

REVIEW

Open Access



Patch-based models and algorithms for image denoising: a comparative review between patch-based images denoising methods for additive noise reduction

Monagi H. Alkinani¹ and Mahmoud R. El-Sakka^{2*}

Abstract

Background: Digital images are captured using sensors during the data acquisition phase, where they are often contaminated by noise (an undesired random signal). Such noise can also be produced during transmission or by poor-quality lossy image compression. Reducing the noise and enhancing the images are considered the central process to all other digital image processing tasks. The improvement in the performance of image denoising methods would contribute greatly on the results of other image processing techniques. Patch-based denoising methods recently have merged as the state-of-the-art denoising approaches for various additive noise levels. In this work, the use of the state-of-the-art patch-based denoising methods for additive noise reduction is investigated. Various types of image datasets are addressed to conduct this study.

Methods: We first explain the type of noise in digital images and discuss various image denoising approaches, with a focus on patch-based denoising methods. Then, we experimentally evaluate both quantitatively and qualitatively the patch-based denoising methods. The patch-based image denoising methods are analyzed in terms of quality and computational time.

Results: Despite the sophistication of patch-based image denoising approaches, most patch-based image denoising methods outperform the rest. Fast patch similarity measurements produce fast patch-based image denoising methods.

Conclusion: Patch-based image denoising approaches can effectively reduce noise and enhance images. Patch-based image denoising approach is the state-of-the-art image denoising approach.

Keywords: Patch-based image denoising, Bilateral filter, Non-local means filtering, Probabilistic patch-based filtering, Dictionary learning filtering, K-SVD, Gaussian patch-PCA filtering, BM3D

1 Review

1.1 Introduction

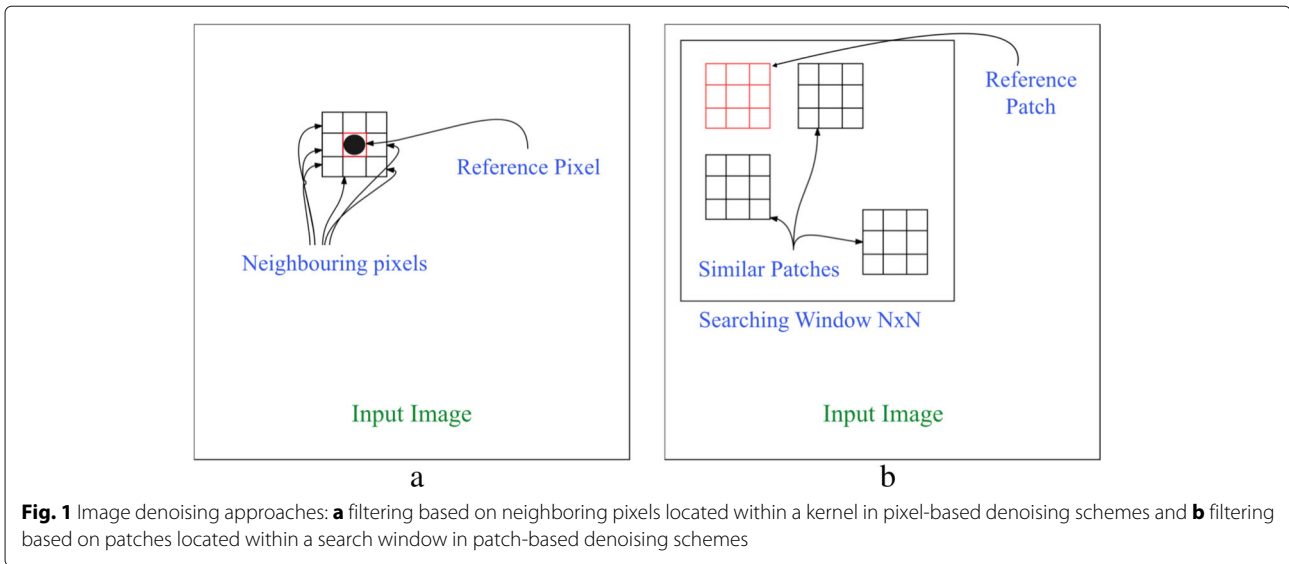
The noise level in digital images may vary from being almost imperceptible to being very noticeable. Image denoising techniques attempt to produce a new image that has less noise, i.e., closer to the original noise-free image. Image denoising techniques can be grouped into two main approaches: pixel-based image filtering and patch-based

image filtering. A pixel-based image filtering scheme is mainly a proximity operation used for manipulating one pixel at a time (pixel-wise) based on its spatial neighboring pixels located within a kernel. On the other hand, in patch-based image filtering, the noisy image is divided into patches, or “blocks,” which are then manipulated separately in order to provide an estimate of the true pixel values (patch-wise) based on similar patches located within a search window. This approach utilizes the redundancy and the similarity among the various parts of the input image. Figure 1 shows the mechanism of the two approaches.

*Correspondence: elsakka@csd.uwo.ca

²Department of Computer Science, Middlesex College, Western University, 1151 Richmond Street, N6A 5B7, London, Ontario, Canada

Full list of author information is available at the end of the article



It is now common in image denoising field to utilize patch-based models and algorithms instead of pixel-based approaches to produce most promising estimate of the noise-free images. However, there are both advantages and disadvantages in the use of patch-based models and algorithms. There are several advantages of patch-based approaches. Smoothing flat regions is the most important aspect. Redundancy between patches enable patch-based approaches to properly smooth flat reigns. A second advantage of using patch-based models and algorithms approaches is that it can preserve fine image details and sharp edges. However, there could be some disadvantages for patch-based models and algorithms. First, although similarity between patches assists in estimating flat regions, so is the averaging. It is, therefore, quite time-consuming to group and compare similar patches. This might mean that each patch has multiple estimates and patches are overlapped. Secondly, while it may be that patterns and textures seem clear with less noise, patch-based models and algorithms usually exploit large number of parameters, which can be extremely difficult to adjust properly. We believe that the advantages of patch-based methods far outweigh their disadvantages, as modern computers are significantly fast, and have large memory spaces.

In this work, the patch-based image denoising schemes are analyzed from two different aspects: (1) the performance of patch-based denoising techniques in terms of image denoising quality and (2) the performance of patch-based denoising techniques in terms of computational time, where various patch-based denoising techniques are addressed.

A literature survey was conducted to evaluate the most recent patch-based denoising improvements for additive

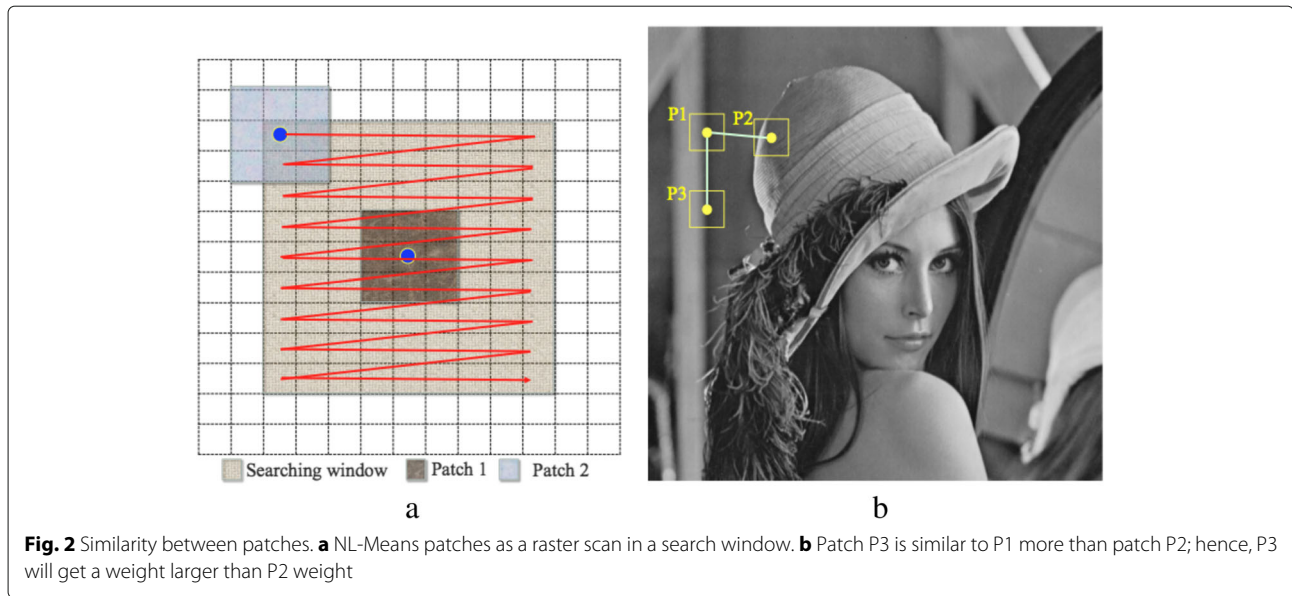
noise. Following the literature survey, there is an empirical study, which is used to evaluate the performance of various patch-based denoising techniques in terms of their accuracy and run times at various noise levels.

2 Patch-based image filtering

In patch-based denoising techniques, the input noisy image is divided into patches (i.e., blocks). The blocks are then manipulated separately in order to provide an estimate of the true pixel values. In this section, various patch-based image denoising algorithms are presented and their efficiency with respect to image denoising are studied.

2.1 Averaging patch-based: non-local means

Non-local means (NL-Means) is a patch-based filter proposed by Buades et al. [6] as a modification to the pixel-wise bilateral filter [4, 16, 38, 50, 58, 61]. Like the bilateral filter, the NL-Means filter blurs the homogeneous areas and preserves edges. The NL-Means filter divides the input image into sub-images and then filters each sub-image separately in a technique that is referred to as being patch-wise. Each sub-image contains several patches. As in the bilateral filter, similarity is measured based on two measurements: (1) the Euclidean distance between the centers of the patches and (2) the luminance distance between the patches. In contrast to the bilateral filter, patches are compared within a search window instead of with the pixels of the neighbors. This is why it is called a non-local method. Patches with similar gray levels have larger weights when they are averaged. Figure 2a shows the NL-Means patches and how to find the similar patches in a raster scan in a search window. Figure 2b illustrates the fact that patches with a similar gray level, for example,



P1 and P3, should be assigned a larger weight than that to be assigned to P2. The edges in NL-Means filtering are preserved regardless of their direction.

The estimated value NL-Means $[v]_i$, for a pixel i , is computed as:

$$\text{NL-Means}[v]_i = \sum_{j \in I} \omega(i, j) [v]_j, \quad (1)$$

where $[v]_i$ and $[v]_j$ are pixel intensities at locations i and j , respectively, and $\omega(i, j)$ is a similarity measure between the pixels i and j . The similarity measure weight satisfies the condition $0 \leq \omega(i, j) \leq 1$ and $\sum_j \omega(j, i) = 1$. The similarity weight depends on the gray level similarity and the Euclidean distance between vectors $N[v]_i$ and $N[v]_j$, where $N[v]_k$ denotes a square neighborhood of fixed size and centered at a pixel k . The weights are described as:

$$\omega(i, j) = \frac{1}{Z(i)} e^{-\frac{\|N[v]_i - N[v]_j\|^2}{h^2}}, \quad (2)$$

where $Z(i)$ is a normalization factor and h is a filtering parameter set depending on the noise level.

The level of noise determines the sizes needed for the search window and patches. For a robust comparison between patches, the size of the patches increases when the noise level is high. Accordingly, the value of the filtering parameter h increases as the size of the patch is increased. Meanwhile, the size of the search window must be increased in order to find more similar patches.

NL-Means filter is considered the cornerstone of many patch-based denoising methods. It can be adapted easily to many other applications, such as, multi-view image denoising. Nevertheless, there are some disadvantages of NL-Means filter. The filter is computationally expensive

due to the large amount of weight computations between similar patches. Another disadvantage is that the NL-Means filter is a spatial domain filter though convolution can be easily implemented in the frequency domain.

NL-Means filter has too many modifications. Improving the way of assigning the weights between patches would improve the performance of the NL-Means method. Hedjam et al. [25] improved the process of adjusting the weights in the NL-Means by using Markovian clustering. Wu et al. [63] used a statistical shrinkage perspective when assigning the weights in NL-Means via using James-Stein [30] shrinkage estimator. Lai and Dou [35] introduced an improved neighborhood pre-classification strategy for optimized weight kernels of NL-Means filter. Khan and El-Sakka [32] introduced a variant of the NL-Means scheme by using a thresholding step to reduce the number of similar patches before weight averaging the patches.

NL-Means is a spatial domain filter, transferring this filter to the frequency domain would help in suppressing more noisy signal. Enríquez and Ponomaryov [17] transferred the patches of the NL-Means to the frequency domain and used discrete cosine transform with a threshold to estimate true patches. Zhong et al. [69] combined the NL-Means with Lee filter [36] for SAR image enhancing. Chan et al. [7] incorporated a median filtering operation indirectly in the NL-Means method for denoising low signal noise ratio (SNR) images. Maruf and El-Sakka [43] projected NL-Means patches into a global feature space before performing a statistical t test to reduce the dimensionality of this feature space. They gathered similar patches globally. Irrera et al. [28] adapted NL-Means for denoising X-ray images (XNL-Means), and then they applied an additional multi-scale contrast enhancement in the frequency domains.

NL-Means filter could be adapted to improve other image processing applications (e.g., segmentation, recognition, and video denoising). Zhan et al. [66] introduced an extension to the NL-Means method for ultrasonic speckle reduction. They assigned the patch similarity weights iteratively in a lower dimensional subspace using principal component analysis (PCA). Xu et al. [65] adapted the NL-Means to be used for microscopy cell images via a frequency transform. Genin et al. [20] adapted a modified version of the NL-Means filter for detecting small objects by background suppression. Background pixels are estimated by a weighted average depending on the similarity between neighborhoods pixels. Kim et al. [33] adapted the NL-Means filter for noise reduction and enhancement of extremely low-light video. They use a motion adaptive temporal filter using gamma correction with adaptive thresholds before the NL-Means filter. Xu et al. [64] adapted the idea of patching from the NL-Means for filtering polarimetric synthetic aperture radar (POL-SAR) images; they use simultaneous sparse coding for transferring the patches into the frequency domain before assigning the weights.

2.2 Probabilistic patch-based filter

The probabilistic patch-based (PPB) filter, which works in the spatial domain, was proposed by Deledalle et al. [12] as an extension of the NL-Means filter. The PPB approach is one of a few denoising techniques that can provide a general denoising methodology for various noise models. Thus, it is more general than the NL-Means and can be applied where there is additive noise or multiplicative speckle noise. The PPB filter is a statistically based similarity scheme that depends on the distribution model of the noise. The weighted average is used for the Gaussian noise distribution in the NL-Means, but the PPB filter applies smoothing based on the maximum likelihood estimator (MLE). The PPB is expressed as a weighted maximum likelihood estimation (WMLE) problem. The weight is derived from the data by improving the isotropy of the filter—non-iterative probabilistic patch-based filter (Non-iPPB)—and it can be iteratively defined based on the similarity of the patches—iterative probabilistic patch-based filter (It-PPB).

2.2.1 Weighted maximum likelihood estimator

Using a weighted maximum likelihood estimation for image denoising is not a new technique; it was first used for image denoising by Polzehl and Spokoiny [51, 52]. PPB redefined the weights in term of a patch-based approach. PPB image denoising is considered to be an estimation \hat{u} of the true image u which originates from the noisy image v . The images are defined over a discrete regular grid Ω . A pixel value is described as i , and its neighbor is j at the location $(x, y) \in \Omega$. The noise model is considered as being

defined by the parametric noise distribution “likelihood” $p(i | \theta_i^*)$, where θ_i^* is an unknown space-varying parameter. The denoising of an image is equivalent to finding the best estimation $\hat{\theta}_i$ for θ_i^* for all pixels. The MLE at each location (x, y) estimates $\hat{\theta}_i$ from a set $S_{\theta_i^*}$ of the distributed random variables around it by:

$$\hat{\theta}_i \triangleq \arg \max_{\theta_i} \sum_{j \in S_{\theta_i^*}} \log p(j | \theta_i), \quad (3)$$

$$\triangleq \arg \max_{\theta_i} \sum_j \delta_{S_{\theta_i^*}}(j) \log p(j | \theta_i),$$

where $\delta_{S_{\theta_i^*}}$ is an indicator function for $S_{\theta_i^*}$ (i.e., $\delta_{S_{\theta_i^*}} = 1$ if $j \in S_{\theta_i^*}$ or 0 otherwise). The indicator function has been derived from the data as weights $\omega(i, j) \geq 0$ in Polzehl and Spokoiny [52] and Polzehl and Tabelow [51], where it is used as weight for adaptive pixel-wise filters. However, the indicator function in PPB is used as weight function to form the WMLE:

$$\hat{\theta}_i \triangleq \arg \max_{\theta_i} \sum_j \omega(i, j) \log p(j | \theta_i). \quad (4)$$

2.2.2 Defining the weight between patches

In Subsection 2.1, the weights in the non-local means filter are defined by comparing the similarities of the two patches $[v]_i$ and $[v]_j$ centered around the two locations i and j , respectively. A weighted Euclidian distance between the two patches defines the level of similarity. The objective of the PPB filter is to generalize and extend the idea of the Euclidean distance weight used in the non-local means filter so that it can be adapted to non-additive noise models. The weights used in the probabilistic patch-based method are estimated using the probability of the two patches in a noisy image having the same parameters. By following the same idea as the weight in the non-local means filter and assuming equal values for i and j in the two statically similar patches $[v]_i$ and $[v]_j$, PPB weights would be defined as:

$$\omega(i, j)^{(\text{PPB})} \triangleq p\left(\theta_{[v]_i}^* = \theta_{[v]_j}^* | v\right)^{1/h}, \quad (5)$$

where $\theta_{[v]_i}^*$ and $\theta_{[v]_j}^*$ are the patches extracted from the image θ^* and h is larger than 0, which indicates the size of the patch in PPB. The h acts as σ in the NL-Means algorithm in order to control the filtering amount. The probability of the similarity of the patches pixels is decomposed into a product of the probabilities of k neighbors: $\prod_k p\left(\theta_{i,k}^* = \theta_{j,k}^* | v_{i,k}, v_{j,k}\right)$.

