j** similar to **i** in a texture image $\tilde{u}$ to recreate a new but similar texture u. The construction of u is done by non-parametric sampling, amounting to an iterative copy–paste process. Let us assume that we already know the values of u on a patch P partially surrounding an unknown pixel **i**. The algorithm of Efros and Leung (1999) 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 **j**. Indeed, these values form a histogram approximating the law of $u(\mathbf{i})$. This algorithm goes back to Shannon's theory of communication (Shannon 2001), in which it was used for the first time to synthesize a probabilistically correct text from a sample.

As proposed by Buades, Coll and Morel (2005*b*), 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 centred at **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 **i**. This simple change leads to the 'non-local means' (NL-means) algorithm, which can therefore be sketched in a few lines: see Algorithm 1.

Buades *et al.* (2005*b*) also proved that the algorithm gave the best possible mean square estimate if the image was modelled as an infinite stationary ergodic spatial process (see Section 5.1 for an exact statement). The algo-

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

rithm was called 'non-local' because it used 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 Section 6, the algorithm of Portilla, Strela, Wainwright and Simoncelli (2003), described in Section 5.6, has better PSNR (peak signal-to-noise ratio) performance. But quality criteria show that NL-means creates fewer artifacts than wavelet-based methods. This may explain why patch-based denoising methods have flourished ever since. To date, 1500 papers have been published on non-local 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 packages.

An exciting recent paper in this exploration of non-local methods raises the following claim (Levin and Nadler 2011): *For natural images, the recent patch-based denoising methods might well be close to optimality.* Levin and Nadler (2011) use a set of 20 000 images containing about 10^{10} patches. This paper provides a second answer to the question of absolute limits raised by Chatterjee and Milanfar (2010): 'Is denoising dead?' The Cramer–Rao-type lower bounds on the attainable RMSE (root mean square error) performance given by Chatterjee and Milanfar (2010) are actually more optimistic: they allow for the possibility of a significant increase in denoising performance. The two types of performance bounds considered by Levin and Nadler (2011) and Chatterjee and Milanfar (2010) 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 by Levin and Nadler (2011) is in fact based on NL-means (Algorithm 1), with the adequate parameter C to account for a Bayesian linear minimum mean square estimation (LMMSE) of the noisy patch given a database of known patches. The only important difference is that the similar patches Q are chosen from a database of 10^{10} patches, instead of from the image itself. Furthermore, by a simple mathematical argument and intensive simulations on patch space, Levin and Nadler are able to approach *the best average estimation error* that can *ever be attained by any patch-based denoising algorithm* (see Section 5.4).

These optimal bounds are nonetheless obtained on a somewhat restrictive definition of patch-based methods. A patch-based algorithm is understood to be an algorithm that denoises each pixel by using knowledge of (a) the patch surrounding it, and (b) the probability density of all existing patches. 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 of denoising all patches and then *aggregating* the estimate of all patches containing a given pixel

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

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 (Dabov, Foi, Katkovnik and Egiazarian 2007), might arguably be very close to the best predicted estimation error is enlightening. Furthermore, doubling the size of the patch used in the paper by Levin and Nadler (2011) would be enough to cover the aggregation step. The difficulty is getting 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 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 largely unexplored territory. Therefore a complete theory or a single definitive algorithm 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 very few principles. More precisely, this paper will present the following recipes, and compare them whenever possible.

- Several families of noise estimation techniques (Section 2).
- The four denoising principles in competition (Section 3).
- Three techniques that improve every denoising method (Section 4).
- Nine complete and recent denoising algorithms. For these algorithms complete recipes will be given (Section 5).
- Three complementary and simple recipes to evaluate and compare denoising algorithms (Section 6).

Using the three comparison recipes, six emblematic or state-of-the-art algorithms, based on reliable and public implementations, will be compared in Section 6. This comparison is followed by a synthesis (Section 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, since no researcher likes finished problems.) With one exception (the BLS-GSM algorithm, Section 5.6), state-of-the-art denoising algorithms are not multiscale. High noise and low noise also remain unexplored.

In a broader perspective, the success of image denoising marks the discovery and exploration of one of the first ever densely sampled high-dimensional probability laws (numerically) accessible to mankind: the 'patch space'. For

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

8×8 patches, by applying a local principal component analysis (PCA) to the patches surrounding a given patch, one can deduce that this space has a dozen significant dimensions (other dimensions corresponding to small eigenvalues). Exploring its structure, as was initiated in Lee, Pedersen and Mumford (2003), seems to be the first step towards 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 discussed here will be available in the journal *Image Processing On Line*, http://www.ipol.im/. In this web journal, each algorithm is given a complete description along with the corresponding source code, and can be run online on arbitrary images. By the time this paper is published, most results and techniques presented here will be effortlessly verifiable and reproducible online.

This Introduction ends with a quick review of many recent contributions of interest on patch-based methods, which nevertheless fall beyond our limited scope.

1.1. Miscellaneous 'patch-based' considerations and applications

Statistical validity

This paper will compare patch-based algorithms according to their structure and their practical performance, which is acceptable in the 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 by Thacker, Manjon and Bromiley (2008), Kervrann, Boulanger and Coupé (2007) and Ebrahimi and Vrscay (2007), who propose a Bayesian interpretation, and by Xu, Xu and Wu (2008), who correct the bias of NL-means. Singer, Shkolnisky and Nadler (2009) give 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 (Maleki, Narayan and Baraniuk 2011*b*), is suitable for 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

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

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 proved (Mammen and Tsybakov 1995) that, for $\alpha \geq 1$,

$$R_m(H^\alpha(C)) \simeq m^{-\frac{2\alpha}{\alpha+1}}. \tag{1.1}$$

For example, for $\alpha = 2$, which corresponds to edges with bounded curvature, the optimal rate is $n^{-\frac{4}{3}}$. This result gives a kind 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. Maleki *et al.* (2011*b*) show that the decay rate is about m^{-1}, close to that obtained with wavelet threshold denoising, better than rates of elementary filters such as linear convolution, the median filter and the bilateral filter, which have rates $m^{-\frac{2}{3}}$. The decay rate of NL-means is nonetheless some distance from the optimal minimax rate of $m^{-4/3}$, which is only attained for $\alpha = 2$ by the wedgelet transform. Maleki, Narayan and Baraniuk (2011*a*) show that an anisotropic non-local 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 improve 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. Therefore, 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 (Darbon *et al.* 2008), fluorescence microscopy (Boulanger, Sibarita, Kervrann and Bouthemy 2008), magnetic resonance imaging (MRI) (Manjón *et al.* 2008, Coupé *et al.* 2008, Wiest-Daesslé *et al.* 2008, Naegel *et al.* 2009), multispectral MRI (Manjón, Robles and Thacker 2007, Buades, Chien, Morel and Osher 2008*a*), and diffusion tensor MRI (DT-MRI) (Wiest-Daesslé *et al.* 2007).

More invariance

Likewise, several papers have explored the degree of invariance that could be applied to image patches. Zimmer, Didas and Weickert (2008) explore a rotationally invariant block matching strategy improving NL-means, and

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

Ebrahimi and Vrscay (2008*a*) use cross-scale (*i.e.*, down-sampled) neighbourhoods in the NL-means filter. See also the paper by Maleki *et al.* (2011*a*), mentioned above as reaching better minimax limits, for which it uses oriented anisotropic patches. Self-similarity has also been explored in the Fourier domain for MRI, by Mayer *et al.* (2008).

Fast patch methods

Several papers have proposed fast and extremely fast (linear) NL-means implementations, using block pre-selection (Mahmoudi and Sapiro 2005, Bilcu and Vehvilainen 2008), Gaussian KD-trees, to classify image patches (Adams, Gelfand, Dolson and Levoy 2009), singular value decomposition (Orchard, Ebrahimi and Wong 2008), the fast Fourier transform, to compute correlation between patches (Wang *et al.* 2006), statistical arguments (Coupé, Yger and Barillot 2006), and approximate search (Barnes, Shechtman, Finkelstein and Goldman 2009), also used for optical flow.

Other image processing tasks

The non-local denoising principle has also been expanded to most image processing tasks: *demosaicing*, the operation which transforms the 'R or G or B' raw image in each camera into an 'R and G and B' image (Buades, Coll, Morel and Sbert 2009*a*, Mairal, Elad and Sapiro 2008); *movie colourization* (Elmoataz, Lézoray, Bougleux and Ta 2008*b*, Lézoray, Ta and Elmoataz 2008); *image inpainting*, by proposing a non-local image inpainting variational framework with a unified treatment of geometry and texture (Arias, Caselles and Sapiro 2009; see also Wong and Orchard 2008); *zooming*, by a fractal-like technique where examples are taken from the image itself at different scales (Ebrahimi and Vrscay 2007); *movie flicker stabilization*, compensating for spurious oscillations in the colours of successive frames (Delon and Desolneux 2009); and *super-resolution*, an image zooming method fusing several frames from a video, or several low-resolution photographs, into a larger image (Protter, Elad, Takeda and Milanfar 2009). The main point of this super-resolution technique is that it does not give an explicit estimate of the motion, in fact allowing for multiple motions, since a block can look like several other patches in the same frame. The very same observation is made by Ebrahimi and Vrscay (2008*b*) on devising a super-resolution algorithm, and also by Elad and Datsenko (2007) and Danielyan, Foi, Katkovnik and Egiazarian (2008). Other classic image non-local applications include image *contrast enhancement*, by applying a reverse non-local heat equation (Buades, Coll and Morel 2006*a*), and *stereo vision*, by performing simultaneous non-local depth reconstruction and restoration of noisy stereo images (Heo, Lee and Lee 2007).

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

The link to PDEs, variational variants

The relationship of neighbourhood filters to classic local PDEs was discussed by Buades, Coll and Morel (2006*b*, 2006*c*), leading to an adaptation of NL-means which avoids the staircase effect. Non-local image-adapted differential operators and non-local variational methods were introduced by Kindermann, Osher and Jones (2006), who proposed denoising and deblurring by non-local functionals. The general goal of this development is in fact to give a variational form to all neighbourhood filters, and to give a non-local form to the total variation as well (Rudin, Osher and Fatemi 1992). Several articles on deblurring have followed this variational approach: Jung and Vese (2009), Mignotte (2008) and Gilboa and Osher (2008) for image segmentation; Boulanger *et al.* (2008) for fluorescence microscopy; Zhang, Burger, Bresson and Osher (2009) again for non-local deconvolution; and Lou, Zhang, Osher and Bertozzi (2008) for deconvolution and tomographic reconstruction. Elad and Datsenko (2007), in a paper dedicated to another notoriously ill-posed problem, that of super-resolution, view the non-local variational principle as an 'emerging powerful family of regularization techniques', and propose use of 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 PDEs has emerged, whose coefficients are in fact image-dependent. For instance, Elmoataz *et al.* (2008*b*) view the image colourization 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, Szlam, Maggioni and Coifman (2006) view the set of patches as a weighted graph, and the weights of the edge between two patches centred at $\mathbf{i}$ and $\mathbf{j}$, 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 neighbourhood filter can be associated with a heat equation on the image graph (Peyré 2009). This approach has been further extended to a variational formulation on patch graphs by Elmoataz, Lézoray and Bougleux (2008*a*). In this same framework Buades *et al.* (2006*a*) proposed image contrast enhancement via a non-local reverse heat equation. Finally, still in this non-local partial differential framework, Bresson and Chan (2008) extend 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.

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

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 is the sum of thermal noise and electronic noise, which are approximately additive and white.

For sufficiently large values of $\tilde{u}(\mathbf{i})$ (*e.g.*, $\tilde{u}(\mathbf{i}) > 1000$), the normal distribution $\mathcal{N}(\tilde{u}(\mathbf{i}), \sqrt{\tilde{u}(\mathbf{i})})$ is an excellent approximation to the Poisson distribution. If $\tilde{u}(\mathbf{i})$ is larger than 10, then the normal distribution is still 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 research 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. The most classic variance stabilizing transformation (VST) is the Anscombe transform (Anscombe 1948), which has the form $f(u_0) = b\sqrt{u_0 + c}$.

The denoising procedure with the standard 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 (Makitalo and Foi 2011).

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,

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

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

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

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

and integrating gives

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

When a linear variance noise model is chosen, this transformation gives 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 (Nowak and Baraniuk 1997, Kolaczyk 1999) propose adaptation of the transform threshold to the local noise level of the Poisson process. Lefkimmiatis, Maragos and Papandreou (2009) have explored a Bayesian approach without applying a VST. Deledalle, Denis and Tupin (2011*a*) argue that for high noise level it is better to adapt NL-means than to apply a VST. They propose replacing the Euclidean distance between patches with a likelihood estimate, taking into account the noise model. This distance can be adapted to each noise model, such as Poisson, Laplace or gamma noise (Deledalle, Tupin and Denis 2010*a*), and to more complex (speckle) noise occurring in radar (SAR) imagery (Deledalle, Tupin and Denis 2010*b*).

Nonetheless, dealing with white uniform Gaussian noise makes the discussion of denoising algorithms far easier. Recent papers on the Anscombe transform by Makitalo and Foi (2011) (for low-count Poisson noise) and Foi (2011) (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 based on signal-dependent noise models. This explains why, in the rest of this paper, the noise is assumed to be uniform, white and Gaussian, having previously applied a VST to the noisy image if necessary. 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 provided by many cameras contain 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 specific adaptation. The key is to have a decent noise model. For this reason, the fundamentals of estimating noise from a single image will be given in Section 2.2.

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

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

Compared to the denoising literature, research on noise estimation is a poor relation. Few papers are dedicated to this topic. Among recent papers we can mention that of Zoran and Weiss (2009), which argues that images are scale-invariant and therefore noise can be estimated by deviation from this assumption. Unfortunately this method is not easily extendable to the estimation of scale-dependent or signal-dependent noise, such as that 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. Efficient methods are described by Foi, Trimeche, Katkovnik and Egiazarian (2008) and Foi, Alenius, Katkovnik and Egiazarian (2007) for Poisson and Gaussian noise.

In this short review we will focus on methods that allow for local, signal-dependent and scale-dependent noise. Indeed, one cannot denoise an image without knowing its noise model. It might be argued that the noise model comes from 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 these operations change the noise model and make it *signal- and scale-dependent*.

The question that arises is: Why are so many researchers 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 procedure permits reliable evaluation of denoising performance, based on a controlled underlying 'true' image. Nevertheless denoising performance may, after all, critically depend on how well we are able to estimate the noise. Most 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 pixel blocks, computes their discrete cosine transform (DCT), quantizes the coefficients, and small coefficients are replaced by zero. Thus JPEG performs frequency-dependent thresholding, 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, demosaicing, white balance and tone curve (contrast change) alter this signal-dependency in a way that depends on the image itself.

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press



(a) (b)

Figure 2.1. Two examples of the ten noise-free images used in the tests: (a) *computer* and (b) *traffic*.

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 proper denoising of 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 is well explained in Liu *et al.* (2008) and Liu, Freeman, Szeliski and Kang (2006). Danielyan and Foi (2009) argue that noise can be estimated by using a denoising algorithm. Again, this procedure is probably too risky for noise and scale-dependent signals.

This section, following Buades, Colom and Morel (2012*a*), 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 estimation of 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 in good lighting conditions and with a low ISO. This means that the number of photons reaching each captor was very high, and the noise level therefore small. To reduce the noise level further, the average of each block of 5×5 pixels was computed, reducing the noise by a factor of 5. Since the images are RGB, taking the mean of the three channels reduces the noise by a further factor of $\sqrt{3}$. The (small) initial noise was therefore reduced by a factor of $5\sqrt{3} \simeq 8.66$, and the images can be considered noise-free. Two images from this noiseless set can be seen in Figure 2.1. The size of each image is 704×469 pixels. For the uniform-noise tests, seven noise levels were applied to these noise-free images: $\sigma \in \{1, 2, 5, 10, 20, 50, 80\}$.

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

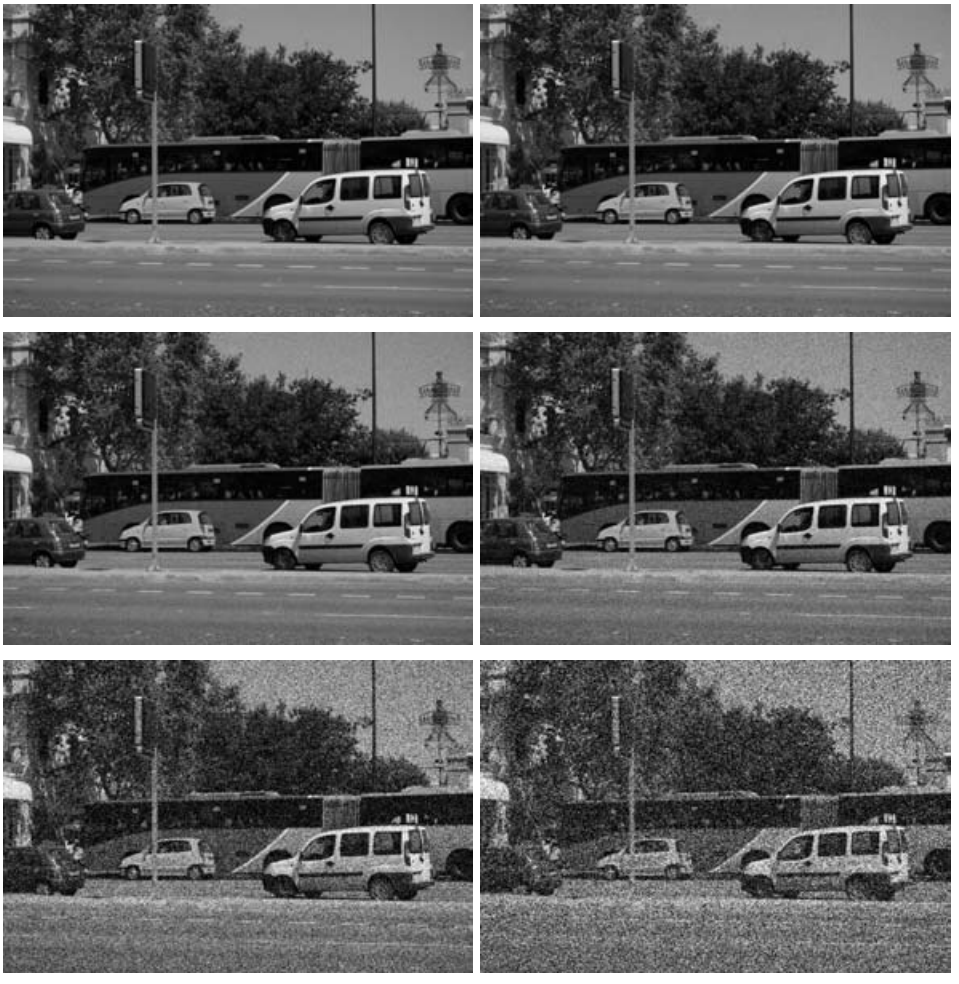


Figure 2.2. Result of adding white homoscedastic Gaussian noise with $\sigma \in \{2, 5, 10, 20, 50, 80\}$ to the noise-free image *traffic*. It may need a zoom in to perceive the noise for $\sigma = 2, 5$.

Figure 2.2 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 of noise estimation proceeds as follows. In Section 2.3 we review in detail the method proposed by Buades *et al.* (2012*a*). This method has all the features of the preceding methods, so we shall be able to make a quick review (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 noise remains somewhat inaccurate, with some 20% error for noises with standard deviation below 2.

