

# Below the Surface of the Non-Local Bayesian Image Denoising Method

Pablo Arias\* and Mila Nikolova\*\*

CMLA, ENS Cachan, Université Paris-Saclay  
94235 Cachan, France  
{arias, nikolova}@cmla.ens-cachan.fr

**Abstract.** The non-local Bayesian (NLB) patch-based approach of Lebrun, Buades, and Morel [12] is considered as a state-of-the-art method for the restoration of (color) images corrupted by white Gaussian noise. It gave rise to numerous ramifications like e.g., possible improvements, processing of various data sets and video. This article is the first attempt to analyse the method in depth in order to understand the main phenomena underlying its effectiveness. Our analysis, corroborated by numerical tests, shows several unexpected facts. In a variational setting, the first-step Bayesian approach to learn the prior for patches is equivalent to a pseudo-Tikhonov regularisation where *the regularisation parameters can be positive or negative*. Practically very good results in this step are mainly due to the aggregation stage – whose importance needs to be re-evaluated.

## 1 Introduction

In this paper we analyse the Non-local Bayesian (NLB) image denoising algorithm introduced by Lebrun, Buades and Morel in [12], which is based on the assumption that sets of similar patches are IID samples from a Gaussian distribution.

In recent years several works have proposed Gaussian models, or Gaussian mixture models (GMMs) as priors for image patches, achieving state-of-the-art results in image denoising and other inverse problems. In [19] a GMM with hundreds of components is used as a fixed prior for image patches. The mixture is learnt via an EM algorithm from a database of two million patches. Instead of a fixed prior [17, 5] proposed to learn the GMM from the input image. Guillemot et al. [9] introduced the *covariance tree*, a hierarchical data structure capturing the covariance of image patches at several scales.

The NLB algorithm differs from the previous approaches in that it does not consider a single model (GMM, covariance tree) for all image patches. Instead, there is a Gaussian model for each patch and its nearest neighbors. This is closely

---

\* The work of P. Arias is funded by BPIFrance and Région Ile de France in the FUI 18 Plein Phare project. P. A. is also partly funded by the “IDI 2016” project funded by the IDEX Paris-Saclay, ANR-11-IDEX-0003-02, the Office of Naval research (ONR grant N00014-14-1-0023) and by ANR-14-CE27-001 (MIRIAM).

\*\* The work of M. Nikolova work was partially funded by the French Research Agency (ANR) under grant No ANR-14-CE27-001 (MIRIAM).

related to the approaches that denoise patches by applying principal component analysis (PCA) on groups of patches [15, 6, 18, 8].

Several extensions of NLB were considered. A multiscale version was developed in [14] to handle the structured noise typical of compression artifacts. Extensions to video were proposed in [4, 2]. NLB was adapted to denoise manifolds-valued images in [10]. A Bayesian hyper-prior model on the mean and the covariance in the NLB approach was proposed for inverse problems in [1].

In the NLB method, for each set of similar patches an empirical Wiener filter is used to estimate the corresponding clean patches. The denoised image results from the aggregation by averaging of these estimated patches. The Wiener filter requires to estimate the mean and the covariance matrix of the *a priori* distribution. The method consists of two steps. In the first step the statistics of the prior are estimated from the noisy patches. The second step is similar but uses the output of the first step as an oracle to estimate the *a priori* statistics.

In this work we carefully dissect the NLB algorithm, focusing on the first step which is responsible for most of the denoising. On the surface, the NLB algorithm appears as a *maximum a posteriori* (MAP) estimation of each patch. But in fact these empirical MAP estimates are poor, as a consequence of the fact that the prior covariance estimate has in general negative eigenvalues which result in an unstable filter. The success of the algorithm resides in that the aggregation phase averages out the errors of the MAP estimates, achieving similar and often better results than the ones obtained with a more standard non-negative definite estimate of the prior covariance matrix. Although the aggregation has been recognized as a good practice in patch-based denoising [13], the results presented below show that it plays a crucial role for the success of the NLB.

## 2 The theory of the algorithm

For an original gray value image  $x \in \mathbb{R}^{M_1 \times M_2}$  we are given an observation contaminated with centered white normal noise  $n$  with known variance  $\sigma^2$ , independent of  $x$ :

$$y = x + n.$$

The approach proposed in [12] relies on the fact that natural images exhibit patch-based self-similarity, and assumes that groups of similar patches follow a normal distribution. The NLB algorithm consists in two main denoising steps: **(1)** compute an oracle image  $\tilde{x}$  using the data  $y$  and  $\sigma^2$ ; **(2)** obtain the denoised image  $\hat{x}$  using the oracle  $\tilde{x}$ , the data  $y$  and  $\sigma^2$ . Each one of these steps consist of two stages: **(a)** denoise individually all patches using a maximum a posteriori (MAP) estimator; **(b)** aggregate the results of **(a)**.

An example of the output of each step is given for the  $256 \times 256$  *parrot* image, shown in Fig. 1. Some noise artifacts reside in the oracle image  $\tilde{x}$ . They are restored in the denoised image.