2.2.3 Iterative WMLE denoising

In order to improve the performance of the PPB algorithm, the probability of a similarity is estimated iteratively. The weight, at each iteration, is expressed as the

product of two terms: (1) the probability of the similarity between the noisy patches as described in Subsection 2.2.2 and (2) the probability of the similarity derived from the previous iteration.

Assume that the previous estimation at t iteration is $\hat{\theta}^{t-1}$ for θ^* . Then, the formula in Eq. 5 can be expressed as:

$$\omega(i, j)^{(It-PPB)} \triangleq p\left(\theta_{[v]_i}^* = \theta_{[v]_j}^* \mid v, \hat{\theta}^{t-1}\right)^{\frac{1}{h}}. \quad (6)$$

This is similar to what was achieved in Subsection 2.2.2, where the probability of the similarity is decomposed into a product of the probabilities of the k neighbors: $\prod_k p\left(\theta_{i,k}^* = \theta_{j,k}^* \mid v_{i,k}, v_{j,k}, \hat{\theta}^{t-1}\right)$.

From the Bayesian framework, the naïve Bayes model can be fitted with the maximum likelihood concept. The probability is estimated using the prior probability and can be presumed to be proportional to the likelihood:

$$p\left(v_{i,k}, v_{j,k}, \hat{\theta}^{t-1} \mid \theta_{i,k}^* = \theta_{j,k}^*\right). \quad (7)$$

The similarity likelihood is computed using:

$$\underbrace{p\left(\theta_{i,k}^* = \theta_{j,k}^* \mid v_{i,k}, v_{j,k}, \hat{\theta}^{t-1}\right)}_{\text{likelihood}} \propto \underbrace{p\left(v_{i,k}, v_{j,k} \mid \theta_{i,k}^* = \theta_{j,k}^*\right)}_{\text{likelihood}} \times \underbrace{p\left(\theta_{i,k}^* = \theta_{j,k}^* \mid \hat{\theta}^{t-1}\right)}_{\text{prior}}. \quad (8)$$

The likelihood term is to compute the degree of similarity between the patches, and the prior term is to compare the two probability distributions from the previous iteration, similar to Polzehl and Spokoiny [51].

The scheme, Fig. 3, shows the procedure of iteratively competing the weights in the PPB algorithm. The procedure for defining the weights is estimated iteratively by (1) the PPB weights estimator (PPBWE) uses the likelihood term and the estimated value from the previous iteration in order to compute the prior term (Eq. 8), (2) WMLE uses the PPBWE estimation and the noisy image in order to estimate the new weight (Eq. 4), and (3) the

PPBWE and the WMLE steps are repeated until there is no difference in the estimations made from the two steps.

2.2.4 Algorithm used in the case of Gaussian noise

The PPB filter can be used for filtering additive white Gaussian noise (AWGN). By assuming the AWGN model, the values of the pixels I of the patch $[v]_i$ are distributed based on the Gaussian distribution $\aleph(u, \sigma^2)$. Here, u is the noiseless image and σ the noise variance. The noiseless image u can be estimated by the weighted average that maximizes the WMLE defined in Eq. 4:

$$\ddot{u}_i^{(WMLE)} = \frac{\sum_j \omega(i, j) I_j^2}{\sum_j \omega(i, j)}. \quad (9)$$

In order to estimate the weighted average $\omega(i, j)$, two terms are considered: the likelihood and the prior terms. The likelihood function is discretized as:

$$\underbrace{p\left(I_{i,k}, I_{j,k} \mid \ddot{u}_{i,k} = \ddot{u}_{j,k}\right)}_{\text{likelihood}} \propto \exp\left(-\frac{|I_i - I_j|^2}{4\sigma^2}\right), \quad (10)$$

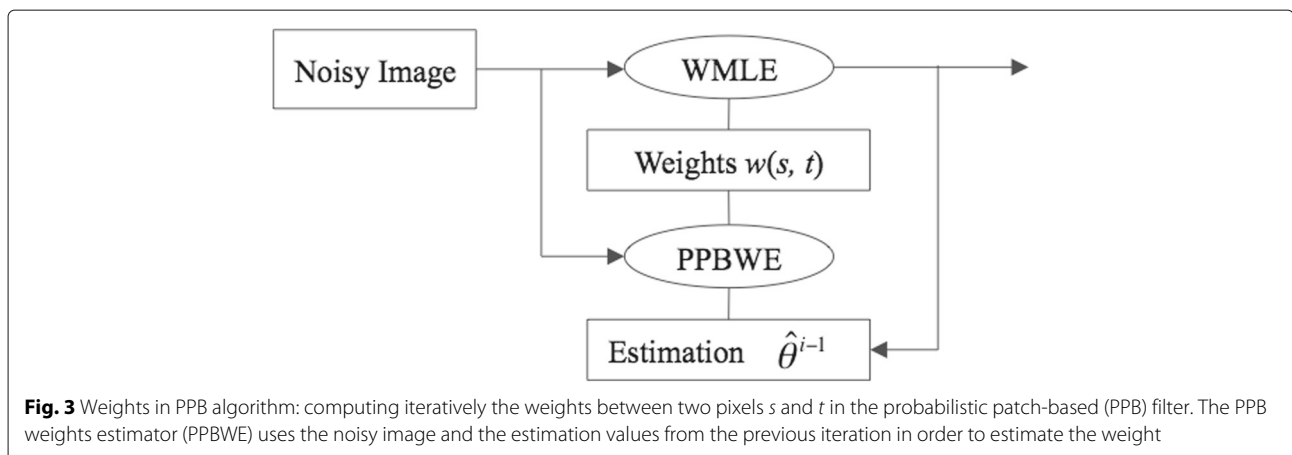
and the prior term is discretized as:

$$\underbrace{p\left(\ddot{u}_{i,k} = \ddot{u}_{j,k} \mid \ddot{u}^{t-1}\right)}_{\text{prior}} \propto \exp\left(-\frac{1}{T} \frac{|\ddot{u}_{i,k}^{t-1} - \ddot{u}_{j,k}^{t-1}|^2}{\sigma^2}\right). \quad (11)$$

By combining the two terms in Eqs. 10 and 11, the weight at any iteration is defined as:

$$\omega(i, j)^{(It-PPB)} = \exp\left[-\sum_n \left(\frac{1}{h} \frac{|I_i - I_j|^2}{4\sigma^2} + \frac{1}{T} \frac{|\ddot{u}_{i,k}^{t-1} - \ddot{u}_{j,k}^{t-1}|^2}{\sigma^2}\right)\right], \quad (12)$$

where n is number of pixels and T is a constant similar to h in Eq. 2. When there is no iteration “posterior term = 0,” the filter performs similar to the NL-Means filter.



PPB filter has several advantages and disadvantages. The main advantage of PPB filtering is that it is a statistical-based approach, which can be utilized for suppressing additive Gaussian noise and/or multiplicative speckle noise. Also, this filter can be adjusted as an iterative filter or not. Nevertheless, the main drawback of this filter is the suppression of fine and dark details when denoising. Another disadvantage of the filter is the high computational cost when used as an iterative filter.

A further extend patch log likelihood (EPLL) filter, similar to probabilistic patch-based filter, was proposed recently by Pappas and Elad [47] via considering a multi-scale prior.

2.3 Dictionary learning

Dictionary learning (DL) is utilized as a replacement for the use of a fixed dictionary for representing data. From the 1970s, data can be represented by using a fixed dictionary, for instance, Fourier of Boussinesq [5] and Wavelets of Haar [23]. In 1996, Olshausen and Field [45] proposed an approach to learn the dictionary from a data in order to optimize the sparsity of the data. The dictionary learning or k-means singular value decomposition (K-SVD) was first adapted to image denoising in 2006 by Aharon et al. [1]. Dictionary learning method finds the best dictionary $D = (d_i)_{i=1}^z$ of z atoms $d_i \in \mathbb{R}^n$ that sparsifies the set $Y = (y_j)_{j=1}^m \in \mathbb{R}^{n \times m}$ of signals $y_j \in \mathbb{R}^n$. In order to filter a noisy image, each signal y_j is considered as a patch extracted from the noisy image. The sparse code of signal data $y = y_j$ for $j = 1, \dots, n$ is obtained by minimizing a constrained optimization ℓ^0 :

$$\min_{\|x\|_0 \leq k} = \frac{1}{2} \|y - Dx\|^2, \quad (13)$$

where $k > 0$ controls the amount of sparsity, and ℓ^0 pseudo-norm is defined by:

$$\|x\|_0 = \{i : x_i \neq 0\}. \quad (14)$$

In dictionary learning, optimization is performed on the dictionary D and the coefficients $X = (x_j)_{j=1}^m \in \mathbb{R}^{p \times m}$ for $j = 1, \dots, n$, where the set of coefficients is x_j of the data y_j . The joint optimization is written as:

$$\arg \min_{D \in \Phi, X \in \chi_k} E(X, D) = \frac{1}{2} \|Y - DX\|^2 = \frac{1}{2} \sum_{j=1}^m \|y_j - Dx_j\|^2, \quad (15)$$

where Φ is the constraint set:

$$\Phi = \{D \in \mathbb{R}^{n \times p} : \forall i \|D_{:,i}\| \leq 1\}. \quad (16)$$

The sparsity constraint is set on χ_k , which is the unit normalization of the dictionary columns:

$$\chi_k = \{X \in \mathbb{R}^{p \times m} : \forall i \|X_{:,i}\|_0 \leq k\}. \quad (17)$$

Peyré and Fadili [49] proposed using a block-coordinate descent minimization approach used by Tseng [59] in order to minimize X and D .

The dictionary learning algorithm depends mainly on three steps: (1) patch extraction, (2) sparse coding, and (3) patch construction. In the first step, several patches are randomly extracted from the whole input image. In the sparse coding step, the energies of the X and D dictionaries are iteratively minimized. Patch averaging and reconstruction occurs in the patch reconstruction step. Each of the three steps has stages:

1. Patch extraction step:

- The mean of each patch is removed from each pixels value.
- Patches are sorted based on their energy, those with a high level of energy are kept by thresholding.
- Patches are reshaped as columns in order to form Y .

2. Sparse coding step:

- Each atom in the columns is normalized in order to form the initial dictionary D .
- Number of the columns is reduced again, before computing the X coefficients.
- The X dictionary is initially started with zeros.
- The coefficients of dictionary X is updated by:

$$X = T(X - \gamma D'((D - X) - Y)),$$

where T is the thresholding.

- The dictionary D is updated again by:

$$D = T(D - \gamma((DX - X) - X')).$$

- K is the number of iterations, which is used to minimize the X and D dictionaries by updating them iteratively.
- Finally, the coefficients of dictionary X are multiplied by dictionary D .

3. Patch construction step:

- The result of the multiplication is a new array, $Y1$.
- The columns of $Y1$ are reshaped to form patches.
- Re-inserting the averages to the patches comes before averaging the patches to replace the noisy patch.

The scheme in Fig. 4 shows the steps of the dictionary learning method for denoising images.

The most important advantage of the DL method is the sparse representation of the input data. One advantage

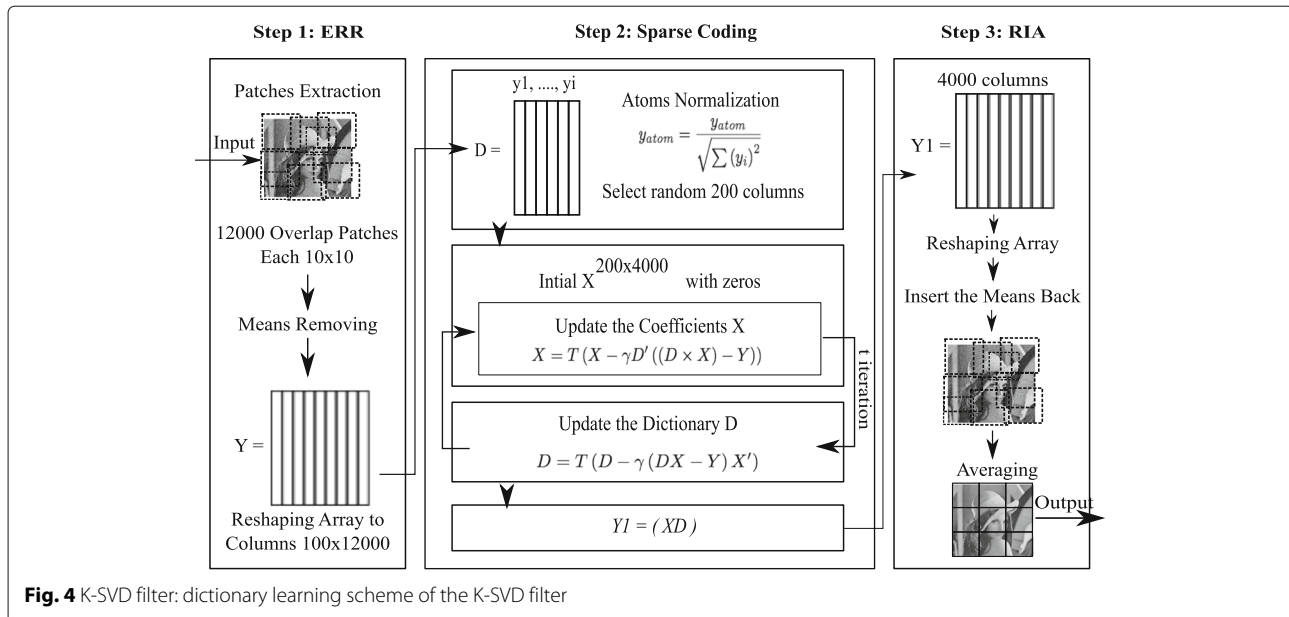


Fig. 4 K-SVD filter: dictionary learning scheme of the K-SVD filter

of the DL is the low number of parameters when compared to other patch-based denoising methods. However, DL has some drawbacks. The main disadvantage of DL is the computational burden due to several iterations for computing the singular value decomposition. Moreover, DL does not handle images with many flat patches very well because the singular value decomposition is more sensitive to textured patches.

The DL method has many modifications. Tian and Wang [57] made DL more sparsely representative in the case of less observation values by proposing an adaptive orthogonal matching pursuit to adaptively ensure the sample size. Some of the modifications aim to adapt the idea of denoising based on dictionary learning to other image processing applications. Chen et al. [10] generalized the idea of the learning dictionary to explore identity information in multiple frames of videos. They generated a sparse representation from multiple video frames for face and body part recognitions. Fu et al. [19] proposed an effective model based on DL for hyperspectral image (HSI) denoising by considering sparsity across the spatial-spectral domain, high correlation across spectra, and non-local self-similarity over space. Kang et al. [31] proposed a feature-based approach for assessing similarity between images. After extracting feature points from an image, they utilize dictionary learning. Then, they measured the similarity between images in terms of sparse representation. A novel self-learning based image decomposition framework was presented by Huang et al. [27]. Their framework performs unsupervised clustering on the observed dictionary via affinity propagation that allows effectively to identify images components with similar context information. The framework can

automatically determine the undesirable random noisy components from true image components directly from a noisy image. Dictionary learning algorithm was adapted to filter Chinese character images by Shi et al. [54]. They divided the image frequency to low and high frequencies. While a Butterworth low-pass filter was utilized to filter low frequency, the K-SVD dictionary learning algorithm was proposed to filter high-frequency parts which consists of structure of Chinese characters.

2.4 Patch-based PCA

Recently, over-complete dictionaries with sparse representation techniques became very widespread in image denoising [1, 40, 41], and they are one of the state-of-the-art denoising algorithms. These methods use over-complete dictionaries derived from enormous image sets or from the noisy image itself. They outperform other denoising techniques due to their ability to provide an appropriate basis for separating noisy signals from the true image signals, so they suppress more noise and preserve edges. Despite the fact that over-complete dictionaries are frequently used for image denoising, such dictionaries are sophisticated and quite expensive in terms of memory usage and time. However, patch-based principal component analysis (PB-PCA) of Deledalle et al. [13] is a modification of the dictionary methods.

PB-PCA uses simple orthogonal dictionaries constructed by using principal component analysis. The results of PB-PCA still do not outperform the over-complete dictionary methods, but PB-PCA shows how simple orthogonal dictionaries can achieve excellent results with less sophistication. PB-PCA simply learns the orthogonal dictionaries from the noisy image via principal

component analysis. The next step is to threshold the patches' coefficients in the dictionaries. This idea is similar to the wavelet denoising methods used in [8, 9, 14], in which they use either hard thresholding or soft thresholding for zeroing the coefficients. Figure 5 shows extracting patches used in the PB-PCA method from an image and grouping them before computing the PCA.