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

2.3. The percentile method

The percentile method, introduced in Ponomarenko *et al.* (2007), 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 in 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 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

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

where

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

Therefore, the values of the associated discrete filter are

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

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

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

Table 2.1. Percentile method results for 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 for medium and high noise. It is nevertheless larger for very low noise ($\sigma = 2$ noise is not visible with the naked eye). Indeed, most photographed objects have some micro-texture everywhere (except perhaps in 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 pixel 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 pixel block size and a DCT with support 3×3. Estimating the best parameters therefore requires a first estimate followed by a second one with the correct parameters.

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

contains (at least) 42 000 elements. These measurements permit, for each interval of means, construction of a histogram of block variance with at least 42 000 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 best results are given by the percentile $p = 0.5\%$, a block size $w = 21$ and a 7×7 support for the DCT transform. 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 learned from simulations. For the 0.5% percentile, 21×21 pixel blocks and the DCT pre-filter operator with support 7×7, the empirical factor derived from pure 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.

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

Algorithm 2 Percentile method algorithm

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

$h =$ FILTER$(\tilde{u})$. Apply high-pass filter to the image.
$a, v =$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. For each interval keep the corresponding values in v.

$S = \emptyset$; $M = \emptyset$.
for each bin **do**
  $v =$ Per(bin, p). Get the p-percentile v of the block variances whose means belong to this bin.
  $m =$ Mean[Per(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 $k = 1 \ldots$ filt **do**
  $S_c[k] =$ FILTER$(S_c[k]$, filt$)$. Filter the noise curve filt times.
end for

For each observed grey-scale value in the image, this noise curve predicts the most likely underlying standard deviation due to noise. Optionally, the noise curve obtained from real images can be filtered. Indeed, it may present some peaks when variances measured for a given grey-scale 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 onto that segment. In general, no more than two filtering iterations are needed. For the comparative tests presented here, the curves were not filtered at all.

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

The pseudo-code for the percentile method is given in Algorithm 2, and the results for the white homoscedastic Gaussian noise in Table 2.1. When the image is tested for white homoscedastic Gaussian noise, only one interval for all grey-scale means is used, whereas in the signal-dependent noise case, the grey-scale interval is divided into seven bins.

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 Figure 2.3. 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 learned 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 learned 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 becomes dominated by noise, that is, most of the variance measured is due to the noise, and not due to micro-textures and edges. It is therefore convenient to avoid applying the learned corrections to direct estimates $\hat{\sigma}$ when $\hat{\sigma} > 10$. Thus, for $\hat{\sigma} > 10$, only the percentile correction is applied. Table 2.2 shows the $\hat{\sigma}$ values estimated with the percentile with learning method. The correction learned 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 one method in detail, as above, namely the percentile method. Most noise estimation methods share the following features.

- They start by applying a high-pass filter, which concentrates the image energy on boundaries, while the noise remains spatially homogeneous.
- They compute the energy for many blocks extracted from this high-passed image.
- They estimate the noise standard deviation from the values of the standard deviations of the blocks.

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

Table 2.2. Percentile method with learning results, with white homoscedastic Gaussian noise added. The correction learned with the mosaic is only applied for $\sigma \in \{1, 2, 5, 10\}$. This method, being local on blocks, extends immediately to estimation of signal-dependent noise, and the performance is similar (Buades *et al.* 2012*a*).

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



Figure 2.3. Mosaic used to learn the correction values in the percentile method.

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

- 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 2.3 shows a classification of the methods according to the preceding criteria. The 'High-pass' column shows the choice of high-pass filter, which can be a discrete differential operator of order two $\left(\frac{\partial^2}{\partial x \partial y}\right)$ in the *estimation of image noise variance* (EINV) method (Rank, Lendl and Unbehauen 1999). It is obtained as a composition of two forward discrete differences. Then we have a discrete Laplacian Δ (Olsen 1993), obtained as the difference between the current pixel value and the average of a discrete neighbourhood, an order-three operator (a difference $\Delta_1 - \Delta_2$ of two different discretizations of the Laplacian (Immerkaer 1996)), a wave associated to a DCT coefficient (Buades *et al.* 2012*a*), and sometimes a non-linear discrete differential operator as in the *median* method (Olsen 1993), 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 was previously applied to all pixels of the image. In the case of the DCT (Ponomarenko *et al.* 2003), the DCT is applied to a block centred on the reference pixel, and the highest-frequency coefficients are kept, for example DCT$(6, 7)$, DCT$(7, 6)$, DCT$(7, 7)$. The most primitive methods, the *block* method (Lee 1981, Mastin 1985), the *pyramid* method (Meer, Jolion and Rosenfeld 1990) and the *scatter* method (Lee and Hoppel 1989), 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 'Block' 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 (Meer *et al.* 1990) uses standard deviations of blocks of all sizes and is unclassifiable. Two methods, *FNVE* (Immerkaer 1996) and the *gradient* method (Bracho and Sanderson 1985, Voorhees and Poggio 1987), do not compute any block standard deviation of the high-passed image before the final estimation.

The 'Percentile' 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 takes into account all the values.

The 'Estimator' column characterizes the estimator, for which there are several variants. The three compared percentile methods (Buades *et al.* 2012*a*) use a very low percentile 0.5% of the block standard deviations. The *average*, *median* (Olsen 1993) and *block* methods (Lee 1981, Mastin 1985) use a 1% percentile of the gradient to select the blocks for which variance is kept, while the high-pass image is a higher-order differential operator. The

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

Table 2.3. Table summarizing all methods. 'Block deviation' means standard deviation of block; 'at percentile 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 pixel 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 1–5) 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. Rows 8–11 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.

Method	Source	High-pass	Block	Estimator	Percentile
percentile with learning	Buades *et al.* (2012*a*), Ponomarenko *et al.* (2007)	DCT 7×7	21	block deviation at percentile	0.5%
percentile	Buades *et al.* (2012*a*), Ponomarenko *et al.* 2007)	DCT 7×7	21	block deviation at percentile	0.5%
block	Lee (1981), Mastin (1985)	none	7	mean of block deviation	1%
average	Olsen (1993)	Δ	3	mean of block deviation	1% of gradient histogram
median	Olsen (1993)	curv	3	mean of block deviation	1% of gradient histogram
scatter	Lee & Hoppel (1989)	none	8	block deviation at	block deviation mode
gradient	Bracho & Sanderson (1985), Voorhees & Poggio (1987)	∇	1	$\lvert\nabla\rvert$ mode	all
EINV	Rank, Lendl & Unbehauen (1999)	$\frac{\partial^2}{\partial x\partial y}$	3	deconvolution of block dev.	all
FNVE	Immerkaer (1996)	$\Delta_1-\Delta_2$	1	RMS	all
DCT-MAD	Donoho & Johnstone (1995)	3-DCT	8	MAD of 3 DCT coefficients	all
DCT-mean	Ponomarenko *et al.* (2003)	3-DCT	8	mean of variances	all
pyramid	Meer, Jolion & Rosenfeld (1990)	none	2^L	block deviation	complex

Table 2.4. White homoscedastic Gaussian noise RMSE results for all methods and for varying σ. The pyramid tests were omitted, being incomplete. As they are obtained as an average over many noiseless images, the differences have been checked to be statistically significant. It is also clear that the ranking of the compared methods may vary with the amount of noise. Nevertheless, the ranks of methods for noises larger than 20 are irrelevant, because all of them work at an acceptable level of precision. Thus, this ranking is mainly relevant to low noise levels, $\sigma = 1, 2, 5, 10$.

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
EINV	1.406	1.159	0.924	0.842	0.656	0.450	0.557
FNVE	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

pyramid method (Meer *et al.* 1990) is quite complex, but uses all standard deviations of all possible blocks in the image. We do not provide the detailed algorithm. The *FNVE* method (Immerkaer 1996) in fact has 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) must correspond to the noise. The *gradient* method (Bracho and Sanderson 1985, Voorhees and Poggio 1987) sets $\hat{\sigma}$ to be the peak of the modulus of the gradient histogram. The *scatter* method (Lee and Hoppel 1989) 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 *EINV* method (Rank *et al.* 1999) does a kind of iterative deconvolution of the histogram of block variances and also extracts its mode.

All the values obtained by these methods are proportional to the noise standard deviation when the image is 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 (Section 2.3).

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

The comparison of the methods which use the highest DCT coefficients, DCT-mean (Ponomarenko *et al.* 2003) and DCT-MAD (Donoho and Johnstone 1995), where MAD stands for *median of absolute deviation*, clearly shows the advantage of a robust estimator: the estimate is obtained by averaging the three medians of absolute deviation of the three highest-frequency DCT coefficients for all blocks.

The ultimate choice of method is of course steered by the RMSE, that is, the root mean square error between the estimated value of σ and σ itself, taken over a representative set of images. As Table 2.4 shows, the ordering of methods by their RMSE is consistent and points to the percentile method as the best one. This method is further 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 $\sigma < 5$.

3. Four denoising principles

In this section we will review the main algorithmic principles proposed for noise removal. All of them of course use a model for the noise, which in our study will always be 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 in fact a smart combination of *all* these ingredients.

3.1. Bayesian patch-based methods

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

$$\tilde{u} = u + n. \tag{3.1}$$

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

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

$$\mathbb{P}(\tilde{u} \mid u) = \frac{1}{(2\pi\sigma^2)^{\frac{M}{2}}} \mathrm{e}^{-\frac{\|u-\tilde{u}\|^2}{2\sigma^2}}, \tag{3.2}$$

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 (Geman and Geman 1984), this prior was a parametric image model describing the stochastic behaviour 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

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

where Z and T are constants and E is called the energy function, and is given by

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

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

$$\arg\max_u \mathbb{P}(u \mid \tilde{u}) = \arg\max_u \mathbb{P}(\tilde{u} \mid u)\mathbb{P}(u),$$

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

$$\arg\min_u \|u - \tilde{u}\|^2 + \frac{2\sigma^2}{T} E(u).$$

This energy is thus a sum of local derivatives of pixels in the image, and is therefore equivalent to a classical Tikhonov regularization (Geman and Geman 1984, Brémaud 1999).

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 learned 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 'neighbourhood' or 'clique', previously used for the Markov field methods. In the non-parametric models the patches are larger, usually 8×8, while the cliques are often confined to 3×3 neighbourhoods. Given a noiseless patch P of u with dimension $\kappa \times \kappa$, and an observed noisy version $\tilde{P}$ of P, the same model gives

$$\mathbb{P}(\tilde{P}|P) = c \cdot \mathrm{e}^{-\frac{\|\tilde{P}-P\|^2}{2\sigma^2}}, \tag{3.3}$$

by the independence of noise pixel values, where P and $\tilde{P}$ are regarded as

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

vectors with κ^2 components and $\|P\|$ denotes the Euclidean norm of P, *i.e.*, the Frobenius norm on $\kappa \times \kappa$ matrices. 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

$$\mathbb{P}(P|\tilde{P}) = \frac{\mathbb{P}(\tilde{P}|P)\mathbb{P}(P)}{\mathbb{P}(\tilde{P})}. \tag{3.4}$$

Given $\tilde{P}$, 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 generally be learned from the image itself, or from a set of images. For example, Chatterjee and Milanfar (2012) apply a clustering method to the set of patches of a given image, and Zoran and Weiss (2011) apply 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 us to associate the likeliest cluster with each observed patch, and then to denoise it with a Bayesian estimate in this cluster. A 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 (Lebrun, Buades and Morel 2011). We shall now discuss this particular case, where all observed patches are noisy.

Why Gaussian? As usual when we have several observations but no particular knowledge of 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, *e.g.*, texture, slight lighting changes, shadows. The Gaussian model, together with a combination of many such random and independent factors, is heuristically justified by the central limit theorem. Thus, for good or ill, 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

$$\mathbb{P}(Q) = c \cdot \mathrm{e}^{-\frac{(Q-\overline{P})^T \mathbf{C}_P^{-1}(Q-\overline{P})}{2}}. \tag{3.5}$$

From (3.2) and (3.4), for each observed $\tilde{P}$ we obtain the following equivalent problems:

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

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

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 (3.1) and the fact that P and the noise n are independent,

$$\mathbf{C}_{\tilde{P}} = \mathbf{C}_P + \sigma^2\mathbf{I}, \quad \mathbb{E}\tilde{Q} = \overline{P}. \tag{3.6}$$

Note that these relations assume a search for patches similar to $\tilde{P}$, at a sufficiently large distance to include all patches similar to P but not so large that it contains outliers. Thus the safe strategy is to search for similar patches at 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 reduces via (3.6) 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 estimate

$$\hat{P}_1 = \overline{\tilde{P}} + \left[\mathbf{C}_{\tilde{P}} - \sigma^2\mathbf{I}\right]\mathbf{C}_{\tilde{P}}^{-1}(\tilde{P} - \overline{\tilde{P}}), \tag{3.7}$$

which resembles a local Wiener filter.

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

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

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

Sections 5.2–5.6 and 5.9 will examine no less than *six Bayesian algorithms* deriving patch-based denoising algorithms from variants of (3.7). The first question, when looking at this formula, is obviously how the matrix $\mathbf{C}_{\tilde{P}}$ can be learned from the image itself. Each method proposes a different approach to learning the patch model.

Of course, other non-Gaussian Bayesian models are possible, depending on patch density assumptions. For example, Raphan and Simoncelli (2010) assume a local exponential density model for the noisy data, and give a convergence proof for the optimal (Bayes) least-squares estimator as the amount of data increases.

3.2. Transform thresholding

Classical transform coefficient thresholding algorithms, such as the DCT or 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 a homoscedastic de-correlated noise remain de-correlated and homoscedastic. For example, the wavelet or DCT coefficients of Gaussian white noise with variance σ^2 remain Gaussian white noise 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 (Antonini, Barlaud, Mathieu and Daubechies 1992), on biorthogonal wavelet transforms (Cohen, Daubechies and Feauveau 1992).

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 $\tilde{U}$ (regarded here as a 'long' vector). Then we have

$$\langle \tilde{U}, G_i \rangle = \langle U, G_i \rangle + \langle N, G_i \rangle, \tag{3.8}$$

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

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

zero mean and variance σ^2:

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

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

$$\mathbf{D}\tilde{U} = \sum_{i=1}^{M} a(i) \langle \tilde{U}, G_i\rangle G_i. \tag{3.9}$$

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

Theorem 3.1. The operator $\mathbf{D}_{\text{inf}}$ minimizing the mean square error (MSE),

$$\mathbf{D}_{\text{inf}} = \arg\min_{\mathbf{D}} \mathbb{E}\{\|U - \mathbf{D}\tilde{U}\|^2\},$$

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

$$a(i) = \frac{|\langle U, G_i\rangle|^2}{|\langle U, G_i\rangle|^2 + \sigma^2}, \tag{3.10}$$

and the corresponding expected MSE is

$$\mathbb{E}\{\|U - \mathbf{D}_{\text{inf}}\tilde{U}\|^2\} = \sum_{i=1}^{M} \frac{|\langle U, G_i\rangle|^2 \sigma^2}{|\langle U, G_i\rangle|^2 + \sigma^2}. \tag{3.11}$$

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 using

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

and the corresponding MSE is

$$\mathbb{E}\{\|U - \mathbf{D}_{\text{inf}}\tilde{U}\|^2\} = \sum_{i} \min(|\langle U, G_i\rangle|^2, \sigma^2). \tag{3.12}$$

A *transform thresholding* algorithm therefore keeps the coefficients with a magnitude larger than the noise, while setting the zero the rest. Note that

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

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 following subsections. For the moment, we shall introduce the classical thresholding filters, which approximate the oracle coefficients by using the noisy ones.

As is classical, the optimal operator (3.10) is called the *Fourier–Wiener filter* 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 (2003) and Yaroslavsky (1996). The noisy image is analysed in a moving window, and at each position of the window its DCT spectrum is computed and modified by using the optimal operator (3.10). 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

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

and thus replacing the optimal attenuation coefficients $a(i)$ with 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 (Mallat 1999). The so-called *hard wavelet thresholding method* (Donoho and Johnstone 1994) is a (non-linear) projection operator setting to zero all wavelet coefficients smaller than a certain threshold. According to the expression of the MSE of a projection operator (3.12), 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 (Pennec and Mallat 2003).

Unfortunately, image features, just like noise, can also cause many small wavelet coefficients, which are nevertheless lower than the threshold. The dramatic cancellation 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

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

by the undue cancellation of some of the small coefficients. These artifacts are sometimes called *wavelet outliers* (Durand and Nikolova 2003). These undesirable effects can be partially avoided with the use of soft thresholding (Donoho 1995), for example,

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

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' (Pennec and Mallat 2003) and 'curvelets' (Starck, Candès and Donoho 2002). This tendency to adapt the transform locally to the image is accentuated by 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 perspective is sketched in the next subsection, on sparse coding.

3.3. Sparse coding

Sparse coding algorithms learn a redundant set $\mathbf{D}$ of vectors called a *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 $\kappa^2 \times n_{\mathrm{dict}}$, where κ^2 is the number of pixels in the patch and $n_{\mathrm{dict}} \geq \kappa^2$. The dictionary patches, which are columns of the matrix, are normalized (in the Euclidean norm). This dictionary may contain the usual orthogonal bases (*e.g.*, discrete cosine transform, wavelets, curvelets), but also patches extracted (or learned) from clean images, or even from the noisy image itself.

The dictionary permits computation of a sparse representation α of each patch P, where α is a coefficient vector of size n_{dict}^2 satisfying $P \approx \mathbf{D}\alpha$. This sparse representation α can be obtained with an ORMP (orthogonal recursive matching pursuit: Cotter, Adler, Rao and Kreutz-Delgado (1999)). The ORMP gives an approximate solution to the (NP-complete) problem

$$\arg\min_{\alpha} \|\alpha\|_0 \quad \text{such that} \quad \|P - \mathbf{D}\alpha\|_2^2 \leq \kappa^2 (C\sigma)^2, \tag{3.13}$$

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, 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 Cotter *et al.* (1999). Details of how this minimization can be achieved are given in Section 5.7 on the K-SVD algorithm. (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.

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

This remark is the starting point of the compressive sampling method; see Candès and Wakin (2008).)

In K-SVD and other current sparse coding algorithms, the previous denoising strategy is used as the first step of a two-step 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.

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 to 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 factor of $\sqrt{m}$.

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

$$\begin{aligned}\mathbb{E}\left\|u(\mathbf{i}) - \sum_{\mathbf{j}} w(\mathbf{j})\tilde{u}(\mathbf{j})\right\|^2 &= \mathbb{E}\left\|\sum_{\mathbf{j}} w(\mathbf{j})(u(\mathbf{i}) - u(\mathbf{j})) - \sum_{\mathbf{j}} w(\mathbf{j})n(\mathbf{j})\right\|^2 \\ &= \sum_{\mathbf{j}} w(\mathbf{j})^2(u(\mathbf{i}) - u(\mathbf{j}))^2 + \sigma^2 \sum_{\mathbf{j}} w(\mathbf{j})^2, \quad (3.14)\end{aligned}$$

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 trade-off between the bias and variance terms of equation (3.14).