### 2.1 Bayesian model for patches

A patch of  $y$  (resp.,  $x$ ) of size  $\sqrt{K} \times \sqrt{K}$  at pixel  $\mathbf{i} = (i_1, i_2)$  is denoted by  $\mathbf{y}_{\mathbf{i}}$  (resp.,  $\mathbf{x}_{\mathbf{i}}$ ). In what follows, we adopt a vector notation for patches. The



**Fig. 1.** From left to right: noisy image  $y$  for  $\sigma = 20$ ; first-step denoised  $\tilde{x}$  (29.57 PSNR), denoised  $\hat{x}$  (29.95 PSNR) and original image.

likelihood of  $\mathbf{x}_i$  given the observed  $\mathbf{y}_i$  is

$$\mathbb{P}(\mathbf{y}_i|\mathbf{x}_i) \propto \exp\left[-\frac{\|\mathbf{x}_i - \mathbf{y}_i\|^2}{2\sigma^2}\right]. \quad (1)$$

With each patch  $\mathbf{x}_i$ , there is associated a set of  $N$  most similar patches  $S(\mathbf{i})$  within a fixed size window. The patch  $\mathbf{x}_i$  is called the reference patch. Similarity is measured with respect to the  $\ell_2$  distance. It is assumed that all patches in  $S(\mathbf{i})$  have a normal distribution sharing the same mean  $\mathbf{m}_{\mathbf{x}_i} \in \mathbb{R}^K$  and the same covariance  $C_{\mathbf{x}_i}$ , see [12]. Thus

$$\mathbb{P}(\mathbf{x}_i) \propto \exp\left(-\frac{1}{2}(\mathbf{x}_i - \mathbf{m}_{\mathbf{x}_i})^T C_{\mathbf{x}_i}^{-1}(\mathbf{x}_i - \mathbf{m}_{\mathbf{x}_i})\right). \quad (2)$$

Using Bayes's rule,  $\mathbb{P}(\mathbf{x}_i|\mathbf{y}_i) \propto \mathbb{P}(\mathbf{y}_i|\mathbf{x}_i)\mathbb{P}(\mathbf{x}_i)$ . The MAP estimate  $\hat{\mathbf{x}}_i^{(i)}$  of the noisy reference patch  $\mathbf{y}_i$ ,

$$\hat{\mathbf{x}}_i^{(i)} = \arg \max_{\mathbf{x}_i} \mathbb{P}(\mathbf{x}_i|\mathbf{y}_i) = (I + \sigma^2 C_{\mathbf{x}_i}^{-1})^{-1} (\mathbf{y}_i + \sigma^2 C_{\mathbf{x}_i}^{-1} \mathbf{m}_{\mathbf{x}_i})$$

is known to provide some denoising. Using that for an invertible matrix  $C$  one has  $(I + \sigma^2 C^{-1})^{-1} C^{-1} = (C + \sigma^2 I)^{-1}$  and  $\sigma^2 (C + \sigma^2 I)^{-1} - I = -C (C + \sigma^2 I)^{-1}$ , the MAP estimate reads also as

$$\hat{\mathbf{x}}_i = \mathbf{m}_{\mathbf{x}_i} + C_{\mathbf{x}_i} (C_{\mathbf{x}_i} + \sigma^2 I)^{-1} (\mathbf{y}_i - \mathbf{m}_{\mathbf{x}_i}). \quad (3)$$

This individual MAP patch denoising is considered as the main phase of the NLB algorithm [12].

*Remark 1.* Extensive numerical tests reported in [11] have shown that the denoising formula (3) can be applied simultaneously to all of the  $N$  most similar patches, i.e. those belonging to  $S_i$ . As a consequence the same patch  $\mathbf{y}_k$  can be denoised several times, if  $\mathbf{k} \in S(\mathbf{i})$  for different  $\mathbf{i}$ . We denote the corresponding MAP estimates as  $\mathbf{x}_k^{(i)}$ .

After all patches are denoised using the local MAP estimator, an aggregation stage is applied by computing for each pixel  $\mathbf{i}$  the mean over all restored patches that contain this pixel.

The main difference of the NLB method compared to other Gaussian-based models for image patches [19, 17, 5] is the way how the matrix  $C_{\mathbf{x}_i}$  is learned from the image itself. The question is analysed in the following sections.

## 2.2 Learning the *a priori* model

We have to distinguish between the first and the second steps. The number of the pixels in the square patch window is denoted by  $K_j$  and the number of similar patches by  $N_j$ , where  $j = 1, 2$  indicates the step.

**First step.** The  $N_1$  patches  $S_1(\mathbf{i})$  similar to the reference patch  $\mathbf{y}_i$  are selected using the  $\ell^2$  distance between  $\mathbf{y}_i$  and all patches  $\mathbf{y}_k$  in a neighborhood around  $\mathbf{y}_i$ . Let us fix at patch  $\mathbf{y}_i$  and its similar patches  $\mathbf{y}_k \in S_1(\mathbf{i})$ .

Since  $\mathbf{x}_i \sim \mathcal{N}(\mathbf{m}_{\mathbf{x}_i}, \mathcal{C}_{\mathbf{x}_i})$  and  $\mathbf{n}_i \sim \mathcal{N}(0, \sigma^2 I)$  and since the noise is independent from the original  $x$ , we have that  $\mathbf{y}_k \sim \mathcal{N}(\mathbf{m}_{\mathbf{x}_i}, \mathcal{C}_{\mathbf{y}_i})$ , with  $\mathcal{C}_{\mathbf{y}_i} = \mathcal{C}_{\mathbf{x}_i} + \sigma^2 I$ , for  $\mathbf{y}_k \in S(\mathbf{i})$ . The maximum likelihood estimates of  $\mathbf{m}_{\mathbf{x}_i}$  and  $\mathcal{C}_{\mathbf{y}_i}$  are given by the sample mean and sample covariance matrix. The ML estimate of the covariance matrix is biased. An unbiased estimator can be obtained by dividing by  $N_1 - 1$  instead of  $N_1$ , yielding

$$\bar{\mathbf{y}}_i = \frac{1}{N_1} \sum_{\mathbf{k} \in S_1(\mathbf{i})} \mathbf{y}_k, \quad \mathcal{C}_{\mathbf{y}_i} = \frac{1}{N_1 - 1} \sum_{\mathbf{k} \in S_1(\mathbf{i})} (\mathbf{y}_k - \bar{\mathbf{y}}_i)(\mathbf{y}_k - \bar{\mathbf{y}}_i)^T, \quad (4)$$

where the superscript  $T$  means transposed. Using that  $\mathcal{C}_{\mathbf{y}_i} = \mathcal{C}_{\mathbf{x}_i} + \sigma^2 I$ , the authors in [12] estimate  $\mathcal{C}_{\mathbf{x}_i}$  as

$$\mathcal{C}_{\mathbf{x}_i} = \mathcal{C}_{\mathbf{y}_i} - \sigma^2 I. \quad (5)$$

This estimate is often not positive definite. There are different ways to avoid this; e.g., clipping the negative eigenvalues to zero has been used on the context of patch-based denoising in [5]. This may also have undesirable effects, see the discussion in [16, p. 406]. New facts are given in subsection 3.3.