For the problem of denoising an image that is interrupted by AWGN, the patch model has the following formula:

$$[v]_i = [u]_i + z_i, i = 1, \dots, n - 1, \tag{18}$$

where $[u]_i$ is the true image patch, z_i is the AWGN noise, $[v]_i$ is the noisy patch, and n is the number of patches. By assuming $[v]_1, \dots, [v]_{n-1}$ are a group of overlapped patches of size $N \times N$ extracted from the noisy image v , the covariance matrix is the sum of:

$$\sum = \frac{1}{n} \sum_{i=1}^n [v]_i [v]_i' - \bar{v} \bar{v}', \tag{19}$$

where

$$\bar{v} = \frac{1}{n} \sum_{i=1}^n [v]_i.$$

In PCA, the singular value decomposition (SVD) of the covariance matrix \sum is processed. Moreover, the eigenvalues g_1, \dots, g_{n-1} of the covariance matrix and the corresponding eigenvectors G_1, \dots, G_{n-1} are calculated. Eigenvectors are called the principal components "axis" of the processed data and are used to form an orthogonal basis, G_i is the i th principal axis of the data. Due to the orthogonal basis of the principal components, an image patch can be decomposed as $[v]_i = \sum_{j=1}^n \langle [v]_i | G_j \rangle G_j$. Figure 6b, c shows the first and last 16 principal axes of the all patches obtained from the house image shown in Fig. 6a.

By assuming that the true image pixels have a low-dimensional subspace and the noise is spread in all directions, projecting the axes into the first axis would suppress the noise in the noisy image. Projecting the axes is called

coefficient thresholding, and it is done by using an appropriate shrinkage function. A general formula for estimating a true image is:

$$[\hat{u}]_i = \bar{v} + \sum_{i=1}^n \eta(\langle [v]_i - \bar{v} | G_i \rangle) G_i, \tag{20}$$

where η is the shrinkage function.

PB-PCA filtering method has been tested with four shrinkage functions: (1) soft thresholding (ST); (2) hard thresholding (HT); (3) Keep or Kill (KoK), and (4) Wiener filter [62]. Figure 7 shows a comparison between the four different projection methods into the basis by PCA for the House and Cameraman images. From Fig. 7, it can be seen that hard thresholding with a large number of axes is the best of the four projection methods. Using a Wiener filter as a shrinkage function for the PCA has been proposed by Muresan and Parks [44, 67]. The Zhang et al. [67] algorithm will be discussed in Subsections 2.4.1.

PB-PCA has three variants based on how each set of patches is collected from the input noisy image before the PCA process. In PB-PCA, the three variants for collecting patches are globally, locally, or hierarchically. Figure 8 shows the best means of collecting the patch sets globally, locally, or hierarchically. These variants are discussed in the following subsections.

2.4.1 Patch-based global PCA

Collecting patches for a PCA can be done globally from the entire noisy image in patch-based global PCA (PGPCA). This approach is faster than the other approaches, which need time to divide the image into sub-images before computing the PCA coefficients. However, it has less filtering quality. Collecting patches globally for PCA filtering has been used in Bacchelli and Papi [2] and Zhang et al. [67]. The Bacchelli and Papi algorithm uses a linear transformation, "a wavelet transform," before computing the PCA in order to achieve better results. The Zhang et al. algorithm collects patches globally, but it computes the PCA in two stages.

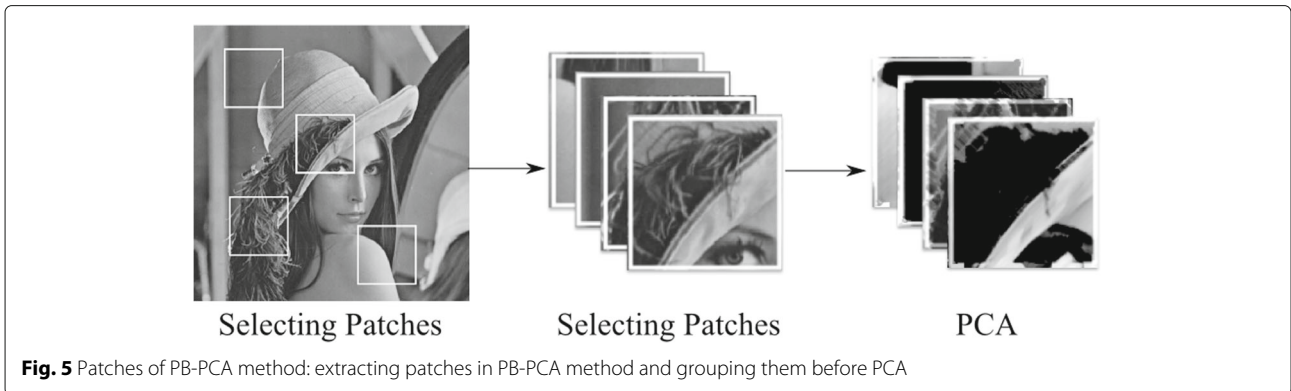


Fig. 5 Patches of PB-PCA method: extracting patches in PB-PCA method and grouping them before PCA

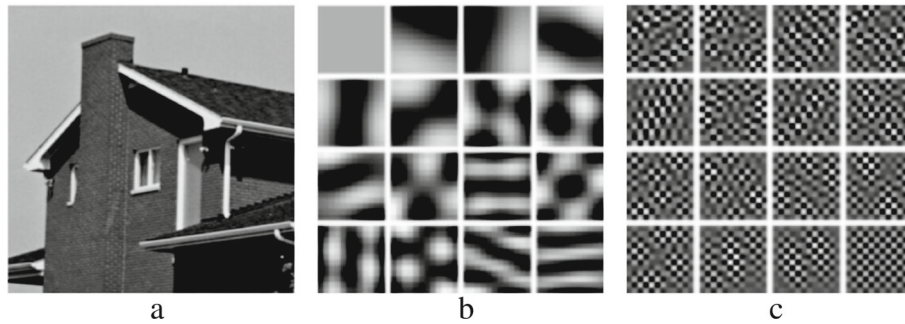


Fig. 6 The principal components “axis” of the house image: **a** is the input image, **b** is the first 16 principal axes of the all patches obtained from the house image, and **c** is the last 16 principal axes of the all patches obtained from the house image [13]

The PGPCA approaches cannot compete with other local or semi-local approaches, which consider the high level of redundancy occurring between neighboring patches. In the global PGPCA, one original basis for the whole image, which impacts negatively on the denoising process. The global PGPCA does not identify the rare patches because they do not exert a strong influence on the total variance. However, an allowance can be made for these limitations by considering the local redundancy between patches.

2.4.2 Patch-based local PCA

In the patch-based local PCA (PLPCA) approach, patches are collected locally in order to overcome the limitations of the global PGPCA approach. The local collection of patches means that the patches are collected within a small region of interest in the noisy image. A fixed search window $N \times N$ is applied to the whole image. Since the patches are overlapped in PLPCA, there will be multiple estimates for a single pixel. Averaging is used to compute a single pixel’s value.

The advantage of this approach is that the orthonormal basis is adapted only to the sub-image and not to the whole image. However, this approach has two limitations: the overfitting and the fact that it is time-consuming. The overfitting problem is due to the limited number of patches on which to compute the PCA. PLPCA is extremely time-consuming because the PCA needs to be computed repeatedly.

Fei et al. [18] collected patches locally, but they improved their approach by using geometric structure clustering to guarantee that only patches with similar properties were gathered. Pal et al. [46] considered patch redundancy in order to improve on the global two-stage PCA approach of Zhang et al. A sliding window that moves with a step $s = \frac{W_p - 1}{2}$, where s is the step size and W_p is the window’s current location was utilized as a modification in order to reduce the time-consuming element of the PLPCA approach. The computational time is divided by s^2 without losing the denoising quality. Zhang et al. [68] proposed using similar patch-based local PCA

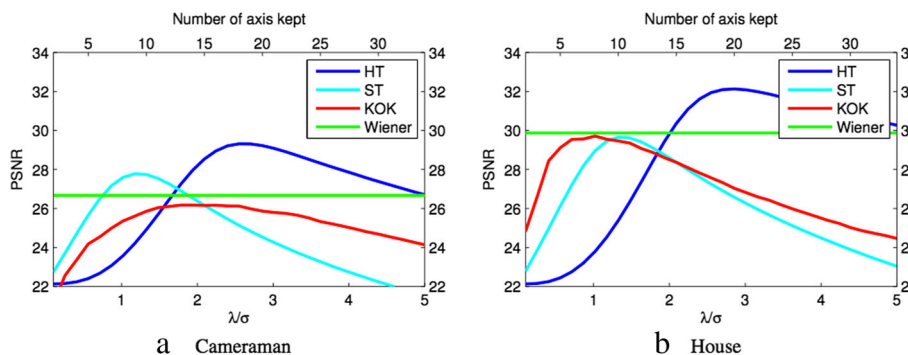


Fig. 7 Different PB-PCA projections: a comparison between PSNR of the different methods of the projections in PB-PCA for the House and Cameraman images. The threshold ratio (λ/σ) into the bottom of x -axis controls the number of axes kept in the upper x -axis. σ is the noise variation, and λ is chosen by cross validation [13]



Fig. 8 Collecting a set of patches in PB-PCA filtering: **a** global PCA, **b** local PCA, and **c** hierarchical PCA

filter with an extended step where a Wiener filter is finally applied.

2.4.3 Patch-based hierarchical PCA

In the patch-based hierarchical PCA (PHPCA) approach, an algorithm builds a hierarchy cluster of the patches. Clustering is the task of grouping a set of patches into the same cluster, i.e., set. There are different cluster models; each model has several clustering algorithms. The models include connectivity models (e.g., hierarchical clustering) and centroid models (e.g., k-means). The hierarchical clustering is based on the concept of the grouping of patches according to a maximum distance between patches. Patches are represented as a dendrogram, which is a Greek word meaning a tree diagram that illustrates the arrangement of the patches. In centroid models, vectors are assigned usually to a number k of fixed clusters. For more information about the clustering models, readers are referred to Chapter 17 in *Introduction to Information Retrieval* by Manning et al. [42].

The objective of the PHPCA approach is to offer a solution that can provide a result between the PGPCA and PLPCA approaches, which is less time-consuming than the local approaches and is more adapted to local sub-images. PHPCA uses a geometric partitioning to first divide the image into four areas, and then it estimates the

principal axis for each area. Each area has its own principal components. This process is repeated until the end of the tree is reached. Several dictionaries share the first axes. Figure 8c shows how an image can be divided to sub-images.

The most important advantage of patch-based PCA filter is its ability to produce a solution for image denoising using a modest orthogonal dictionary with PCA of the input data. Moreover, a patch-based PCA filter provides various grouping stages to fit user needs—local, global, or hierarchical. The main disadvantage of PB-PCA is its susceptibility to overfitting due to few training patches when performing the PCA locally. Also, PB-PCA is quite expensive in terms of time since PCA needs to be performed repeatedly.

2.5 Sparse 3D transform-domain collaborative filtering

Block matching 3D algorithm (BM3D) is the state-of-the-art denoising technique by Dabov et al. [11]. It is based on modified sparse representation in the frequency domain. BM3D groups the patches into 3D data arrays instead of into 2D arrays, then it applies a modified sparse representation in the frequency domain. Collaborative filtering is used to deal with the 3D arrays. BM3D’s algorithm depends on two steps: (1) collaborative hard thresholding and (2) the collaborative Wiener filtering. The two steps

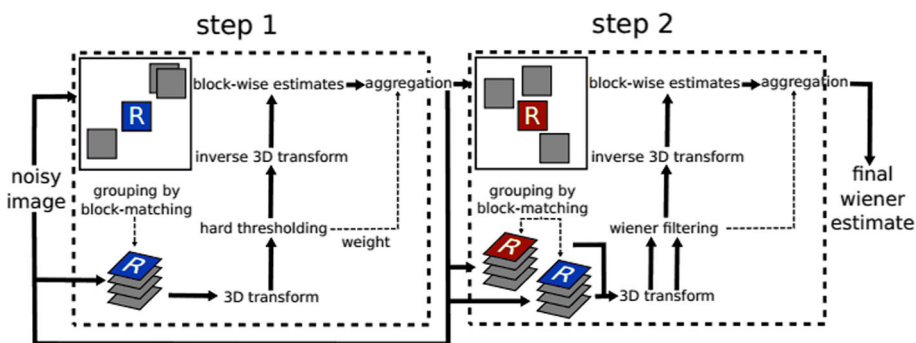


Fig. 9 BM3D filtering: the two steps of BM3D filtering [11]

Table 1 Parameters set for the original BM3D

	Symbol	Description	Fast BM3D	Normal BM3D	
				$\sigma \leq 40$	$\sigma \geq 40$
Step 1 (hard) parameters	T_{hard}^{2D}	2D transform	Bior1.5	Bior1.5	DCT
	T_{hard}^{3D}	3D transform	Haar	Haar	Haar
	N_1^{hard}	Patch size	8	8	12
	N_2^{hard}	3D array size	16	16	16
	$N_{\text{step1}}^{\text{hard}}$	Patch step	6	3	4
	$N_{\text{search}}^{\text{hard}}$	Window size	25	39	39
	$N_{\text{step2}}^{\text{hard}}$	Window step	6	1	1
	$N_{\text{Prev.}}^{\text{hard}}$	Small window	3	–	–
	β^{hard}	Kaiser window	2.0	2.0	2.0
	λ_{2D}	2D thresholding	0	0	2.0
	λ_{3D}	3D thresholding	2.7	2.7	2.8
	$T_{\text{match}}^{\text{hard}}$	Distance	2500	2500	5000
	Step 2 (Wiener) parameters	T_{Wiener}^{2D}	2D transform	DCT	DCT
T_{Wiener}^{3D}		3D transform	Haar	Haar	Haar
N_1^{Wiener}		Patch size	8	8	11
N_2^{Wiener}		3D array size	16	32	32
$N_{\text{step1}}^{\text{Wiener}}$		Patch step	5	3	6
$N_{\text{search}}^{\text{Wiener}}$		Window size	25	39	39
$N_{\text{step2}}^{\text{Wiener}}$		Window step	5	1	1
$N_{\text{Prev.}}^{\text{Wiener}}$		Small window	2	–	–
β^{Wiener}		Kaiser window	2.0	2.0	2.0
$T_{\text{match}}^{\text{Wiener}}$		Distance	400	400	3500

allow the BM3D to suppress more noise and to preserve more detail. The amount of noise is suppressed in the thresholding step, and the details are restored in the second step. Collaborative hard thresholding has three functions: (1) 3D transform, (2) shrinkage, and (3) 3D inverse transform. The patches in the 3D arrays are overlapped, so a weighted average is used to obtain one estimation for each pixel. Aggregation is the averaging procedure. A significant filter is obtained by using the BM3D algorithm. The scheme in Fig. 9 shows the two steps of BM3D filtering. Below, the two steps of the BM3D algorithm are described. First, the collaborative hard thresholding step is explained. Then, using the collaborative Wiener filtering is discussed.