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

Averaging spatially close pixels

An initial rather trivial idea is to average the closest pixels to a given pixel. This amounts to convolving the image with a fixed radial positive kernel. The archetype 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. But nearby pixels do not necessarily share their colours. Thus, the first error term in (3.14) can quickly increase. This approach is only valid for pixels for which 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 (Lee 1983) or neighbourhood filter (Yaroslavsky 1985). These filters average only nearby pixels of $\mathbf{i}$ having also a similar colour value. We shall denote these filters by YNF (Yaroslavsky neighbourhood filter). Their formula is simply

$$\mathrm{YNF}_{h,\rho}\,\tilde{u}(\mathbf{i}) = \frac{1}{C(\mathbf{i})} \sum_{\mathbf{j}\in B_\rho(\mathbf{i})} \tilde{u}(\mathbf{j})\, \mathrm{e}^{-\frac{|\tilde{u}(\mathbf{i})-\tilde{u}(\mathbf{j})|^2}{h^2}}, \tag{3.15}$$

where $B_\rho(\mathbf{i})$ is a ball of centre $\mathbf{i}$ and radius $\rho > 0$, $h > 0$ is the filtering parameter, and

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

is the normalization factor. The parameter h controls the degree of colour similarity that needs to be taken into account in the average. According to the Bayesian interpretation of the filter we should have $h = \sigma$. The filter (3.15), due to Yaroslavsky and Lee, has been reinvented several times, and has received the alternative names of *SUSAN filter* (Smith and Brady 1997) and *bilateral filter* (Tomasi and Manduchi 1998). The relatively minor difference in these algorithms is that instead of considering a fixed spatial neighbourhood $B_\rho(\mathbf{i})$, they weigh the spatial distance to the reference pixel $\mathbf{i}$ by a Gaussian.

Neighbourhood filters choose the 'neighbouring' 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 (3.14) 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 neighbourhood) is broad enough, then the average

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

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 Buades (2006).

Theorem 3.2. Assume that $n(\mathbf{i})$ are independent identically distributed, with zero mean and variance σ^2. Then a noise n filtered by the neighbourhood filter YNF_h satisfies

$$\operatorname{Var} \mathrm{YNF}_{h,\rho}\, n = f\left(\frac{h}{\sigma}\right)\sigma^2,$$

where

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

and

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

The function $f(x)$ is decreasing with $f(0) = 1$ and

$$\lim_{x\to\infty} f(x) = 0$$

(see Figure 3.1). 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 that in fact

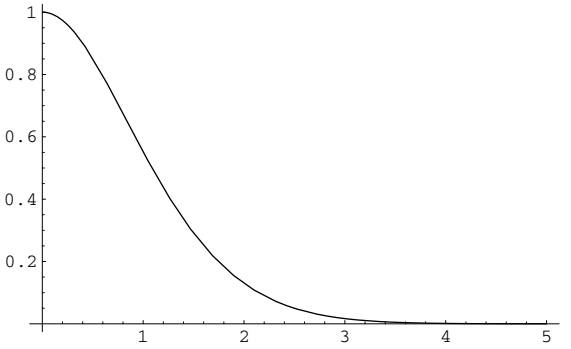


Figure 3.1. Noise reduction function $f(x)$ given by Theorem 3.2.

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

belong to other objects. This explains the observed decaying performance of the neighbourhood filter when the noise standard deviation or the search zone $B(\mathbf{i}, \rho)$ increase too much.

The image model underlying neighbourhood filters is that of *image self-similarity*, namely the presence in the image of pixels $\mathbf{j}$ which have the same law as $\mathbf{i}$. In Section 5.1 we will introduce the NL-means algorithm (Buades *et al.* 2005*b*), which can be seen as an extension of the neighbourhood filters attenuating their main drawbacks. In NL-means, the 'neighbourhood of a pixel $\mathbf{i}$' is defined as any set of pixels $\mathbf{j}$ in the image such that a patch around $\mathbf{j}$ looks like a patch around $\mathbf{i}$. In other words NL-means estimates the value of $\mathbf{i}$ as an average of the values of all the pixels $\mathbf{j}$ whose neighbourhood looks like the neighbourhood of $\mathbf{i}$.

4. Noise reduction: generic tools

This section describes four generic tools that permit an increase in the performance of *any* denoising principle. We shall illustrate them for DCT denoising. Starting from the application of a simple DCT transform threshold, the four generic tools will be 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 of m. Such an aggregation strategy was the main proposition of the *translation-invariant wavelet thresholding algorithm* (Coifman and Donoho 1995). 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 us to attribute to each estimator a weight inversely proportional to its variance (Nemirovski 2000). 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, however, most DCT coefficients are cancelled and the noise is completely removed. The proposition of Guleryuz (2007) 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 by Yu and Sapiro (2011)

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

one can test an implementation of DCT denoising. They use 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' (Yu and Sapiro 2011).

The same risk of 'halos' occurs with non-aggregated NL-means (Section 5.1), since patches containing edges have far fewer similar instances in the image than flat patches. Thus the non-local averaging is made over fewer 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 (Dabov *et al.* 2007). As a consequence, an aggregation step is applied in all patch-based denoising algorithms. This weighted aggregation favours, 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 superior noise reduction by increasing the number of values being averaged to obtain the final estimate, or selecting those estimates with lower variance. Kervrann and Boulanger (2008) considered the whole bias + variance decomposition in order to also adapt the search zone of neighbourhood filters or NL-means. Since the bias term depends on the original image, it cannot be computed in practice, and Kervrann and Boulanger proposed minimization of both bias and variance by choosing the smallest spatial neighbourhood 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. Van De Ville and Kocher (2009) give a closed-form expression of Stein's unbiased estimator of the risk (SURE) for NL-means. (See also the generalizations of the SURE estimator to the non-Gaussian case in Raphan and Simoncelli (2007).) The aim is to select globally the best bandwidth for a given image. Duval, Aujol and Gousseau (2011) also use the SURE technique to minimize the risk by selecting the bandwidth locally. Deledalle, Duval and Salmon (2012) apply the same technique to combine the results of NL-means with different window sizes and shapes. A similar treatment can be found in Raphan and Simoncelli (2010), 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

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

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 neighbourhood 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 to compute more accurate colour distances between pixels. Thus the second step neighbourhood filter is given by

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

where $\tilde{u}$ is the observed noisy image and $\hat{u}_1$ the image previously denoised by (3.15).

Similarly, for linear transform Wiener-type methods, the image is first denoised by its classical definition, which amounts to approximating the oracle coefficients of Theorem 3.1 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 (3.9) is given by

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

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

Alternatives and extensions: 'twicing' and Bregman iterations

In the recent review paper by Milanfar (2011), many denoising operators are formalized in a general linear framework, noting that they can be associated with a doubly stochastic diffusion matrix W with non-negative coefficients. In NL-means, for example, 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 by Milanfar (2011) 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 the better estimate given by the previously denoised image. One does not generally observe much improvement by iterating the oracle method more than once. Milanfar (2011) points out another generic tool, used at least for total

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

variation denoising: so-called 'twicing', a term due to Tukey (1977). 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. Milanfar (2011) shows that the Bregman iterations (Osher *et al.* 2004) 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 with all colour channels simultaneously can be proposed. This solution is adopted by averaging filters such as neighbourhood 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 of variables is expressed as a linear transform by multiplication of the RGB vector by the matrices

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

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 noise reduction permits better performance of the algorithm in this component. It also permits 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.

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

4.4. Trying all generic tools on an example

This subsection incrementally applies the above generic denoising tools to the DCT sliding window, to illustrate how these additional tools permit drastic improvement of algorithm performance. We start with the basic DCT 'neighbourhood filter', as proposed by Yaroslavsky and Eden (2003). Its principle is to denoise a patch around each pixel, and to keep only the central denoised pixel.

Figure 4.1 displays the denoised images obtained by incrementally applying each of the following ingredients.



Figure 4.1. (a, b) Original and noisy images with additive Gaussian white noise of standard deviation 25. (c–h) Close-ups of denoised images by sliding DCT thresholding filters 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 PSNRs are 26.85, 27.33, 30.65, 30.73, 31.25, respectively.

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

- A basic DCT thresholding algorithm using the neighbourhood 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 and 'halo' effects are removed.
- Additional iteration using 'oracle' estimation: residual noise is totally removed and the sharpness of details is increased.

The PSNRs 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 achieving competitive results. This last version of the DCT denoising algorithm, incorporating all the proposed generic tools, will be the one used in Section 6. 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 Yu and Sapiro (2011), which does not use the oracle step.

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 5.1 and 5.2. This similarity is true for patches whose centres are within one pixel of the centre of the reference patch. In that case the self-similarity boils down to a local image regularity assumption. Such regularity is guaranteed by Shannon and Nyquist's sampling conditions, which require the image to be blurry. In a much more general sense inspired by neighbourhood filters, one can define the 'neighbourhood of a pixel **i**' to be any set of pixels **j** in the image such that a patch around **j** looks like a patch around **i**. All pixels in that neighbourhood can be used for predicting the value at **i**, as was first

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

Algorithm 3 DCT denoising algorithm. DCT coefficients lower than 3σ are cancelled 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 $\tilde{u}$, noise standard deviation σ.
Optional. Prefiltered image $\hat{u}_1$ for 'oracle' estimation.
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

$$w_{\tilde{P}} = 1/\{\text{number of non-zero DCT coefficients}\}.$$

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

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

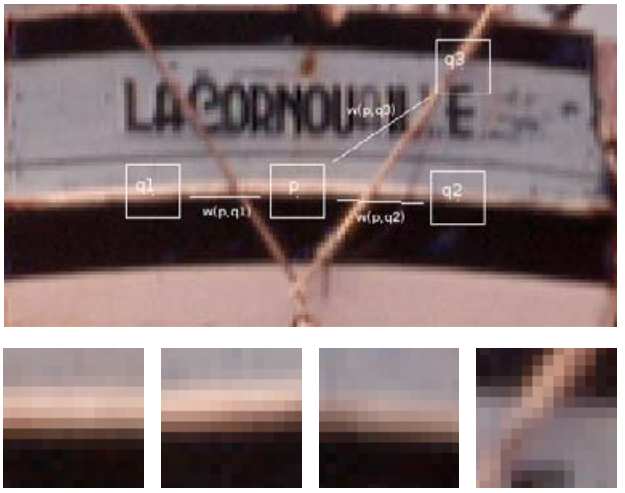


Figure 5.1. Here $q1$ and $q2$ have a large weight because their similarity windows are similar to that of p. On the other hand the weight $w(p, q3)$ is much smaller because the intensity grey-scale values in the similarity windows are very different.

shown in Efros and Leung (1999) for the synthesis of texture images. This self-similarity hypothesis is a generalized periodicity assumption. The use of self-similarities has in fact been well known in information theory from its foundation. In his paper 'A mathematical theory of communication', Shannon (2001) analysed the local self-similarity (or redundancy) of natural written language, and gave what was probably the first stochastic text synthesis algorithm. The Efros–Leung texture synthesis method adapted this algorithm to images, and NL-means (Buades, Coll and Morel 2004) seems to be the first adaptation of the same idea to denoising.[2]

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

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

[2] Nevertheless, some researchers have pointed out to us the report by De Bonet (1997) as giving an early discovery that intuition could use signal redundancy. This very short paper describes an experiment in a few sentences. It suggests that region redundancy on both sides of an edge can be detected, and used for image denoising. Nevertheless, no algorithm is specified in the paper.

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

The concentration of the noise law gives NL-means the edge over neighbourhood filters, 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

Weissman *et al.* (2005) proposed a 'universal denoiser' for digital images, and proved 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'. Ordentlich *et al.* (2003) presented an implementation valid for binary images with an impulse noise, with excellent results. Awate and Whitaker (2006) 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, that is, if one can find many samples of every image detail. It can be proved (Buades *et al.* 2005*b*) that if the image is a fairly general stationary and mixing random process, for every pixel **i**, NL-means converges to the conditional expectation of **i** knowing its neighbourhood, 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 can instead preserve straight edges, but flat zones present many artifacts. One might consider combining these methods to improve both results. A Gaussian convolution could be applied in flat zones, while an anisotropic filter could be applied on straight edges. However, other types of filters should be designed to specifically restore corners, or curved edges, or periodic texture. Figure 5.2 illustrates how NL-means chooses the right weight configuration for each type of image self-similarity.

NL-means is easily extended to the denoising of image sequences and video, indiscriminately involving pixels belonging to a space–time neighbourhood. The algorithm favours pixels with a similar local configuration. When the similar configuration moves, so do the weights. Thus, as shown by Buades, Coll and Morel (2008*b*), the algorithm is able to follow moving similar configurations without any explicit motion computation (see Figure 5.3).

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

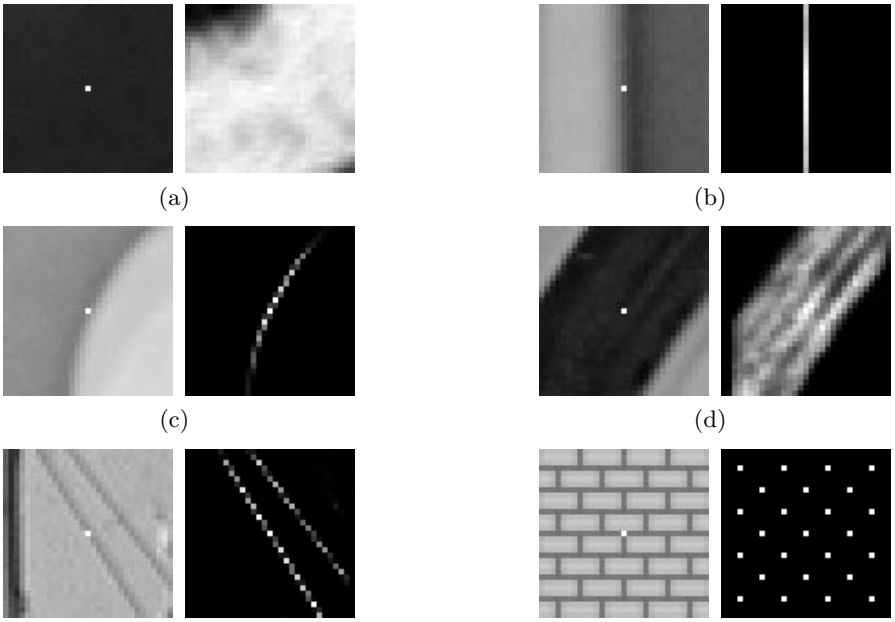


Figure 5.2. The right-hand image of each pair shows the weight distribution used to estimate a centred patch of the left-hand image by NL-means. (a) In flat zones, the weights are uniformly distributed, and NL-means acts like a low-pass isotropic filter. (b) On straight edges, the weights are distributed in the direction of the edge (as for anisotropic filters). (c) On curved edges, the weights favour pixels belonging to the same contour. (d) In a flat neighbourhood, the weights are distributed in a grey-scale neighbourhood (just as for neighbourhood filters). For (e) and (f), the weights are distributed across the more similar configurations, even though they are far away from the observed pixel. This behaviour justifies the term 'non-local'.

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 underlying 'true' image for the physical motion. Legitimate information about the colour of a given pixel should exist only along its physical trajectory. But 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 succeeding or preceding 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 Buades *et al.* (2008*b*) for more details of this discussion).

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

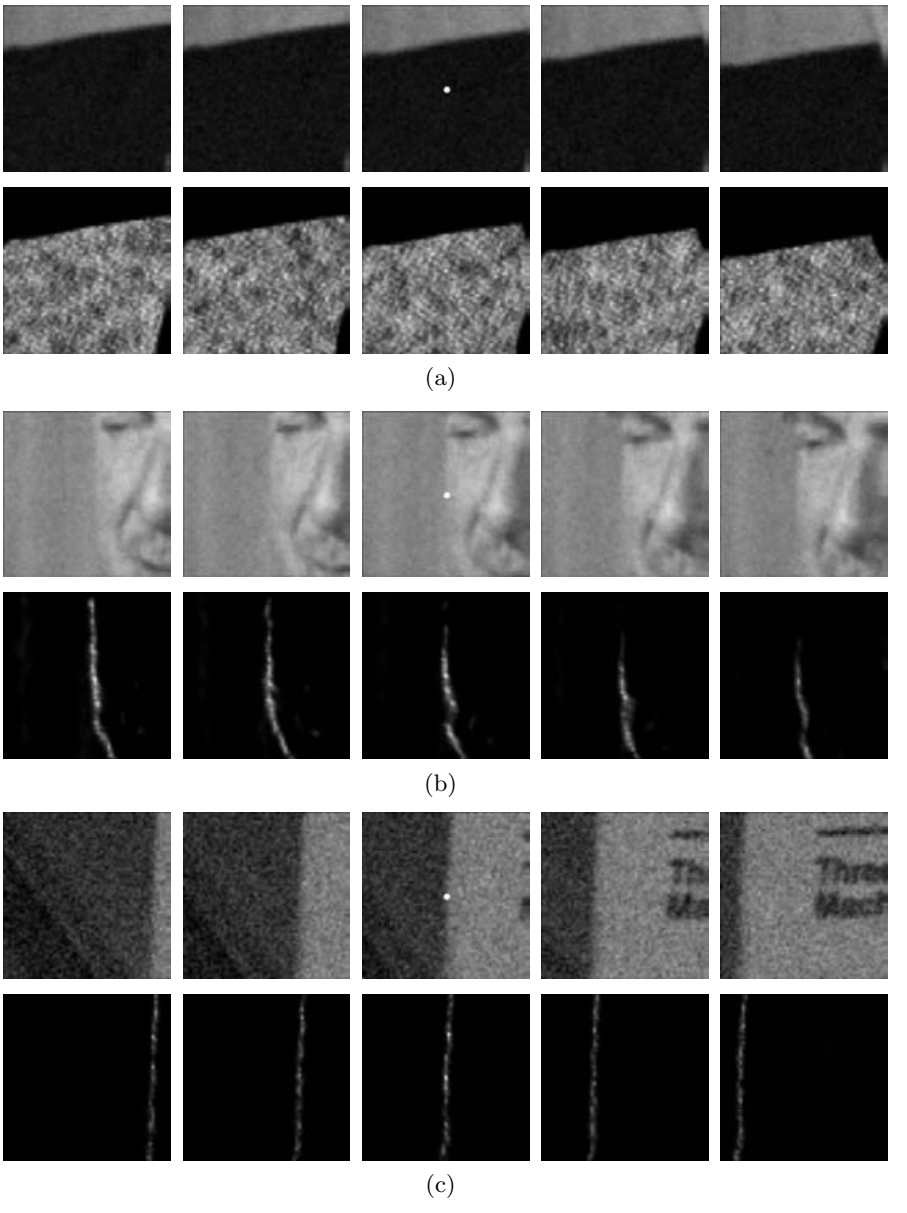


Figure 5.3. Weight distribution of NL-means applied to a movie. The upper row of each group (a), (b) and (c) shows a five-frame image sequence. The lower row shows the weight distribution used to estimate the central pixel (in white) of the middle frame is shown. The weights are equally distributed over 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 exploited for better denoising performance by using more pixels to compute the average.

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

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

Input. Noisy image $\tilde{u}$, noise standard deviation σ.
Output. Denoised image.