In spite of the theoretical background explained above, extensive experiments in [11] have shown that using the MAP estimator yields artifacts in flat areas.

*Remark 2.* [Flat areas] *Flat areas are detected by applying a  $\chi^2$  Gaussianity test to the set of all pixels of the patches in  $S(\mathbf{i})$ . If a flat area is detected, the patches in  $S(\mathbf{i})$  are estimated as constant patches, where the constant value is the average of all pixel values of all patches in  $S(\mathbf{i})$ .*

**Second step.** This step is grounded on the presumption that the *oracle* estimate  $\tilde{x}$  is faithful enough to provide two informations on the unknown ground truth image. These are: (1) the distances between patches in the *oracle* are well estimated, so the set of the  $N_2$  patches similar to  $\mathbf{y}_i$  is computed from the *oracle* estimate  $\tilde{x}$ ; (2) the covariance matrices for patches provided by the *oracle* image  $\tilde{x}$  are “nearly true”, so these matrices are computed using only the *oracle* image.

## 3 An oracle image in the first step

In this section we study in detail the first step of the algorithm.

### 3.1 Statistical model and filtering for 1st step patch denoising

Replacing  $C_{\mathbf{x}_i}$  and  $\mathbf{m}_{\mathbf{x}_i}$  in (3) by their estimates (4), we end up with the following Wiener-like filter

$$\tilde{\mathbf{x}}_{\mathbf{k}}^{(i)} = \bar{\mathbf{y}}_i + (C_{\mathbf{y}_i} - \sigma^2 I) C_{\mathbf{y}_i}^{-1} (\mathbf{y}_{\mathbf{k}} - \bar{\mathbf{y}}_i). \quad (6)$$

This filter is applied to all patches  $\mathbf{k} \in S_1(\mathbf{i})$ . After simplification,  $\tilde{\mathbf{x}}_{\mathbf{k}}^{(i)} = \mathbf{y}_{\mathbf{k}} - \sigma^2 C_{\mathbf{y}_i}^{-1} (\mathbf{y}_{\mathbf{k}} - \bar{\mathbf{y}}_i)$ . Since  $C_{\mathbf{y}_i} \succeq 0$  its eigen-decomposition is of the form

$$C_{\mathbf{y}_i} = U_{\mathbf{y}_i} \text{diag}(\lambda_{\mathbf{y}_i}) U_{\mathbf{y}_i}^T. \quad (7)$$

Here  $U_{\mathbf{y}_i}$  is an orthonormal matrix (i.e. it satisfies  $U_{\mathbf{y}_i}^T U_{\mathbf{y}_i} = U_{\mathbf{y}_i} U_{\mathbf{y}_i}^T = I$ ) containing the eigenvectors of  $C_{\mathbf{y}_i}$  and  $\lambda_{\mathbf{y}_i} \in \mathbb{R}_{\geq 0}^K$  are the corresponding eigenvalues of  $C_{\mathbf{y}_i}$  in decreasing order. The estimate of  $C_{\mathbf{x}_i}$  in (5) reads as

$$C_{\mathbf{x}_i} = C_{\mathbf{y}_i} - \sigma^2 I = U_{\mathbf{y}_i} \text{diag}(\lambda_{\mathbf{y}_i} - \sigma^2) U_{\mathbf{y}_i}^T. \quad (8)$$

Thus the Wiener-like denoising in (6) can be interpreted as a filtering in the basis of the eigenvectors  $U_{\mathbf{y}_i}$  of  $C_{\mathbf{y}_i}$ :

$$U_{\mathbf{y}_i}^T (\mathbf{x}_{\mathbf{k}}^{(i)} - \bar{\mathbf{y}}_i) = \text{diag} \left( \frac{\lambda_{\mathbf{y}_i} - \sigma^2}{\lambda_{\mathbf{y}_i}} \right) (U_{\mathbf{y}_i}^T (\mathbf{y}_{\mathbf{k}} - \bar{\mathbf{y}}_i)). \quad (9)$$

where division in the diagonal matrix is vector-wise.

A filtering of this form is stable and the learnt prior model is consistent if and only if  $C_{\mathbf{y}_i} - \sigma^2 I$  has positive eigenvalues.