2.5.1 Step 1: thresholding

Grouping: Similar to the NL-Means, a search window is used to determine the similarity between the patches. The search window is used in order to benefit from the high redundancy among the neighboring patches. There are several different grouping techniques. A number of these techniques have been discussed in this survey. Some other useful techniques for grouping could be considered for

patch grouping, such as vector quantization [21], k -means clustering [39], self-organizing maps [34], and others discussed in this survey [29]. However, grouping in BM3D is based on the similarity distance between patches, the “Euclidian distance.” The grouping stage is the first of the two steps in which similar patches are gathered to form 3D arrays. Similarity is computed according to the distances between the patches. Patches with a distance that is below a fixed threshold are considered to be similar and are grouped into the 3D array. Before measuring the distance, a coarse pre-filtering is used to linearly transform the patches using a 2D linear transformation such as multiple wavelet transforms [15, 53]. The formulation in Eq. 21 is used to compute the similarity distance between patches,

$$\text{Dst}([v]_i, [v]_j) = \frac{\|\gamma^{2D}(T_{\text{hard}}^{2D}([v]_i)) - \gamma^{2D}(T_{\text{hard}}^{2D}([v]_j))\|_2^2}{(N_1^{\text{hard}})^2}, \quad (21)$$

where $[v]_i, [v]_j$ are respectively the reference patches at i and its neighbors at j , T_{hard}^{2D} is the 2D linear transform,

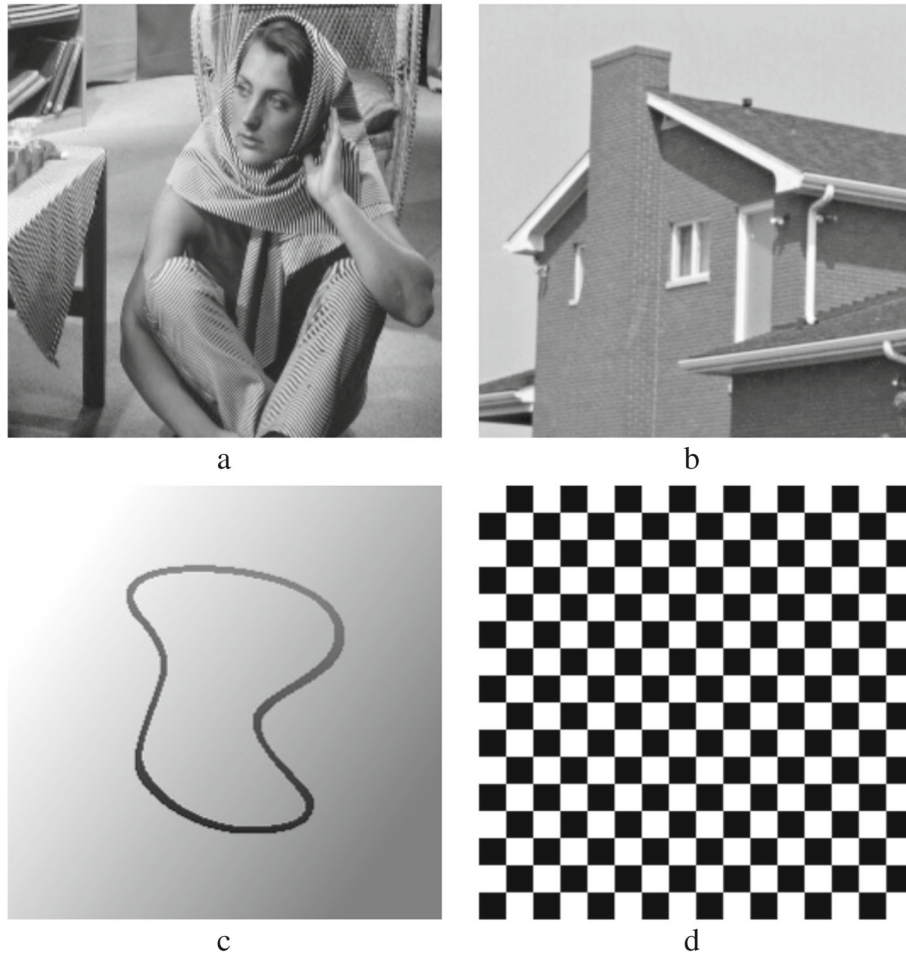


Fig. 10 The four used images in the experiment: **a** *Barbara* image 512×512 , **b** *House* image 256×256 , **c** *CurvedBand* image 257×257 , and **d** *Chessboard* image 256×256

γ^{2D} is a hard-thresholding operator equals to $\lambda_{2D} \times \sigma$, and $(N_1^{\text{hard}})^2$ is the patch size $N \times N$. σ is the estimated noise standard deviation. γ^{2D} makes all coefficients with absolute value less than the threshold ($\lambda_{2D} \times \sigma$) equal to zero, and it leaves the other coefficients unchanged. After computing the Euclidian distance, grouping the similar patches into a 3D array is required. The formulation in Eq. 22 is used for gathering similar patches.

$$3D S_i^{\text{hard}} = \left\{ j \in \Omega : \text{Dst}([v]_i, [v]_j) \leq T_{\text{match}}^{\text{hard}} \right\}, \quad (22)$$

where $3D S_i^{\text{hard}}$ is the constructed 3D array contains similar patches and $T_{\text{match}}^{\text{hard}}$ is the maximum distance between two similar patches. The maximum grouped patches size are restricted to N_2^{hard} . The next stage is to apply the collaborative filter by (1) performing a 2D linear transform then a 1D linear transform, (2) shrinkage, and (3) inverting the 1D transform and the 2D linear transform.

Collaborative filtering: Once the 3D array is built, a collaborative filter is used for suppressing the noise. A 3D transform is applied to the 3D array, before the shrinkage of the transforming coefficients. The 2D transformation in the grouping stage is applied along both horizontal and vertical lines for each patch, and then a third transformation is conducted along the third dimension of the 3D array for the 3D transform. The formulation of the collaborative filter is:

$$3D \hat{u}_{S_i^{\text{hard}}} = T_{\text{hard}}^{3D-1} \left(\gamma^{3D} \left(T_{\text{hard}}^{3D} \left(3D S_i^{\text{hard}} \right) \right) \right), \quad (23)$$

where T_{hard}^{3D} is the 3D linear transform of the first (hard) step, T_{hard}^{3D-1} is the inverse of 3D transformation, and γ^{3D} is a hard-thresholding operator equals to $\lambda_{3D} \times \sigma$. γ^{3D} makes all coefficients with absolute value less than the threshold ($\lambda_{3D} \times \sigma$) equal to zero.

Aggregation weights: At this stage, overlapped patches in the 3D array $(3D \hat{u}_{S_i^{\text{hard}}})$ have multiple estimates for each pixel in the reference patch at the location i . A weighted averaging procedure is required to provide an estimate for each pixel. Weights in BM3D are inversely proportional to the total variance of the patches in the $3D \hat{u}_{S_i^{\text{hard}}}$ array. When the total variance is high, a small weight is assigned to the patch.

The amount of the additive noise is independent when processing the collaborative filter in step 1 and step 2. Thus, the total variance is not the same after applying the collaborative filter in the first and second steps. In step 1, the total variance is computed by $\sigma^2 \times N_{\text{non-zero}}^{\text{hard}}$, where $N_{\text{non-zero}}^{\text{hard}}$ is the number of non-zero coefficients after the hard thresholding. The total variance calculated in step 2 depends on the results of the Wiener filter coefficients, the weights of step 2 will be explained on the facing page. However, the weights for step 1 is equal to:

$$\omega_i^{\text{hard}} = \begin{cases} \frac{1}{\sigma^2 \times N_{\text{non-zero}}^{\text{hard}}}, & \text{if } \rightarrow N_{\text{non-zero}}^{\text{hard}} \geq 1 \\ 1, & \text{otherwise} \end{cases} \quad (24)$$

2.5.2 Step two: Wiener filter coefficients

Grouping: Grouping in the second step is in some ways similar to the grouping in the first step; but here, the power spectrums of the first step are grouped, not just the patches from the noisy image. The same formula is used:

$$3D S_i^{\text{Wiener}} = \left\{ j \in \Omega : \frac{\| \hat{u}_{S_{[v]_i}^{\text{hard}}} - \hat{u}_{S_{[v]_j}^{\text{hard}}} \|^2}{(N_1^{\text{Wiener}})^2} \leq T_{\text{match}}^{\text{Wiener}} \right\}, \quad (25)$$

where $\hat{u}_{S_{[v]_i}^{\text{hard}}}$ and $\hat{u}_{S_{[v]_j}^{\text{hard}}}$ are the estimated sub-images from step 1 respectively at locations i and j , respectively. At this stage there are two groups: (1) a group of similar patches derived from the noisy image and (2) a group of similar patches derived from the first step.

Collaborative filtering: After grouping the patches, a 3D transform is applied to the 3D array of the grouped patches. A Wiener shrinkage is applied to the transform coefficients of the 3D array. The definition of the Wiener shrinkage coefficients of the power spectrum of the first step is shown in the equation:

$$3D W_{S_i^{\text{Wiener}}} = \frac{|T_{3D}^{\text{Wiener}}(3D S_i^{\text{Wiener}})|^2}{|T_{3D}^{\text{Wiener}}(3D S_i^{\text{Wiener}})|^2 + \sigma^2}, \quad (26)$$

where T_{3D}^{Wiener} is the 3D linear transform and $3D S_i^{\text{Wiener}}$ is the result of Eq. 25. The final stage in the collaborative

Wiener filtering of the second step is to multiply the Wiener shrinkage coefficients element-by-element by the 3D transform coefficients of the noisy image. The inverse of the 3D transform is applied. Multiplication and the inverse of the 3D transform are shown in the equation:

$$3D \hat{u}_{S_i^{\text{Wiener}}} = T_{3D}^{\text{Wiener}^{-1}} \left(3D W_{S_i^{\text{Wiener}}} \times (T_{3D}^{\text{Wiener}}(3D v_i)) \right), \quad (27)$$

where $3D v_i$ is the 3D transform coefficients of the noisy data.

Aggregation weights: Adjusting the weights in this step is not like the first step, which depends on the number of non-zero coefficients reached after the hard thresholding.

Table 2 The performance of the denoising algorithms with various noise levels (σ)

	Method	= 10		= 20		= 40		= 60	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Barbara Image	Noisy	24.795	0.7285	19.764	0.5248	16.328	0.3365	14.924	0.2503
	Bilateral	19.760	0.7806	19.314	0.7473	15.435	0.4370	12.782	0.1250
	NL-Means	25.106	0.9286	21.594	0.8450	17.634	0.6711	15.584	0.5387
	K-SVD	27.200	0.9204	23.353	0.8547	19.435	0.7255	17.087	0.6050
	BM3D	31.987	0.9287	30.244	0.8935	27.021	0.8075	24.962	0.7399
	Non-itPPB	24.241	0.9224	21.527	0.8581	17.631	0.6962	15.645	0.5709
	It-PPB	23.524	0.9128	20.750	0.8556	17.926	0.7291	16.273	0.6114
	PGPCA	27.744	0.9239	23.873	0.8565	20.591	0.7518	18.636	0.6747
	PLPCA	28.884	0.9340	25.304	0.8817	21.766	0.7768	19.769	0.7182
	PHPCA	28.663	0.9324	25.035	0.8779	21.636	0.7837	19.554	0.7136
House Image	Noisy	24.408	0.6177	19.658	0.3979	16.762	0.2431	15.424	0.1805
	Bilateral	26.867	0.8680	25.566	0.8004	18.486	0.3089	10.857	0.2504
	NL-Means	29.931	0.8836	26.872	0.8056	22.595	0.6400	19.864	0.4920
	K-SVD	30.294	0.8950	27.015	0.8285	23.143	0.7372	20.559	0.6374
	BM3D	29.861	0.9016	28.380	0.8448	26.864	0.7952	25.960	0.7618
	Non-itPPB	29.700	0.8854	27.205	0.8292	23.377	0.7202	20.812	0.6064
	It-PPB	29.144	0.8844	26.658	0.8283	23.684	0.7464	21.893	0.6576
	PGPCA	30.281	0.8983	27.114	0.8420	23.769	0.7586	21.826	0.7252
	PLPCA	30.902	0.9031	27.491	0.8404	23.800	0.7326	22.064	0.7139
	PHPCA	30.791	0.9012	27.484	0.8412	23.987	0.7573	22.026	0.7212
CurvedBand Image	Noisy	19.221	0.5012	15.068	0.2963	11.875	0.1752	10.465	0.1320
	Bilateral	34.645	0.9862	30.923	0.8872	21.083	0.2830	15.021	0.0978
	NL-Means	37.916	0.9631	32.627	0.8778	27.058	0.6670	23.570	0.5029
	K-SVD	35.287	0.9730	31.079	0.9383	26.702	0.8429	24.255	0.7578
	BM3D	35.080	0.9869	30.407	0.9696	26.082	0.9339	24.384	0.9189
	Non-itPPB	37.496	0.9785	34.169	0.9440	29.715	0.8258	25.760	0.6895
	It-PPB	36.731	0.9785	33.041	0.9508	29.736	0.8619	27.213	0.7557
	PGPCA	33.661	0.9640	29.573	0.9364	26.165	0.8592	24.457	0.8579
	PLPCA	34.526	0.9555	30.250	0.9220	26.362	0.7991	24.879	0.8261
	PHPCA	34.635	0.9612	30.171	0.9291	26.488	0.8444	24.705	0.8380
Chessboard Image	Noisy	15.513	0.5585	11.341	0.4397	8.139	0.3181	6.742	0.2503
	Bilateral	47.898	0.9853	37.586	0.9118	23.162	0.5564	15.662	0.3774
	NL-Means	25.050	0.9760	24.319	0.9435	22.771	0.8837	20.794	0.8371
	K-SVD	38.509	0.9609	32.666	0.8973	26.796	0.7995	23.076	0.7272
	BM3D	42.827	0.9898	38.422	0.9689	33.357	0.9300	28.304	0.9033
	Non-itPPB	21.668	0.9701	21.442	0.9407	20.398	0.8745	18.887	0.8280
	It-PPB	21.073	0.9601	20.676	0.9254	19.554	0.8631	18.216	0.8160
	PGPCA	34.036	0.9759	29.664	0.9537	23.812	0.8334	19.584	0.7675
	PLPCA	33.467	0.9411	28.836	0.9059	23.285	0.8150	20.276	0.7968
	PHPCA	33.408	0.9421	29.288	0.9227	23.893	0.8332	20.220	0.7873

Bold font indicates the best performance

The weights, here, depend on the Wiener shrinkage coefficients; the weights are assigned as:

$$\omega_i^{\text{Wiener}} = \sigma^{-2} \left\| W_{S_i}^{\text{Wiener}} \right\|_2^{-2}. \quad (28)$$

The parameters set for the original BM3D is shown in Table 1; the table has some non-discussed parameters:

- $N_{\text{step1}}^{\text{hard}}$: step size for searching the patches inside the search window
- $N_{\text{step2}}^{\text{hard}}$: step size for moving the search window
- $N_{\text{prev}}^{\text{hard}}$: small search window width for fast BM3D
- β^{hard} : the Kaiser window function one parameter for reducing the borders effect
- Haar: a Haar transform
- DCT: a discrete cosine transform
- Bior1.5: a biorthogonal wavelet

BM3D filter has advantages and disadvantages. The main advantage of BM3D filtering is that it yields great results with less loss of detail due to the smoothing and sharpening stages. The BM3D filter is a fast method because it computes similarity between patches before the actual filtering procedure. Despite its many advantages, BM3D has several drawbacks. The performance of BM3D decreases with a high noise level ($\sigma > 40$), and it produces images with many artifacts. BM3D is more complex and less flexible to be adapted for domain-specific image processing applications. In addition, it is not easy to be parallelized. In addition, it has many parameters, and adjusting them optimally is a challenging task.

BM3D has two filtering steps and more than 20 parameters. Improving the way of adjusting any of the 20 parameters would participate on improving the output of the BM3D method. BM3D modifications would be categorized into four main categorizes: (1) modifications consider improving the shrinkage function, (2) modifications consider improving the transforms, (3) modifications consider improving the image similarity measures when collecting similar patches, and (4) modifications consider improving the Wiener filtering stages. Suwabe et al. [55] modified the way of collecting similar patches in the BM3D from non-locally to globally. They proposed using iterative filtering with Chebyshev polynomial approximation (CPA) in order to collect the patches from the whole noisy image. Bashar and El-Sakka [3] replaced the fixed hard thresholding scheme with a learning-based adaptive hard thresholding scheme that considers the context of corresponding blocks. Hasan and El-Sakka [24] improved the Wiener filter of BM3D by maximizing the structural similarity (SSIM) [60] between patches instead of using the mean square error (MSE). Moreover, they introduced a 3D zigzag thresholding.