Set parameter $\kappa = 3$: size of patches.
Set parameter $\lambda = 31$: size of search zone for similar patches.
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 neighbourhood 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 $\tilde{Q}\mathrm{e}^{-\frac{d(\tilde{P},\tilde{Q})^2}{h^2}}$ to $\hat{P}$ and $\mathrm{e}^{-\frac{d(\tilde{P},\tilde{Q})^2}{h^2}}$ 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

5.2. Non-local Bayesian denoising

It is clear that (3.7) in Section 3.1,

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

is 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 by Lebrun *et al.* (2011) achieves this. Let $\mathcal{P}(\tilde{P})$ be the set of patches $\tilde{Q}$ similar to the patch $\tilde{P}$, which have been 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

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

large numbers, we have

$$\begin{aligned} \mathbf{C}_{\tilde{P}} &\simeq \frac{1}{\#\mathcal{P}(\tilde{P}) - 1} \sum_{\tilde{Q} \in \mathcal{P}(\tilde{P})} \big(\tilde{Q} - \overline{\tilde{P}}\big)\big(\tilde{Q} - \overline{\tilde{P}}\big)^T, \\ \overline{\tilde{P}} &\simeq \frac{1}{\#\mathcal{P}(\tilde{P})} \sum_{\tilde{Q} \in \mathcal{P}(\tilde{P})} \tilde{Q}. \end{aligned} \tag{5.1}$$

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 estimate is used as the 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 estimate $\mathbf{C}_{\hat{P}_1}$ for $\mathbf{C}_P$, the covariance of the cluster containing P, and a new estimate 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 (3.6) and (3.7) that we can obtain a second improved denoised patch, namely

$$\hat{P}_2 = \overline{\tilde{P}}^1 + \mathbf{C}_{\hat{P}_1}\big[\mathbf{C}_{\hat{P}_1} + \sigma^2 \mathbf{I}\big]^{-1}(\tilde{P} - \overline{\tilde{P}}^1), \tag{5.2}$$

where

$$\begin{aligned} \mathbf{C}_{\hat{P}_1} &\simeq \frac{1}{\#\mathcal{P}(\hat{P}_1) - 1} \sum_{\hat{Q}_1 \in \mathcal{P}(\hat{P}_1)} \big(\hat{Q}_1 - \overline{\tilde{P}}^1\big)\big(\hat{Q}_1 - \overline{\tilde{P}}^1\big)^T, \\ \overline{\tilde{P}}^1 &\simeq \frac{1}{\#\mathcal{P}(\hat{P}_1)} \sum_{\hat{Q}_1 \in \mathcal{P}(\hat{P}_1)} \tilde{Q}. \end{aligned} \tag{5.3}$$

We denote the denoised patches by $\overline{\tilde{P}}$ in (5.1) and $\overline{\tilde{P}}^1$ in (5.2). Indeed, in (5.2), the denoised version of $\tilde{P}$ is computed as the average of noisy patches $\tilde{Q}$ whose denoised patch is similar to $\hat{P}_1$.

In short, the estimates (3.7) and (5.2) appear to be equivalent, but in practice they are not. $\mathbf{C}_{\hat{P}_1}$, obtained after a first denoising step, is a better estimate 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. Since all the above quantities are computable from the noisy image, we obtain the two-step Algorithm 5.

As pointed out by Buades, Lebrun and Morel (2012*b*), the non-local Bayes algorithm is only an interpretation (with some generic improvements such as aggregation) of the PCA-based algorithm proposed by Zhang, Dong, Zhang and Shi (2010). This paper has a self-explanatory title: 'Two-stage image denoising by principal component analysis with local pixel grouping'. It is equivalent to applying PCA on the patches similar to $\tilde{P}$, followed by a Wiener filter on the coefficients of $\tilde{P}$ for this PCA, or applying formula (3.7) with the covariance matrix of the similar patches. Indeed, the PCA simply computes the eigenvalues of the empirical covariance matrix. Thus,

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

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:

$$\mathbf{C}_{\tilde{P}} \simeq \frac{1}{\#\mathcal{P}(\tilde{P}) - 1} \sum_{\tilde{Q} \in \mathcal{P}(\tilde{P})} \big(\tilde{Q} - \overline{\tilde{P}}\big)\big(\tilde{Q} - \overline{\tilde{P}}\big)^T,$$

$$\overline{\tilde{P}} \simeq \frac{1}{\#\mathcal{P}(\tilde{P})} \sum_{\tilde{Q} \in \mathcal{P}(\tilde{P})} \tilde{Q}.$$

Obtain the first-step estimate:

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

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:

$$\mathbf{C}_{\hat{P}_1} \simeq \frac{1}{\#\mathcal{P}(\hat{P}_1) - 1} \sum_{\hat{Q}_1 \in \mathcal{P}(\hat{P}_1)} \big(\hat{Q}_1 - \overline{\tilde{P}}^1\big)\big(\hat{Q}_1 - \overline{\tilde{P}}^1\big)^T,$$

$$\overline{\tilde{P}}^1 \simeq \frac{1}{\#\mathcal{P}(\hat{P}_1)} \sum_{\hat{Q}_1 \in \mathcal{P}(\hat{P}_1)} \tilde{Q}.$$

Obtain the second-step patch estimate:

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

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 $\mathbf{i}$.

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

the method in Zhang *et al.* (2010) finds its Bayesian interpretation. A study of the compared performance of local PCA versus global PCA for two-stage image denoising (TSID) is in fact proposed by Deledalle, Salmon and Dalalyan (2011*b*).

5.3. Patch-based locally optimal Wiener (PLOW)

In the non-local Bayes method of Section 5.2, a local model is estimated in a neighbourhood of each patch. In the PLOW method of Chatterjee and Milanfar (2012), however, the idea is to learn from the image a sufficient number of patch clusters, in fact 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 the fact that similar patches can have varying contrast, the inter-patch distance is photometrically neutral, and Chatterjee and Milanfar call it a 'geometric distance'. The clustering phase is accelerated by dimension reduction obtained by applying PCA 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 patches that are geometrically similar but not necessarily photometrically similar, the method identifies the photometrically similar patches for each patch in the cluster as being those whose quadratic distance to the reference patch are within the bounds allowed by noise. Then an LMMSE estimate (Kay 1993) is obtained for the reference patch by a variant of (3.7). 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 an oracle to obtain the covariance matrices of the sets of patches.

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 a sparse patch dictionary on which to decompose any given patch, from a single image or from a small set of images. The shotgun methods instead learn 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 Zoran and Weiss (2011), this patch space is organized as a Gaussian mixture with about 200 components. Shotgun methods have begun to be used in several image restoration methods: for example in Hays and Efros (2007), for image inpainting, with a fairly explicit title, 'Scene completion using millions of photographs'.

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

Algorithm 6 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 $\hat{\sigma} = 1.4826 \times \text{median}(|\nabla\tilde{U} - \text{median}(\nabla\tilde{U})|)$;
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 $w_{ij} = \mathrm{e}^{-\frac{\|\tilde{Q}_i - \tilde{Q}_j\|^2}{h^2}}$.
    Compute the LMMSE estimator for the noisy patch $\tilde{Q}_i$ (slightly more complex than usual, 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.

The approach of Levin and Nadler (2011) 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 would 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).

To evaluate the MMSE this experiment uses a set of 20 000 images from the LabelMe dataset (Russell, Torralba, Murphy and Freeman 2008). The

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

Algorithm 7 Shotgun NL-means

Input. Noisy image $\tilde{u}$ in vectorial form; 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 (5.4).
end for
At each pixel $\mathbf{i}$ get $\hat{u}(\mathbf{i})$ as $\hat{P}(\mathbf{i})$, where the patch P is centred 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 is not considered in Levin and Nadler (2011).)

method, even though certainly impractical, is of exquisite simplicity. Given a clean patch P, the noisy patch $\tilde{P}$ with Gaussian noise, of standard deviation σ, has probability distribution

$$\mathbb{P}(\tilde{P} \mid P) = \frac{1}{(2\pi\sigma^2)^{\frac{\kappa^2}{2}}} \mathrm{e}^{-\frac{\|P-\tilde{P}\|^2}{2\sigma^2}}, \tag{5.4}$$

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,

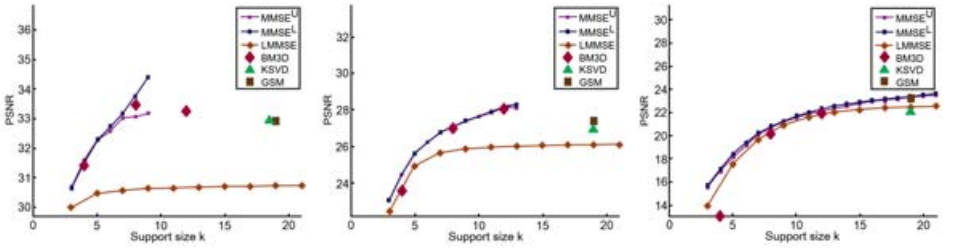
$$\hat{P} = \mathbb{E}[P \mid \tilde{P}] = \int \mathbb{P}(P \mid \tilde{P}) P \,\mathrm{d}P = \int \frac{\mathbb{P}(\tilde{P} \mid P)}{\mathbb{P}(\tilde{P})} \mathbb{P}(P) P \,\mathrm{d}P. \tag{5.5}$$

Using a huge set of M natural patches (with a distribution supposedly approximating the real natural patch density), we can approximate the terms in (5.5) by $\mathbb{P}(P)\,\mathrm{d}P \simeq \frac{1}{M}$ and $\mathbb{P}(\tilde{P}) \simeq \frac{1}{M}\sum_i \mathbb{P}(\tilde{P} \mid P_i)$, which in view of (5.4) 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 simply 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 database. The final algorithm is summarized in Algorithm 7.

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press



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

Figure 5.4. From Levin and Nadler (2011). The PSNR ($-10\log_{10}(\text{MMSE})$) of several denoising algorithms – K-SVD (Mairal *et al.* 2009*a*), BM3D (Dabov *et al.* 2007), Gaussian scale mixture (Portilla *et al.* 2003) – compared with the PNSR predicted by MMSE^L and MMSE^U. The performance of all algorithms is bounded by the MMSE^U estimate, but BM3D approaches this upper bound to within fractional dB values. Nevertheless, the performance bounds consider more restrictive patch-based algorithms than the class to which BM3D belongs. Thus the actual distance to optimality may be greater.

However, as mentioned earlier, the main focus of Levin and Nadler (2011) is elsewhere: they use 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* $\tilde{P}$*, and given the law* $p(P)$ *of all possible patches, 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 Levin and Nadler (2011) turns out to be very close to results obtained with BM3D (see Section 5.8), and they conclude that for small window sizes, or moderate to high noise levels, the quest for the best denoising algorithm might be close to an end. More precisely, only fractions of decibels separate the current best algorithms from these demonstrated upper bounds. The EPLL method (Zoran and Weiss 2011) 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 lower and upper bounds of Levin and Nadler (2011) can be estimated from a sufficient set of natural images.

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

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

error is

$$\mathrm{MSE} = \int \mathbb{P}(P) \int \mathbb{P}(\tilde{P} \mid P) \|P - \hat{P}\|^2 \, \mathrm{d}\tilde{P} \, \mathrm{d}P. \tag{5.6}$$

Conversely, one can start from a noisy patch $\tilde{P}$ and measure the variance of $\mathbb{P}(P \mid \tilde{P})$ around it. According to Levin and Nadler (2011), this amounts to computing the sum of weighted distances between the restored $\hat{P}$ and all possible P explanations:

$$\mathrm{MSE} = \int \mathbb{P}(\tilde{P}) \int \mathbb{P}(P \mid \tilde{P}) \|P - \hat{P}\|^2 \, \mathrm{d}\tilde{P} \, \mathrm{d}P. \tag{5.7}$$

This last equation follows from (5.6) by Bayes' rule. For each noisy $\tilde{P}$ one can define its MMSE:

$$\mathrm{MMSE}(\tilde{P}) = \mathbb{E}[\|\hat{P} - \tilde{P}\|^2 \mid \tilde{P}] = \int \mathbb{P}(P \mid \tilde{P})(P - \hat{P})^2 \, \mathrm{d}P. \tag{5.8}$$

The main interest of this formulation is that it allows us to prove that the MMSE, out of all the denoising algorithms, is the one that minimizes the overall MSE. Indeed, differentiating (5.7) with respect to $\hat{P}$ returns the MMSE estimator (5.5). The best overall MMSE achievable by any given denoising algorithm is therefore

$$\mathrm{MMSE} = \int \mathbb{P}(\tilde{P}) \mathbb{E}[\|\hat{P} - \tilde{P}\|^2 \mid \tilde{P}] \, \mathrm{d}\tilde{P} = \int \mathbb{P}(\tilde{P}) \mathbb{P}(P \mid \tilde{P})(P - \hat{P})^2 \, \mathrm{d}P \, \mathrm{d}\tilde{P}. \tag{5.9}$$

The goal of Levin and Nadler (2011) 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 lower bound on the MMSE from the two MSE formulations (5.6)–(5.7). 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

$$\mathrm{MMSE}^U = \frac{1}{M} \sum_j \|\hat{P}_j - P_j\|^2 \tag{5.10}$$

and

$$\mathrm{MMSE}^L = \frac{1}{M} \sum_j \frac{\sum_i \mathbb{P}(\tilde{P}_j \mid P_i) \|\hat{P}_j - P_i\|^2}{\sum_i \mathbb{P}(\tilde{P}_j \mid P_i)}. \tag{5.11}$$

A striking feature of both estimates is that MMSE^U uses explicit knowledge of the original noise-free patch P_j, while MMSE^L does not. Since MMSE^U simply measures the error for a given denoising algorithm, it

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

obviously provides an upper bound for the MMSE of any other denoising algorithm. As Levin and Nadler (2011) 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, Levin and Nadler (2011) give 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, Levin and Nadler (2011) 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.

5.5. The expected patch log likelihood (EPLL) method

The patch Gaussian mixture model

This other shotgun method (Zoran and Weiss 2011) is an almost literal application of the *piecewise linear estimator* (PLE) method (Yu, Sapiro and Mallat 2012) (see Section 5.9). But it is indeed shotgun, that is, it is applied to a huge set of patches instead of the image itself. A Gaussian mixture model is learned 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 public-domain MATLAB code.[3] Thus were learned: 200 means (in fact they are all zero), 200 full covariance matrices and 200 mixing weights, which constitute the Gaussian mixture model of this set of patches. Figure 5.5 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 learned, 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

$$\text{EPLL}_{\mathbb{P}}(U) = \sum_i \log \mathbb{P}(\mathbf{P}_i U),$$

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

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

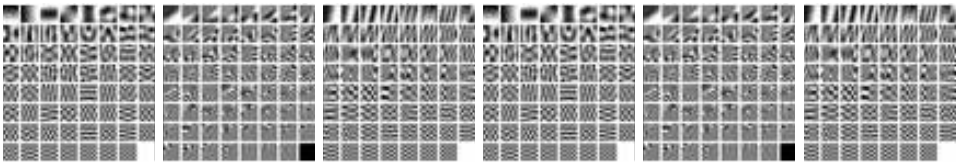


Figure 5.5. Eigenvectors of six randomly selected covariance matrices from the learned Gaussian mixture model, sorted by eigenvalue from largest to smallest (from Zoran and Weiss (2011)). The authors notice the similarity of these basis elements to DCT, but also that many seem to model texture boundaries and edges at various orientations.

where $\mathbf{P}_i$ is a matrix which extracts the ith 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 ith patch under the prior $\mathbb{P}$. Assuming a patch 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

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

According to Zoran and Weiss (2011), 'This equation has the familiar form of a likelihood term and a prior term, but note that $\mathrm{EPLL}_{\mathbb{P}}(U)$ 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 introducing auxiliary patch variables Z^i, $i = 1, \ldots, M$, one for each patch P_i, and minimizing 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

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

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 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

value $\tilde{U}(\mathbf{j})$ and of the patch denoised values $Z_i(\mathbf{j})$:

$$U(\mathbf{j}) = \frac{\lambda \tilde{U}(\mathbf{j}) + \sum_{P_i \ni \mathbf{j}} Z_i(\mathbf{j})}{\lambda + \beta k^2}, \tag{5.12}$$

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.

Once the Gaussian mixture model is 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, and 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 procedure shown in Algorithm 8.

Zoran and Weiss (2011) comment that this is one iteration of the 'hard version' of the EM algorithm for finding the modes of a Gaussian mixture (Carreira-Perpinan 2000). The method can be used for denoising, and several experiments seem to indicate that it equals the performance of BM3D and LLSC (Mairal *et al.* 2009*a*).

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

The basic idea of this algorithm is modelling a noiseless 'wavelet coefficient neighbourhood', P, by a *Gaussian scale mixture* (GSM), which is defined by

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

where U is a zero-mean Gaussian random vector and z is an independent positive scalar random variable. The wavelet coefficient neighbourhood

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

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 again adopt 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-squares* (BLS) estimator. For this reason, the method will be called here BLS-GSM (Bayes least-squares estimate of Gaussian scale mixture; Portilla *et al.* called it simply BLS). Without loss of generality it is assumed that $\mathbb{E}z = 1$, 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 an $n_1 \times n_2$ image, the pyramid $\mathcal{P}$ is generated in $\log_2(\min(n_1, n_2) - 4)$ 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). Using polar coordinates (r, θ) in the Fourier domain, the low-pass and high-pass isotropic filters are

$$l(r) = \begin{cases} 1 & 0 \le r < 0.5, \\ \cos(\frac{\pi}{2}(-\log_2 r - 1)) & 0.5 \le r < 1, \\ 0 & 1 \le r \le \sqrt{2}, \end{cases} \tag{5.13}$$

and

$$h(r) = \begin{cases} 0 & 0 \le r < 0.5, \\ \cos(\frac{\pi}{2}(\log_2 r)) & 0.5 \le r < 1, \\ 1 & 1 \le r \le \sqrt{2}. \end{cases} \tag{5.14}$$

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

$$a_k(r, \theta) = h(r) g_k(\theta), \quad k \in [0, K-1], \tag{5.15}$$

where $K = 8$, and

$$g_k(\theta) = \frac{(K-1)}{\sqrt{K[2(K-1)]}} \left[2 \cos\left(\theta - \frac{\pi k}{K} \right) \right]^{K-1}. \tag{5.16}$$

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 five-scale pyramid consisting of 49 subbands: eight high-pass oriented sub-bands, from $\mathcal{P}^1$ to $\mathcal{P}^8$, eight bandpass

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

oriented sub-bands for each scale, from $\mathcal{P}^9$ to $\mathcal{P}^{48}$, in addition to one low-pass non-oriented residual sub-band, $\mathcal{P}^{49}$. (Without loss of generality we shall keep this number 49 as a landmark, but the number of course depends on the image size.) Assume now that the image has been corrupted by independent additive Gaussian noise. Therefore, a typical neighbourhood of wavelet coefficients can be represented by

$$\tilde{P} = P + N = \sqrt{z}U + N, \tag{5.17}$$

where noise, N, and P are considered to be independent. Define $p_s(i,j)$ to be the sample at position (i,j) of the sub-band $\mathcal{P}^s$, the sub-bands being enumerated as $s = 1, \ldots, 49$, for example. The neighbourhood of the wavelet coefficient $p_s(i,j)$ is composed of its spatial neighbours for the same sub-band s. It could also have contained 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 neighbourhood is quite limited. Portilla *et al.* claim 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 neighbourhood 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 to this. First, the neighbourhood of coarsest-scale coefficients (without any coarser scale) necessarily has only nine surrounding coefficients. Second, the boundary coefficients are processed using special steps described below. Using the observed noisy vector, $\tilde{P}$, an estimate of P can be obtained by

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

This estimate 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 $\mathbb{E}(P \mid \tilde{P})$ is computed using

$$\mathbb{E}(P \mid \tilde{P}) = \sum_{i=1}^{13} \mathbb{P}(z_i \mid \tilde{P}) \mathbb{E}(P \mid \tilde{P}, z_i). \tag{5.18}$$

The only question left is how to compute the conditional probability and the conditional expectation, $\mathbb{P}(z_i \mid \tilde{P})$ and $\mathbb{E}(P \mid \tilde{P}, z_i)$. For each sub-band $\mathcal{P}^s$, except for 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 neighbourhood. If n^s denotes the size of neighbourhood at sub-band $\mathcal{P}^s$ (so n^s is 10 or 9, as explained above), $\mathbf{C}_N^s$ is an $n^s \times n^s$ matrix which can be experimentally generated by first decomposing a delta function

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

$\sigma\delta$ on the steerable pyramid. Here σ is the known noise variance and δ is an $n_1 \times n_2$ image defined by

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

(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 $\mathbf{N}_s$ to be the matrix which has as its rows all neighbourhoods 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 to eliminate the boundary coefficients. The covariance matrix $\mathbf{C}_N^s$ of the neighbourhood samples for each sub-band is computed using

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

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 (5.17) and the assumption $\mathbb{E}z = 1$, 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 its negative eigenvalues to zero. We can now calculate $\mathbb{E}(P \mid \tilde{P}, z_i)$. Using the fact that P and N are independent Gaussian random variables and also that the noise is additive, $\mathbb{E}(P \mid \tilde{P}, z_i)$ is simply a local Wiener estimate,

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

where the matrix fraction notation is understood to mean $\frac{\mathbf{C}}{\mathbf{W}} := \mathbf{C}\mathbf{W}^{-1}$. Clearly it would be cumbersome to compute as many matrix inversions as z_is. Fortunately, with a bit of linear algebra this computation can be rendered common to all z_i. Let $\{\mathbf{Q}, \mathbf{\Lambda}\}$ be 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 $\mathbf{M} = \mathbf{S}\mathbf{Q}$, $\mathbf{v} = \mathbf{M}^{-1}\tilde{P}$. Then we have

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

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

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

The interesting point is that one can calculate $\mathbf{M}$, $\mathbf{\Lambda}$ and $\mathbf{v}$ once for each sub-band. The scalar final formulation of the above equation is

$$\mathbb{E}(P \mid \tilde{P}, z_i)_c = \sum_{j=1}^{n^s} \frac{z_i m_{c,j}\lambda_{j,j}v_j}{z_i\lambda_{j,j}+1}, \tag{5.19}$$

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 neighbourhood.