**The eigenvalues of the estimated prior covariances  $C_{\mathbf{y}_i} - \sigma^2 I$ .** It appears that  $C_{\mathbf{y}_i} - \sigma^2 I$  nearly always has negative eigenvalues even for *non-flat* patches, whatever the level of the noise. Table 1 gives more details for five images and different values of  $\sigma \in \{1, 5, 10, 20, 40\}$ . The parameter choice follows the recommendations in [11]. Patches are square; for  $\sigma < 20$  one takes  $(K_1, N_1) = (9, 27)$  and for  $20 \leq \sigma < 50$   $(K_1, N_1) = (25, 75)$ ; as recommended,  $N_i = 3K_i$  and the search window is a square of size  $\lfloor N_i/2 \rfloor$ . Flat areas are detected as described in Remark 2. As in [12], with each index  $\mathbf{i}$  in the image (except at the boundary) there is associated a reference patch  $\mathbf{y}_i$ . The set of indexes corresponding to *non-flat reference patches* is denoted by  $J$ . Column (a) contains the least eigenvalues of  $C_{\mathbf{y}_i}$  for each image and for each noise level  $\sigma$  over all patches in the relevant set  $J$ . Since the eigenvalues are ordered decreasingly, the least one is  $K_1$ . All these least eigenvalues are much smaller than the corresponding  $\sigma^2$ . Column (b) shows the percentage of matrices  $C_{\mathbf{y}_i} - \sigma^2 I$  over  $J$  with at least one negative eigenvalue, which also reads as  $\#\{\mathbf{y}_i, \mathbf{i} \in J : g(\lambda_{K_1}) < 0\}$ . Observe that these percentages are large, going up to 100 %. What is more, the number of negative weights in  $g(\lambda_{\mathbf{y}_i})$  for non-flat reference patches is important: column (c) presents the *average* number of negative weights in  $g(\lambda_{\mathbf{y}_i})$  taken over  $J$ .

**Consequences.** Since  $C_{\mathbf{x}_i} = C_{\mathbf{y}_i} - \sigma^2 I$  has negative eigenvalues it *is not a covariance matrix*, and the learned prior  $\mathbb{P}(\mathbf{x}_i)$  *is not a probability*.

Table 1 shows that most of the noisy prior covariances estimated by the denoising algorithm have a large number of negative eigenvalues, and thus they have a significant impact in the pseudo-Wiener denoising in (6) and (9).

		(a) $\min_{\mathbf{y}_i, \mathbf{i} \in J} \lambda_{\mathbf{y}_i}(K_1)$	(b) $\#\{\mathbf{y}_i : g(\lambda_{K_1}) < 0\}$	(c) $\overline{\#\{k \in J : g(\lambda_k) < 0\}}$
boat (512 × 512)	$\sigma = 1$	0.087	10.25 %	1.702 %
	$\sigma = 5$	0.883	96.23 %	40.75 %
	$\sigma = 10$	3.747	99.85 %	53.47 %
	$\sigma = 20$	31.88	100 %	55.13 %
	$\sigma = 40$	122.7	100 %	62.25 %
man (1024 × 1024)	$\sigma = 1$	0.059	10.37 %	2.584 %
	$\sigma = 5$	0.909	95.84 %	41.41 %
	$\sigma = 10$	3.833	99.64 %	54.65 %
	$\sigma = 20$	29.01	100 %	56.83 %
	$\sigma = 40$	134.6	100 %	62.23 %
parrot (256 × 256)	$\sigma = 1$	0.051	45.43 %	16.59 %
	$\sigma = 5$	1.201	80.75 %	29.84 %
	$\sigma = 10$	4.811	93.88 %	42.26 %
	$\sigma = 20$	35.22	100 %	48.16 %
	$\sigma = 40$	134.9	100 %	89.42 %
peppers (512 × 512)	$\sigma = 1$	0.126	13.48 %	1.914 %
	$\sigma = 5$	0.772	97.67 %	46.82 %
	$\sigma = 10$	3.309	99.88 %	56.94 %
	$\sigma = 20$	34.05	100 %	56.63 %
	$\sigma = 40$	127.3	100 %	62.62 %
stream (512 × 512)	$\sigma = 1$	0.093	5.194 %	0.856 %
	$\sigma = 5$	0.942	67.77 %	20.44 %
	$\sigma = 10$	4.354	97.14 %	39.19 %
	$\sigma = 20$	29.55	100 %	50.38 %
	$\sigma = 40$	130.5	100 %	61.69 %

**Table 1.** (a) The least eigenvalue of  $C_{\mathbf{y}_i}$  for all reference patches in non-flat areas. (b) Percentage of reference patches  $\mathbf{y}_i$  in non-flat areas with at least one negative weight in  $g(\lambda_{\mathbf{y}_i})$ . (c) Average number of negative weights in  $g(\lambda_{\mathbf{y}_i})$  for reference patches  $\mathbf{y}_i$  in non-flat areas.

*Pseudo-Tikhonov regularization.* Observe that  $\tilde{\mathbf{x}}_{\mathbf{k}}^{(i)}$  in (6) is the unique stationary point of the objective  $F$  below

$$F(\mathbf{x}) = \|\mathbf{x} - \mathbf{y}_i\|^2 + \sigma^2(\mathbf{x} - \bar{\mathbf{y}}_i)^T (C_{\mathbf{y}_i} - \sigma^2 I)^{-1} (\mathbf{x} - \bar{\mathbf{y}}_i). \quad (10)$$

Using the eigen-decomposition in (7) and the fact that  $U_{\mathbf{y}_i}$  is orthonormal,  $F$  also reads as

$$F(\mathbf{x}) = \|U_{\mathbf{y}_i}^T (\mathbf{x} - \mathbf{y}_i)\|^2 + \sigma^2(\mathbf{x} - \bar{\mathbf{y}}_i)^T U_{\mathbf{y}_i} (\text{diag}(\lambda_{\mathbf{y}_i} - \sigma^2 I))^{-1} U_{\mathbf{y}_i}^T (\mathbf{x} - \bar{\mathbf{y}}_i). \quad (11)$$