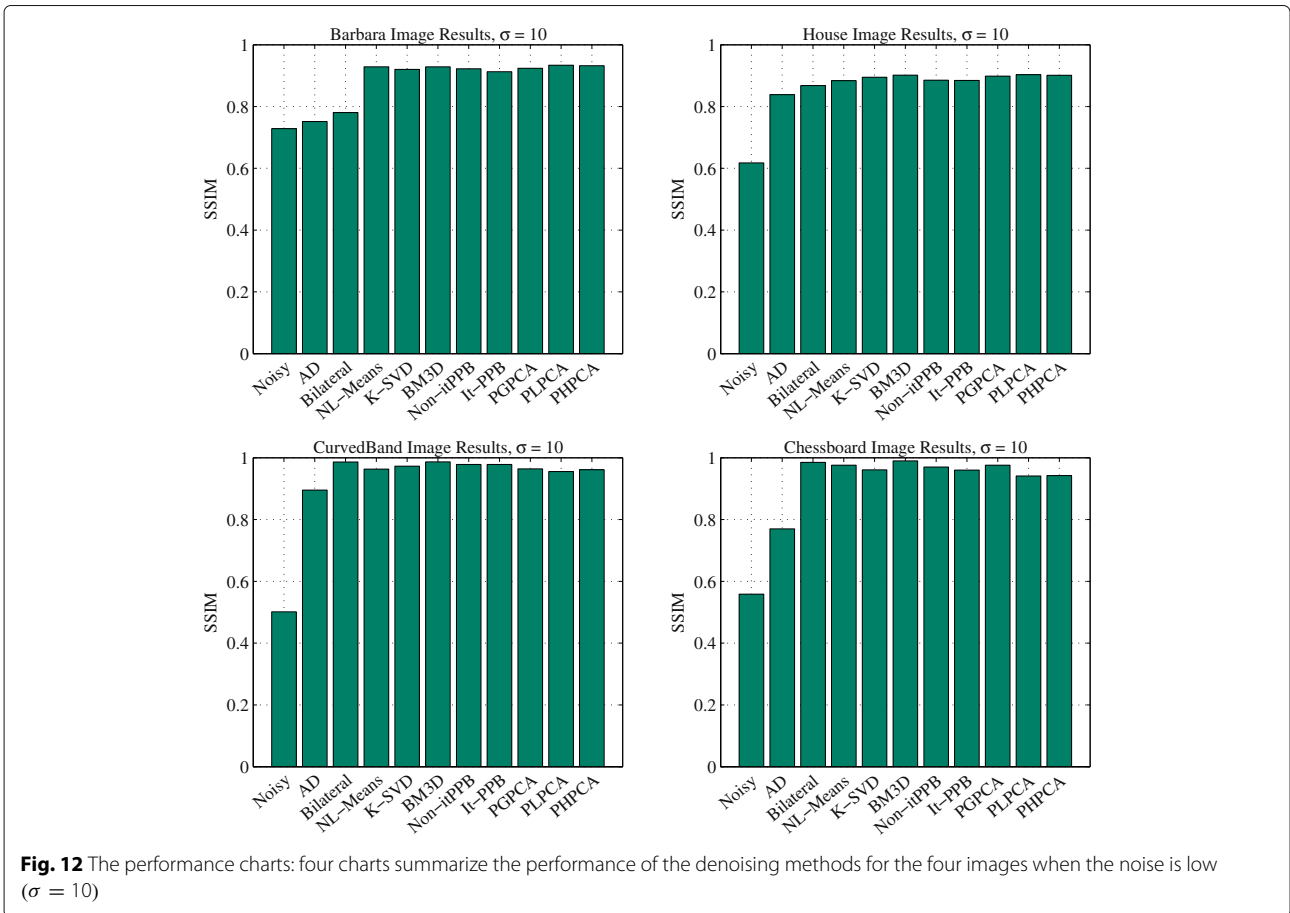
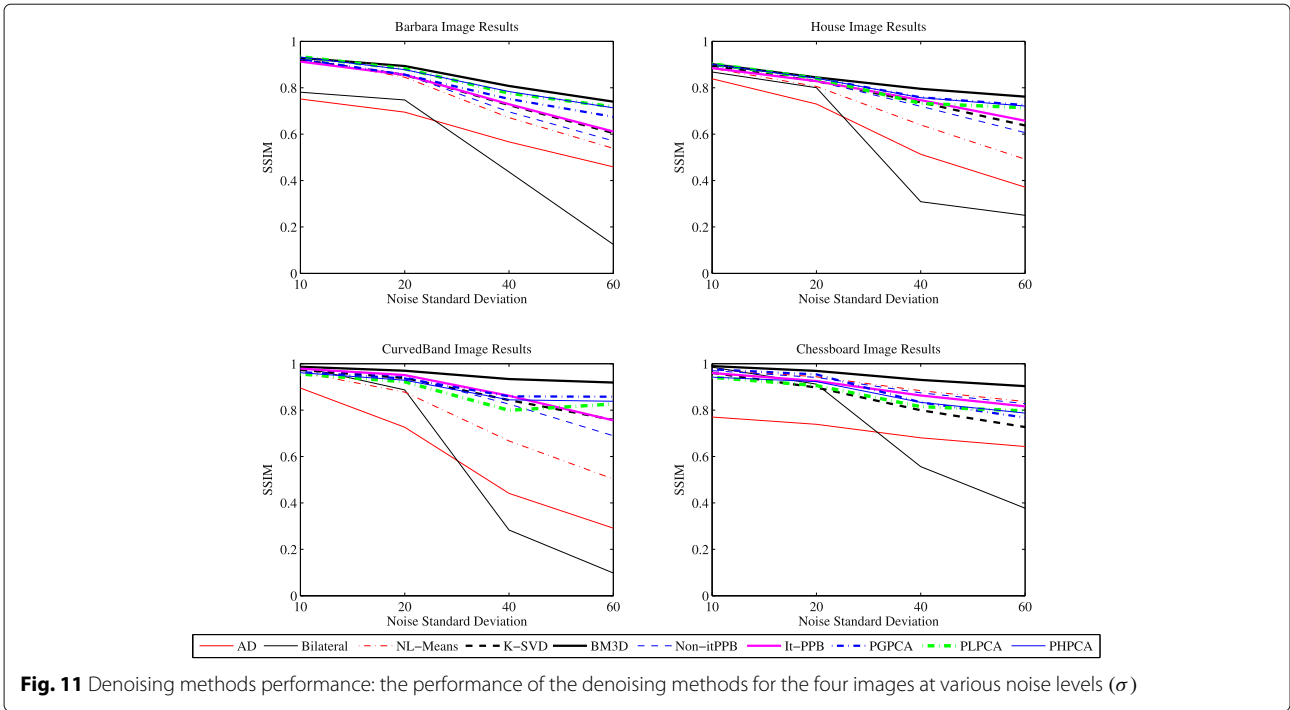
3 Results and discussions

In this section, various denoising methods are compared aiming to reduce additive white Gaussian noise (AWGN). The objective of this section is to experimentally study the performance of these methods, where the performance is

Table 3 Execution time in seconds shows the speed of the denoising methods applied to the four images at various noise levels (σ)

	Method	= 10	= 20	= 40	= 60	Average
		Seconds	Seconds	Seconds	Seconds	Seconds
Barbara Image	AD	0.05	0.05	0.05	0.05	0.05
	Bilateral	0.48	0.48	0.48	0.49	0.48
	NL-Means	44.13	46.16	43.66	46.58	45.13
	K-SVD	1125.85	1163.52	1116.98	1113.34	1129.92
	BM3D	3.13	3.16	3.21	3.96	3.37
	Non-it PPB	10.26	9.93	10.37	10.16	10.18
	It-PPB	32.10	28.60	30.23	28.68	29.90
	PGPCA	5.35	5.27	6.79	6.82	6.06
	PLPCA	9.42	9.05	15.43	16.99	12.72
	PHPCA	7.62	7.57	9.71	9.77	8.67
House Image	AD	0.05	0.05	0.05	0.05	0.05
	Bilateral	0.48	0.48	0.48	0.48	0.48
	NL-Means	44.23	45.50	43.76	43.45	44.24
	K-SVD	356.09	344.96	338.84	339.82	344.93
	BM3D	0.84	0.82	0.80	0.98	0.86
	Non-it PPB	10.97	10.02	10.35	10.10	10.36
	It-PPB	30.10	29.38	29.16	28.63	29.32
	PGPCA	1.32	1.34	1.65	1.65	1.49
	PLPCA	2.28	2.28	3.65	3.73	2.99
	PHPCA	1.76	1.71	2.15	2.12	1.93
CurvedBand Image	AD	0.05	0.05	0.05	0.05	0.05
	Bilateral	0.48	0.48	0.49	0.49	0.48
	NL-Means	42.84	45.93	43.42	45.83	44.50
	K-SVD	349.84	342.92	340.05	341.39	343.55
	BM3D	0.32	0.82	0.84	0.99	0.74
	Non-it PPB	9.88	9.99	9.95	10.06	9.97
	It-PPB	30.66	28.67	28.74	28.66	29.18
	PGPCA	1.34	1.39	1.63	1.68	1.51
	PLPCA	2.37	2.19	3.84	3.81	3.05
	PHPCA	1.75	1.74	2.19	2.13	1.95
Chessboard Image	AD	0.05	0.05	0.05	0.05	0.05
	Bilateral	0.47	0.48	0.48	0.47	0.48
	NL-Means	42.39	45.58	44.00	42.93	43.73
	K-SVD	1161.89	1155.13	1115.27	1112.00	1136.07
	BM3D	0.79	0.32	0.30	0.43	0.46
	Non-it PPB	13.18	10.33	10.01	10.05	10.89
	It-PPB	41.24	37.00	28.81	29.25	34.07
	PGPCA	0.79	0.86	1.03	0.98	0.91
	PLPCA	1.35	1.36	2.03	2.11	1.71
	PHPCA	1.10	1.04	1.32	1.32	1.20

Bold font indicates the best performance



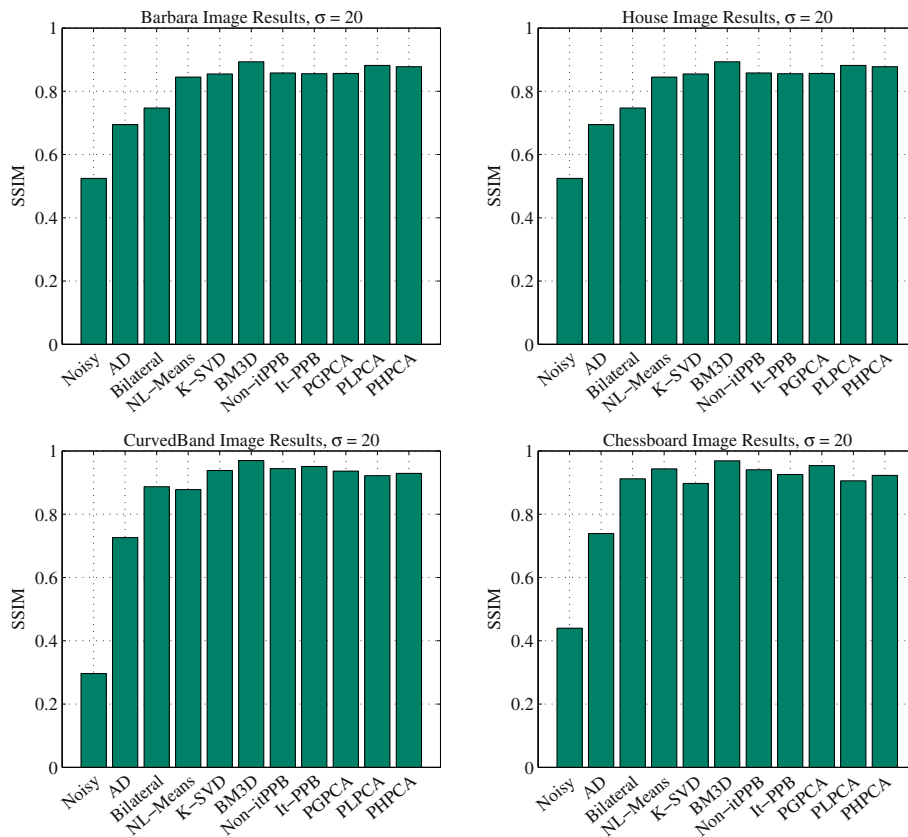


Fig. 13 The performance charts: four charts summarize the performance of the denoising methods for the four images when the noise is low ($\sigma = 20$)

assessed at various noise levels. The issue of time consumption is also addressed. Four images are used to run this experiment. The images have been chosen carefully to help in distinguishing between the methods. The first two of the four images are natural scene images, Barbara and House; the other two are synthetic images, CurvedBand and Chessboard. The Chessboard image is a binary image while the other three images are gray-scale images. The four images are shown in Fig. 10. The fine details in Barbara image helps in demonstrating how various methods preserve the image clarity, whereas the sharp edges in the House image helps in demonstrating how various methods preserve edges. The gray gradations in CurvedBand image provide insight into the amount of smoothing that has been applied to images. The methods are also tested with the binary pattern repetitions in the Chessboard image. MatLab is used for this experiment. The computer's processor is an Intel (R) Core(TM) i7 CPU @ 3.40 GHz. In Subsections 3.1 and 3.2, the methods are evaluated both quantitatively and qualitatively.

3.1 Quantitative evaluation

In order to make an objective comparison between the results, two image similarity matrices are used: (1) the structural similarity and (2) the peak signal-to-noise ratio (PSNR). These quality metrics have been chosen for their capability to assess the level of the additive Gaussian noise degradation. The best result for SSIM is 1, while the PSNR has good result when its value is high. Equations 29 and 30 show the formulas for these two quality metrics, respectively:

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}, \quad (29)$$

where x is the true reference image, y is the noisy image, μ_x and μ_y are the mean of the true reference image block and the noisy image block, respectively, σ_x and σ_y are the variance/covariance of the true reference image block and the noisy image block, respectively, and C_1

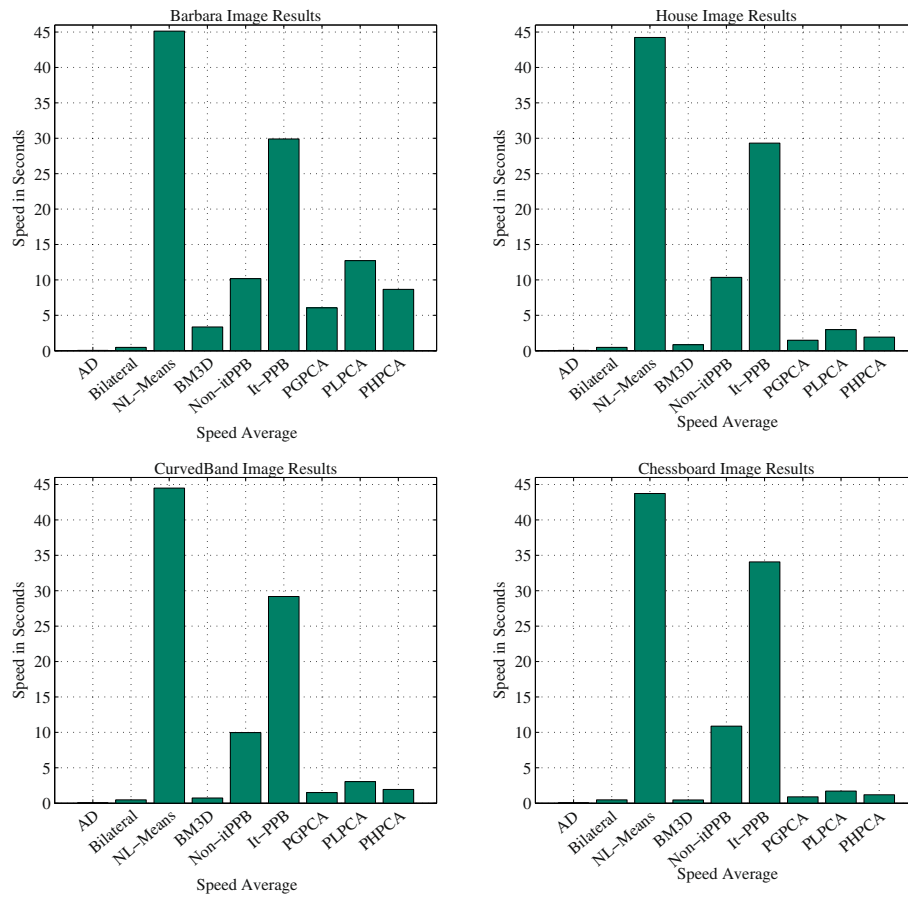


Fig. 14 The efficiency charts: four charts show the average of the consumed time in seconds for various denoising methods excluding K-SVD

and C_2 are constants used to avoid instability. The peak signal-to-noise ratio is defined as:

$$\text{PSNR} = 10 \log \left(\frac{(2^n - 1)^2}{\text{MSE}} \right), \quad (30)$$

where MSE is the mean squared error and n is an integer number representing the number of bits per pixel. When $n = 8$, i.e., in case of gray-scale images, the PSNR formula is reduced to:

$$\text{PSNR} = 10 \log \left(\frac{255^2}{\text{MSE}} \right). \quad (31)$$

A study conducted by Hore and Ziou [26] has revealed that SSIM is less sensitive to additive noise than PSNR. They used F -score test to compare between SSIM and PSNR which works for AWGN. Thus, the final conclusion in our study is driven based on SSIM.

The experimental results of the denoising methods are shown in Tables 2 and 3. The tables show the performance of the patch-based denoising methods along with

the famous pixel-based denoising methods: anisotropic diffusion (AD) by Perona and Malik [48] and the bilateral filter. In both tables, the methods are sorted from the oldest to the most recent.

The results in Table 2 are computed by measuring the differences between the original images and the denoised images. The default parameters shown in Table 1 were used for the BM3D method, these values are suggested by the authors of BM3D. The noise standard deviation of the noisy images, which methods depends heavily on it, is required to be adjusted accurately before the denoising process. In case the noise estimation was not given, Ghazi and Erdogan [22], Tai and Yang [56], and Liu et al. [37] offer more information about noise estimation. The highest values of SSIM and PSNR are highlighted with a bold font. Table 3 shows the execution time of the methods in seconds. The fastest patch-based methods are highlighted with a bold font.