The second component of (5.18) is $\mathbb{P}(z_i \mid \tilde{P})$, which can be obtained using Bayes' rule. Here $p_z(z)$ denotes the density function of the random variable z,

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

or its discrete form,

$$\mathbb{P}(z_i \mid \tilde{P}) = \frac{\mathbb{P}(\tilde{P} \mid z_i)p_z(z_i)}{\sum_{j=1}^{13} \mathbb{P}(\tilde{P} \mid z_j)p_z(z_j)}, \tag{5.20}$$

where the density of the observed noisy neighbourhood 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

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

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

$$\mathbb{P}(\tilde{P} \mid z_i) = \frac{\mathrm{e}^{-\frac{1}{2}\sum_{j=1}^{n^s}\frac{v_j^2}{z_j\lambda_{j,j}+1}}}{\sqrt{\Pi_{j=1}^{n^s}(z_i\lambda_{j,j}+1)}}, \tag{5.21}$$

The only question left is the form of $p_z(z)$. Portilla *et al.* (2003), after a somewhat puzzling discussion, adopt a 'non-informative Jeffreys prior', $p_z(z) \simeq \frac{1}{z}$. Since this function cannot be a density, being non-integrable, the function is cut off to zero near $z = 0$. The algorithm of Portilla *et al.* (2003) is given here as Algorithm 9.

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

Algorithm 9 Portilla *et al.* wavelet neighbourhood 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 neighbourhood coefficient (10 or 9).

Apply the wavelet pyramid (5.13)–(5.16), respectively, to the noise image δ and to the observed image.
Regroup the obtained wavelet coefficients to obtain $\tilde{P}^s$, the wavelet coefficient neighbourhoods of rank s and N^s the noise wavelet coefficient neighbourhoods of rank s.
for each filter index s **do**
Compute $\mathbf{C}_N^s$ and $\mathbf{C}_{\tilde{P}}^s$, the noise and observation covariance matrices of N_s and $\tilde{P}_s$. (The subscript s is omitted hereafter.) 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 neighbourhood $\tilde{P}$ and $i \in \{1, \ldots, 13\}$ **do**
Compute $\mathbf{M} = \mathbf{S}\mathbf{Q}$, $\mathbf{v} = \mathbf{M}^{-1}\tilde{P}$.
Using (5.19) obtain $\mathbb{E}(P \mid \tilde{P}, z_i)_c = \sum_{j=1}^{n^s} \frac{z_i m_{c,j} \lambda_{j,j} v_j}{z_i \lambda_{j,j} + 1}$, 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 neighbourhood.
Apply (5.20) to get

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

using the value obtained by (5.21) for

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

By (5.18) finally obtain $\mathbb{E}(P \mid \tilde{P}) = \sum_{i=1}^{13} \mathbb{P}(z_i \mid \tilde{P}) \mathbb{E}(P \mid \tilde{P}, z_i)$, 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 neighbourhood coefficients $\mathbb{E}(P \mid \tilde{P})$ by the inverse steerable pyramid.

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

As we shall see in Section 7, in spite of its formalism, this method is in fact extremely similar to other patch-based Bayesian methods. It has received a more recent extension, reaching state-of-the-art performance, in Lyu and Simoncelli (2009): this paper proposes an extension of the above method modelling the wavelet coefficients as a global random field of Gaussian scale mixtures.

5.7. K-SVD

The K-SVD method was introduced by Aharon, Elad and Bruckstein (2006), whose whole objective was to optimize the quality of sparse approximations of vectors in a learnt dictionary. Although they noted the relevance of the technique to image processing tasks, it was Elad and Aharon (2006) who made a detailed study of the denoising of grey-scale images. Since then, the adjustment to colour images has been treated by Mairal *et al.* (2008). Let us note that Mairal *et al.* (2008) 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 Mairal (2010) and Mairal *et al.* (2009*b*).

The algorithm is divided into three steps. In the two first steps an optimal dictionary and a sparse representation are 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 the usual orthogonal basis (*e.g.*, the discrete cosine transform, or 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 $\kappa^2 \times n_{\text{dict}}$, with $n_{\text{dict}} \geq \kappa^2$, whose columns (the 'atoms of the dictionary') are normalized in the 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_{dict}) 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

$$\arg\min_{\alpha_{\mathbf{i}}} \|\alpha_{\mathbf{i}}\|_0 \quad \text{such that} \quad \|\mathbf{R_i}\tilde{U} - \mathbf{D}\alpha_{\mathbf{i}}\|_2^2 \leq \kappa^2 (C\sigma)^2, \tag{5.22}$$

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

Algorithm 10 K-SVD algorithm for grey-scale images

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

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

for k=1 to K **do**

An ORMP is applied to the vectors $\mathbf{R_i}\tilde{U}$ in such a 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 residual

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

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:

$$(\hat{d}_l, \hat{\alpha}^l) = \arg\min_{d_l, \alpha^l} \|\mathbf{E}_l - d_l\alpha^l\|_F^2.$$

A truncated SVD is applied to the matrix $\mathbf{E}_l$. It partially provides $\mathbf{U}$, $\mathbf{V}$ (orthogonal matrices) and $\mathbf{\Delta}$ (filled in with zeros 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

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

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 a 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

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

approximation. The goal is to decrease the quantity

$$\sum_{\mathbf{i}} \|\mathbf{D}\alpha_{\mathbf{i}} - \mathbf{R}_{\mathbf{i}}\tilde{U}\|_2^2 \tag{5.23}$$

while keeping the sparsity of the vectors $\alpha_{\mathbf{i}}$. We will denote by $\hat{d}_l$ ($1 \leq l \leq n_{\text{dict}}$) the columns of the dictionary $\hat{\mathbf{D}}$. First, the quantity (5.23) 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 residual

$$e_{\mathbf{i}}^l = \mathbf{R}_{\mathbf{i}}\tilde{U} - \hat{\mathbf{D}}\hat{\alpha}_{\mathbf{i}} + \hat{d}_l\hat{\alpha}_{\mathbf{i}}(l), \tag{5.24}$$

which is the error committed by deciding to omit $\hat{d}_l$ from the representation of the patch $R_{\mathbf{i}}\tilde{U}$. Thus $e_{\mathbf{i}}^l$ is a vector of size κ^2.

These residuals 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

$$\sum_{\mathbf{i}} \|\hat{\mathbf{D}}\hat{\alpha}_{\mathbf{i}} - \hat{d}_l\hat{\alpha}_{\mathbf{i}}(l) + d_l\alpha^l - \mathbf{R}_{\mathbf{i}}\tilde{U}\|_2^2 = \|\mathbf{E}_l - d_l\alpha^l\|_F^2, \tag{5.25}$$

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 it is plausible that minimizing (5.25) amounts to reducing the approximation error caused by $\hat{d}_l$. It is well known that the minimization of such a Frobenius norm consists of a rank-one approximation, which always admits a solution, obtained in practice using the SVD. Using the SVD of $\mathbf{E}_l$,

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

where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices and $\boldsymbol{\Delta}$ is a diagonal matrix whose diagonal elements are 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 $\boldsymbol{\Delta}(1,1)$.

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

$$\hat{U} = \arg\min_{U_0 \in \mathbb{R}^M} \lambda\|U_0 - \tilde{U}\|_2^2 + \sum_{\mathbf{i}} \|\hat{\mathbf{D}}\hat{\alpha}_{\mathbf{i}} - \mathbf{R}_{\mathbf{i}}U_0\|_2^2$$

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

using the closed formula

$$\hat{U} = \left(\lambda \mathbf{I} + \sum_{\mathbf{i}} \mathbf{R}_{\mathbf{i}}^T \mathbf{R}_{\mathbf{i}}\right)^{-1} \left(\lambda \tilde{U} + \sum_{\mathbf{i}} \mathbf{R}_{\mathbf{i}}^T \hat{\mathbf{D}} \hat{\alpha}_{\mathbf{i}}\right). \tag{5.26}$$

For each pixel this amounts to averaging its initial noisy value with the average of all estimates obtained with all patches containing it. The parameter λ controls the trade-off between these two values and thus measures the fidelity to the initial noisy image.

Mairal *et al.* (2008) proposed direct extension of 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 to 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 describe the algorithm for grey-scale images: the colour version simply requires an adaptation of the Euclidean norm to the colour space.

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 from DCT denoising is that a set of similar patches are used to form a three-dimensional block, which is filtered by using a three-dimensional transform, hence the name *collaborative filtering*. The method has four steps: (a) finding the image patches similar to a given image patch and grouping them in a three-dimensional block, (b) three-dimensional linear transform of the three-dimensional block, (c) shrinkage of the transform spectrum coefficients, and (d) inverse three-dimensional transformation. This three-dimensional filter therefore filters out simultaneously all two-dimensional image patches in the three-dimensional 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 of 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, as described in Section 4. The second difference is that the new three-dimensional 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

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

Algorithm 11 BM3D first iteration algorithm for grey-scale images

Input. Noisy image $\tilde{u}$, noise standard deviation σ.
Output. Basic estimate $\hat{u}_1$ 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 for similar patches.
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 neighbourhood 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 biorthogonal spline wavelet (Bior 1.5) is applied on every patch 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 biorthogonal 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**, using the weights $\omega_{\tilde{P}}$.

end for

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

Algorithm 12 BM3D second iteration algorithm for grey-scale images

Input. Noisy image $\tilde{u}$; noise standard deviation σ; basic estimate $\hat{u}_1$ 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 for similar patches.
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 $\hat{P}_1$ centred at **i**, of size $\kappa \times \kappa$ in the initial and basic estimation images.

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

Select the number of similar patches as done in the first step.

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 3D transform (denoted by τ_{3D}) is applied on both 3D groups:

- First a 2D DCT is applied on every patch contained in $\mathcal{P}(\tilde{P})$ and $\mathcal{P}(\hat{P}_1)$;
- Then a Walsh–Hadamard transform is applied along the third dimension of $\mathcal{P}(\tilde{P})$ and $\mathcal{P}(\hat{P}_1)$.

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

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

denoised image obtained at the first step to approximate the true coefficients given by Theorem 3.1. The final aggregation step is identical to that of the first step.

The algorithm is extended to colour images via 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.

Here we have described the basic implementation given in the original paper, and which will also be used in Section 6. However, BM3D has several more recent variants that improve its performance. As for NL-means, there is a variant with shape-adaptive patches (Dabov, Foi, Katkovnik and Egiazarian 2009). In this algorithm, called BM3D-SAPCA, the sparsity of image representation is improved in two respects. First, it employs image patches (neighbourhoods) which can have a 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 neighbourhoods. 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 (Katkovnik, Egiazarian and Astola 2006).

The very recent development of BM3D is presented in Katkovnik, Danielyan and Egiazarian (2011) and Danielyan, Katkovnik and Egiazarian (2012), 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 by Yu, Sapiro and Mallat (2010, 2012) 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 only studying the denoising problem, we shall take $\mathbf{A}_i$ to be the identity. The (straightforward) extension including a linear perturbation operator is beyond our scope.

The patch density law is modelled 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 to be independently drawn from one of these Gaussians with an unknown index k and the density function

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

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

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

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

- estimating the Gaussian parameters $(\mu_k, \mathbf{C}_k)_{1 \le k \le K}$ from the degraded data $\tilde{P}_i$,
- identifying the index k_i of the Gaussian distribution generating the patch P_i,
- estimating 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 (Yu *et al.* 2012) 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

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

$\hat{P}_i^k$ 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 to be known for all patches i, and again permit estimation of the Gaussian models $(\mu_k, \mathbf{C}_k)_{1\le k\le K}$. According to the terminology of Section 4.2, this subsection gives the *oracle* allowing us to estimate the patches by a Wiener-type filter in the E-step.

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

$$\begin{aligned}(\hat{P}_i, k_i) &= \arg\max_{P,k} \log \mathbb{P}(P_i \mid \tilde{P}_i, \mathbf{C}_k) \\ &= \arg\max_{P,k} \big(\log \mathbb{P}(\tilde{P}_i \mid P_i, \mathbf{C}_k) + \log \mathbb{P}(P_i \mid \mathbf{C}_k)\big) \\ &= \arg\min_{P_i,k} \big(\|P_i - \tilde{P}_i\|^2 + \sigma^2 (P_i - \mu_k)^T \mathbf{C}_k^{-1} (P_i - \mu_k) + \sigma^2 \log|\mathbf{C}_k|\big)\end{aligned} \tag{5.27}$$

where the second equation follows from Bayes' rule and the third one assumes white Gaussian noise with diagonal covariance 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

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

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 (5.2) already seen in Section 5.2. Then the best Gaussian model k_i is selected as

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

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

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

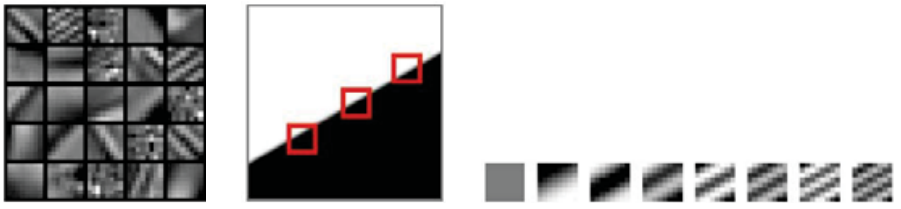
This yields the empirical estimate

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

which are the estimates (5.3) also used in Section 5.2.

Finally, the above MAP-EM algorithm is iterated and Yu *et al.* observe

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press



(a) (b) (c)

Figure 5.6. From Yu *et al.* (2012). (a) Typical dictionary atoms learned from the classic image Lena with K-SVD. (b, c) 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).

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. Noting that having the adequate Gaussians describing the patch space amounts to having a good set of PCA bases for intuitive patch clusters, Yu *et al.* 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 into 18 classes of patches containing an edge or an oriented texture and one class containing the more isotropic patches.

The study by Yu *et al.* (2012) gives an interpretation of the patch dictionary methods such as K-SVD and fuses them with Bayesian methods and the Wiener method. In particular, Yu *et al.* show how the K-SVD method actually learns patches that are quite similar to oriented patches obtained by the above procedure, as illustrated in Figure 5.6. This analysis provides the framework for the synthesis 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 neighbourhood Gaussian scale mixture (BLS-GSM) algorithm as specified in Algorithm 9,
- the classical vector-valued NL-means as specified in Algorithm 4,

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

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

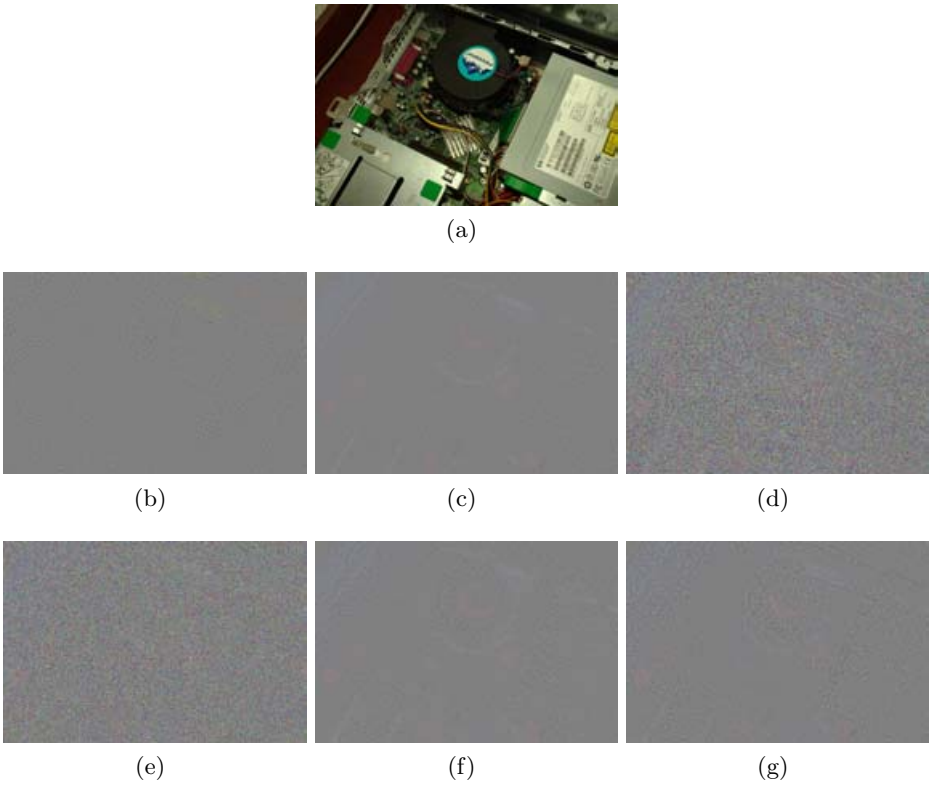
The comparison, using public-domain IPOL algorithms when possible, will be based on four quantitative and qualitative criteria: visualization of the *method noise*, namely the part of the image that the algorithm has taken out as noise; the visual verification of the *noise-to-noise* principle; the *mean square error* or *PSNR* tables; and last but but not least, the *visual quality* of the restored images, which 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 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. The procedure was introduced by Buades, Coll and Morel (2005*a*), who called this difference *method noise*. They pointed out that the method noise should look like a noise, at least in the 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, such as Gaussian convolution, anisotropic filters, neighbourhood filters or wavelet thresholding algorithms, a closed formula permits mathematical analysis of the method noise, and thus gives an explanation of the observed structure of image differences when applying the denoising method (Buades *et al.* 2005*b*). Such an analysis is unfortunately not available, and not easy for the state-of-the-art algorithms 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 white noise with standard deviation larger than 5 or 10 has been added to the original.