We denote  $\lambda_{\mathbf{y}_i} := (\lambda_1, \dots, \lambda_K)$ . Let  $\mathbf{u}_{\mathbf{y}_i, k}$  stand for the  $k$ th column of  $U_{\mathbf{y}_i}$ . Then

$$F(\mathbf{x}) = \sum_{k=1}^K (\mathbf{u}_{\mathbf{y}_i, k}^T (\mathbf{x} - \mathbf{y}_i))^2 + \beta(\lambda_k) (\mathbf{u}_{\mathbf{y}_i, k}^T (\mathbf{x} - \bar{\mathbf{y}}_i))^2, \quad (12)$$

where  $\beta(\lambda) = \sigma^2 / (\lambda - \sigma^2)$ . The function  $\lambda \mapsto \beta(\lambda)$  is continuous on  $\mathbb{R} \setminus \{0\}$  with

$$\lim_{\lambda \searrow 0} \beta(\lambda) = -1 \quad \lim_{\lambda \nearrow \sigma^2} \beta(\lambda) = -\infty \quad \lim_{\lambda \searrow \sigma^2} \beta(\lambda) = \infty \quad \beta(2\sigma^2) = 1 \quad \lim_{\lambda \rightarrow +\infty} \beta(\lambda) = 0.$$

The objective  $F$  is a quadratic *convex-concave function*. It can be seen as a pseudo-Tikhonov regularization in the basis  $U_{\mathbf{y}_i}$  where the regularization parameters  $\beta(\lambda_k)$  can be positive or negative. Let us remind that classical Tikhonov regularization with all  $\beta > 0$  entails oversmoothing; see e.g., [3, sec. 3.2]. Table 1(c) shows that  $F$  in (12) has many negative regularization parameters.

*Pseudo-Wiener filtering.* Equation (9) reads also as

$$U_{\mathbf{y}_i}^T \tilde{\mathbf{x}}_{\mathbf{k}}^{(i)} = \text{diag} \left( \frac{\lambda_{\mathbf{y}_i} - \sigma^2}{\lambda_{\mathbf{y}_i}} \right) (U_{\mathbf{y}_i}^T (\mathbf{y}_{\mathbf{k}} - \bar{\mathbf{y}}_i)) + U_{\mathbf{y}_i}^T \bar{\mathbf{y}}_i. \quad (13)$$

Since  $U_{\mathbf{y}_i}$  is an orthonormal matrix, the noise statistics of  $\mathbf{y}_i$  and of  $U_{\mathbf{y}_i}^T \mathbf{y}_i$  are the same. The weighting coefficients in the diagonal matrix in (13) are given by a function  $g : \mathbb{R}_{>0} \rightarrow \mathbb{R}$

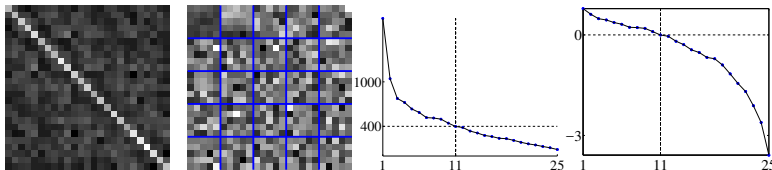
$$g(\lambda) := (\lambda - \sigma^2)/\lambda = 1 - \sigma^2/\lambda \quad (14)$$

which is differentiable, strictly increasing with  $\lim_{\lambda \searrow 0} g(\lambda) = -\infty$ ,  $g(\sigma^2) = 0$  and  $\lim_{\lambda \rightarrow +\infty} g(\lambda) = 1$ . The weights  $g(\lambda)$  are negative for  $\lambda < \sigma^2$  and rapidly decrease with  $\lambda \searrow 0$ . This  $g$  is related to  $\beta$  in (12) by  $g(\lambda) = (\beta(\lambda) + 1)^{-1}$ .

Thus the components of  $U_{\mathbf{y}_i}^T \mathbf{x}_i^{(i)}$  in (13) satisfy

$$\mathbf{u}_{\mathbf{y}_i, k}^T (\tilde{\mathbf{x}}_{\mathbf{k}}^{(i)} - \bar{\mathbf{y}}_i) = g(\lambda_k) \mathbf{u}_{\mathbf{y}_i, k}^T (\mathbf{y}_{\mathbf{k}} - \bar{\mathbf{y}}_i), \quad 1 \leq k \leq K_1. \quad (15)$$

Fig. 2 shows a sample covariance matrix  $C_{\mathbf{y}_i}$  with its eigenvectors and eigenvalues  $\lambda_{\mathbf{y}_i}$  for a non-flat patch of the noisy *parrot* image in Fig. 1 where  $\sigma = 20$ . The eigenvectors are plotted in a raster order. The weights  $g(\lambda_{\mathbf{y}_i})$  in (14) are depicted in the same figure. More than half of the eigenvalues are smaller than  $\sigma^2$ , resulting in negative eigenvalues in  $C_{\mathbf{x}_i}$  and negative filter coefficients  $g(\lambda)$ . The resulting filter gives large negative weights to the eigenvectors associated to the smallest eigenvalues, which look like noise with no apparent structure.



**Fig. 2.** For a non-flat patch in the noisy *parrot* image in Fig. 1, with  $\sigma = 20$ , from left to right: sample covariance matrix  $C_{\mathbf{y}_i}$ ; its eigenvectors in raster order; the eigenvalues  $\lambda_{\mathbf{y}_i}$  and the resulting weighting function  $g(\lambda_{\mathbf{y}_i})$ . Notice the large filter weights  $g(\lambda)$  corresponding to  $\lambda < \sigma^2$ .

*Remark 3.* According to the Eckart-Young theorem, the capability of the eigenvectors of  $C_{\mathbf{y}_i}$  to encode important features of the patch  $\mathbf{y}_i$  decreases as the eigenvalue decreases. As far as the negative weights  $g(\lambda_k)$  decrease, insignificant eigenvectors with inverted signs are amplified.