The performance of the denoising methods varies depending on the noise level and the scene details inside noisy images. For example, the pixel-based bilateral filter outperforms the famous patch-based NL-Means method

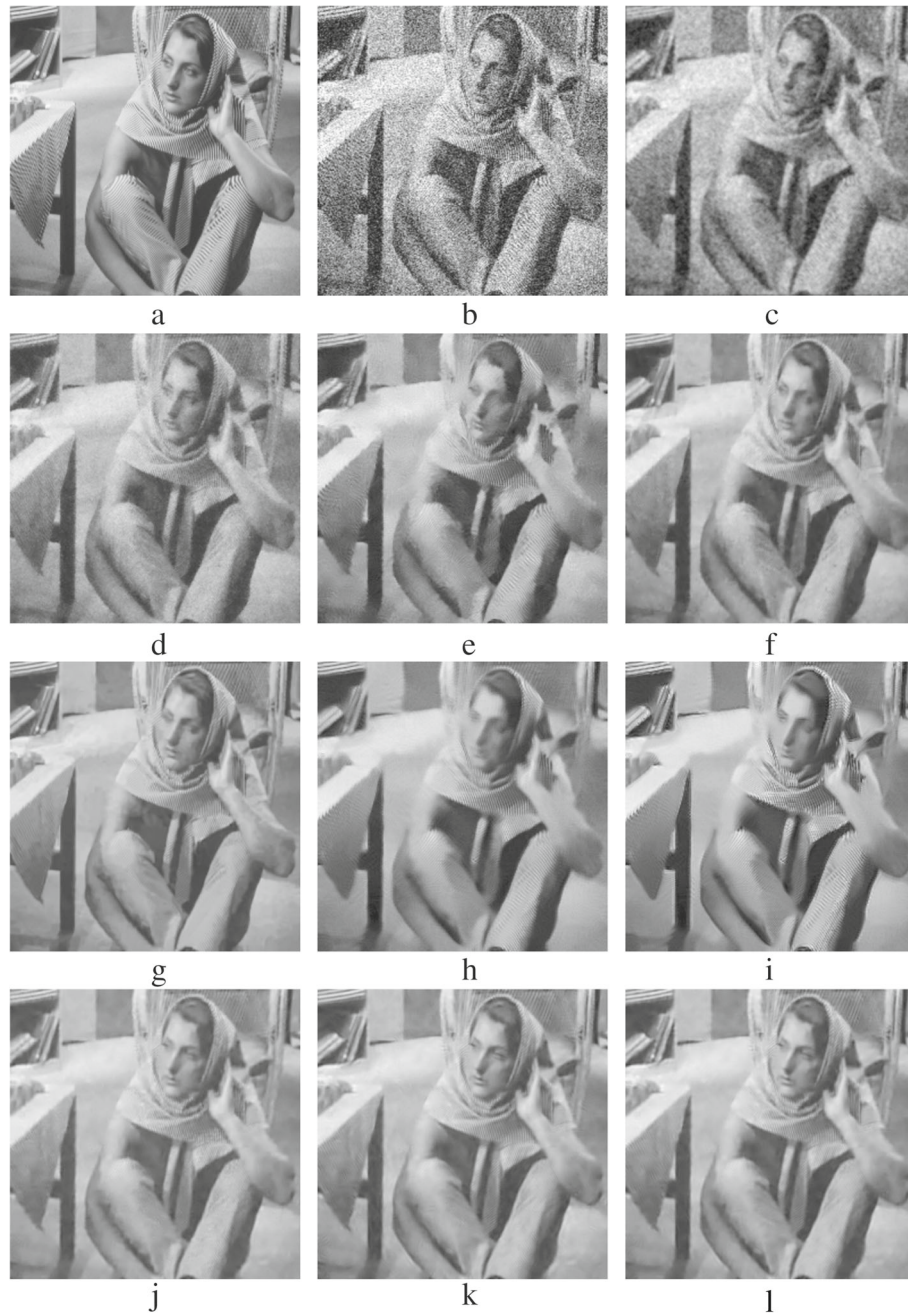


Fig. 15 Denoised *Barbara* images: **a** Original *Barbara* image; **b** AWG noise, $\sigma = 40$; **c** AD; **d** bilateral filtering, **e** NL-Means filtering, **f** K-SVD, **g** BM3D, **h** non-itPPB, **i** it-PPB, **j** PGPCA, **k** PLPCA, and **l** PHPCA

when denoising the Chessboard image at $\sigma \leq 10$ because of the large number of flat regions. Using pixel-based methods is not recommended when standard deviation of a noisy image is above 40. Although promising results have been achieved by BM3D when denoising flat and textures scene images at high noise levels $\sigma > 10$, the performance of BM3D decreases when denoising images have a lot of fine details regions at $\sigma \leq 10$;

see the results of *Barbara* and *House* images in Table 2.

Figure 11 shows the performance of the methods with various levels of noise on each of the four images. The charts show that BM3D is the best method (from SSIM point of view) when the noise level is high. BM3D is the best whether it is used for denoising natural scene or synthetic images. PCA patch-based methods come second

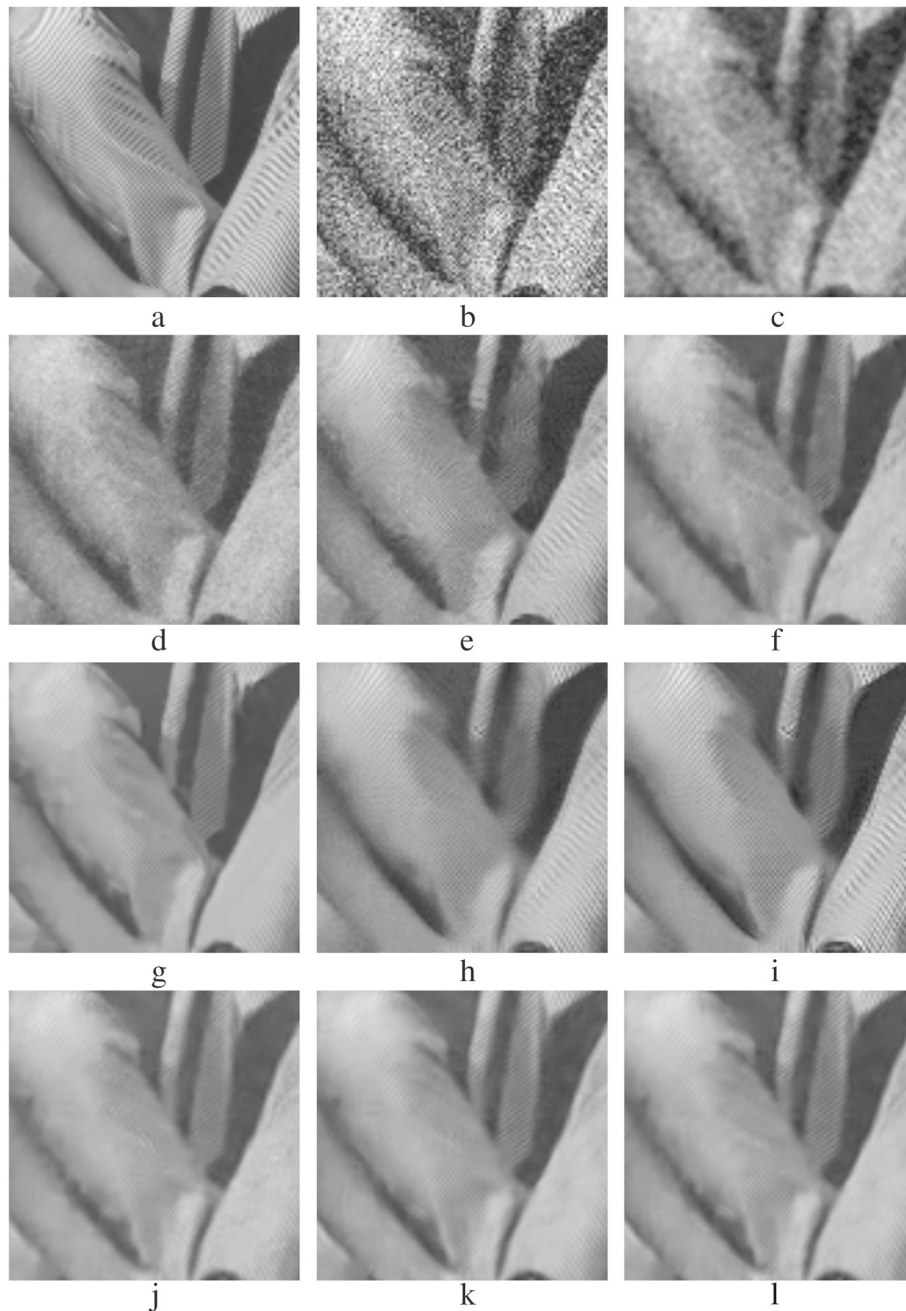


Fig. 16 Zoomed images of the denoised *Barbara* image shown in Fig. 15: **a** original *Barbara*'s pant; **b** AWG noise, $\sigma = 40$; **c** AD (Perona & Malik); **d** bilateral filtering; **e** NL-Means filtering; **f** K-SVD; **g** BM3D; **h** Non-itPPB; **i** It-PPB; **j** PGPCA; **k** PLPCA; and **l** IHPPCA

after BM3D when they are used for natural scene images. The results of K-SVD and the iterative PPB are similar.

The two lines at the bottom of each chart presents the performance of pixel-wise methods: anisotropic diffusion and the bilateral filter. From the charts, we can conclude that the block-wise denoising methods perform better than the pixel-wise methods. When using $\sigma = 10$ and $\sigma = 20$, the results are similar except with the

pixel-wise methods. The performance of the methods will be discussed in the following two paragraphs.

Bar charts are used in Fig. 12 to represent the methods performance when the noise level is low $\sigma = 10$. The charts show that patch-based denoising methods are similar. The pixel-wise denoising methods perform better when denoising the synthetic images due to its flat areas and the absent of details. In contrast, the patch-based

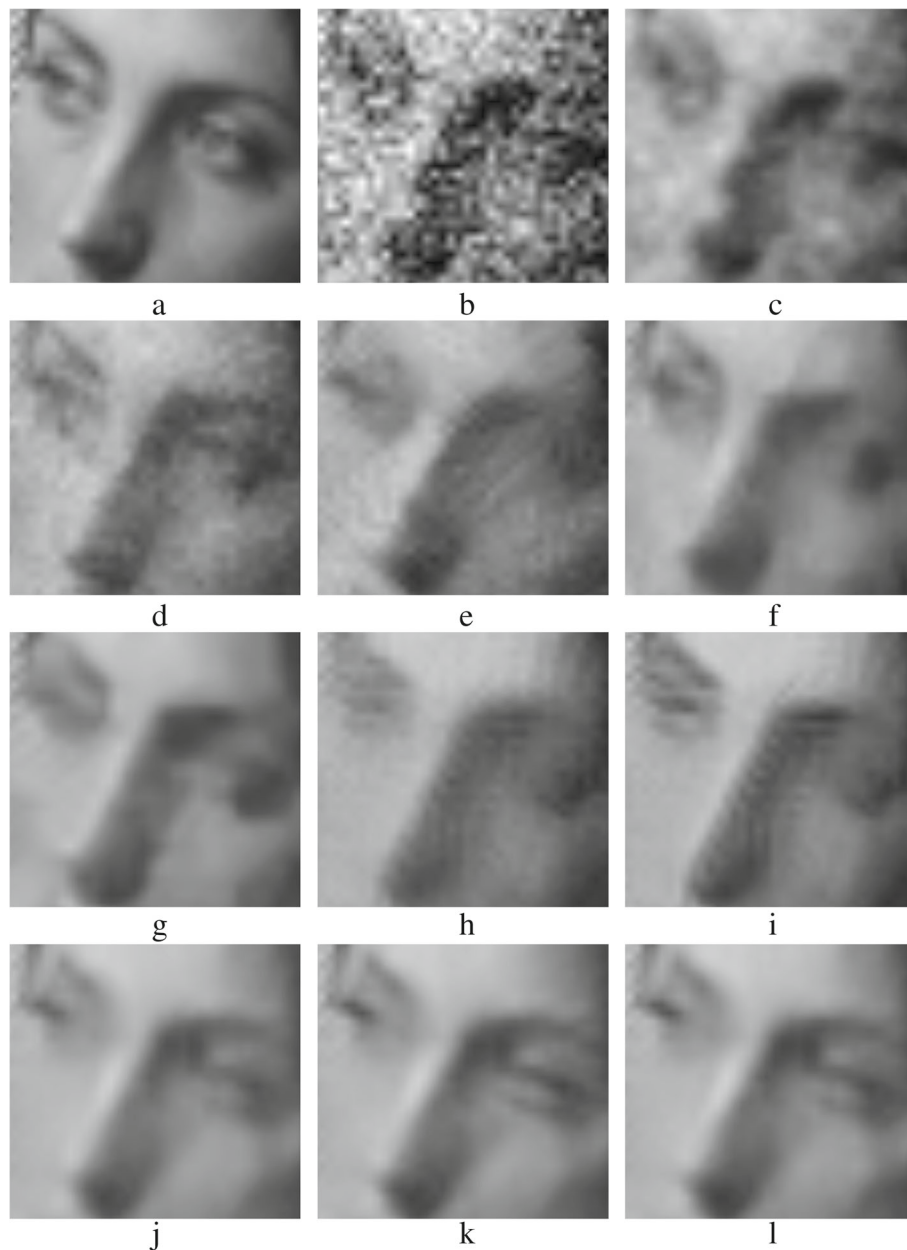


Fig. 17 Zoomed images of the denoised *Barbara* image shown in Fig. 15: **a** original *Barbara*'s eye cover fold; **b** AWG noise, $\sigma = 40$; **c** AD (Perona & Malik); **d** bilateral filtering; **e** NL-Means filtering; **f** K-SVD; **g** BM3D; **h** Non-itPPB; **i** It-PPB; **j** PGPCA; **k** PLPCA; and **l** PHPCA

denoising methods achieved better results when denoising the natural scene images. When the noise level is low $\sigma = 10$, the top two denoising methods for denoising the natural scene images are PHPCA and BM3D, whereas the top two denoising methods for denoising the synthetic images are BM3D and bilateral filter. Yet, the differences are insignificant.

The charts in Fig. 13 show the methods performance when $\sigma = 20$. By increasing the noise level, the contrast between methods becomes obvious unlike when the

noise is low $\sigma = 10$. BM3D achieved better results than other methods in all images. The patch-based denoising methods perform better than pixel-wise methods.

Figure 14 contains charts that illustrate the average execution time aspect for each image. The K-SVD methods are very time-consuming. They are about ten times as expensive as any other methods. Thus, K-SVD is excluded from the charts to make it easier to distinguish between methods. The NL-Mean comes second after the K-SVD. Although, the BM3D uses two stages to perform

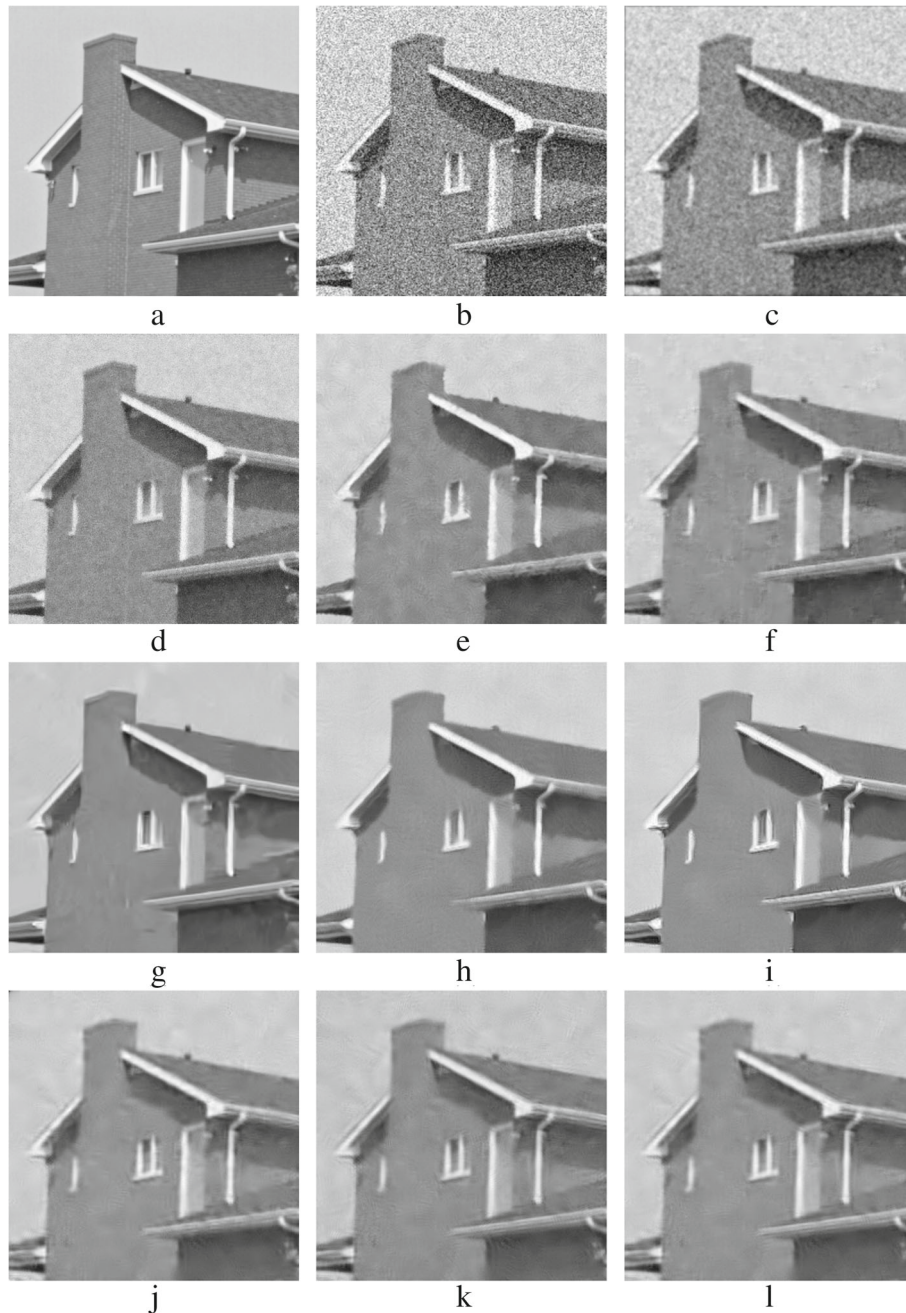


Fig. 18 Denoised *House* images: **a** original *House* image; **b** AWG noise, $\sigma = 40$; **c** AD (Perona & Malik); **d** bilateral filtering; **e** NL-Means filtering; **f** K-SVD; **g** BM3D; **h** Non-itPPB; **i** lt-PPB; **j** PGPCA; **k** PLPCA; and **l** PHPCA

the denoising step, it is the fastest patch-based method. The time consumed for various values of noise is almost the same, so the high level of noise does not greatly affect the time consumption of the various methods.