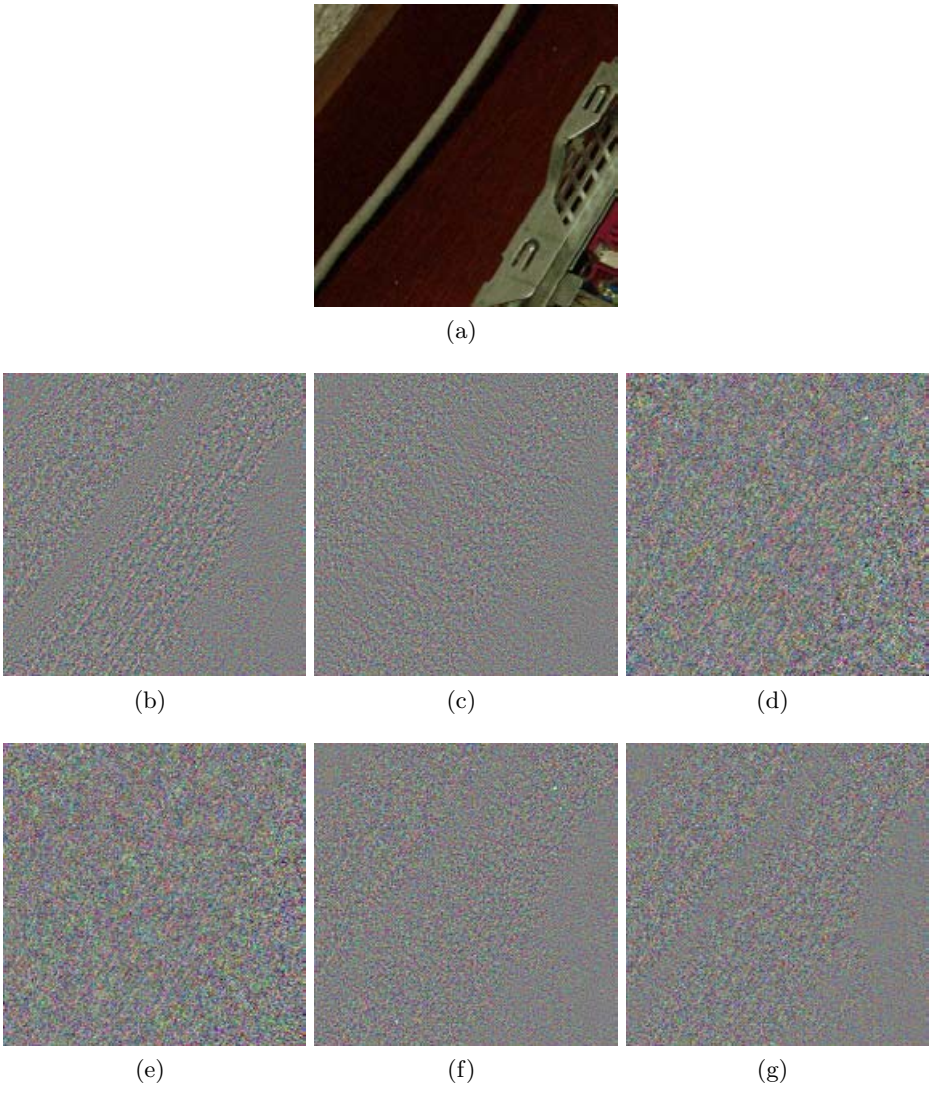
https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press



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

Figure 6.1. Method noise. The noisy image was obtained by adding Gaussian white noise of standard deviation 5. (a) Slightly noisy image, (b) DCT sliding window (StD = 4.69), (c) BLS-GSM (StD = 4.28), (d) NL-means (StD = 5.78), (e) K-SVD (StD = 5.67), (f) BM3D (StD = 4.25) and (g) non-local Bayes (StD = 4.28). All methods have a difference similar to white noise even if the magnitudes of the NL-means and K-SVD differences are 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 make little change to the coefficients larger than those predicted by noise, noise is not removed in textured and edge zones. This can be easily seen in Figure 6.2, where a piece of the residual noise has been enlarged.

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press



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

Figure 6.2. Enlargement of the method noise difference of Figure 6.1. (a) Slightly noisy image, and the method noise for (b) DCT sliding window, (c) BLS-GSM, (d) NL-means, (e) K-SVD, (f) BM3D and (g) 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.

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

Figure 6.1 displays the method noise for the state-of-the-art algorithms compared in this section, when 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. In a first visual inspection, we see that all methods have a difference similar to white noise. This is an outstanding property of these algorithms, which is not shared by classical denoising techniques such as anisotropic filtering, total variation minimization or wavelet thresholding (see Buades *et al.* (2005*b*) for a more detailed study). We also see immediately 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 Figure 6.1), which is $\simeq 5.7$ for NL-means and K-SVD, $\simeq 4.7$ for DCT denoising, and $\simeq 4.25$ for the other algorithms. DCT-denoising, BLS-GSM, BM3D and non-local Bayes keep the transform coefficients that are larger than those predicted by noise. This explains why they remove little noise in textured or edge regions. This fact can be easily seen in Figure 6.2, where a piece of the residual noise of Figure 6.1 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 by Buades *et al.* (2008*b*), requires a denoising algorithm to transform white noise into white noise. This paradoxical requirement seems to be the best way to characterize artifact-free algorithms. The transformation of white noise into any correlated signal creates structure and artifacts. Only white noise is perceptually devoid of structure, as was pointed out by Attneave (1954).

The noise-to-noise of classical denoising algorithms was studied by Buades *et al.* (2008*b*), who showed that neighbourhood filters and, asymptotically, NL-means transform white noise into 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 longer 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 superimposed wavelets or cosines if the DCT is used. The mathematical analysis of the rest of the 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 exhibit it is as follows.

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

Table 6.1.

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

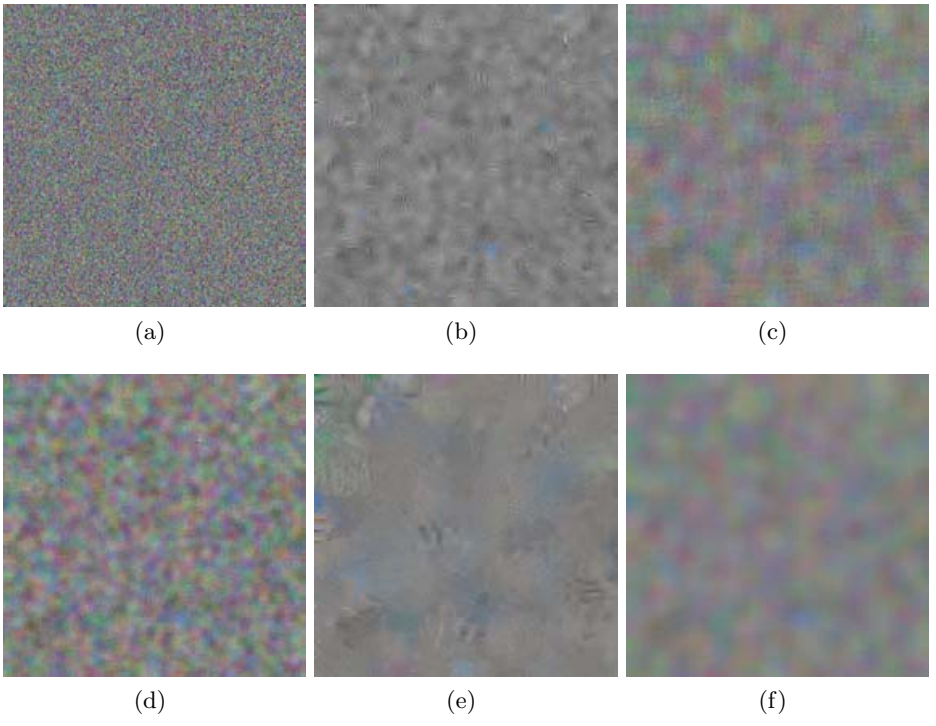


Figure 6.3. The noise-to-noise principle: a three-channel colour noise image filtered by the state-of-the-art methods. (a) The noise image (flat, with independent homoscedastic noise added on each channel). Then, the same image denoised by (b) DCT sliding window, (c) NL-means, (d) K-SVD, (e) BM3D and (f) non-local Bayes. The more the denoised image of a noise image looks like a noise image the better. Indeed, structured noise creates artifacts. BLS-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 a multiscale version could cope with the low-frequency remaining noise.

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

- 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 of slightly different values:[4] $RGB = (127, 128, 129)$.
- To reduce the variations due to the random nature of the noise, the tests are performed on relatively large noise images. The chosen size is 1024×1024. The PSNR and RMSE results then become fairly independent of the simulated noise.
- Noise is added to each channel independently. It is therefore 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 of 5. A small part with size 256×256 of the denoised image is shown in close-up in Figure 6.3.

The results in PSNR and RMSE are summarized in Table 6.1. The 'order' of performance of the methods is almost respected, except for TV denoising, which shows a really good result compared to K-SVD. Figure 6.3 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 non-local Bayes has no artifacts. Indeed, it detects flat patches and replaces them with their mean. This trick could in fact be applied to all algorithms. Last but not least, each method leaves a sizeable low-frequency noise, which could be removed with a multiscale 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 control of the absence of artifacts and the correct reconstruction of edges, texture and fine structure. Figures 6.5–6.7 display the noisy and denoised images for the algorithms under comparison for noise standard deviations of 20, 30 and 40.

Figure 6.5 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

[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-scale image with a single channel.

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press



(a) alley (b) computer (c) dice



(d) flowers (e) girl (f) traffic



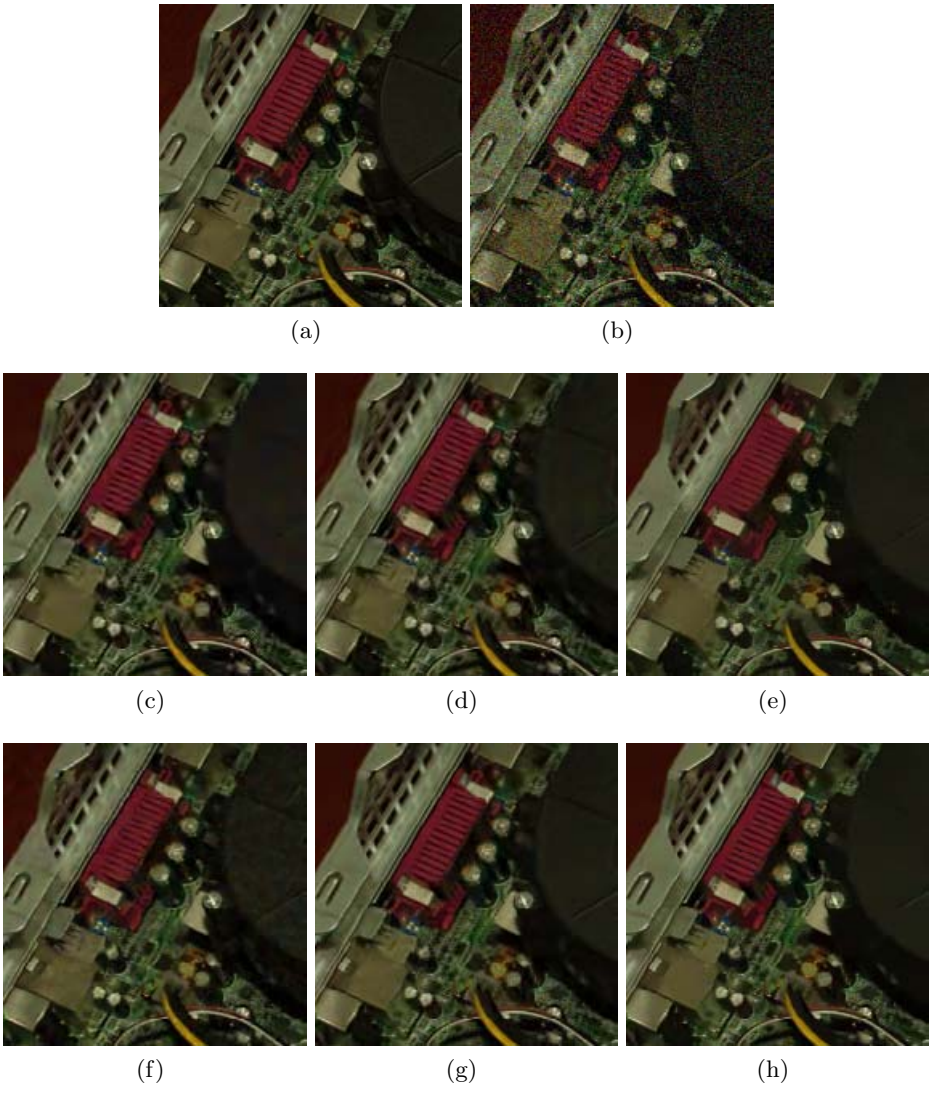
(g) trees (h) Valldemossa

Figure 6.4. A set of noiseless images used for the comparison tests.

most local algorithms and therefore have more trouble 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 though each uses 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 from 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 experiment, BM3D and non-local Bayes give similar performance, superior to the rest of algorithms.

Figures 6.6 and 6.7 again illustrate the low-frequency colour artifacts of DCT, BLS-GSM and K-SVD. In these figures, DCT and BLS-GSM also suffer from a strong Gibbs effect near all image boundaries. This Gibbs effect is nearly unnoticeable in the denoised image by K-SVD, since the use of the whole dictionary permits better reconstruction of edges when the right atoms are present in the dictionary. The image denoised by NL-means

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press



(a) (b)

(c) (d) (e)

(f) (g) (h)

Figure 6.5. Comparison of visual quality. The noisy image was obtained by adding Gaussian white noise of standard deviation 20. (a) Original, (b) noisy, (c) DCT sliding window, (d) BLS-GSM, (e) NL-means, (f) K-SVD, (g) BM3D and (h) non-local Bayes.

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

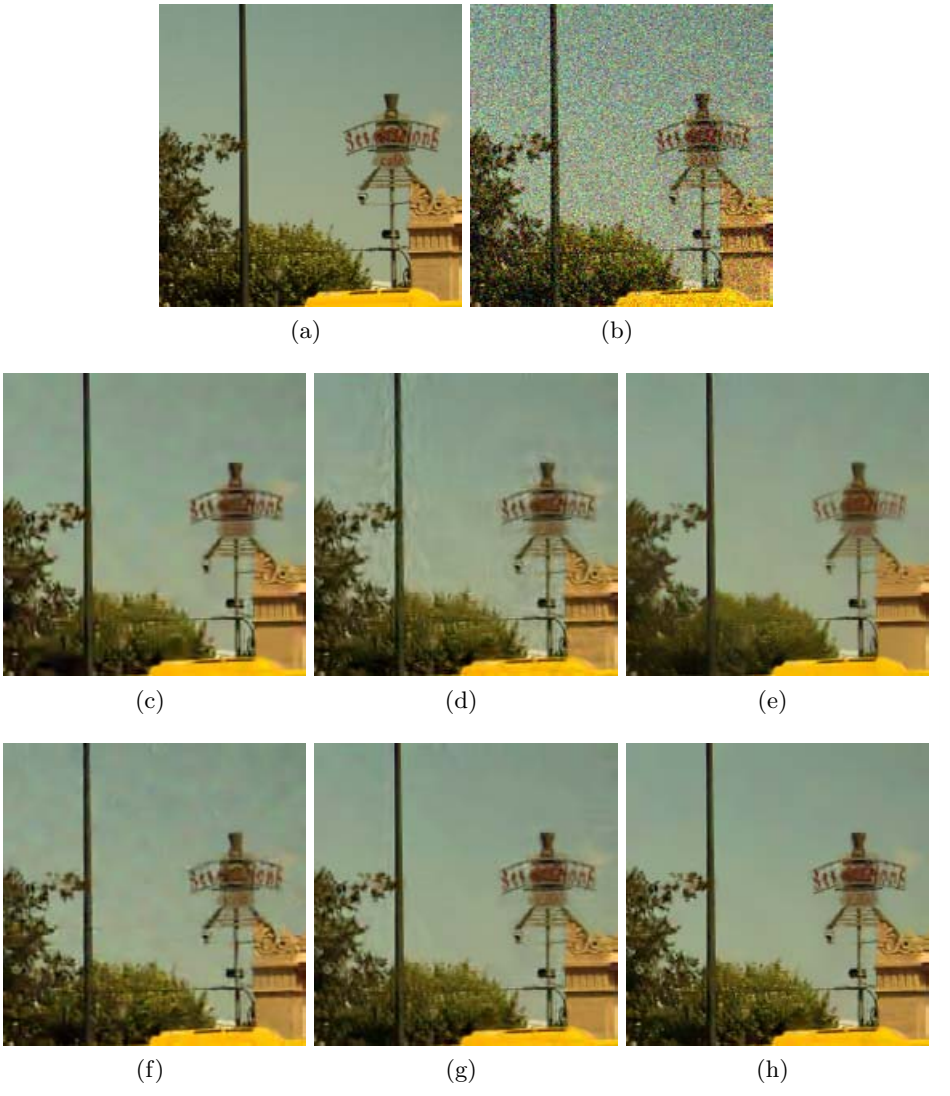
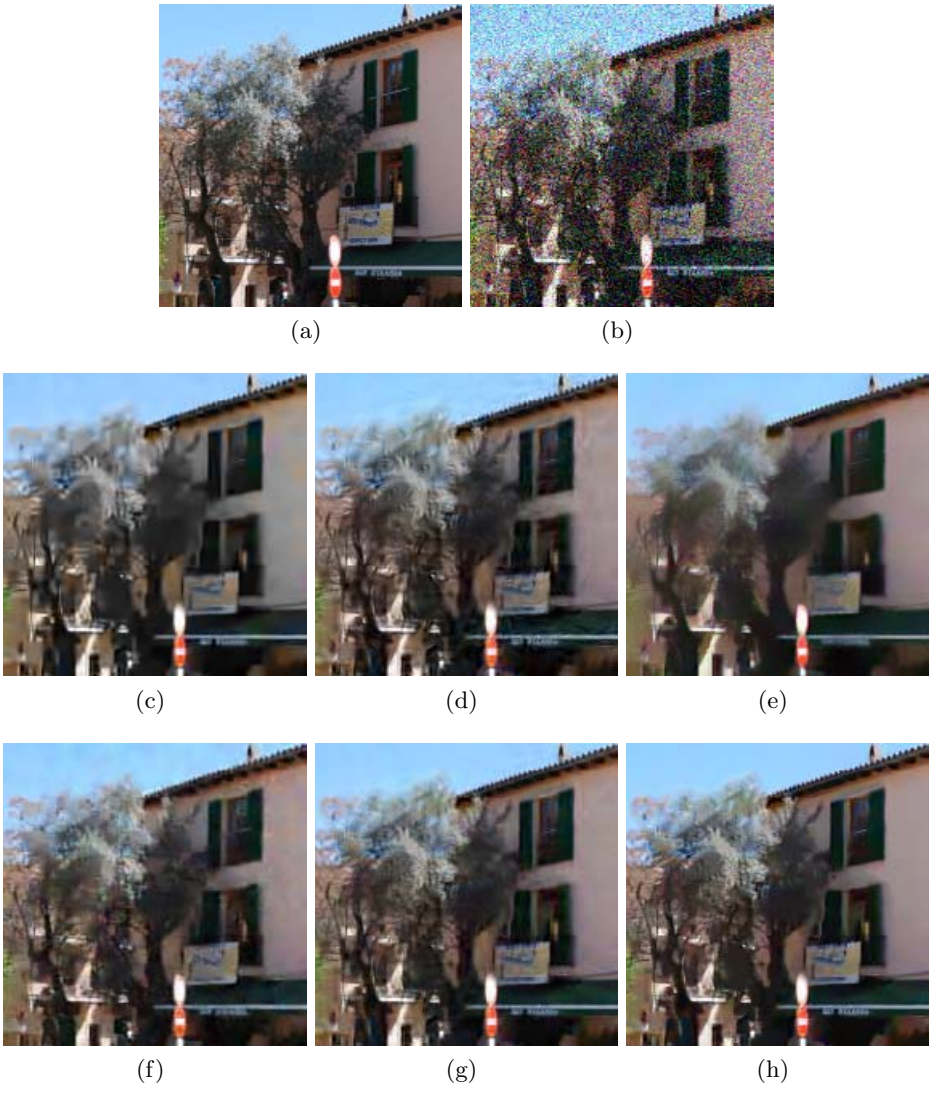


Figure 6.6. Comparison of visual quality. The noisy image was obtained by adding Gaussian white noise of standard deviation 30. (a) Original, (b) noisy, (c) DCT sliding window, (d) BLS-GSM, (e) NL-means, (f) K-SVD, (g) BM3D and (h) non-local Bayes.