### 3.2 Aggregation

The process of individual patch MAP estimation terminates when all patches have been visited. A patch can be estimated multiple times using (6). The set of all MAP estimates of a patch  $\mathbf{x}_{\mathbf{k}}$  reads as  $X_{\mathbf{k}} := \{\mathbf{i} : \mathbf{k} \in S_1(\mathbf{i})\}$ . All these patch estimates are aggregated by averaging them on their corresponding locations on the image. We denote by  $H$  the local indexes within a  $\sqrt{K} \times \sqrt{K}$  patch,  $H = \{\mathbf{h} = (h_1, h_2) : 0 \leq h_1, h_2 \leq \sqrt{K} - 1\}$ . A pixel  $\mathbf{j}$  is contained in all the patches  $\mathbf{x}_{\mathbf{j}-\mathbf{h}}$  with  $\mathbf{h} \in H$ . The value of the oracle image  $\tilde{x}$  at pixel  $\mathbf{j}$  results from averaging all estimated values for that pixel:

$$\tilde{x}(\mathbf{j}) = \frac{1}{\mathcal{A}_{\mathbf{j}}} \sum_{\mathbf{h} \in H} \sum_{\mathbf{i} \in X_{\mathbf{j}-\mathbf{h}}} \tilde{\mathbf{x}}_{\mathbf{j}-\mathbf{h}}^{(\mathbf{i})}(\mathbf{h}), \quad \text{with} \quad \mathcal{A}_{\mathbf{j}} = \sum_{\mathbf{h} \in H} \# X_{\mathbf{j}-\mathbf{h}}. \quad (16)$$

There is a large variability in the aggregation weights across the image. Some patches have a high degree of self-similarity with their surroundings, and appear in many nearest neighbors sets  $S_1$ . Thus their aggregation weights  $\mathcal{A}_{\mathbf{i}}$  can be quite higher than those of other patches with singular patterns. Several patch based methods have a similar aggregation of the patch estimates [7, 19, 17], and it was identified in [13] as one of the key ingredients of state-of-the-art image denoising strategies.

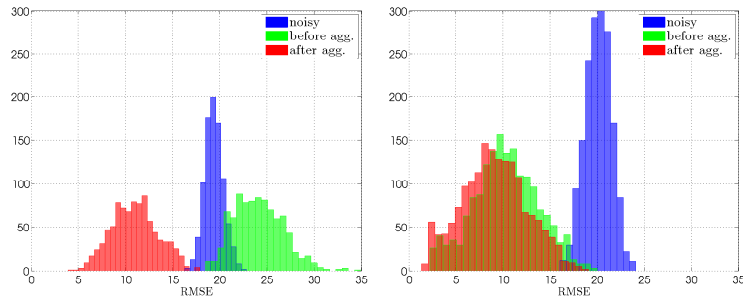
To evaluate the impact of the different stages of the denoising algorithm, we track the non-flat patches during the denoising of the *parrot* image with  $\sigma = 20$ . In Figure 3 we show the histograms of the RMSE of the non-flat patches in the input noisy image, (1) after the individual MAP estimates but before the aggregation, and (2) after their aggregation in the output oracle image. The RMSE before the aggregation (BA) is computed as

$$\text{RMSE}_{\text{BA}} = \left( \frac{1}{C} \sum_{\mathbf{i} \in J} \sum_{\mathbf{k} \in S_1(\mathbf{i})} \|\tilde{\mathbf{x}}_{\mathbf{k}}^{(\mathbf{i})} - \mathbf{x}_{\mathbf{k}}\|_2^2 \right)^{\frac{1}{2}}, \quad (17)$$

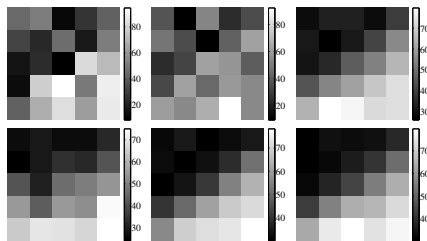
where  $C = \sum_{\mathbf{i} \in J} \# S_1(\mathbf{i})$  for  $J$  the set of non-flat patches. The RMSE after the aggregation is computed similarly, with the patch of the oracle  $\tilde{\mathbf{x}}_{\mathbf{k}}$  instead of  $\tilde{\mathbf{x}}_{\mathbf{k}}^{(\mathbf{i})}$ . The left figure corresponds to the first step of the NLB and the right to the second. It is seen that the MAP formula (6) in the first step gives quite a poor result, *with a RMSE worse than that of the noisy input image*. This is a consequence of the unreliable estimate of the covariance matrix. The aggregation stage has a crucial role to obtain the first-step denoised patch  $\tilde{\mathbf{x}}_{\mathbf{i}}$ . Most of the denoising work is done during the first step of the algorithm; see also Fig. 1.

Fig. 4 shows different stages of the denoising of a patch  $\mathbf{y}_{\mathbf{i}}$ : the MAP estimate (6), the aggregation stage in (16), followed by the second step MAP and the final denoised patch  $\hat{\mathbf{x}}_{\mathbf{i}}$ . Fig. 2 already showed that eigenvectors with small positive and also negative eigenvalues have a random structure. In Fig. 4, the significant contribution of such eigenvectors can be clearly seen: The step 1 MAP estimate before the aggregation is not better than the noisy patch. However, the aggregation of the MAP estimates yields a good approximation of the clean patch, very similar to the output after the second denoising iteration. Most of the denoising is carried out by the aggregation of the MAP estimates in the first stage.





**Fig. 3.** Histograms of the RMSE for non-flat patches, for the *parrot* image with  $\sigma = 20$ .



**Fig. 4.** Restoration of a noisy patch  $\mathbf{y}_i$  of the image in Fig. 1. Top: noisy patch  $\mathbf{y}_i$ , step 1 MAP  $\tilde{\mathbf{x}}_i^{(i)}$ , step 1 agg.  $\tilde{\mathbf{x}}_i$ . Bottom: original, step 2 MAP  $\hat{\mathbf{x}}_i^{(i)}$ , step 2 agg.  $\hat{\mathbf{x}}_i$ .