3.2 Qualitative evaluation

The evaluation in this subsection is a subjective evaluation, where the quality of the denoised images is

addressed via the visual perception. Denoised images with AWGN ($\sigma = 40$) are chosen to perform this evaluating. Figures 15, 18, 19, and 21 show the denoised images.

The results of denoising *Barbara's* image are shown in Fig. 15. Figure 15g shows the best achieved result by using BM3D method. While homogeneous regions are properly smoothed in BM3D, the sharp edges are preserved. The K-SVD and the PCA patch-based methods

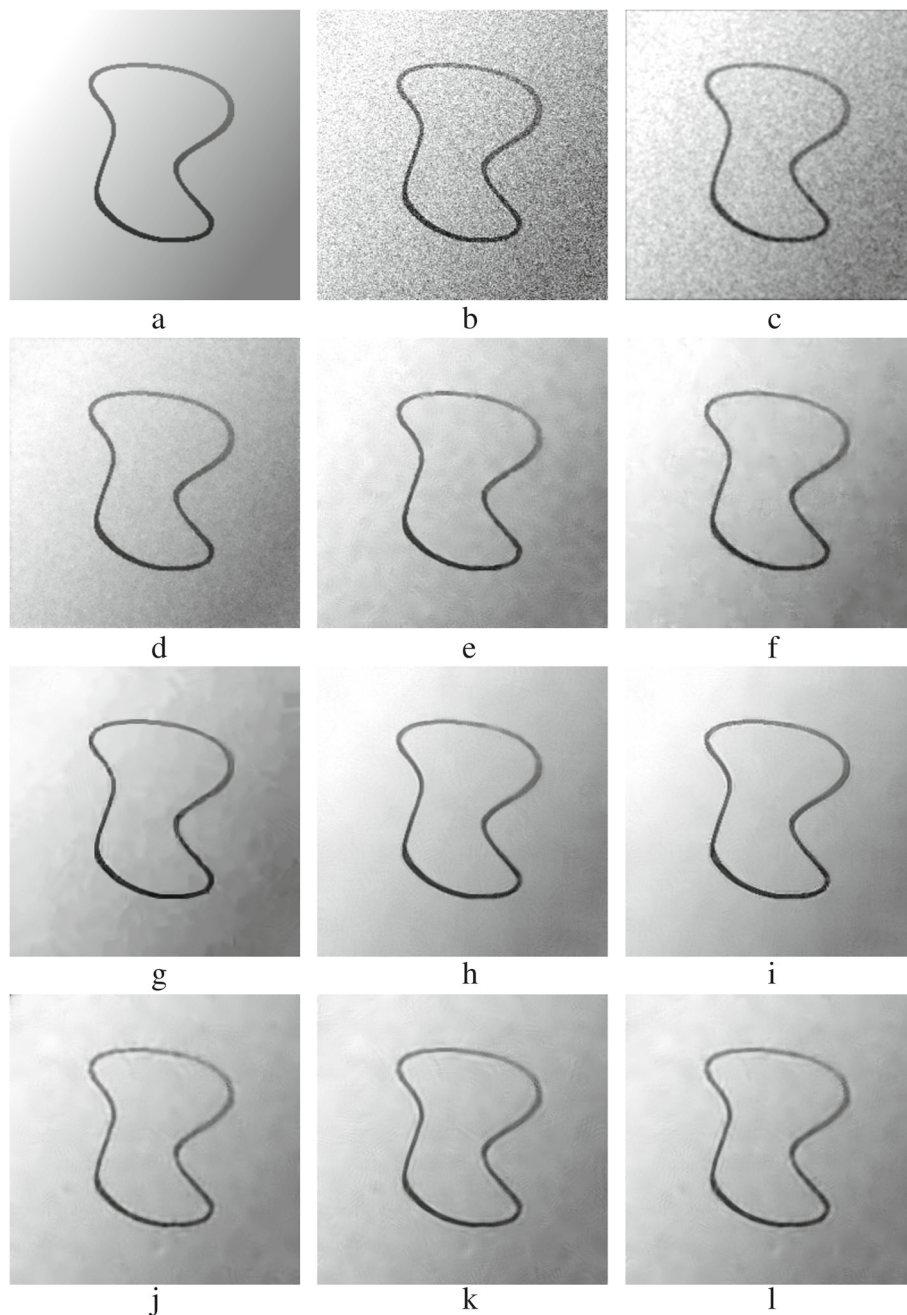


Fig. 19 Denoised *CurvedBand* images: **a** original *CurvedBand* image; **b** AWG noise, $\sigma = 40$; **c** AD (Perona & Malik); **d** bilateral filtering; **e** NL-Means filtering; **f** K-SVD; **g** BM3D; **h** Non-itPPB; **i** It-PPB; **j** PGPCA; **k** PLPCA; and **l** PHPCA

come after BM3D, Fig. 15f, j, k, l shows K-SVD and PCA patch-based results. They apply a good smoothing to the image, but some edges are destroyed. The non-local mean method applies less smoothing; thus, the textures shown in Fig. 15e in Barbara's pants are preserved; a zoomed version of Fig. 15e is shown in Fig. 16e. Figure 15h, i shows how the PPB methods apply a good smoothing, but they fail to preserve sharp edges and textures, i.e., Barbara's

eye cover fold. Zoomed versions of Fig. 15h, i are shown in Fig. 17h, i.

Figure 18 shows the results of denoising the House image. The best results are achieved by using BM3D and PPB methods, Fig. 18g–i shows these results. K-SVD and PLPCA methods have disappointing results, unlike their results when they are used for denoising Barbara image. Figure 18f, k show K-SVD and PCA methods results.

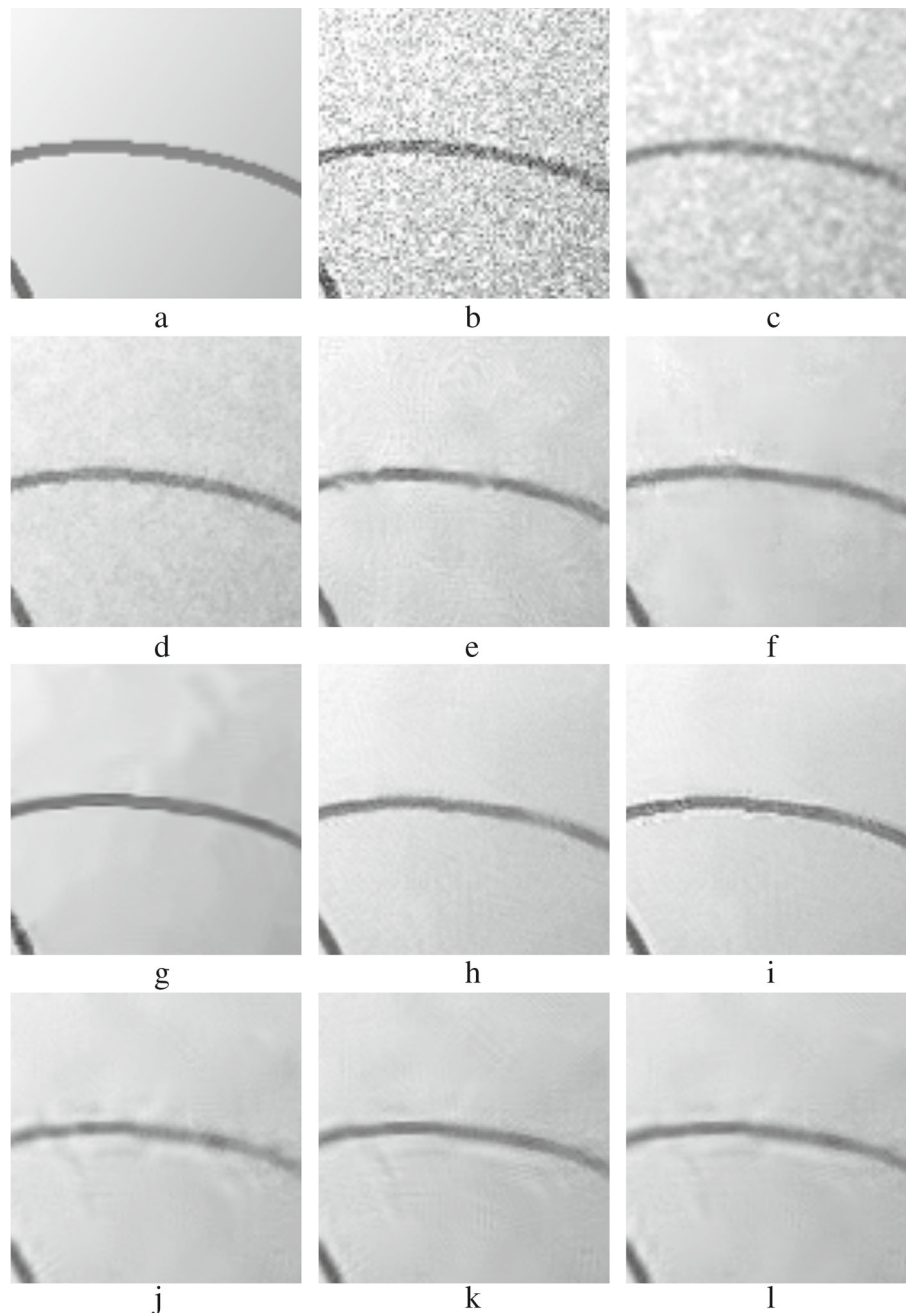


Fig. 20 Zoomed images of the denoised *CurvedBand* image shown in Fig. 19: **a** original *CurvedBand* image; **b** AWG noise, $\sigma = 40$ **c** AD (Perona & Malik); **d** bilateral filtering; **e** NL-Means filtering; **f** K-SVD; **g** BM3D; **h** Non-itPPB; **i** It-PPB; **j** PGPCA; **k** PLPCA; and **l** PHPCA

The edges are preserved when using K-SVD, but K-SVD fails to smooth properly flat areas, i.e., the sky in the House image. Thin edges are not preserved with PCA patch-based methods.

Figure 19 shows the denoised *CurvedBand* image. PPB methods and BM3D have the best results, the results are shown in Fig. 19g–i. Unlike BM3D, PPB methods succeed more in smoothing the gray gradations; zoomed versions

of Fig. 19g–i are shown in Fig. 20g–i. K-SVD and PCA patch-based methods have similar results, Fig. 19f, j, k, l shows these results. They preserve parts of the curve, but they do not smooth properly flat areas; zoomed versions of Fig. 19f, j, k, l are shown in Fig. 20f, j, k, l.

Figure 21 shows the denoised *Chessboard* image. Figure 21g shows BM3D result, BM3D achieves the best result when denoising the binary *Chessboard*'s image.

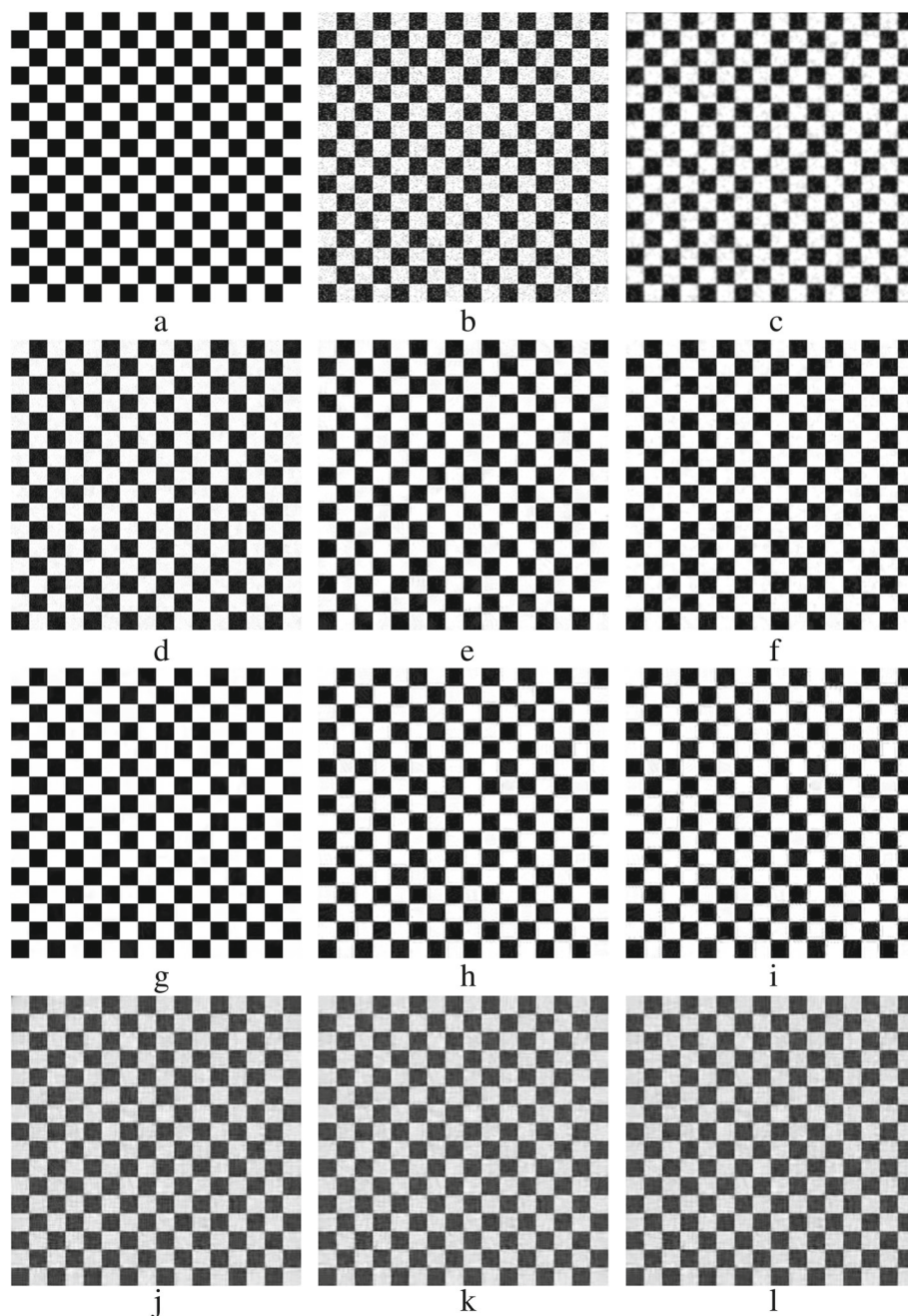


Fig. 21 Denoised *Chessboard* images: **a** original *Chessboard* image; **b** AWG noise, $\sigma = 40$; **c** AD (Perona & Malik); **d** bilateral filtering; **e** NL-Means filtering; **f** K-SVD; **g** BM3D; **h** Non-itPPB; **i** It-PPB; **j** PGPCA; **k** PLPCA; and **l** PHPCA

NL-Means method has the second good result, unlike its performance on the other images. Figure 21e shows the result of NL-Means method. With the K-SVD method shown in Fig. 21f, edges are preserved and the flat areas are smoothed appropriately. Disappointed results are achieved by using PCA patch-based methods; Fig. 21j, k, l shows the result of using PCA patch-based methods.

4 Summary of contribution

This paper provides a review of the state-of-art patch-based denoising techniques and compares their effectiveness for denoising natural, synthetic, and binary images. Effort is drawn on details like smoothing flat regions and objects and preserving details such as edges, lines, and textures. The features, strengths, and limitations of

the patch-based denoising techniques are also presented. Moreover, the review covers the issue of time. Finally, based on years of experience, we believe that this review will be helpful for researchers to choose suitable denoising techniques to be adapted further for their image processing applications.

5 Future research directions

Like any other image denoising approaches, many important research directions should remain in patch-based image denoising. Improving patch similarity measures is suggested for grouping accurately similar patches. Moreover, the research should also produce tools with better shrinkage functions to suppress noise and preserve fine details. Finally, developing effective image transform strategies that meet the needs to differentiate between a true and noisy signal is strongly recommended.

6 Conclusions

Among the best denoising methods is patch-based denoising method, which includes BM3D, NL-Means, and K-SVD. This paper has dealt with the efficiency of each of these methods when compared to other patch-based denoising methods. Experimentally, BM3D method gives the best result; it performs very well on all images and at all levels of noise. K-SVD and PCA methods come second to BM3D. Although the sparsity is learned from the data itself in K-SVD, K-SVD fails to compete with BM3D. NL-Means gives an encouraging result at low levels of noise. Furthermore, PPB methods preserve the fine details but fails to do so with respect to the sharp edges.

K-SVD is incomparable with the other compared methods in terms of its time consumption. As it is very expensive. In contrast, BM3D execution time is the best among the patch-based denoising methods because it computes all similarity between whole patches first before starting the actual denoising process.

Abbreviations

AD: Anisotropic diffusion; AWGN: Additive white Gaussian noise; BM3D: Block matching 3D algorithm; CPA: Chebyshev polynomial approximation; DL: Dictionary learning; EPLL: Extend patch log likelihood; HSI: Hyperspectral image; HT: Hard thresholding; It-PPB: Iterative probabilistic patch-based; K-SVD: K-means singular value decomposition; KoK: Keep or Kill; MLE: Maximum likelihood estimator; MSE: Mean square error; MSSIM: Mean structural similarity; NL-Means: Non-local means; NSERC: Natural sciences and engineering research council of Canada; Non-iPPB: Non-iterative probabilistic patch-based; PB-PCA: Patch-based principal component analysis; PCA: Principal components analysis; PGPCA: Patch-based global PCA; PHPCA: Patch-based hierarchical PCA; PLPCA: Patch-based local PCA; POL-SAR: Polarimetric synthetic aperture radar; PPB: Probabilistic patch-based; PPBWE: PPB weights Estimator; PSNR: Peak signal-to-noise ratio; SNR: Signal-to-noise ratio; SSIM: Structural similarity; ST: Soft thresholding; SVD: Singular value decomposition; WMLE: Weighted maximum likelihood estimation; XNL-Means: X-ray non-local means

Acknowledgements

This research is partially funded by the Natural Sciences and Engineering Research Council of Canada (NSERC). This research is also partially funded by Jeddah University, Jeddah, Saudi Arabia. These supports are greatly appreciated.