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press



(a) (b)

(c) (d) (e)

(f) (g) (h)

Figure 6.7. Comparison of visual quality. The noisy image was obtained by adding Gaussian white noise of standard deviation 40. (a) Original, (b) noisy, (c) DCT sliding window, (d) BLS-GSM, (e) NL-means, (f) K-SVD, (g) BM3D and (h) non-local Bayes.

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

has no visual artifacts but is more blurred than those given by BM3D and non-local Bayes, which clearly have superior performance to the rest of the algorithms. The image denoised by BM3D has some Gibbs effect near edges, which sometimes degrades the visual quality of the solution. The non-local Bayes image shows no artifacts. It often preserves better textures than BM3D, in which 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 non-local 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:

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

These numerical quality measurements are the most objective, since they do not rely on any visual interpretation. Tables 6.2 and 6.3 display the PSNR of state-of-the-art denoising methods using the images in Figure 6.4 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 merely 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 very close, the way generic tools are implemented, and the degree of sophistication with which each principle is implemented, do matter. For example, two of our readers, Alessandro Foi and Vladimir Katkovnik,[5] have pointed out to us that an experimental analysis carried out exclusively on colour images does not permit a comparison between the different strategies devised to take advantage of spatial redundancy. They suggest complementing the denoising results on colour images with experiments on grey-scale images. Then it would be possible to: (1) compare the degree of success of these different denoising principles in

[5] Personal communication.

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

Table 6.2. 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.

	NL-Bayes	BM3D	BLS-GSM	K-SVD	NL-means	DCT denoising
				$\sigma = 2$		
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
				$\sigma = 5$		
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
				$\sigma = 10$		
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

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

Table 6.3. PSNR table for $\sigma = 20$, 30 and 40.

	NL-Bayes	BM3D	BLS-GSM	K-SVD	NL-means	DCT denoising
				$\sigma = 20$		
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
				$\sigma = 30$		
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
				$\sigma = 40$		
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

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

exploiting spatial redundancy, and (2) evaluate the effectiveness of the various ways in which these grey-scale algorithms are extended to colour data.

In short, Foi and Katkovnik do not share our analysis or our conclusions drawn from the experimental results, because these results are very much influenced by the way colour data are treated, while many of the conclusions are applied concerning the relative effectiveness in exploiting spatial redundancy.

For the same reasons, they also disagree with the taxonomy summarized in Table 7.1 (page 564), where it seems that the extension to colour is to be considered as a feature of a particular algorithm. Some methods are applied to colour data in a very simple non-adaptive way and thus cannot be expected to fully decorrelate the colour channels. For instance, this is the case for BM3D, which uses a YUV/Opp colour transformation. Data-adaptive colour transformations for multispectral data are considered in Danielyan, Foi, Katkovnik and Egiazarian (2010). This adaptive method provides substantially better results than a standard colour transformation.

Another reason for being cautious is that all methods in existence do in fact have variants, and we are using the basic algorithms as stated in their original papers. For example, it is shown in Hou, Zhao, Yang and Cheng (2011) that BM3D can be slightly improved for heavy noise > 40 by changing the method parameters.

In other words, the following PSNR comparison of colour images must be taken 'as 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, BLS-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 poor visual quality compared to NL-means. In all cases, BM3D and non-local Bayes have better PSNR performance than the others. Because of superior noise reduction in flat zones and the presence of fewer 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 artifacts near boundaries can probably be eliminated by the same trick as for non-local Bayes, namely detecting and giving a special treatment to flat three-dimensional 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 neighbourhood, they give several evaluations for each pixel. Thus, they all use an aggregation step. There is

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

only one method for which the aggregation is not explicitly stated as such, the wavelet neighbourhood (BLS-GSM) algorithm. Nevertheless, closer examination shows that it denoises not just one but some 49 wavelet channels for a 512×512 image. The applied 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 neighbourhood' contains a 3×3 patch of a wavelet channel, complemented by 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 initially looks different from the other algorithms, it relies on similar principles: it estimates patch models to denoise them, and aggregates the results. But it is also 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.

Why is its performance 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 about a PSNR improvement. Portilla *et al.* (2003) do not specify whether there is an aggregation step, but a first aggregation step is possible (the second aggregation being implicit in the reconstruction step from all redundant channels). 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 7.1 shows a synopsis of the ten methods that have been thoroughly discussed. The classification criteria are as follows.

The method's denoising principle

Our task here is to show that, in spite of the different language used for each method, the underlying principles are, in fact, very similar. The dominant principle is to compute a linear minimum least-squares estimator (LMMSE) after building a Bayesian patch model. As a matter of fact, even if this is not always explicit, *all* methods follow the same LMMSE estimator principle very closely. For example, the DCT threshold is simply a Wiener thresholding version of the Bayesian LMMSE. This threshold is used because the DCT of the underlying noiseless image is unknown. The same argument

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

applies for NL-means, which was interpreted as an LMMSE in Section 5.1. A close examination of K-SVD can convince the user that this algorithm is very close to PLE, 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, PLE, PLOW and EPLL actually deliver a dictionary, which is the union of several orthogonal bases of patches. For each noisy patch, PLE, PLOW and EPLL select one of the bases, for which the patch will be sparse. In short, like K-SVD, they compute a sparse representation for each patch in an overcomplete dictionary. In this argument, we follow the simple and intelligent interpretation proposed for the PLE method by Yu *et al.* (2010, 2012), who summarize their method 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 learned 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 overcomplete 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, each learned from some 64 patches. Thus, the open question is: How many clusters and how many learning patches are truly necessary to obtain the best PSNR? The disparity between these figures is certainly too large to be realistic.

Finally, we must wonder if transform thresholding methods fit into the united view of all algorithms. The Bayesian–Gaussian estimate used by most algorithms mentioned can be interpreted as a Wiener filter on the eigenvector basis of the Gaussian. It sometimes includes 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 three-dimensional block obtained by grouping similar patches. Clearly, it has found, by a

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

Table 7.1. Synoptic table of all methods considered.

Method	Denoising principle	Patches	Size	Aggr.	Oracle	Col.
DCT	transform threshold	one	8	yes	yes	yes
NL-means	average	neighbourhood	3	yes	yes	no
NL-Bayes	Bayes	neighbourhood	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	neighbourhood	8–12	yes	yes	yes
PLE	Bayes, 19 clusters	image	8	yes	yes	yes

rather systematic exploration, the right two-dimensional 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. NL-means, non-local Bayes, shotgun-NL, and BM3D denoise a patch after comparing it with 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, globally process the 'patch space' and construct patch models. Nevertheless, this difference is easily reduced. Indeed, PLE, 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 with each other, but they are compared with the clusters. Similarly, dictionary-based methods such as K-SVD propose over-complete dictionaries learned 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 classifying this patch. This is suggested by Yu *et al.* (2012), for PLE: 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 modelling is elaborated in Zhou *et al.* (2011).

Patches

The 'Patches' column in Table 7.1 indicates the number of patches used for the denoising method, and where they are found. Trivial DCT uses only the current patch to denoise it. NL-means, non-local Bayes and BM3D compare the reference patch with a few hundred patches in its spatial neighbourhood.

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

PLE, PLOW, BLS-GSM and K-SVD compare each noisy patch to a learned model of all image patches. Finally, shotgun-NL and EPLL involve a virtually infinite number of patches in the estimation. Surprisingly enough, the performance of all methods is relatively similar. Thus, the huge numbers used to denoise in shotgun-NL and EPLL clearly depend on the fact that the patches were not learned from the image itself, and their number can arguably be considerably reduced.

Size (*of patches*)
The 'Size' column compares the patch sizes. All methods without exception try to deduce the correct value of a given pixel $\mathbf{i}$ by using a neighbourhood of $\mathbf{i}$ called a 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 of 8×8 can be insufficient, 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). Little work has been done 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 in 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 via aggregation and via the oracle strategy.

Algorithms compared by complexity and information
Current research focuses on obtaining the best possible results, and perhaps the optimal results. We have followed this path and have completely ignored complexity issues in this comparison. For example, the 'shotgun' patch methods are not reproducible in an acceptable time. But 'all is fair in love and war'. The question of how to obtain the best acceptable results must be solved first, by every possible means, before fast algorithms are devised. On the other hand, 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 On Line*. At least two of them give state-of-the-art results. Thus, we hold the view that complexity is not a central issue in the current debate. Another question that emerged in this study is the *amount of information needed to*

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

achieve optimal denoising. Here, we have observed that the methods divide into two groups. The simplest one (DCT denoising) uses only one image patch and gets results only 1 dB from optimal results. The classic non-local methods only use a larger neighbourhood of a given pixel, in spite of their 'non-local' epithet. Then, an intermediate class of methods uses all image patches simultaneously. The shotgun methods use virtually all possible existing image patches. 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

Most patch-based image denoising methods follow one paradigm: they unite the transform thresholding method and a Markovian–Bayesian estimation. This unification is complete when the patch space is assumed to be a Gaussian mixture. Each Gaussian is associated with its orthonormal basis of patch eigenvectors. Thus, transform thresholding (or a Wiener filter) is applied to these local orthogonal bases.

This method seems to be almost optimal. It yields satisfactory results for an interval of standard deviations ranging from 5 to 40. (These figures are valid for current image formats, with a range of $[0, 255]$.) Small noise (below 5) and large noise (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, such as photogrammetry and stereovision, the precision varies inversely with the signal-to-noise ratio. Thus, even with good-quality stereo pairs, it is relevant to remove even more noise. For high noise, near-optimal denoising can be obtained by applying a global Wiener filter using the second-order statistics of the image. Nevertheless, all existing filters produce too many artifacts for large noise.

The multiscale aspect of denoising has been explored only on three dyadic scales (since most patch methods use 8×8 patches). It may be insufficient. The success of denoising methods suggests that the statistical exploration of images has been advancing slowly. The exploration of the huge 'patch space' is only starting. Its structure remains largely unknown, and we know little of its geometry. Its representation by a sum of Gaussians, or by a Gaussian scale mixture, is only a first rough approximation.

Are image denoising algorithms close to achieving their optimal bounds? For the ranges of noise that we have tested, the image visual improvement obtained by state-of-the-art denoising methods is undeniable and sometimes spectacular. For movies, which are much more redundant, this effect is even more impressive. Can we deduce from the arguments developed by Levin and Nadler (2011) and Chatterjee and Milanfar (2010) that the cur-

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

rent methods are close to optimality? In these papers optimal bounds for *all* patch-based methods are proposed. Nevertheless, a closer examination shows that existing methods are probably further away from optimality than is understood. Indeed, all state-of-the-art patch-based methods use the aggregation step, which doubles the size of the neighbourhood effectively used. Thus, in a fair comparison, the shotgun Bayesian estimate (Levin and Nadler 2011) should use 16×16 patches. We might be facing the curse of dimensionality.

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 into the text. Research was partly financed by the MISS project of Centre National d'Etudes Spatiales, the US Office of Naval Research under grant N00014-97-1-0839, and by the European Research Council's 'Twelve Labours' advanced grant.

REFERENCES[6]

A. Adams, N. Gelfand, J. Dolson and M. Levoy (2009), Gaussian KD-trees for fast high-dimensional filtering. In *ACM Transactions on Graphics: TOG*, Vol. 28, ACM, p. 21.

M. Aharon, M. Elad and A. M. Bruckstein (2006), 'K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation', *IEEE Trans. Signal Processing* **54**, 4311–4322.

F. Anscombe (1948), 'The transformation of Poisson, binomial and negative-binomial data', *Biometrika* **35**, 246–254.

M. Antonini, M. Barlaud, P. Mathieu and I. Daubechies (1992), 'Image coding using wavelet transform', *IEEE Trans. Image Processing* **1**, 205–220.

P. Arias, V. Caselles and G. Sapiro (2009), A variational framework for non-local image inpainting. In *Proc. 7th International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition*, Springer, pp. 345–358.

F. Attneave (1954), 'Some informational aspects of visual perception', *Psych. Rev.* **61**, 183–193.

S. Awate and R. Whitaker (2006), 'Unsupervised, information-theoretic, adaptive image filtering for image restoration', *IEEE Trans. Patt. Anal. Machine Intell.* **28**, 364–376.

[6] The URLs cited in this work were correct at the time of going to press, but the publisher and the authors make no undertaking that the citations remain live or are accurate or appropriate.

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

C. Barnes, E. Shechtman, A. Finkelstein and D. Goldman (2009), PatchMatch: A randomized correspondence algorithm for structural image editing. In *ACM Transactions on Graphics: TOG*, Vol. 28, ACM, p. 24.

R. Bilcu and M. Vehvilainen (2008), Combined non-local averaging and intersection of confidence intervals for image denoising. In *15th IEEE International Conference on Image Processing*, pp. 1736–1739.

J. Boulanger, J. Sibarita, C. Kervrann and P. Bouthemy (2008), Non-parametric regression for patch-based fluorescence microscopy image sequence denoising. In *5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro, 2008*, pp. 748–751.

R. Bracho and A. Sanderson (1985), Segmentation of images based on intensity gradient information. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 19–23.

P. Brémaud (1999), *Markov Chains: Gibbs Fields, Monte Carlo Simulation, and Queues*, Vol. 31 of *Texts in Applied Mathematics*, Springer.

X. Bresson and T. Chan (2008), Non-local unsupervised variational image segmentation models. UCLA CAM Report 08-67.

A. Buades (2006), Image and film denoising by non-local means. PhD thesis, Universitat Illes Balears.

A. Buades, A. Chien, J. Morel and S. Osher (2008*a*), 'Topology preserving linear filtering applied to medical imaging', *SIAM J. Imaging Sci.* **1**, 26–50.

A. Buades, B. Coll and J. Morel (2004), Image data process by image noise reduction and camera integrating the means for implementing this process. French Patent 0404837.

A. Buades, B. Coll and J. Morel (2005*a*), 'A non-local algorithm for image denoising', *IEEE Computer Vision and Pattern Recognition* **2**, 60–65.

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

A. Buades, B. Coll and J. Morel (2006*a*), Image enhancement by non-local reverse heat equation. Preprint CMLA 2006-22.

A. Buades, B. Coll and J. Morel (2006*b*), 'Neighborhood filters and PDE's', *Numer. Math.* **105**, 1–34.

A. Buades, B. Coll and J. Morel (2006*c*), 'The staircasing effect in neighborhood filters and its solution', *IEEE Trans. Image Processing* **15**, 1499–1505.

A. Buades, B. Coll and J. Morel (2008*b*), 'Nonlocal image and movie denoising', *Internat. J. Computer Vision* **76**, 123–139.

A. Buades, B. Coll, J. Morel and C. Sbert (2009*a*), 'Self-similarity driven color demosaicking.', *IEEE Trans. Image Processing* **18**, 1192–1202.

A. Buades, M. Colom and J. Morel (2012*a*), 'Multiscale signal dependent noise estimation', *Image Processing On Line* (www.ipol.im).

A. Buades, M. Lebrun and J. Morel (2012*b*), 'Implementation of the 'non-local Bayes' image denoising algorithm', *Image Processing On Line* (www.ipol.im).

A. Buades, Y. Lou, J. Morel and Z. Tang (2009*b*), A note on multi-image denoising. In *Local and Non-Local Approximation in Image Processing, 2009: LNLA 2009*, IEEE, pp. 1–15.

E. Candès and M. Wakin (2008), 'An introduction to compressive sampling', *IEEE Signal Process. Magazine* **25**, 21–30.

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

M. Carreira-Perpinan (2000), 'Mode-finding for mixtures of Gaussian distributions', *IEEE Trans. Patt. Anal. Machine Intell.* **22**, 1318–1323.
P. Chatterjee and P. Milanfar (2010), 'Is denoising dead?', *IEEE Trans. Image Processing* **19**, 895–911.
P. Chatterjee and P. Milanfar (2012), 'Patch-based near-optimal image denoising', *IEEE Trans. Image Processing* **21**, 1635–1649.
C. Chevalier, G. Roman and J. Niépce (1854), *Guide du Photographe*, C. Chevalier.
A. Cohen, I. Daubechies and J. Feauveau (1992), 'Biorthogonal bases of compactly supported wavelets', *Comm. Pure Appl. Math.* **45**, 485–560.
R. Coifman and D. Donoho (1995), Translation-invariant de-noising. In *Wavelets and Statistics*, Lecture Notes in Statistics, Springer, pp. 125–125.
S. Cotter, R. Adler, R. Rao and K. Kreutz-Delgado (1999), 'Forward sequential algorithms for best basis selection', *IEE Proc. Vision, Image and Signal Processing* **146**, 235–244.
P. Coupé, P. Yger and C. Barillot (2006), Fast non local means denoising for 3D MRI images. In *Medical Image Computing and Computer-Assisted Intervention: MICCAI 2*, pp. 33–40.
P. Coupé, P. Yger, S. Prima, P. Hellier, C. Kervrann and C. Barillot (2008), 'An optimized blockwise nonlocal means denoising filter for 3-D magnetic resonance images', *IEEE Trans. Medical Imaging* **27**, 425–441.
K. Dabov, A. Foi, V. Katkovnik and K. Egiazarian (2007), 'Image denoising by sparse 3D transform-domain collaborative filtering', *IEEE Trans. Image Processing* **16**, 2080–2095.
K. Dabov, A. Foi, V. Katkovnik and K. Egiazarian (2009), BM3D image denoising with shape-adaptive principal component analysis. In *Proc. Workshop on Signal Processing With Adaptive Sparse Structured Representations: SPARS 09*, Vol. 49.
A. Danielyan and A. Foi (2009), Noise variance estimation in nonlocal transform domain. In *Proc. International Workshop on Local and Non-Local Approximation in Image Processing: LNLA 2009*, pp. 41–45.
A. Danielyan, A. Foi, V. Katkovnik and K. Egiazarian (2008), Image and video super-resolution via spatially adaptive block-matching filtering. In *Proc. International Workshop on Local and Non-Local Approximation in Image Processing*.
A. Danielyan, A. Foi, V. Katkovnik and K. Egiazarian (2010), Denoising of multispectral images via nonlocal groupwise spectrum-PCA. In *Proc. Fifth European Conference on Colour in Graphics, Imaging, and Vision and of the 12th International Symposium on Multispectral Colour Science*, pp. 261–266.
A. Danielyan, V. Katkovnik and K. Egiazarian (2012), 'BM3D frames and variational image deblurring', *IEEE Trans. Image Processing* **21**, 1715–1728.
J. Darbon, A. Cunha, T. Chan, S. Osher and G. Jensen (2008), Fast nonlocal filtering applied to electron cryomicroscopy. In *5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pp. 1331–1334.
J. De Bonet (1997), 'Noise reduction through detection of signal redundancy', *Rethinking Artificial Intelligence*.
C.-A. Deledalle, L. Denis and F. Tupin (2011*a*), 'NL-InSAR: Nonlocal interferogram estimation', *IEEE Trans. Geoscience and Remote Sensing* **49**, 1441–1452.