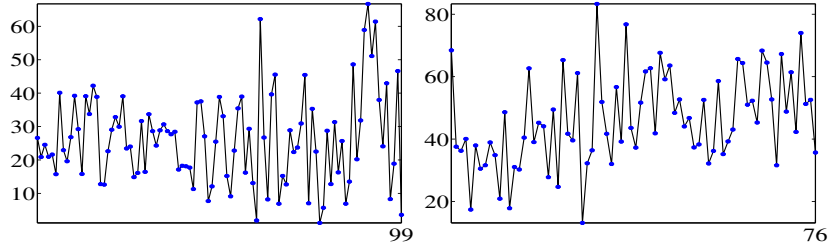
In Fig. 5 we show all the estimates that are aggregated for two pixels in the patch shown in Fig. 4. We refer to these plots as *pixel paths*. These values correspond to the MAP estimates (Eq. (6)) of patches that contain the pixels. It is quite a general rule that a pixel has a large number of updates. The variance of the updates is huge, which results from the negative weights in the pseudo-Wiener filter. The surprising fact is that in spite of this huge variance the final value is reasonable. By manipulating the values – removing outliers based on the histogram – one can get better results. Since the number of updates is not known, an automatic outlier rejection does not seem simple.

### 3.3 The effect of the thresholding the negative weights

In the light of the results after the first step MAP estimation (Figs. 3 and 4) one is curious to better realize the role of the negative eigenvalues in the estimated matrix  $C_{\mathbf{x}_i}$ . A common approach is to clip negative eigenvalues to zero [5]. Our aim here is to go into the question using a family of estimates of the covariance matrix by discarding small eigenvalues in  $C_{\mathbf{y}_i} - \sigma^2 I$  (see [2]):

$$C_{\mathbf{x}_i}^\tau = U_{\mathbf{y}_i} \text{diag} (T_\tau (\lambda_{\mathbf{y}_i} - \sigma^2)) U_{\mathbf{y}_i}^T, \quad (18)$$

where  $\tau \in \mathbb{R}$  and  $T_\tau(s) = s$  if  $s \geq \tau\sigma^2$  and  $T_\tau(s) = 0$  otherwise. We can then investigate the performance of the algorithm under variations of  $\tau$ .



**Fig. 5.** *Pixel paths* for two pixels in the patch  $\mathbf{y}_i$  shown in Fig. 4. These are the values that are aggregated according to (16).

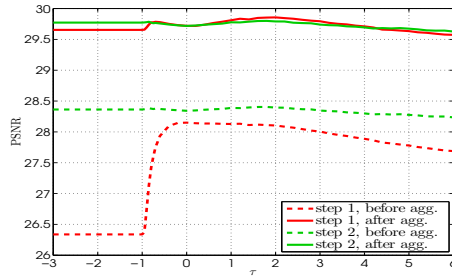
In Fig. 6 we test the estimators  $C_{\mathbf{x}_i}^\tau$  for different values of  $\tau$  and evaluate the PSNR on non-flat patches after the individual MAP estimation (dashed lines), and after the aggregation (continuous lines) for both denoising steps. As expected the plots become constant below  $\tau = -1$ , since all eigenvalues are larger than  $-\sigma^2$  ( $C_{\mathbf{y}_i}$  is positive definite). This is exactly the first-step MAP estimate (5) in NLB [12]. When  $\tau$  increases, more and more entries in the diagonal matrix in (18) are zeroed out, together with the corresponding filter coefficients  $g(\lambda_{\mathbf{y}_i})$ . For  $\tau$  large enough  $C_{\mathbf{x}_i} \rightarrow 0$  and the MAP denoising reduces to the  $\tilde{\mathbf{x}}_{\mathbf{k}}^{(i)} = \bar{\mathbf{y}}_i$ . If  $\tau = 0$  we recover estimator  $(C_{\mathbf{y}_i} - \sigma^2)_+$  of [5].

The plot confirms the bad quality of the MAP estimates in the first denoising iteration when  $\tau \leq -1$ , and the impressive correction caused by the aggregation. Regarding the output of the individual MAP estimates in the first step, the best results are obtained for  $\tau = 0$  (1.8dB higher than with  $\tau \leq -1$ ). However, this is not the case after the aggregation (continuous red curve), where slightly better results (around 0.1dB higher) correspond to  $\tau \approx 2$ , and also to  $\tau \approx -0.7$ . The PSNRs for the second step of the algorithm are almost independent of  $\tau$ . The negative eigenvalues in the first step do not have a negative effect on the performance of the whole algorithm. In fact, allowing some small negative eigenvalues seems slightly better than keeping all the positive eigenvalues. Although we show our analysis on a single example, we have found that this is the generally the case.

## 4 Second step denoising

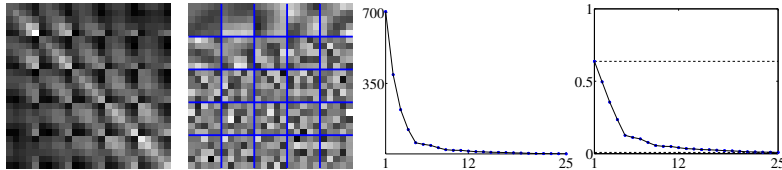
The second step denoising is essentially a Wiener filter where the *a priori* statistics are estimated using the oracle image resulting from the first step. In addition, the oracle image is also used to define the set of similar patches  $S_2(\mathbf{i})$ . As in the first step, following Remark 1, all patches in  $S_2(\mathbf{i})$  are denoised as proposed in (3). The *a priori* covariance matrix is estimated as the sample covariance matrix of the patches from the oracle image  $C_{\tilde{\mathbf{x}}_i}$ , which are assumed noiseless. The coefficients of the resulting Wiener filter matrix on the basis of eigenvectors  $U_{\tilde{\mathbf{x}}_i}$  have the well-known expression  $f(\lambda) = \lambda/(\lambda + \sigma^2)$ .