Funding

This research is partially funded by the Natural Sciences and Engineering Research Council of Canada (NSERC). This research is also partially funded by Jeddah University, Jeddah, Saudi Arabia.

Authors' contributions

MES suggested the idea; MHA carried out the experiments. MES continuously supervised the experiments and suggested modifications. MHA prepared the manuscript draft, and MES edited and proofread the manuscript. Both authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Author details

¹Department of Computer Science, University of Jeddah, Asfan Road, 285, Dhahban 23881, Jeddah, Saudi Arabia. ²Department of Computer Science, Middlesex College, Western University, 1151 Richmond Street, N6A 5B7, London, Ontario, Canada.

Received: 31 August 2016 Accepted: 26 July 2017

Published online: 24 August 2017

References

1. M Aharon, M Elad, A Bruckstein, K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans. Sig. Process.* **54**(11), 4311–4322 (2006). doi:10.1109/TSP.2006.881199
2. S Bacchelli, S Papi, Image denoising using principal component analysis in the wavelet domain. *J. Comput. Appl. Math.* **189**, 606–621 (2006). doi:10.1016/j.cam.2005.04.030. Proceedings of The 11th International Congress on Computational and Applied Mathematics
3. F Bashar, MR El-Sakka, in *Proceedings of the 11th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*. BM3D image denoising using learning-based adaptive hard thresholding, vol. 3:VISAPP, (Rome, 2016), pp. 204–214
4. EP Bennett, L McMillan, in *ACM SIGGRAPH 2005 Papers, ACM, New York, NY, USA, SIGGRAPH '05*. Video enhancement using per-pixel virtual exposures, (2005), pp. 845–852. doi:10.1145/1186822.1073272. http://doi.org/10.1145/1073204.1073272
5. J Boussinesq, *Théorie analytique de la chaleur* (Gauthier-Villars, Paris, 1903)
6. A Buades, B Coll, JM Morel, in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, vol 2*. A non-local algorithm for image denoising, (2005), pp. 60–65. doi:10.1109/CVPR.2005.38
7. C Chan, R Fulton, DD Feng, S Meikle, in *IEEE Nuclear Science Symposium Medical Imaging Conference*. Median non-local means filtering for low SNR image denoising: application to pet with anatomical knowledge, (2010), pp. 3613–3618. doi:10.1109/NSSMIC.2010.5874485
8. S Chang, B Yu, M Vetterli, Adaptive wavelet thresholding for image denoising and compression. *IEEE Trans. Image Process.* **9**(9), 1532–1546 (2000a). doi:10.1109/83.862633
9. S Chang, B Yu, M Vetterli, Spatially adaptive wavelet thresholding with context modeling for image denoising. *IEEE Trans. Image Process.* **9**(9), 1522–1531 (2000b). doi:10.1109/83.862630
10. YC Chen, VM Patel, PJ Phillips, R Chellappa, Dictionary-based face and person recognition from unconstrained video. *IEEE Access.* **3**, 1783–1798 (2015). doi:10.1109/ACCESS.2015.2485400
11. K Dabov, A Foi, V Katkovnik, K Egiazarian, Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Trans. Image Process.* **16**(8), 2080–2095 (2007). doi:10.1109/TIP.2007.901238
12. CA Deledalle, L Denis, F Tupin, Iterative weighted maximum likelihood denoising with probabilistic patch-based weights. *IEEE Trans. Image Process.* **18**(12), 2661–2672 (2009). doi:10.1109/TIP.2009.2029593
13. CA Deledalle, J Salmon, A Dalalyan, in *Proceedings of the British Machine Vision Conference, BMVA Press*, ed. by J Hoey, S McKenna, and E Trucco. Image denoising with patch based PCA: local versus global, (2011), pp. 25.1–25.10. doi:10.5244/C.25.25

14. D Donoho, De-noising by soft-thresholding. *IEEE Trans. Inf. Theory*. **41**(3), 613–627 (1995). doi:10.1109/18.382009
15. DL Donoho, IM Johnstone, Ideal spatial adaptation by wavelet shrinkage. *Biometrika*. **81**, 425–455 (1994)
16. F Durand, J Dorsey, Fast bilateral filtering for the display of high-dynamic-range images. *ACM Trans. Graph.* **21**(3), 257–266 (2002). doi:10.1145/566654.566574. <http://doi.org/10.1145/566654.566574>
17. AEP Enriquez, V Ponomaryov, in *2016 International Conference on Electronics, Communications and Computers (CONIELECOMP)*. Image denoising using block matching and discrete cosine transform with edge restoring, (2016), pp. 140–147. doi:10.1109/CONIELECOMP.2016.7438566
18. X Fei, W Huang, K Wang, Z Wei, in *Advances in Image and Graphics Technologies, Communications in Computer and Information Science*, vol. 363, ed. by T Tan, Q Ruan, X Chen, H Ma, and L Wang. Patch-based image denoising with geometric structure clustering (Springer, Berlin, Heidelberg, 2013), pp. 85–91. doi:10.1007/978-3-642-37149-3
19. Y Fu, A Lam, I Sato, Y Sato, in *2015 IEEE International Conference on Computer Vision (ICCV)*. Adaptive spatial-spectral dictionary learning for hyperspectral image denoising, (2015), pp. 343–351. doi:10.1109/ICCV.2015.47
20. L Genin, F Champagnat, GL Besnerais, L Coret, in *2011 18th IEEE International Conference on Image Processing*. Point object detection using a NL-means type filter, (2011), pp. 3533–3536. doi:10.1109/ICIP.2011.6116477
21. A Gersho, On the structure of vector quantizers. *IEEE Trans. Inf. Theory*. **28**(2), 157–166 (1982). doi:10.1109/TIT.1982.1056457
22. MM Ghazi, H Erdogan, Image noise level estimation based on higher-order statistics. *Multimedia Tools Appl.* **76**(2), 2379–2397 (2017). doi:10.1007/s11042-015-3169-1. <http://dx.doi.org/10.1007/s11042-015-3169-1>
23. A Haar, Zur Theorie der orthogonalen Funktionensysteme. *Math. Ann.* **69**(3), 331–371 (1910). doi:10.1007/bf01456326. <http://dx.doi.org/10.1007/bf01456326>
24. M Hasan, MR El-Sakka, *Structural similarity optimized Wiener filter: a way to fight image noise* (Springer International Publishing, Cham, 2015), pp. 60–68. doi:10.1007/978-3-319-20801-5. http://dx.doi.org/10.1007/978-3-319-20801-5_7
25. R Hedjam, RF Moghaddam, M Cheriet, in *2009 16th IEEE International Conference on Image Processing (ICIP)*. Markovian clustering for the non-local means image denoising, (2009), pp. 3877–3880. doi:10.1109/ICIP.2009.5414041
26. A Hore, D Ziou, in *Pattern Recognition (ICPR), 2010. 20th International Conference on*. Image quality metrics: Psnr vs. ssim, (2010), pp. 2366–2369. doi:10.1109/ICPR.2010.579
27. DA Huang, LW Kang, YCF Wang, CW Lin, Self-learning based image decomposition with applications to single image denoising. *IEEE Trans. Multimedia*. **16**(1), 83–93 (2014). doi:10.1109/TMM.2013.2284759
28. P Irrera, I Bloch, M Delplanque, A flexible patch based approach for combined denoising and contrast enhancement of digital X-ray images. *Med. Image Anal.* **28**, 33–45 (2016). doi:10.1016/j.media.2015.11.002. <http://www.sciencedirect.com/science/article/pii/S1361841515001681>
29. AK Jain, MN Murty, PJ Flynn, Data clustering: a review. *ACM Comput. Surv.* **31**(3), 264–323 (1999). doi:10.1145/331499.331504
30. W James, C Stein, in *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*. Estimation with quadratic loss (University of California Press, Berkeley, 1961), pp. 361–379. <http://projecteuclid.org/euclid.bsm/1200512173>
31. LW Kang, CY Hsu, HW Chen, CS Lu, CY Lin, SC Pei, Feature-based sparse representation for image similarity assessment. *IEEE Trans. Multimedia*. **13**(5), 1019–1030 (2011). doi:10.1109/TMM.2011.2159197
32. A Khan, MR El-Sakka, in *Proceedings of the 11th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*. Non-local means using adaptive weight thresholding, (2016), pp. 67–76. doi:10.5220/0005787100670076
33. M Kim, D Park, DK Han, H Ko, A novel approach for denoising and enhancement of extremely low-light video. *IEEE Trans. Consum. Electron.* **61**(1), 72–80 (2015). doi:10.1109/TCE.2015.7064113
34. T Kohonen, MR Schroeder, TS Huang, *Self-Organizing Maps*, 3rd edn. (Springer-Verlag New York, Inc., Secaucus, 2001)
35. R Lai, X Dou, in *Image and Signal Processing (CISP), 2010 3rd International Congress on*. vol 2. Improved non-local means filtering algorithm for image denoising, (2010), pp. 720–722. doi:10.1109/CISP.2010.5646842
36. JS Lee, Digital image smoothing and the sigma filter. *Comput. Vis. Graph. Image Proc.* **24**(2), 255–269 (1983). doi:10.1016/0734-189X(83)90047-6. <http://www.sciencedirect.com/science/article/pii/0734189X83900476>
37. C Liu, WT Freeman, R Szeliski, SB Kang, in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. vol 1. Noise estimation from a single image, (2006), pp. 901–908. doi:10.1109/CVPR.2006.207
38. YS Liu, PQ Yu, JH Yong, H Zhang, JG Sun, in *Proceedings of the First international conference on Computational and Information Science*. Bilateral filter for meshes using new predictor (Springer-Verlag, Berlin, Heidelberg, CIS'04, 2004), pp. 1093–1099. http://dx.doi.org/10.1007/978-3-540-30497-5_168
39. J MacQueen, in *Proc. Fifth Berkeley Symp. on Math. Statist. and Prob., Univ. of Calif. Press*. vol 1. Some methods for classification and analysis of multivariate observations, (1967), pp. 281–297
40. J Mairal, G Sapiro, M Elad, Learning multiscale sparse representations for image and video restoration. *Multiscale Model. Simul.* **7**(1), 214–241 (2008). doi:10.1137/070697653. <http://dx.doi.org/10.1137/070697653>
41. J Mairal, F Bach, J Ponce, G Sapiro, A Zisserman, in *Computer Vision, 2009 IEEE 12th International Conference on*. Non-local sparse models for image restoration, (2009), pp. 2272–2279. doi:10.1109/ICCV.2009.5459452
42. CD Manning, P Raghavan, H Schütze, *Introduction to Information Retrieval*. (Cambridge University Press, New York, 2008), p. 0521865719. <http://dl.acm.org/citation.cfm?id=1394399>
43. GM Maruf, MR El-Sakka, *Improved non-local means algorithm based on dimensionality reduction* (Springer International Publishing, Cham, 2015), pp. 43–50. doi:10.1007/978-3-319-20801-5
44. D Muresan, T Parks, in *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*. vol 1. Adaptive principal components and image denoising, (2003), pp. 1-101–4. doi:10.1109/ICIP.2003.1246908
45. BA Olshausen, DJ Field, Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*. **381**(6583), 607–609 (1996). doi:10.1038/381607a0
46. S Pal, R Mahakud, M Sahoo, PCA based image denoising using LPG. *IJCA Spec. Issue 2nd Natl. Conf Comput. Commun. Sens. Netw. (CCSN)*. **3**, 20–25 (2011). Published by Foundation of Computer Science, New York, USA
47. V Pappas, M Elad, Multi-scale patch-based image restoration. *IEEE Trans. Image Process.* **25**(1), 249–261 (2016). doi:10.1109/TIP.2015.2499698
48. P Perona, J Malik, Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern. Anal. Mach. Intell.* **12**(7), 629–639 (1990). doi:10.1109/34.56205
49. G Peyré, J Fadili, in *Proc. of Sampta'11*. Learning analysis sparsity priors, (2011). <http://hal.archives-ouvertes.fr/hal-00542016>
50. T Pham, L van Vliet, in *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*. Separable bilateral filtering for fast video preprocessing. pp 4 pp.–, (2005). doi:10.1109/ICME.2005.1521458
51. J Polzehl, V Spokoyny, Propagation-separation approach for local likelihood estimation. *Probab. Theory Related Fields*. **135**(3), 335–362 (2006). doi:10.1007/s00440-005-0464-1
52. J Polzehl, K Tabelow, Adaptive smoothing of digital images: the R package adimpro. *J. Stat. Softw.* (2007). <http://www.jstatsoft.org/article/view/v019i01>
53. O Rioul, M Vetterli, Wavelets and signal processing. *IEEE Signal Proc. Mag.* **8**(4), 14–38 (1991). doi:10.1109/79.91217
54. Z Shi, B Xu, X Zheng, M Zhao, A chinese character structure preserved denoising method for chinese tablet calligraphy document images based on ksvd dictionary learning. *Multimedia Tools Appl.* 1–16 (2016). doi:10.1007/s11042-016-4284-3. <http://dx.doi.org/10.1007/s11042-016-4284-3>
55. K Suwabe, M Onuki, Y Iizuka, Y Tanaka, in *2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. Globalized BM3D using fast eigenvalue filtering, (2015), pp. 438–442. doi:10.1109/GlobalSIP.2015.7418233
56. SC Tai, SM Yang, in *2008 3rd International Symposium on Communications, Control and Signal Processing*. A fast method for image noise estimation using Laplacian operator and adaptive edge detection, (2008), pp. 1077–1081. doi:10.1109/ISCCSP.2008.4537384
57. Y Tian, Z Wang, in *Fuzzy Systems and Knowledge Discovery (FSKD), 2013 10th International Conference on*. An adaptive orthogonal matching pursuit algorithm based on redundancy dictionary, (2013), pp. 578–582. doi:10.1109/FSKD.2013.6816263

58. C Tomasi, R Manduchi, in *Computer Vision, 1998. Sixth International Conference on*. Bilateral filtering for gray and color images, (1998), pp. 839–846. doi:10.1109/ICCV.1998.710815
59. P Tseng, Convergence of a block coordinate descent method for nondifferentiable minimization. *J. Optim. Theory Appl.* **109**(3), 475–494 (2001). doi:10.1023/A:1017501703105
60. Z Wang, A Bovik, H Sheikh, E Simoncelli, Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* **13**(4), 600–612 (2004). doi:10.1109/TIP.2003.819861
61. B Weiss, Fast median and bilateral filtering. *ACM Trans. Graph.* **25**(3), 519–526 (2006). doi:10.1145/1141911.1141918. <http://doi.acm.org/10.1145/1141911.1141918>
62. N Wiener, *Extrapolation, Interpolation, and Smoothing of Stationary Time Series* (The MIT Press, 1964). <http://dl.acm.org/citation.cfm?id=1097023>
63. Y Wu, B Tracey, P Natarajan, JP Noonan, James–Stein type center pixel weights for non-local means image denoising. *IEEE Signal Process. Lett.* **20**(4), 411–414 (2013). doi:10.1109/LSP.2013.2247755
64. B Xu, Y Cui, B Zuo, J Yang, J Song, Polarimetric SAR image filtering based on patch ordering and simultaneous sparse coding. *IEEE Trans. Geosci. Remote Sens.* **54**(7), 4079–4093 (2016). doi:10.1109/TGRS.2016.2536648
65. J Xu, J Hu, X Jia, A multistaged automatic restoration of noisy microscopy cell images. *IEEE Journal of Biomedical and Health Informatics.* **19**(1), 367–376 (2015). doi:10.1109/JBHI.2014.2305445
66. Y Zhan, M Ding, L Wu, X Zhang, Nonlocal means method using weight refining for despeckling of ultrasound images. *Signal Process.* **103**, 201–213 (2014). doi:10.1016/j.sigpro.2013.12.019. <http://www.sciencedirect.com/science/article/pii/S0165168413005070>. Image Restoration and Enhancement: Recent Advances and Applications
67. L Zhang, W Dong, D Zhang, G Shi, Two-stage image denoising by principal component analysis with local pixel grouping. *Pattern Recogn.* **43**(4), 1531–1549 (2010). doi:10.1016/j.patcog.2009.09.023
68. Y Zhang, J Liu, M Li, Z Guo, Joint image denoising using adaptive principal component analysis and self-similarity. *Inf. Sci.* **259**, 128–141 (2014). <http://dx.doi.org/10.1016/j.ins.2013.08.002>. <http://www.sciencedirect.com/science/article/pii/S0020025513005458>
69. H Zhong, J Zhang, G Liu, Robust polarimetric SAR despeckling based on nonlocal means and distributed lee filter. *IEEE Trans. Geosci. Remote Sens.* **52**(7), 4198–4210 (2014). doi:10.1109/TGRS.2013.2280278

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com