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

C.-A. Deledalle, V. Duval and J. Salmon (2012), 'Non-local methods with shape-adaptive patches (NLM-SAP)', *J. Math. Imaging Vision*, to appear.

C.-A. Deledalle, J. Salmon and A. Dalalyan (2011*b*), Image denoising with patch based PCA: Local versus global. In *Proc. British Machine Vision Conference*, BMVA Press, pp. 25.1–25.10.

C.-A. Deledalle, F. Tupin and L. Denis (2010*a*), Poisson NL means: Unsupervised non local means for Poisson noise. In *17th IEEE International Conference on Image Processing: ICIP*, pp. 801–804.

C.-A. Deledalle, F. Tupin and L. Denis (2010*b*), Polarimetric SAR estimation based on non-local means. In *IEEE International Geoscience and Remote Sensing Symposium: IGARSS*, pp. 2515–2518.

J. Delon and A. Desolneux (2009), Flicker stabilization in image sequences. hal.archives-ouvertes.fr.

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

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

D. Donoho and I. Johnstone (1994), 'Ideal spatial adaptation by wavelet shrinkage', *Biometrika* **81**, 425–455.

S. Durand and M. Nikolova (2003), 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*, pp. 145–152.

V. Duval, J. Aujol and Y. Gousseau (2011), 'A bias-variance approach for the nonlocal means', *SIAM J. Imaging Sci.* **4**, 760.

M. Ebrahimi and E. Vrscay (2007), Solving the inverse problem of image zooming using 'self-examples'. In *Image Analysis and Recognition: ICIAR 2007*, Vol. 4633 of *Lecture Notes in Computer Science*, Springer, pp. 117–130.

M. Ebrahimi and E. Vrscay (2008*a*), Examining the role of scale in the context of the non-local-means filter. In *Proc. International Conference on Image Analysis and Recognition: ICIAR 2008*, Vol. 5112 of *Lecture Notes in Computer Science*, Springer, pp. 170–181.

M. Ebrahimi and E. Vrscay (2008*b*), Multi-frame super-resolution with no explicit motion estimation. In *Proc. 2008 International Conference on Image Processing, Computer Vision, and Pattern Recognition: IPCV 2008*, pp. 455–459.

A. Efros and T. Leung (1999), Texture synthesis by non parametric sampling. In *Proc. International Conference on Computer Vision*, Vol. 2, pp. 1033–1038.

M. Elad and M. Aharon (2006), 'Image denoising via sparse and redundant representations over learned dictionaries', *IEEE Trans. Image Processing* **15**, 3736–3745.

M. Elad and D. Datsenko (2007), 'Example-based regularization deployed to super-resolution reconstruction of a single image', *Comput. J.* **52**, 15–30.

A. Elmoataz, O. Lézoray and S. Bougleux (2008*a*), 'Nonlocal discrete regularization on weighted graphs: A framework for image and manifold processing', *IEEE Trans. Image Processing* **17**, 1047–1060.

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

A. Elmoataz, O. Lézoray, S. Bougleux and V. Ta (2008*b*), Unifying local and non-local processing with partial difference operators on weighted graphs. In *International Workshop on Local and Non-Local Approximation in Image Processing*, pp. 11–26.

A. Foi (2011), Noise estimation and removal in MR imaging: The variance-stabilization approach. In *IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pp. 1809–1814.

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

A. Foi, M. Trimeche, V. Katkovnik and K. Egiazarian (2008), 'Practical Poissonian–Gaussian noise modeling and fitting for single-image raw-data', *IEEE Trans. Image Processing* **17**, 1737–1754.

S. Geman and D. Geman (1984), 'Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images', *IEEE Pat. Anal. Mach. Intell.* **6**, 721–741.

G. Gilboa and S. Osher (2008), 'Nonlocal linear image regularization and supervised segmentation', *Multiscale Model. Simul.* **6**, 595–630.

O. Guleryuz (2007), 'Weighted averaging for denoising with overcomplete dictionaries', *IEEE Trans. Image Processing* **16**, 3020–3034.

S. Harris (1966), 'Image evaluation and restoration', *J. Optical Soc. Amer.* **56**, 569–570.

J. Hays and A. Efros (2007), Scene completion using millions of photographs. In *ACM Transactions on Graphics: TOG*, Vol. 26, ACM, p. 4.

Y. Heo, K. Lee and S. Lee (2007), Simultaneous depth reconstruction and restoration of noisy stereo images using non-local pixel distribution. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8.

Y. Hou, C. Zhao, D. Yang and Y. Cheng (2011), 'Comments on "Image denoising by sparse 3D transform-domain collaborative filtering" ', *IEEE Trans. Image Processing* **20**, 268–270.

J. Immerkaer (1996), 'Fast noise variance estimation', *Computer Vision and Image Understanding* **64**, 300–302.

M. Jung and L. Vese (2009), Nonlocal variational image deblurring models in the presence of Gaussian or impulse noise. In *Proc. SSVM 2009*, Vol. 5567 of *Lecture Notes in Computer Science*, Springer, pp. 401–412.

V. Katkovnik, K. Egiazarian and J. Astola (2006), *Local Approximation Techniques in Signal and Image Processing*, SPIE.

V. Katkovnik, A. Danielyan and K. Egiazarian (2011), Decoupled inverse and denoising for image deblurring: Variational BM3D-frame technique. In *Proc. IEEE International Conference on Image Processing: ICIP 2011*, pp. 3453-3456.

S. Kay (1993), *Fundamentals of Statistical Signal Processing: Estimation Theory*, Prentice-Hall.

C. Kervrann and J. Boulanger (2008), 'Local adaptivity to variable smoothness for exemplar-based image regularization and representation', *Internat. J. Computer Vision* **79**, 45–69.

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

C. Kervrann, J. Boulanger and P. Coupé (2007), Bayesian non-local means filter, image redundancy and adaptive dictionaries for noise removal. In *Proc. SSVM '07*, Vol. 4485 of *Lecture Notes in Computer Science*, Springer, pp. 520–532.

S. Kindermann, S. Osher and P. Jones (2006), 'Deblurring and denoising of images by nonlocal functionals', *Multiscale Model. Simul.* **4**, 1091–1115.

E. Kolaczyk (1999), 'Wavelet shrinkage estimation of certain Poisson intensity signals using corrected thresholds', *Statist. Sin.* **9**, 119–135.

M. Lebrun, A. Buades and J. Morel (2011), 'Study and analysis of NL-PCA', *Image Processing On Line* (www.ipol.im).

A. Lee, K. Pedersen and D. Mumford (2003), 'The nonlinear statistics of high-contrast patches in natural images', *Internat. J. Computer Vision* **54**, 83–103.

J. Lee (1981), 'Refined filtering of image noise using local statistics', *Computer Graphics and Image Processing* **15**, 380–389.

J. Lee (1983), 'Digital image smoothing and the sigma filter', *Computer Vision, Graphics, and Image Processing* **24**, 255–269.

J. Lee and K. Hoppel (1989), Noise modelling and estimation of remotely-sensed images. In *Proc. International Geoscience and Remote Sensing Symposium*, IEEE, pp. 1005–1008.

S. Lefkimmiatis, P. Maragos and G. Papandreou (2009), 'Bayesian inference on multiscale models for Poisson intensity estimation: Application to photo-limited image denoising', *IEEE Trans. Image Processing* **18**, 1724–1741.

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

O. Lézoray, V. Ta and A. Elmoataz (2008), 'Nonlocal graph regularization for image colorization', *19th International Conference on Pattern Recognition: ICPR 2008*, pp. 1–4.

C. Liu, W. Freeman, R. Szeliski and S. Kang (2006), Noise estimation from a single image. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2006*, Vol. 1, pp. 901–908.

C. Liu, R. Szeliski, S. Kang, C. Zitnick and W. Freeman (2008), 'Automatic estimation and removal of noise from a single image', *IEEE Trans. Patt. Anal. Machine Intell.* **30**, 299–314.

Y. Lou, X. Zhang, S. Osher and A. Bertozzi (2008), Image recovery via nonlocal operators. UCLA CAM Report 08-35.

S. Lyu and E. Simoncelli (2009), 'Modeling multiscale subbands of photographic images with fields of Gaussian scale mixtures', *IEEE Trans. Patt. Anal. Machine Intell.* **31**, 693–706.

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

J. Mairal (2010), Représentations parcimonieuses en apprentissage statistique, traitement d'image et vision par ordinateur. PhD thesis, Ecole Normale Supérieure de Cachan.

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

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

J. Mairal, F. Bach, J. Ponce, G. Sapiro and A. Zisserman (2009*b*), Non-local sparse models for image restoration.. In *ICCV'09*, pp. 2272–2279.
J. Mairal, M. Elad and G. Sapiro (2008), 'Sparse representation for color image restoration', *IEEE Trans. Image Processing* **17**, 53–69.
M. Makitalo and A. Foi (2011), 'Optimal inversion of the Anscombe transformation in low-count Poisson image denoising', *IEEE Trans. Image Processing* **20**, 99–109.
A. Maleki, M. Narayan and R. Baraniuk (2011*a*), Anisotropic nonlocal means denoising. Arxiv preprint arXiv:1112.0311.
A. Maleki, M. Narayan and R. Baraniuk (2011*b*), Suboptimality of nonlocal means on images with sharp edges. In *49th Annual Allerton Conference on Communication, Control, and Computing*, pp. 299-305.
S. Mallat (1999), *A Wavelet Tour of Signal Processing*, Academic press.
E. Mammen and A. Tsybakov (1995), 'Asymptotical minimax recovery of sets with smooth boundaries', *Ann. Statist.* **23**, 502–524.
J. Manjón, J. Carbonell-Caballero, J. Lull, G. García-Martí, L. Martí-Bonmatí and M. Robles (2008), 'MRI denoising using non-local means', *Medical Image Analysis* **12**, 514–523.
J. Manjón, M. Robles and N. Thacker (2007), Multispectral MRI denoising using non-local means. In *Proc. MIUA*, Vol. 7, pp. 41–45.
G. Mastin (1985), 'Adaptive filters for digital image noise smoothing: An evaluation', *Computer Vision, Graphics, and Image Processing* **31**, 103–121.
G. Mayer, E. Vrscay, M. Lauzon, B. Goodyear and J. Mitchell (2008), 'Self-similarity of images in the Fourier domain, with applications to MRI', In *Proc. International Conference on Image Analysis and Recognition: ICIAR 2008*, Vol. 5112 of *Lecture Notes in Computer Science*, Springer, pp. 43–52.
P. Meer, J. Jolion and A. Rosenfeld (1990), 'A fast parallel algorithm for blind estimation of noise variance', *IEEE Trans. Patt. Anal. Machine Intell.* **12**, 216–223.
Y. Meyer (1993), *Wavelets: Algorithms and Applications*, SIAM.
M. Mignotte (2008), 'A non-local regularization strategy for image deconvolution', *Pattern Recognition Letters* **29**, 2206–2212.
P. Milanfar (2011), 'A tour of modern image filtering', *IEEE Signal Processing Magazine*. Preprint at http://users.soe.ucsc. edu/∼milanfar/publications/.
B. Naegel, A. Cernicanu, J. Hyacinthe, M. Tognolini and J. Vallée (2009), 'SNR enhancement of highly-accelerated real-time cardiac MRI acquisitions based on non-local means algorithm', *Med. Image Anal.* **13**, 598–608.
A. Nemirovski (2000), Topics in non-parametric statistics. In *Lectures on Probability Theory and Statistics: Saint-Flour 1998*, Vol. 1738 of *Lecture Notes in Mathematics*, Springer, pp. 85–277.
R. Nowak and R. Baraniuk (1997), 'Wavelet-domain filtering for photon imaging systems', *IEEE Trans. Image Processing* **8**, 666–678.
S. Olsen (1993), 'Estimation of noise in images: An evaluation', *CVGIP: Graphical Models and Image Processing* **55**, 319–323.
J. Orchard, M. Ebrahimi and A. Wong (2008), Efficient non-local-means denoising using the SVD. In *Proc. IEEE International Conference on Image Processing*, pp. 1732–1735.

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

E. Ordentlich, G. Seroussi, S. Verdu, M. Weinberger and T. Weissman (2003), A discrete universal denoiser and its application to binary images. In *International Conference on Image Processing*, Vol. 1, pp. 117–120.

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

E. Pennec and S. Mallat (2003), Geometrical image compression with bandelets. In *Proc. SPIE 2003*, Vol. 5150, pp. 1273–1286.

G. Peyré (2009), 'Manifold models for signals and images', *Computer Vision and Image Understanding* **113**, 249–260.

N. Ponomarenko, V. Lukin, S. Abramov, K. Egiazarian and J. Astola (2003), Blind evaluation of additive noise variance in textured images by nonlinear processing of block DCT coefficients. In *Proc. SPIE*, Vol. 5014, p. 178.

N. Ponomarenko, V. Lukin, M. Zriakhov, A. Kaarna and J. Astola (2007), An automatic approach to lossy compression of AVIRIS images. In *IEEE International Geoscience and Remote Sensing Symposium*, pp. 472–475.

J. Portilla, V. Strela, M. Wainwright and E. Simoncelli (2003), 'Image denoising using scale mixtures of Gaussians in the wavelet domain', *IEEE Trans. Image Processing* **12**, 1338–1351.

M. Protter, M. Elad, H. Takeda and P. Milanfar (2009), 'Generalizing the non-local-means to super-resolution reconstruction', *IEEE Trans. Image Processing* **18**, 36–51.

K. Rank, M. Lendl and R. Unbehauen (1999), 'Estimation of image noise variance', *IEE Proc. Vision, Image and Signal Processing* **146**, 80–84.

M. Raphan and E. Simoncelli (2007), Learning to be Bayesian without supervision. In *Advances in Neural Information Processing Systems*, Vol. 19, pp. 1145–1152.

M. Raphan and E. Simoncelli (2010), An empirical Bayesian interpretation and generalization of NL-means. Technical Report TR2010-934, Courant Institute of Mathematical Sciences.

W. Richardson (1972), 'Bayesian-based iterative method of image restoration', *J. Optical Soc. Amer.* **62**, 55–59.

L. Rudin, S. Osher and E. Fatemi (1992), 'Nonlinear total variation based noise removal algorithms', *Physica D* **60**, 259–268.

B. Russell, A. Torralba, K. Murphy and W. Freeman (2008), 'LabelMe: A database and web-based tool for image annotation', *Internat. J. Computer Vision* **77**, 157–173.

C. Shannon (2001), A mathematical theory of communication. In *ACM SIGMOBILE Mobile Computing and Communications Review*, Vol. 5, pp. 3–55.

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

S. Smith and J. Brady (1997), 'SUSAN: A new approach to low level image processing', *Internat. J. Computer Vision* **23**, 45–78.

J. Starck, E. Candès and D. Donoho (2002), 'The curvelet transform for image denoising', *IEEE Trans. Image Processing* **11**, 670–684.

A. Szlam, M. Maggioni and R. Coifman (2006), A general framework for adaptive regularization based on diffusion processes on graphs. Yale technical report.

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

N. Thacker, J. Manjon and P. Bromiley (2008), A statistical interpretation of non-local means. In *5th International Conference on Visual Information Engineering*, IEEE, pp. 250–255.

C. Tomasi and R. Manduchi (1998), Bilateral filtering for gray and color images. In *Sixth International Conference on Computer Vision, 1998*, IEEE, pp. 839–846.

J. Tukey (1977), *Exploratory Data Analysis*, Addison-Wesley.

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

H. Voorhees and T. Poggio (1987), Detecting textons and texture boundaries in natural image. In *Proc. First IEEE International Conference on Computer Vision*, pp. 250–258.

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

T. Weissman, E. Ordentlich, G. Seroussi, S. Verdu and M. Weinberger (2005), 'Universal discrete denoising: Known channel', *IEEE Trans. Inform. Theory* **51**, 5–28.

N. Wiest-Daesslé, S. Prima, P. Coupé, S. Morrissey and C. Barillot (2007), Non-local means variants for denoising of diffusion-weighted and diffusion tensor MRI. In *Proc. 10th International Conference on Medical image Computing and Computer-Assisted Intervention: MICCAI 2007*, Vol. 4792 of *Lecture Notes in Computer Science*, Springer, pp. 344–351.

N. Wiest-Daesslé, S. Prima, P. Coupé, S. Morrissey and C. Barillot (2008), Rician noise removal by non-local means filtering for low signal-to-noise ratio MRI: Applications to DT-MRI. In *Medical Image Computing and Computer-Assisted Intervention: MICCAI 2008*, Vol. 5242 of *Lecture Notes in Computer Science*, Springer, pp. 171–179.

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

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

L. Yaroslavsky (1985), *Digital Picture Processing*, Springer.

L. Yaroslavsky (1996), Local adaptive image restoration and enhancement with the use of DFT and DCT in a running window. In *Proc. SPIE*, Vol. 2825, p. 2.

L. Yaroslavsky and M. Eden (2003), *Fundamentals of Digital Optics*, Birkhäuser.

G. Yu and G. Sapiro (2011), 'DCT image denoising: A simple and effective image denoising algorithm', *Image Processing On Line* (www.ipol.im).

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

G. Yu, G. Sapiro and S. Mallat (2012), Solving inverse problems with piecewise linear estimators: From Gaussian mixture models to structured sparsity. *IEEE Trans. Image Processing*, to appear.

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press

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

X. Zhang, M. Burger, X. Bresson and S. Osher (2009), Bregmanized nonlocal regularization for deconvolution and sparse reconstruction. UCLA CAM Report 09-03.

M. Zhou, H. Yang, G. Sapiro, D. B. Dunson and L. Carin (2011), Dependent hierarchical beta process for image interpolation and denoising. In *Proc. Journal of Machine Learning Research*, pp. 883–891.

S. Zimmer, S. Didas and J. Weickert (2008), 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*, IEEE, pp. 135–142.

D. Zoran and Y. Weiss (2009), Scale invariance and noise in natural images. In *IEEE 12th International Conference on Computer Vision*, pp. 2209–2216.

D. Zoran and Y. Weiss (2011), From learning models of natural image patches to whole image restoration. In *2011 IEEE International Conference on Computer Vision* (ICCV), pp. 479–486.

https://doi.org/10.1017/S0962492912000062 Published online by Cambridge University Press