Fig. 7 shows a sample covariance matrix  $C_{\tilde{\mathbf{x}}_i}$  with its eigenvectors and eigenvalues, and the weights  $f(\lambda_{\tilde{\mathbf{x}}_i})$ . The covariance matrix corresponds to the reference patch in Fig. 2. The filter weights concentrate their mass on the largest



**Fig. 6.** PSNR of all non-flat patches after different stages of the algorithm, using the thresholded covariance matrices  $C_{\bar{x}_i}^T$ , as a function of  $\tau$ . See text for details.

eigenvalues, which correspond to smooth eigenvalues. This is contrary to what happens in the first step, as shown in Fig. 2. Thus, unlike to the the first step, eigenvectors with a random structure are much less present in the second step MAP estimates.



**Fig. 7.** For the same non-flat patch as in Fig. 2 we show, from left to right: sample covariance matrix  $C_{\bar{x}_i}$ ; its eigenvectors in raster order; the eigenvalues  $\lambda_{\bar{x}_i}$  and the resulting weighting function  $f(\lambda_{\bar{x}_i})$ . Compare to the weights in Fig. 2.

The final denoised image is obtained by aggregation as described in subsection 3.2. Although the second step has a contribution to the quality of the final result, arguably most of the denoising is done in the first step, as can be seen by the Figures 1, 3 and 4.

## 5 Discussion

Several patch-based methods for image denoising follow Bayesian approaches with a Gaussian (or GMM) prior for patches. The validity of the Gaussian model is questionable, but these methods produce state-of-the art results and are appealing due to their simplicity. In the case of NLB, the indefinite prior covariance estimate causes poor individual patch MAP estimates. However, they systematically lead to sets of pixel estimates whose mean (aggregate value) is visibly consistent. A good understanding of the NLB algorithm requires models that include the aggregation stage.

## References

1. Aguerrebere, C., Almansa, A., Gousseau, Y., Musé, P.: A hyperprior bayesian approach for solving image inverse problems. HAL (2016)
2. Arias, P., Morel, J.M.: Video denoising via empirical Bayesian estimation of space-time patches (Jan 2017), [http://dev.ipol.im/~pariasm/video\\_nlbayes/vnlb\\_files/vnlb\\_preprint.pdf](http://dev.ipol.im/~pariasm/video_nlbayes/vnlb_files/vnlb_preprint.pdf), preprint
3. Aubert, G., Kornprobst, P.: Mathematical problems in image processing. Springer-Verlag, Berlin, 2 edn. (2006)
4. Buades, A., Lisani, J.L., Miladinović, M.: Patch-based video denoising with optical flow estimation. *IEEE Transactions on Image Processing* 25(6), 2573–2586 (June 2016)
5. Chatterjee, P., Milanfar, P.: Patch-based near-optimal image denoising. *IEEE Trans. Image Process.*21(4), 1635–1649 (2012)
6. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: BM3D image denoising with shape-adaptive principal component analysis. In: *Proceedings of the Workshop on Signal Processing with Adaptive Sparse Structured Representations*. pp. 1–7 (2009)
7. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising by sparse 3d transform-domain collaborative filtering. *IEEE Trans. on IP* 16(8), 2080–2095 (2007)
8. Deledalle, C.A., Salmon, J., Dalalyan, A.: Image denoising with patch based pca: local versus global. In: *Proceedings of the British Machine Vision Conference*. pp. 25.1–25.10 (2011)
9. Guillemot, T., Almansa, A., Boubekur, T.: Covariance trees for 2D and 3D processing. In: *IEEE Conference on Computer Vision and Pattern Recognition*. pp. 556–563 (2014)
10. Laus, F., Nikolova, M., Persch, J., Steidl, G.: A nonlocal denoising algorithm for manifold-valued images using second order statistics. *SIAM J. Imaging Sci.*(2017)
11. Lebrun, M., Buades, A., Morel, J.M.: Implementation of the “nonlocal Bayes” (nl-bayes) image denoising algorithm. *Image Processing On Line* 3 (2013)
12. Lebrun, M., Buades, A., Morel, J.M.: A nonlocal Bayesian image denoising algorithm. *SIAM J. Imaging Sci.*6(3), 1665–1688 (2013)
13. Lebrun, M., Colom, M., A. Buades, A., Morel, J.M.: Secrets of image denoising cuisine. *Acta numerica* 21 (Apr. 2012)
14. Lebrun, M., Colom, M., Morel, J.M.: The noise clinic: a blind image denoising algorithm. *Image Processing Online (IPOL)* 5 (Jan. 2015)
15. Muresan, D.D., Parks, T.W.: Adaptive principal components and image denoising. In: *Proceedings of the IEEE International Conference on Image Processing*. vol. 1, pp. 1–101–4 vol.1 (Sept 2003)
16. Yaroslavsky, L.: *Theoretical Foundations of Digital Imaging Using MATLAB*. CRC Press (2013)
17. Yu, G., Sapiro, G., Mallat, S.: Solving inverse problems with piecewise linear estimators: From Gaussian mixture models to structured sparsity. *IEEE Trans. Image Process.*21(5), 2481–2499 (2012)
18. Zhang, L., Dong, W., Zhang, D., Shi, G.: Two-stage image denoising by principal component analysis with local pixel grouping. *Pattern Recognition* 43(4), 1531–1549 (2010)
19. Zoran, D., Weiss, Y.: From learning models of natural image patches to whole image restoration. In: *Computer Vision (ICCV)*. pp. 479–486. *IEEE International Conference on* (Nov. 2011)